xmanas07 ... 3 minutes ago

..

images 25 minutes ago

README.md 3 minutes ago

README.md

Digital-electronics-1

úkol 1: Preparation tasks

Dec. equivalent	B[1:0]	A[1:0]	B > A	B = A	B < 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
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

Dec. equivalent	B[1:0]	A[1:0]	B > A	B = A	B < A
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

Canonical SoP and PoS

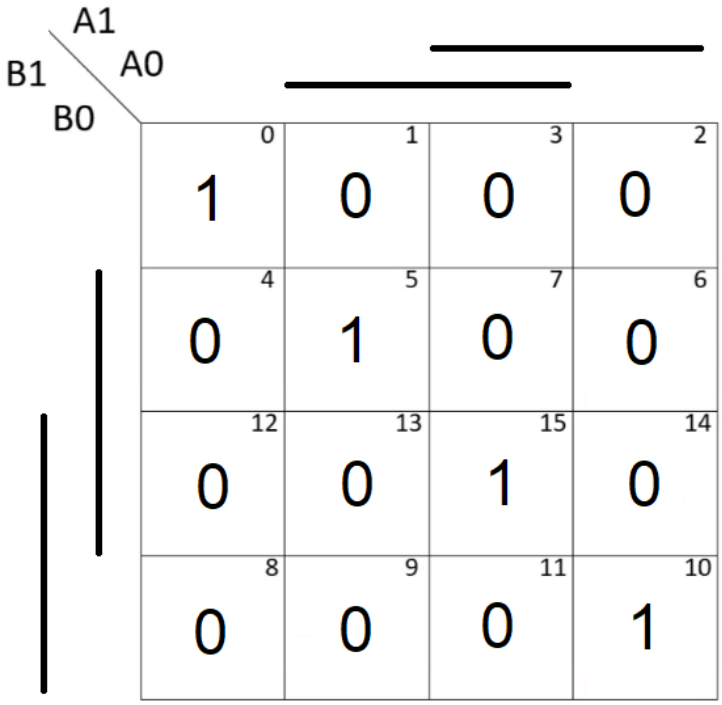
$$equals_{SoP}^{canon.} = (\overline{b_1} \cdot \overline{b_0} \cdot \overline{a_1} \cdot \overline{a_0}) + (\overline{b_1} \cdot b_0 \cdot \overline{a_1} \cdot a_0) + (b_1 \cdot \overline{b_0} \cdot a_1 \cdot \overline{a_0}) + (b_1 \cdot b_0 \cdot a_1 \cdot a_0)$$

$$less_{PoS}^{canon.} = (b_1 + b_0 + a_1 + a_0) \cdot (b_1 + \overline{b_0} + a_1 + a_0) \cdot (b_1 + \overline{b_0} + a_1 + \overline{a_0}) \cdot (\overline{b_1} + b_0 + a_1 + a_0) \cdot (\overline{b_1} + b_0 + a_1 + \overline{a_0}) \cdot$$
$$\cdot (\overline{b_1} + b_0 + \overline{a_1} + a_0) \cdot (\overline{b_1} + \overline{b_0} + a_1 + a_0) \cdot (\overline{b_1} + \overline{b_0} + a_1 + \overline{a_0}) \cdot (\overline{b_1} + \overline{b_0} + \overline{a_1} + a_0) \cdot (\overline{b_1} + \overline{b_0} + \overline{a_1} + \overline{a_0})$$

