README.md

Ivo Točený, 222683 - gates-01

1. part - Link to GitHub repository

My GitHub repository

2. part - Verification of De Morgan's laws

Equations from equation maker for more readability

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

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

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

Source code from design.vhd of Architecture

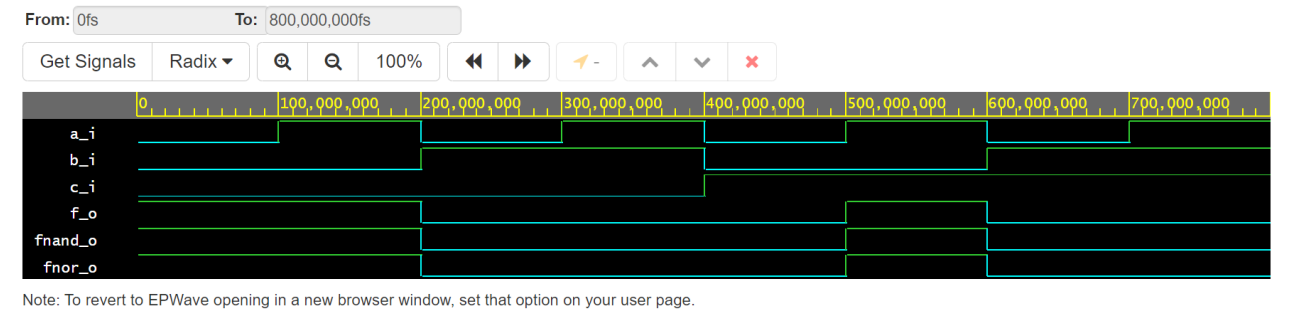
```
-------------------------------------------------------------------------
-- Architecture body for basic gates
-------------------------------------------------------------------------
-- Usage of De Morgan laws on function f using nands and nors
```

```
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;
```

Table of function values of set variables

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 simulated time waveforms of f, fnand, fnor to see that it matches



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Link to my EDA Playground of De Morgan's laws

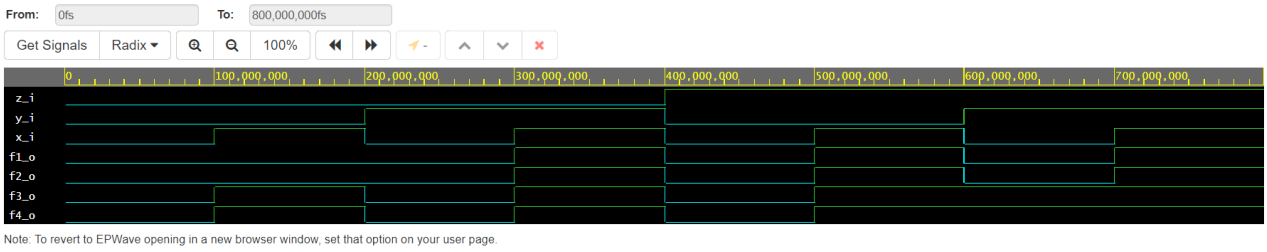
De Morgan's Laws EDA Playground

3. part - Verification of Distributive laws

Source code from design.vhd of Architecture

```
-------------------------------------------------------------------------
-- Architecture body for basic gates
-------------------------------------------------------------------------
-- Usage of De Morgan laws on function f using nands and nors
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;
```

Screenshot with simulated time waveforms to see if f1 = f2, f3 = f4, which does



Link to my EDA Playground of Distributed laws

Distributed laws