

# Sprawozdanie z laboratorium

<b>Przedmiot</b>	Modelowanie i Analiza Systemów
<b>Temat laboratorium</b>	Model idealnego przetwornika A/C typu Sigma-Delta
<b>Numer laboratorium</b>	4
<b>Imię i nazwisko</b>	Maciej Stanek
<b>Numer indeksu</b>	122352
<b>Data wykonania</b>	17 maja 2018
<b>Data sprawozdania</b>	3 czerwca 2018

**Zadanie 1:** Zaprojektuj i zaimplementuj w VHDL-AMS modele idealnych przetworników sigma-delta I i II rzędu. Do wyjścia modulatora podłącz model filtra decymacyjnego, który został opracowany w poprzednich zadaniach. Elementy układu (entity) powinny mieć porty typu **terminal**. Integratory zaimplementuj wykorzystując opis w dziedzinie Z.

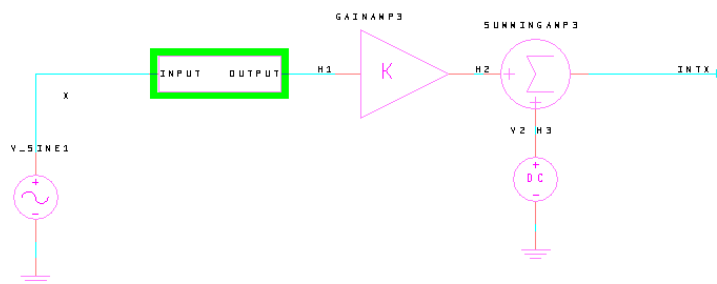
Zgodnie z zaleceniem prowadzącego, implementacji modulatora dokonano w dziedzinie ciągłej. Wymagało to przygotowania układu całkującego działającego na obiektach typu **terminal**.

Listing 1. Układ całkujący.

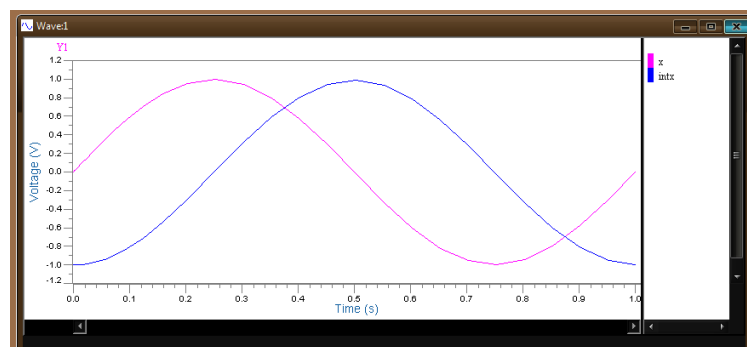
```
library ieee;
use ieee.std_logic_1164.all;
use ieee.electrical_systems.all;
use ieee.math_real.all;

entity analog_int is
  port(
    terminal input : in electrical;
    terminal output : out electrical);
end entity;

architecture default of analog_int is
  quantity v1 across input;
  quantity v2 across i2 through output;
begin
  v2 == v1'integ;
end architecture;
```

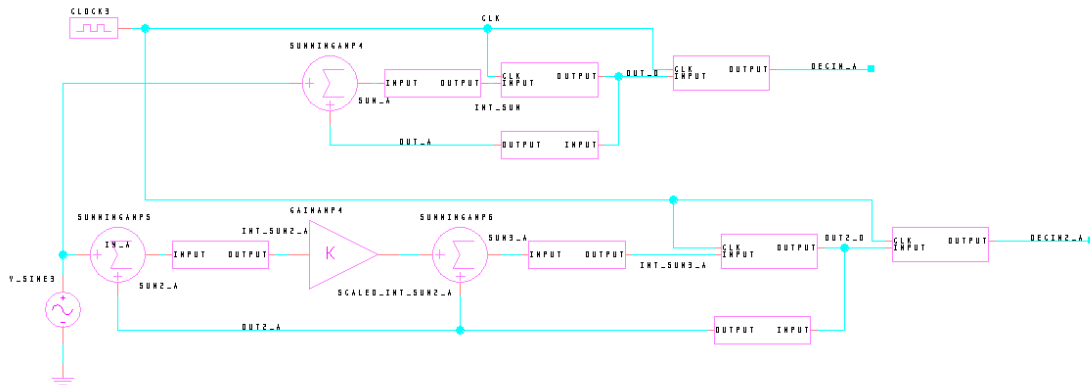


Rysunek 1. Układ testujący moduł całkujący.



