1. Preparation tasks

2-bit comparator truth table

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
10	1 0	1 0	0	1	0

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

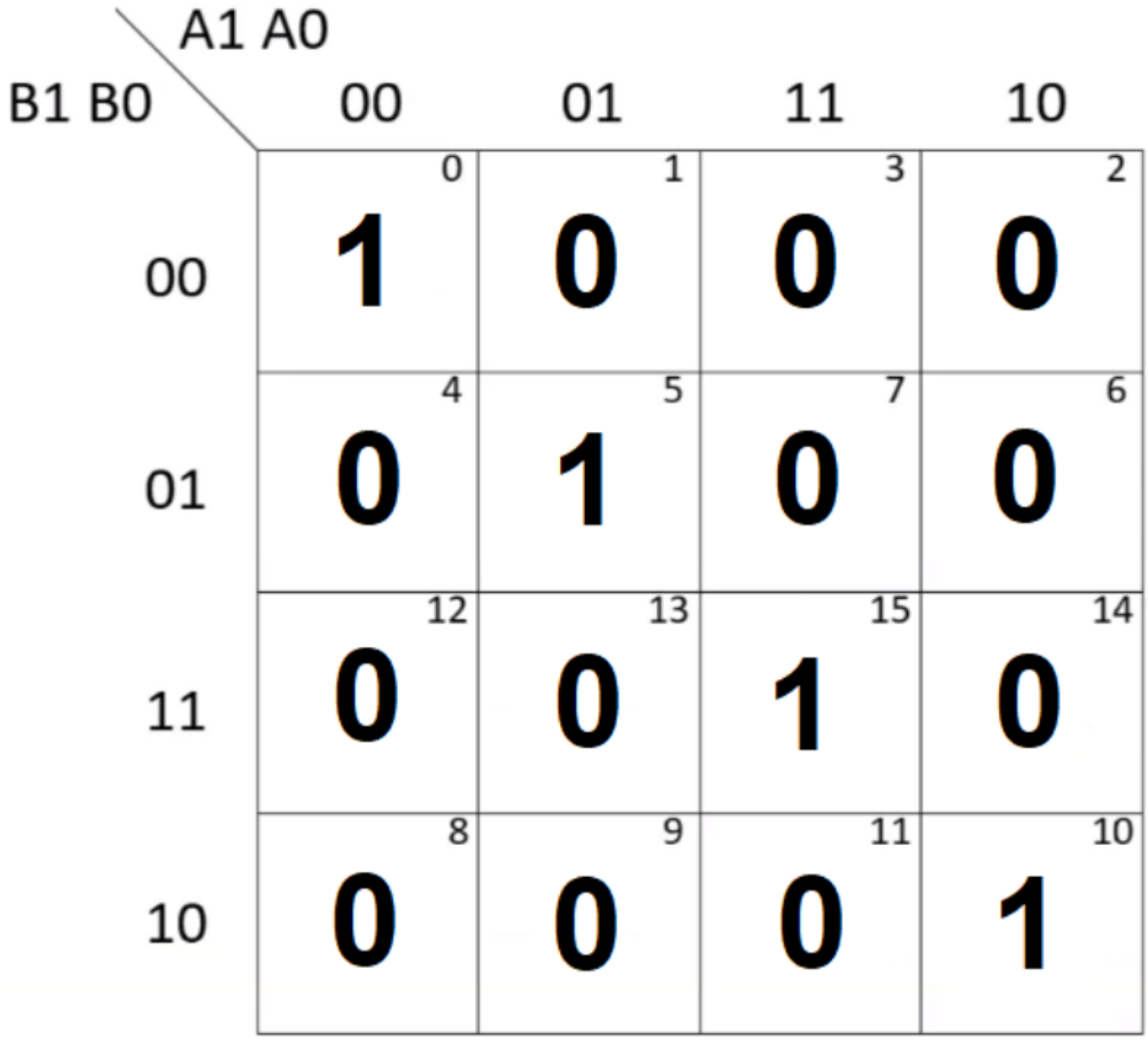
$$equals_{SoP}^{canon.} = (\overline{b_1} \cdot \overline{b_0} \cdot \overline{a_1} \cdot \overline{a_0}) + (\overline{b_1} \cdot b_0 \cdot \overline{a_1} \cdot a_0) + (b_1 \cdot \overline{b_0} \cdot a_1 \cdot \overline{a_0}) + (b_1 \cdot b_0 \cdot a_1 \cdot a_0)$$

