

# Лабораторні роботи

## Лабораторна робота 1 Основи мови VHDL. Модель генератора сигналів

**Мета роботи:** одержати знання та навички у розробці поведінкових моделей генераторів тестових сигналів для ЦОС, навчитись користуватись VHDL-симулятором.

### Теоретичні відомості

Різноманітні генератори сигналів використовуються для діагностування, тестування та налагодження алгоритмів і пристроїв ЦОС. Часто використовується генератор синусоїдальних сигналів через те, що такий сигнал проходить через лінійну систему без зміни своєї форми. Шумовий сигнал має широкий спектр і тому дослідження його проходження через лінійну систему дає змогу оцінити її частотні характеристики. Також шум додається до сигналу, щоб виявити фільтруючі властивості системи. Пропускаючи через систему сигнал спеціальної форми, можна перевірити її якість, так як система з гідною передаточною характеристикою зберігає форму сигналу.

В лабораторній роботі слід розробити генератор сигналів спеціальної форми *Gen* та виконати з ним ряд дослідів, включивши його у схему на рис. 4.1.

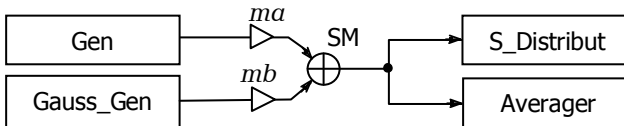


Рис. 1.1. Схема для лабораторної роботи 1

Інші блоки схеми на рис.4.1. — це генератор гаусового шуму *Gauss\_Gen*, блок аналізу статистичного розподілення сигналу *S\_Distribut* та блок осереднення сигналу *Averager*.

В основу генератора гаусового шуму покладено виконання центральної граничної теореми. Згідно з нею, якщо випадкова величина є сумою досить великого числа незалежних між собою доданків, кожен з яких є незначним за величиною, то ця сума має розподіл, який наближається до нормального.

Блок *Gauss\_Gen* має дві частини: генератор чисел з рівномірним розподілом та суматор, який складає до  $np$  згенерованих чисел.

Генерація чисел з рівномірним розподілом в інтервалі  $[0.0,1.0]$  виконується за конгруентним алгоритмом з використанням процедури UNIFORM. Одержана сума ділиться на  $np/2$  і видається як відлік шуму, який розподілено в інтервалі  $[-1.0,1.0]$ . Тут  $np$  – це настроювальна константа. При  $np = 1$  генератор видає шум з рівномірним розподілом, при  $np = 2$  – з трикутним розподілом, а при  $np \geq 10$  – практично з гаусовим розподілом.

Сигнали від генераторів *Gen* та *Gauss\_Gen* складаються у суматорі *SM* після множення на масштабні коефіцієнти  $ma$ ,  $mb$ , відповідно (рис. 4.1). В результаті, одержуємо сигнал  $x(n)$ , що досліджується.

Статистичні властивості сигналу  $x(n)$  можна оцінити завдяки побудові гістограми розподілу його вірогідності. Обчислення гістограми виконують наступним чином. Інтервал зміни величини  $x(n)$ , наприклад,  $a < x(n) < b$ , ділять на  $k$  підінтервалів рівної довжини.  $i$ -му інтервалу, включаючи 0-й підінтервал  $x(n) \leq a$  та  $k+1$  – й підінтервал  $b \leq x(n)$ , ставиться у відповідність натуральне число  $M_i$ , яке дорівнює кількості відліків  $x(n)$ , які попадають у відповідний підінтервал. Підінтервал часто називають інтервалом групування або кишенею.

Нехай ширина підінтервалу дорівнює  $c = (b - a)/k$ , а границя  $i$ -го інтервалу –  $d_i = a + i \cdot c$ . Тоді після аналізу усієї послідовності  $x(n)$  у  $i$ -й кишені одержимо число таких  $x(n)$ , що  $d_{i-1} \leq x(n) < d_i$ . Виключеннями є 0-а кишеня з кількістю відліків  $x(n) < a$  та  $k+1$ -а кишеня з числом таких  $x(n)$ , що  $b \leq x(n)$ .

При обробці  $x(n)$  на комп'ютері організують масив цілих чисел  $M$  розміром  $k+2$ . Якщо  $a \leq x(n) < b$ , то обчислюють номер кишені

$$i = \left[ \frac{(x(n)-a)}{c} \right] + 1. \quad (4.1)$$

Тут дужки  $[\ ]$  означають цілу частину числа. Одержавши індекс за формулою(4.1) у  $i$ -ту комірку, тобто кишеню  $M(i)$  додають 1. Такі обчислення повторюють для усіх  $x(n)$ .

На основі одержаних результатів можна для послідовності  $x(n)$  довжиною  $N$  обчислити вибірккову функцію щільності вірогідності за формулою

$$P_i = \frac{M(i) \cdot k}{N(b - a)}. \quad (4.2)$$

Вибір параметрів  $a$ ,  $b$ ,  $k$  залежить головним чином від уяви про розподіл сигналу  $x(n)$ , способу збору даних та їх кількості. Напевне, що кількість кишень недоцільно вибирати більше за  $\sqrt{N}$  чи за

кількість рівнів квантування сигналу. Параметри  $a$ ,  $b$ ,  $k$  уточнюються після циклів пробної побудови гістограми.

Побудова гістограми виконується блоком `S_Distribut`. Він має настроювальні константи `channels`, тобто  $k$ , `dmin`, тобто  $a$  та `drange`, який дорівнює  $b - a$ .

Якщо сигнал  $x(n)$  – періодичний, то  $x(n+T_0) = x(n)$ , де  $T_0$  – його період. Реальний сигнал  $x'(n)$  має додаткову складову у вигляді шумового сигналу  $q(n)$ , тобто  $x'(n) = x(n) + q(n)$ . Блок осереднення сигналу `Averager` виконує додавання періодів сигналу  $x'(n)$ . Якщо шум  $q(n)$  є послідовністю випадкових чисел з нормальним законом розподілу, то після додавань  $N$  періодів сигналу  $x'(n)$  його складова  $q(n)$  збільшиться менше, ніж у  $N$  разів. Таким чином, після  $N$  додавань періодів сигналу  $x'(n)$  та нормалізації результату діленням на  $N$  для  $n = 0, \dots, T_0 - 1$  та  $k = 0, \dots, NT_0 - 1$  одержимо результат

$$\begin{aligned} y(n) &= \frac{1}{N} \sum_{i=0}^{N-1} (x(n+T_0 i) + q(k)) = x(n) + \frac{1}{N} \sum_{i=0}^{N-1} q(k) = \\ &= x(n) + m(q(k)) + \frac{1}{\sqrt{N}} q'(k), \end{aligned} \quad (4.3)$$

де  $m(q(k))$  – математичне очікування шуму  $q(k)$ ,  $q'(k)$  – осереднений шумовий сигнал. Отже, бачимо, що осередненням сигналу можна знизити рівень шуму до  $\sqrt{N}$  разів. Тому осереднення періодичного сигналу часто використовується для його фільтрації від шуму. При цьому важливими умовами є прямування до нуля математичного очікування шуму та строга рівність довжини масиву  $y(n)$ , де накопичується осереднений сигнал, періоду  $T_0$  сигналу  $x(n)$ .

### Завдання для роботи

1. Розробити модуль `Gen`, який генерує періодичну функцію, що має два інтервали:

$$f(i) = \begin{cases} f_1 & \text{на першому інтервалі довжиною } N_1, \\ f_2 & \text{на другому інтервалі довжиною } N_2. \end{cases}$$

Тут період функції дорівнює  $T_0 = N_1 + N_2$ . Функції  $f_1$ ,  $f_2$  та їх параметри  $N_1$ ,  $N_2$  беруться з таблиці 4.1 в залежності від номера варіанту роботи. Модуль описується мовою VHDL як об'єкт і архітектура. Параметри  $N_1$ ,  $N_2$  повинні бути настроювальними константами, тобто `generics`.

2. Розробити стенд для випробувань, що відповідає структурі на рис. 4.1. Стенд описується структурним стилем мовою VHDL з застосуванням засобу `Block Diagram` системи `ActiveHDL`.

Таблиця 1.1 Функція, яка реалізується в модулі Gen

№ вар.	$f_1$	$f_2$
1	$\sin(2\pi i/N_1) + \sin(6\pi i/N_1)/3$	$\sin(2\pi i/N_2) + \sin(6\pi i/N_2)/3$
2	$i/N_1$	$1 - i/N_2$
3	$i^2 / N_1^2$	$(N_2 - i)^2 / N_2^2$
4	$\sqrt{i/N_1}$	$\sqrt{(N_2 - i)/N_2}$
5	$\sin^2(2\pi i/N_1)$	$\sin^2(2\pi i/N_2)$
6	$\exp(- i - N_1/2 )$	$-\exp(- i - N_2/2 )$
7	$1/(1 + (10 \cdot (i - N_1)/N_1)^2)$	$1/(1 + (10 \cdot i/N_2)^2)$
8	$1 - \exp(-10 \cdot i/N_1)$	$\exp(-10 \cdot i/N_2)$
9	$\frac{4\sin(8\pi(i-N_1/2)/N_1)}{i-N_1/2+\delta}$	$\frac{4\sin(8\pi(i-N_2/2)/N_2)}{i-N_2/2+\delta}$
10	$\ln(i+1)/\ln N_1$	$\ln(N_2 - i + 1)/\ln N_2$
11	$i/N_1$	$\sqrt{(N_2 - i)/N_2}$
12	$i^2 / N_1^2$	$1/(1 + (10 \cdot i/N_2)^2)$
13	$\sqrt{i/N_1}$	$\exp(-10 \cdot i/N_2)$
14	$\sin^2(2\pi i/N_1)$	$-\sin^2(2\pi i/N_2)$
15	$\exp(- i - N_1/2 )$	$\sin^2(2\pi i/N_1)$
16	$1/(1 + (10 \cdot (i - N_1)/N_1)^2)$	$\sqrt{(N_2 - i)/N_2}$
17	$1 - \exp(-10 \cdot i/N_1)$	$(N_2 - i)^2 / N_2^2$
18	$\frac{4\sin(8\pi(i-N_1/2)/N_1)}{i-N_1/2+\delta}$	$-\frac{4\sin(8\pi(i-N_2/2)/N_2)}{i-N_2/2+\delta}$
19	$\ln(i+1)/\ln N_1$	$\exp(-10 \cdot i/N_2)$
20	$i/N_1$	$\exp(-10 \cdot i/N_2)$
21	$i^2 / N_1^2$	$\ln(N_2 - i + 1)/\ln N_2$
22	$\sqrt{i/N_1}$	$\exp(-10 \cdot i/N_2)$
23	$\sin(2\pi i/N_2) - \sin(6\pi i/N_2)/3$	$\sin(2\pi i/N_2) - \sin(6\pi i/N_2)/3$
24	$\exp(- i - N_1/2 )$	$-\frac{4\sin(8\pi(i-N_2/2)/N_2)}{i-N_2/2+\delta}$
25	$1/(1 + (10 \cdot (i - N_1)/N_1)^2)$	$\exp(-10 \cdot i/N_2)$
26	$1 - \exp(-10 \cdot i/N_1)$	$\sqrt{(N_2 - i)/N_2}$
27	$\frac{4\sin(8\pi(i-N_1/2)/N_1)}{i-N_1/2+\delta}$	$\exp(- i - N_1/2 )$
28	$\ln(i+1)/\ln N_1$	$(N_2 - i)^2 / N_2^2$

При цьому структура складається з розробленого модуля *Gen* та готових модулів генератора гаусового шуму *Gauss\_Gen*, блока аналізу статистичного розподілення сигналу *S\_Distribut*, блока осереднення сигналу *Averager*, генератора синхроімпульсів *CLK\_GEN* та суматора сигналів *ADD2*.

3. Встановити початкові значення настроювальних констант в модулях: період синхросигналу  $tclk=10\text{ ns}$  – в *CLK\_GEN*,  $N1 = N_1 = 100$ ,  $N2 = N_2 = 200$  – в *Gen*, математичне очікування  $m=0.0$ , вид генерації  $pn=10$  в *Gauss\_Gen*, підсилення каналів сигналу від *Gen* –  $ma = 1.0$ , сигналу від *Gauss\_Gen* –  $mb=3.0$  – в модулі *ADD2*. В модулі *S\_Distribut* кількість кишень  $channels = 102$ , значення нижньої межі  $dmin$  та діапазону зміни сигналу  $drange$  задати з урахуванням діапазону зміни заданої функції  $f(i)$  та доданої до неї величини шуму з урахуванням  $ma$  та  $mb$ . Період сигналу, що осереднюється в модулі *Averager*, дорівнює  $N = T_o = N_1 + N_2$ .

4. Скопіювати побудовану модель стенду для випробувань. Виконати моделювання для 16 періодів сигналу. Номер періоду сигналу видається на вихід *Naver* модуля *Averager*. Якщо необхідно, скоректувати параметри  $dmin$ ,  $drange$ .

5. Фрагменти графіків сигналів на виходах модулів *ADD2*, *S\_Distribut*, та *Averager* зберегти у звіті лабораторної роботи разом з параметрами модулів.

6. Повторити пп. 3,4,5 з різними значеннями настроювальних констант  $N1$ ,  $N2$ ,  $N\ mb$ ,  $m$ ,  $pn$  та для різної тривалості моделювання (кількості періодів сигналу). Виконати також моделювання для випадку  $N \neq N_1 + N_2$ .

7. Проаналізувати одержані графіки сигналів з виходів модулів *S\_Distribut* та *Averager*. При цьому перевірити справдження формули (4.3). Сформулювати залежності між формою і характером сигналів та формою кривої їх розподілення на виході модуля *S\_Distribut*. Зробити висновки по роботі.

### Приклад виконання роботи

Нехай задано функцію

$$f(i) = \begin{cases} \arctg(i/N_1) & \text{на першому інтервалі довжиною } N_1, \\ \cos(i/N_2) & \text{на другому інтервалі довжиною } N_2. \end{cases}$$

Тоді модуль, який її генерує, описується як наступний:

```
entity Gen is generic(n1:natural:=100;
                       n2:natural:=200);
port(CLK : in STD_LOGIC;
      RST : in STD_LOGIC;
```

```

DATA_OUT : out REAL:=0.0;
START : out BIT);
end Gen;
architecture MODEL of Gen is
    signal ct2:natural;
begin
    process(CLK,RST)
        begin
            if RST='1' then
                ct2<=0;
                START<='1';
            elsif CLK='1' and CLK'event then
                START<='0';
                if ct2=n1+n2-1 then
                    ct2<=0;
                else
                    ct2<=ct2+1;
                end if;
                if ct2<=n1 and ct2>0 then
                    DATA_OUT<= arctan(real(ct2)/real(n1));
                else
                    DATA_OUT<= cos(MATH_PI*real(ct2-n1)/real(n2));
                end if;
            end if;
        end process;
    end MODEL;

```

Побудована схема стенду для випробувань показана на рис.4.2.

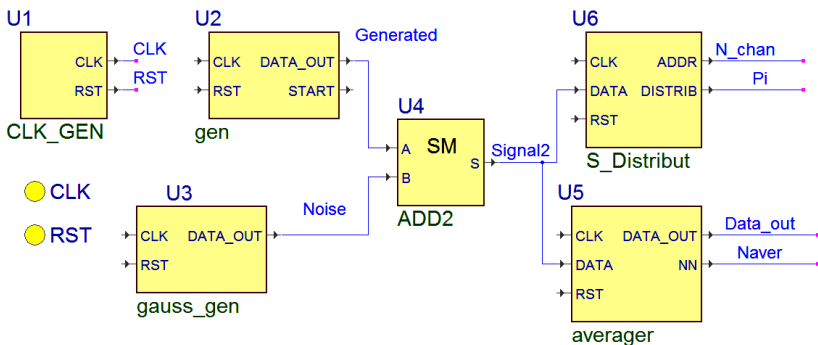


Рис. 1.2. Схема стенду для випробувань

На рис. 4.3. показані графік сигналу, згенерованого модулем Gen, графік сигналу з доданим гаусовим шумом, амплітуда якого відноситься як 3:1 та сигнал після 100 циклів осереднення у модулі

**Averager.** Аналіз графіків показує, що дійсно, після 100 осереднень періодичного сигналу рівень доданого гаусового шуму зменшується приблизно у 10 разів.

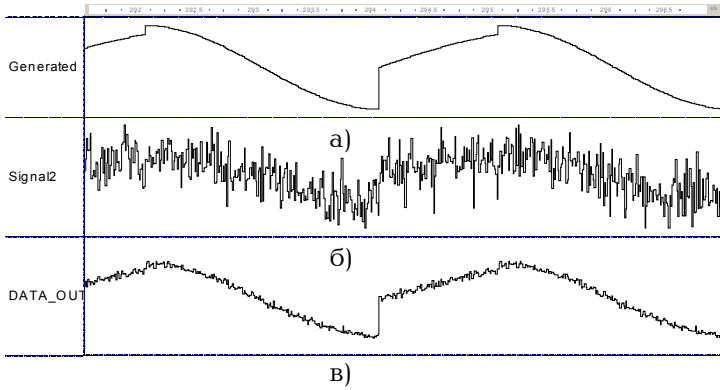


Рис. 3. Графіки згенерованого сигналу (а), сигналу з доданим гаусовим шумом (б), та сигналу після осереднення (в)

На рис. 4.4. показані графіки сигналів, одержаних на виході модуля *S\_Distribut* для сигналів без шуму, з доданим шумом з рівнем 0,3 та рівнем 3. Графіки показують, що сигнал має позитивну постійну складову. Перший графік має форму, характерну для сигналу без шуму, другий графік показує наявність в сигналі шуму, а третій — що в сигналі переважає шум, причому саме шум з імовірно гаусовим розподілом.