Rysunek 2. Test modułu całkującego.

Następnie wykonano modulatory w dwóch żądanych konfiguracjach. Wymagało to modyfikacji kody decymatora, który został przygotowany na minionych zajęciach.



Rysunek 3. Modulatory sigma-delta z dołączonymi decymatorami. Od góry kolejno układy pierwszego i drugiego rzędu.

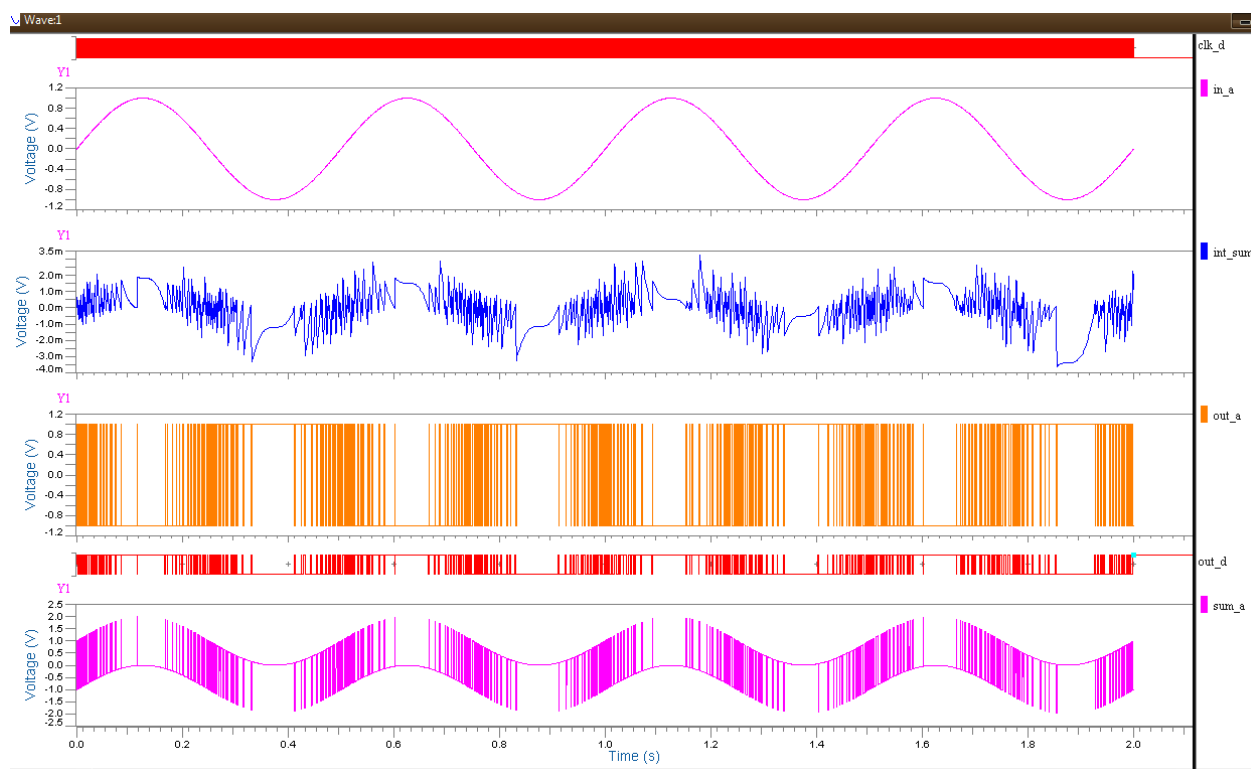
Listing 2. Układ decymacyjny: zmodyfikowany kod źródłowy.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
use ieee.electrical_systems.all;
use ieee.math_real.all;

