

Міністерство освіти і науки України
Національний університет “Львівська політехніка”

Є. Ваврук, О. Лашко, Р. Попович

АЛГОРИТМИ ТА ЗАСОБИ ОБРОБКИ СИГНАЛІВ

Навчальний посібник

Львів, СПОЛОМ, 2021

УДК 004.383.3+519.65

Ва 12

*Рекомендовано Навчально-методичною радою Національного університету
“Львівська політехніка” як навчальний посібник для студентів спеціальності
123 “Комп’ютерна інженерія” (протокол № 42 від 23.05.2019).*

Рецензенти:

Кочан Р. В. – докт. техн. н., проф., професор кафедри спеціалізованих комп’ютерних систем Національного університету “Львівська політехніка”;
Куриляк Д. Б. – докт. фіз.-мат. н., с.н.с., завідувач відділу фізичних основ діагностики матеріалів Фізико-механічного інституту ім. Г. В. Карпенка НАН України;
Новосядлий С. П. – докт. техн. н., проф., професор кафедри комп’ютерної інженерії та електроніки Прикарпатського національного університету ім. В. Стефаника.

Ваврук, Євгеній Ярославович.

Алгоритми та засоби обробки сигналів : навч. посібн. / Ваврук Є., Лашко О., Попович Р. – Львів : СПОЛОМ, 2021. – 240 с. : іл. – Бібл.: с. 237–239 (34 назви).

Розглянуто основні питання проектування алгоритмічних, програмних та апаратних засобів опрацювання сигналів, вивчення яких допоможе студентам спеціальності “Комп’ютерна інженерія” здобути початкові практичні навички в оптимальному виборі алгоритмів, їх програмній та апаратній реалізації.

Для викладачів та студентів вищих навчальних закладів четвертого рівня акредитації.

ISBN 978-966-919-717-7

© С. Ваврук, О. Лашко, Р. Попович, 2021
© Вид-во «СПОЛОМ», 2021

ЗМІСТ

ВСТУП	6
РОЗДІЛ 1. ЗАГАЛЬНІ ПІДХОДИ ДО ОПРАЦЮВАННЯ СИГНАЛІВ	8
1.1. Узагальнена схема системи цифрового опрацювання сигналів	8
1.2. Основні типи сигналів.....	9
1.3. Специфічні сигнали в обробці сигналів	15
1.4. Особливості організації обчислювальних засобів	21
1.5. Завдання для самоперевірки	24
РОЗДІЛ 2. БАЗОВІ АЛГОРИТМИ ОБРОБКИ ТА ЇХ ОПЕРАЦІЇ	26
2.1. Опрацювання сигналів у часовій області подання.....	26
2.2. Опрацювання сигналів в частотній області подання.....	34
2.3. Швидке перетворення Фур'є.....	40
2.4. Деякі інші ортогональні тригонометричні перетворення	49
2.5. Завдання для самоперевірки.....	52
РОЗДІЛ 3. МОДУЛЯЦІЯ ТА ДЕМОДУЛЯЦІЯ СИГНАЛІВ	54
3.1. Класифікація видів модуляції	54
3.2. Основні характеристики видів модуляції	55
3.3. Обчислення параметрів модульованих сигналів.....	62
3.4. Завдання для самоперевірки.....	63
РОЗДІЛ 4. ЦИФРОВА ОБРОБКА СИГНАЛІВ ПРИ ДЕКІЛЬКОХ ШВИДКОСТЯХ	65
4.1. Області застосування методів обробки при декількох швидкостях	65
4.2. Зміна частоти дискретизації	65
4.3. Перетворення частоти дискретизації з нецілим кроком.....	69
4.4. Багатокасадне перетворення частоти дискретизації.....	72
4.5. Завдання для самоперевірки	74
РОЗДІЛ 5. ПРОЕКТУВАННЯ ЦИФРОВИХ ФІЛЬТРІВ	75
5.1. Лінійні дискретні системи обробки сигналів: опис в часовій області.....	75
5.2. Лінійні дискретні системи обробки сигналів: опис в частотній області	82
5.3. Основні операції фільтрації та типи фільтрів.....	87
5.4. Розрахунок і побудова цифрових СІХ-фільтрів з частотною вибіркою.....	91
5.5. Завдання для самоперевірки	97
РОЗДІЛ 6. ОПРАЦЮВАННЯ МОВНИХ СИГНАЛІВ	99
6.1. Мовні технології.....	99
6.2. Найпростіший розпізнавач мови.....	101

6.3. Попереднє опрацювання мовних сигналів.....	103
6.4. Алгоритм динамічного часового вирівнювання для розпізнавання слів із невеликого словника	108
6.5. Розпізнавання злитної мови з великим словником	122
6.6. Завдання для самоперевірки	132
РОЗДІЛ 7. ОБРОБКА НЕРУХОМИХ ЗОБРАЖЕНЬ.....	134
7.1. Стиск нерухомих зображень з використанням дискретних косинусних перетворень	134
7.2. Поняття хвилькового перетворення	141
7.3. Стиск зображень з використанням методу кодування областей хвилькового перетворення	148
7.4. Стиск зображень з використанням методу дерев нулів хвилькового перетворення	148
7.5. Адаптивні хвилькові перетворення	160
7.6. Завдання для самоперевірки	164
РОЗДІЛ 8. ОРГАНІЗАЦІЯ ЦИФРОВИХ СИГНАЛЬНИХ ПРОЦЕСОРІВ ...	167
8.1. Вимоги до процесорів при опрацюванні сигналів	168
8.2. Типова структура цифрового сигнального процесора	170
8.3. Інтерфейси сигнальних процесорів	171
8.4. Завдання для самоперевірки	183
РОЗДІЛ 9. ПРОЕКТУВАННЯ ПРОЦЕСОРА ШПФ НА ЦИФРОВИХ СИГНАЛЬНИХ ПРОЦЕСОРАХ.....	185
9.1. Процедури визначення параметрів процесора	185
9.2. Основи вибору сигнальних процесорів.....	188
9.3. Проектування процесора ШПФ	191
9.4. Завдання для самоперевірки	195
РОЗДІЛ 10. ПРОЕКТУВАННЯ ЗАСОБІВ ОПРАЦЮВАННЯ СИГНАЛІВ НА ПЛІС.....	197
10.1. Однокристална реалізація на ПЛІС алгоритму ШПФ	197
10.2. Оцінка продуктивності вузла реалізації алгоритму ШПФ на ПЛІС	198
10.3. Приклад розробки процесора ШПФ на ПЛІС	199
10.4. Завдання для самоперевірки	206
РОЗДІЛ 11. РЕАЛІЗАЦІЯ АЛГОРИТМІВ ОПРАЦЮВАННЯ СИГНАЛІВ НА НЕЙРОПРОЦЕСОРАХ.....	208
11.1. Основні характеристики нейрокомп'ютерів.....	208
11.2. Визначення продуктивності нейрокомп'ютерів	212
11.3. Організація паралельних обчислень алгоритмів ШПФ на нейрокомп'ютері NM6403	213
11.4. Завдання для самоперевірки	220

РОЗДІЛ 12. ДІАГНОСТИКА І КОНТРОЛЬ ПРОЦЕСОРІВ І СИСТЕМ ОПРАЦЮВАННЯ СИГНАЛІВ	222
12.1. Особливості діагностики та контролю процесорів та систем опрацювання сигналів	222
12.2. Ієрархічність засобів діагностики та контролю процесорів та систем опрацювання сигналів	226
12.3. Процес формування АЧХ	231
12.4. Визначення і дослідження виду АЧХ.....	233
12.5. Завдання для самоперевірки	235
СПИСОК ЛІТЕРАТУРИ.....	237

ВСТУП

В останні роки цифрове опрацювання сигналів (ЦОС) суттєво впливає на такі ключові технологічні галузі, як телекомунікації, цифрове телебачення і засоби інформації, біомедицина і цифровий звукозапис. Сьогодні ЦОС є основою багатьох новітніх видів цифрових розробок і різних застосувань в інформаційному середовищі (наприклад, цифровий мобільний зв'язок, цифрові відеокамери, телебачення і системи звукозапису). Поряд з тим ЦОС ширше впроваджується і в класичні галузі застосування (радіолокація, геофізика, опрацювання мовних сигналів, сейсмологія, системи зв'язку і передачі даних, медична і технічна діагностика, системи керування). Такому широкому застосуванню сприяли успіхи в розробці швидких алгоритмів і методів ЦОС (наприклад, хвилькові перетворення), досягнення в технології конструювання інтегральних схем (архітектура нових цифрових сигнальних процесорів (ЦСП, digital signal processor, DSP), програмовані логічні інтегральні схеми (ПЛІС), системи на кристалі), використання програмних пакетів (MATLAB, SCILAB, бібліотеки програм на мові C).

Суть ЦОС як області науки пролягає у розв'язку на обчислювальній машині чотирьох основних задач: представлення сигналів в зручній для сприйняття формі; виділення із сигналів корисної інформації; внесення в сигнали корисної інформації; формування сигналів із заданими параметрами.

Основні переваги ЦОС полягають в такому: можливості реалізації складних методів та алгоритмів опрацювання сигналів, недоступних для аналогових пристроїв; забезпеченні високої точності опрацювання; гнучкості й універсальності засобів, розвиненому користувачькому інтерфейсові тощо.

Головна проблема ЦОС полягає у підвищенні швидкодії при реалізації певного набору математичних операцій над сигналами. Цей набір, як правило, визначається проблемною орієнтацією засобів ЦОС. Для розв'язання цієї задачі комплексно застосовують два напрямки – підвищення ефективності

обчислювальних алгоритмів і вдосконалення архітектури комп'ютерних засобів. При цьому виходять з детального аналізу особливостей проблемної області.

Даний навчальний посібник укладений відповідно до навчальних програм з дисциплін “Обробка сигналів”, “Методи та засоби опрацювання сигналів”, “Проектування комп'ютерних засобів обробки сигналів та зображень”.

Метою посібника є допомогти засвоїти основні методи та алгоритми опрацювання сигналів, принципи та шляхи проектування апаратних і програмних комп'ютерних засобів опрацювання сигналів, набути початкові практичні навички проектування таких засобів.

У результаті вивчення згаданих дисциплін студенти повинні знати: типи і особливості алгоритмів опрацювання сигналів, математичні основи їх реалізації, характеристики елементної бази та основи реалізації алгоритмів на її основі, особливості і етапи розробки цифрових пристроїв на процесорах і вузлах, орієнтованих на задачі опрацювання сигналів; мови програмування та мови опису апаратних засобів; пакети моделювання, опрацювання і фільтрації сигналів. Також мають вміти: вибирати найефективніший метод і алгоритм, здійснювати математичне формування алгоритму та методу його розв'язання, розробляти блок-схеми алгоритмів, розробляти програмну реалізацію виконання алгоритмів, розраховувати технічні параметри апаратних і програмних засобів, проектувати структурні та функціональні схеми процесорів та вузлів опрацювання сигналів; моделювати і опрацьовувати сигнали різної форми.

РОЗДІЛ 1. ЗАГАЛЬНІ ПІДХОДИ ДО ОПРАЦЮВАННЯ СИГНАЛІВ

1.1. Узагальнена схема системи цифрового опрацювання сигналів

До основних галузей, де використовують опрацювання сигналів та зображень, належать:

1. Радіолокація – виявлення, фільтрація та накопичення сигналу.
2. Гідроакустика – контроль водяного простору, підводна сейсмологія, гідронавігація.
3. Зв'язок – підвищення надійності, пропускну здатності, скритності, виділення символів, згортка символних послідовностей, скорочення надлишковості, підвищення завадостійкості, керування транспортними засобами.
4. Геофізика – пошук нафтоносних (водоносних) шарів.
5. Біомедицина – візуалізація органів, діагностика.
6. Системи керування виробничими процесами – керування, знімання даних, передача інформації.
7. Промислова діагностика – неруйнівний контроль, візуалізація стану вузлів, діагностика віддалених об'єктів.

Узагальнена структурна схема системи ЦОС наведена на рис. 1.1. Вона відображає послідовність процедур, необхідних для перетворення вхідного аналогового сигналу у вихідний аналоговий сигнал за заданим алгоритмом, засобами цифрової обчислювальної техніки.

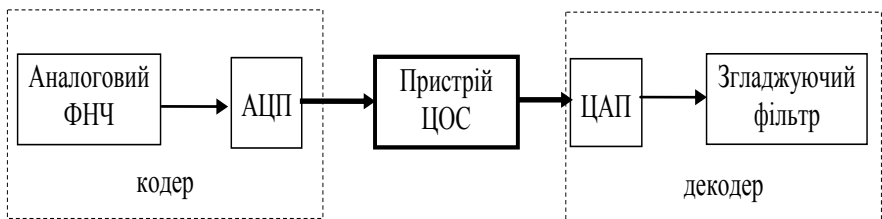


Рис. 1.1. Узагальнена структурна схема системи ЦОС.

В ЦОС виділяють три основні етапи: формування цифрового сигналу з вхідного аналогового; перетворення цифрового сигналу за заданим алгоритмом; формування вихідного аналогового сигналу з цифрового. Відповідно в схемі є три функціональні блоки: кодер, пристрій ЦОС, декодер.

Пристрій ЦОС може бути реалізований апаратно або програмно. В першому випадку – у вигляді спеціалізованого пристрою, а в другому – у вигляді програми на універсальному комп'ютері або на ЦСП. На сьогодні переважає програмна реалізація.

1.2. Основні типи сигналів

Під *сигналом* розуміємо будь-яку змінну, котра передає або містить якийсь вид інформації, і яку можна, наприклад, переносити, виводити на екран або виконувати з нею якісь дії.

За своєю природою, сигнали можуть бути випадкові або детерміновані. До детермінованих відносять сигнали, значення яких у будь-який момент часу або в довільній точці простору є апріорно відомими або можуть бути досить точно визначені (обчислені) за відомою або передбачуваною функцією, навіть якщо ми не знаємо її явного виду.

Випадкові сигнали в принципі не мають певного закону зміни своїх значень у часі або в просторі. Для кожного конкретного моменту (відліку) випадкового сигналу можна знати тільки ймовірність того, що він прийме яке-небудь значення в якій-небудь певній області можливих значень. Закон розподілу (функція розподілу – імовірність того, що випадкова величина прийме значення менше аргументу функції, або густина розподілу – похідна функції розподілу) далеко не завжди відомий [22,23].

Найбільш прості характеристики законів розподілу – середнє значення випадкових величин (математичне очікування) і дисперсія (математичне очікування квадрата відхилення від середнього), що характеризує розкид значень випадкових величин щодо середнього значення. Параметри динаміки випадкових

сигналів (процесів) у часі характеризуються функціями автокореляції (кількісна оцінка взаємозв'язку значень випадкового сигналу на різних інтервалах) або автоковаріації (те ж, при центруванні випадкових сигналів). Аналогічною мірою взаємозв'язку двох випадкових процесів і ступеня їхньої подібності по динаміці розвитку є взаємокореляція (кроскореляція) або взаємоковаріація (кросковаріація). Максимальне значення взаємної кореляції досягається при збігу двох сигналів. При затримці одного із сигналів стосовно іншого положення максимуму кореляційної функції дає можливість оцінити величину цієї затримки.

Математично сигнали описують функціями часу, тип яких залежить від типу сигналу. Основні типи – аналоговий, дискретний і цифровий сигнали [2].

Аналоговим називають сигнал, неперервний у часі і значеннях. Такий сигнал описується неперервною або кусочно неперервною функцією $f(t)$ при цьому і аргумент і функція можуть приймати довільні значення: $-\infty \leq t \leq \infty$; $-\infty \leq f(t) \leq \infty$.

Дискретизований – сигнал, дискретний у часі і неперервний у значенні. Такий сигнал описується решітчастою функцією – послідовністю $f_\delta(\Delta t)$, яка визначена тільки в конкретні значення аргументу і може приймати довільні значення.

Цифровий сигнал – це дискретизований сигнал, який набуває значень зі скінченної кількості рівнів квантування. Такий сигнал описується квантованою решітчастою функцією – квантованою послідовністю $f_q(\Delta t)$, відліки якої в кожен момент часу набувають дискретне значення рівнів квантування.

Цифрові сигнали формуються з аналогових за два етапи. Перший з них – *дискретизація*, другий – *квантування*.

Дискретизація – це послідовне вимірювання амплітудних значень сигналу через певні інтервали часу Δt (при цьому отримуються послідовні відліки $f_\delta(\Delta t)$). Умови, за яких можливе повне відновлення аналогового сигналу за його цифровим еквівалентом зі збереження всієї вхідної інформації, що містилася в сигналі, виражаються теоремами Котельникова, Найквіста, Уїттекера, Шеннона,

сутність яких практично однакова.

Для того, щоб визначити принцип дискретизації слід зрозуміти основні поняття представлення сигналів в часовій та частотній областях.

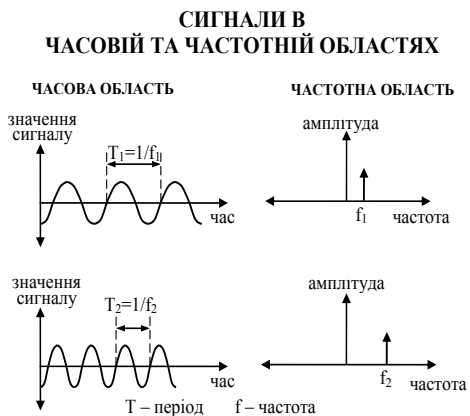


Рис.1.2. Часова та частотна області представлення сигналів.

Часова область зручна при зображенні змін сигналу в часі. Ми всі знаємо, що таке синусоїди. Кожна синусоїда характеризується трьома параметрами: амплітудою, початковою фазою й частотою. Одна синусоїда має одну частоту. Частота – це параметр, що показує як часто сигнал повторює сам себе. Зворотним до частоти є період. Він відповідає тривалості, що займає в часі один повтор сигналу. На рис. 1.2 показані дві синусоїди з різними частотами й, отже, різними періодами.

