Ivo-Toceny-222683 / **Digital-electronics-1**

Code Issues Pull requests Actions Projects Wiki Security

main Digital-electronics-1 / Labs / 02-logic /

Ivo-Toceny-222683 Updated README 02-logic waveform image ... 1 minute ago **History**

..

images 5 minutes ago

README.md 1 minute ago

README.md

Ivo Točený, 222683 - 02-logic

0. part - Link to GitHub repository

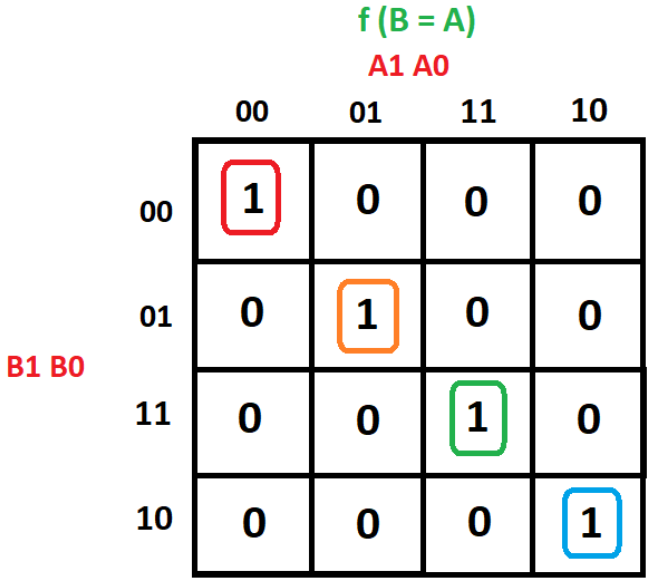
My GitHub 02-logic repository

1. part - Binary comparator truth table

Dec. equivalent	B[1:0]	A[1:0]	B is greater than A	B equals A	B is less than A
0	0 0	0 0	0	1	0
1	0 0	0 1	0	0	1
2	0 0	1 0	0	0	1
3	0 0	1 1	0	0	1
4	0 1	0 0	1	0	0
5	0 1	0 1	0	1	0

Dec. equivalent	B[1:0]	A[1:0]	B is greater than A	B equals A	B is less than A
6	0 1	1 0	0	0	1
7	0 1	1 1	0	0	1
8	1 0	0 0	1	0	0
9	1 0	0 1	1	0	0
10	1 0	1 0	0	1	0
11	1 0	1 1	0	0	1
12	1 1	0 0	1	0	0
13	1 1	0 1	1	0	0
14	1 1	1 0	1	0	0
15	1 1	1 1	0	1	0

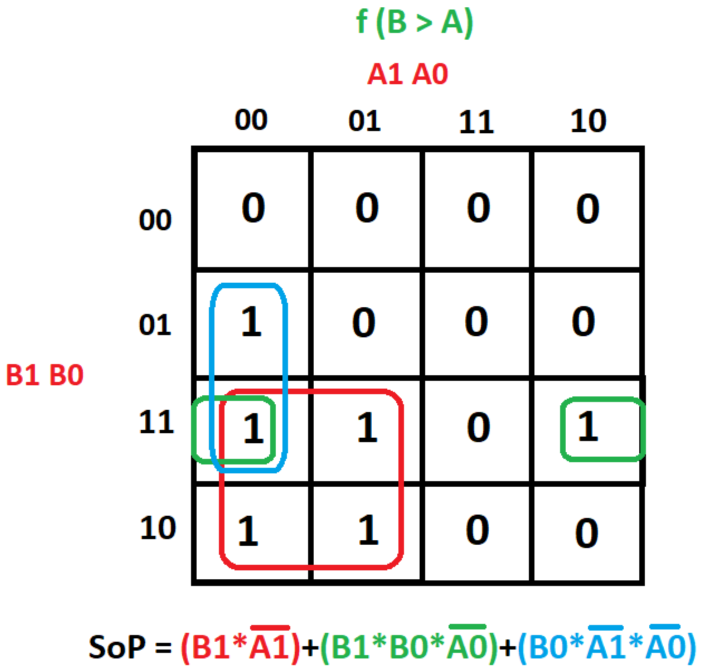
2. part - 2-bit comparator



f (B = A)

