

# Лабораторна робота 1

## Генератор синусоїдального сигналу

### 1 Мета:

Мета – одержати знання і практичний досвід в розробці генераторів синусоїдальних сигналів для сучасних спеціалізованих ЕОМ, щоб одержати досвід програмування і відлагоджування на мові VHDL.

### 2 Теоретичні відомості

Цифровий генератор синусоїдального сигналу широко використовується в ЦОС-застосуваннях як джерело синусоїдального, косинусоїдального сигналу для частотного перетворення, дискретного перетворення Фур'є, в радіоприймачах, радарях, мобільних телефонах і т.і.

Наступні алгоритми таких генераторів широко використовуються:

- обчислення функції синусоїди;
- генератори на основі таблиці;
- алгоритми генерації;
- комбіновані алгоритми.

Обчислення синусоїдальної функції – звичайний спосіб генерації синусоїдального сигналу в ПЕОМ і інших програмно керованих ЕОМ. Тут для цієї мети звичайно використовується відповідна команда сопроцесора з плаваючою комою. В спеціалізованому процесорі використовуються різні алгоритми апроксимації, як наприклад, ряд Тейлора, алгоритм цифра за цифрою, інтерполяції алгоритми і т.і. Наприклад, функція синусоїди і косинусоїди на інтервалі  $|x| < 1$  може бути розрахована як

$$\begin{aligned}\sin(\pi x/2) &= 1,57063x - 0,64323x^3 + 0,07271x^5, \\ \cos(\pi x/2) &= 0,9994 - 1,22279x^2 + 0,22399x^4,\end{aligned}\tag{1}$$

з похибкою, яка менша ніж 0,06%. Недолік цього способу полягає в великій складності обчислення (в прикладі вище – до 6 операцій множення і 2 додавання для синусоїдальної функції). Крім того, функція визначена для кутів менше ніж  $\pi/2$ , і додаткові обчислення потрібні для пошуку функції в інших діапазонах. Переваги способу – широкий діапазон генерованих частот.

Генератор на основі таблиці – найпростіший і найбільш широко вживаний. Суть способу – побудувати таблицю  $M$  відліків синусоїдальної функції, яка формує один період синусоїдального сигналу. Це означає, що за адресою  $i$  записується значення  $S_i = \sin(2\pi i/M)$ . Генерація сигналу полягає в читанні відліків, адресації їх лічильником приросту адреси. Приріст  $k=1,2,\dots,M/2$  такого лічильника пропорційний частоті  $f = kf_s/M$ , результуючого синусоїдального сигналу, де  $f_s$  – частота виборки (квантування). Точність як вихідного синусоїдального сигналу так і його встановленої частоти залежать від об'єму таблиці  $M$  і розрядності відліків  $S_i$ .

Алгоритм генерації генерує синусоїдальний сигнал, використовуючи деякі основні залежності трансцендентних функцій. Він звичайно реалізується як вирішення деякого дифференційного рівняння. Наприклад, наступне дифференційне рівняння, апроксимоване як різницеве рівняння

$$y(i) = 2\cos(b)y(i-1) - 1 \cdot y(i-2), \quad i=0,1,\dots,\tag{2}$$

моделює рекурсивний цифровий фільтр другого порядку на границі режимів підсилення і насичення. Такий алгоритм генерує синусоїдальний сигнал при початкових умовах

$$y(-1) = -\sin(b); \quad y(-2) = -\sin(2b);\tag{3}$$

або косинусоїдальний сигнал при умовах

$$y(-1) = \cos(b); \quad y(-2) = \cos(2b);$$

з частотою  $f = b f_s / (2\pi)$  Гц. Теоретично рівняння (2) представляє собою стабільний генератор синусоїдального сигналу, тобто він працює без згасання і обмеження вихідного сигналу, якщо добуток  $1 \cdot y(i-2)$  дорівнює точно множенню на 1 [1], що звичайно можна одержати без складнощів. Але коефіцієнти синусоїди і косинусоїди повинні округлятися при машинному представленні в такий спосіб, щоб відлік вихідної синусоїди для нульового кута строго дорівнював нулю, тобто

$$y(0) = 2\cos(b)(-\sin(b)) - (-\sin(2b)) = 0. \quad (4)$$

На Рис.1 показана структура потоків даних, яка вирішує рівняння (2).