Синусоїда описується формулою $x(t) = A \sin(\omega t + \varphi)$, де ω – кругова частота (вимірюється в радіанах на секунду). Формула зв'язку між круговою і лінійною частотою (вимірюється в Герцах): $\omega = 2\pi f$ (якщо вимірювати кругову частоту в градусах на секунду, то $\omega = 360f$).

Частотна область зручна при зображенні частотного складу сигналів. Кожна синусоїда, представлена на рис. 1.2, має одну частоту. Отже, у частотній

області кожна синусоїда представляється тільки однією частотною складовою. Її амплітуда (на рис. 1.2. – пряма зі стрілкою нагору) у частотній області пропорційна амплітуді синусоїди в часовій області. Частота f_1 відповідає частоті першої синусоїди, а f_2 – другої. Чим вища частота синусоїди, тим далі по осі частот вона розташовується. Словосполучення “частотна складова” для стислості заміняють просто на “частоту”, якщо зрозуміло, що мова йде про складову частотного спектра, а не про поняття частоти як такої.

Реальні сигнали являють собою комбінацію з безлічі синусоїд з різними частотами, амплітудами й початковими фазами. Виходить, у частотній області реальний сигнал містить безліч частотних складових.

Спектр (*частотний*) сигналу відображає його точний частотний склад. Така інформація необхідна для правильної обробки сигналу. Частотний спектр сигналів можна обчислювати й оцінювати

Завжди виникає необхідність обмежувати спектр реальних сигналів. Можливість такого обмеження обумовлена особливостями частотного розподілу енергії сигналу: основна частина його енергій зосереджена в області $f \leq f_c$. Тобто амплітуди спектральних складових, починаючи з певної частоти $f > f_c$ суттєво зменшуються. Вибір f_c визначається конкретним типом сигналу та задачею, що вирішується. Так, у гладких функціях спектр спадає стрімко, а у функцій з різкими перепадами (“рвані” функції) спектр спадає повільно, оскільки для представлення розривів та зламів функції потрібні синусоїди з високими частотами. При обробці аудіо та відеосигналів вибір f_c залежить від особливостей сприйняття цих сигналів.

Частотний діапазон, або смуга частот це – проміжок між мінімальною та максимальною частотою, присутніми в сигналі.

Ширина *смуги* сигналу – це різниця між його найвищою й найнижчою частотами, при яких амплітуди перевищують задане значення. У цьому випадку це $2f_c$. Знати ширину смуги сигналу дуже корисно. Таким чином, ширина смуги не дає інформації про значення частот, що містяться в сигналі. Спектр сигналу

дозволяє їх побачити. Таким чином, два сигнали з однаковою шириною смуги можуть мати два зовсім різних спектри [10].

Теорема Котельникова. Якщо сигнал $x(t)$ обмежений смугою $(0, f_{\text{граничне}})$, то він може бути відтворений з як завгодно великою точністю за відліками, що взяті з частотою дискретизації $2f_c$:

$$\Delta t \leq \frac{1}{2f_c} \quad \text{або} \quad f_{\text{д}} \geq 2f_c. \quad (1.1)$$

Іншими словами: якщо сигнал має спектр, обмежений частотою f_c то після дискретизації сигналу з частотою не менше $2f_c$ можна відновити вхідний неперервний сигнал за отриманим цифровим абсолютно точно. Для цього слід інтерполювати дискретний сигнал між відліками функціями спеціального виду, які ще називають базисом Котельникова:

$$f(t) = \sum_{n=-\infty}^{\infty} f(n\Delta t) \frac{\sin\left(\pi \frac{t-n\Delta t}{\Delta t}\right)}{\pi \frac{t-n\Delta t}{\Delta t}}.$$

Якщо умова теореми Котельникова порушується, у дискретному сигналі виникає ефект маскування (підміни) дійсних частот “гаданими” більш низькими частотами. Цей ефект також називають накладанням спектрів, або елайсингом (aliasing). При накладанні спектрів, відновлений сигнал буде виглядати так, ніби частоти, що знаходяться вище половини частоти дискретизації, віддзеркалилися від половини частоти дискретизації, перейшли в нижню частину спектру і наклалися на його нижні складові (частоти) [1,3].

Для того, щоб уникнути накладання спектрів можна використовувати більш високу частоту дискретизації. Однак вибір надто малого Δt приводить до надлишковості обчислень.

На практиці, як вже згадувалося, перед оцифруванням сигналів здійснюється обмеження їх спектру за допомогою фільтру нижніх частот (НЧ-фільтр). Ці фільтри видаляють з сигналу всі частоти, що лежать вище заданої, яка ще називається *частотою зрізу* фільтра. Для анти-елейсингових фільтрів частота

зрізу встановлюється рівною половині частоти дискретизації.

Фізично, перетворення сигналу в цифрову форму здійснюється аналого-цифровими перетворювачами (АЦП) а в них, як правило, завжди вбудовується анти-елайсинговий фільтр.

За теоремою Котельникова, верхня (гранична) частота f_z аналогового сигналу не повинна перевищувати половину частоти дискретизації f_0 цього сигналу. З цього випливає, що дискретні сигнали є сенс розглядати в області $[0; f_0/2]$, яка називається *основною смугою частот*, або *основним діапазоном частот*. Частоту $f_N = f_0/2$ прийнято називати *частотою Найквіста*.

Квантування проводиться з метою представлення точних значень сигналу (відліків) у вигляді двійкових чисел скінченої розрядності – цифрових відліків $f_u(\Delta t)$. Для цього динамічний діапазон дискретного сигналу розбивається на скінчену кількість дискретних рівнів – рівнів квантування, і кожному відліку за певним правилом присвоюється значення одного з найближчих рівнів. Рівні квантування кодуються двійковими числами, що мають розрядність b , яка залежить від кількості рівнів квантування $R \leq 2^b$. Звідси випливає, що $b = \lceil \log_2 R \rceil$, де $\lceil \cdot \rceil$ – ціле значення, отримане методом заокруглення до більшого.

Наприклад, якщо обрано 5 рівнів квантування (без врахування знаку), то відліки будуть кодуватися 4-розрядними двійковими числами: один розряд знаковий і три значущих ($b = \lceil \log_2 5 \rceil = 3$).

Сукупність квантованих відліків $f_u(n\Delta t)$, $n=0,1,\dots$ називають, як було вже згадано, цифровим сигналом. Відстань між двома сусідніми рівнями квантування називається кроком квантування. Квантування приводить до появи похибки $e_{кв} = f_u - f_0$, оскільки має місце операція заокруглення.

Квантування, як і дискретизацію, реалізують за допомогою АЦП. У цьому разі використовують двійкову систему подання при рівномірній шкалі з певним числом розрядів. Збільшення кількості розрядів підвищує точність вимірів і розширює динамічний діапазон вимірюваних сигналів. Втрачена через брак розрядів АЦП інформація невідновна, й існують лише оцінки похибки, наприклад,

через потужність шуму, породженого помилкою в останньому розряді.

Для того щоб оцінити вплив перешкоди, вводиться поняття “відношення сигнал-шум” – відношення потужності сигналу до потужності шуму (вимірюють у децибелах):

$$SNR = 20 \cdot \lg \left(\frac{P_c}{P_w} \right) \text{ (дБ)}. \quad (1.2)$$

Найчастіше використовують 8-, 10-, 12-, 16-, 20- і 24-х розрядні АЦП. *Кожен додатковий розряд покращує відношення сигнал-шум на 6 децибел.* Однак збільшення кількості розрядів знижує швидкість дискретизації й збільшує вартість апаратури.

Важливим аспектом є також *динамічний діапазон*, який визначається як відношення максимального рівня сигналу до мінімального, з яким може працювати система аналогово-цифрового перетворювача. Динамічний діапазон найчастіше виражається в децибелах через кількість бітів перетворювача:

$$D = 20 \cdot \lg 2^b. \quad (1.3)$$

Іноді, динамічний діапазон визначається через потужність сигналу. Наприклад в цифрових системах записування і відтворення звуку він може визначатися як відношення максимальної потужності сигналу до мінімальної, яку можна відрізнити від потужності шуму. Якщо використовується лише АЦП, *апертурний час* – це час перетворення АЦП, тобто період часу, протягом якого аналоговий вхід повинен залишатися постійним, щоб можна було виконати перетворення. Відносно схеми вибірки-зберігання, це час необхідний для запису, після того, як віддана команда на зберігання.

Для зворотного перетворення використовується цифро-аналоговий перетворювач (ЦАП), основні характеристики якого (розрядність, частота дискретизації, число каналів і т.п.) аналогічні характеристикам АЦП.

1.3. Специфічні сигнали в обробці сигналів

Фінітні сигнали. Сигнал, що визначений лише на деякому часовому проміжку і не існує поза ним, називається фінітним, тобто при $t > T$, сигнал рівний нулю.

Періодичні сигнали. Періодичні сигнали – це математична модель. Реально вони не можуть існувати, оскільки в такому випадку вони повинні були б проявлятися на безмежному відрізку часу. Це означає, що якщо в математичну модель процесу закладена періодичність то всі його характеристики можна дослідити на одному періоді.

У дискретному варіанті маємо: $x(n) = x(n + k \cdot N)$, $k = 0, \pm 1, \pm 2, \pm \dots$, де N – період, $0 < N < \infty$, n, N – цілі числа, що визначають номер відліку.

В загальному випадку, для переходу від неперервного до дискретного сигналу відбувається дискретизація з кроком Δt . Тому для при поданні цифрових сигналів слід пам'ятати, що номер відліку на часовій осі відповідає $n = n\Delta t$.

Парні і непарні сигнали. Парним називається сигнал (функція), для якого виконується умова: $x(-t) = x(t)$, а непарним – для якого виконується умова: $x(-t) = -x(t)$.

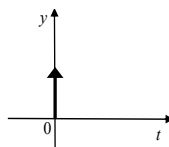
Елементарні сигнали. З математичної точки зору, елементарними називаються функції, які мають невелику кількість параметрів. В обробці сигналів елементарні сигнали посідають важливе місце, оскільки вони дозволяють легко і наглядно інтерпретувати те чи інше фізичне явище або процес. Тобто, якщо треба дослідити деяке складне явище, то складний сигнал розбивається, при можливості, на елементарні. Розглянемо найбільш вживані елементарні сигнали, їх ще називають *типовими*.

I. Неперервний випадок

Дельта –функція (δ-функція):

(функція Дірака)

$$\delta(t) = \begin{cases} \infty, & t = 0 \\ 0, & t \neq 0 \end{cases}$$

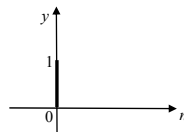


Це нескінченно вузький імпульс з нескінченно великою амплітудою, розташований в нульовому значенні аргументу. При цьому, площа цього імпульсу дорівнює одиниці. Розмірність δ -функції обернена розмірності її аргументу, тобто δ -функція часу має розмірність частоти (1/с).

Дискретний аналог

Одиничний імпульс

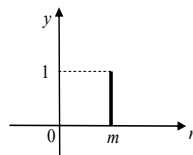
$$u_0(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases}$$



(функція Кронекера)

Сигнал рівний одиниці, якщо аргумент рівний нулю, та рівний нулю в інших випадках. Важливість цього сигналу в тому, що він має стробуючу властивість, або її ще називають властивістю селективності, або фільтруючою властивістю одиничного імпульсу. Для розуміння цієї властивості слід розглянути затриману дельта-функцію, або затриманий одиничний імпульс. Вони описуються відповідними виразами:

$$u_0(n-m) = \begin{cases} 1, & n = m \\ 0, & n \neq m \end{cases}$$



Фільтруюча властивість полягає в тому, що виконується наступна рівність:

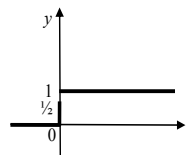
$$x(m) = \sum_{n=0}^{\infty} x(n)u_0(n-m).$$

Це визначення показує як подати дискретну послідовність через суму членів цієї ж послідовності.

II. Функція одиничного стрибка $\sigma(t)$, вона ще називається функцією Хевісайда, або функцією вмикання.

Аналоговий випадок

$$\sigma(t) = \begin{cases} 0, & t < 0 \\ 1/2, & t = 0 \\ 1, & t > 0 \end{cases}$$

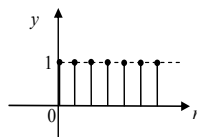


Сигнал рівний нулю для від'ємних значень аргументу і одиниці для додатних. При нульовому значенні аргументу, функцію вважають або невизначеною, або рівною $\frac{1}{2}$. Цю функцію зручно використовувати при створенні математичних описів для сигналів скінченної тривалості. Загалом, довільну кускову функцію можна подати у виді одиничного стрибка.

Дискретний випадок: *цифровий одиничний стрибок*.

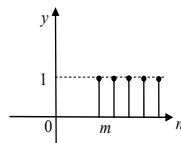
Сигнал рівний одиниці при всіх невід'ємних значеннях n :

$$u_1(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases}$$



Затриманий одиничний стрибок описується аналогічно до одиничного імпульсу:

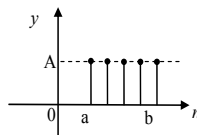
$$u_1(n-m) = \begin{cases} 1, & n \geq m \\ 0, & n < m \end{cases}$$



III. *Прямокутний імпульс*.

Дискретний (цифровий) випадок:

$$y(n) = \begin{cases} A, & a < n < b \\ 0, & t < a, n > b \end{cases}$$



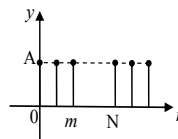
Прямокутний імпульс, як кускову функцію можна легко подати через функцію Хевісайда:

$$y(n) = A(u_1(n-a) - u_1(n-b)).$$

IV. *Послідовність прямокутних імпульсів*.

Розглядається сигнал, період якого дорівнює N :

$$y(n) = \begin{cases} A, & 0 < n \leq m \\ 0, & m < n < N \end{cases}$$



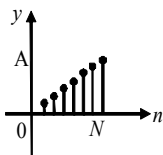
Для послідовності прямокутних імпульсів прийнято розглядати відношення тривалості періоду (N) до тривалості самого імпульсу (m). Цей параметр називається *щільність* (шпаруватість) (англ. porosity або on-off time ratio), а обернена величина – *наповненість*: $k=N/m$.

У зв'язку з цим, важливим є випадок послідовності прямокутних імпульсів коли щільність рівна 2. Такий сигнал називається *меандр*. В ньому тривалість імпульсів та проміжків між ним є рівною.

V. Несиметричний трикутний імпульс.

Аналоговий випадок:

$$y(t) = \begin{cases} A \frac{t}{\tau}, & 0 \leq t \leq \tau \\ 0, & t < 0, t > \tau \end{cases}$$



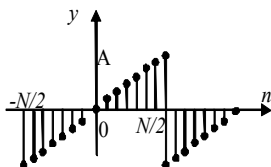
Дискретний (цифровий) випадок

$$y(n) = \begin{cases} A \frac{n}{N}, & 0 < n \leq N \\ 0, & n < 0, n > N \end{cases}$$

VI. Послідовність несиметричних трикутних імпульсів –

пиллоподібний сигнал.

Тут, як правило розглядаються несиметричні трикутні імпульси, центровані відносно початку відліку часу з періодом повторення N . Тому на графіку присутня від'ємна частина сигналу. В межах одного періоду такий сигнал описується лінійною функцією.



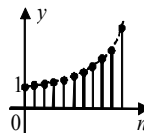
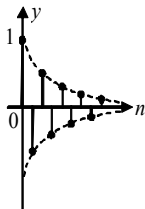
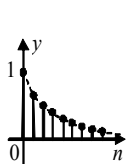
VII. Загальна показникова функція.

Це ще один важливий елементарний сигнал, який є суттєвим для опису сигналів:

$$y(n) = \begin{cases} a^n, & n \geq 0 \\ 0, & n < 0 \end{cases}, \quad \text{де } a \text{ – дійснозначна константа.}$$

Залежно від значення константи a та її знаку дискретна показникова функція матиме вигляд:

$|a| < 1, a > 0$ – спадаюча; $|a| < 1, a < 0$ – спадаюча
 знакопостійна; $|a| < 1, a < 0$ – спадаюча
 знакозмінна; $|a| > 1$ – зростаюча;



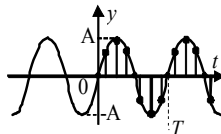
$a=1$ – цифровий одиничний стрибок (функція Хевісайда)

$a=-1$ – знакозмінна послідовність одиниць

VIII. Гармонійний сигнал (гармонійне коливання).

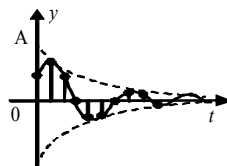
Дискретна синусоїда отримується з аналогової шляхом заміни неперервного часу дискретними відліками, тобто підстановкою $n\Delta t$ замість t :

$$y(n) = y(n\Delta t) = A \sin(\omega n \Delta t + \varphi) = A \sin(2\pi f n \Delta t + \varphi) = A \sin(\omega n + \varphi)$$



IX. Затухаюче коливання.

$$y(n) = A e^{-an} \sin(\omega n + \varphi)$$



X. Комплексний гармонійний сигнал (комплексна експонента).

Аналоговий випадок:

$$y(t) = A e^{j\omega t}, \quad -\infty \leq t \leq \infty$$

Дискретний (цифровий) випадок

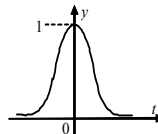
$$y(n) = A e^{j\omega n \Delta t}$$

Враховуючи формулу Ейлера, цей сигнал можна описати двома дійсними послідовностями – косинусоїдою (дійсна частина) та синусоїдою (уявна частина):

$$y(n) = A \cos(\omega n \Delta t) + j \sin(\omega n \Delta t).$$

XI. Гаусівський імпульс.

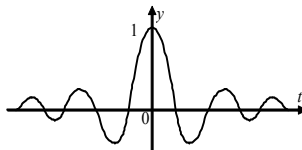
$$y(t) = Ae^{-a^2 t^2}$$



Цей імпульс має нескінчену тривалість в обох напрямках на часовій осі. Важливою властивістю цієї функції є те, що спектр даної функції також є Гаусівським імпульсом.