A1 A0

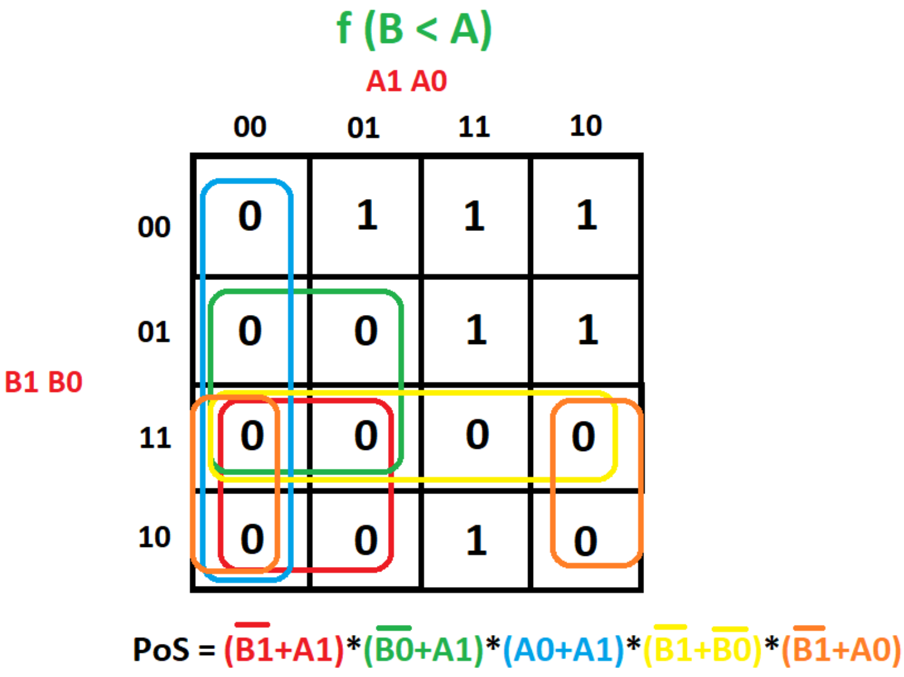
B1 B0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1



f (B > A)

B1 B0 \ A1 A0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$$SoP = (B1*\overline{A1})+(B1*B0*\overline{A0})+(B0*\overline{A1}*\overline{A0})$$



f (B < A)

B1 B0 \ A1 A0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$$PoS = (\overline{B1}+A1)*(\overline{B0}+A1)*(A0+A1)*(\overline{B1}+\overline{B0})*(\overline{B1}+A0)$$

3. part - 4-bit comparator

design.vhd

```
------------------------------------------------------------------------
-- Entity declaration for 4-bit binary comparator
------------------------------------------------------------------------
entity comparator_4bit is
```

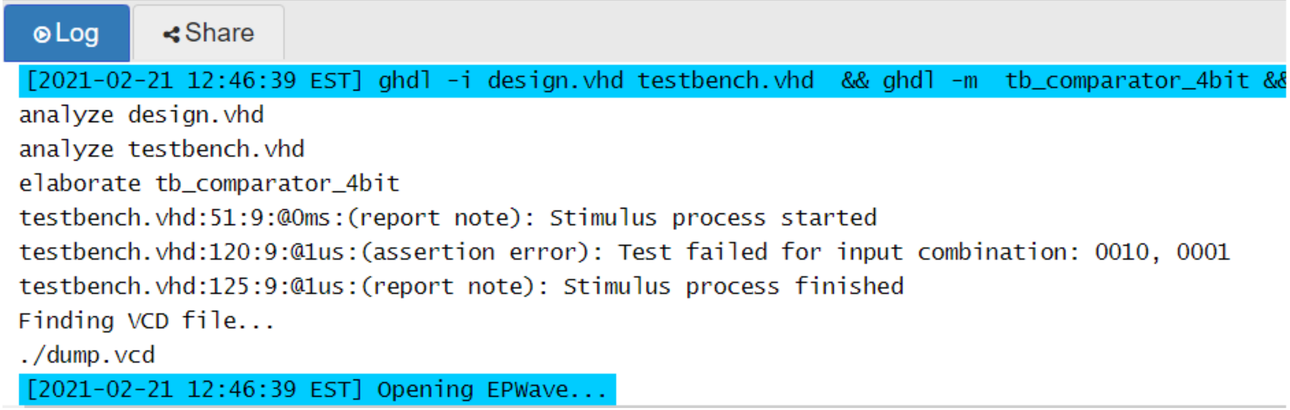
```
    port(
        a_i           : in  std_logic_vector(4 - 1 downto 0);
        b_i           : in  std_logic_vector(4 - 1 downto 0);


        -- COMPLETE ENTITY DECLARATION

                B_greater_A_o    : out std_logic;
                B_equals_A_o    : out std_logic;
        B_less_A_o    : out std_logic       -- B is less than A
    );
end entity comparator_4bit;

------------------------------------------------------------------------
-- Architecture body for 4-bit binary comparator
------------------------------------------------------------------------
architecture Behavioral of comparator_4bit is
begin
    B_less_A_o   <= '1' when (b_i < a_i) else '0';
    B_equals_A_o   <= '1' when (b_i = a_i) else '0';
    B_greater_A_o   <= '1' when (b_i > a_i) else '0';

end architecture Behavioral;
```