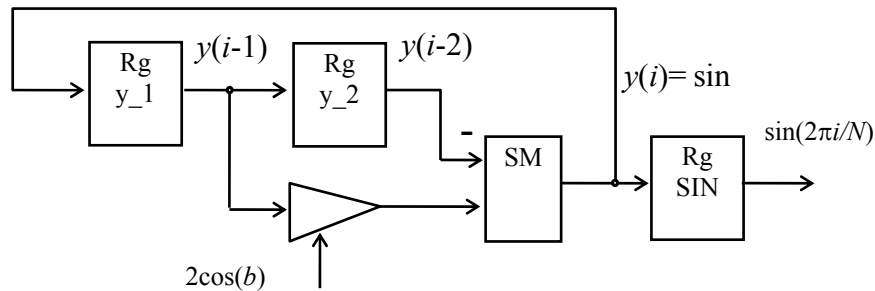


Рис.1. Структура для реалізації рівняння (2)

Ця структура може бути описана наступною VHDL-програмою.

```

library IEEE;
use IEEE.std_logic_1164.all;
entity SINE_GEN
  port (CLK: in STD_LOGIC;
        RESET: in STD_LOGIC;
        SIN: out integer range -2**15 to 2**15-1);
end SINE_GEN;

architecture SIMPLE of SINE_GEN is
  signal y_1, y_2 : integer range -2**15 to 2**15-1;
  constant sinb: integer := -286;
  constant sin2b: integer := -572;
  constant cosb: integer := 32767;
begin
  process(CLK,RESET)
  variable SINE: integer;
  begin
    if RESET = '1' then
      y_1 <= sinb;
      y_2 <= sin2b;
      SIN <= 0;
    elsif CLK = '1' and CLK'event then
      SINE_GEN := 2*cosb*y_1 / 2**15
      y_2 <= y_1;
      y_1 <= SINE;
      SIN <= SINE;
    end if;
  end process;
end SIMPLE;

```

Синусоїдальна функція так само, як і звичайний сигнал має бути представленою в діапазоні від  $-1.0$  до  $1.0$ . Але в VHDL сигнали представляються як цілі числа або бітові вектори. Тому в цій моделі всі значення необхідно масштабувати з коефіцієнтом  $2^{-15}$ . Початкові значення сигналів  $y_1$ ,  $y_2$  і констант  $\cos b = 2^{-15} \cos(b)$  для різних частот показані в наступній таблиці 1.

Таблиця 1

$-\sin b$	$-\sin 2b$	$\cos b$	Обчислений період синусоїдального сигналу, тактів	Одержаний період синусоїдального сигналу, тактів	Одержана амплітуда	Зміщення постійної складової
19261	31165	26510	10	10	32767	0
2856	5690	32642	72	72	32422	-6
286	572	32767	720	703	28808	-87

Цей генератор доволі простий. Він часто використовується для генерації сигналу однієї частоти. Недоліки цього генератора :

- невеликий діапазон зміни частоти, яка обмежена розрядністю даних і формулою (4);
- амплітуда сигналу різна для різних частот, і присутнє деяке зміщення (постійна складова, див. таблицю 1);
- для множини частот коефіцієнти повинні бути обчислені і/або запам'ятовані;
- дуже висока ( $>0,13f_s$ ) і занадто низька ( $<0,001f_s$ ) частоти не можуть генеруватися без великих похибок.

Інша схема генерації – на основі широко відомих тригонометричних формул:

$$\begin{aligned} \sin(x+y) &= \sin x \cos y + \cos x \sin y; \\ \cos(x+y) &= \cos x \cos y - \sin x \sin y. \end{aligned} \quad (5)$$

Тут  $\sin x$ ,  $\cos x$  – це відліки генерованих сигналів,  $y$  – кут, на який відрізняються сусідні відліки, тобто цей кут представляє задану частоту.

Недолік цього алгоритму полягає в його нестабільності через неточне представлення і обчислення синусоїдального і косинусоїдального відліків. Це означає що

$$\sin^2 x + \cos^2 x \neq 1 \quad \text{і} \quad \sin^2 y + \cos^2 y \neq 1$$

через похибки усікання результатів (2). Цей недолік може бути мінімізований додаванням деякої нелінійності в цю схему, яка буде зменшувати або збільшувати амплітуду сигналу за законом зворотнього зв'язку.

В комбінованих алгоритмах використовується суперпозиція перелічених вище алгоритмів. Наприклад, розглянемо генератор, частота якого повинна регулюватись точно. Тоді такий генератор може будуватись як два генератори, перший з них генерує синусоїдальний і косинусоїдальний сигнал з високою частотою, а інший – генерує низьку частоту. результуючий сигнал одержується змішуванням сигналів, використовуючи рівняння (5).