XII. Сигнал вигляду:

$$y(t) = \frac{\sin(t)}{t}$$



Даний сигнал має ідеальний низькочастотний спектр. Тобто його спектральна функція є сталою в певній смузі частот, починаючи з нульової, і не існує поза цією смугою (фактично спектр є прямокутним імпульсом).

Всі наведені елементарні сигнали є вивченими [10,22,23] і мають відомі характеристики. Існують таблиці спектрів цих функцій, що дозволяє швидко аналізувати сигнали, розкладені на елементарні складові.

1.4. Особливості організації обчислювальних засобів

Методи аналізу архітектур обчислювальних засобів. Технічно системи керування та опрацювання інформації реалізуються як комплекс спеціалізованих і універсальних засобів обчислювальної техніки. Головними вимогами при їх проектуванні є адаптація структури на виконання задач в режимі реального часу (РРЧ) й забезпечення вагогабаритних показників, споживаної потужності та оптимального співвідношення між обсягами апаратних засобів та програмного

забезпечення. Такий підхід забезпечує найкращі швидкісні та вагогабаритні параметри вузлів, однак робить їх частково вузькоспеціалізованими. Тому при проектуванні та аналізі систем вибирається метод, який забезпечує виконання вищенаведених вимог, і одночасно дозволяє реалізувати апаратні та програмні засоби на базі універсальних вузлів, або можливості оперативного переналаштування спеціалізованих вузлів на виконання інших задач. При синтезі бортових систем користуються змістовними (евристичними, інтуїтивними) та формальними (алгоритмічними) методами, більшість з яких орієнтована на не взаємозв'язані моделі систем. Такі моделі відображають різні боки системної організації, але не враховують цілісність систем в функціональному і структурному аспектах та концепцію їх розвитку. В цьому напрямку розглядаються два підходи до проектування систем: структурно-функціональний і функціонально-структурний. Перший напрямок зводиться до розбиття складних систем на окремі структурні рівні (підсистеми) для визначення їх функціонального призначення. Він є обмеженим, як концепція синтезу систем, що не дозволяє ефективно використовувати його при створенні систем цифрового опрацювання. Другий напрямок розглядає питання розвитку та взаємодії програмних і апаратних засобів, аналізу функцій систем і передбачає: цілісний підхід до аналізу і синтезу багаторівневих систем; облік взаємозв'язку функцій і структури об'єктів (при визначаючій ролі функції по відношенню до структури); облік енергетичних і інформаційних зв'язків як між елементами системи, так і системи з зовнішнім середовищем; розгляд системи в розвитку.

Процесори в задачах опрацювання сигналів та зображень. Структура системи визначається в основному специфікою задач і алгоритмів, що вирішуються комплексом. Загальний варіант структурної схеми системи комплексного опрацювання інформації наведено на рис. 1.2, де Д – давачі, П – приймач, ВАО – вузол аналогової обробки, ВД – вузол дискретизації, ОБ – обчислювальний вузол, ВК – вузол керування, ВР – вузол реєстрації, відображення, додаткового опрацювання, ВП – виконавчий пристрій,

$\underline{x} = \{x_1, \dots, x_n\}$ – дискретна вибірка. Блок давачів виконує функції перетворення вхідних даних $\alpha_i(t)$ у вихідні сигнали $S_1(\alpha_i, \beta, t)$, де β – вектор неінформаційних параметрів сигналів. В приймачі виконується попереднє опрацювання суміші сигналів і завад $\xi(t) = S_1(\alpha_i, \beta, t) + n_1(t)$, різні типи фільтрації зовнішніх $n_1(t)$ і внутрішніх $n_2(t)$ завад, нормування вихідних процесів і аналогово-цифрове перетворення $x(t)$. Функції інших вузлів залежать від області застосування системи. У цьому разі, частина обчислювальних функцій і алгоритмів виконується вузлами ОВ і ВК. В даній структурі вузли ВД, ОВ, ВК, ВР можуть бути реалізовані на базі універсальних процесорів, ЦСП, ПЛІС, систем на кристалі [10].

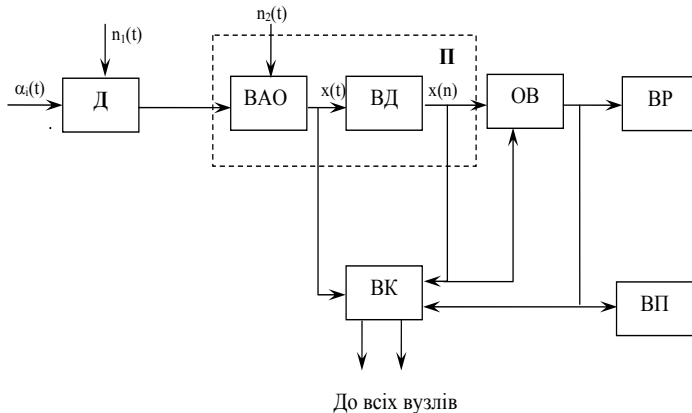


Рис. 1.2. Структурна схема системи комплексного опрацювання інформації.

Напрямки реалізації апаратних засобів на базі яких проектуються засоби опрацювання є такими:

- програмовані процесори;
- процесори на базі комплектів спеціалізованих НВІС функціональних вузлів;
- спеціалізовані НВІС процесори, які виконують конкретні алгоритми ЦОС;
- процесори на базі НВІС програмованих середовищ;
- процесори ЦОС (ЦСП), які конструктивно вмонтовані в комп'ютер;
- процесори на базі ПЛІС.

Шляхи та засоби підвищення продуктивності засобів цифрового опрацювання наведені в табл.1.1.

Таблиця. 1.1. Шляхи та засоби підвищення продуктивності

Шляхи підвищення ефективності	Засоби підвищення ефективності		
	Підвищення регулярності структури	Розширення функціональних можливостей	Інші
Підвищення продуктивності	–	–	Розпаралелення процесу обчислення, апаратно-мікропрограмна реалізація укрупнених операторів
Мінімізація втрат при обміні інформацією	–	Режим розподілу ресурсів, укрупнення операторів	–
Зменшення складності апаратних засобів	Зменшення складності зв'язків за рахунок використання НВІС	–	–
Зменшення складності апаратних засобів за рахунок збільшення об'ємів виробництва	Використання модульного конструювання	Інваріантність до типу комп'ютера, збільшення числа операторів, що ефективно реалізуються, використання багатфункціональних модулів	–

1.5. Завдання для самоперевірки

1. Теорема Котельникова представляється виразом:

А) $\Delta t \leq \frac{2}{f_2}$; Б) $\Delta t \leq \frac{1}{2f_2}$; В) $\Delta t \geq \frac{1}{4f_2}$; Г) $\Delta t \leq \frac{1}{f_2}$; Д) немає правильної

відповіді

2. Відомо, що сигнал оцифровано з частотою 22 кГц. Який діапазон частот може бути коректно переданий таким цифровим записом ?

А) від 0 до 22 кГц; Б) від 22 до 44 кГц; В) від 1 до 10кГц; Г) від 0 до 44 кГц; Д) немає правильної відповіді.

3. Скільки ненульових частотних складових є в сигналі:

$$x(t) = 20 + 0.7 \cos(0.2t) + 3 \sin(60t - \pi / 60) ?$$

А) 1; Б) 3; В) жодної; Г) 4; Д) немає правильної відповіді.

4. Який сигнал описує така функція: $u_1(n) = \begin{cases} 1, & n \geq 0 \\ 0, & n < 0 \end{cases} ?$

А) одиничний цифровий імпульс; Б) затриманий одиничний цифровий імпульс; В) пилоподібний сигнал; Г) цифровий одиничний стрибок; Д) немає правильної відповіді.

5. Як називається така функція: $y(n) = \begin{cases} 0.1^n, & n \geq 0 \\ 0, & n < 0 \end{cases} ?$

А) зростаюча дискретна показникова функція; Б) спадаюча знакопостійна дискретна показникова функція; В) функція Гаусса; Г) спадаюча знакозмінна дискретна показникова функція; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1). Б; 2) В; 3) Д; 4) Г; 5) Б.

РОЗДІЛ 2. БАЗОВІ АЛГОРИТМИ ОБРОБКИ ТА ЇХ ОПЕРАЦІЇ

2.1. Опрацювання сигналів у часовій області подання

Обробка цифрових сигналів виконується або спеціальними процесорами, або на універсальних комп'ютерах за спеціалізованими програмами. Існують численні алгоритми ЦОС як загального типу – для сигналів у їх класичній часовій формі (телекомунікації, зв'язок, телебачення та ін.), так і спеціалізовані – у всіляких галузях науки й техніки (геоінформатиці, геології й геофізиці, медицині, біології, військовій справі тощо). Однак усі згадані алгоритми, як правило, блокового типу, тобто побудовані на як завгодно складних комбінаціях досить невеликого набору типових цифрових операцій. Це відображено в табл. 2.1.

Розглянемо основні типи алгоритмів.

Лінійна згортка (convolution) – основна операція ЦОС, особливо в режимі реального часу. Вираз згортки у вигляді суми аналогічний інтегралу згортки, що використовується в теорії лінійних аналогових систем. Проте, дискретна згортка має не лише теоретичне значення за аналогією з інтегралом згортки, а і безпосередньо застосовується при реалізації дискретних систем. В загальному випадку, для двох послідовностей $x(n)$ та $h(n)$ згортка $y(n)$ визначається виразом:

$$y(n) = x(n) \otimes h(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) \text{ або } y(n) = h(n) \otimes x(n) = \sum_{k=-\infty}^{\infty} x(n-k) \cdot h(k),$$

де \otimes – умовне позначення операції згортки.

У системах обробки сигналів послідовність $x(n)$ – це дані, які опрацьовують (наприклад, сигнал на вході системи), а $h(n)$ називають *ядром згортки (kernel)*.

Всі сигнали, що обробляються на комп'ютері, мають скінчену тривалість. Тому доцільно розглядати згортку для вхідних послідовностей $x(n)$ та $h(n)$ довжиною N та M відліків, відповідно. Якщо сигнал $x(n)$ відмінний від нуля лише на відрізьку від 0 до $N-1$ (що відповідає довжині N), а ядро згортки визначене на відрізьку від $-m_1$ до m_2 включно, тобто $M=m_1+m_2+1$, то при підстановці у формулу

Таблиця 2.1. Перелік основних алгоритмів і застосованих операцій

(РЛ – радіолокація, ГА – гідроакустика, ЗВ – зв'язок, ГФ – геофізика,
 БМ – біомедицина, СК – системи керування)

Алгоритми обробки і застосовувані операції	Області застосування					
	РЛ	ГА	ЗВ	ГФ	БМ	СК
Згортка	+		+	+		
ШПФ	+	+	+	+	+	
Перемноження елементів векторів	+	+			+	+
Додавання елементів векторів	+	+				+
Перетворення координат	+	+	+			
Обчислення тригонометричних функцій	+	+	+	+		+
Пошук максимальних значень	+					+
Операції над матрицями	+	+	+		+	
Табличне перетворення відліків зображення	+					
Сортування	+					+
Отримання квадратного кореня	+	+	+	+		+
Піднесення до степеня	+	+	+	+		+
Статистична обробка	+					
Нормування	+	+				
Розрахунок нормованих коефіцієнтів кореляції		+				
Операції повертання координат			+			
Логарифмування				+		
Потенціювання				+		
Номинальні розкиди координат					+	
Інтерполяція даних					+	+

згортки, отримаємо сигнал $y(n)$, який буде відмінний від нуля на відрізьку від $-m_1$ до $N-1+m_2$, включно. Таким чином, довжина результуючого сигналу буде $L=N+M-1$ відлік, тобто сума довжини вхідного сигналу та ядра згортки мінус одиниця. Тому в формулі згортки межі підсумовування зміняться:

$$y(n) = x(n) \otimes h(n) = \sum_{k=0}^{L-1} x(k) \cdot h(n-k) = \sum_{k=0}^{L-1} x(n-k) \cdot h(k) \quad (2.1)$$

Далі перелічено властивості згортки.

1. $x(n) \otimes h(n) = h(n) \otimes x(n)$ - можна переставляти місцями вхідний сигнал і ядро згортки (комутативність)

2. $x(n) \otimes h(n) \otimes h_1(n) = x(n) \otimes (h(n) \otimes h_1(n))$ - можна проводити згортку з ядром, яке є згорткою (асоціативність) .

3. $x(n) \otimes h(n) + x(n) \otimes h_1(n) = x(n) \otimes (h(n) + h_1(n))$ (дистрибутивність).

Кореляція (correlation) існує у двох формах: автокореляції й взаємної кореляції. *Взаємо-кореляційна функція* (ВКФ, *cross-correlation function*, CCF). Часто виникає задача виявлення одного сигналу в іншому. Наприклад, може бути відомо, що певна зовнішня подія генерує в давачі сигнал визначеної форми. Проте події можуть відбуватися майже одночасно, а сигнали від них – накладатися, перекриватися. Крім того, на виході давача може бути присутній шум, що ускладнить виявлення потрібних сигналів. Для надійного виявлення таких сигналів застосовується метод кореляційного аналізу. Формула обчислення взаємо-кореляційної функції:

$$B_{xg}(n) = y(n) = \sum_{k=-\infty}^{\infty} x(n+k) \cdot g(k) \quad (2.2)$$

Тут реальні межі підсумовування визначаються довжиною сигналу g .

Авто-кореляційна функція (АКФ, *correlation function*, CF). Використовується для того, щоб знайти кореляцію сигналу самого з собою вона показує найбільш імовірні відрізки повторення форми сигналу (так звані приховані періодичності). АКФ є кількісною інтегральною характеристикою форми сигналу, подає інформацію про структуру сигналу і його динаміку в часі.

Вона, по суті, є частковим випадком ВКФ для одного сигналу і в точці n являє собою скалярний добуток сигналу і його копії, зсунутої в часі на n відліків. Формула обчислення авто-кореляційної функції:

$$B_{xx}(n) = y(n) = \sum_{k=-\infty}^{\infty} x(n+k) \cdot x(k) \quad (2.3)$$

АКФ має максимальне значення при $n=0$ (множення сигналу на самого себе), і є парною функцією $B_{xx}(-n)=B_{xx}(n)$, тому значення АКФ для від'ємних координат зазвичай не обраховуються.

Лінійна фільтрація. Лінійна фільтрація використовується для придушення шумів, спектр яких не перетинається зі спектром сигналу і дозволяє виділити в чистому вигляді функцію, що описує явище, яке досліджується.

В часовій області, лінійна цифрова фільтрація – це згортка сигналу з імпульсною характеристикою фільтра:

$$s(k) = \sum_{n=0}^N h(n) \cdot y(k-n), \quad (2.4)$$

де $h(n)$, $n=0, 1, 2, \dots, N$ – коефіцієнти фільтра, $y(k)$ і $s(k)$ – вхід і вихід фільтра.

На практиці розрізняють два основних типи лінійних фільтрів: з скінченною (СІХ) і нескінченною (НІХ) імпульсними характеристиками. НІХ-фільтри мають рекурсивну структуру, яка описується різницеvim рівнянням

$$y(n) = \sum_{i=0}^M b_i x(n-i) - \sum_{i=1}^M a_i y(n-i), \quad n \geq 0; \quad (2.5)$$

де x та y – вхідна та вихідна реалізації процесу; a_i і b_i – постійні коефіцієнти, що характеризують властивості фільтра ($a_M \neq 0$); M – порядок фільтру.

СІХ-фільтри реалізуються на основі нерекурсивної структури, яка описується формулою (2.6) і може бути отримана з формули (2.5) якщо a_i ($i = \overline{1, M}$) покласти рівними нулю:

$$y(n) = \sum_{i=0}^M b_i x(n-i), \quad n \geq 0. \quad (2.6)$$

У цьому разі b_i можна назвати дискретною імпульсною характеристикою системи (фільтра).

Медіанна фільтрація. Медіанна фільтрація є нелінійним способом опрацювання одномірних та двомірних послідовностей (вибірок) і використовується для зменшення рівня імпульсних шумів. В порівнянні з лінійною, медіанна фільтрація має такі переваги: зберігає різкі перепади сигналу і добре згладжує імпульсний шум. Процес виконання фільтрації цього типу виконується за три етапи: сортування виборок реалізації в ковзаючому “вікні”; вибір середнього значення у “вікні” (медіана); заміна відліку, розташованого в середині вікна значенням медіан [2,23].

Модуляція сигналів. Системи реєстрації, обробки, інтерпретації, зберігання і використання інформаційних даних стають усе більше розподіленими, що вимагає комунікації даних по високочастотних каналах зв'язку. Як правило, інформаційні сигнали є низькочастотними й обмеженими по ширині спектра, на відміну від широкосмугових високочастотних каналів зв'язку, розрахованих на передачу сигналів від безлічі джерел одночасно із частотним поділом каналів. Перенос спектра сигналів з низькочастотної області у виділену для їхньої передачі область високих частот виконується операцією модуляції. При модуляції, значення інформаційного (*модульованого*) сигналу переносяться на певний параметр високочастотного (*несучого*) сигналу.

