

# Sprawozdanie z laboratorium

<b>Przedmiot</b>	Modelowanie i Analiza Systemów
<b>Temat laboratorium</b>	Generator sygnałów testujących i prosty układ kombinacyjny
<b>Numer laboratorium</b>	1
<b>Imię i nazwisko</b>	Maciej Stanek
<b>Numer indeksu</b>	122352
<b>Data wykonania</b>	9 marca 2018
<b>Data sprawozdania</b>	30 marca 2018

Listing 1. Implementacja generatora z zadania pierwszego wraz z pustą architekturą

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;

entity gen is port(
    clk: out std_logic;
    s1: out std_logic;
    s2: out std_logic;
    s3: out std_logic_vector(1 downto 0);
    s4: out integer);
end entity;

architecture default of gen is
    signal clkgen: std_logic := '0';
begin
    clkgen <= not clkgen after 20 ns;
    clk <= clkgen;

    s1 <= '1', '0' after 10 ns, '1' after 20 ns;
    s2 <= '1', '0' after 10 ns, '1' after 30 ns, '0' after 50 ns;
    s3 <= "01", "00" after 10 ns, "11" after 20 ns;
    s4 <= 0, 2 after 10 ns, 5 after 20 ns;
end architecture;

architecture empty of gen is
begin
end architecture;

```

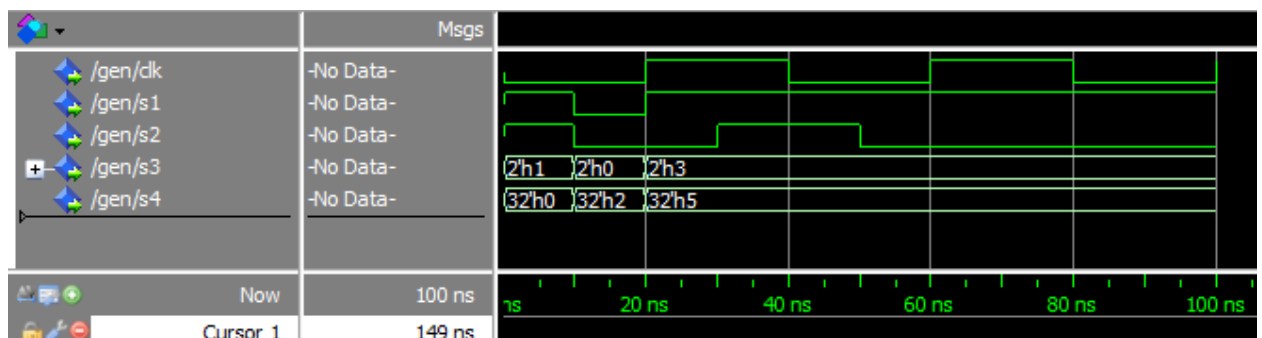
Listing 2. Skrypt testujący generator z zadania drugiego

```

vcom gen.vhd
vsim work.gen
foreach x [list clk s1 s2 s3 s4] {
    add wave -position insertpoint sim:/gen/$x
}