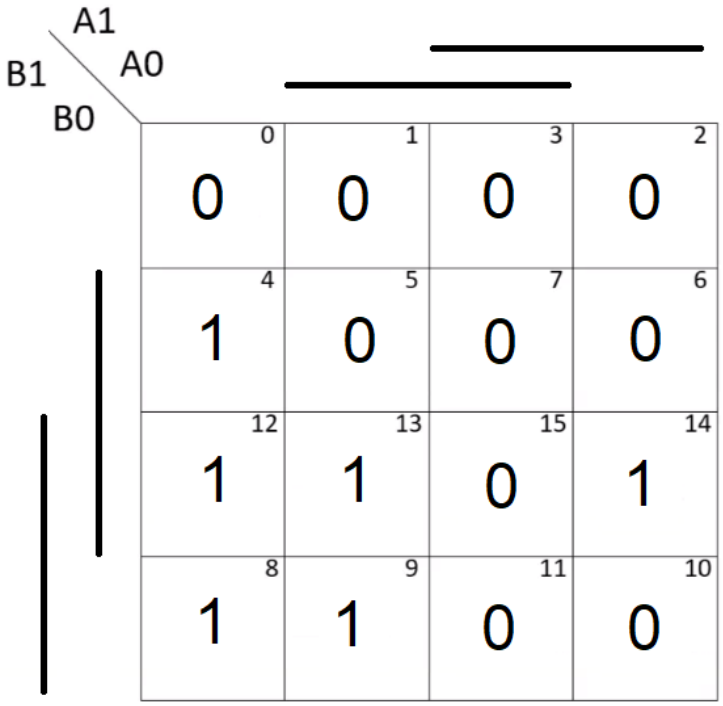
úkol 2: A 2-bit comparator

Karnaugh maps

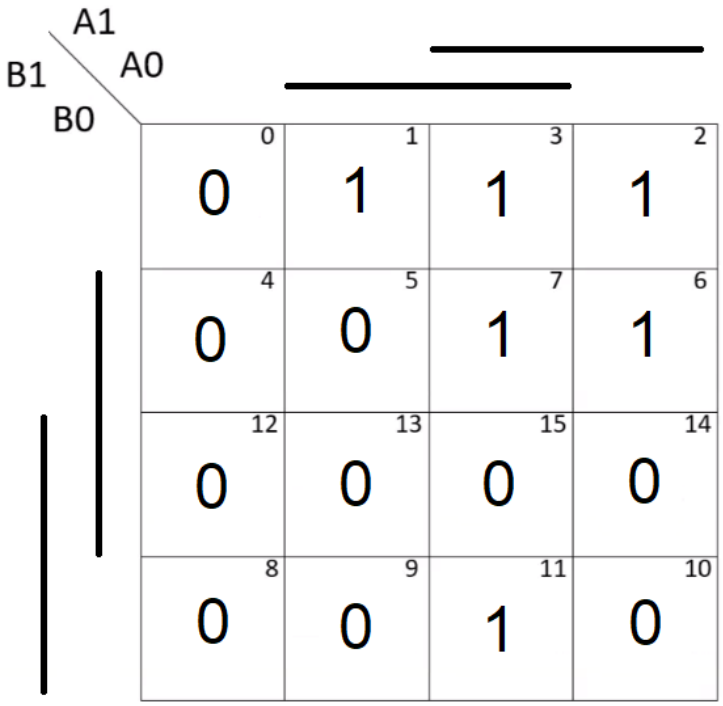
B equals A



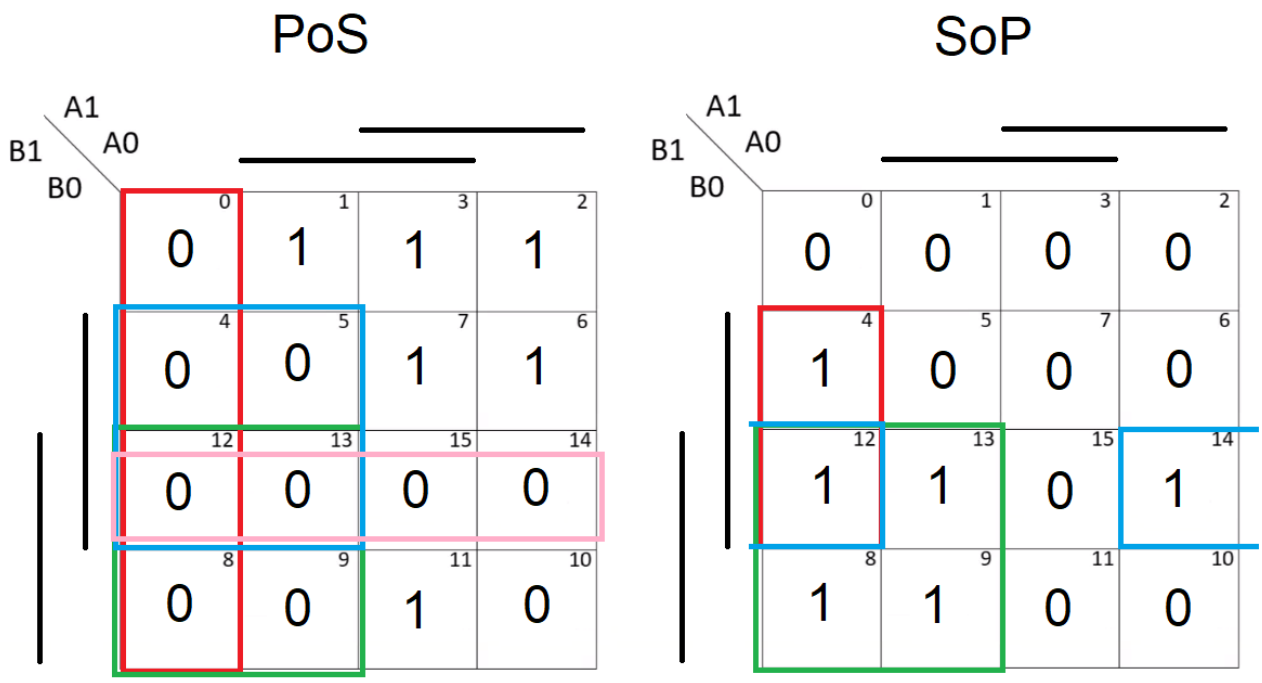
B is greater than A



B is less than A



SoP and PoS



$$greater_{SoP}^{min.} = (b_1 \cdot \overline{a_1}) + (b_1 \cdot b_0 \cdot \overline{a_0}) + (b_0 \cdot \overline{a_1} \cdot \overline{a_0})$$

$$less_{PoS}^{min.} = (\overline{b_1} + a_1) \cdot (\overline{b_0} + a_1) \cdot (\overline{b_1} + \overline{b_0}) \cdot (a_1 + a_0) \cdot (\overline{b_1} + a_0)$$

playground link

https://www.edaplayground.com/x/EaLP

úkol 3: 4-bit comparator

VHDL architecture

```
entity comparator_2bit is
    port(
        a_i           : in  std_logic_vector(4 - 1 downto 0);
        b_i           : in  std_logic_vector(4 - 1 downto 0);


        -- COMPLETE THE 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_2bit;

------------------------------------------------------------------------
-- Architecture body for 2-bit binary comparator
------------------------------------------------------------------------
architecture Behavioral of comparator_2bit is
begin
```

```
    B_less_A_o   <= '1' when (b_i < a_i) else '0';


    -- WRITE "GREATER" AND "EQUALS" ASSIGNMENTS HERE

    B_greater_A_o   <= '1' when (b_i > a_i) else '0';
    B_equals_A_o    <= '1' when (b_i = a_i) else '0';


end architecture Behavioral;
```

VHDL stimulus process (`design.vhd`)

```
    p_stimulus : process
    begin
        -- Report a note at the beginning of stimulus process
        report "Stimulus process started" severity note;


                s_b <= "0000"; s_a <= "0000"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_
                report "Test failed for input combination: 0000, 0000" severity error

                s_b <= "0000"; s_a <= "0001"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_
                report "Test failed for input combination: 0000, 0001" severity error

                s_b <= "0000"; s_a <= "0010"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0010" severity error

                s_b <= "0000"; s_a <= "0011"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0011" severity error