Функціональні перетворення сигналів. Під функціональними перетворенням мають на увазі такі зміни сигналу, які дозволяють подати його у іншій області дослідження, або як комбінацію простіших, елементарних, типових сигналів. Такі перетворення приводять до того, що можна легко визначити властивості та кількісні характеристики досліджуваних сигналів чи систем. Функціональні перетворення передбачають дуже широкий клас операцій, а їх виконання займає найбільше часу в реальних системах обробки сигналів [3,22,24].

Одним з можливих шляхів аналізу складного сигналу є подання його через контрольовану суму елементарних сигналів. При цьому спираються на теорему Вейерштрассе, яка полягає у наступному: якщо функція неперервно диференційована і обмежена у часі, то її можна подати у вигляді ряду:

$$x(t) = \sum_k a_k \varphi_k(t), \quad (2.7)$$

де φ_k – ортогональна система функцій, a_k – коефіцієнти ряду.

Важливим в аналізі сигналів та систем є можливість описати їх у різних областях. Так, аналогові і цифрові сигнали та лінійні системи можна описувати в часовій області.

Сигнали описуються в цій області функціями часу, а лінійні системи – характеристиками. Характеристика лінійної системи визначається як її реакція на деякий тестовий сигнал, тобто характеристика – це сигнал, що описується функцією часу. Співвідношення вхід/вихід лінійної системи описується лінійним рівнянням, яке встановлює зв'язок між вхідними і вихідними сигналами – функціями часу. Тип функцій часу визначається типом сигналу: неперервна функція $x(t)$ описує аналоговий сигнал, послідовність (решітчаста функція) $x(n\Delta t)$ описує дискретний сигнал.

Окрім часової, сигнали і системи можуть описуватися і в інших областях (в областях інших незалежних змінних), при цьому відповідні функції часу перетворюються у функції іншої змінної. Традиційно, сигнали і системи описують в області комплексної змінної. Для опису аналогових сигналів у p -області використовують перетворення Лапласа:

$$X(p) = L\{x(t)\} = \int_0^{\infty} x(t)e^{-pt} dt \quad (2.8)$$

Для опису дискретних сигналів у p -області можна використовувати дискретне перетворення Лапласа $D\{x(n\Delta t)\}$, але на практиці переходять до z -перетворення:

$$X(z) = Z\{x(n\Delta t)\} = \sum_{n=0}^{\infty} x(n\Delta t)z^{-n} \quad (2.9)$$

Використання саме z -перетворення, а не дискретного перетворення Лапласа зумовлено простішим математичним описом: воно дозволяє отримати алгебраїчні співвідношення, а не трансцендентні.

Для опису аналогових сигналів у частотній області використовується перетворення Фур'є:

$$X(\omega) = F\{x(t)\} = \int_0^{\infty} x(t) \cdot e^{-j\omega t} dt \quad (2.10)$$

Для опису дискретних сигналів у частотній області використовується ряд Фур'є (дискретно неперервне перетворення Фур'є):

$$X(\omega) = X(e^{j\omega\Delta t}) = \sum_{n=0}^{\infty} x(n\Delta t) e^{-j\omega n\Delta t} \quad (2.11)$$

Нормування сигналів. Одним з прикладів операції нормування сигналів є ділення комплексних чисел:

$$U_H(n,i) = \frac{U(n,i)}{U^*(i)} = \frac{\operatorname{Re}U(n,i)\operatorname{Re}U^*(i) + \operatorname{Im}U(n,i)\operatorname{Im}U^*(i)}{[\operatorname{Re}U^*(i)]^2 + [\operatorname{Im}U^*(i)]^2} + j \frac{\operatorname{Im}U(n,i)\operatorname{Re}U^*(i) - \operatorname{Re}U(n,i)\operatorname{Im}U^*(i)}{[\operatorname{Re}U^*(i)]^2 + [\operatorname{Im}U^*(i)]^2} \quad (2.12)$$

Сортування. В основі – дві операції: порівняння і пересилання даних.

Таблиця 2.2. Базові операції для задач керування та опрацювання інформації

Алгоритм	Базові операції
Реалізація фільтрів, обчислення кореляційної і взаємкореляційної функцій	Додавання, віднімання, множення, обчислення суми парних добутків
Алгоритми перетворення	Множення, додавання комплексних чисел; обчислення операцій ШПФ, ШПХ, множення послідовностей комплексних чисел
Алгоритми обчислення: коефіцієнтів взаємної кореляції, координат, відстаней, виконання вагового множення, тощо	Обчислення тригонометричних функцій, добування квадратного кореня, піднесення до степеня, ділення, сортування

Більшості з названих алгоритмів властиві регулярність, рекурсивність і локальність, що робить їх придатними для реалізації на НВІС. Зокрема, до локально рекурсивних алгоритмів відносяться алгоритми множення матриць, згортки, НІХ-фільтрація, сортування вибором; до глобально-рекурсивних – алгоритми швидких перетворень, бітове сортування [10].

На основі аналізу алгоритмів в табл. 2.2 виділено набір базових операцій для опрацювання сигналів. Детальніше конкретні алгоритми описані і реалізовані в подальших розділах посібника.

Основні параметри типових операцій. Основні параметри типових операцій наведені в табл. 2.3, де N – розмірність масиву інформації.

Таблиця 2.3. Параметри типових операцій

Тип операції	Складність виконання	Тип обчислювальних засобів	Необхідна швидкодія (MOPS)
Лінійні операції: просторова фільтрація, згортка, виявлення контурів, скалярний добуток, КІХ-фільтрація.	N	Скалярний	$10^2 - 10^5$
Операції другого порядку: лінійні перетворення, перетворення Фур'є, кореляція, перемноження матриці на вектор, сортування, медіанна фільтрація.	N^2	Векторний	$10^3 - 10^7$
Операції вищих порядків: матричні операції, спектральні обчислення, адаптивні операції, розв'язання задач лінійних алгебраїчних рівнянь	N^3	Матричний	$10^4 - 10^8$

Аналіз наведених задач і алгоритмів їх розв'язання показує, що вони мають такі особливості:

- широкий динамічний і частотний діапазон сигналів, що обробляються;
- велика інтенсивність і постійність потоків даних;

- великий об'єм обчислень з перевагою обчислювальних операцій над логічними;
- регулярність і рекурсивність алгоритмів;
- можливість розпаралелення алгоритмів в часі та в просторі;
- в основі більшості алгоритмів лежить математична операція множення/додавання;
- постійне ускладнення нових алгоритмів і підвищення вимог до точності даних;
- структура даних дозволяє застосувати їх векторне опрацювання з використанням просторового та часового паралелізму.

2.2. Опрацювання сигналів в частотній області подання

Більшість сигналів зручно аналізувати, розклавши їх на синусоїдальні, тобто частотні складові, які ще називають гармоніками. Так, наприклад, працює людське вухо – розкладає звук на окремі коливання різної частоти.

Одним з основних методів частотного аналізу й обробки сигналів є перетворення Фур'є. Фактично, перетворення Фур'є – це розклад функцій на синусоїди, тобто гармоніки. Розрізняють поняття “перетворення Фур'є” і “ряд Фур'є”. Перетворення Фур'є припускає неперервний розподіл частот, ряд Фур'є задається на дискретному наборі частот. Сигнали також можуть бути задані набором часових відліків або як неперервна функція часу. Це дає чотири варіанти перетворень:

- перетворення Фур'є з неперервним або з дискретним часом (частота є неперервною);
- ряд Фур'є з неперервним часом або з дискретним часом (частота є дискретною).

Найбільш практичною з погляду цифрової обробки сигналів є дискретизація і у часовій, і в частотній області, але не слід забувати, що вона є апроксимацією неперервного перетворення. Неперервне перетворення Фур'є дозволяє точно

представляти будь-які явища. Сигнал, представлений рядом Фур'є, може бути тільки періодичний. Сигнали довільної форми можуть бути представлені рядом Фур'є тільки наближено, тому що при цьому передбачається періодичне повторення розглянутого інтервалу сигналу за межами його задання. На стиках періодів при цьому можуть виникати розриви й злами сигналу, і виникати помилки обробки, викликані явищем Гіббса, для мінімізації яких застосовують певні методи (вагові вікна, продовження інтервалів задання сигналів тощо).

Отже, в частотній області дискретний сигнал $x(n\Delta t)$ описується своїм Фур'є-відображенням (Фур'є-образом), яке визначається за допомогою перетворення Фур'є (фактично, ряду Фур'є):

$$X(\omega) = X(e^{j\omega\Delta t}) = \sum_{n=0}^{\infty} x(n) e^{-jn\Delta t\omega}, \quad (2.13)$$

Фур'є-образ $X(\omega)$ називають комплексним спектром або просто *спектром* дискретного сигналу. Маючи відомий спектр сигналу, сам початковий сигнал $x(n\Delta t)$ знаходять за допомогою оберненого дискретно-неперервного перетворення Фур'є:

$$x(n) = \frac{\Delta t}{2\pi} \int_{-\pi/\Delta t}^{\pi/\Delta t} X(\omega) e^{j\omega n\Delta t} d\omega. \quad (2.14)$$

Комплексну функцію $X(\omega)$ можна виразити через її модуль і аргумент (як і довільне комплексне число):

$$X(\omega) = |X(\omega)| \cdot e^{j\arg(X(\omega))} \quad (2.15)$$

де $|X(\omega)| = \sqrt{\{\operatorname{Re}(X(\omega))\}^2 + \{\operatorname{Im}(X(\omega))\}^2}$ називають *амплітудним спектром*, а $\arg(X(\omega)) = -\arctg \frac{\operatorname{Im}(X(\omega))}{\operatorname{Re}(X(\omega))}$ – *фазовим спектром*.

Далі сформульовано важливі результати, які використовують при цифровому опрацюванні сигналів, – теорему про згортку та теорему (рівність) Парсеваля.

Теорема про згортку: Спектр згортки двох сигналів дорівнює добутку спектрів цих сигналів.

Тобто, якщо: $y(n) = x(n) \otimes g(n)$
то спектр сигналу $Y(\omega) = X(\omega) \cdot G(\omega)$. (2.16)

Спектр добутку сигналів дорівнює згортці спектрів цих сигналів, з точністю до нормуючого множника.

Тобто, якщо: $y(n) = x(n) \cdot g(n)$
то спектр сигналу $Y(\omega) = \frac{1}{2\pi} X(\omega) \otimes G(\omega)$. (2.17)

Теорема (рівність) Парсеваля.

Теорема встановлює зв'язок між енергією дискретного сигналу, обчисленою у часовій та у частотній областях. Щоб сформулювати цю теорему, потрібні такі поняття.

Енергія та потужність – це кількісні характеристики сигналу, які дозволяють оцінити вплив цього сигналу протягом визначеного часу або усереднено.

Якщо, $x(t)$ – довільний сигнал, то його *енергія* визначається:

- для неперервного випадку: $E = \int_0^{\infty} |x(t)|^2 dt$, зокрема, для дійсних сигналів

$$E = \int_0^{\infty} x(t)^2 dt;$$

- для дискретного випадку: $E = \sum_{n=0}^{\infty} |x(n\Delta t)|^2$.

Потужність сигналу для неперервного випадку визначається виразом:

Миттєва потужність: $p(t) = x^2(t)$

Середня потужність $P_{cp} = \frac{1}{T} \int_0^T x^2(t) dt$, тобто $P_{cp} = \frac{E}{T}$ (T – тривалість сигналу).

Енергія сигналу може бути скінченою або нескінченою. Так, довільний сигнал скінченної тривалості буде мати *скінчену* енергію (якщо він не містить

дельта-функцій або нескінчених гілок), а довільний періодичний сигнал навпаки має нескінчену енергію.

Якщо енергія сигналу нескінченна, то можна визначити середню потужність цього сигналу на всій часовій осі:

$$P_{cp} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-T/2}^{T/2} x^2(t) dt \quad (2.18)$$

Корінь квадратний з середньої потужності дає середньоквадратичне (діюче) значення сигналу:

$$\sigma_x = \sqrt{P_{cp}}. \quad (2.19)$$

Виходячи з наведених означень, формулюємо *теорему Парсеваля*:

$$E = \sum_{n=0}^{\infty} |x(n)|^2 = \frac{\Delta t}{2\pi} \int_{-\pi/\Delta t}^{\pi/\Delta t} |X(\omega)|^2 d\omega. \quad (2.20)$$

Якщо обробка сигналу проводиться в частотній області, ця рівність дозволяє обчислити енергію сигналу безпосередньо за його амплітудним спектром, не виконуючи обернене перетворення Фур'є.

Зв'язок між спектрами аналогового і дискретного сигналів. Співвідношення між спектрами аналогового і дискретного сигналів задається так:

$$X_d(\omega) = X(e^{j\omega\Delta t}) = \frac{1}{\Delta t} \sum_{m=-\infty}^{\infty} X_a \left[j \left(\omega + m \frac{2\pi}{\Delta t} \right) \right]. \quad (2.21)$$

З нього випливає, що спектр дискретного сигналу (з точністю до постійного множника $1/\Delta t$, тобто f_d) рівний сумі спектрів аналогового сигналу, зсунутих на частотній осі на всі можливі частоти, які кратні частоті дискретизації $m\omega_d = m \frac{2\pi}{\Delta t}$, де $m = 0, \pm 1, \pm 2, \dots$. Іншими словами, спектр дискретного сигналу – це нескінченна сума копій спектру аналогового сигналу, зсунутих один відносно одного на частоту дискретизації [22,23].

Вплив частоти дискретизації на співвідношення між спектрами аналогового і дискретного сигналів відображений на рис. 2.1. У цьому разі вважаємо, що спектр аналогового сигналу обмежений граничною частотою $\omega_c = 2\pi f_c$.

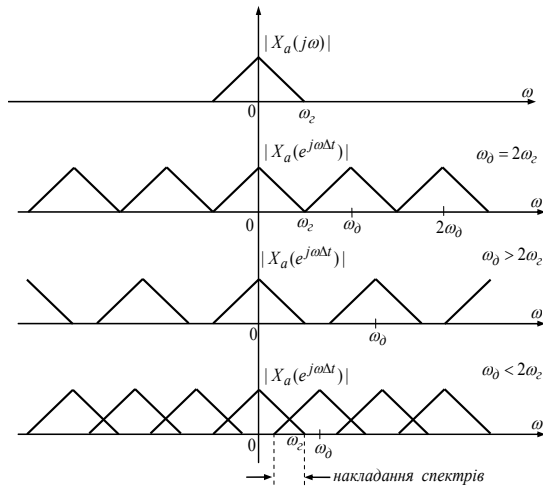


Рис. 2.1. Спектр дискретного сигналу для різної частоти дискретизації.

Наведені на рис. 2.1 графіки дозволяють зробити такі висновки:

1. якщо частота дискретизації $\omega_d \geq 2\omega_c$, то в основній смузі частот $[0; \omega_d/2]$, спектри аналогового і дискретного сигналів співпадають.

2. якщо частота дискретизації $\omega_d < 2\omega_c$, то відбувається накладання спектрів, тому в основній смузі частот $[0; \omega_d/2]$ спектр дискретного сигналу є спотвореним спектром аналогового сигналу.

Дискретизація вузькосмугових сигналів.

Вузькосмуговим (відносно вузькосмуговими) називають сигнал, ширина спектру ΔF якого значно менша, ніж його середня частота f_{cp} :

$$\frac{\Delta F}{f_{cp}} \ll 1.$$

Це завжди має місце в радіоканалах і в багатоканальних системах з частотним розділенням [2,23]. Наприклад, той самий телефонний сигнал можна трактувати як вузькосмуговий, при його передачі на несучій частоті 128 кГц з використанням амплітудної модуляції. Його спектр зосереджений в області від $f_{\min} = 128 + 0.3 = 128.3$ до $f_{\max} = 128 + 3.4 = 131.4$ кГц і має ширину

$$\Delta F = f_{\max} - f_{\min} = 3.1 \text{ кГц},$$

а середня частота $f_{cp} = (f_{\max} + f_{\min})/2 = (131.4 + 128.3)/2 = 129.85 \text{ кГц}$. Очевидно, що виконується умова

$$\Delta F / f_{cp} = 3.1/129.85 \approx 0.024 \ll 1.$$

Зауважимо, що іноді сигнал називають вузькосмуговим, якщо відношення максимальної частоти спектру до мінімальної не більше двох: $(f_{\max} / f_{\min}) \leq 2$, що не суперечить попередньому визначенню.

В подібних випадках частота дискретизації, вибрана безпосередньо за правилом $f_d \geq 2f_z$ буде надлишковою, особливо для радіосигналів, коли вона може становити сотні мегагерц, внаслідок чого обробка в реальному масштабі часу виявиться неможливою або через складність алгоритмів обробки, або внаслідок технічних можливостей сучасної елементної бази. Крім того, очевидно, що така висока частота дискретизації і не потрібна, оскільки інформація про сигнал міститься не в частоті f_{\max} , а в огинаючій, або фазі, які змінюються в часі повільно (при відносно низьких частотах модуляції). Тому у таких випадках використовується *узагальнена теорема Котельникова*:

$$\frac{2f_{\max}}{q} \leq f_d \leq \frac{2f_{\min}}{q-1} \quad (2.22)$$

де q – номер діапазону частот, який може приймати лише цілі значення:

$$q = 1, 2, \dots, \left\lfloor \frac{f_{\max}}{f_{\max} - f_{\min}} \right\rfloor$$

а $\lfloor x \rfloor$ – найближче менше від x ціле число. Тобто $q_{\max} \leq \frac{f_{\max}}{f_{\max} - f_{\min}}$.

Таким чином, значення частоти дискретизації f_d може вибиратися з ряду допустимих діапазонів. Зокрема, при $q=1$ маємо класичну теорему Котельникова.