### 3. Приклад розробки генератора

Розглянемо розробку генератора синусоїдального сигналу, який видає синусоїдальний і косинусоїдальний сигнали. Його параметри:

- частота –  $0,01f_s$ ,
- розмір таблиці синусоїд – 16,
- діапазон аргументу синусоїдальної функції в таблиці –  $0 - \pi$ ,
- розрядність аккумулятора фази – 16 біт,
- вихідна розрядність – 16.

Структура генератора показана на Рис.2

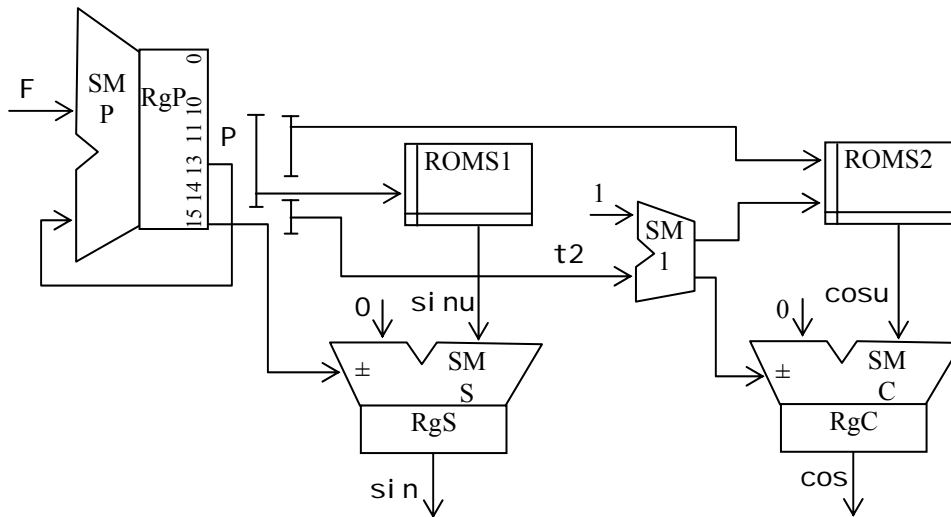


Рис.2. Генератор синусоїдального сигналу

Суматор SMP з регістром RgP реалізує акумулятор фази з приростом  $F$ . ПЗП ROMS1, ROMS2 зберігають половину періоду синусоїдальної функції. ROMS1 адресується 5 старшими бітами RgP крім найстаршого. Тому весь період синусоїдального сигналу складається з 32 відліків.

Суматор SMS інвертує код синусоїди, щоб генерувати негативне значення результуючого синусоїдального сигналу. Він інвертує коли 15-й біт RgP дорівнює 1. В іншій ситуації цей суматор пропускає дані без змін.

Суматор SM1 додає 1 в 2 старших розряди коду RgP, і тому він зсуває значення фази на  $90^\circ$ . Цей спосіб реалізує адресацію, яка забезпечує генерацію косинусоїдальної функції за допомогою таблиці синусів. Суматор SMC і ПЗП ROMS2 генерують косинусоїдальну функцію. Відліки синусоїди і косинусоїди буферизуються в регістрах RgS і RgC, відповідно.

Кожен період сигналу в акумуляторі фази виникає переповнення. Щоб генерувати сигнал з частотою  $0,01f_s$ , потрібно 100 тактів синхросерії, щоб одержати переповнення. Така ситуація виникає, коли приріст фази дорівнює:  $F = \lfloor 2^{16} / 100 \rfloor = 655$ . Відносна похибка представлення частоти через усікання коду фази дорівнює:  $\delta_f = (2^{16} / 100 - F) / F = 5,5 \cdot 10^{-4}$ .

Цей генератор описується в наступному коді.

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_signed.all;
entity SIN_GEN
  port(CLK : in STD_LOGIC;
        RST : in STD_LOGIC;
        F : in STD_LOGIC_VECTOR(15 downto 0);
        SIN_O : out STD_LOGIC_VECTOR(15 downto 0);
        COS_O : out STD_LOGIC_VECTOR(15 downto 0));
end SIN_GEN;
architecture beh of SIN_GEN is
  type ARR array (0 to 15) of integer range -2**15 to 2**15-1;
  constant SIN_TABLE:ARR:=
    (0, 6392,12539,18204, 23170,27245,30273,32138, --таблиця синусів
     32767,32138,30273,27245, 23170,18204,12539,6392);
  signal t:ARR:= (0,others=>0);
  signal P,sinu,cosu:STD_LOGIC_VECTOR(15 downto 0);
  signal t2:STD_LOGIC_VECTOR(1 downto 0);
begin

```