                s_b <= "0000"; s_a <= "0100"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0100" severity error

                s_b <= "0000"; s_a <= "0101"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0101" severity error

                s_b <= "0000"; s_a <= "0110"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0110" severity error

                s_b <= "0000"; s_a <= "0111"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_
                report "Test failed for input combination: 0000, 0111" severity error
```

.

```
        s_b <= "1111"; s_a <= "1000"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1000" severity error

        s_b <= "1111"; s_a <= "1001"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1001" severity error

        s_b <= "1111"; s_a <= "1010"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1010" severity error

        s_b <= "1111"; s_a <= "1011"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1011" severity error

        s_b <= "1111"; s_a <= "1100"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1100" severity error

        s_b <= "1111"; s_a <= "1101"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1101" severity error

        s_b <= "1111"; s_a <= "1110"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_
        report "Test failed for input combination: 1111, 1110" severity error

        s_b <= "1111"; s_a <= "1111"; wait for 100 ns;
        assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_
        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;
```

Simulation console output

```
[2021-02-23 07:06:10 EST] ghdl -i design.vhd testbench.vhd  && ghdl -m  tb_comparator_2bit && ghdl -r  tb_comparator_2bit   --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_2bit
testbench.vhd:51:9:@0ms:(report note): Stimulus process started
testbench.vhd:59:16:@200ns:(assertion error): Test failed for input combination: 0000, 0001
testbench.vhd:1079:9:@25600ns:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-23 07:06:14 EST] Opening EPWave...
Done
```

Eda playground link

https://www.edaplayground.com/x/9mQM