Відомо [10,22,23], що зі збільшенням номеру діапазону q частота f_d зменшується і спектр сигналу, залишаючись незмінним, пропорційно зсувається вліво до нової основної смуги частот $[0, f_d/2]$. Ця обставина дозволяє вибрати таку f_d , при якій відстань Δf_{cp} між середніми частотами сусідніх копій спектру

дискретного сигналу буде максимальною і становитиме $f_d/2$. Це забезпечується лише в тому випадку, коли середня частота спектру сигналу виявиться в точці $f_{cp} = f_d/4$.

2.3. Швидке перетворення Фур'є

Дискретне перетворення Фур'є. За допомогою обчислювальної техніки, неперервна функція може бути визначеною лише в дискретних точках. Механізм (алгоритм, метод) обчислення спектру скінченної послідовності на періоді ω_d в дискретних точках називається дискретним перетворенням Фур'є (ДПФ).

ДПФ скінчених (фінітних) сигналів. Як вже було сказано раніше, періодичні сигнали (а в дискретному випадку – періодичні послідовності) – це лише математична модель, тобто, реально, їх не існує. На практиці, ми маємо справу з сигналами скінченної тривалості. Тому, нас цікавить як обчислити спектр скінченної послідовності у дискретних точках (дискретний спектр). Перш за все, слід вияснити скільки точок спектру слід брати, оскільки недостатня їх кількість приведе до втрати інформації, а надлишкова – до невиправданого зростання обсягів обчислень. Відповідь на це дає наведена далі теорема.

Теорема Котельникова в частотній області: неперервний спектр скінченного сигналу повністю визначається послідовністю своїх відліків, взятих з інтервалом $\Delta\omega$:

$$\Delta\omega \leq \frac{2\pi}{T_c} \quad \text{або} \quad \Delta f \leq \frac{1}{T_c}. \quad (2.23)$$

де T_c – тривалість сигналу.

Оскільки вхідний сигнал є дискретним, тобто визначений у N точках, $T_c = N\Delta t$. Тому отримуємо мінімально достатню кількість рівномірно віддалених відліків спектру на періоді ω_d :

$$\frac{\omega_d}{\Delta\omega} = \frac{2\pi}{\Delta t} \cdot \frac{2\pi}{T_c} = \frac{2\pi}{\Delta t} \cdot \frac{2\pi}{N\Delta t} = N.$$

Отже, на періоді ω_d неперервний спектр $X(\omega)$ скінченного сигналу, довжиною N , повністю визначається послідовністю своїх N рівномірно віддалених відліків $X(k\Delta\omega)$, $k = 0, 1, \dots, (N-1)$. Маючи ці відліки, можемо точно відтворити неперервний спектр. При цьому величина f_d / N називається роздільною здатністю за частотою.

Таким чином, дискретним перетворенням Фур'є називається пара взаємо однозначних перетворень:

- пряме дискретне перетворення Фур'є (використовують скорочення ДПФ)

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{nk} \quad k = 0, 1, \dots, (N-1) \quad (2.24)$$

- обернене дискретне перетворення Фур'є (ОДПФ)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-nk} \quad n = 0, 1, \dots, (N-1) \quad (2.25)$$

де

- $x(n)$, $n = 0, 1, \dots, (N-1)$ – послідовність відліків в часовій області (дійсна або комплексна);

- $X(k)$, $k = 0, 1, \dots, (N-1)$ – дискретні коефіцієнти Фур'є – один період послідовності в частотній області;

- k – номер відліку послідовності $X(k)$, що відповідає частоті $k\Delta\omega$;

- W_N^{nk} – повертаючий множник, для якого виконується рівність

$$W_N^{nk} = e^{-j \frac{2\pi}{N} nk}$$

Послідовності $x(n)$ та $X(k)$ називають N -точковими. Відліки послідовності $X(k)$ називають відліками ДПФ.

Порівнюючи ДПФ та ОДПФ для скінчених та періодичних послідовностей, бачимо, що вони співпадають, хоча трактування в них – різне. У випадку періодичної послідовності:

- N -точкова послідовність $x(n)$ - це один період даної послідовності;

- N -точкова послідовність $X(k)$ - це N комплексних гармонік періодичного спектру на періоді;

а у випадку скінченної послідовності:

- N -точкова послідовність $x(n)$ - це скінченна послідовність довжиною N відліків;

- N -точкова послідовність $X(k)$ - це N рівномірно віддалених відліків неперервного періодичного спектру $X(\omega)$ на періоді ω_d .

ДПФ як згортка сигналу з базисними функціями. Оскільки комплексна експонента може бути представлена у виді дійсної та уявної частини (формула Ейлера), то основне рівняння ДПФ може бути записане так:

$$X(k) = \sum_{n=0}^{N-1} x(n) [\cos(2\pi nk / N) - j \sin(2\pi nk / N)] = \sum_{n=0}^{N-1} x(n) \cos(2\pi nk / N) - j \sum_{n=0}^{N-1} x(n) \sin(2\pi nk / N)$$

$\text{Re}(X(k))$
 $\text{Im}(X(k))$

Таким чином, вихідний спектр ДПФ $X(k)$ є результатом обчислення згортки вхідних відліків сигналу (у часовій області), і набором з N пар гармонійних функцій (косинус і синус). Косинусоїди й синусоїди є базисними функціями даного перетворення. Випадок $N=8$ проілюстровано на рис. 2.2.

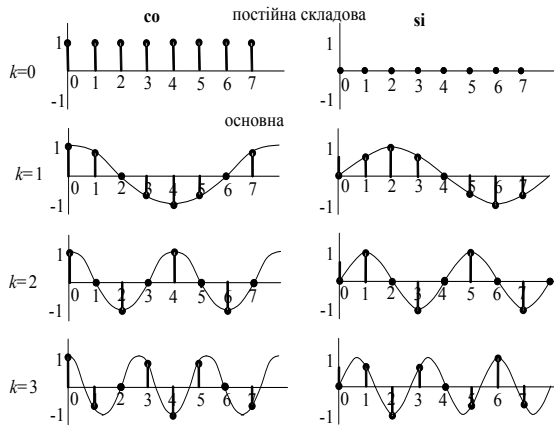


Рис. 2.2. Перші чотири базисні функції ДПФ для $N=8$.

Наведемо утворення дійсної частини перших чотирьох точок спектра (показані тільки косинусоїдальні гармонійні базисні функції). Подібна ж

процедура використовується для обчислення уявної частини спектра на основі синусоїдальних функцій.

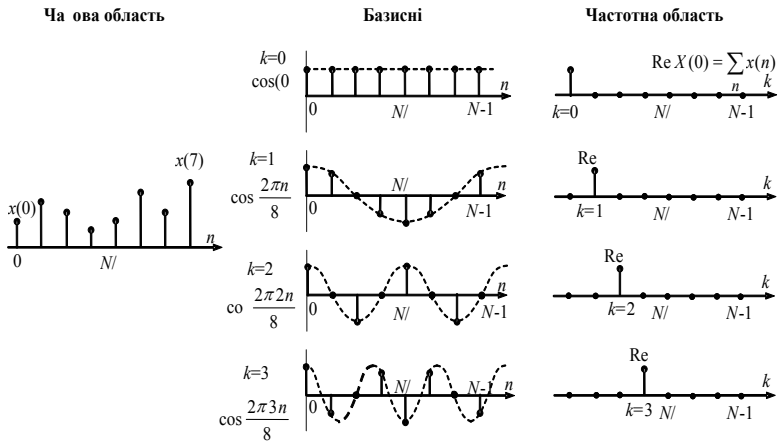


Рис. 2.3. Згортка відліків у часовій області з базисними функціями при ДПФ для $N=8$.

Перша точка $X(0)$ є простою сумою вхідних відліків у часовій області, оскільки $\cos(0)=1$. Зауважимо, що $X(0)$ – це середнє значення відліків у часовій області, або постійна складова (для уявної частини вона буде відсутня, оскільки $\sin(0)=0$). Друга точка $\text{Re}(1)$ отримана множенням кожного відліку з часової області на відповідне значення косинусоїди, що має один повний період на інтервалі N , з наступним підсумовуванням результатів. Третя точка $\text{Re}(2)$ отримана множенням кожного відліку з часової області на відповідну точку косинусоїди, що має два повних періоди на інтервалі N , з наступним підсумовуванням результатів. Аналогічно, четверта точка $\text{Re}(3)$ отримана множенням кожного відліку з часової області на відповідну точку косинусоїди із трьома повними періодами на інтервалі N і підсумовуванням результатів. Цей процес триває, поки не будуть обчислені всі N вихідних відліків. Подібна процедура, але з використанням синусоїд, застосовується для обчислення уявної частини частотного спектру.

Припустимо, що вхідний сигнал є косинусоїдальним, що має період N , тобто він містить один повний період у нашій вибірці. Також прийнемо його амплітуду й фазу ідентичними першої косинусоїдальній базисній функції $\cos(2\pi n/8)$. Вихідний спектр містить одну ненульову точку $\text{Re}(1)$, а всі інші точки $\text{Re}(k)$ є нульовими. Припустимо, що тепер вхідна косинусоїда зміщена вправо на 90° . Значення згортки між нею й відповідної базисною косинусоїдальною функцією дорівнює нулю. Але алгоритм перетворення передбачає обчислення згортки з базисною функцією $\sin(2\pi n/8)$, необхідне для одержання $\text{Im} X(1)$. Це показує, чому необхідно розраховувати і дійсну і уявну частини спектра для визначення амплітуди і фази частотного спектра. Зауважимо, що згортка синусоїдальної/косинусоїдальної функції будь-якої частоти, відмінної від частоти базової функції, дає нульове значення і для $\text{Re} X(k)$, і для $\text{Im} X(k)$.

Подібна процедура застосовується при обчисленні зворотного ДПФ для відновлення відліків у часовій області $x(n)$ із відліків у частотній області $X(k)$. Відповідне рівняння має такий вигляд:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) \cos(2\pi nk / N) + j \frac{1}{N} \sum_{k=0}^{N-1} X(k) \sin(2\pi nk / N).$$

$\text{Re}(x(n))$

$\text{Im}(x(n))$

Існує два основних типи ДПФ: дійсне ДПФ і комплексне ДПФ[24]. Попередні рівняння, описують комплексне ДПФ, де як вхідні, так і вихідні величини є комплексними числами. Оскільки на практиці, вхідні відліки в часовій області є дійсними і не мають уявної частини, уявна частина вхідних відліків завжди приймається рівною нулю. Вихід ДПФ $X(k)$ містить дійсну і уявну компоненти, які можуть бути перетворені в амплітуду й фазу. Дійсне ДПФ є спрощенням комплексного ДПФ. Вихідний спектр ДПФ може бути представлений або в полярній системі координат (амплітудою й фазою), або в алгебраїчній формі (дійсною і уявною частинами). Обидві зазначених форми перебувають у взаємно однозначній відповідності.

Швидким перетворення Фур'є (ШПФ) називають групу алгоритмів, які суттєво зменшують кількість арифметичних операцій при обчисленні дискретного

перетворення Фур'є [3,24]. Тобто, ці алгоритми мають менший порядок обчислювальної складності, в порівнянні з безпосереднім обчисленням.

Варто підкреслити, що ШПФ є точним, а не наближеним методом обрахунку коефіцієнтів ДПФ (а в більшості випадків, за рахунок зменшення кількості операцій він приводить до менших похибок заокруглення, тобто є навіть точнішим).

Оскільки алгоритмів багато, то як правило, в їх назвах є уточнення, який саме мається на увазі: ШПФ за основою 2, ШПФ за основою 4, ШПФ Винограда і т.д. Найширше використовується алгоритм ШПФ за основою 2, відомий як алгоритм Кулі-Тьюкі (вперше оприлюднений у 1965 році). Існують два, еквівалентні за ефективністю, алгоритми ШПФ за основою 2: з прорідженням за часом та з прорідженням за частотою.

ШПФ за основою 2 та прорідженням за часом (ШПФ2t).

1. Розбиття задачі на підзадачі. Розбиття (поділ) вхідної послідовності в цьому виді ШПФ здійснюється за принципом парності/непарності відліків. Тобто початкова сума розбивається на дві вдвічі меншої розмірності, перша з яких містить відліки з парними індексами, а друга – з непарними.

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn} = \sum_{n=0}^{N/2-1} x(2n) W_N^{k2n} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^{k(2n+1)}.$$

Оскільки $W_N^{k2n} = W_{N/2}^{kn}$; $\left(\begin{array}{l} -j2\pi nk \\ \text{бо: } e^{\frac{-j2\pi nk}{N}} = e^{\frac{-j2\pi nk}{N/2}} \end{array} \right)$

а $W_N^{k(2n+1)} = W_{N/2}^{kn} \cdot W_N^k$, $\left(\begin{array}{l} -j2\pi(2n+1)k \\ \text{бо: } e^{\frac{-j2\pi(2n+1)k}{N}} = e^{\frac{-j2\pi nk}{N}} + \frac{-j2\pi k}{N} = e^{\frac{-j2\pi nk}{N}} \cdot e^{\frac{-j2\pi k}{N}} = W_{N/2}^{kn} \cdot W_N^k \end{array} \right)$

можемо записати:

$$X(k) = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{kn} + \sum_{n=0}^{N/2-1} x(2n+1) W_N^k W_{N/2}^{kn} = \sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{kn} + W_N^k \sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{kn}$$

Очевидно, що справедливі співвідношення

$$\sum_{n=0}^{N/2-1} x(2n) W_{N/2}^{kn} = \text{ДПФ}_{N/2}\{x(2n)\} = X_0(k);$$

$$\sum_{n=0}^{N/2-1} x(2n+1) W_{N/2}^{kn} = \text{ДПФ}_{N/2}\{x(2n+1)\} = X_1(k).$$

Таким чином, отримуємо формулу розкладу алгоритму ШПФ2:

$$X(k) = X_0(k) + W_N^k X_1(k) \quad k = 0, 1, \dots, N-1. \quad (2.26)$$

Вона показує, як задачу розмірності N звести до двох підзадач розмірності $N/2$.

2. Рекурсія. Рекурсивно продовжуючи за формулою розкладу розбиття перетворення меншої розмірності, тобто $X_0(k)$ та $X_1(k)$ аж до отримання двоточкових перетворень, синтезуємо потрібний алгоритм ШПФ.

3. Виділення однакових виразів. Цей етап виконується для того, щоб однакові вирази обчислювати лише один раз. Тут беруться до уваги властивості періодичності симетричності повертаючих множників: періодичність: $W_N^{(k+N)n} = W_N^{kn}$ та симетричність $W_N^{k+N/2} = -W_N^k$.

Крім того, самі послідовності $X_0(k)$ та $X_1(k)$ є періодичні з періодом $N/2$ (оскільки вони є ДПФ для $N/2$ точок), тобто: $X_0(k+N/2) = X_0(k)$, $X_1(k+N/2) = X_1(k)$. Тому в загальній формулі розкладу для другої половини коефіцієнтів, тобто для $k+N/2$, будемо мати:

$$X(k+N/2) = X_0(k+N/2) + W_N^{k+N/2} X_1(k+N/2) = X_0 - W_N^k X_1(k).$$

Таким чином, маємо основну процедуру швидкого алгоритму:

$$\begin{aligned} X(k) &= X_0(k) + W_N^k X_1(k) \\ X(k+N/2) &= X_0(k) - W_N^k X_1(k) \end{aligned} \quad k = 0, 1, \dots, N/2-1.$$

Вона визначає правило обрахунку N -точкового ДПФ за результатами двох ДПФ розмірності $N/2$ для парних і непарних вхідних відліків.

Основною операцією алгоритму ШПФ, яку ще називають *базовою операцією* (БО) є одночасне (паралельне) обчислення двох відліків ДПФ – k -ого та $k+N/2$ -ого, за наведеною вище формулою. Граф БО (рис. 2.4) отримав назву “метелик”.

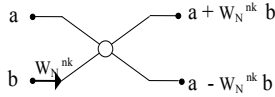


Рис.2.4. Базова операція алгоритму ШПФ з часовим прорідженням.

Використовуючи основну процедуру ШПФ і продовжуючи розбиття кожної з отриманих підпослідовностей, рекурсивно дійдемо до виконання двоточкового перетворення, яке здійснюється безпосередньо за базовою операцією. Випадок $N=8$ проілюстровано на рис. 2.5 та рис. 2.6.

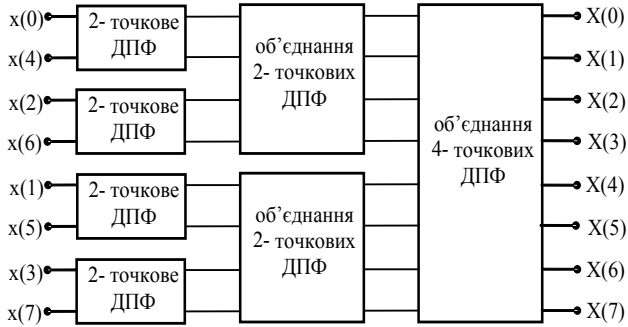


Рис. 2.5. Структура 8-точкового ШПФ2t.

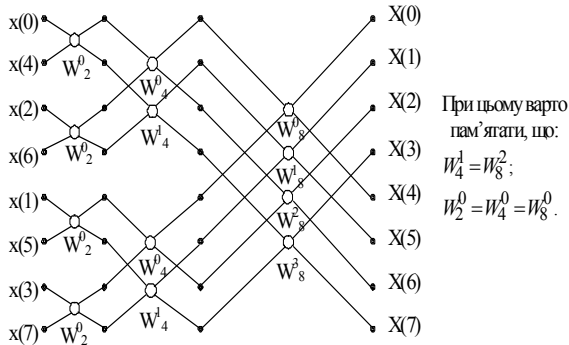


Рис.2.6. Граф алгоритму 8-точкового ШПФ2t.