force -freeze sim:/gen/clk 0 0, 1 20 -r {40 ns}
force -freeze sim:/gen/s1 1 0, 0 10, 1 20
force -freeze sim:/gen/s2 1 0, 0 10, 1 30, 0 50
force -freeze sim:/gen/s3 01 0, 00 10, 11 20
force -freeze sim:/gen/s4 0 0, 2 10, 5 20
run 100 ns

```



Rysunek 1. Wyniki symulacji generatorów z zadania pierwszego i drugiego.

Rysunek 2. Tablica Karnaugh dla wyjść EQ i GE

$$ge = (A_1 + A_0 + \bar{B}_0)(A_1 + \bar{B}_1)(A_0 + \bar{B}_1 + \bar{B}_0)$$

$$eq = \bar{A}_1 \bar{A}_0 \bar{B}_1 \bar{B}_0 + \bar{A}_1 A_0 \bar{B}_1 B_0 +$$

$$+ A_1 A_0 \bar{B}_1 B_0 + A_1 \bar{A}_0 \bar{B}_1 \bar{B}_0$$

Rysunek 3. Równania wyjść EQ i GE wyznaczone na podstawie tablicy Karnaugh

Listing 3. Implementacja sumatora w architekturach behawioralnej i strukturalnej

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;

entity uklad_kombinacyjny_porownujacy_dwie_liczby_dwubitowe is port(
    a:    in std_logic_vector(1 downto 0);
    b:    in std_logic_vector(1 downto 0);
    eq: out std_logic;
    ge: out std_logic);
end entity;

architecture behavioral of uklad_kombinacyjny_porownujacy_dwie_liczby_dwubitowe is
begin
    compare: process(a, b)
    begin
        if a >= b then
            ge <= '1';
        else
            ge <= '0';
        end if;
        if a = b then
            eq <= '1';
        else
            eq <= '0';
        end if;
    end process;
end architecture;

architecture structural of uklad_kombinacyjny_porownujacy_dwie_liczby_dwubitowe is
    signal ge0, ge1, ge2: std_logic;
    signal eq0, eq1, eq2, eq3: std_logic;
begin
    ge0 <= a(1) or a(0) or not b(0);
    ge1 <= a(1) or not b(1);
    ge2 <= a(0) or not b(1) or not b(0);
    ge <= ge0 and ge1 and ge2;

    eq0 <= not a(1) and not a(0) and not b(1) and not b(0);
    eq1 <= not a(1) and a(0) and not b(1) and b(0);
    eq2 <= a(1) and a(0) and b(1) and b(0);
    eq3 <= a(1) and not a(0) and b(1) and not b(0);
    eq <= eq0 or eq1 or eq2 or eq3;
end architecture;

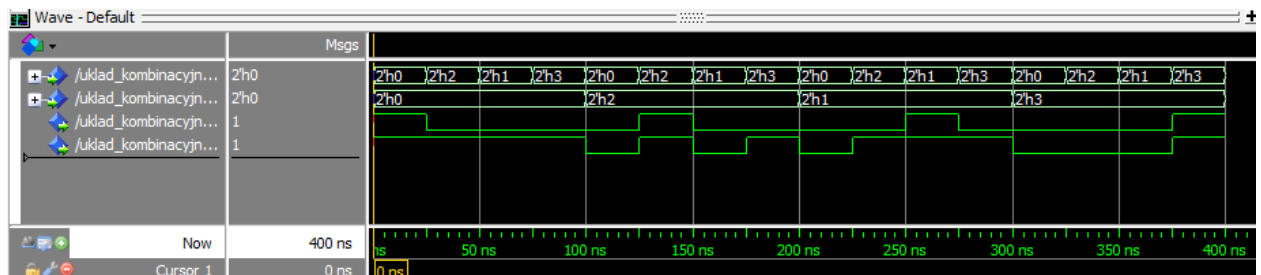
```

Listing 4. Skrypt testujący sumator z zadania trzeciego

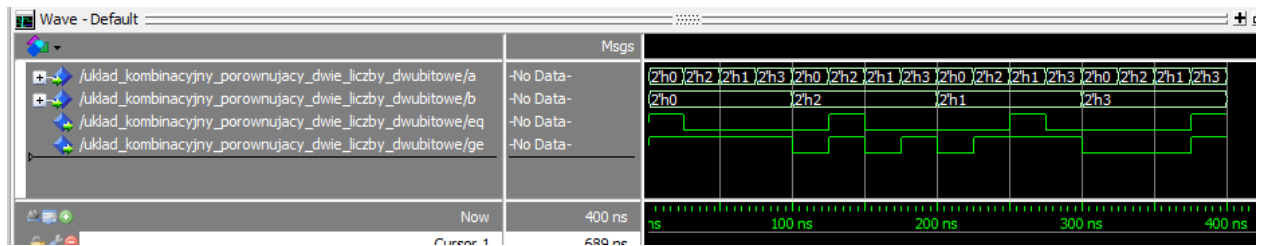
```

set name uk lad_kombinacyjny_porownujacy_dwie_liczby_dwubitowe
#set arch behavioral
set arch structural
vcom $name.vhd
vsim "work.$name\($arch\)"
foreach x [list a b eq ge] {
    add wave -position insertpoint sim:/$name/$x
}
set T 25
foreach sig [list a(1) a(0) b(1) b(0)] {
    set 2T [expr 2*$T]
    force -freeze sim:/$name/$sig 0 0, 1 $T -r "$2T ns"
    set T $2T
}
run $2T ns

```



Rysunek 4. Wynik działania architektury strukturalnej



Rysunek 5. Wynik działania architektury behawioralnej