TaaviSalum / Digital-electronics-1

Test

MIT License

0 stars 0 forks

Star Unwatch

Code Issues Pull requests Actions Projects Wiki Security

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

TaaviSalum Small fix for exercise 3 ... 18 hours ago History

..

Pictures 19 hours ago

README.md 18 hours ago

README.md

Digital-electronics-1

Laboratory #2

Exercise 1: 2-bit comparator truth table

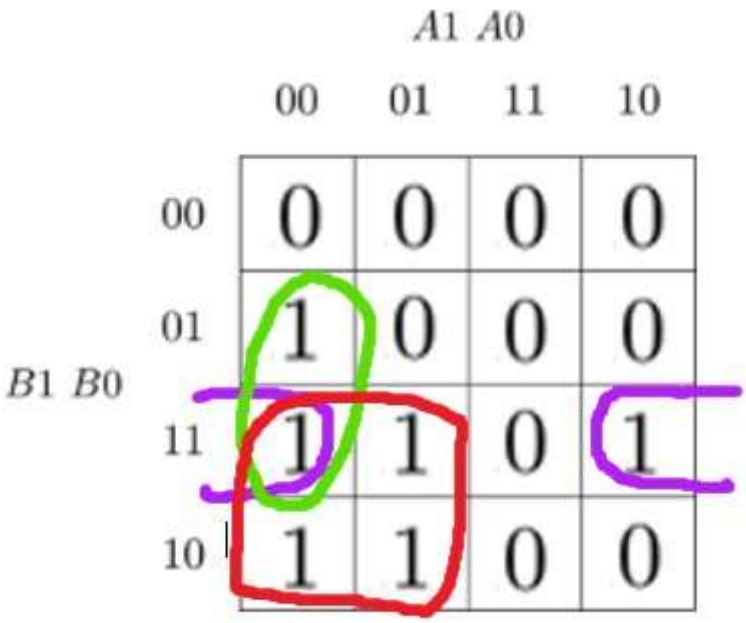
Dec.equivalent	B[1:0]	A[1:0]	B is greater than A	B equals A	B is less than A
0	0 0	0 0	0	1	0
1	0 0	0 1	0	0	1

Dec.equivalent	B[1:0]	A[1:0]	B is greater than A	B equals A	B is less than A
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
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

