

Masauso Lungu 209533

Lab 2: Combinational Logics

Links

[My github repository](#)

[My EDAPlayground](#)

Preparation tasks

Digital or Binary comparator compares the digital signals A, B presented at input terminal and produce outputs depending upon the condition of those inputs. Complete the truth table for 2-bit *Identity comparator* (B equals A), and two *Magnitude comparators* (B is greater than A, B is less than A). Note that, such a digital device has four inputs and three outputs/functions.

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
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

Karnaugh maps and simplified equations

Handwritten Karnaugh maps for a 2-bit comparator with inputs A_1, A_0 and B_1, B_0 .

Map 1: $B > A$

	$A_1 A_0$ 00	01	11	10
$B_1 B_0$ 00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

Map 2: $B < A$

	$A_1 A_0$ 00	01	11	10
$B_1 B_0$ 00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

Map 3: $B = A$

	$A_1 A_0$ 00	01	11	10
$B_1 B_0$ 00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

(ii) simplified equations

@ greater^{min}_{SOP} = $b_1 \bar{a}_1 + b_0 \bar{a}_1 \bar{a}_0 + b_1 b_0 \bar{a}_0$

(b) less^{min}_{POS} = $(a_1 + a_0) \cdot (\bar{b}_1 + \bar{b}_0) \cdot (\bar{b}_0 + a_1) \cdot (\bar{b}_1 + a_0) + (\bar{b}_1 + a_1)$

4-bit comparator design.vhd

```

-----
-- 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;

```

4-bit comparator stimulus process testbench.vhd

```

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

```

```

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

-- 2nd test values
s_b <= "0000"; s_a <= "0001"; wait for 100 ns;
-- Expected output
'1')) assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A =
-- If false, then report an error
report "Test failed for input combination: 0000, 0001" severity error;

-- 3rd test values
s_b <= "0001"; s_a <= "0000"; wait for 100 ns;
-- Expected output
'0')) assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A =
-- If false, then report an error
report "Test failed for input combination: 0001, 0000" severity error;

-- 4th test values
s_b <= "0001"; s_a <= "0001"; wait for 100 ns;
-- Expected output
'0')) assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_A =
-- If false, then report an error
report "Test failed for input combination: 0001, 0001" severity error;

-- 5th test values
s_b <= "0001"; s_a <= "0101"; wait for 100 ns;
-- Expected output
'1')) assert ((s_B_greater_A = '0') and (s_B_equals_A = '0') and (s_B_less_A =
-- If false, then report an error
report "Test failed for input combination: 0001, 0101" severity error;

-- 6th test values
s_b <= "0011"; s_a <= "0001"; wait for 100 ns;
-- Expected output
'0')) assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A =
-- If false, then report an error
report "Test failed for input combination: 0011, 0001" severity error;

-- 7th test values
s_b <= "0011"; s_a <= "0011"; wait for 100 ns;
-- Expected output
'0')) assert ((s_B_greater_A = '0') and (s_B_equals_A = '1') and (s_B_less_A =

```

```

-- If false, then report an error
report "Test failed for input combination: 0011, 0011" severity error;

-- 8th test values
s_b <= "0011"; s_a <= "0100"; 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, 0001" severity error;

-- 9th test values
s_b <= "1101"; 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: 1101, 1001" severity error;

-- 10th 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;

```

Error reported

```

analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_4bit
testbench.vhd:42:9:@0ms:(report note): Stimulus process started
testbench.vhd:112:9:@1us:(assertion error): Test failed for input combination: 1111, 1111
testbench.vhd:118:9:@1us:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-23 18:09:58 EST] Opening EPWave...
Done

```