Оцінка обчислювальної складності алгоритму ШПФ. Таким чином, N -точкове ШПФ $2t$ буде складатися з $\log_2 N$ етапів, на кожному з яких буде виконуватися $N/2$ БО. Тобто всього буде виконано $\log_2 N \cdot N/2$ БО. Враховуючи, що для виконання одного “метелика” треба виконати 1 операцію комплексного множення то 2 операції комплексного додавання, оцінимо складність такого алгоритму. $\mu_2 = 1; \alpha_2 = 2; N_{BO} = \frac{N}{2} \cdot \log_2 N$.

Обчислювальна складність N -точкового алгоритму ШПФ $2t$ така:

кількість операцій множення $\mu_N = \frac{N}{2} \cdot \log_2 N$,

кількість операцій додавання $\alpha_N = N \cdot \log_2 N$.

Реально, при $N=1024$ кількість операцій зменшується на два порядки: ДПФ $_{1024} = 2^{20}$; ШПФ $_{1024} = 2^{10} * 10 = 5 * 2^{11}$.

Порівнявши характеристики складності можна говорити, що побудований алгоритм є швидким, оскільки він зменшує характеристики складності на порядок: ДПФ $\rightarrow O(N^2)$, ШПФ $\rightarrow O(N \log_2 N)$.

ШПФ за основою 2 та прорідженням за частотою (ШПФ $2f$).

Побудова даного алгоритму відрізняється від попереднього принципом поділу вхідних послідовностей, порядком виконання БО та самою БО. В алгоритмі ШПФ $2f$ вхідні дані розділяються на першу та другу половини (від $x(0)$ до $x(N/2-1)$ та від $x(N/2)$ до $x(N-1)$), а етапи проходження будуть відповідати графу ШПФ $2t$, взятому справа наліво. При цьому вхідні відліки будуть йти в природному порядку, а вихідні – в біт-зворотньому порядку проходження номерів. Базова операція алгоритму ШПФ $2f$ показана рис. 2.7.

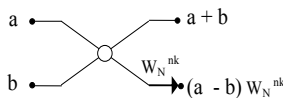


Рис.2.7. Базова операція алгоритму ШПФ з частотним прорідженням.

Для прикладу, алгоритм 8-точкового ШПФ $2f$ зображений на рис. 2.8.

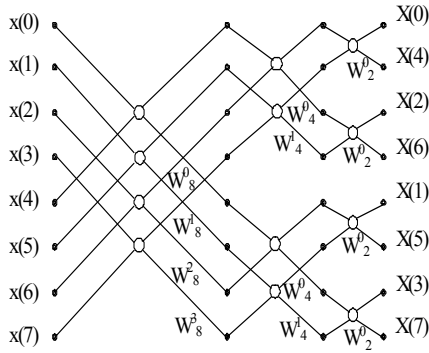


Рис.2.8. Граф алгоритму 8-точкового ШПФ2f.

Очевидно, що обчислювальна складність цього алгоритму така ж як і для ШПФ2f.

Крім ШПФ за основою 2, можна будувати ШПФ за основою 4, 8, при цьому довжини вхідних послідовностей мають бути степенями цих чисел. Алгоритми за більшою основою потребують менше операцій комплексного множення, але передбачають складнішу схему адресації та більше повертаючих множників [3,9,24].

2.4. Деякі інші ортогональні тригонометричні перетворення

Перетворення Фур'є має певні незручності: по-перше, для дійсних сигналів $x(t)$ значення $X(\omega)$ є комплексним, по-друге, ядро прямого $e^{-j\omega t}$ і оберненого $e^{j\omega t}$ перетворення мають різний вигляд, що ускладнює розробку засобів їх реалізації. Тому в практиці цифрової обробки сигналів, крім перетворення Фур'є, застосовуються також і інші дискретні ортогональні тригонометричні перетворення (ОТП), кожне з яких має свої переваги та недоліки [3,9,10,24].

Використання їх зумовлене тим, що вони враховують конкретні види вхідних сигналів, а також володіють необхідними властивостями, що дозволяють максимально спростити процес обчислень для переходу в спектральну область, а отже зекономити час та ресурси. До найпоширеніших ОТП можна віднести: дискретне перетворення Хартлі (ДПХ), дискретне косинус- і синус-перетворення

Фур'є (ДКПФ і ДСПФ), дискретне косинусне перетворення (ДКП), дискретне синусне перетворення (ДСП). Для кожного з них існують власні швидкі алгоритми обчислень, до того ж їх можна обраховувати за допомогою стандартного ШПФ.

Дискретне перетворення Хартлі.

$$H(k) = \sum_{n=0}^{N-1} x(n)(C_N^{kn} + S_N^{kn}) \quad - \text{аналіз}, \quad x(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k)(C_N^{kn} + S_N^{kn}) \quad - \text{синтез},$$

$$\text{де } n, k = 0, 1, \dots, N-1, \quad C_N^{kn} = \cos\left(\frac{2\pi kn}{N}\right); \quad S_N^{kn} = \sin\left(\frac{2\pi kn}{N}\right).$$

Фактично ми маємо : $W_N^r = C_N^r - jS_N^r$. Інакше кажучи, в перетворенні Хартлі явна одиниця опущена, а все решта – залишається.

Переваги перетворення Хартлі такі: не використовуються комплексні числа; пряме та обернене перетворення збігаються, що спрощує їх реалізацію. Але для більшості випадків (моделювання, дослідження електричних кіл) використовується ДПФ. Тому, з метою побудови і універсалізації виконання перетворень доцільно знати зв'язок між ДПФ і ДПХ. Наслідком існування такого переходу є можливість легкого представлення ДПХ через ДПФ.

Дискретні косинус- і синус- перетворення Фур'є.

Дані види перетворень існують для вхідних даних, що володіють властивістю симетричності. Ці перетворення, фактично є частинами ДПХ, проте виконуються як самостійні ОТП.

$$ДКПФ_N\{x(n)\} = H^c(k) = \sum_{n=0}^{N-1} x(n)C_N^{kn}, \quad ДСПФ_N\{x(n)\} = H^s(k) = \sum_{n=0}^{N-1} x(n)S_N^{kn}$$

$$\text{де } n, k = 0, 1, \dots, N-1, \quad C_N^{kn} = \cos\left(\frac{2\pi kn}{N}\right) \quad S_N^{kn} = \sin\left(\frac{2\pi kn}{N}\right).$$

Для забезпечення симетричності, вхідна послідовність розбивається на парну і непарну частини:

$$x(n) = x^c(n) + x^s(n), \quad n = 0, 1, \dots, N-1;$$

$$x^c(n) = x^c(N-n) = \frac{x(n) + x(N/n)}{2}, \quad n = 0, 1, \dots, N/2;$$

$$x^s(n) = -x^s(N-n) = \frac{x(n) - x(N/n)}{2}, \quad n=1, \dots, N/2-1, \quad x^s(0) = x^s(N/2) = 0.$$

Таким чином, маємо

$$H^c(k) = \text{ДКП}\Phi_N\{x^c(n)\}, \quad k, n=0, 1, \dots, N/2+1;$$

$$H^s(k) = \text{ДСП}\Phi_N\{x^s(n)\}, \quad k, n=1, \dots, N/2-1.$$

Ці перетворення є симетричними, тобто обернене ДКПФ і ДСПФ повністю збігається з прямим, з точністю до множника $1/N$:

$$x^c(n) = (1/N)\text{ДКП}\Phi_N\{H^c(k)\}; \quad x^s(n) = (1/N)\text{ДСП}\Phi_N\{H^s(k)\}.$$

Таким чином, перетворення комплексної послідовності подається у виді суми перетворень двох дійсних, кожне з яких передбачає у загальному випадку виділення парної і непарної частини:

$$\begin{aligned} x(n) &= \text{ОДП}\Phi_N\{X(k)\} = (1/N)\text{ДПХ}_N\{H(k)\} = x^c(n) + x^s(n) = \\ &= (1/N)\text{ДКП}\Phi_N\{H^c(k)\} + (1/N)\text{ДСП}\Phi_N\{H^s(k)\}, \quad n=1, \dots, N/2-1. \end{aligned}$$

Парна N -точкова послідовність повністю задається $N/2+1$ першими відліками, а непарна – $N/2-1$ відліками, починаючи з другого. Якщо ж вхідна послідовність відразу є повністю парною, або непарною, то достатньо виконати лише одне відповідне перетворення.

Дискретні косинусне і синусне перетворення.

Пряме та обернене ДКП послідовності дійсних чисел $x(n)$, описуються такими виразами:

$$L(k) = \sum_{n=0}^{N-1} x(n)C_N^{nk}, \quad x(n) = \sum_{k=0}^{N-1} P_k L(k)C_N^{nk},$$

де $C_N^{kn} = \cos\left(\frac{\pi(2n+1)k}{2N}\right)$ – елементи матриці ДКП; $n, k=0, 1, \dots, N-1$, $P_0 = 1/N$,

$$P_k = 1/N, k \neq 0.$$

Пряме та обернене ДСП послідовності дійсних чисел $x(n)$, описуються наведеними далі співвідношеннями:

$$Q(k) = \sum_{n=0}^{N-1} x(n)S_N^{nk}, \quad x(n) = \sum_{k=0}^{N-1} P_k Q(k)S_N^{nk},$$

де $S_N^{kn} = \sin\left(\frac{\pi(2n+1)k}{2N}\right)$ – елементи матриці ДСП; $n, k = 0, 1, \dots, N-1$, $P_0 = 1/N$,
 $P_k = 1/N, k \neq 0$.

Зв'язок між ДКП та ДСП впливає з властивостей тригонометричних функцій \sin та \cos і має такий вигляд:

$$Q(N-n) = \text{ДКП}_N\{(-1)^n x(n)\}.$$

Тобто, для обчислення ДСП достатньо замінити послідовність $x(n)$ на $x_1(n) = (-1)^n x(n)$, знайти ДКП послідовності $x_1(n)$, а далі переставити отримані елементи спектральної послідовності у зворотному порядку.

Треба зазначити, що на практиці найчастіше використовується ДКП, оскільки воно ефективно концентрує енергію сигналу в невеликій частині спектральних коефіцієнтів. Так, на базі ДКП побудовано відомий засіб стиску звукових сигналів MPEG, а також перші версії засобу стиску зображень JPEG (до JPEG-2000).

2.5. Завдання для самоперевірки

1. Лінійна дискретна згортка двох сигналів описується так:

$$\text{А) } y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k); \quad \text{Б) } y(n) = \sum_{k=-\infty}^{\infty} x(n) \cdot h(n-k) \quad \text{В) } y(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot x(n-k);$$

$$\text{Г) } y(n) = \sum_{k=-\infty}^{\infty} h(k) \cdot h(n-k); \quad \text{Д) немає правильної відповіді.}$$

2. Пряме перетворення Фур'є дозволяє:

А) подати сигнали в часовій області; Б) подати сигнали в частотній області;
 В) описати сигнали в полярних координатах; Г) знайти різницю двох сигналів;
 Д) немає правильної відповіді.

3. Чому дорівнює спектр добутку сигналів ?

А) добутку спектрів цих сигналів; Б) сумі спектрів цих сигналів; В) різниці спектрів цих сигналів;
 Г) згортці спектрів цих сигналів; Д) немає правильної відповіді.

4. Швидке перетворення Фур'є це:

А) метод отримання кореляційних характеристик сигналу; Б) спосіб обчислення похибки оцифрування сигналів; В) спосіб отримання передаточної характеристики лінійної дискретної системи; Г) група алгоритмів, які зменшують кількість арифметичних операцій при обрахунку дискретного перетворення Фур'є; Д) немає правильної відповіді

5. Роздільною здатністю за частотою називають:

А) відношення граничної частоти сигналу до частоти дискретизації; Б) відношення кількості точок ШПФ до подвоєної граничної частоти сигналу; В) відношення частоти дискретизації до кількості відліків; Г) добуток частоти дискретизації та кількості відліків; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1). А; 2) Б; 3) Г; 4) Г; 5) В.

РОЗДІЛ 3. МОДУЛЯЦІЯ ТА ДЕМОДУЛЯЦІЯ СИГНАЛІВ

3.1. Класифікація видів модуляції

Модуляція (лат. *modulatio* – мірність, розмірність) – процес зміни одного або декількох параметрів високочастотного модульованого коливання за законом інформаційного низькочастотного повідомлення (сигналу) [1]. У результаті спектр керуючого сигналу переноситься в область високих частот, оскільки для ефективної трансляції в ефір необхідно щоб усі приймально-передавальні станції працювали на різних частотах і “не заважали” один одному. Це процес “завантаження” інформаційного коливання на апріорно відому несучу. Інформація, яку слід передати, закладена в керуючому (модулюючому) сигналі. Роль носія інформації виконує високочастотне (несуче) коливання. Як несучі можуть бути використані коливання різної форми (прямокутні, трикутні і т.д.), однак найчастіше застосовують гармонійні коливання.

Класифікація видів модуляції [1].

За видом інформаційного сигналу:

- аналогова (неперервна) модуляція (аналоговий сигнал);
- дискретна модуляція (дискретний сигнал);
- маніпуляція (0-1);

За видом переносника (або несучої частоти):

- гармонійна (синусоїдальний сигнал);
- імпульсна (прямокутний періодичний імпульс);
- цифрова.

За видом параметрів несучої частоти, які зазнають зміни під дією інформаційного сигналу:

- амплітудна модуляція;
- частотна модуляція;
- фазова модуляція;

- широтна модуляція;
- широтно-імпульсна модуляція.

3.2. Основні характеристики видів модуляції

Енергетична ефективність (потенційна завадостійкість) характеризує достовірність переданих даних при впливі на сигнал адитивного білого гауссівського шуму, за умови, що послідовність символів відновлена ідеальним демодулятором. Визначається мінімальним співвідношенням сигнал / шум, яке необхідне для передачі даних через канал з ймовірністю помилки, що не перевищує задану. Енергетична ефективність визначає мінімальну потужність передавача, необхідну для достовірної роботи. Характеристикою методу модуляції є крива енергетичної ефективності – залежність ймовірності помилки ідеального демодулятора від співвідношення сигнал / шум.

Спектральна ефективність – співвідношення швидкості передачі даних до використаної смуги частот пропускання радіоканалу.

Приклади: AMPS: 0,83 NMT: 0,46 GSM: 1,35

Стійкість до впливів каналу передачі характеризує достовірність переданих даних при впливі на сигнал специфічних викривлень: завмирання внаслідок багатопроменевого поширення, обмеження смуги, зосереджені по частоті або часу завади, ефект Доплера та ін.

Лінійність підсилювачів. Для посилення сигналів з деякими видами модуляції можуть бути використані нелінійні підсилювачі, що дозволяє істотно знизити енергоспоживання передавача, при цьому рівень позасмугового випромінювання не перевищує допустимих меж.

Складність *реалізації* – визначається обчислювальним ресурсом, необхідним для реалізації алгоритму демодуляції, і вимогами до характеристик аналогової частини.

При *гармонійній модуляції* несучим є гармонійне коливання. Розрізняють *амплітудну* (АМ), *частотну* (ЧМ) і *фазову* (ФМ) модуляції.

При *імпульсній модуляції* як несучу використовують послідовність імпульсів. При *неперервній модуляції* параметр носія який модулюється під впливом повідомлення, що передається, може приймати довільне значення в деякому неперервному інтервалі своїх значень, а при *дискретній* – скінченна кількість значень із деякого інтервалу своїх значень.

До складних видів імпульсної модуляції відноситься *дельта-модуляція* (ДМ) і *імпульсно-кодова модуляція* (ІКМ). Окремим випадком модуляції є *маніпуляція сигналів*, при якій в якості модулюючого (інформаційного) сигналу використовується послідовність одно- або двополярних прямокутних імпульсів. Використовують її в системах передачі дискретної інформації.

Модуляцію носія (неперервну чи дискретну) здійснюють за допомогою перетворювача, що називається *модулятором*. Процес відновлення впливу, здійсненого при модуляції на носій за значеннями одного або декількох його параметрів, називається *демодуляцією*, а перетворювач, за допомогою якого здійснюється процес демодуляції, – *демодулятором*. Пристій, який одночасно виконує обидві операції, називається модемом (*модулятор-демодулятор*).

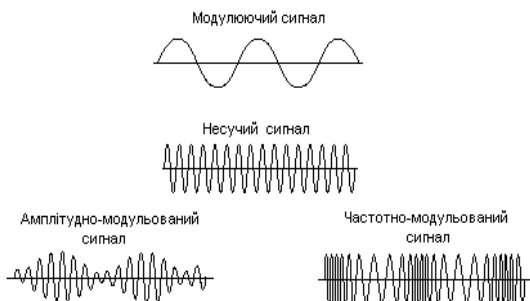


Рис. 3.1. Графіки амплітудної та частотної модуляція сигналів.

Розрізняють амплітудну модуляцію, при якій в залежності від миттєвої величини інформаційного сигналу змінюється амплітуда вихідного сигналу, і частотну модуляцію, при якій частота вихідного сигналу дещо змінюється в

порівнянні з частотою сигналу-носія, пропорційно миттєвій величині інформаційного сигналу.

Графіки, які ілюструють модуляцію сигналів, наведені на рис. 3.1.

Модуляція застосовується для узгодження інформаційного сигналу з параметрами лінії зв'язку.

Повідомлення низької частоти можна передавати і безпосередньо (без використання переносника високої частоти, тобто без модуляції). Проте модуляція розширює можливості передачі повідомлень з таких причин:

- збільшується кількість повідомлень, які можуть передаватись по одній лінії зв'язку шляхом використання частотного розділення сигналів і під несучих частот;

- підвищується достовірність сигналів, що передаються, при використанні завадостійких типів модуляції;

- підвищується ефективність випромінювання сигналу при передачі по радіоканалу.