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

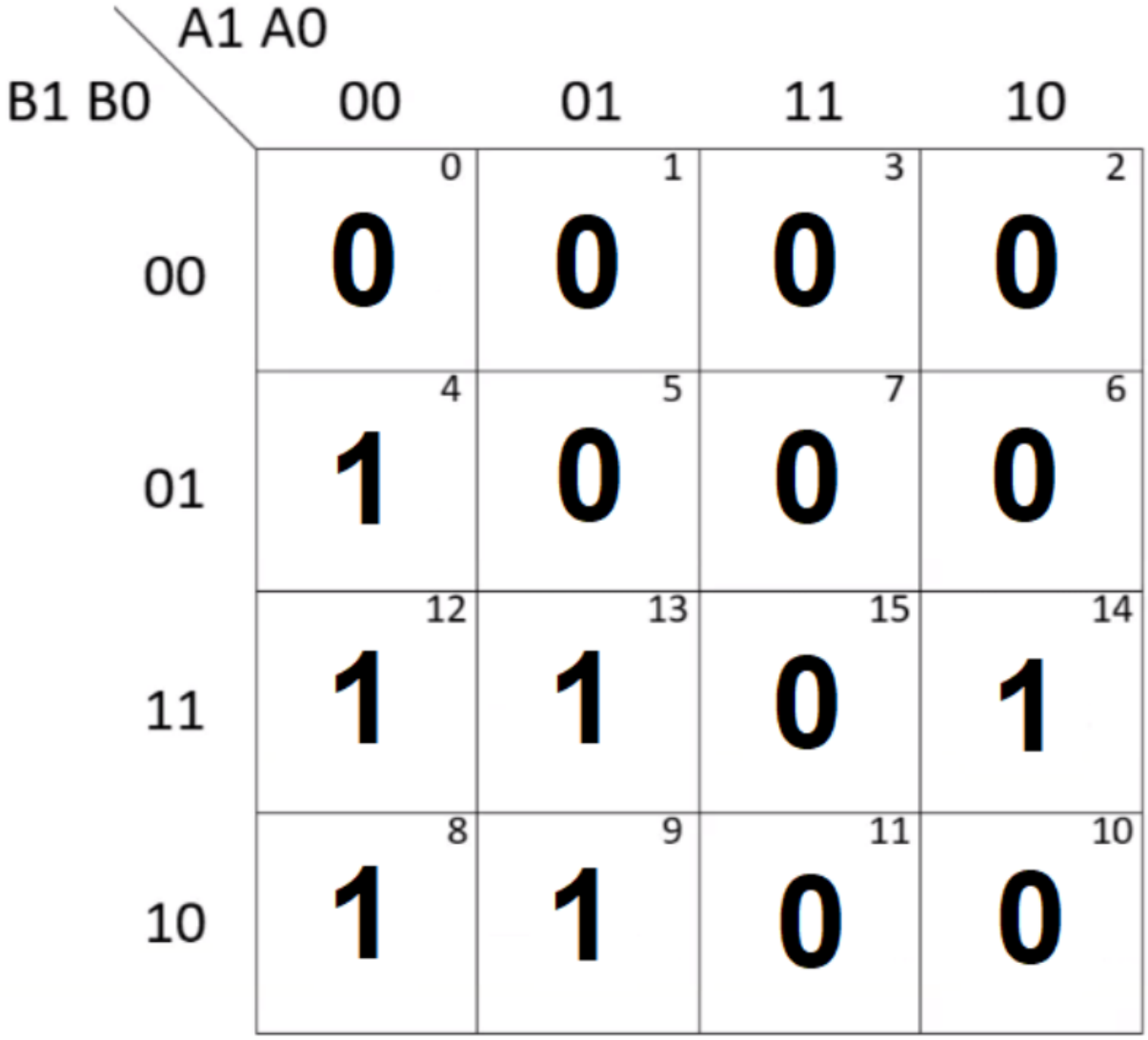
2. 2-bit comparator

Karnaugh maps

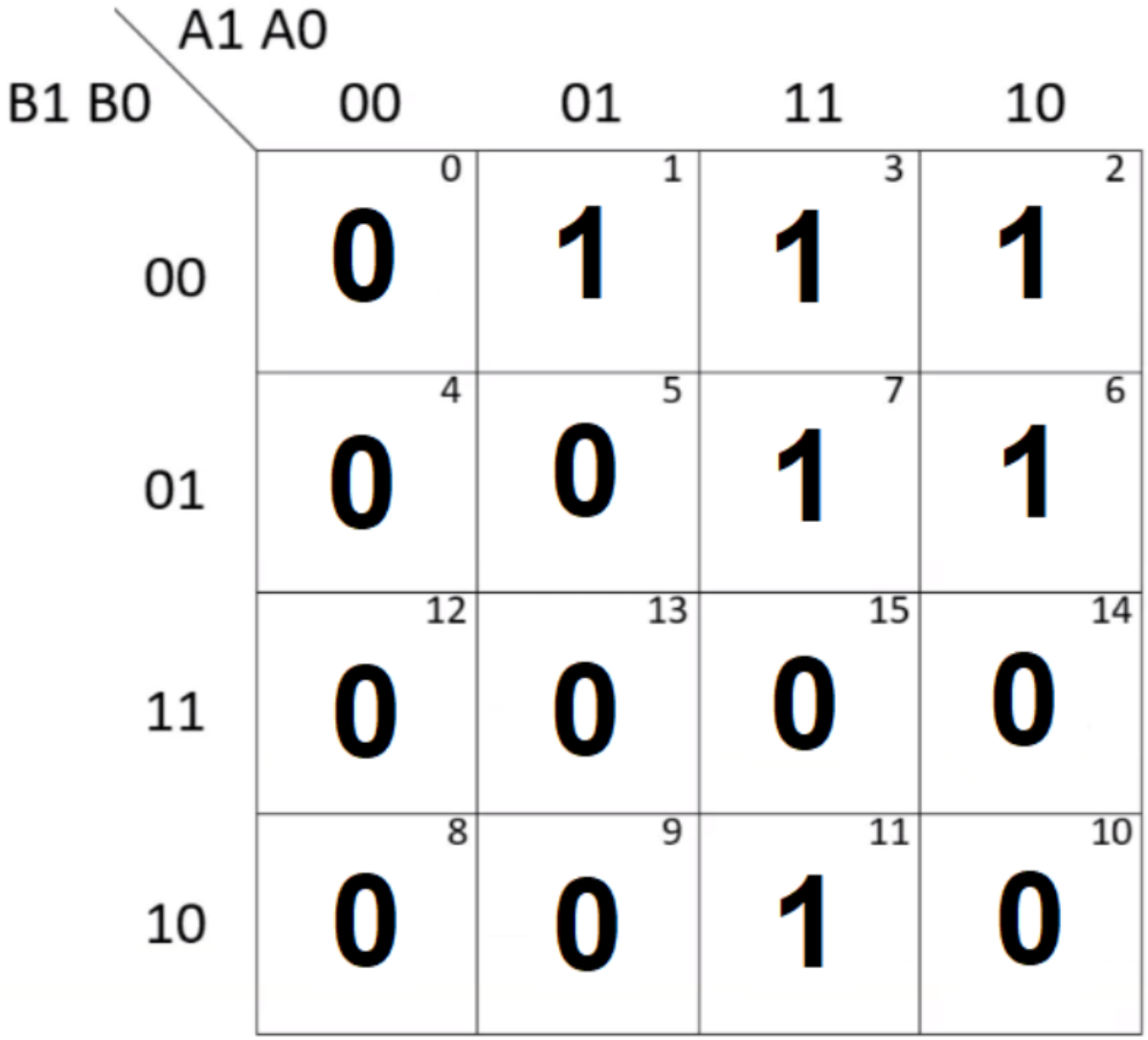
B equals A



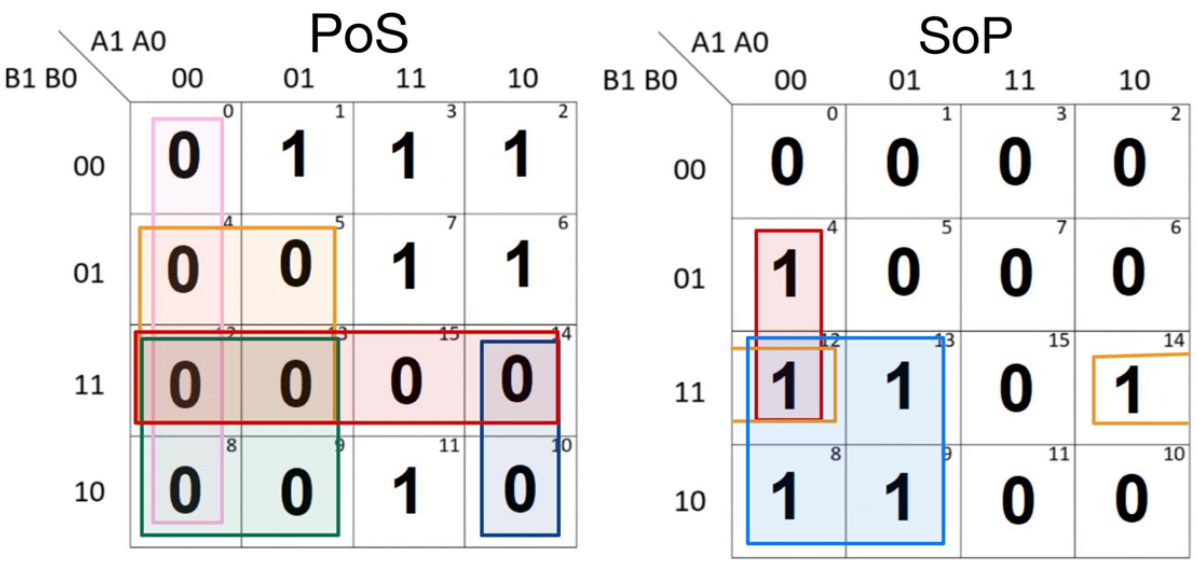
B is greater than A



B is less than A



PoS and SoP function



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

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

EDA Playground

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

3. 4-bit binary comparator

VHDL architecture (`design.vhd`)

```
architecture Behavioral of comparator_2bit is

begin
    B_less_A_o          <= '1' when (b_i < a_i) else '0';
    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 (`testbench.vhd`)

```
    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_l

        -- If false, then report an error
                report "Test failed for input combination: 0000, 0000" severity e

        -- WRITE OTHER TESTS HERE
                s_b <= "0000"; s_a <= "0001"; wait for 100 ns;
                assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_l