```

process
    variable j: real;
begin
    for i in 0 to 15 loop
        j:=real(i);
        t(i)<=integer(FLOOR(32767.9* SIN(MATH_2_PI*j/32.0)));
    end loop;
    wait;
end process;

CT_PHASE:process(CLK,RST) begin    -- лічильник фази
    if RST='1' then
        P<="X"0000";
    elsif CLK='1' and CLK'event then
        P<=P+F;
    end if;
end process;

sinu<=CONV_STD_LOGIC_VECTOR(SIN_TABLE(CONV_INTEGER('0'&P(14 downto 11))),16);

t2<=P(15 downto 14) +1;
cosu<=CONV_STD_LOGIC_VECTOR(SIN_TABLE(CONV_INTEGER('0'&t2(0)&P(13 downto 11))),16);

process(CLK,RST) begin    -- регістр з перетворенням в доповнюючий код
    if RST='1' then
        SIN_O<=(others=>'0');
        COS_O<=(others=>'0');
    elsif CLK='1' and CLK'event then
        if P(15)='0' then
            SIN_O<= sinu;
        else
            SIN_O<=0 - sinu;
        end if;
        if t2(1)='0' then
            COS_O<= cosu;
        else
            COS_O<=0 - cosu;
        end if;
    end if;
end process;
end beh;

```

#### 4. Виконання лабораторної роботи

Генератор повинен генерувати як синусоїдальний, так і косинусоїдальний сигнали. Амплітуда вихідного сигналу повинна бути стабільною. Генератор повинен генерувати множину частот  $f_1, f_2, i f_3$ , які задаються 2-бітовим вхідним кодом. Вхідний і вихідні сигнали повинні представлятися кодами типу STD\_LOGIC.

Кожен варіант роботи має множину параметрів, які нумеруються цілими числами. Ці номери одержуються з номеру залікової книжки студента. Розглянемо 3 останні цифри  $a_2, a_1, a_0$ , номеру залікової книжки. Тоді номер варіанту

$$N = 100a_2 + 10a_1 + a_0 = 2^9b_9 + 2^8b_8 + 2^7b_7 + 2^6b_6 + 2^5b_5 + 2^4b_4 + 2^3b_3 + 2^2b_2 + 2^1b_1 + b_0,$$

де  $b_i$  – це біти номеру  $N$  в двійковому представленні.

Тип генератору вибирається з Таблиці 1.

Таблиця 1

b <sub>0</sub>	0	1
Тип	На основі таблиці	Генератор за алгоритмом (2)

Набір частот представлений в Таблиці 2.

Таблиця 2

b <sub>3</sub> , b <sub>2</sub> , b <sub>1</sub>	Частот генератора, частка $f_c$		
000	0,001	0,01	0,03
001	0,002	0,02	0,04
010	0,003	0,03	0,05
011	0,004	0,04	0,06
100	0,005	0,05	0,07
101	0,006	0,06	0,08
110	0,007	0,07	0,09
111	0,008	0,08	0,1

Генератор повинен мати розрядність, що задана в наступній таблиці

Таблиця 3

b <sub>4</sub> , b <sub>3</sub> , b <sub>2</sub>	Розрядність синусоїдального сигналу	Розрядність аккумулятора фази	Розрядність коефіцієнтів, суматорів	Розрядність адреси таблиці
000	8	16	12	4
001	8	18	14	4
010	10	20	16	5
011	10	22	18	5
100	12	24	20	5
101	12	26	22	6
110	16	28	24	6
111	16	32	28	6

## 5. Розробка стенду для іспитів і режими тестування

Генератор тестується в середовищі ActiveHDL з додаванням стимулюючих сигналів для входів CLK і RST, з дослідженням графіків вихідних синусоїдального і косинусоїдального сигналів. Частота синхросигналу CLK повинна бути встановлена на 100 МГц.

Для кожної заданої частоти частота вихідного синусоїдального сигналу повинна вимірюватись, вимірюючи період вихідного графік сигналу. Розраховується похибка цієї частоти.

## 6. Звіт про лабораторну роботу

Звіт про лабораторну роботу повинен вміщувати:

- мету роботи,
- опис генератора,
- VHDL - текст,
- графіки тестування,
- висновки.

## Література

1. Отнес Р., Энноксон Л. Прикладной анализ временных рядов. –М.:Мир. –1982. – 428 с.
2. Рабинер Л., Гоулд Б. Теория и применение цифровой обработки сигналов. –М.:Мир. –1978. – 848 с.