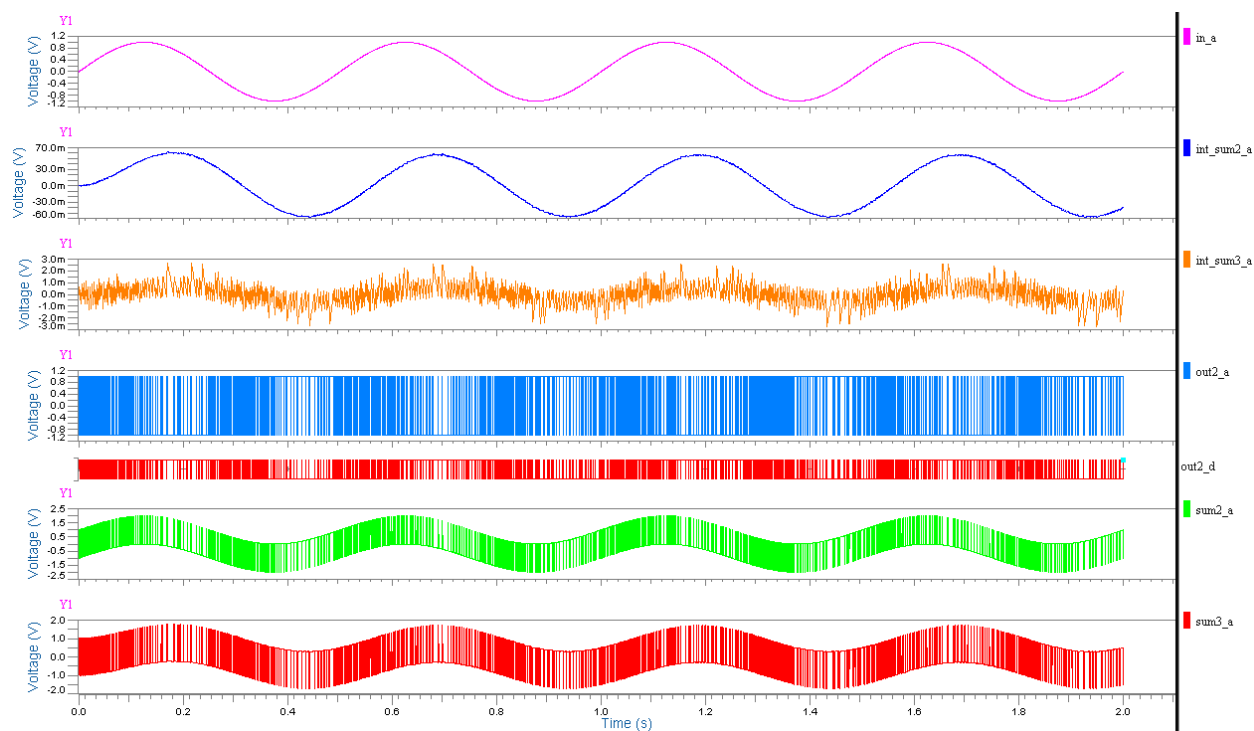
entity decim is
    generic(osr : integer := 32);
    port(clk : in std_logic;
         input : in std_logic;
         terminal output : out electrical);
end entity;

architecture default of decim is
    quantity v across i through output;
    signal decimated : real := 0.0;
begin
    process(clk)
        variable progress : integer := 0;
        variable result : integer := osr;
    begin
        if rising_edge(clk) then
            if input = '1' then
                result := result + 1;
            end if;
            if input = '0' then
                result := result - 1;
            end if;
            progress := progress + 1;
            if progress = osr then
                progress := 0;
                decimated <= real(result - osr) / real(osr);
                result := osr;
            end if;
        end if;
    end process;