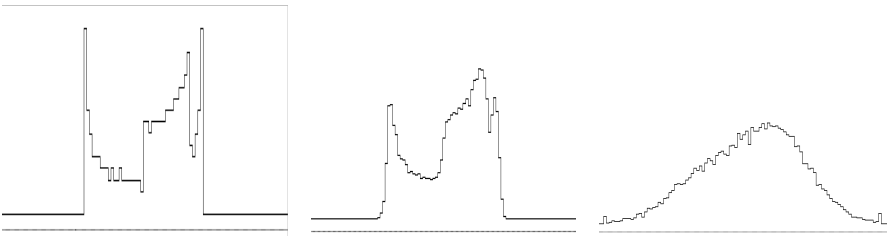


Рис. 1.4. Графіки розподілу сигналу без шуму (а) та сигналу з доданим гаусовим шумом (б), (в)

## Лабораторна робота 2

### Дослідження властивостей Z-перетворення

**Мета роботи:** одержати знання про спектральні властивості дискретних сигналів і передаточних функцій та навички у їх дослідженні за допомогою VHDL.

#### Теоретичні відомості

У підрозділі 2.2 вже розглядалися деякі властивості z-перетворення, які використовуються для формування алгоритму обробки сигналу. Далі розглядатимуться деякі його уточнення. z-перетворення широко використовується в ЦОС через те, що сигнал виду  $x(n) = z^n$  є **власною функцією** для лінійної системи, яка інваріантна до зсуву. Наведемо деякі приклади такого сигналу.

При  $z = 1$ : ...,  $x(-2) = 1$ ,  $x(-1) = 1$ ,  $x(0) = 1$ ,  $x(1) = 1$ ,  $x(2) = 1$ , ...

При  $z = 0.5$ : ...,  $x(-2) = 4$ ,  $x(-1) = 2$ ,  $x(0) = 1$ ,  $x(1) = 0.5$ ,  $x(2) = 0.25$ , ...

При  $z = e^{-j\pi/2}$ : ...,  $x(-2) = -1$ ,  $x(-1) = -j$ ,  $x(0) = 1$ ,  $x(1) = j$ ,  $x(2) = -1$ , ...

Те, що функція є власною для системи, означає, що після обробки сигналу, який має форму цієї функції, одержимо сигнал, який має таку саму форму, але затриманий у часі та має іншу амплітуду. Покажемо, що  $z^n$  є власною функцією для системи, що описується імпульсною реакцією. Якщо такий сигнал подати на вхід системи, то одержимо, згідно з (2.1) та властивостями лінійної системи, що інваріантна до зсуву, вихідний сигнал

$$y(n) = \sum_{k=-\infty}^{\infty} h(k)z^{n-k} = \sum_{k=-\infty}^{\infty} h(k)z^{-k}z^n = z^n \sum_{k=-\infty}^{\infty} h(k)z^{-k} \quad (1.4)$$

Отже, бачимо, що функція  $z^n$  пройшла незмінною на вихід системи. Згідно з (2.5) та (4.4), **передаточна функція** системи дорівнює

$$H(z) = \sum_{k=-\infty}^{\infty} h(k)z^{-k}. \quad (1.5)$$

Так само, як і  $z$ , функція  $H(z)$  є комплексною функцією з амплітудою  $|H(z)|$  та фазою  $\arg(H(z))$ . Функція  $H(z)$  має дві множини особливих точок. Перша множина – це такі точки  $q_1, \dots, q_M$ , для яких  $|H(q_i)| = 0$  і тому називаються **нулями** функції  $H(z)$ . Друга множина – це такі точки  $r_1, \dots, r_N$ , для яких  $|H(r_i)| = \infty$  і тому вони називаються **полюсами** функції  $H(z)$ .



Щоб знайти нулі та полюси  $H(z)$ , вираз (2.5) перетворюють у дріб з позитивними степенями при  $z$ . Якщо  $M - N = L$ , то чисельник (2.5) множать на  $z^M$ , а знаменник — на  $z^N$ . Після цього одержують дріб

$$H(z) = z^{-L} \cdot \frac{\sum_{r=0}^M b_r z^{M-r}}{\sum_{k=0}^N a_k z^{N-k}}, \quad (1.6)$$

чисельник та знаменник якого є поліномами з позитивними степенями при  $z$ , де  $a_0 = 1$ , множник  $z^{-L}$  представляє собою затримку на  $L$  тактів і, як правило, може бути усунений з розгляду. Ці поліноми прирівнюють до нуля та знаходять корені цих рівнянь  $q_1, \dots, q_M$  та  $r_1, \dots, r_N$ , відповідно. Маючи множини нулів та полюсів, функцію  $H(z)$  можна виразити через них за допомогою формули

$$H(z) = K \cdot \frac{(z - q_1) \dots (z - q_M)}{(z - r_1) \dots (z - r_N)} = K \cdot \frac{(1 - q_1 z^{-1}) \dots (1 - q_M z^{-1})}{(1 - r_1 z^{-1}) \dots (1 - r_N z^{-1})}, \quad (1.7)$$

де  $K$  — **коефіцієнт підсилення** системи. Отже, множини нулів та полюсів визначають передаточну функцію системи.

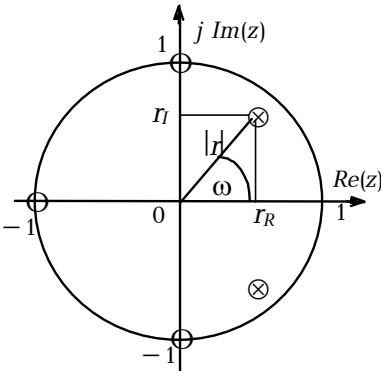


Рис. 1.5. Нулі та полюси фільтра нижніх частот

$|r| \leq 1$ , а кількість полюсів з ненульовими уявними координатами завжди парна.

Модуль передаточної функції  $|H(z)|$  у тривимірному просторі можна представити як нескінченну мембрану, що пружно розтягується, яка натягнута над комплексною площиною на одиничній висоті. Точки мембрани, які відповідають нулям функції, притягнуті до комплексної площини, а точки, що означають полюси, відтягнуті вгору до нескінченності. Якщо на мембрані зобразити концентрич-

Графічно нулі та полюси зображають як точки на комплексній площині у вигляді кружечків та хрестиків, відповідно. На рис. 4.5 показано приклад розміщення нулів та полюсів деякого фільтра нижніх частот. Нуль чи полюс, як корінь рівняння, знаходять як пару реального та уявного чисел:  $r = r_R + jr_I$ . Для зручності корінь представляють у полярних координатах парю амплітуди  $|r|$  та фази  $\omega$ :  $r = |r|e^{-j\omega}$ . При цьому фаза  $\omega$  відповідає певній частоті сигналу чи передаточної функції. Амплітуда полюса, як правило, не перевищує одиниці, тобто

ні кола, то після її деформації згідно з розміщенням нулів та полюсів як на рис. 4.5, одержимо зображення, як на рис. 4.6. Товста лінія на поверхні  $|H(z)|$  (рис. 4.6) відображається на комплексну площину як коло з одиничним радіусом. Вона будується при

$$z^{-n} = e^{-j\omega n} = \cos(-\omega n) + j \sin(-\omega n). \quad (1.8)$$

При підстановці послідовності  $e^{-j\omega n}$  в (2.4) одержимо дискретне перетворення Фур'є, а при підстановці в (4.5) чи (4.6) – передаточну функцію

$$|H(\omega)| = |H(e^{-j\omega n})|, \quad (4.9)$$

тобто залежність коефіцієнту передачі системи від частоти  $\omega$ . Цю функцію називають **амплітудо-частотною характеристикою** (АЧХ) системи. При цьому крива, яка на рис. 4.6 розміщена на циліндрі одиничного радіуса з віссю  $\omega|H(z)|$ , розкручується на площині графіка АЧХ у вигляді безкінечної періодичної кривої з періодом  $2\pi$ , як на рис. 4.7.

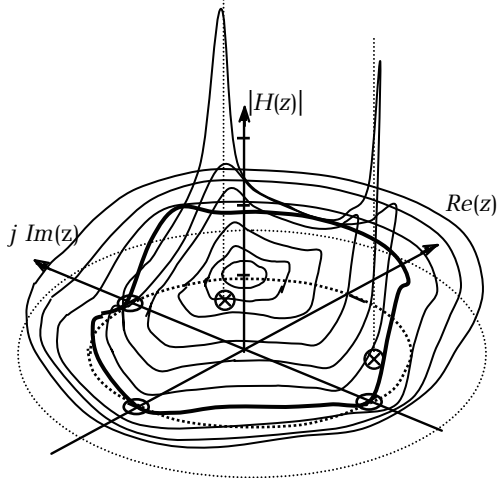


Рис. 6. Функція  $|H(z)|$  у трьохвимірному просторі

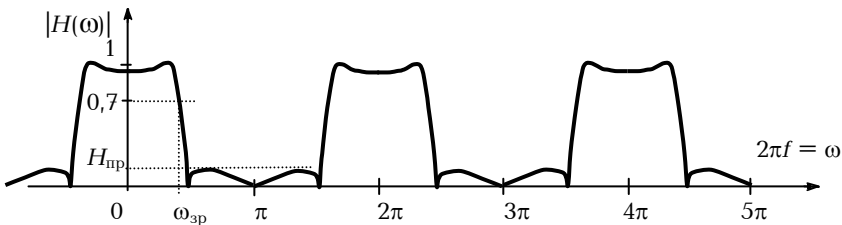


Рис. 7. Амплітудо-частотна характеристика фільтра нижніх частот

Для зручності вісь частот графіка АЧХ нормують, поділивши аргумент на  $2\pi$ , так що одиниця на цій осі відповідає частоті дискретизації сигналу. Аналогічно АЧХ, обчислюють **фазо-частотну функцію** (ФЧХ) системи

$$\arg(H(\omega)) = \arctg\left(\frac{\text{Im}(H(e^{-j\omega}))}{\text{Re}(H(e^{-j\omega}))}\right). \quad (1.10)$$

Для АЧХ цифрових фільтрів домовлено, що у **смузі пропускання** АЧХ має рівень не нижче  $\sqrt{0,5} \approx 0,7$ . Це означає, що у цій смузі потужність фільтрованого сигналу не знижується менше за рівень 0,5 від номінальної потужності. Відповідна частота границі смуги називається **частотою зрізу** ( $\omega_{зр}$  на рис. 4.7). Максимальний рівень фільтрованого сигналу у смузі непропускання називається **рівнем заглушення** фільтру ( $H_{пр}$  на рис. 4.7).

Через те, що більшість систем ЦОС є системами з послідовно з'єднаних блоків, стало зручним представляти АЧХ у логарифмічному масштабі, тобто  $H(\omega) = 20 \cdot \lg |H(e^{-j\omega n})|$ . Тоді результуюча АЧХ системи дорівнює сумі логарифмічних АЧХ її складових. При цьому відлік рівнів АЧХ ведуть у децибелах. Наприклад, рівню 1 відповідає 0 дБ, рівню 0,7 – рівень –3дБ., рівню 0,1 – рівень –20 дБ.

У цій лабораторній роботі досліджуються характеристики передаточної функції

$$H_1(z) = \frac{b + a(b+1)z^{-1} + z^{-2}}{1 + a(b+1)z^{-1} + bz^{-2}} \quad (1.11)$$

та інших передаточних функцій на її основі.  $H_1(z)$  відповідає **фазовому фільтру**, нулі та полюси якого показані на рис. 8. Нулі та полюси функції (1.8) мають однаковий кут  $\omega$ . Якщо модуль її полюса  $r$ , то модуль нуля дорівнює оберненій величині, тобто  $1/r$ . В результаті цього нулі повністю компенсують вплив полюсів і тому АЧХ дорівнює 1 у всьому діапазоні частот.

Положення полюсів та нулів визначається коефіцієнтами  $a, b$ , причому

$$a = -\cos(\omega); \quad b = r^2. \quad (1.12)$$

Фазовий фільтр не змінює амплітуди сигналу, але перекичує його фазу. Причому чим ближче параметр  $b$  до одиниці, тим різкіше перекичування фази на частоті  $\omega$ . Якщо з'єднати паралельно два фазових фільтра з характеристиками  $H_1(z), H_2(z)$ , наприклад:

$$H_3(z) = (H_1(z) + H_2(z))/2, \quad (1.13)$$

то в їх загальній передаточній характеристиці  $H_3(z)$  нулі перестають компенсувати полюси. Обидва канали  $H_1(z), H_2(z)$  пропускають два сигнали без зміни їх амплітуд. Але на частотах, де різниця фаз дорівнює  $\pi, 3\pi, \dots$ , сигнали компенсують один одного і результуючий сигнал є нульовим. Навпаки, на частотах, де різниця фаз  $0, 2\pi, 4\pi, \dots$ ,

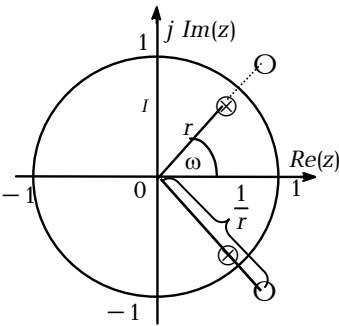


Рис. 1.8. Нулі та полюси фазового фільтра

цей сигнал залишається без змін. Завдяки цьому, в результуючій АЧХ  $|H(\omega)|$  з'являються смуги пропускання та непропускання.

У будь-якій лінійній системі, що інваріантна до зсуву, можна збільшити кожну затримку в  $k$  разів. Тоді замість передаточної функції  $H(z)$  розглядається функція з **кратними затримками**  $H(z^k)$ . Ефект такого збільшення затримок полягає в тому, що масштаб графіків АЧХ, ФЧХ вздовж осі частот зменшується у  $k$  разів.

Нехай  $k=2$ , тоді на графіку на рис. 4.7 слід замінити відліки  $\pi$  на  $\pi/2$ ,  $2\pi$  на  $\pi$  і т.д. В результаті, кількість смуг пропускання та непропускання в частотній області, що розглядається, подвоюється.

Нехай маємо  $H_4(z) = H_3(z^k)$ . Якщо АЧХ  $|H_4(\omega)|$  має кілька смуг пропускання, то робочу смугу можна виділити за допомогою **маскуючого фільтра**  $H_M(z)$ , який з'єднаний з ним послідовно. Тобто результуюча функція

$$H(z) = H_4(z) \cdot H_M(z) . \quad (1.14)$$

Для цього АЧХ маскуючого фільтра повинна мати смуги непропускання у тих діапазонах частот, де знаходяться смуги пропускання в АЧХ  $|H_4(\omega)|$ , які необхідно заглушити.

Для обчислення передаточних функцій бажано використати пакети IEEE.MATH\_REAL та IEEE.MATH\_COMPLEX. В останньому пакеті визначені типи, константи та функції, які оперують з комплексними даними.

### Завдання для роботи

1. Розробити програму на VHDL, яка будує графіки  $|H_1(z)|$  для різних значень  $|z|$  для експериментального визначення положення нулів та полюсів функції (4.11) з параметрами, заданими у табл. 4.2. Обчислити нулі та полюси функції (4.11) і порівняти їх з практично одержаними значеннями. Для визначених положень нулів та полюсів перевірити відношення (4.12) та те, що модуль нуля дорівнює оберненій величині модуля полюса, а їх кути співпадають.

2. Розробити програму на VHDL, яка будує графіки  $|H_3(\omega)|$ ,  $\arg(H_3(\omega))$  для заданої функції (4.13) і одержати ці графіки. Визначити ширину смуги пропускання, ширину перехідної смуги, рівень заглушення у смузі непропускання.

3. За допомогою програми на VHDL побудувати графіки  $|H_4(\omega)|$ ,  $\arg(H_4(\omega))$ ,  $H_4(z) = H_3(z^k)$  для заданого параметра  $k$ . Визначити ширину смуги пропускання, ширину перехідної смуги, рівень заглушення у смузі непропускання і порівняти їх з параметрами, які одержані в п.2.