Передачу інформації по лінії (каналу) зв'язку здійснюють за допомогою модуляції певних параметрів носія. Як носій можуть застосовуватись гармонійні сигнали і послідовність імпульсів. Вид і назву модуляції визначають в залежності від параметра, що модулюється, та форми носія.

Амплітудною модуляцією (АМ) називають утворення сигналу шляхом зміни амплітуди гармонійного коливання (несучої) пропорційно миттєвим значенням напруги чи струму іншого, більш низькочастотного сигналу (повідомлення).

Несуче коливання гармонійних видів модуляції можна представити як

$$u = U_0 \cos(\omega_0 t + \varphi_0), \quad (3.1)$$

де U_0 , ω_0 і φ_0 – відповідно постійні амплітуда, кутова частота і початкова фаза гармонічного коливання.

Модулюючий (інформаційний) сигнал $f(t)$ при амплітудній модуляції впливає на постійну амплітуду коливань несучої частоти U_0 , змінюючи її пропорційно величині модулюючого сигналу:

$$U_0 + \Delta U \cdot f(t), \quad (3.2)$$

де ΔU – найбільше відхилення амплітуди АМ-коливання. Тоді АМ-коливання можна записати у вигляді

$$U_{AM} = [U_0 + \Delta U \cdot f(t)] \cos \omega_0 t = U_0 \left[1 + \frac{\Delta U}{U_0} f(t) \right] \cos \omega_0 t, \quad (3.3)$$

де відношення $\frac{\Delta U}{U_0} = m_a$ називається коефіцієнтом амплітудної модуляції. Щоб

уникнути явища перемодуляції, при якому на виході модулятора різко розширюється спектр модульованого сигналу, коефіцієнт m_a вибирається меншим одиниці. З урахуванням цього, вираз (3.3) виглядатиме так:

$$U_{AM} = U_0 [1 + m_a f(t)] \cos \omega_0 t. \quad (3.4)$$

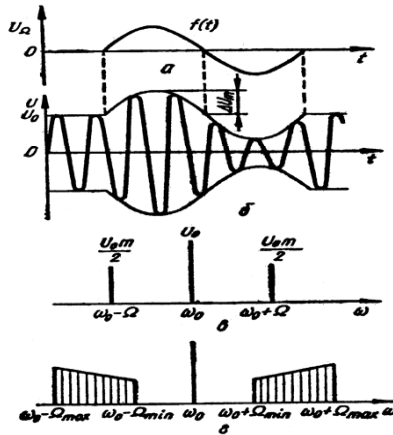


Рис. 3.2. Графіки модулюючого (а) і модульованого (б) сигналів;
спектри амплітуд АМ-сигналу
для простого (в) і складного (г) модулюючих сигналів.

Якщо інформаційний сигнал представляє собою гармонійне коливання однієї частоти з одиничною амплітудою $f(t) = \cos \Omega t$ (рис. 3.2, а), то, з урахуванням (3.4), отримаємо:

$$U_{AM} = U_0 [1 + m_a \cos \Omega t] \cos \omega_0 t = U_0 \cos \omega_0 t + \frac{m_a U_0}{2} \cos(\omega_0 + \Omega)t + \frac{m_a U_0}{2} \cos(\omega_0 - \Omega)t, \quad (3.5)$$

де перший доданок описує немодульоване несуче коливання, другий і третій – доданки з частотами $\omega_0 + \Omega$ і $\omega_0 - \Omega$ – верхню і нижню бокові частоти. Графік амплітудно модульованого коливання наведений на рис. 3.2, б. Спектр АМ-сигналу для розглянутого випадку містить несучу і дві бокові частоти (див рис. 3.2, в). Ширина спектру при цьому рівна 2Ω .

При модуляції несучої частоти складним сигналом, що має широкий спектр частот, АМ-коливання буде містити верхню і нижню бокові полоси частот (див. рис. 3.2, з). При цьому ширина спектра визначається значенням подвоєної максимальної частоти спектру модулюючого сигналу:

$$\Delta\omega_{AM} = 2\Omega_{\max}.$$

Так як бокові смуги частот є дзеркальним відображенням один одного відносно несучої частоти і несуть одну і ту ж інформацію про інформаційний сигнал, то на практиці для зменшення смуги, що займає АМ-сигнал, підвищення завадостійкості та більш ефективного використання лінії зв'язку передачу, як правило, здійснюють на одній боковій смузі, зрізаючи одну з бокових полос відповідним фільтром. Така передача отримала назву *передачі на одній боковій смузі* і застосовується в багатьох системах передачі інформації. У цьому разі смуга частот, що передаються, скорочується не менш, ніж в два рази. Переваги вказаного способу передачі полягають в такому:

- у два рази збільшується кількість каналів зв'язку;
- у чотири рази збільшується завадостійкість;
- при прийомі до двох разів зменшується потужність завади.

Частотна модуляція.

У цьому разі як несуче коливання використовуємо таке $u = U_0 \cos(\omega_0 t + \varphi_0)$.

При частотній модуляції модулюючий сигнал $f(t) = \cos \Omega t$ впливає на частоту ω_0 несучого коливання, яка змінюється в часі відносно свого центрального значення за законом інформаційного сигналу $f(t)$:

$$\omega = \omega_0 + \Delta\omega \cdot f(t), \quad (3.6)$$

де $\Delta\omega$ – девіація частоти, тобто відхилення кутової частоти від центрального значення ω_0 . З врахуванням (3.6) отримуємо:

$$U = U_0 \cos((\omega_0 + \Delta\omega \cdot f(t))t + \varphi_0) \quad (3.7)$$

Відношення $\frac{\Delta\omega}{\Omega} = m_f$ називається *індексом (коефіцієнтом) частотної модуляції*. Так як до виразу для модульованого сигналу входить постійна частота, то амплітуду коливання носія можна описати виразом

$$U = U_m \cos \theta, \quad (3.8)$$

де θ – фаза коливання, пов'язана з частотою співвідношенням $\omega = \frac{d\theta}{dt}$.

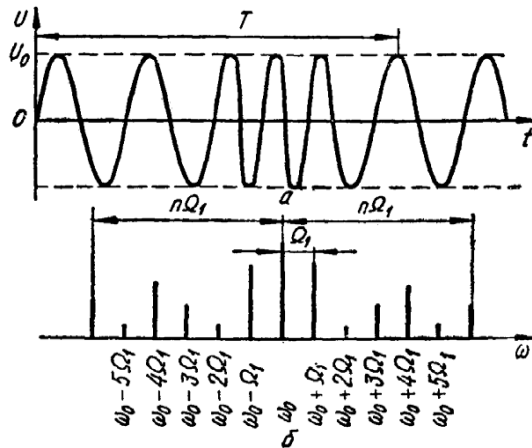


Рис.3.3. Частотно-модульований сигнал (а) і його спектр (б).

Вигляд ЧМ-сигналу з постійною амплітудою наведений на рис. 3.3, а. При ЧМ фаза коливання відповідає такому виразу:

$$\theta = \int_0^t \omega dt = \int_0^t [\omega_0 + \Delta\omega \cdot f(t)] dt = \omega_0 t + \Delta\omega \int_0^t f(t) dt. \quad (3.9)$$

Підставивши вираз (3.9) в (3.8), отримаємо

$$U_{\text{ЧМ}} = U_m \cos \left[\omega_0 t + \Delta \omega \int_0^t f(t) dt \right]. \quad (3.10)$$

Якщо модулююче коливання є інформаційним сигналом одної частоти Ω , то спектр ЧМ-сигналу (рис. 3.3,б) має дві бокові смуги, які містять нескінченну послідовність гармонічних коливань, віддалених один від одного на Ω і спадних за амплітудою при віддаленні від несучої частоти.

Приблизно ширину спектру ЧМ-сигналу можна визначити з виразу

$$\Delta \omega_{\text{ЧМ}} \approx 2m_f \Omega.$$

Видно, що чим більший індекс частотної модуляції m_f , тим вужчий, практично, необхідний спектр ЧМ-сигналу.

В залежності від вибраного індексу модуляції m_f розрізняють *вузькосмугову частотну модуляцію* з малими індексами m_f і *широкосмугову* – з великими індексами m_f . При вузькосмуговій ЧМ ширина спектра наближається до АМ, а при широкосмуговій – набагато більше, ніж при АМ.

Основною перевагою широкосмугової ЧМ є висока завадостійкість, значно більша ніж при АМ, так як частота сигналу менше зазнає впливу завад, ніж амплітуда.

Демодуляція АМ-сигналу може здійснюватися декількома методами. Найпростіший метод – симулювати роботу аналогового двопівперіодного детектора. Обчислюємо модуль вхідного АМ-сигналу, а потім згладжуємо отримані однополярні косинусоїдальні імпульси, пропускаючи їх через ФНЧ. Проте даний метод, не буде працювати правильно у випадку перемодуляції. Інший метод – синхронне детектування, суть якого полягає в домноженні частоти сигналу на опорне коливання з несучою частотою:

$$y(t) = S_{\text{АМ}}(t) \cos(\omega_0 t + \varphi_0) = A(t) \cos^2(\omega_0 t + \varphi_0) = \frac{1}{2} A(t) + \frac{1}{2} A(t) \cos 2(\omega_0 t + \varphi_0) \quad (3.11)$$

Результат перемноження містить дві складові. Перша складова – це вагова амплітудна функція, а друга – АМ сигнал з несучою частотою $2\omega_0$. Цей

високочастотний сигнал можна легко видалити шляхом пропускання сигналу через ФНЧ.

3.3. Обчислення параметрів модульованих сигналів

Для обчислення параметрів модульованих сигналів використовують дані, які наведені далі.

$$\text{Вхідний сигнал, } \sum_{n=1}^4 \sin\left(\frac{2\pi\omega_n t}{F_s}\right), t=0..1000$$

Частоти вхідного сигналу, ω_n , Гц: 0,06; 1,65; 4,85; 8,3

Тип функції несучого коливання: $U_0 \cdot \cos(\omega_0 t / F_s + \varphi_0)$

Частота несучого коливання, ω_0 , Гц: 40π

Амплітуда несучого коливання, U_0 : 31

Фаза несучого коливання, φ_0 : $\pi/31$

Частота дискретизації, F_s , Гц: 875

Коефіцієнт модуляції, m_a : 0,5

Спочатку формуємо модулюючий сигнал:

$$y = \sum_{n=1}^4 \sin\left(\frac{2\pi\omega_n t}{F_s}\right) = \sin(0,06 \cdot 2\pi \cdot t / 40\pi) + \sin(1,65 \cdot 2\pi \cdot t / 40\pi) + \\ + \sin(4,85 \cdot 2\pi \cdot t / 40\pi) + \sin(8,3 \cdot 2\pi \cdot t / 40\pi)$$

та несучий сигнал:

$$x = U_0 \cdot \cos(\omega_0 t / F_s + \varphi_0) = 31 \cdot \cos(2 \cdot 40 \cdot \pi \cdot t / 40\pi + \pi / 31).$$

Далі знаходимо АМ-сигнал:

$$U_{AM}(t) = U_0 [1 + m_a f(t)] \cos \omega_0 t = 31 \cdot [1 + 0,5y(t)] \cos(2 \cdot 40 \cdot \pi \cdot t / 40\pi + \pi / 31).$$

Після цього обчислюємо перетворення Фур'є для сигналу $U_{AM}(t)$. Тоді виконуємо демодуляцію сигналу $U_{AM}(t)$ згідно з виразом (3.11):

$$\begin{aligned} y_{DM}(t) &= U_{AM}(t) \cos(2 \cdot 40 \cdot \pi \cdot t + \pi/31) = y(t) \cos^2(2 \cdot 40 \cdot \pi \cdot t + \pi/31) = \\ &= \frac{1}{2} y(t) + \frac{1}{2} y(t) \cos(2 \cdot 80 \cdot \pi \cdot t + 2\pi/31) \end{aligned}$$

На завершальному кроці виконуємо фільтрацію сигналу $y_{DM}(t)$. Для цього, зокрема, можна скористатися відповідними функціями середовища SCILAB.

3.4. Завдання для самоперевірки

1. Модуляція це:

А) метод апроксимації сигналу; Б) частотне представлення сигналу; В) метод квантування аналогового сигналу; Г) процес зміни параметрів високочастотного модульованого коливання за законом інформаційного низькочастотного повідомлення; Д) немає правильної відповіді.

2. В якості несучого сигналу може виступати:

А) гармонійне коливання; Б) пилоподібний сигнал; В) послідовність прямокутних імпульсів; Г) послідовність одиничних імпульсів; Д) немає правильної відповіді.

3. При амплітудній модуляції інформаційний сигнал впливає на:

А) постійну амплітуду коливань несучої частоти; Б) частоту несучого коливання; В) фазу несучого коливання; Г) відхилення кутової частоти несучого коливання; Д) немає правильної відповіді

4. При частотній модуляції інформаційний сигнал впливає на:

А) постійну амплітуду коливань несучої частоти; Б) частоту несучого коливання; В) фазу несучого коливання; Г) відхилення кутової частоти несучого коливання; Д) немає правильної відповіді

5. При фазовій модуляції інформаційний сигнал впливає на:

А) постійну амплітуду коливань несучої частоти; Б) частоту несучого коливання; В) фазу несучого коливання; Г) відхилення кутової частоти несучого коливання; Д) немає правильної відповіді

Правильні відповіді до завдань:

1). Г; 2) А,В; 3) А; 4) Б; 5) В.

РОЗДІЛ 4. ЦИФРОВА ОБРОБКА СИГНАЛІВ ПРИ ДЕКІЛЬКОХ ШВИДКОСТЯХ.

4.1. Області застосування методів обробки при декількох швидкостях

Зростаюча потреба в сучасних системах обробки з більш ніж однією частотою дискретизації зумовила появу нової підобласті ЦОС – обробка при декількох швидкостях (multirate processing) [1]. У цьому разі використовують дві основні операції: децимація й інтерполяція, що дозволяють ефективно чергувати швидкості передачі даних. Децимація зменшує частоту дискретизації, ефективно стискаючи дані, залишаючи тільки необхідну інформацію. Інтерполяція, навпаки, збільшує частоту дискретизації. Часте перетворення даних під нову частоту передачі полегшує їхню обробку (наприклад, робить обчислювально більш ефективною) або забезпечує сумісність із іншою системою. Приклад 1: Якщо зменшити частоту дискретизації сигналу з 100 до 10 КГц без втрати бажаної інформації, то витрати обчислювальних ресурсів при наступній обробці сигналу зменшаться в 10 разів. Приклад 2: Нехай потрібно відтворити звуковий компакт-диск, що має частоту дискретизації 44,1 КГц, у студії, яка підтримує частоту 48 КГц. Тоді інформацію з даного компакт-диска потрібно спочатку перетворити з частотою 48 КГц, використовуючи методи обробки при декількох швидкостях.

Області застосування обробки при декількох швидкостях є, зокрема, такими [1]: обробка аудіосигналів – АЦП/ЦАП з вибіркою із запасом (oversampling), відтворення компакт-дисків і збір даних; цифровий зв'язок – трансмультиплексори, приймачі; біомедицина – вузькосмугова фільтрація електро-кардіограм й енцефалограм плоду.

4.2. Зміна частоти дискретизації

Зменшення частоти дискретизації: децимація із цілим кроком. На рис. 4.1(а) наведена блок-схема децимації сигналу $x(n)$ із цілим кроком M .

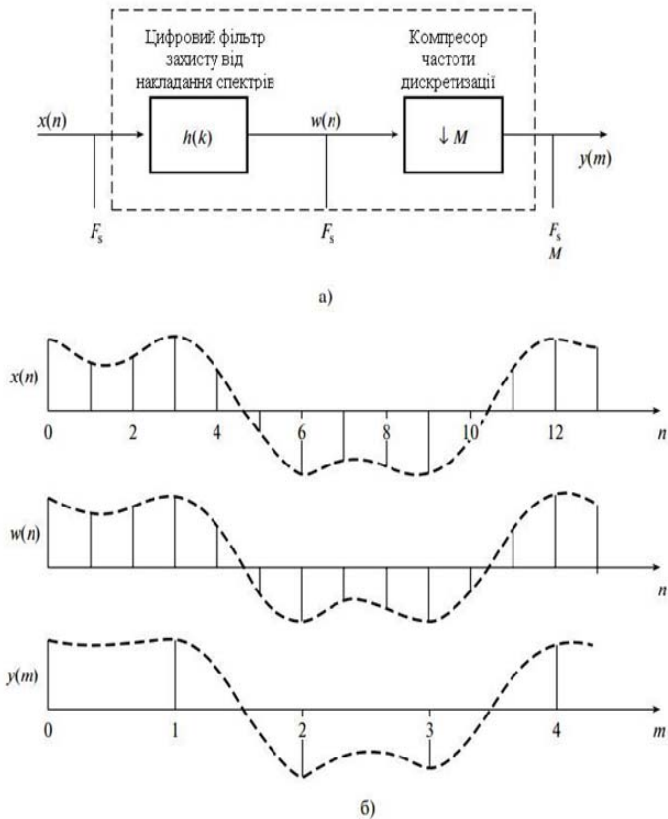


Рис. 4.1. Блок-схема децимації з кроком M (а). Графіки процесу децимації при $M = 3$ (б).

На рис.4.1(а) наведені цифровий фільтр захисту від накладення спектрів $h(k)$ і схема стиску (компресор) частоти дискретизації, позначена стрілкою, що спрямована вниз, з коефіцієнтом децимації M . Компресор частоти дискретизації знижує частоту дискретизації з F_s до F_s/M . Для запобігання накладання при більш низькій частоті вибірки використовується цифровий фільтр, що попередньо обмежує смугу вхідного сигналу до $F_s/2M$. Таким чином, сигнал $x(n)$ спочатку обмежується по смузі. Зниження частоти дискретизації досягається за рахунок того, що з кожних M вибірок сигналу $w(n)$, що фільтрується, відкидається $M-1$ вибірка. Вхід і вихід процесу децимації зв'язані наступним співвідношенням:

$$y(m) = w(mM) = \sum_{k=-\infty}^{\infty} h(k)x(mM - k), \quad (4.1)$$

де $w(n) = \sum_{k=-\infty}^{\infty} h(k)x(n - k)$.

