Digital-electronics-1

Link to my Digital-electronics-1 repository

Exercise 1.1: Verification of De Morgan's laws:

f(c,b,a) = ((not b) and a) or ((not c) and (not b))

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

### VHDL Code:

-- 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 -- 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));

end architecture dataflow;

Waveform #1:

EDA Playground example

Exercise 1.2: Verification of De Morgan's laws:

f(c,b,a)NAND = ((not b) nand a) nand ((not c) nand (not b))

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

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

### VHDL Code:

-- 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 -- Output function); end entity gates;

-- Architecture body for basic gates

architecture dataflow of gates is begin f_o <= ((not b_i) nand a_i) nand ((not c_i) nand (not b_i));

end architecture dataflow;

Waveform #2:

EDA Playground example

Exercise 1.3: Verification of De Morgan's laws:

f(c,b,a)NOR = ((not b) nor a) nor ((not c) nor (not b))

c	b	a	f(c,b,a)NOR
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

### VHDL Code:

-- 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 -- Output function); end entity gates;

```
-- Architecture body for basic gates

architecture dataflow of gates is begin f_o <= ((not b_i) nor a_i) nor ((not c_i) nor (not b_i));

end architecture dataflow;
```

Waveform #3:

EDA Playground example

Exercise 2.1: Verification of Distributive laws:

(a and b) or (a and c) = a and (b or c)

c	b	a	(a and b) or (a and c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

c	b	a	a and (b or c)
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

### VHDL Code:

```
-- 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; -- Output function f_m : out std_logic --
```

Output function); end entity gates;

-- Architecture body for basic gates

architecture dataflow of gates is begin f_o <= (a_i and b_i) or (a_i and c_i); f_m <= a_i and (b_i or c_i);

end architecture dataflow;

Waveform #4:

EDA Playground example

Exercise 2.2: Verification of Distributive laws:

(a or b) and (a or c) = a or (b and c)

c	b	a	(a or b) and (a or c)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

---

c	b	a	a or (b and c)
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

### VHDL Code:

-- 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; -- Output function f_m : out std_logic -- Output function ); end entity gates;

-- Architecture body for basic gates

architecture dataflow of gates is begin f_o <= (a_i or b_i) and (a_i or c_i); f_m <= a_i or (b_i and c_i);

end architecture dataflow;
```

Waveform #5:

EDA Playground example