Таблиця 1.2. Параметри та функції до лабораторної роботи

№ вар.	$a$	$b$	$H_2(z)$	$k$	$H_M(z)$
1	-0,3125	0,81	-1	2	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4})/5$
2	-0,125	0,81	-1	2	$(1 - z^{-1} + z^{-2} - z^{-3} + z^{-4})/5$
3	-0,625	0,81	-1	2	$(1 + 4z^{-1} + 6z^{-2} + 4z^{-3} + z^{-4})/16$
4	-0,875	0,81	-1	2	$(1 - 4z^{-1} + 6z^{-2} - 4z^{-3} + z^{-4})/16$
5	-0,3125	0,49	$-z^{-1}$	2	$(1 + 4z^{-1} + 6z^{-2} + 4z^{-3} + z^{-4})/16$
6	-0,75	0,49	$-z^{-1}$	2	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4})/5$
7	-0,5	0,49	$-z^{-1}$	2	$(-1 + 3z^{-1} + 5z^{-2} + 3z^{-3} - z^{-4})/9$
8	-0,25	0,49	$-z^{-1}$	2	$(1 + 5z^{-1} + 10z^{-2} + 10z^{-3} + 5z^{-4} + z^{-5})/32$
9	-0,8	0,49	$z^{-1}$	2	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6} + z^{-7})/8$
10	0,5	0,49	$z^{-1}$	2	$(1 + z^{-1})(1 + z^{-1} + z^{-2} + z^{-3})/8$
11	-0,25	0,25	$z^{-1}$	3	$(2 + 5z^{-1} + 7z^{-2} + 5z^{-3} + 2z^{-4})/21$
12	-0,5	0,25	$z^{-1}$	3	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5})/6$
13	-0,125	0,25	$z^{-1}$	3	$(1 + z^{-1} + 2z^{-2} + z^{-3} + z^{-4})/6$
14	-0,75	0,25	$z^{-1}$	3	$(1 - z^{-1} + z^{-3} - z^{-4})/4$
15	-0,75	0,25	$z^{-1}$	3	$(1 + 0.7z^{-1} - 0.7z^{-3} - z^{-4})/3.4$
16	-0,625	0,25	$z^{-1}$	3	$(1 - z^{-1} + z^{-2} - z^{-3} + z^{-4} - z^{-5})/6$
17	-0,75	0,25	$z^{-1}$	3	$(1 - 0.7z^{-1} + 0.7z^{-3} - z^{-4})/3.4$
18	-0,5	0,25	$z^{-1}$	4	$(1 + z^{-1} + z^{-3} + z^{-4})/4$
19	0,25	0,25	$z^{-1}$	4	$(-1 + z^{-1} + z^{-3} - z^{-4})/4$
20	-0,625	0,25	$z^{-1}$	4	$(1 - z^{-1} + z^{-3} - z^{-4})/4$
21	-0,5	0,25	$z^{-1}$	4	$(-1 + 2z^{-2} - z^{-4})/4$
22	-0,75	0,25	$z^{-1}$	4	$(1 + 2z^{-1} + 2z^{-2} + 2z^{-3} + z^{-4})/8$
23	-0,6	0,25	$z^{-1}$	4	$(1 - 2z^{-1} + 2z^{-2} - 2z^{-3} + z^{-4})/8$
24	0	0,25	$z^{-1}$	4	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} + z^{-5} + z^{-6})/7$
25	-0,1	0,25	$z^{-1}$	4	$(1 - z^{-2} + z^{-4} - z^{-6})/4$
26	-0,15	0,25	$z^{-1}$	4	$(1 + 1.4z^{-1} + z^{-2} - z^{-4} - 1.4z^{-5} - z^{-6})/6.8$
27	-0,5	0,25	$z^{-1}$	5	$(1 + z^{-1} + z^{-2} + z^{-3} + z^{-4})/5$
28	-0,6	0,25	$z^{-1}$	5	$(-3 - 2z^{-1} + 5z^{-2} + 5z^{-3} - 2z^{-4} - 3z^{-5})/20$
29	-0,75	0,25	$z^{-1}$	5	$(3 - 2z^{-1} + 5z^{-2} + 5z^{-3} - 2z^{-4} + 3z^{-5})/20$

4. Для заданої функції  $H_M(z)$  за допомогою програми на VHDL побудувати графіки  $|H_M(\omega)|$ ,  $\arg(H_M(\omega))$ , а також графіки  $|H(\omega)|$ ,  $\arg(H(\omega))$  для функції (4.14). Проаналізувати ширину смуги пропускання, рівень заглушення у смугі непропускання для різних значень параметра  $a$ . Дослідним шляхом визначити діапазон зміни

параметра  $a$ , у якому забезпечується заглушення не менше 20 дБ і такий параметр  $a$ , при якому досягається максимальне заглушення. Намалювати структуру фільтра.

5. Проаналізувати одержані графіки. Зробити висновки по роботі.

### Приклад виконання роботи

Нехай задано  $a = -0.5$ ,  $b = 0.64$ ,  $k = 2$ ,  $H_2(z) = z^{-1}$ ,

$$H_M(z) = \frac{(1+z^{-1})(1-z^{-7})}{7(1-z^{-1})}.$$

Для обчислення передаточних функцій розроблено ряд функцій з використанням пакетів IEEE.MATH\_REAL та IEEE.MATH\_COMPLEX. Обчислення послідовності  $z^n$  виконується з використанням наступних функцій, які перевантажують одна одну

--число  $Z$  радіусом  $MAG$  в степені кута  $j*fi$

```
function Z(mag,fi:real) return COMPLEX_POLAR is
begin
```

```
    return mag*exp(COMPLEX_TO_POLAR(MATH_CBASE_J)*(fi));
```

```
end Z;
```

--число  $Z$  радіусом 1 в степені кута  $j*fi$

```
function Z(fi:real) return COMPLEX_POLAR is
begin
```

```
    return exp(COMPLEX_TO_POLAR(MATH_CBASE_J)*(fi));
```

```
end Z;
```

Наступна функція обчислює  $H_1(z)$  за формулою (4.11), але для позитивних степенів  $z$ , тобто з урахуванням (4.6). Вона необхідна для сканування простору  $|H_1(z)|$  при пошуку нулів та полюсів, коли встановлюється різна величина  $mag$  амплітуди та фази  $fi$  комплексного вектора  $z$ . Також вона використовується для розрахунків АЧХ та ФЧХ.

-- обчислює  $(bZ^2+a(b+1)Z+1)/(Z^2+c(b+1)Z+b)$

```
function Allpass2(b, a: real; --коефіцієнти
    mag, fi: real) -- амплітуда, кут для  $z$ 
return COMPLEX_POLAR is
variable tn,td: COMPLEX_POLAR;
```

```
begin
```

```
    td:=COMPLEX_TO_POLAR(COMPLEX'(b,0.0))
    +a*(b+1.0)*Z(mag,fi) + Z(mag**2,2.0*fi);
    tn:=COMPLEX_TO_POLAR(COMPLEX'(1.0,0.0))
    +a*(b+1.0)*Z(mag,fi) + b*Z(mag**2,2.0*fi);
    return tn/td ;
```

```
end Allpass2;
```

Наступна функція обчислює  $H_1(z^2)$  (1.11) для від'ємних степенів  $z$  з одиничною амплітудою, тобто вона обчислює  $H_1(\omega)$ , де  $\omega = \pi$ . Тому вона використовується лише для розрахунків АЧХ та ФЧХ.

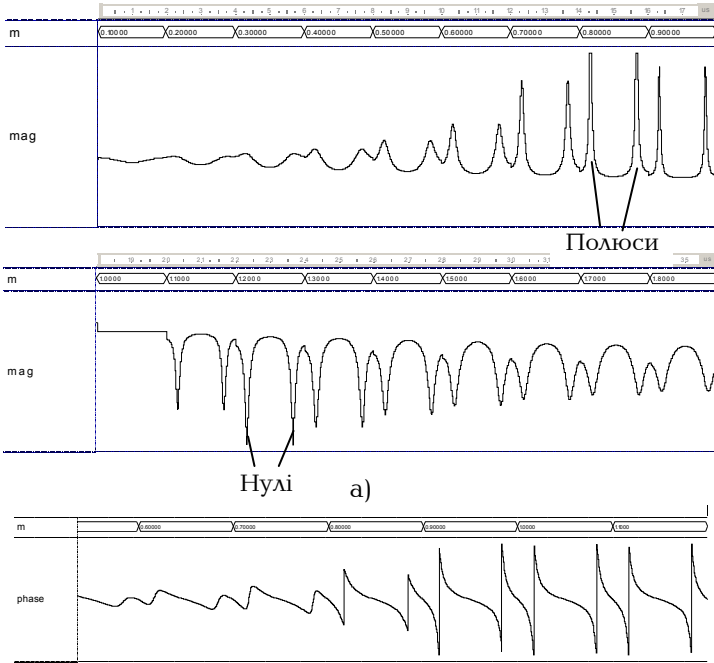
```
function Allpass2x2(b,c:real; fi:real) return COMPLEX_POLAR is
  variable tn,td: COMPLEX_POLAR;
begin
  tn:=COMPLEX_TO_POLAR(COMPLEX'(b,0.0))
  +c*(b+1.0)*Z(-2.0*fi) + Z(-4.0*fi);
  td:=COMPLEX_TO_POLAR(COMPLEX'(1.0,0.0))
  +c*(b+1.0)*Z(-2.0*fi) + b*Z(-4.0*fi);
  return tn/td ;
end Allpass2x2;
```

Для сканування простору  $|H_1(z)|$  використовується наступний фрагмент VHDL-програми.

```
signal clk:bit;
signal m,ph:real:=0.0;
signal a,b,logm,phase,mag:real:=0.0;
begin
  clk<=not clk after 5ns; --генератор синхросигналів
  process(CLK)
    variable p,phas:real:=0.0;
    variable Hz: COMPLEX_POLAR;
  begin
    a<=-0.5; b<=0.64; -- параметри H(z)
    if clk='1' and clk'event then
      phas:= phas+0.001; -- лічильник фази (частоти)
      p:=phas*MATH_PI*2.0; -- нормована фаза
      m<= trunc(phas)*0.1+0.1; -- амплітуда Z
      ph<=phas; -- сигнал фази
    end if;
    Hz:=Allpass2(b,a,m,p); --власне H(z)
    mag<=abs(Hz); -- амплітуда H(z)
    phase<=Hz.ARG; -- фаза H(z)
    logm<=20.0*log10(abs(Hz)); -- H(z) у децибелах
  end process;
```

Після запуску програми на моделювання одержано графік  $|H_1(z)|$  (рис. 4.9), у якому на кількох інтервалах, позначених  $m$ , розміщено криві  $|H_1(m,ph)|$ , де  $m$  – довжина,  $ph$  – нормований кут (фаза, частота) вектора  $z$ , причому  $m = 0.1, 0.2, \dots$ , повний оберт вектора  $z$  відбувається коли  $ph$  – ціле число. Перша половина  $|H_1(m,ph)|$  – для  $m = 0.1, \dots, 0.9$ , а друга половина – для  $m = 1.0, \dots, 1.9$ . Таким чином, кожен інтервал відповідає замкненій кривій, як на рис. 4.6. Там же аналогічно показано графік  $\arg(H_1(z))$  для  $m = 0.6, \dots, 1.1$ .

Аналіз графіків на рис. 9 показує, що полюси функції  $H_1(z)$  мають фазу  $\pm 2\pi \cdot rh = \pm 2\pi \cdot 0.1645 = \pm 1.0336$  і знаходяться в точках  $r_1 = 0.8 \angle 1.0336$  та  $r_2 = 0.8 \angle -1.0336$ . Нулі функції  $H_1(z)$  мають таку саму фазу і дорівнюють  $q_1 = 1.3 \angle 1.0336$ ,  $q_2 = 1.3 \angle -1.0336$ . Слід відмітити, що  $-\cos(1.0336) = -0.512 \approx a$ ,  $1/0.8 = 1.25 \approx 1.3$  та  $0.8^2 = 0.64 = b$ , тобто справджуються відношення (4.12).



б)

Рис. 9. Розгортка функції  $|H_1(z)|$  (а) та функції  $\arg(H_1(z))$  (б)

Аналіз графіка  $\arg(H_1(z))$  показує, що при куті  $rh = 0$  фаза дорівнює 0, при поступовому збільшенні  $rh$  фаза плавно зменшується. Але якщо  $rh$  наближається до положення полюсів, фаза стрімко досягає значення  $\pm\pi$ .

Для обчислення  $|H_3(\omega)|$  у попередній програмі один оператор присвоювання змінній замінюється на наступний оператор, що відповідає формулі (4.13) .



$$Hz:=(Allpass2(b,a,1.0,p)+Z(-1.0*p))/2.0 .$$

Тут  $H_2(z) = z^{-1}$  – теж передаточна функція фазового фільтра, тому що  $|H_3(z)| = 1$  для усіх  $z$ .

Після запуску програми на моделювання одержано графіки  $|H_3(\omega)|$  та  $\arg(H_3(\omega))$  (рис. 4.10) для частот  $\omega = (0, 2\pi)$  або для нормованих частот  $ph = (0, 1)$ . Як бачимо, додавання передаточної функції  $H_2(z)$  створює АЧХ фільтра нижніх частот з частотою зрізу  $f_{зр} = 0,156 \approx \arg(r_1) = \frac{1}{2\pi} 1.0336 = 0,164$  частки частоти дискретизації. Вимірний за графіком  $\log_{10}$  рівень заглушення складає 8.5 дБ.

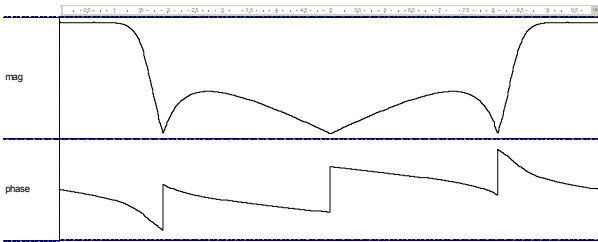


Рис. 10. Графіки АЧХ  $|H_3(\omega)|$  та ФЧХ  $\arg(H_3(\omega))$

Так само, як і у попередньому випадку, графік  $|H_4(\omega)|$  (рис.4.11,а) будується за допомогою заміни оператора на наступний оператор

$$Hz:=(Allpass2x2(b,a,p)+Z(-2.0*p))/2.0;$$

$$\begin{aligned} \text{Графік } |H_M(\omega)| \text{ (рис.4.11,б) будується за допомогою оператора} \\ Hz:=(COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))+Z(-p))* \\ (COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))-Z(-7.0*p)) \\ /(COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))-Z(-p))/14.0; \end{aligned}$$

Видно, що цей графік представляє собою повторений двічі графік на рис.4.10, що є характерним для передаточних функцій з кратними затримками.

Графік результуючої АЧХ  $|H_M(\omega)|$  (рис.4.11,в) будується за допомогою оператора

$$\begin{aligned} Hz:=(Allpass2x2(b,a,p)+Z(-2.0*p))* \\ (COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))+Z(-p))* \\ (COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))-Z(-7.0*p)) \\ /(COMPLEX\_TO\_POLAR(COMPLEX'(1.0,0.0))-Z(-p))/28.0; \end{aligned}$$

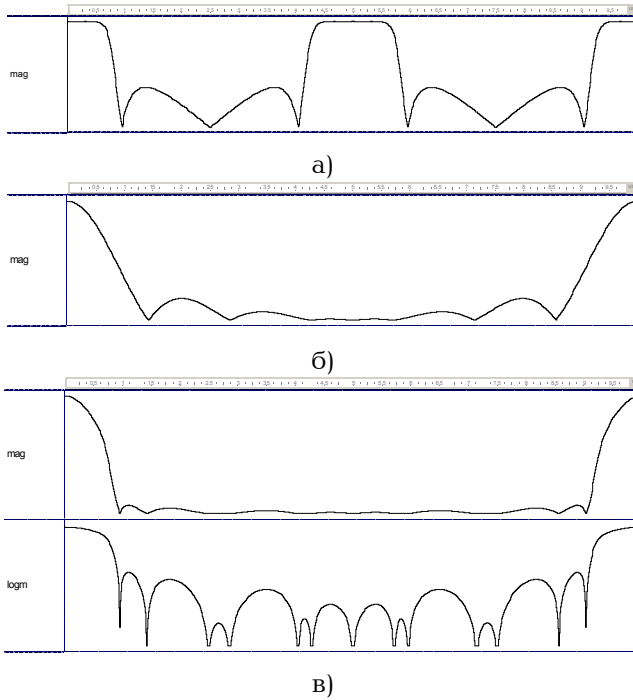


Рис. 1.11. Графік АЧХ  $|H_4(\omega)|$  (а),  $|H_M(\omega)|$  (б) та результуючої АЧХ  $|H(\omega)|$  в лінійному та логарифмічному масштабах

Як свідчить рис. 4.11, функція  $H_M(z)$  маскує  $H_4(z)$ , в результаті чого одержано передаточну функцію  $H(z)$  (1.5) фільтра нижніх частот з частотою зрізу  $f_{зр} = 0,06$  від частоти дискретизації та рівнем заглушення більше 23 дБ.

## Лабораторна робота 3

### Фільтри зі скінченною імпульсною характеристикою

**Мета роботи:** одержати знання про фільтри зі скінченною імпульсною характеристикою та навички їх розрахунку, програмування та моделювання за допомогою VHDL.

#### Теоретичні відомості

Такі лінійні системи, як фільтри, поділяються на СІХ- та НІХ-фільтри (див. підрозділ 2.2). СІХ-фільтри мають ряд наступних переваг:

- СІХ-фільтри завжди стабільні, тому що у них немає полюсів і зворотніх зв'язків, через які відбувається збудження;
- завжди гарантована побудова фільтра з лінійною фазою;
- помилки округлення в СІХ-фільтрах мають значно менший вплив на результат, ніж у НІХ-фільтрах;
- СІХ-фільтри ефективно реалізуються в мікропроцесорах, які мають команди множення з додаванням до акумулятора.

АЧХ  $|H(\omega)|$  фільтра задає конкретний коефіцієнт підсилення на певних частотах, а ФЧХ  $\varphi(\omega) = \arg(H(\omega))$  – вплив на затримку чи зсув фази сигналу на цих частотах. Функція **групової затримки** визначена як

$$T_d(\omega) = - \frac{d\varphi(\omega)}{d\omega}. \quad (1.15)$$

Фільтр з лінійною фазою має ФЧХ, яку можна виразити формулою:

$$\varphi(\omega) = - a\omega \text{ або } \varphi(\omega) = \pi - a\omega. \quad (1.16)$$

Тому, згідно з (4.15), у фільтра з лінійною фазою групова затримка сигналу є сталою  $a$  для усіх частот. Такий фільтр не має перекручування фази сигналу і, відповідно, не втрачає своєї форми (якщо його спектр попадає у смугу пропускання), тому що усі його частотні компоненти затримуються на одну і ту саму затримку.

За формою АЧХ фільтри діляться на фільтри нижніх частот (ФНЧ), високих частот (ФВЧ), смугові фільтри (СФ) та режекторний (смугово-непропускний) фільтр (РФ). Через те, що АЧХ фільтра з реальними коефіцієнтами є симетричною функцією від  $\omega$ , така АЧХ задається у діапазоні  $0 \leq \omega \leq \pi$ .

