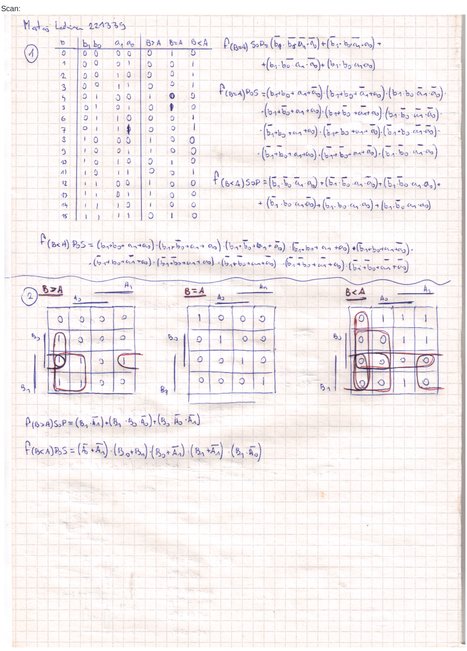
Digital-electronics-1 Matej Ledvina (221339)

Link to my github

02-logic

Preparation and 2bit comparator:

1. Scan:



2. EDA demonstration link

4bit comparator:

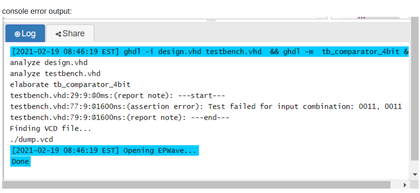
1. design.vhd:

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


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_less_A_o      : out std_logic;
        B_equals_A_o    : out std_logic;
        B_greater_A_o   : out std_logic
    );
end entity comparator_4bit;


architecture Behavioral of comparator_4bit is
begin
        B_less_A_o      <= '1' when (b_i < a_i) else '0';
    B_equals_A_o        <= '1' when (b_i = a_i) else '0';
    B_greater_A_o       <= '1' when (b_i > a_i) else '0';


end architecture Behavioral;
```

2. testbench.vhd:

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

entity tb_comparator_4bit is
end entity tb_comparator_4bit;

architecture testbench of tb_comparator_4bit is

    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
    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
        );


    p_stimulus : process
    begin

        report "---start---" 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 <= "0001"; s_a <= "0000"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0001, 0000" severity error;
        s_b <= "0001"; s_a <= "0001"; 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: 0001, 0001" severity error;
        s_b <= "0001"; 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: 0001, 0010" severity error;
        s_b <= "0001"; 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: 0001, 0011" severity error;
        s_b <= "0010"; s_a <= "0000"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0010, 0000" severity error;
        s_b <= "0010"; s_a <= "0001"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0010, 0001" severity error;
        s_b <= "0010"; s_a <= "0010"; 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: 0010, 0010" severity error;
        s_b <= "0010"; 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: 0010, 0011" severity error;
        s_b <= "0011"; s_a <= "0000"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0011, 0000" severity error;
        s_b <= "0011"; s_a <= "0001"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0011, 0001" severity error;
        s_b <= "0011"; s_a <= "0010"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '0') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0011, 0010" severity error;
        s_b <= "0011"; s_a <= "0011"; wait for 100 ns;
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '1') and (s_B_less_A = '0'))
        report "Test failed for input combination: 0011, 0011" severity error;
        report "---end---" severity note;
        wait;
    end process p_stimulus;

end architecture testbench;
```

3. console error output:



4. EDA demonstration link