                report "Test failed for input combination: 0000, 0001" severity e

                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_l
                report "Test failed for input combination: 0000, 0010" severity e

                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_l
                report "Test failed for input combination: 0000, 0011" severity e

                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_l
                report "Test failed for input combination: 0000, 0100" severity e
```

```
            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_l
            report "Test failed for input combination: 0000, 0101" severity e

            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_l
            report "Test failed for input combination: 0000, 0110" severity e

            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_l
            report "Test failed for input combination: 0000, 0111" severity e

            s_b <= "0000"; s_a <= "1000"; wait for 100 ns;
            assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_l
            report "Test failed for input combination: 0000, 1000" severity e

            s_b <= "0000"; s_a <= "1001"; wait for 100 ns;
            assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_l
            report "Test failed for input combination: 0000, 1001" severity e

        -- chyba
            s_b <= "0000"; s_a <= "1010"; wait for 100 ns;
            assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_l
            report "Test failed for input combination: 0000, 1010" severity e

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

Simulator console output

```
[2021-02-23 10:20:02 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;
analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_2bit
testbench.vhd:51:9:@0ms:(report note): Stimulus process started
testbench.vhd:84:9:@600ns:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-23 10:20:02 EST] Opening EPWave...
Done
```

EDA Playground

https://www.edaplayground.com/x/J9UL