АЧХ ідеального ФНЧ показана на рис. 4.12,а. Діапазони  $0 \leq \omega \leq \omega_{зр}$  та  $\omega > \omega_{зр}$  називаються **смугами пропускання** та

**непропускання**, відповідно, а частота  $\omega_{зр}$ , яка розділяє ці смуги – **частотою зрізу**. Ідеальний фільтр нижніх частот у діапазоні  $0 \leq \omega \leq \omega_{зр}$  має АЧХ  $|H(\omega)| = 1$ , а у діапазоні  $\omega > \omega_{зр}$  –  $|H(\omega)| = 0$  (рис. 4.12,а).

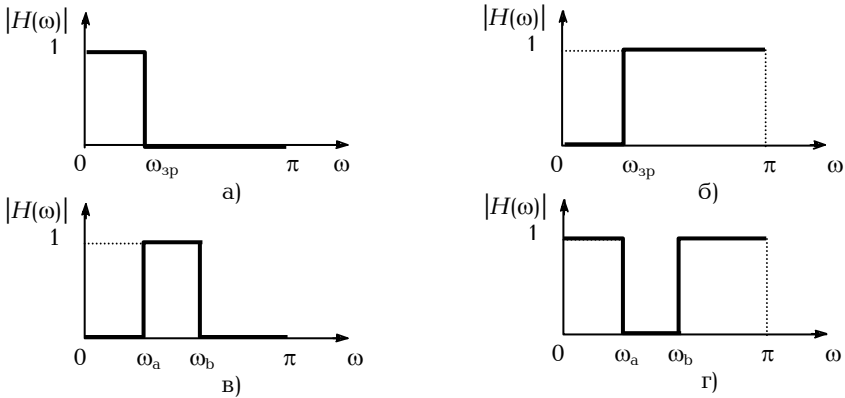


Рис. 1.12. Графіки АЧХ фільтрів нижніх частот (а), верхніх частот (б), смугового (в) та режекторного (г) фільтрів

В АЧХ смугового фільтра частоти  $\omega_a$ ,  $\omega_b$  називаються нижньою та верхньою частотами зрізу. Такий фільтр пропускає частоти у діапазоні  $(\omega_a, \omega_b)$  і заглушує частоти в інших діапазонах (рис. 4.12,в). Режекторний (смугово-непропускаючий) фільтр – навпаки – заглушує частоти в діапазоні  $(\omega_a, \omega_b)$  (рис. 4.12,г).

У попередній лабораторній роботі приділялась увага фазовому фільтру, у якого  $|H(\omega)| = 1$  для усіх частот. Особливим випадком фазового фільтру є **фільтр Гільберта**, який виконує зсув фази вхідного сигналу на кут  $\pi/2$ .

Існують багатосмугові фільтри, які мають кілька смуг пропускання та непропускання. Гребінчастий фільтр – це різновид багатосмугового фільтра, який має кілька смуг, що рівновіддалені за частотою, і тому його АЧХ за формою нагадує гребінець.

Насправді неможливо досягнути ідеальної форми АЧХ, як на рис. 1.12. На практиці АЧХ задається вихідними даними смуг пропускання  $(0, \omega_n)$ , непропускання  $(\omega_n, \pi)$ , **перехідної смуги**  $(\omega_n, \omega_n)$  рівнів пульсацій у смузі пропускання  $\delta_1$  та у смузі непропускання  $\delta_2$  (рис. 1.13).

Ці рівні пульсацій, як правило, задаються у відносних одиницях – децибелах як похибка амплітуди у смузі пропускання  $A_1$  та мінімальний рівень заглушення фільтра  $A_2$ :

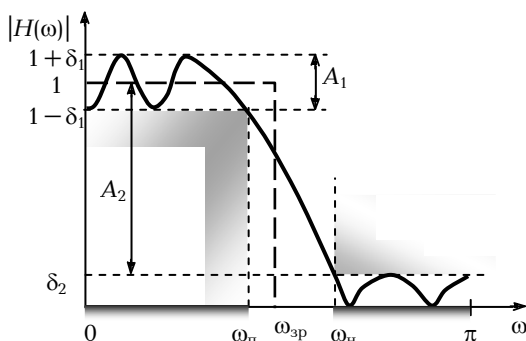


Рис. 1.13. Графік АЧХ реального фільтра нижніх частот

$$A_1 = 20 \log \left( \frac{1 + \delta_1}{1 - \delta_1} \right), \text{ дБ}; \quad A_2 = 20 \log \delta_2, \text{ дБ}. \quad (1.17)$$

Згідно з (2.5), передаточна функція СІХ-фільтра визначається формулою

$$H(z) = \sum_{r=0}^M b_r z^{-r} \quad (1.18)$$

Якщо  $M$  – парне число, тобто  $N = 2M$ , то (4.16) можна переписати як

$$H(z) = \sum_{r=-N}^N b_{r+N} z^{-r-N} = z^{-N} \sum_{r=-N}^N h_r z^{-r}. \quad (1.19)$$

Нехай коефіцієнти  $h_r$  мають симетрію:  $h_r = h_{-r}$ . Тоді АЧХ, згідно з (4.8) та (4.19) обчислюється за формулою

$$H(\omega) = e^{-j\omega N} \sum_{r=-N}^N h_r e^{-j\omega r} = e^{-j\omega N} \left( h_0 + 2 \sum_{r=1}^N h_r \cos(\omega r) \right), \quad (1.20)$$

де множник  $e^{-j\omega N}$  означає лише затримку на  $N$  тактів і може бути усунений з аналізу.

Якщо  $M$  – непарне число, то в (4.20) буде відсутній окремий член  $h_0$ . Якщо коефіцієнти імпульсної реакції  $h_r$  – реальні, то і  $H(\omega)$  – реальна функція. ФЧХ такого фільтра дорівнює

$$\varphi(\omega) = \begin{cases} -\omega N & \text{при } H(\omega) \geq 0, \\ \pi - \omega N & \text{при } H(\omega) < 0. \end{cases} \quad (1.21)$$

Отже при умові симетрії імпульсної реакції ФЧХ є лінійною, і через кожні  $\pi$  радіан зміни кута  $\varphi(\omega)$  значення АЧХ змінює знак. Навіть коли імпульсна реакція – антисиметрична, тобто коли

$h_0 = 0, h_r = -h_{-r}$ , ФЧХ є лінійною. Згідно з (4.15), групова затримка такого СІХ-фільтра дорівнює  $Td(\omega) = M/2$ , тобто вона відповідає середньому члену послідовності імпульсної реакції.

Властивості симетрії імпульсної реакції, як правило, використовуються для зменшення обсягів обчислень. Згідно з (2.2), СІХ-фільтр обчислюється як

$$y(n) = \sum_{r=0}^M b_r x(n-r). \quad (4.22)$$

При урахуванні симетрії коефіцієнтів  $b_r$  вираз (4.20) можна переписати як

$$y(n) = \sum_{r=0}^{M/2-1} b_r [x(n-r) + x(n-M+1+r)]. \quad (4.23)$$

Порівнявши (4.22) і (4.23), видно, що кількість множень у останньому випадку зменшена удвічі. Сигнальний граф такого фільтра для випадку непарного  $M$  показано на рис. 4.14. При парному  $M$  сигнальний граф фільтра має на один помножувач менше.

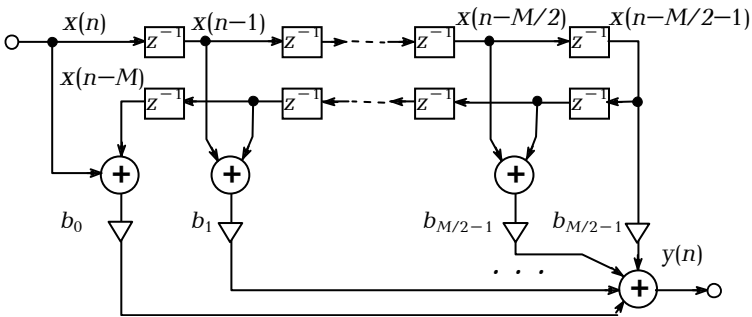


Рис. 4.14. Сигнальний граф СІХ-фільтра з симетричною імпульсною реакцією непарної довжини

СІХ-фільтри з різними АЧХ відрізняються значеннями та кількістю своїх коефіцієнтів  $b_i$ . Існують кілька методів синтезу набору цих коефіцієнтів. Найбільш універсальним та ефективним є метод Паркса й Маклеллана, оснований на процедурі поліноміальної оптимізації Ремеза.

Початковими даними для синтезу коефіцієнтів  $b_i$  або  $h_i$  (4.19) цим методом є їх число  $M$ , діапазони частот пропускання  $(0, \omega_p)$  та непропускання  $(\omega_s, \pi)$ , як наприклад, для ФНЧ, похибка амплітуди у смузі пропускання  $A_1$ , мінімальний рівень заглушення фільтра  $A_2$  (4.17) або ступінь важливості мінімізації похибок  $\delta_1, \delta_2$  (рис. 4.12).

Згідно з методом, АЧХ (4.20) представляється як поліном

$$H(\omega) = \sum_{i=0}^N a_i \cos^i(\omega). \quad (4.24)$$

Метод полягає у виборі ряду  $N+1$  частот  $\omega_k$ , складанні та вирішенні системи рівнянь (4.24). Результатами рішень цих рівнянь є коефіцієнти  $a_i$ , які на частотах  $\omega_k$  дають похибку  $\delta_1 = 0$  або  $\delta_2 = 0$ . Після цього обчислюється АЧХ для інших частот, що знаходяться між  $\omega_k$  та  $\omega_{k+1}$ . Якщо похибка такої поліноміальної апроксимації не переважає  $A_1$  та  $A_2$ , то за коефіцієнтами  $a_i$  знаходяться шукані  $b_i$ . Якщо ні — то за певним алгоритмом вибираються інші частоти  $\omega_k$  і процес оптимізації повторюється, починаючи з вирішення системи рівнянь. При використанні замість  $A_1, A_2$  ступенів важливості мінімізації похибок, наприклад,  $w_1, w_2$ , процес оптимізації націлений на мінімізацію критерія  $w_1 \max(\delta_1) + w_2 \max(\delta_2)$ .

Попереднє значення  $M$  можна оцінити за емпіричною формулою [4]:

$$M \approx \frac{-10 \lg(\delta_1 \cdot \delta_2)}{2,324 \Delta\omega} - 13, \quad (4.25)$$

де  $\Delta\omega = |\omega_n - \omega_{n-1}|$  — ширина перехідної смуги. Якщо після обчислення коефіцієнтів фільтру не досягаються гідні рівні  $A_1, A_2$ , то збільшується число  $M$  або збільшується дистанція між  $\omega_n$  та  $\omega_{n-1}$ . Оскільки поведінка АЧХ у перехідній смузі не контролюється за методом, то при надмірних значеннях  $|\omega_n - \omega_{n-1}|$ , можливі небажані перекручення АЧХ.

Метод Паркса й Маклеллана реалізований програмно у багатьох засобах математичних САПР. У даній роботі рекомендується застосувати систему Scilab, яка функціонально еквівалентна системі Matlab, але на відміну від останньої, є безкоштовною. Обчислення коефіцієнтів смугового фільтра виконується за наступними командами.

```
hn=eqfir(M, [0 f1; f2 f3; f4 0.5], [0 1 0], [1 1 1]);
[hm,fr] = frmag(hn, 256);
plot(fr,hm);
hn=
```

За першою командою знаходиться вектор коефіцієнтів  $h_i$ , число яких  $M$ . Діапазони частот задані двійками  $0 f1; f2 f3; f4 0.5$ , де частоти  $f_i$  задаються частками повного кута, який нормований до одиниці, тобто розрахункова частота  $f = \omega/2\pi$ . Числа у векторі  $[0 1 0]$  дорівнюють  $H(f)$  у відповідних діапазонах частот, а у векторі  $[1 1 1]$  — коефіцієнтам важливості  $w_i$ . Другий оператор обчислює таблицю з 256 аргументів і значень  $H(f)$ , а третій — виконує вивід графіка  $H(f)$ . За четвертим оператором виводиться вектор знайдених коефі-

цієнтів фільтра. Кількість елементів векторів у першому операторі залежить від виду фільтра. Так, у ФНЧ їх два, а у гребінчастого фільтра — подвоєна кількість смуг пропускання.

У попередній лабораторній роботі АЧХ системи обчислювалась на основі алгоритму ЦОС, заданого формулою (2.2), шляхом підстановки (4.9) у формулу відповідної передаточної функції (2.5). Однак такий спосіб не підходить для визначення АЧХ готового фільтра. АЧХ такого фільтра, як правило, відрізняється від АЧХ, яка визначена за формулою, через квантування даних та коефіцієнтів, додавання похибок обчислень, нарешті, через несправність або відмову елементів фільтра. Тому визначення АЧХ фільтра є типовою процедурою його діагностики та тестування.

Для аналізу чи вимірювання АЧХ та ФЧХ на вхід системи, що перевіряється, слід подавати комплексний сигнал  $e^{-j\omega n}$  (4.8), який називають **аналітичним**. Відповідно, система повинна бути спроможною обробляти такий сигнал. Але переважна більшість технічних систем ЦОС призначені для обробки лише реальних сигналів.

Часто використовують простий спосіб перевірки, коли замість аналітичного сигналу подають лише його складову —  $\cos(\omega n)$ , а АЧХ вимірюють на виході системи, як максимум результуючого сигналу  $Re(H(\omega))$ , тобто в моменти, коли друга складова —  $\sin(\omega n) = 0$ . Недолік цього способу — у неточності вимірювання максимуму сигналу.

Більш точний спосіб полягає у одержанні уявної складової  $Im(H(\omega))$  після пропускання результату  $Re(H(\omega))$  через фільтр Гільберта. Але такий фільтр вносить суттєві перекручення в частотну характеристику при  $\omega \rightarrow 0$ .

Тому для аналізу систем ЦОС слід застосувати сигнальний граф на рис. 4.15. В ньому використовуються два ідентичних екземпляри системи  $H(z)$ , на які подаються синусна та косинусна складова аналітичного сигналу. Відповідно, з виходів систем знімаються складові аналітичного сигналу відгуку системи. На відміну від аналогових систем, для такого дослідження завжди можна одержати дві ідентичні цифрові системи.

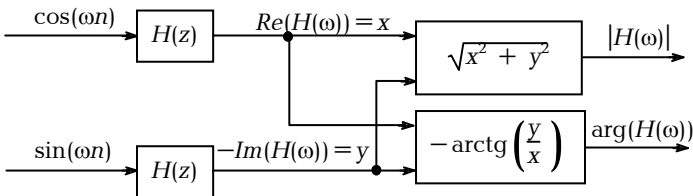


Рис. 4.15. Вимірювання АЧХ, ФЧХ реальної системи  $H(z)$



Переважає більшість СІХ-фільтрів обчислюється в комп'ютерах у арифметиці цілих чисел. При програмуванні такого фільтру синтезують набір коефіцієнтів, що задовольняють задані вимоги і представлені з плаваючою комою. Потім вибирають кількість бітів квантування коефіцієнтів  $n_k$ , вхідних даних  $n_x$  та результатів  $n_y$ .

Як правило,  $n_x, n_y \geq \log_2 10 \cdot D/20$ , де  $D$  – динамічний діапазон зміни сигналу, дБ. Тобто на кожні 6 децибел динамічного діапазону припадає не менше одного розряду. Коефіцієнти масштабують та округлюють, так що цілі коефіцієнти дорівнюють

$$b'_i = \lfloor 2^{n_x} \cdot b_i + 0.5 \rfloor . \quad (4.26)$$

Результати фільтра обчислюють за формулою (4.22) чи (4.23) з вибором такої розрядності суми, щоб не виникало переповнення. Причому розрядність добутку дорівнює  $n_d = n_k + n_x$ , а розрядність суматора повинна бути не меншою за  $n_c = \log_2 S + n_k + n_x$ , де  $S$  – теоретично можливий максимальний результат формули (4.22). Нескладно довести, що  $S$  дорівнює сумі модулів усіх коефіцієнтів фільтра, тобто

$$n_c = \log_2 \left( \sum_{i=0}^M |b_i| \right) + n_k + n_x. \quad (4.27)$$

Оскільки  $n_c$  може бути доволі великим числом, а вірогідність досягнення результатом (4.22) максимального значення досить мала, то на практиці  $n_c$  вибирають дещо меншим, а додавання в (4.22) виконують за алгоритмом накопичення з **насищенням**. За цим алгоритмом, якщо виникає переповнення суми, то результат замінюється максимальним числом розрядності  $n_c$  з відповідним знаком. Результат фільтра  $y(n)$  береться як старші  $n_y$  розрядів суми (4.22) з відкиданням молодших розрядів, тобто з усіканням.

Округлення коефіцієнтів та усікання результату незворотно перекручують АЧХ фільтра. Тому слід обчислити АЧХ та ФЧХ за сигнальним графом на рис. 4.15 і порівняти їх з початковими характеристиками. Якщо характеристики фільтра не задовольняють задані вимоги, то слід збільшити значення  $n_k$  та  $n_c$ , можливо –  $M$  і повторити цикл синтезу коефіцієнтів. Іноді вдається покращити АЧХ, завдяки корекції молодших розрядів коефіцієнтів  $b'_i$ .

### Завдання для роботи

1. Синтезувати за допомогою системи Scilab набір коефіцієнтів СІХ-фільтра з симетричною імпульсною характеристикою, який задовольняє задані параметри АЧХ. Параметри фільтра вибрати з табл. 4.3 за варіантом. Порядок фільтра визначити за (4.25), але не більше 60. Показати на графіку АЧХ, що фільтр задовольняє завдання.

2. Написати програму моделі СІХ-фільтра з синтезованим набором коефіцієнтів, що виконує обчислення з плаваючою комою. Ввімкнути модель у стенд для випробувань, згідно з рис. 4.15 та одержати графіки АЧХ і ФЧХ фільтра. Порівняти графік АЧХ з графіком, який одержано в п. 1.

