2-bit comparator

2-bit comparator truth table

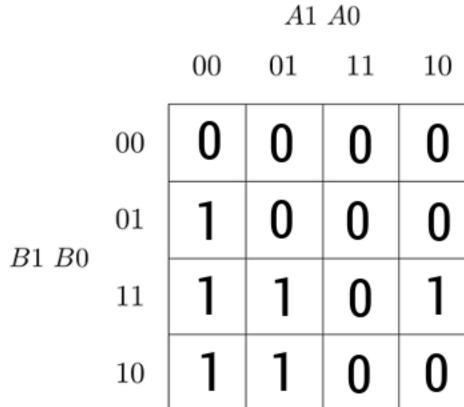
Dec. equivalent	B [1:0]	A [1:0]	B is greater than A	B equals A	B is less than A
0	0 0	0 0	0	1	0
1	0 0	0 1	0	0	1
2	0 0	1 0	0	0	1
3	0 0	11	0	0	1
4	0 1	0 0	1	0	0
5	0 1	0 1	0	1	0
6	0 1	1 0	0	0	1
7	0 1	1 1	0	0	1
8	1 0	0 0	1	0	0
9	1 0	0 1	1	0	0
10	1 0	1 0	0	1	0
11	1 0	1 1	0	0	1
12	1 1	0 0	1	0	0
13	1 1	0 1	1	0	0
14	1 1	1 0	1	0	0

0

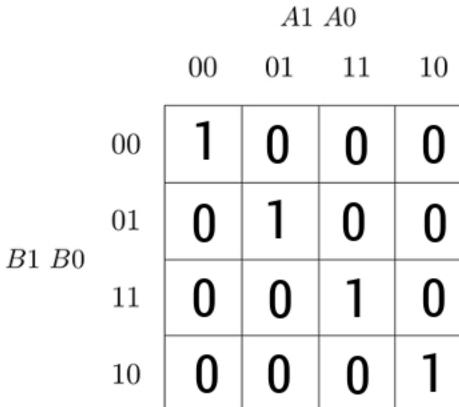
Karnaugh map - B is greater than A

1 1

15

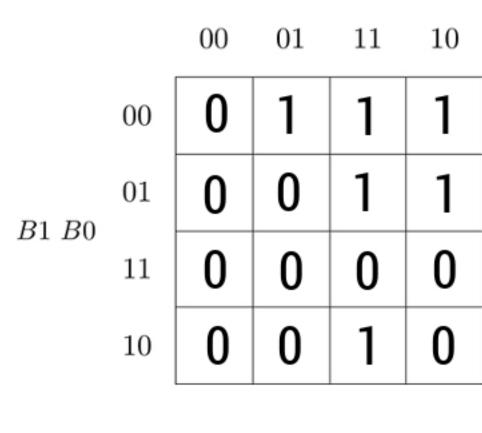


Karnaugh map - B equals A



A1 A0

Karnaugh map - B is less than A



 $greater_{SoP} = \overline{A1} \cdot B1 + \overline{A0} \cdot \overline{A1} \cdot B0 + \overline{A0} \cdot B0 \cdot B1$ $less_{PoS} = (A0 + A1) \cdot (\overline{B0} + A1) \cdot (\overline{B1} + A1) \cdot (\overline{B1} + A0) \cdot (\overline{B1} + \overline{B0})$

Equations of simplified SoP form of the "greater than" function and simplified PoS form of the "less than"

```
EDA playground link:
```

https://www.edaplayground.com/x/8UA4 4-bit comparator

entity comparator_4bit is

port(

function:

VHDL architecture from design file

```
: in std_logic_vector(4 - 1 downto 0);
                     : in std_logic_vector(4 - 1 downto 0);
         B_greater_A_o : out std_logic;
                                        -- B is greater than A
                                      -- B equals A
         B_equals_A_o : out std_logic;
         B_less_A_o : out std_logic
                                         -- B is less than A
     );
  end entity comparator_4bit;
  architecture Behavioral of comparator_4bit is
     B_{ess} = (b_i < a_i)  else '0';
     B_greater_A_o <= '1' when (b_i > a_i) else '0';
     B_{equals} = (b_i = a_i) else (0);
  end architecture Behavioral;
VHDL stimulus process from testbench file
  p_stimulus : process
     begin
```

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_A = '0'))
            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 = '0') and (s_B_less_A = '1'))
            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_A = '1'))
            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_A = '1'))
            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_A = '1'))
            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_A = '1'))
            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_A = '1'))
            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_A = '1'))
            report "Test failed for input combination: 0000, 0111" severity error;
            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_less_A = '1'))
            report "Test failed for input combination: 0000, 1000" severity error;
            s_b <= "0000"; s_a <= "1001"; wait for 100 ns;
            assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_A = '1'))
            report "Test failed for input combination: 0000, 1001" severity error;
   report "Stimulus process finished" severity note;
   wait;
end process p_stimulus;
```

Console output

```
analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_4bit
testbench.vhd:51:9:@Oms:(report note): Stimulus process started
testbench.vhd:100:16:@1us:(assertion error): Test failed for input combination: 0000, 1001
testbench.vhd:106:9:@1us:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-23 05:43:04 EST] Opening EPWave...
Done
```

EDA playground link: