LAB01 - Gates

Link to my GitHub repository

https://github.com/xruine00/Digital-electronics-1

De Morgan's laws

Equations for the implementation of gates:

$$f(c,b,a) = \overline{b}\,a + \overline{c}\,\overline{b}$$

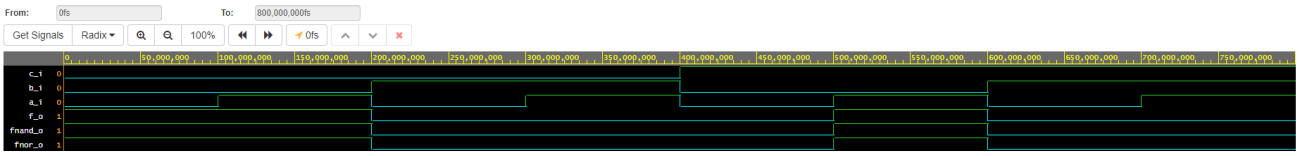
$$f(c,b,a)_{\mathrm{NAND}} = \overline{\overline{\overline{b}\,a} \cdot \overline{\overline{c}\,\overline{b}}}$$

$$f(c,b,a)_{\mathrm{NOR}} = \overline{b + \overline{a}} + \overline{c + b}$$

The truth table:

c	b	a	f(c,b,a)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

Screenshot with time waveforms:



Defined functions in VHDL:

```
library ieee;               -- Standard library
use ieee.std_logic_1164.all;-- Package for data types and logic operations

-------------------------------------------------------------------------
-- Entity declaration for basic gates
-------------------------------------------------------------------------
entity gates is
    port(
        a_i     : in  std_logic;         -- Data input
        b_i     : in  std_logic;         -- Data input
        c_i     : in  std_logic;         -- Data input
        f_o     : out std_logic;         -- output function
        fnand_o : out std_logic;         -- NAND output function
        fnor_o  : out std_logic           -- NOR output function
    );
end entity gates;

-------------------------------------------------------------------------
-- Architecture body for basic gates
-------------------------------------------------------------------------
architecture dataflow of gates is
begin
    f_o     <= ((not b_i) and a_i) or ((not c_i) and (not b_i));
    fnand_o <= ((b_i nand b_i) nand a_i) nand ((c_i nand c_i) nand (b_i nand
b_i));
    fnor_o  <= ((b_i nor (a_i nor a_i)) nor (c_i nor b_i)) nor ((b_i nor (a_i nor
a_i)) nor (c_i nor b_i));

end architecture dataflow;
```

Link to the EDA Playground:

https://www.edaplayground.com/x/ezmf

Distributive laws

Equations for the implementation of gates:

$$f1 = x \cdot y + x \cdot z$$
$$f2 = x \cdot (y + z)$$
$$f3 = (x + y) \cdot (x + z)$$
$$f4 = x + (y \cdot z)$$

$$f(1) = f(2) \rightarrow x \cdot y + x \cdot z = x \cdot (y + z)$$
$$f(3) = f(4) \rightarrow (x + y) \cdot (x + z) = x + (y \cdot z)$$

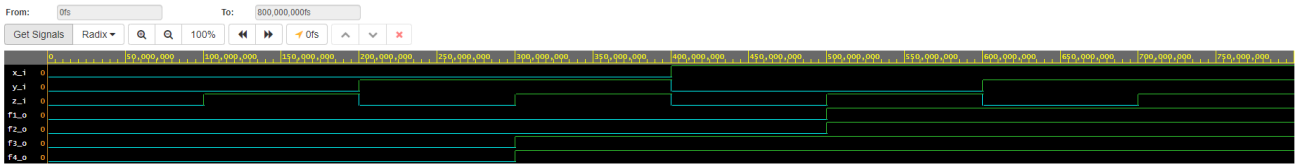
The truth table for f1 and f2:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

The truth table for f3 and f4:

x	y	z	f(x,y,z)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Screenshot with time waveforms:



Defined functions in VHDL:

```
library ieee;               -- Standard library
use ieee.std_logic_1164.all;-- Package for data types and logic operations

-------------------------------------------------------------------------
-- Entity declaration for basic gates
-------------------------------------------------------------------------
entity gates is
    port(
        x_i     : in  std_logic;         -- Data input
        y_i     : in  std_logic;         -- Data input
        z_i     : in  std_logic;         -- Data input
        f1_o    : out std_logic;         -- Function 1
        f2_o    : out std_logic;         -- Function 2
        f3_o    : out std_logic;         -- Function 3
        f4_o    : out std_logic         -- Function 4
    );
end entity gates;

-------------------------------------------------------------------------
-- Architecture body for basic gates
-------------------------------------------------------------------------
architecture dataflow of gates is
begin
    f1_o  <= (x_i and y_i) or (x_i and z_i);
    f2_o  <= x_i and (y_i or z_i);
    f3_o  <= (x_i or y_i) and (x_i or z_i);
    f4_o  <= x_i or (y_i and z_i);

end architecture dataflow;
```

Link to the EDA Playground:

https://www.edaplayground.com/x/GG_2