3. Представити коефіцієнти фільтра цілими числами згідно з (4.24). Написати програму моделі СІХ-фільтра з синтезованим набором коефіцієнтів, що виконує обчислення з цілими числами. Перевірити відношення (4.27), яке не повинне бути більшим за 32. Ввімкнути модель у стенд для випробувань, згідно з рис. 4.15 та одержати графіки АЧХ і ФЧХ фільтра. Порівняти графіки з графіками, які одержано в п. 2.

4. Підключити фільтр до виходу генератора сигналів, який розглядався у першій лабораторній роботі і одержати графік результуючого сигналу. Пояснити, чому він має перевернутий форму.

5. Проаналізувати одержані графіки. Зробити висновки по роботі.

Для реалізації стенда для випробувань доцільно використати готові віртуальні модулі FilterTB, які додаються до роботи. Наприклад, модуль FilterTB\_r призначений для тестування модулів фільтрів порти яких пропускають сигнали з плаваючою комою. Такий модуль має наступний інтерфейс.

```
entity FilterTB is
  generic(fsampl:integer := 1000;
          fstrt:integer:=0;
          deltaf:integer:=20;
          maxdelay:integer:=100;
          slowdown:integer:=3;
          magnitude:real:=1000.0);
  port( CLK      : in  STD_LOGIC;
        RST      : in  STD_LOGIC;
        START    : in  STD_LOGIC;
        RERSP    : in  real;
        IMRSP    : in  real;
        REO      : out real;
        IMO      : out real;
        FREQ     : out integer;
        MAGN     : out real;
        LOGMAGN  : out real;
        PHASE    : out real;
        ENA     : inout STD_LOGIC);
end FilterTB;
```

Аналогічний інтерфейс мають модулі FilterTB\_I, FilterTB\_V для тестування фільтрів сигналами, що представляються цілими числами та векторами бітів. Призначення настроювальних констант і портів роз'яснено у таблиці 4.4.

Таблиця 1.3. Параметри фільтра до лабораторної роботи

№ вар.	Вид фільтра	$A_1, \text{дБ}$	$A_2, \text{дБ}$	$f_{\text{п}}, f_{\text{п1}}$	$f_{\text{п}}, f_{\text{п1}}$	$f_{\text{п2}}$	$f_{\text{п2}}$	$n_x, n_y$	$n_k$
1	ФНЧ	0,1	60	0,3	0,42			16	12
2	ФНЧ	0,2	57	0,25	0,4			16	12
3	ФНЧ	0,4	54	0,25	0,35			16	12
4	ФНЧ	1	51	0,2	0,4			15	11
5	ФНЧ	2	48	0,2	0,3			14	10
6	ФНЧ	3	45	0,15	0,25			13	9
7	ФНЧ	4	42	0,15	0,2			12	8
8	ФВЧ	0,1	60	0,42	0,3			16	12
9	ФВЧ	0,2	57	0,4	0,25			16	12
10	ФВЧ	0,4	54	0,35	0,25			16	12
11	ФВЧ	1	51	0,4	0,2			15	11
12	ФВЧ	2	48	0,3	0,2			14	10
13	ФВЧ	3	45	0,25	0,15			13	9
14	ФВЧ	4	42	0,2	0,1			12	8
15	СФ	0,5	51	0,3	0,2	0,35	0,45	16	12
16	СФ	1	48	0,25	0,15	0,35	0,45	16	12
17	СФ	2	45	0,25	0,15	0,3	0,4	16	12
18	СФ	3	42	0,2	0,1	0,3	0,4	15	11
19	СФ	4	51	0,2	0,1	0,3	0,4	14	10
20	СФ	5	48	0,15	0,05	0,25	0,35	13	9
21	СФ	6	45	0,1	0,03	0,2	0,28	12	8
22	РФ	0,5	51	0,3	0,39	0,49	0,41	16	12
23	РФ	1	48	0,25	0,35	0,49	0,40	16	12
24	РФ	2	45	0,25	0,35	0,45	0,38	16	12
25	РФ	3	42	0,2	0,3	0,49	0,4	15	11
26	РФ	4	51	0,2	0,3	0,45	0,35	14	10
27	РФ	5	48	0,15	0,25	0,40	0,30	13	9
28	РФ	6	45	0,1	0,2	0,4	0,3	12	8
29	РФ	0,5	42	0,1	0,15	0,35	0,25	12	8

Таблиця 4.4. Віртуальний модуль для випробувань фільтрів

fsampl	Частота дискретизації $f_o$ , наприклад 1000 кГц
fstrt	Початкова частота діапазону аналізу
deltaf	Приріст частоти $d$ , так що через $k$ кроків генератор видаватиме частоту $f_o + k \cdot d$
maxdelay	Затримка у кількості відліків від початку генерації частоти, після якої виконується оцінка результатів фільтрації. Має бути більшою за подвоєну максимальну групову затримку фільтра, який тестується.
slowdown	Коефіцієнт зменшення швидкості фільтра відносно тактової частоти. Якщо відіні відліки поступають у кожному такті, то $slowdown=1$ , якщо вони приходять у кожному парному такті, то $slowdown=2$ і т.д.
magnitude	Амплітуда синусного і косинусного сигналів, що генеруються.
REO,IMO	Синусний та косинусний сигнали, що генеруються
RERSP, IMRSP	Сигнали з виходів фільтрів, які є відгуками на синусний та косинусний сигнали, відповідно
FREQ	Код частоти сигналів, що генеруються, який дорівнює $f_o + k \cdot d$
MAGN	Обчислена амплітуда комплексного сигналу RERSP, IMRSP на частоті FREQ
LOGMAGN	Обчислена амплітуда комплексного сигналу RERSP, IMRSP на частоті FREQ у логарифмічному масштабі, тобто у децибелах, причому сигнал з амплітудою magnitude приймається за 0 дБ.
PHASE	Обчислена фаза комплексного сигналу RERSP, IMRSP на частоті FREQ, представлена у діапазоні $\pm\pi$
RST,START	Сигнали початкового встановлення
ENA	Сигнал дозволу прийому RERSP, IMRSP при $slowdown > 1$

### Приклад виконання роботи

Нехай задано: ФНЧ,  $A_1 = 1$  дБ,  $A_2 = 48$  дБ,  $f_{п1} = 0.21$ ,  $f_{п2} = 0.29$ ,  $n_x = 12$ ,  $n_k = 10$ .

Знайдено  $\Delta\omega = 2\pi(0.29 - 0.21) = 0.5027$ . Вирішивши рівняння (4.17), знайдено  $\delta_1 = 0.05$ ,  $\delta_2 = 0.004$ . Після підстановки цих значень в (4.25) одержано  $M = 21$ .

Сформовано команду для Scilab щоб знайти набір коефіцієнтів

$$hn=eqfir(21, [0 0.21; 0.29 0.5], [1 0], [4 50]);$$

де вагові коефіцієнти [4 50] – відповідають відношенню  $\delta_1$  та  $\delta_2$ .

Після невдалої спроби одержати коефіцієнти, що задовольняють вимоги, було збільшено кількість коефіцієнтів  $M = 22$ .

Розроблено наступний модуль СІХ-фільтра, в якому використані знайдені коефіцієнти фільтра.

```

library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity FIR_filter is           -- СІХ-фільтр парного порядку
    generic(nn:natural:=22); -- порядок фільтра
    port(CLK : in STD_LOGIC;
         RST : in STD_LOGIC;
         X : in REAL;           -- вхід фільтра
         Y : out REAL);       -- вихід фільтра
end FIR_filter;
architecture model of FIR_filter is
    constant m: natural:=nn/2; -- кількість коефіцієнтів
    type Tarr is array (0 to m-1) of real;
    constant b: Tarr:= (           -- набір коефіцієнтів
        0.0099895, 0.0238865, 0.0071111, - 0.0255226, - 0.0145117,
        0.0407322, 0.0307317, - 0.0684982, - 0.0734177, 0.1589393,
        0.4370358);
    signal xf,xb:Tarr; -- ланцюжки затримок вхідного сигналу
begin
    DELAY:process(CLK,RST) begin
        if RST='1' then
            xf<=(others=>0.0);
            xb<=(others=>0.0);
        elsif CLK='1' and CLK'event then
            xf<=X & xf(0 to m-2);
            xb<=xb(1 to m-1) & xf(m-1);
        end if;
    end process;

    ADD_MUL:process(xf,xb)
        variable acc:real;
    begin
        acc:=0.0;
        for i in 0 to m-1 loop
            acc:=acc + b(i)*(xf(i)+xb(i));
        end loop;
        Y<=acc;
    end process;
end model;

```

Модуль фільтра був вставлений у сенд для випробувань, що показаний на рис. 4.16. За допомогою цього сенда було побудовано графіки АЧХ та ФЧХ, які показано на рис.4.17.

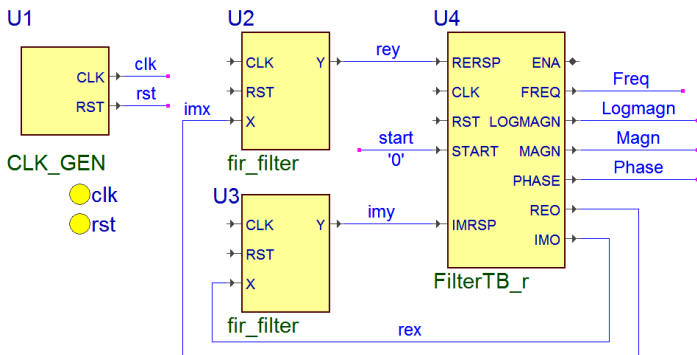


Рис.4.16. Стенд для іспитів СІХ-фільтра

За допомогою фільтра було профільтровано сигнал з генератора, який було побудовано в першій лабораторній роботі. Результат фільтрації показано на рис. 4.18.

Як видно, фільтр типу ФНЧ дещо згладжує різкі переходи сигналу. Невеликі коливання в межах різких переходів пов'язані з недостатнім притисненням сигналу з частотою, яка близька до 0,5 частоти дискретизації.

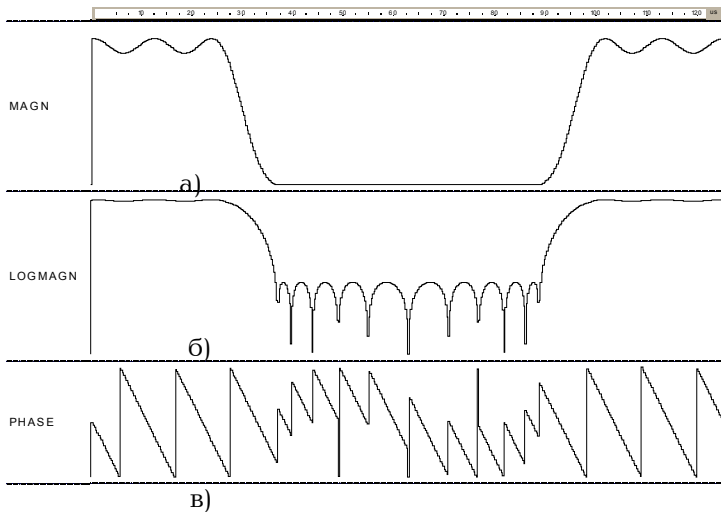


Рис.4.17. АЧХ (а), АЧХ у логарифмічному масштабі (б) та ФЧХ (в) синтезованого СІХ-фільтра

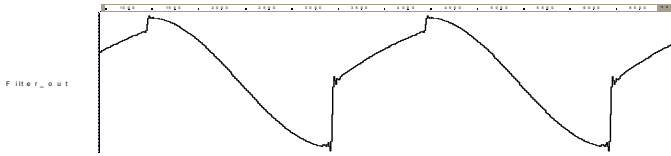


Рис. 4.18. Результат фільтрації сигналу СІХ-фільтром

Далі модель `FIR_filter`, яка працює з плаваючою комою, була перероблена для обчислень з цілими числами у модель `FIR_filter_I`. Для цього, крім заміни типів сигналів та змінних, були змасштабовані коефіцієнти згідно з (4.24):

**constant b:** `Tarri:= ( 10, 24, 7, -26, -15, 42, 31, -70, -75, 162, 448);`

Також результат був поділений на масштабний коефіцієнт:

`Y<=acc/2**10;`

Після включення в стенд для випробувань і подання тестової послідовності з амплітудою 2048, що відповідає розрядності  $n_x$ , була одержана АЧХ, як на рис. 4.16.

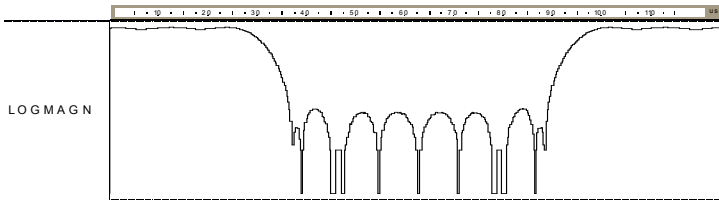


Рис.4.19. АЧХ у логарифмічному масштабі для СІХ-фільтра з  $n_x = 12$  і  $n_k = 10$

Порівняння АЧХ на рис. 4.17 та 4.19 показує, що округлення коефіцієнтів до 10 розрядів погіршило  $A_2$  (43 дБ замість 48 дБ) і не змінило інших показників. Отже, синтез СІХ-фільтра пройшов успішно. Для підвищення його якості слід збільшити розрядність коефіцієнтів.

## Лабораторна робота 4

### Фільтри з нескінченною імпульсною характеристикою

**Мета роботи:** одержати знання про фільтри з нескінченною імпульсною характеристикою та навички їх розрахунку, програмування та моделювання за допомогою VHDL.

#### Теоретичні відомості

НІХ-фільтри (див. підрозділ 2.2) мають наступні переваги над СІХ-фільтрами:

- у кілька разів менші обчислювальні витрати та об'єм пам'яті;
- менша групова затримка сигналу;
- має пряму аналогію з фізичними моделями та аналоговими фільтрами.

До недоліків НІХ-фільтрів належать нелінійність ФЧХ, необхідність забезпечення стабільності обчислень і як результат – підвищена точність розрахунків. Наявність зворотніх зв'язків в ГСПД НІХ-фільтра обмежує можливості розпаралелювання обчислень.

Реалізація НІХ-фільтра – це обчислення різницевого рівняння (2.2). Якщо одиничний імпульс подати на вхід  $x$  фільтра, то на його виході  $y$  одержимо його імпульсну реакцію  $h(n)$  (2.3). Через те, що результат  $y(n)$  залежить від попередніх результатів  $y(n-k)$ , імпульсна реакція виходить нескінченною. Це стало причиною назви таких фільтрів.

На відміну від СІХ-фільтрів, НІХ-фільтри мають крім нулів також полюси передаточної функції. Наявність пари комплексних полюсів  $r_1 = \text{Re}(r) + j\text{Im}(r)$ ,  $r_2 = \text{Re}(r) - j\text{Im}(r)$  задає резонансні властивості фільтра на частоті, яка визначається фазовою координатою полюсів  $\varphi = \arg(r)$ . Причому чим ближче полюс до одиничного кола, тобто чим більше  $|r|$  наближається до 1 – тим значніше проявляється резонанс (див. рис. 4.5). При цьому проміжні результати фільтра можуть зростати до  $\theta$  разів у порівнянні з амплітудою вхідного сигналу, де  $\theta$  – **коефіцієнт добротності**. У крайньому випадку, коли  $|r| = 1$ ,  $\theta \rightarrow \infty$ , тобто фільтр стає нестійким. Отже, нерівність  $|r| < 1$  є **критерієм стійкості** НІХ-фільтра, якого обов'язково слід дотримуватись.

У попередній лабораторній роботі можна було пересвідчитись, що якість фільтрації зростає зі збільшенням довжини імпульсної реакції. Отже, добравши якісні коефіцієнти для (2.2) можна одер-



жати НІХ-фільтр, ефективність якого буде суттєво вища ніж у СІХ-фільтра.

Пошук коефіцієнтів — це задача апроксимації передаточної функції (2.5), яка повинна наближатись до заданої форми на рис. 4.12. За методом апроксимації НІХ-фільтри розділяють на фільтри Бесселя, Батерворта, Чебишева та еліптичні. Ці методи були розроблені для синтезу аналогових фільтрів і були адаптовані до розробки цифрових фільтрів. Фільтр Бесселя має гладку АЧХ у смугах пропускання та непропускання і є аналогом ланцюгового РС-фільтра. Фільтр Батерворта має незначну пульсацію у смузі пропускання. Фільтр Чебишева першого роду має мінімізовані рівновеликі пульсації у смузі пропускання і гладку АЧХ в смузі непропускання, а фільтр Чебишева другого роду — навпаки — мінімізовані рівновеликі пульсації у смузі непропускання.

Еліптичний фільтр має мінімізовані рівновеликі пульсації як у смузі пропускання, так і непропускання. Серед усіх фільтрів він має мінімальну кількість коефіцієнтів. Але він має найбільше пере-кручування фази і, відповідно, нерівномірну групову затримку (4.15). Як результат, сигнал, який профільтровано еліптичним фільтром, суттєво змінює форму.

Як правило, НІХ-фільтр роблять багатоланковим, тобто його передаточна функція визначається формулою (2.6), а його структура — як на рис. 3.2. Такий спосіб реалізації фільтра дає змогу дещо зменшити вимоги до точності обчислень, тому що загальний коефіцієнт добротності фільтра  $\theta$  розділяється між ступеннями. Крім того, кожна наступна ланка фільтрує похибки обчислень попередніх ланок і результируючий шум округлення стає меншим, ніж у фільтрів, реалізованих іншим способом.

$k$ -та ланка фільтра може бути пешого чи другого порядку з передаточною функцією:

$$H_1(z) = \frac{b_{0,k} + b_{1,k}z^{-1}}{1 + a_{1,k}z^{-1}} ; H_2(z) = \frac{b_{0,k} + b_{1,k}z^{-1} + b_{2,k}z^{-2}}{1 + a_{1,k}z^{-1} + a_{2,k}z^{-2}} .$$

Для зменшення кількості множень коефіцієнти чисельників нормалізують і чисельник стає рівним

$$1 + b_{1,k}z^{-1}/b_{0,k} + b_{2,k}z^{-2}/b_{0,k} . \quad (4.28)$$

При цьому результат фільтрації множать на загальний коефіцієнт підсилення  $b = b_{0,1} \cdot b_{0,2} \dots$ .

Ланка першого порядку має реальні нуль та полюс. Ланка другого порядку має, як правило, комплексні полюси. Найчастіше у цього фільтра нулі реальні, але у еліптичного фільтра та фільтра Чебишева другого роду — нулі завжди комплексні. Якщо фільтр — ФНЧ з реальними нулями  $q_{1,2} = 1$ , то чисельник  $H_2(z)$  дорівнює

$1+2z^{-1}+z^{-2}$ , у фільтра високих частот нулі  $q_{1,2} = -1$  і чисельник дорівнює  $1-2z^{-1}+z^{-2}$ . У смугового фільтра нулі  $q_1 = -1$ ,  $q_2 = 1$ , які відповідають чисельнику  $1-z^{-2}$ . Якщо смуговий фільтр – багатоланковий то у однієї ланки може бути чисельник  $H_2(z)$  як у ФНЧ, а у іншої – як у ФВЧ.

Згідно з вимогою критерія стійкості НІХ-фільтра, коефіцієнти знаменника  $H_2(z)$  повинні задовольняти нерівність

$$1 > a_{2,k} > |a_{1,k}| - 1. \quad (4.29)$$

Умовою того, що корені – комплексні, є

$$|0,5a_{1,k}/\sqrt{a_{2,k}}| < 1. \quad (4.30)$$

Для пошуку коефіцієнтів можна використати наступну функцію системи Scilab:

`[cells, fact, zzeros, zpoles]=eqiir(ftype, approx, [om1, om2, om3, om4], delpa, deltas);`

де `ftype` – тип фільтра: 'lp' – ФНЧ, 'hp' – ФВЧ, 'bp' – СФ, 'sb' – РФ; `approx` – вид фільтру: 'butt' – Баттерворта, 'cheb1' – Чебишева першого типу, 'cheb2' – Чебишева другого типу, 'ellip' – еліптичний;  $0 \leq om1 \leq om2 \leq om3 \leq om4 \leq \pi$  – частоти початку і кінця смуг пропускання чи непропускання у радіанах; `delpa`, `deltas` –  $\delta_1, \delta_2$  – коливання АЧХ у смугах пропускання та непропускання (рис. 4.12); `cells` – вказує, що слід знайти коефіцієнти багатоланкового фільтру; `fact` – коефіцієнт нормалізації, тобто загальний коефіцієнт підсилення  $b$ ; `zzeros`, `zpoles` – множини нулів та полюсів, відповідно. Крім того, ця функція виводить формулу передаточної функції у формі (2.6) і (4.28).

### Завдання для роботи

1. Синтезувати за допомогою системи Scilab набір коефіцієнтів НІХ-фільтра, який задовольняє задані параметри АЧХ. Параметри фільтра вибрати з табл. 4.3 за варіантом. Вид апроксимації – Баттерворта (0), Чебишева першого роду (1), Чебишева другого роду (2) еліптичного фільтра (3) вибрати за формулою  $N \bmod 4$ , де  $N$  – номер варіанта. Одержані набори нулів та полюсів відобразити на графіку  $z$ -площини.

2. Розробити модель ланки фільтра другого порядку, описану мовою VHDL. Коефіцієнти фільтра повинні задаватись як настроювальні константи.

3. Розробити модель багатоланкового НІХ-фільтра структурним стилем. Настроїти ланки моделі згідно з набором коефіцієнтів фільтра. Виміряти АЧХ у лінійному та логарифмічному масштабі, а також ФЧХ так само, як це робилось у лабораторній роботі 3.

Виміряти внутрішні сигнали ланок і оцінити величину коефіцієнта добротності ланок  $\theta_i$ . Порівняти загальний коефіцієнт добротності  $\theta$  фільтра з величиною, оберненою до **fact**.

4. Підключити фільтр до виходу генератора сигналів, який розглядався у першій лабораторній роботі і одержати графік результуючого сигналу. Пояснити його форму.

5. Округлити коефіцієнти фільтра до двох розрядів після коми і знов виміряти АЧХ. Пояснити її форму.

6. В одній з ланок змінити коефіцієнти так, щоб не справджувалась нерівність (4.29) і щоб  $|r| \approx 1$ . Виміряти вихідний сигнал і дослідити ефект збудження.

7. Проаналізувати одержані графіки. Зробити висновки по роботі.

### Приклад виконання роботи

Нехай задано: СФ,  $A_1 = 0.3$  дБ,  $A_2 = 54$  дБ,  $f_{н1} = 0.05$   $f_{п1} = 0.11$ ,  $f_{н2} = 0.14$ ,  $f_{п2} = 0.2$ .

Вирішивши рівняння (4.17), знайдено  $\delta_1 = 0.02$ ,  $\delta_2 = 0.002$ .

Сформовано команду для Scilab, щоб знайти набір коефіцієнтів

```
[cells,fact,zzeros,zpoles]=
    eqiir('bp','cheb1',[0.1*%pi,0.22*%pi,0.28*%pi, 0.4*%pi],0.02,0.002)
```

За цією командою одержано формулу передаточної функції:

$$\frac{1 + 2z + z^2}{0.9487295 - 1.2250177z + z^2} \quad \frac{1 + 2z + z^2}{0.8857866 - 1.2789722z + z^2}$$

$$\frac{1 - 2z + z^2}{0.8948900 - 1.401471z + z^2} \quad \frac{1 - 2z + z^2}{0.9582494 - 1.5204099z + z^2},$$

причому **fact** = 0.0000507.

Графік нулів та полюсів (рис. 4.20) одержано за командами Scilab:

```
ze =[- 1., - 1., - 1., - 1., 1., 1., 1., 1.];
pol =[0.6125089 + %i*0.7573390, 0.6125089 - %i*0.7573390,
      0.6394861 + %i*0.6905390, 0.6394861 - %i*0.6905390,
      0.7007355 + %i*0.6354996, 0.7007355 - %i*0.6354996,
      0.7602049 + %i*0.6167154, 0.7602049 - %i*0.6167154];
n= [0:0.01:2*%pi]';
plot(cos(n),sin(n),'k', real(ze),imag(ze),'ko',real(pol),imag(pol),'kx')
```

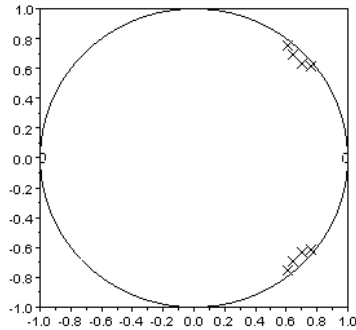


Рис. 4.20. Графік розміщення нулів та полюсів

Опис VHDL-моделі ланки фільтра приведено у розділі 2.3. Структура фільтра показана на рис. 3.2.

Одержані графіки АЧХ та ФЧХ показані на рис. 4.21 при встановленій затримці  $\text{maxdelay}=400$ .

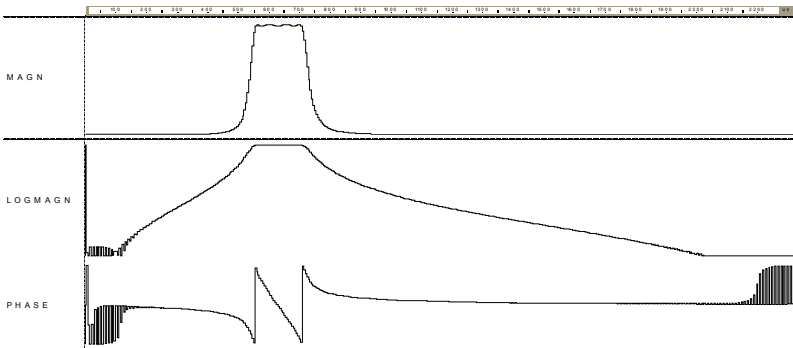


Рис. 4.21. Графіки АЧХ у лінійному та логарифмічному масштабі та ФЧХ НІХ-фільтра

Аналіз графіків на рис. 4.21 показує, що одержані характеристики задовольняють задані, причому  $f_{\text{Н1}} = 0.05$ ,  $f_{\text{П1}} = 0.109$ ,  $f_{\text{П2}} = 0.141$ ,  $A_2 = 58$  дБ на межі смуги непропускання і плавно спадає до рівня 100 дБ.

Шляхом вимірювання амплітуди максимальних сигналів на входах і виходах ланок та обчислення їх відношень визначені добротності ланок  $\theta_1=40$ ,  $\theta_2=13$ ,  $\theta_3=7$ ,  $\theta_4=5.4$ . Загальна доброт-

ність  $\theta = \theta_1 \cdot \theta_2 \cdot \theta_3 \cdot \theta_4 = 19656$ . Вона доволі точно збігається з величиною  $1/\text{fact} = 19723$ .

На рис. 4.22 показаний відгук фільтра на тестовий сигнал, такий як на рис. 4.3. Видно, що фільтр виділив частоти зі спектром, який має ширину смуги пропускання. Сплески сигналу відповідають різким переходам у тестовому сигналі, тобто тим місцям, де зосереджені високі частоти.

Після округлення коефіцієнтів до двох десяткових знаків після коми одержано АЧХ, як на рис. 4.23.

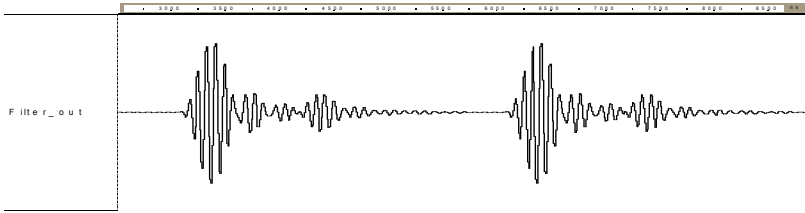


Рис. 4.22. Графік відгуку НІХ-фільтра на тестовий сигнал.

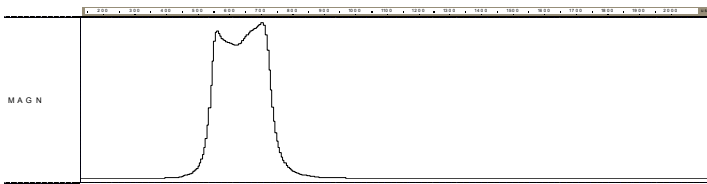


Рис. 4.23. Графік АЧХ НІХ-фільтра з округленими коефіцієнтами

Як видно з рис. 4.23, неточне представлення коефіцієнтів привело до спотворення АЧХ у смугі пропускання, через те, що передаточна функція була сформована з чебишевською оптимізацією пульсацій саме у смугі пропускання.

## Лабораторна робота 5 Швидке перетворення Фур'є

**Мета роботи:** одержати знання про методи спектрального оцінювання, алгоритм ШПФ та навички їх програмування та моделювання за допомогою VHDL.

### Теоретичні відомості

У розділі 2.2 розглядалось  $z$ -перетворення дискретних сигналів  $x(n)$ , що дає представлення сигналу через суму власних функцій. Згідно з (2.4), при  $z = e^{j\omega} = e^{j2\pi fT}$   $z$ -перетворення представляє собою дискретне перетворення Фур'є:

$$X(f) = T \sum_{n=-\infty}^{\infty} x(n)e^{-j2\pi fnT}, \quad (4.31)$$

де  $T$  – період дискретизації сигналу  $x(n)$ ,  $X(f)$  – спектр сигналу. Слід відмітити, що спектр  $X(f)$  – періодична безперервна функція, період якої дорівнює частоті дискретизації  $F_s = 1/T$ . Для неї існує обернена функція – зворотне перетворення Фур'є

$$x(n) = \frac{1}{T} \int_0^{F_s} X(f)e^{j2\pi fn} df. \quad (4.32)$$

Вирази (4.29) і (4.30) для кінечної послідовності  $x(n)$  довжиною  $N$  відліків виглядають як зрізані ряди Фур'є

$$F(x) = X(k) = T \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N}, \quad (4.33)$$

$$F^{-1}(X) = x(n) = \frac{1}{NT} \sum_{k=0}^{N-1} X(k)e^{j2\pi kn/N}. \quad (4.34)$$

Власне, (4.33) і (4.34) називаються в ЦОС як пряме і зворотне **дискретне перетворення Фур'є** (ДПФ). Обидві послідовності  $x(n)$  та  $X(k)$  є періодичними з періодом  $N$ . В них параметр  $T$  введено для можливості обчислення каліброваної спектральної щільності енергії

$$E(k) = X^2(k) = \left( T \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \right)^2;$$

та спектральної щільності потужності

$$\tilde{S}(k) = \frac{E(k)}{NT} = \frac{T}{N} \left( \sum_{n=0}^{N-1} x(n)e^{-j2\pi kn/N} \right)^2. \quad (4.35)$$

Функцію (4.34) часто використовують для вимірювання спектральних властивостей сигналу. Метод оцінки спектру на основі обчислення ДПФ виборок сигналу довжиною  $N$  називається методом періодограм, а одержаний спектр – періодограмою.

**Періодограма**  $\tilde{S}(k)$  дорівнює спектральній щільності потужності періодичного сигналу з періодом  $N$ . Для дійсного сигналу це лише наближена оцінка спектральної щільності  $S(k)$ . При цьому, якщо  $x(n)$  називають відліками, то  $X(k)$  чи  $S(k)$  – називають **бінами**. Безпосереднє обчислення (4.35) означає, що у доволі довгому дискретному сигналі  $x(n)$ , так званім, прямокутним **часовим вікном**  $w(n)$  вирізається фрагмент довжиною  $N$ , який вважається періодичним:

$$x'(n) = x(n) \cdot w(n),$$

причому  $w(n) = 1$  при  $0 \leq n < N$  та  $w(n) = 0$  – при інших  $n$ . Як результат, сигнал, що аналізується, стає несхожим на початковий сигнал. В ньому з'являються розриви в точках  $iN$ . Отже, результуючий спектр є згорткою справжнього спектру та спектру вікна

$$F(x'(n)) = F(x(n)) * F(w(n)). \quad (4.36)$$

Через це у спектральній щільності  $\tilde{S}(k)$  крім основної пелюстки складової спектру з'являються бокові пелюстки. Поява цих пелюсток називається розтіканням або просочуванням потужності і призводить до зміщення оцінок спектру сусідніх бінів відносно біна основної пелюстки.

Важливим параметром методу спектрального оцінювання є його роздільна здатність, тобто здатність роздільно вимірювати спектральні відгуки двох синусоїдальних сигналів, які близькі за амплітудою та частотою. Вважається, що спектральні піки розділені, якщо провал між ними складає величину не менше 3дБ.

Згідно з (4.33), інтервал часу дослідження сигналу дорівнює  $T_e = NT$ . Відповідно, відстань між бінами  $B_e = 1/NT$  характеризує максимальну роздільну здатність. Звідси видно, що спектральна роздільна здатність  $B_e$  в герцах обернено пропорційна інтервалу часу сигналу  $T_e$ , що обробляється в секундах, тобто  $T_e B_e = 1$ .

Для зменшення впливу просочування потужності використовуються вікна спеціальної форми. Це еквівалентно тому, що ввімкнено фільтр, який фільтрує спектр за формулою (4.36) і своїм впливом мінімально спотворює спектр. Усі вікна симетричні відносно центру і мають спадання підсилення на своїх краях, що призводить принаймні до мінімізації розривів у сигналі, що аналізується.

При застосуванні часового вікна роздільна здатність зменшується в  $\alpha$  разів у порівнянні з еквівалентною шириною смуги  $B_e$ . Отже, при обробці випадкових сигналів слід розглядати ефективну

статистичну ширину смуги  $B_s = \alpha B_e$ . Вона дорівнює ширині смуги основної пелюстки на рівні 3 дБ.

У табл. 4.5. представлені часові вікна, які застосовуються при спектральному аналізі методом періодограм для  $-N/2 \leq n < N/2$ . Максимальний рівень бокових пелюсток  $\Delta S_s$  показує, синусоїду якого рівня неможливо розпізнати у періодограмі, якщо у сигналі є синусоїда з амплітудою 0 дБ у сусідніх бінах.

Таблиця. 4.5. Часові вікна для спектрального аналізу

№ пп.	Вікно	Функція $w(n)$	$\Delta S_s$ , дБ	$\alpha$	A
1	Прямокутне	1	-13,3	0,88	1
2	Трикутне	$1 - 2 n /N$	-26,5	1,28	$\sqrt{3}$
3	Макса, Фока, Берг'є	$\frac{\sin(2\pi n/N)}{2\pi n/N}$	-26	1,24	1,49
4	Ханна	$0.5 + 0.5 \cos \frac{2\pi n}{N}$	-31,5	1,44	1,63
5	Хеммінга	$0.54 + 0.46 \cos \frac{2\pi n}{N}$	-43	1,3	1,59
6	Парзена	$1 - 6\frac{4n^2}{N^2} + 6\frac{8 n ^3}{N^3},  n  \leq N/4$ $2\left(1 - \frac{2 n }{N}\right)^3, N/4 <  n $	-51	1,84	1,93
7	Блекмана	$0.42 + 0.5 \cos \frac{2\pi n}{N} + 0.08 \cos \frac{4\pi n}{N}$	-55	1,64	1,81
8	Лапласа - Гауса	$\exp\left(-\frac{n^2}{18N^2}\right)$	-55	1,64	1,84
9	Наттола	$0,36358 + 0,48918 \cos \frac{2\pi n}{N} +$ $+ 0,1366 \cos \frac{4\pi n}{N} + 0,01064 \cos \frac{6\pi n}{N}$	-98	1,7	
10	Карре - Руйс	$\frac{\sin(2\pi n/N)}{2\pi n/N} \cdot (1 - 2 n /N)$	-	1,64	1,98

Оскільки множення сигналу на вікно дещо заглушує сигнал, амплітуда спектральних складових зменшується в A разів. Тому для корекції результатів аналізу функцію вікна або сигнал чи результат ДПФ слід помножити на A.

При аналізі випадкових сигналів слід враховувати статистичний показник якості оцінки Q, який дорівнює відношенню середніх



рівнів шуму і сигналу. Доведено, що для сигналу з гаусовим шумом спектральне оцінювання є слухним, якщо

$$QT_e B_s \geq 1 \quad (4.37)$$

Нехай  $Q = 0,1$ , тоді роздільна здатність, що досягається, оцінюється як  $B_s \approx 10/T_e = 10B_e$ . Тому для слухної оцінки спектру слід збільшувати величину  $T_e$ . Для цього можна збільшувати довжину виборки  $N$  у  $M = 1/Q$  разів, а потім результат одержати як осереднення  $M$  сусідніх бінів.

Інший метод — метод періодограм Велча — оснований на тому, що  $N \cdot M$  відліків сигналу розділяється на  $M$  виборок, для яких обчислюється спектр  $\tilde{S}_i(k)$ ,  $i=1, \dots, M$ . Результируючий спектр є осередненням спектрів виборок:

$$\tilde{S}(k) = \frac{1}{M} \sum_{i=1}^M \tilde{S}_i(k) \quad (4.38)$$

ДПФ (4.33) основане на операції повороту комплексного вектора. Поворот виконується як множення комплексного числа на коефіцієнт  $e^{-j2\pi pn/N}$ , який скорочено зображається як  $W_N^{pn}$ .

Нехай вхідний сигнал  $x(n)$  представляє собою комплексну синусоїду — вектор довжиною  $r_p$ , що в кожному такті повертається на кут  $2\pi p/N$ , тобто у  $n$ -му такті він дорівнює  $r_p W_N^{pn}$ . Якщо ці вектори-відліки повертати у протилежний бік на кут  $-2\pi k/N$  і накопичувати в акумуляторі, то результат  $\sum r_p W_N^{-pn} \cdot W_N^{kn}$  досягне максимуму  $\sum r_p$  при умові, що  $p = k$ , а якщо  $p \neq k$ , то результат прямує до нуля. Недолік ДПФ полягає у великій кількості обчислень —  $N^2$  комплексних множень та додавань, якщо необхідно знайти результати для усіх  $N$  частотних каналів або бінів.

Отже, у нульовому частотному каналі вхідні відліки повертають на кут 0, у  $k = 1$  — му каналі — на кут  $2\pi n/N$ , у 2-му каналі — на кут  $4\pi n/N$  і т.д. Причому у різних каналах трапляються однакові повороти тих самих даних, наприклад, у всіх каналах — на кут 0, у половині каналів — на кут  $\pi/2$ . Крім того, якщо непарні відліки у непарних каналах повернути на кут  $2\pi n/N$ , то решта поворотів у парах сусідніх каналів співпадають. Будь-який поворот для ДПФ довжини  $2^{\log_2 N}$  можна представити як послідовність  $\log_2 N$  поворотів на кут  $2^q \pi/N$ ,  $q = 1, 2, \dots$ . У парних каналах кількість поворотів можна скоротити удвічі, якщо додати відліки  $x(n)$  та  $x(n+N/2)$ .

На таких особливостях ДПФ ґрунтуються алгоритми швидкого перетворення Фур'є (ШПФ). Так, ДПФ можна представити як додавання двох сум від парних та непарних вхідних даних:

$$\begin{aligned}
 X(k) &= \sum_{n=0}^{N/2-1} x(2n)W_N^{2nk} + \sum_{n=0}^{N/2-1} x(2n+1)W_N^{(2n+1)k} = \\
 &= \sum_{n=0}^{N/2-1} x(2n)W_{N/2}^{nk} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1)W_{N/2}^{nk}.
 \end{aligned}
 \tag{4.39}$$

Тут множник  $T$  з (4.33) усунуто як загальний нормуючий коефіцієнт. Як видно, (4.39) — це два ДПФ для  $N/2$  даних кожне, причому результати другого ДПФ повернуті на кут  $-2\pi k/N$ . Аналогічно розкладаються  $N/2$ -точкові ДПФ на  $N/4$ -точкові ДПФ. З кожним таким розкладанням кількість операцій зменшується приблизно удвічі. Найменшим ДПФ є двоточкове ДПФ:

$$\begin{aligned}
 X(0) &= x(0) + x(1); \\
 X(1) &= x(0) - x(1).
 \end{aligned}$$

З урахуванням множення на коефіцієнт повороту  $W_N^q = W^q$  одного з операндів базова операція ШПФ записується як

$$\begin{aligned}
 X(0) &= x(0) + x(1)W^q; \\
 X(1) &= x(0) - x(1)W^q.
 \end{aligned}$$

ГСПД базової операції ШПФ та її умовне позначення показані на рис. 4.24. За форму графа базової операції її часто називають "метеликом" (butterfly). За розмір ДПФ базової операції таке ШПФ називають ШПФ за основою 2. Таке ШПФ є найпоширенішим. Менш поширеним є ШПФ за основою 4, яке має на чверть меншу кількість операцій. ШПФ, які мають інші основи, включаючи комбіновані основи, використовуються рідко.

Один з графів алгоритмів ШПФ для  $N = 8$  показано на рис. 4.25. На ньому пунктиром виділено графи

4-точкового ШПФ, які ілюструють обчислення за формулою (4.39). Виділення серед вхідних даних, що представляють сигнал у часовому просторі, в (4.39) парних чи непарних відліків називається проріджуванням. Через це даний алгоритм називається ШПФ з проріджуванням по часу. У інших алгоритмів ШПФ виконується проріджування, починаючи з частотних результатів і тому вони називаються ШПФ з проріджуванням по частоті.

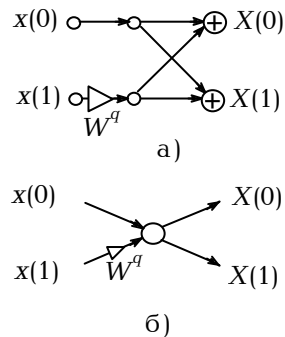


Рис. 4.24. ГСПД базової операції ШПФ (а) та її умовне позначення (б)

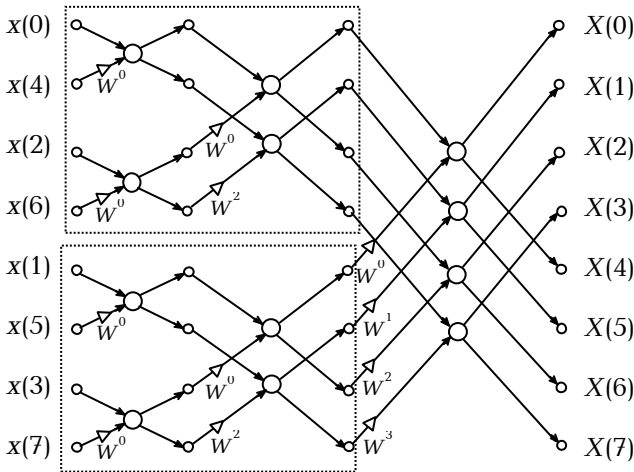


Рис. 4.25. Граф алгоритму ШПФ для  $N = 8$

Стовпці точок на графі можна розглядати як масиви даних, що обробляються. Якщо номер даного у масиві представити двійковим числом, то щоб з натурального створити **двійково-інверсний порядок** масиву, слід дзеркально переставити двійкові розряди номеру даних. Наприклад, номер  $a_3a_2a_1a_0$  перетворюється на номер  $a_0a_1a_2a_3$ . Тоді за нумерацією даних у стовпцях видно, що результуючий масив на рис. 4.25 має натуральний порядок, а вхідні дані — двійково-інверсний. Стовпець базових операцій називається **ітерацією** алгоритму ШПФ.

Нескладно порахувати, що у 8-точковому ШПФ лише 12 комплексних додавань та множень.  $N$ -точкове ШПФ має  $0,5N \log_2 N$  комплексних додавань та множень замість  $N^2$  таких операцій при обчисленні ДПФ за формулою (4.33). Отже, ШПФ є ефективним алгоритмом.

Алгоритм зворотнього ШПФ одержують при заміні поворотних коефіцієнтів на  $W^{-q}$  та корекції знаків у базовій операції згідно з формулою оберненого ДПФ (4.34). Але для обчислення оберненого ДПФ можна використати також пряме ШПФ, якщо усі результати помножити на  $N$  та, крім  $X(0)$ , поміняти місцями:  $X(1)$  з  $X(N-1)$ ,  $X(2)$  з  $X(N-2)$  і т.д. [1].

Досі малась на увазі обробка за допомогою ДПФ комплексного, тобто аналітичного сигналу. На практиці найчастіше ДПФ застосовується до реального сигналу. Для цього уявна частина вхідних даних представляється нулем:  $\text{Im}(x(n)) = 0$ . Спектр такого сигналу для парного  $N$  має симетрію відносно спектрального відліку  $X(N/2)$ :

$$X^*(N - k) = X(k) ; \quad (4.40)$$

тобто

$$\begin{aligned} \operatorname{Re}(X(N - k)) &= \operatorname{Re}(X(k)) ; & k = 1, \dots, N/2 - 1; \\ \operatorname{Im}(X(N - k)) &= -\operatorname{Im}(X(k)); \\ \operatorname{Im}(X(0)) &= \operatorname{Im}(X(N/2)) = 0. \end{aligned}$$

Існують алгоритми обчислення ДПФ реальних послідовностей, в яких завдяки використанню властивості (4.40) загальна кількість операцій зменшена майже удвічі.

У лабораторній роботі ШПФ виконується програмою FFTbehR.VHD, яка описана нижче. У VHDL-симуляторах зазвичай цифрові сигнали спостерігаються у вигляді графіків, які зберігаються у процесі моделювання алгоритму. Але алгоритм ШПФ припускає, що дані надходять і видаються відразу у вигляді масивів. Тому доцільно обчислювати ШПФ як ланцюжок процедур. Перша з них називається Gather\_C і накопичує вхідний масив комплексних даних. Друга процедура – FFT – виконує алгоритм ШПФ, а заключна процедура Scatter\_C – видає результуючий масив у вигляді послідовності даних.

Зазначені процедури виконуються об'єктом FFTbeh. Інтерфейс цього об'єкта й перелік використовуваних стандартних бібліотек виглядає так:

```

library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_ARITH.all;
use IEEE.MATH_REAL.all;
use IEEE.MATH_COMPLEX.all;
entity FFTbeh is
  generic(NN:positive:=8;           -- 2**NN - довжина перетворення
    IFFT:natural:=0; -- 0 - пряме, 1 - зворотне перетворення
    pt:integer;      -- масштаб результату
    overs:integer:=0 -- 0,1 – без перекриття вхідних масивів;
                    -- 2 – перекриття 1/2, 4 – перекриття 1/4...
                    );-- -1, -2 – половина масиву занулюється для згортки

  port (
    CLK: in STD_LOGIC;      -- синхросерія
    RST: in STD_LOGIC;      -- встановлення в 0
    START: in STD_LOGIC;    -- початок накопичення масиву
    EDIN: in STD_LOGIC;     -- строб даних
    DIN_RE: in real;        -- реальна частина даних
    DIN_IM: in real;        -- уявна частина даних
    DOUT_RE: out real:=0;   -- реальна частина спектру
    DOUT_IM: out real:=0;   -- уявна частина спектру
    NUM: out integer:=0;    -- номер вихідного відліку
    READY: out STD_LOGIC);-- початок виводу результатів
end FFTbeh;

```

Настроювальна змінна  $N = 2^{NN}$  задає довжину перетворення. При IDCT = 0 виконується пряме перетворення, а при 1 – зворотнє. Змінна PT задає масштаб результату. Отже, програма виконує обчислення (4.33), (4.34):

$$X(k) = \frac{1}{2^{PT}} \sum_{n=0}^{N-1} x(n) e^{-j(-1)^{IDCT} 2\pi kn/N}.$$

де  $x(n) = \text{DIN\_RE} + j * \text{DIN\_IM}$  – відлік комплексного вхідного даного,  $X(k) = \text{DOUT\_RE} + j * \text{DOUT\_IM}$  – відлік комплексного спектру.

За сигналом START починається накопичення масиву вхідних даних. Закінчення обчислень ШПФ вказується сигналом READY. Вихідні відліки спектру супроводжуються номером NUM, за котрим знаходиться значення частоти певної спектральної лінії. Сигнал EDIN стробує прийом вхідних даних.

Поведінка об'єкта FFTbeh\_R описана в архитектурі FFT\_R. Для представлення масивів даних в ній оголошені наступні типи масивів комплексних і реальних даних.

```
constant N: integer:=2**NN;           -- довжина перетворення
type MEMOC is array (0 to N) of complex; --комплексний масив
type MEMOC_2 is array (0 to N/2-1 ) of complex;
type MEMOR is array (0 to N-1) of real;  -- реальний масив
```

Далі описана процедура, за якою накопичується масив вхідних комплексних даних.

```
procedure GATHER_C(signal CLK,START,EDIN:in std_logic;
signal DR,DI: in integer;           --реальна та уявна частини
           -- відліків вхідних даних
signal rdy: out std_logic;         --сигнал готовності
signal datact: inout natural;      --номер даного
signal RAM:out MEMOC)              -- робочий масив
is
begin
  wait until CLK= '1'; -- запуск процедури за фронтом CLK
  if EDIN = '1' then -- дозвіл прийому
    RAM(datact)<=(DR,DI); -- запис комплексного даного
    if datact < RAM'right then
      datact<=datact+1;  -- лічильник даних
    end if;
    if START='1' then
      datact<=RAM'left; --встановлення лічильника даних
    end if;
    rdy<='0';
    if datact =RAM'right-1 then
      rdy<='1';        --масив готовий
```

```

        end if;
    end if;
end procedure;

```

У цій процедурі при кожному її запуску при EDIN = '1' в масив RAM за лічильником datact записується чергова пара даних .

Процедура SCATTER\_C виглядає як

```

procedure SCATTER_C(signal CLK,START,EDIN : in std_logic;
    pt: natural; --масштаб результату
    signal RAM: in MEMOC; --вхідний масив
    signal datact: inout natural; --лічильник даних
    signal rdy: out std_logic; -- кінець виводу масиву
    signal DR,DI: out integer) -- вихідні дані
is
    variable s:real:=real(2**pt); -- масштабний коефіцієнт
begin
    wait until CLK='1';
    if START='1' then
        datact<=RAM'left;
    elsif EDIN='1' then
        RDY<='0';
        DR<= integer(RAM2(datact).RE/s);
        DI<= integer(RAM2(datact).IM/s);
        if datact< RAM'right then
            datact<=datact+1;
        end if;
        if datact=RAM'right-1 then
            RDY<='1';
        end if;
    end if;
end procedure;

```

Ця процедура виконується так само, як і попередня. Але навпаки, накопичені у масиві дані послідовно видаються на її вихід.

Нарешті, процедура, що обчислює ШПФ масиву даних виглядає так:

```

procedure FFT(N:positive; -- розмір перетворення =2**N
    signal rdy:in std_logic; -- запуск перетворення
    signal RAM_I: in MEMOC; -- вхідний масив даних
    signal RAM_O:out MEMOC) -- вихідний масив бітів
is
    constant NN:positive:=2**N;
    variable TWIDDLE:MEMOC_2; -- масив поворотних коефіцієнтів
    variable RAM:MEMOC; -- робочий масив
    variable a,b,c,d,w: complex;

```

```

variable al:real;
variable base,itera,dataact,twiddlelect,twiddleinc: natural;
variable delta:integer:=1;
variable addrf: std_LOGIC_VECTOR(N-1 downto 0);
variable addr: std_LOGIC_VECTOR(N-1 downto 0);
begin
-- формування таблиці поворотних коефіцієнтів
for i in 0 to NN/2-1 loop
    al:= (MATH_PI*real(2*i))/real(NN);
    if IDCT=0 then
        TWIDDLE(i):=(COS(al),-SIN(al)); -- для прямого ШПФ
    else
        TWIDDLE(i):=(COS(al), SIN(al));-- для зворотнього ШПФ
    end if;
end loop;

loop -- ОСНОВНИЙ ЦИКЛ
    wait until RDY='1'; -- початок ШПФ
-- двійково-інверсна перестановка вхідних даних
    addrf:=(others=>'0'); -- вектор адреси для прямого порядку
    dataact:=0;
    for i in 1 to NN loop
        for ind in 0 to N-1 loop -- інверсія порядку бітів
            addr(ind):=addrf(N-1-ind); -- інверсна адреса
        end loop;
    RAM(conv_integer(unsigned(addri))):=RAM_I(conv_integer(unsigned(addrf)));
    addrf:=unsigned(addrf)+1;
end loop;
-- власне ШПФ
    itera:=0;
    delta:=1;
    twiddleinc:=0;
    for itera in 1 to N loop -- початок ітерації
        base:=0;
        twiddlelect:=0;
        for butterfly in 0 to NN/2 - 1 loop
            a:=RAM(base); -- початок базової операції
            base:=base+delta;
            b:=RAM(base);
            w:=TWIDDLE(twiddlelect);
            c:=a + b * w; -- власне метелик
            d:=a - b * w;
            RAM(base-delta):=c;
            RAM(base):=d; -- кінець базової операції
        -- модифікація параметрів базової операції
        base:= base + delta;

```

```

        if base >= NN then
            base:=base-NN+1;
            twiddlelect:=twiddlelect+twiddleleinc;
        end if;

        end loop;           -- кінець ітерації
        -- модифікація параметрів ітерації
        delta:=delta*2;
        if itera=1 then
            twiddleleinc:=NN/4;
        else
            twiddleleinc:=twiddleleinc/2;
        end if;
    end loop;
    RAM_O<=RAM;           -- результати ШПФ
end loop;
end procedure;

```

Тут виконується алгоритм ШПФ за основою 2 із проріджуванням у часі із заміщенням даних, як на рис. 4.25. Процедура має дві ділянки: перша з них виконується однократно на початку обчислень, а друга — власне ШПФ — виконується періодично. При виконанні першої ділянки формується таблиця обертаючих коефіцієнтів, розмір якої дорівнює довжині перетворення і яка потім використовується алгоритмом ШПФ.