testbench.vhd

```
------------------------------------------------------------------------
-- Entity declaration for testbench
------------------------------------------------------------------------
entity tb_comparator_4bit is
    -- Entity of testbench is always empty
end entity tb_comparator_4bit;

------------------------------------------------------------------------
-- Architecture body for testbench
------------------------------------------------------------------------
architecture testbench of tb_comparator_4bit is

    -- Local signals
    signal s_a       : std_logic_vector(4 - 1 downto 0);
    signal s_b       : std_logic_vector(4 - 1 downto 0);
    signal s_B_greater_A : std_logic;
    signal s_B_equals_A  : std_logic;
    signal s_B_less_A    : std_logic;

begin
    -- Connecting testbench signals with comparator_4bit entity (Unit Under Test)
    uut_comparator_4bit : entity work.comparator_4bit
        port map(
            a_i           => s_a,
            b_i           => s_b,
            B_greater_A_o => s_B_greater_A,
            B_equals_A_o  => s_B_equals_A,
            B_less_A_o    => s_B_less_A
        );

    --------------------------------------------------------------------
    -- Data generation process
    --------------------------------------------------------------------
    p_stimulus : process
    begin
        -- Report a note at the begining of stimulus process
        report "Stimulus process started" severity note;


        -- First test values
        s_b <= "0000"; s_a <= "0000"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_A = '0'
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0000" severity error;

        -- Second test values
                s_b <= "0000"; s_a <= "0001"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A = '1'
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0001" severity error;

        -- Third test values
        s_b <= "0000"; s_a <= "0010"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A = '1'
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0010" severity error;

        -- 4th test values
        s_b <= "0000"; s_a <= "0011"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A = '1'
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0011" severity error;

        -- 5th test values
        s_b <= "0001"; s_a <= "0000"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'
        -- If false, then report an error
        report "Test failed for input combination: 0001, 0000" severity error;
```

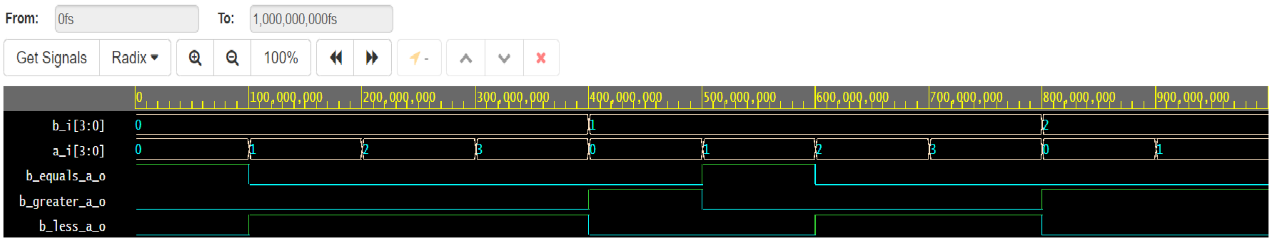
```
        -- 6th test values
        s_b <= "0001"; s_a <= "0001"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_A = '0'
        -- If false, then report an error
        report "Test failed for input combination: 0001, 0001" severity error;

        -- 7th test values
        s_b <= "0001"; s_a <= "0010"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A = '1'
        -- If false, then report an error
        report "Test failed for input combination: 0001, 0010" severity error;

        -- 8th test values
        s_b <= "0001"; s_a <= "0011"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A = '1'
        -- If false, then report an error
        report "Test failed for input combination: 0001, 0011" severity error;

        -- 9th test values
        s_b <= "0010"; s_a <= "0000"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'
        -- If false, then report an error
        report "Test failed for input combination: 0010, 0000" severity error;

        -- 10th test values - INTENTIONAL ERROR
        s_b <= "0010"; s_a <= "0001"; wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '1') and (s_B_less_A = '0'
        -- If false, then report an error
        report "Test failed for input combination: 0010, 0001" severity error;

        -- Report a note at the end of stimulus process
        report "Stimulus process finished" severity note;
        wait;
    end process p_stimulus;

end architecture testbench;
```

Screenshot of Console with successful intentional error message



```
[2021-02-21 12:46:39 EST] ghdl -i design.vhd testbench.vhd  && ghdl -m  tb_comparator_4bit &&
analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_4bit
testbench.vhd:51:9:@0ms:(report note): Stimulus process started
testbench.vhd:120:9:@1us:(assertion error): Test failed for input combination: 0010, 0001
testbench.vhd:125:9:@1us:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-21 12:46:39 EST] Opening EPWave...
```

Screenshot of simulated waveforms from EDA



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

Link to EDA playground of 4-bit comparator simulation

My EDA Playground sim