david3891 / **Digital-electronics-1**

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

main

Digital-electronics-1 / Labs / 02-logic / **README.md**

david3891 Update README.md

1 contributor

Delete this file

Raw Blame

```
289 lines (226 sloc) | 9.91 KB
```

Labs 02-logic

1) Pravdivostní tabulka

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

Dec. equivalent	B[1:0]	A[1:0]	B is greater than A	B equals A	B is less than 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

2) Karnaughovy mapy

1) Karnaughova mapa pro funkci "rovno"

			A1,A0		
		0 0	***0 1***	***1 1***	***1 0***
	0 0	1	0	0	0
B1,B0	***0 1***	0	1	0	0
	1 1	0	0	1	0
	1 0	0	0	0	1

2) Karnaughova mapa pro funkci "větší než"

			A1,A0		
		0 0	***0 1***	***1 1***	***1 0***
	0 0	0	0	0	0
B1,B0	***0 1***	1	0	0	0
	1 1	1	1	0	1
	1 0	1	1	0	0

většíSoP = (B1./A1) + (/A0.B1.B0) + (/A1./A0.B0)

"/" -> negace

3) Karnaughova mapa pro funkci "menší než"

			A1,A0		
		0 0	***0 1***	***1 1***	***1 0***
	0 0	0	1	1	1
B1,B0	***0 1***	0	0	1	1
	1 1	0	0	0	0
	1 0	0	0	1	0

menšíPoS = (A1+/B1).(A1+/B0).(A1+A0).(/B1+/B0).(A0+/B1)

"/" -> negace

3) 4-bit komparátor

VHDL kód (design.vhd)

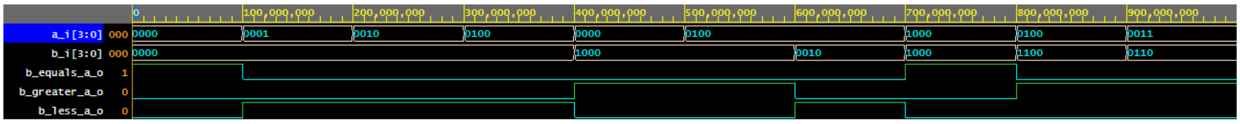
```
------------------------------------------------------------------------
--
-- Example of 2-bit binary comparator using the when/else assignment.
-- EDA Playground
--
-- Copyright (c) 2020-2021 Tomas Fryza
-- Dept. of Radio Electronics, Brno University of Technology, Czechia
-- This work is licensed under the terms of the MIT license.
--
------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

------------------------------------------------------------------------
-- Entity declaration for 2-bit binary comparator
------------------------------------------------------------------------
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);

        -- COMPLETE THE ENTITY DECLARATION

        B_greater_A_o : out std_logic;       -- B is greater than A
        B_equals_A_o  : out std_logic;       -- B equals A
        B_less_A_o    : out std_logic        -- B is less than A
    );
end entity comparator_4bit;
```

```vhdl
------------------------------------------------------------------------
-- Architecture body for 2-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;
```

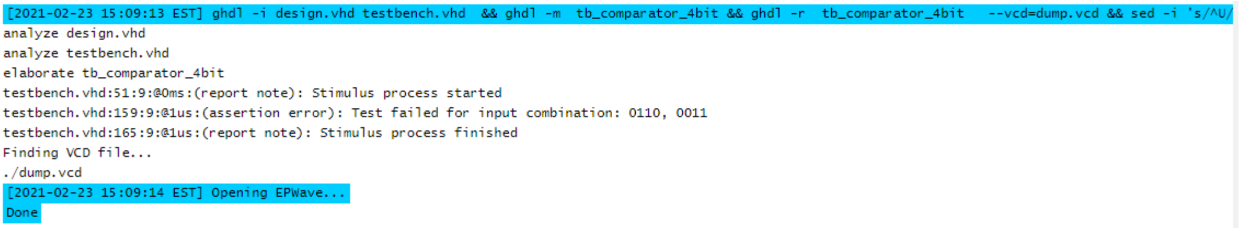
VHDL kód (testbench.vhd)

```vhdl
------------------------------------------------------------------------
--
-- Testbench for 2-bit binary comparator.
-- EDA Playground
--
-- Copyright (c) 2020-2021 Tomas Fryza
-- Dept. of Radio Electronics, Brno University of Technology, Czechia
-- This work is licensed under the terms of the MIT license.
--
------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;

------------------------------------------------------------------------
-- Entity declaration for testbench
------------------------------------------------------------------------
entity tb_comparator_4bit is
    -- Entity of testbench is always empty
end entity tb_comparator_4bit;

-----------------------------------------------------------------------
-- Architecture body for testbench
------------------------------------------------------------------------
architecture testbench of tb_comparator_4bit is

    -- Local signals
    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
    -- Connecting testbench signals with comparator_2bit entity (Unit Under Test)
    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
        );

    --------------------------------------------------------------------
    -- Data generation process
    --------------------------------------------------------------------
    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_less_A =
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0000" severity error;


        -- Second test values
        s_b <= "0000";
        s_a <= "0001";
        wait for 100 ns;
        -- Expected output
        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;


        -- Third test values
        s_b <= "0000";
        s_a <= "0010";
        wait for 100 ns;
        -- Expected output
        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, 0010" severity error;


        -- Fourth test values
        s_b <= "0000";
        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 =
        -- If false, then report an error
        report "Test failed for input combination: 0000, 0100" severity error;
        -- Fifth test values
        s_b <= "1000";
        s_a <= "0000";
        wait for 100 ns;
        -- Expected output
        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: 1000, 0000" severity error;


        -- Sixth test values
        s_b <= "1000";
        s_a <= "0100";
        wait for 100 ns;
        -- Expected output
        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: 1000, 0100" severity error;


        -- Seventh test values
        s_b <= "0010";
        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 =
        -- If false, then report an error
        report "Test failed for input combination: 0010, 0100" severity error;


        -- Eighth test values
        s_b <= "1000";
        s_a <= "1000";
        wait for 100 ns;
        -- Expected output
        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: 1000, 1000" severity error;


        -- Ninth test values
        s_b <= "1100";
        s_a <= "0100";
        wait for 100 ns;
        -- Expected output
        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: 1100, 0100" severity error;


        -- Tenth test values
        s_b <= "0110";
        s_a <= "0011";
        wait for 100 ns;
        -- Expected output
        assert ((s_B_greater_A = '1') and (s_B_equals_A = '1') and (s_B_less_A =
        -- If false, then report an error
        report "Test failed for input combination: 0110, 0011" severity error;


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

end architecture testbench;
```

Výstup ze simulace



Vypsaný error



```
[2021-02-23 15:09:13 EST] ghdl -i design.vhd testbench.vhd  && ghdl -m  tb_comparator_4bit && ghdl -r  tb_comparator_4bit   --vcd=dump.vcd && sed -i 's/^U/
analyze design.vhd
analyze testbench.vhd
elaborate tb_comparator_4bit
testbench.vhd:51:9:@0ms:(report note): Stimulus process started
testbench.vhd:159:9:@1us:(assertion error): Test failed for input combination: 0110, 0011
testbench.vhd:165:9:@1us:(report note): Stimulus process finished
Finding VCD file...
./dump.vcd
[2021-02-23 15:09:14 EST] Opening EPWave...
Done
```

Link na EDA Playground

(https://www.edaplayground.com/x/ADLQ)