Друга ділянка являє собою нескінченний цикл. Цикл запускається, як тільки готовий масив вихідних даних, який підготовлено процедурою GATHER\_C. Після запуску дані пересилаються із вхідного масиву у робочий масив. При цьому виконується двійкова інверсія адрес запису. Слід зазначити просту й оригінальну можливість двійкової інверсії адреси, що надає мова VHDL. Адреса представляється бітовим вектором і біти в ньому переставляються в інверсному порядку безпосередньо. У звичайних мовах програмування для цього необхідно виконати досить складний і не завжди очевидний алгоритм.

Як і всі алгоритми ШПФ, даний алгоритм являє собою гніздо циклів. У внутрішньому циклі виконуються базові операції ШПФ, а зовнішній цикл утворений NN ітераціями алгоритму ШПФ.

Виконавча частина архітектури містить виклики вищеописаних процедур:

**begin**

```

INPUT:GATHER_C(CLK,START,EDIN, DR=>DIN_Re,DI=>DIN_IM,
              rdy=>idatardy,datact=>datact_I,RAM=>RAM1);
SPECTRUM:FFT(N,idatardy,RAM_I=>RAM1,RAM_O=>RAM2);
OUTPUT: SCATTER_C(CLK,idatardy,EDIN,PT,RAM=>RAM2,DATACT=>datact_o,

```



```

rdy=>rdy,DR=>DOUT_RE,DI=>DOUT_IM);
DEL:process(CLK) begin
    if CLK='1' and CLK'event and EDIN='1' then
        READY<=idatardy;
        NUM<= dataact_o;
    end if;
end process;
end FFT_int;

```

Для розуміння виконання програми варто нагадати, що паралельний виклик процедури, у якій застосовується оператор **wait**, еквівалентний оператору процесу, тіло якого складається з тіла процедури. Таким чином, всі три процедури, як і еквівалентні їм процеси, виконуються паралельно в деякому конвеєрному режимі. При цьому дані й керування передаються від процедури до процедури як між ланками конвеєра. У результаті, об'єкт FFTbeh обробляє безперервний потік даних, сегментуючи його на блоки довжиною  $2**NN$ .

Процес DEL виконує затримку сигналу готовності READY і потоку номерів NUM відліків результату на один такт, щоб вони відповідали вихідним відлікам сигналів DOUT\_RE та DOUT\_IM .

### Завдання для роботи

1. Розробити VHDL-модуль множення на часові вікно. Модуль виконує додавання трьох вхідних сигналів:  $x(n) = A(n) + B(n) + C(n)$ . Результуючий потік реального сигналу  $x(n)$  помножується на часові вікно заданого типу або на прямокутне вікно (пропускає сигнал без змін). Тип вікна і його довжина  $N$  береться з таблиці 4.6. Модуль повинен також видавати сигнал START, який позначає початок вікна і призначений для запуску модуля ШПФ. Графік вікна занести у звіт.

2. Розробити VHDL-модуль, який осереднює  $M$  сусідніх виборок спектру  $X^2(k) = \tilde{S}_i(k)$  довжиною  $N$  відліків за формулою (4.38) або пропускає  $\tilde{S}_i(k)$  без обробки. Також модуль повинен видавати результат у логарифмічному масштабі, тобто у децибелах.

3. Скласти стенд для випробувань, який має два генератори синусоїдальних сигналів, генератор шумового сигналу, генератор спеціального сигналу, модуль множення навікно, модуль ШПФ FFTbeh та модуль осереднення. Генераторами спеціального та синусоїдального сигналів мають бути модуль, побудований у першій лабораторній роботі та модуль FilterTB\_r, відповідно. Довжина ШПФ встановлюється рівною  $N$  (табл.4.6).

Таблиця 4.6. Варіанти завдань

№ вар.	Вид вікна	$N$	$M$	№ вар.	Вид вікна	$N$	$M$
1	Трикутне	512	8	16	Блекмана	512	49
2	те саме	1024	9	17	те саме	1024	16
3	–	2048	10	18	–	2048	8
4	Макса, Фока, Берг'є	512	16	19	Лапласа – Гауса	512	100
5	те саме	1024	25	20	те саме	1024	64
6	–	2048	36	21	–	2048	49
7	Ханна	512	49	22	Натгола	512	16
8	те саме	1024	16	23	те саме	1024	25
9	–	2048	8	24	–	2048	36
10	Хеммінга	512	100	25	Карпе – Руйс	512	49
11	те саме	1024	64	26	те саме	1024	64
12	–	2048	49	27	–	2048	8
13	Парзена	512	16	28	Хеммінга	512	9
14	те саме	1024	25	29	те саме	1024	10
15	–	2048	36	30	–	2048	16

4. Підключити до реального входу модуля FFTbeh вихід модуля множення на вікно, а до уявного входу – нуль. Входи модуля множення на вікно підключити до виходів двох генераторів синусоїдального сигналу з однаковими амплітудами вихідних сигналів, що дорівнюють 1.0. Частота першого генератора дорівнює  $f_1 = 16/N$ , частота другого –  $f_2 > f_1$  встановлюється близькою до  $f_1$ . Частота генератора регулюється відношенням  $f = \text{fstrt}/\text{fsampl}$  при  $f_2 \text{ fsampl} \neq N$ . Параметр генераторів  $\text{Maxdelay} > NM$ .

Дослідити спектр сигналу для прямокутного вікна при відсутності осереднення. Підібрати таку частоту  $f_2$ , щоб її різниця  $\Delta f = f_2 - f_1$  була мінімальною і щоб піки частот у спектрі можна було розрізнити, тобто між піками має бути провал не менше 3 дБ. Записати  $\Delta f$  як роздільну здатність спектральної оцінки.

Встановити  $f_1 = 13/(0.8N)$ ,  $f_2 = 16/(0.73N)$ . Встановити таку амплітуду синусоїди з частотою  $f_2$ , пік якої можна розрізнити у спектрі рядом з піком  $f_1$ . Виміряти різницю амплітуд сигналів з частотами  $f_2$ ,  $f_1$  у децибелах і записати її як динамічний діапазон спектральної оцінки  $D_s$ . Виміряти також різницю амплітуд сигналів з частотами 0 та  $f_1$  у децибелах і записати її як динамічний діапазон вікна  $D_w$ .

5. Повторити п. 4 для заданого часового вікна. Перевірити справдження параметрів  $\Delta S_s$ ,  $\alpha$ ,  $A$  для заданого вікна.

6. Повторити п. 5 з доданим шумом, максимальна амплітуда якого у 30 разів менша за амплітуду першого синусоїдального сигналу.

7. Повторити п. 6, виконавши  $M$ -кратне осереднення спектру .

Порівняти результати п. 5 і п. 7 та перевірити справдження формули (4.33).

8. Підключити до входу модуля множення на вікно лише генератор спеціального сигналу, його параметри  $N_1, N_2$  зменшити у 20 разів і виміряти спектр сигналу, застосувавши задане вікно. Перевірити формулу (4.40).

9. Проаналізувати одержані графіки. Зробити висновки по роботі.

### Приклад виконання роботи

Нехай задано: вікно  $w(n) = (2,18 - (8n/N)^2 e^{-|4n/N|})$ ,  $N = 1024$ ,  $M = 64$ .

Графік вікна показано на рис. 4.26.

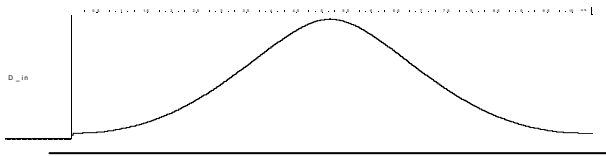


Рис. 4.26. Часове вікно для спектрального аналізу

Для прямокутного вікна одержано  $\Delta f = 1,60$  біна, динамічний діапазон  $D_s = 21,6$  дБ,  $D_w = 32,7$  дБ.

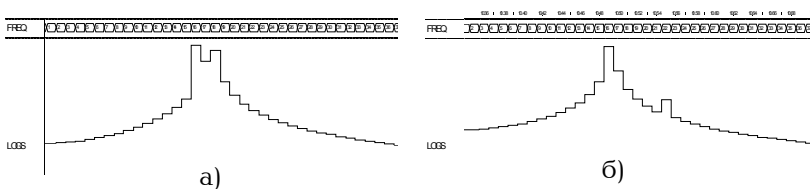


Рис. 4.27. Спектр при мінімальній різниці  $\Delta f$  (а) та максимальній різниці амплітуд (б) при прямокутному вікні

Для заданого вікна одержано  $\Delta f = 2,16$  біна, динамічний діапазон  $D_S = 34,5$  дБ,  $D_W = 68,5$  дБ.

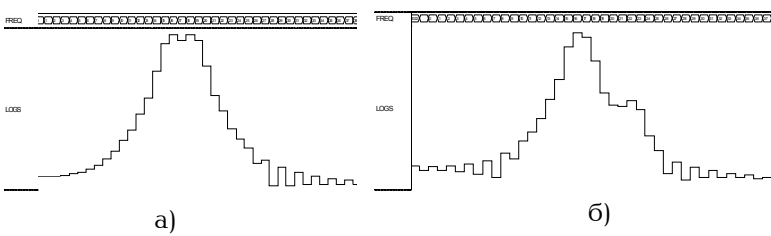


Рис. 4.28. Спектр як на рис. 4.27 при застосуванні спеціального вікна

Для заданого вікна при наявності гаусового шуму одержано  $\Delta f = 2,5$  біна, динамічний діапазон  $D_S = 34,7$  дБ,  $D_W = 53,9$  дБ.

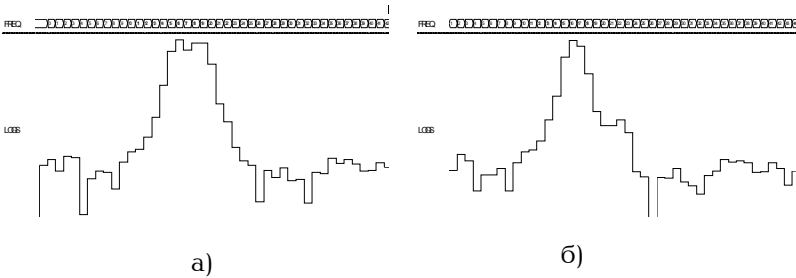


Рис. 4.29. Спектри як на рис. 4.28 при доданому шумові

Для заданого вікна при наявності гаусового шуму і 64 ітераціях осереднення одержано  $\Delta f = 2,45$  біна, динамічний діапазон  $D_S = 28,4$  дБ,  $D_W = 42,1$  дБ.

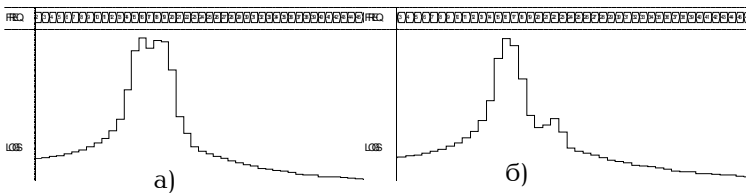


Рис. 4.30. Спектри як на рис. 4.29 після 64 осереднень

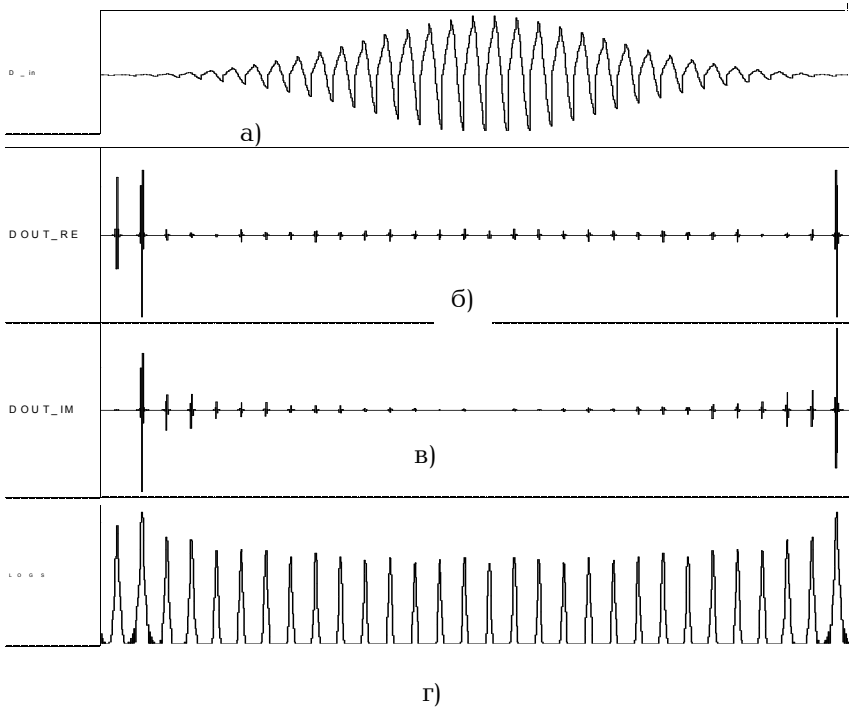


Рис. 4.31. Сигнал після множення на вікно (а), реальна (б) та уявна (в) частини його спектру й модуль спектру в логарифмічному масштабі (г)

Результати вимірювання спектру сигналу спеціальної форми: амплітуда нульової частоти (постійна складова)  $\tilde{S}(0) = 43,2$  дБ, амплітуда першої гармоніки  $\tilde{S}(34) = 51,3$  дБ, амплітуда другої та третьої гармонік  $\tilde{S}(68) = 36,5$  дБ,  $\tilde{S}(102) = 35,2$  дБ. Аналіз вимірів показує, що період сигналу дорівнює  $1024/34 = 30,1$  відліків, що доволі точно співпадає з заданим у генераторі періодом. Частоти гармонік сигналу кратні частоті першої гармоніки, що свідчить про незмінність періоду сигналу. Сигнал має постійну складову, потужність якої у 6 разів (на 8,1 дБ) менше потужності основної (першої) гармоніки. Друга та третя гармоніки за потужністю у 30 та у 40 разів, тобто на 14,8 і 16,1 дБ менші за першу гармоніку. Це означає, що сигнал є далеким від синусоїдального і має несиметричну форму.

Загальні висновки наступні. Застосування спеціального часового вікна збільшує динамічний діапазон вимірювання спектру з 21,6 дБ до 34,5 дБ, а при доволі великій відстані між бінами, у яких шукаються сигнали — до 68,5 дБ. Роздільна здатність такого вікна складає від 2,16 до 2,5 біна в залежності від наявності шуму у сигналі. Застосування осереднення суттєво покращує слушність оцінки спектру. Після 64 осереднень стає видно, що генератор шуму має рівномірний спектр, рівень якого на 53 дБ нижче за рівень сигналу, тобто його спектральна щільність потужності у  $2,4 \cdot 10^5$  менша.

## Літэратура

1. Введение в цифровую обработку сигналов / Под ред. Р.Богнера и А. Константиноидиса. — М.:Мир. — 1976. — 216 с.
2. Макс Ж. Методы и техника обработки сигналов при физических измерениях. Т1. Основные принципы и классические методы. — М.:Мир. — 1983. — 311 с.
3. Марпл С. Л. мл. Цифровой спектральный анализ и его приложения. — М.:Мир. — 1990. — 584 с.
4. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. — М: Техносфера. — 2006. — 856с.
5. Поляков А.К. Языки VHDL и VERILOG в проектировании цифровой аппаратуры. — М.: Солон, 2003.- 320 с.
6. Сато Ю. Без паники! Цифровая обработка сигналов. — М.: ИД Додэка—XXI. — 2005. — 175 с.
7. Сергиенко А.М. VHDL для проектирования вычислительных устройств. —К.: —"ДиаСофт". —2003. —210 с.
8. Сэломон Д. Сжатие данных, изображений и звука. — М: Техносфера. — 2004. — 368с.
9. Яне Б. Цифровая обработка изображений. — М.: Техносфера. — 2007. — 583 с.
10. Aldec's New Co-Simulation Wizard for Simulink Offers Support for Mixed HDL/System Design. October 22, 2003. Режим доступа: <http://www.aldec.com>
11. Kuo S. M., Lee B. H. Wenshun T. Real-Time Digital Signal Processing. Implementations and Applications. — Wiley. — 2006. — 646 p.
12. Regalia P., Mitra S.K., Vaidyanathan P.P. The Digital All-Pass Filter: A Versatile Signal Processing Building Block // Proc. IEEE. — 1988. —V.76. —№1. — p.19–37.
13. Сайт Matlab. Режим доступа: <http://matlab.exponenta.ru>
14. Сайт VHDL-AMS. Режим доступа: <http://www.vhdl-ams.org>