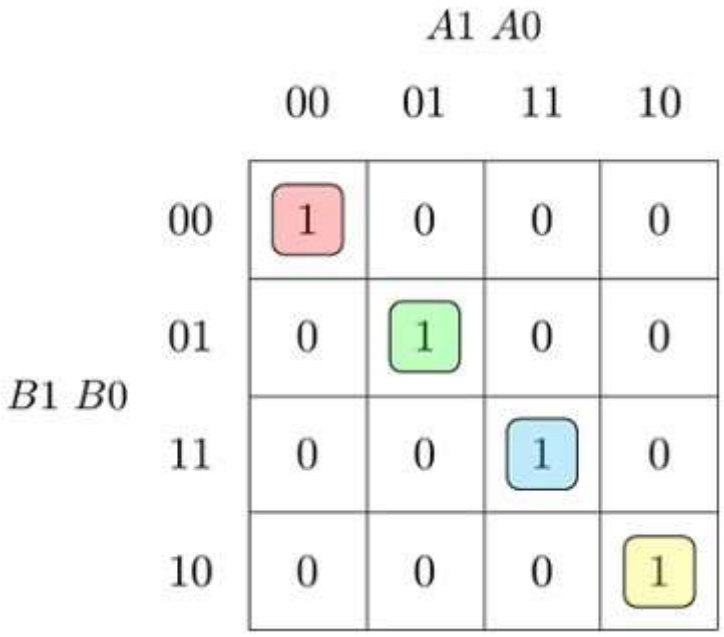
Exercise 2: 2-bit comparator

Karnaugh maps

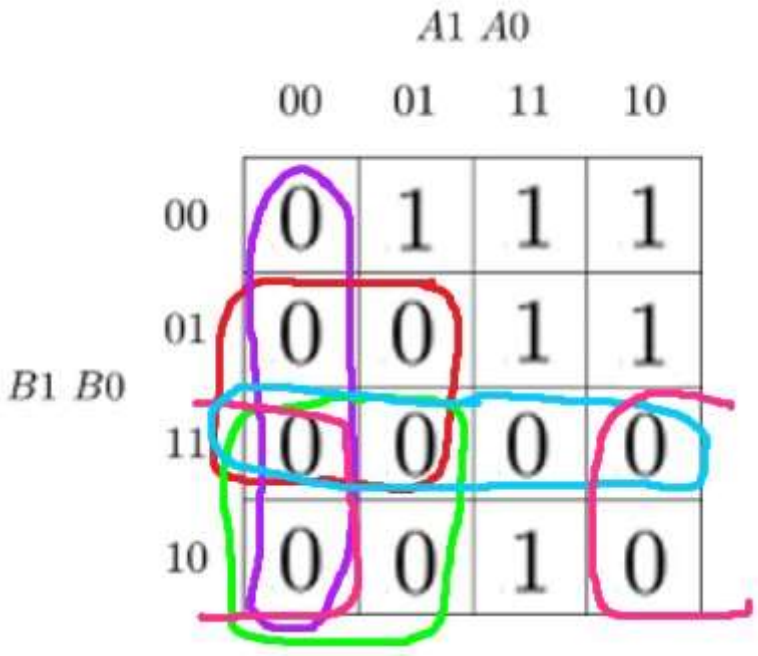
Greater than function Karnaugh map



Equal to function Karnaugh map



Less than function Karnaugh map



Simplified SoP form of the "greater than" function

$$Greater_{SoP} = (\bar{b_1} \cdot b_0 \cdot \bar{a_1} \cdot \bar{a_0}) + (b_1 \cdot \bar{b_0} \cdot \bar{a_1} \cdot \bar{a_0}) + (b_1 \cdot \bar{b_0} \cdot \bar{a_1} \cdot a_0) + (b_1 \cdot b_0 \cdot \bar{a_1} \cdot \bar{a_0}) + (b_1 \cdot b_0 \cdot \bar{a_1} \cdot a_0) + (b_1 \cdot b_0 \cdot a_1 \cdot \bar{a_0})$$

Simplified PoS form of the "less than" function

$$Less_{PoS} = (b_1 + b_0 + a_1 + a_0) \cdot (b_1 + \bar{b_0} + a_1 + a_0) \cdot (b_1 + \bar{b_0} + a_1 + \bar{a_0}) \cdot (\bar{b_1} + b_0 + a_1 + a_0) \cdot (\bar{b_1} + b_0 + a_1 + \bar{a_0}) \cdot (\bar{b_1} + b_0 + \bar{a_1} + a_0) \cdot (\bar{b_1} + \bar{b_0} + a_1 + a_0) \cdot (\bar{b_1} + \bar{b_0} + a_1 + \bar{a_0}) \cdot (\bar{b_1} + \bar{b_0} + \bar{a_1} + a_0) \cdot (\bar{b_1} + \bar{b_0} + \bar{a_1} + \bar{a_0}) \cdot$$

EDA playground example: https://www.edaplayground.com/x/8Qjr

Exercise 3: 4-bit binary comparator

VHDL design code:

```
library ieee;
use ieee.std_logic_1164.all;

------------------------------------------------------------------------
-- 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);
        B_greater_A_o : out std_logic;      -- B is greater than A
        B_equals_A_o  : out std_logic;      -- B is equal to A
        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_greater_A_o <= '1' when (b_i > a_i) else '0';
    B_equals_A_o  <= '1' when (b_i = a_i) else '0';
    B_less_A_o    <= '1' when (b_i < a_i) else '0';

end architecture Behavioral;
```

VHDL testbench code:

```
library ieee;
use ieee.std_logic_1164.all;

------------------------------------------------------------------------
-- 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;


        -- WRITE OTHER TESTS HERE

        -- Second 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;

        -- Third test values
        s_b <= "0011"; 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: 0011, 0000" severity error;

        -- Fourth test values
        s_b <= "0100"; s_a <= "1100"; 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: 0100, 1100" severity error;

        -- Fifth test values
        s_b <= "0000"; s_a <= "1000"; 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, 1000" severity error;

        -- Sixth test values
        s_b <= "1010"; s_a <= "1010"; 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: 1010, 1010" severity error;

        -- Seventh test values
        s_b <= "1100"; s_a <= "1000"; 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: 1100, 1000" severity error;

        -- Eight test values
        s_b <= "0001"; s_a <= "1001"; 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, 1001" severity error;

        -- Ninth test values
        s_b <= "0000"; s_a <= "1111"; 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, 1111" severity error;

        -- Tenth test values
        s_b <= "1111"; s_a <= "1111"; 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: 1111, 1111" 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;
```

Simulator console output

```
[2021-02-22 12:44:52 EST] ghdl -i design.vhd testbench.vhd  && ghdl -m
tb_comparator_4bit && ghdl -r  tb_comparator_4bit   --vcd=dump.vcd && sed -i
```

```
's/^U/X/g; s/^-/X/g; s/^H/1/g; s/^L/0/g' dump.vcd
analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_4bit
testbench.vhd:40:9:@0ms:(report note): Stimulus process started
testbench.vhd:98:9:@800ns:(assertion error): Test failed for input combination:
0001, 1001
testbench.vhd:117:9:@1us:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-22 12:44:53 EST] Opening EPWave...
Done
```

EDA playground example: https://www.edaplayground.com/x/6xjT