david3891 / Digital-electronics-1

MIT License

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Digital-electronics-1 / Labs / 01-gates / README.md

david3891 Update README.md

1 contributor

Raw Blame

149 lines (97 sloc) | 3.71 KB

Labs 01-gates

De'Morgans law

tabulka

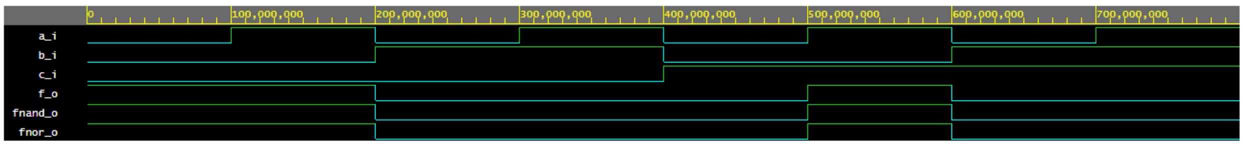
c	b	a	f(c,b,a)
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0

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

VHDL kód

```
------------------------------------------------------------------------
--
-- Example of basic OR, AND, XOR gates.
-- Nexys A7-50T, Vivado v2020.1, EDA Playground
--
-- Copyright (c) 2019-2020 Tomas Fryza
-- Dept. of Radio Electronics, Brno University of Technology, Czechia
-- This work is licensed under the terms of the MIT license.
--
------------------------------------------------------------------------

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;         -- OR output function
        fnand_o: out std_logic;         -- AND output function
        fnor_o : out std_logic          -- XOR 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 <= not( not ((not b_i) and a_i) and not((not c_i) and (not b_i)));
    fnor_o <= (not(b_i or (not a_i))) or (not(c_i or b_i));

end architecture dataflow;
```

obrázek



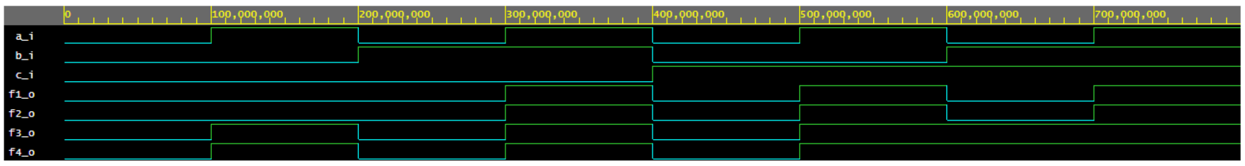
link na EDA Playground

(https://www.edaplayground.com/x/jVpY)

Distributive laws

```
------------------------------------------------------------------------
--
-- Example of basic OR, AND, XOR gates.
-- Nexys A7-50T, Vivado v2020.1, EDA Playground
--
-- Copyright (c) 2019-2020 Tomas Fryza
-- Dept. of Radio Electronics, Brno University of Technology, Czechia
-- This work is licensed under the terms of the MIT license.
--
------------------------------------------------------------------------

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;
        b_i             : in  std_logic;
        c_i             : in  std_logic;
        f1_o            : out std_logic;
        f2_o            : out std_logic;
        f3_o            : out std_logic;
        f4_o            : out std_logic
    );
end entity gates;

------------------------------------------------------------------------
-- Architecture body for basic gates
------------------------------------------------------------------------
architecture dataflow of gates is
begin


    f1_o <= (a_i and b_i) or (a_i and c_i);
    f2_o <= a_i and (b_i or c_i);
    f3_o <= (a_i or b_i) and (a_i or c_i);
    f4_o <= a_i or (b_i and c_i);


end architecture dataflow;
```

obrázek



link na EDA Playground

(https://www.edaplayground.com/x/pPR8)