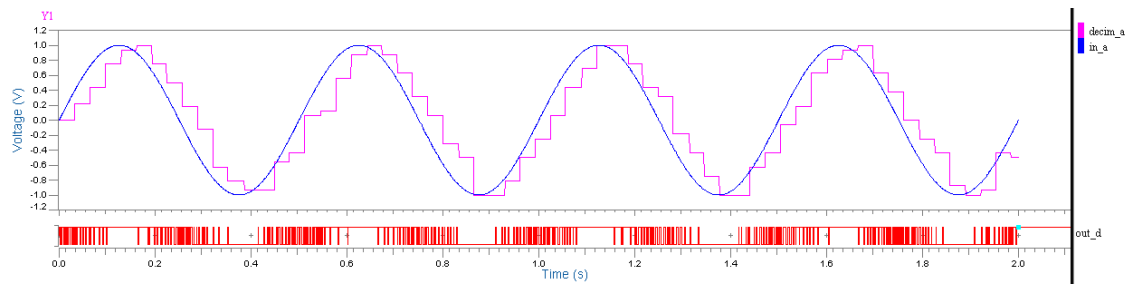
    decimated == v;
end architecture;
```



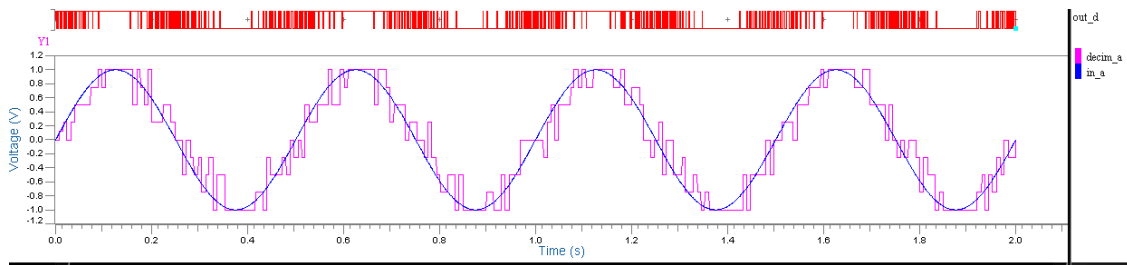
Rysunek 4. Działanie modulatora sigma-delta pierwszego rzędu. Sygnał out\_d to sygnał wyjściowy modulatora.



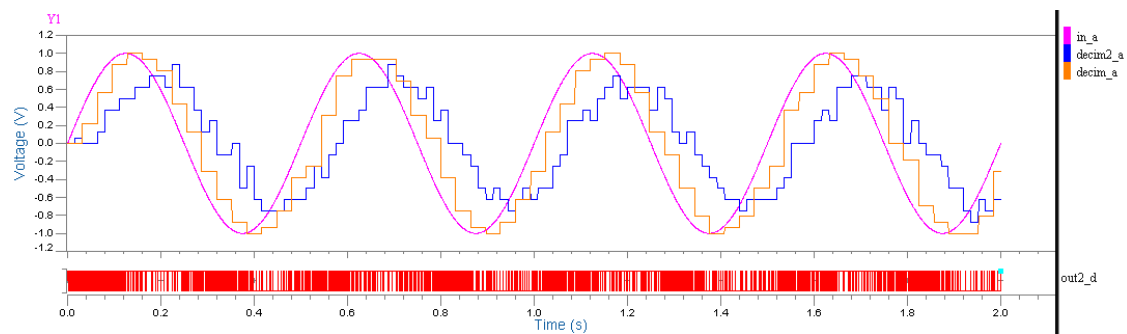
Rysunek 5. Działanie modulatora sigma-delta drugiego rzędu. Sygnał out2\_d to sygnał wyjściowy modulatora.



Rysunek 6. Sygnał przetworzony w modulatorze pierwszego rzędu i odtworzony układem decymacyjnym. Współczynnik OSR wynosi domyślne 32.



Rysunek 7. Sygnał przetworzony w modulatorze pierwszego rzędu i odtworzony układem decymacyjnym. Współczynnik OSR wynosi 8.



Rysunek 8. Sygnał przetworzony w modulatorze drugiego rzędu i odtworzony układem decymacyjnym (niebieski) porównany z sygnałem przetworzonym przez modulator pierwszego rzędu (pomarańczowy). Współczynnik OSR dla obydwu konfiguracji wynosi domyślne 32.