На рис. 4.1(б) проілюстровано процес децимації для випадку $M=3$, де з кожних трьох вибірок $x(n)$ відкидаються дві. Відзначимо, що, по суті, децимація – це операція стиску даних. Спектральне подання процесу децимації наведено на рис. 4.2(а), де передбачається, що на вхід подається широкосмуговий сигнал $x(n)$. На рис. 4.2(б) пунктиром зображені дзеркальні компоненти сигналу, які призвели б до накладання, якби сигнал $x(n)$ не був обмежений по смузі перед децимацією.

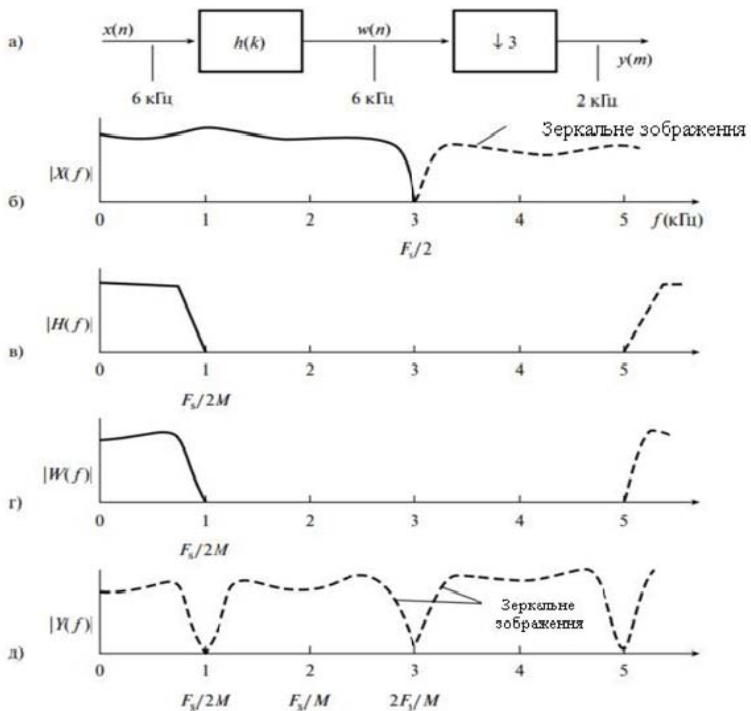


Рис. 4.2. Спектральна інтерпретація децимації сигналу із частоти 6 кГц до частоти 2 кГц.

Збільшення частоти дискретизації: інтерполяція із цілим кроком.

Інтерполяція – це цифровий еквівалент процесу цифро-аналогового перетворення, коли із цифрових вибірок, поданих на цифро-аналоговий перетворювач, за допомогою інтерполяції відновлюється аналоговий сигнал. При цифровій інтерполяції, втім, процес породжує специфічні значення.

Для даного сигналу $x(n)$ із частотою дискретизації F_s процес інтерполяції збільшує частоту дискретизації в L раз, тобто до LF_s . Блок-схема інтерполятора наведена на рис. 4.3 (а), графіки процесу інтерполяції із кроком $L=3$ у часовій області наведені на рис. 4.3 (б-г). До складу інтерполятора входить експандер частоти дискретизації, позначений стрілкою, яка спрямована нагору, з коефіцієнтом інтерполяції L , що показує, у скільки разів збільшується частота дискретизації.

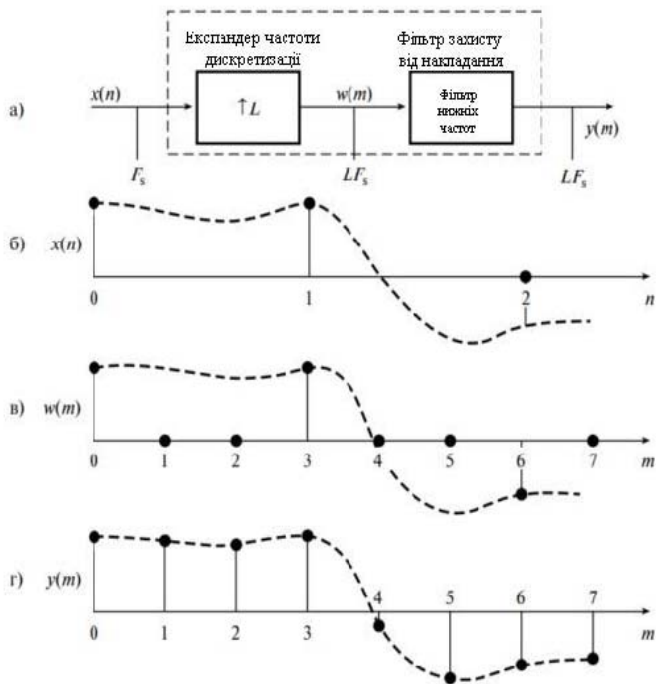


Рис. 4.3. Блок-схема інтерполятора та графіки інтерполяції з кроком $L=3$ у часовій області.

Для кожної вибірки сигналу $x(n)$ експандер вводить $(L-1)$ нульову вибірку,

у результаті чого формується новий сигнал $w(m)$ з частотою дискретизації LF_s . Далі цей сигнал пропускається через фільтр нижніх частот для видалення дзеркальних частот, введених при збільшенні частоти, і на виході формується сигнал $y(m)$. Введення $(L-1)$ нулів приводить до поширення енергії кожної вибірки сигналу по L вихідних вибірках, тобто кожна вибірка послабляється в L разів. Для компенсації цього ефекту можна, наприклад, помножити кожен вибірку $y(n)$ на L . Процес інтерполяції характеризується наступним зв'язком входу і виходу:

$$y(m) = \sum_{k=-\infty}^{\infty} h(k)w(m-k), \quad (4.2)$$

де

$$w(n) = \begin{cases} x(m/L, m=0, \pm L, \pm 2L, \dots \\ 0 \end{cases}$$

Процес інтерполяції в часовій області ілюструється на рис. 4.3 (б-г) для простого випадку $L=3$. У цьому разі кожна вибірка $x(n)$ породжує три вихідні вибірки (дві нульові вибірки вводять експандер).

Інтерпретація даного процесу в частотній області наведена на рис. 4.4. Функції $X(f)$, $W(f)$ і $Y(f)$ є частотними характеристиками сигналів $x(n)$, $w(m)$ і $y(m)$ відповідно. $H(f)$ – амплітудна характеристика фільтра придушення дзеркальних частот. Даний фільтр необхідний для видалення дзеркальних компонентів, позначених в $W(f)$ пунктиром. На даному етапі варто відзначити, що процеси децимації й інтерполяції дуальні, тобто взаємно обернені. Дана властивість дуальності означає, що інтерполятор можна легко одержати з еквівалентного дециматора і навпаки.

4.3. Перетворення частоти дискретизації з нецілим кроком

У деяких застосуваннях потрібно змінити частоту дискретизації в неціле число раз. Приклад: цифрове аудіо, де може вимагатися передача даних з одного запам'ятовуючого пристрою на інший; причому системи підтримують різні

частоти дискретизації (можливо, для запобігання незаконного копіювання матеріалів). Зокрема, це передача даних із системи відтворення компакт-дисків (44,1 КГц) на цифрову аудіоплівку (digital audio tape, DAT) (48 КГц). Для виконання необхідного перетворення частоту дискретизації компакт-дисуку варто збільшити в 48/44,1 рази.

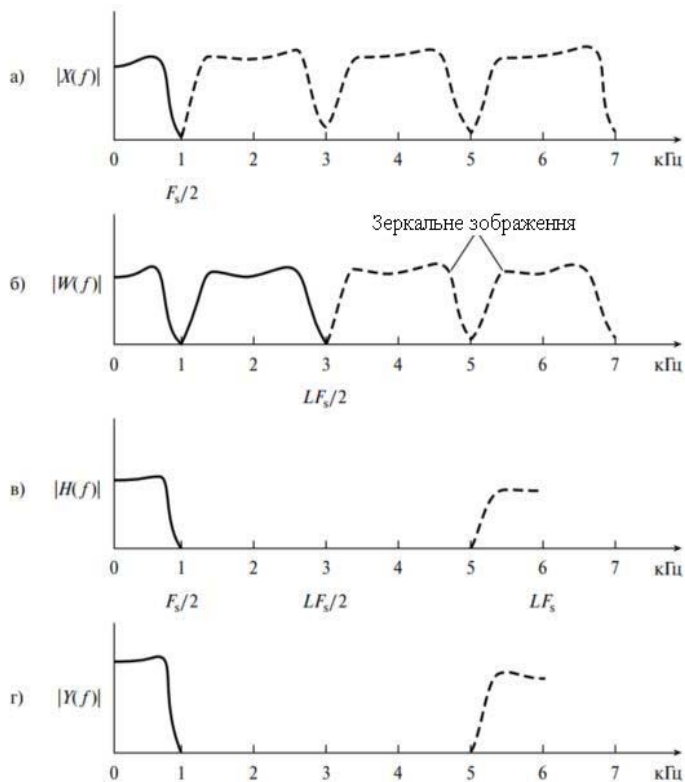


Рис. 4.4. Спектральна інтерпретація інтерполяції сигналу з 2 до 6 кГц.

На практиці подібні нецілі множники представляються раціональним числом, тобто відношенням двох цілих чисел, скажемо L і M , таких що L/M максимально близьке до бажаного множника. На рис. 4.5 наведені графіки збільшення частоти дискретизації з 2 кГц в $3/2$ рази.

Необхідне перетворення частоти дискретизації проводиться за два етапи: 1)

інтерполяція даних із кроком L і 2) децимація із кроком M . Необхідно, щоб процес інтерполяції передував децимації, оскільки в протилежному випадку при децимації зникнуть деякі необхідні частотні компоненти. У наведеному вище прикладі (з компакт-диску на цифрову аудіоплівку) перетворення частоти в 48/44,1 рази можна досягти так: інтерполювати із кроком $L=160$, а потім провести децимацію із кроком $M=147$. Тобто, спочатку швидкість даних компакт-диска збільшуємо в $L=160$ разів до 7056 кГц, а потім зменшуємо в $M=147$ разів до 48 кГц.

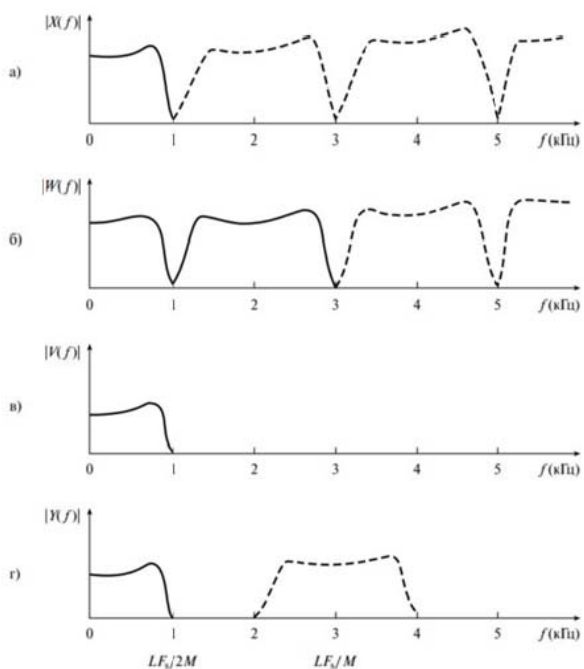


Рис. 4.5. Графіки збільшення частоти дискретизації з 2 кГц в 3/2 рази

На рис. 4.6 (а) наведена блок-схема перетворення частоти дискретизації з нецілим кроком. Оскільки два фільтри нижніх частот $h_1(k)$ і $h_2(k)$ з'єднані касадно і мають загальну частоту дискретизації, їх можна об'єднати, і тоді вийде один узагальнений конвертер частоти дискретизації. Відповідна блок-схема

наведена на рис. 4.6 (б). На рис. 4 (в-е) наведені графіки процесу інтерполяції з нецілим кроком для ($L=3, M=2$). Якщо $M > L$, то вироблена конвертером операція називається децимацією з нецілим кроком, а якщо $M < L$ – інтерполяцією. Якщо $M=1$, дія узагальненої системи зводиться до простої інтерполяції із цілим кроком, описаної раніше, а при $L=1$ вона зводиться до децимації із цілим кроком.

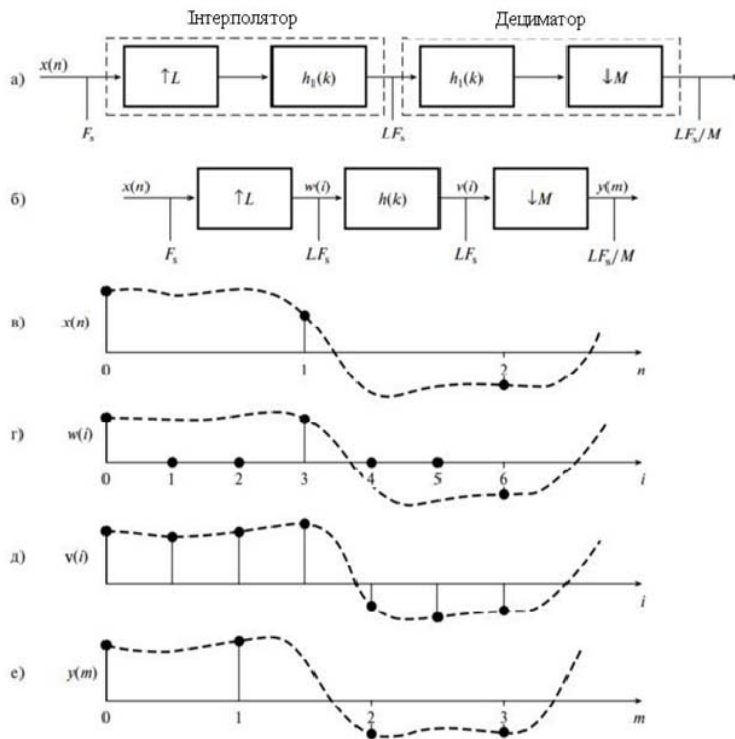


Рис. 4.6. Блок-схеми та графіки перетворення частоти дискретизації з нецілим кроком.

4.4. Багатокаскадне перетворення частоти дискретизації

Раніше описано, що зміна частоти дискретизації відбувалася відразу з використанням єдиного коефіцієнта децимації або інтерполяції. Якщо потрібна суттєва зміна частоти дискретизації, то такий підхід неефективний; набагато краще перетворити частоту за кілька етапів (використовуючи кілька каскадів).

Багатокаскадний підхід використовується у більшості практичних систем обробки при декількох швидкостях.

Він поступово знижує або збільшує частоту дискретизації, що приводить до істотного зниження вимог до фільтрів захисту від накладення спектрів або придушення дзеркальних частот кожного каскаду. Загальний крок децимації M виражається як добуток менших кроків:

$$M = M_1 M_2 \dots M_I,$$

де ціле число M_i – крок децимації i -го каскаду ($i = 1, 2, \dots, I$). Блок-схема i -каскадного процесу децимації наведена на рис. 4.7.

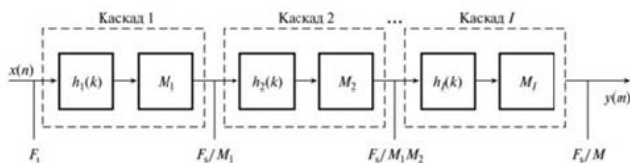


Рис. 4.7. Блок-схема процесу багатокаскадної децимації.

Кожен каскад – це незалежний дециматор, позначений пунктирним прямокутником. Якщо $M \gg 1$, то багатокаскадний підхід приводить до значного зниження вимог до обчислень, пам'яті та характеристик фільтрів, використовуваних у дециматорах, а, отже, до використання фільтрів, менш чутливих до ефектів скінченної розрядності.

Названі переваги можливі за рахунок збільшення складності розробки і реалізації системи.

На практиці, зміни частоти дискретизації реалізуються у двох чи більше каскадах для досягнення максимальної обчислювальної ефективності або мінімальних вимог до пам'яті. Цифрові фільтри, використовувані в системах з обробкою при декількох швидкостях, мають не такі строгі специфікації, що виражається в меншій кількості коефіцієнтів, а, отже, меншій чутливості до ефектів скінченної розрядності. Основна перевага систем з обробкою при декількох швидкостях – це можливість використання переваг цифрової обробки

сигналів. Зокрема, йдеться про використання ЦОС для обмеження смуги сигналу майже до частоти Найквіста зі значним загасанням і без порушення умов теореми про частотне подання. Переваги такого підходу виявилися в багатьох областях, включаючи відтворення компакт-дисків, цифрову фільтрацію, збір даних тощо.

4.5. Завдання для самоперевірки

1. Зменшення частоти дискретизації називається:
А) інтерполяція; Б) децимація; В) модуляція; Г) апроксимація; Д) немає
правильної відповіді.
2. Збільшення частоти дискретизації називається:
А) інтерполяція; Б) децимація; В) модуляція; Г) апроксимація; Д) немає
правильної відповіді.
3. При перетворенні частоти дискретизації з нецілим кроком проводиться:
А) лише інтерполяція; Б) лише децимація; В) децимація після інтерполяції;
Г) інтерполяція після децимації; Д) немає правильної відповіді.
4. Багато каскадне перетворення частоти дискретизації проводиться при:
А) незначній зміні частоти дискретизації; Б) високих вимогах до фільтрів;
В) невеликій обчислювальній ефективності; Г) значній зміні частоти
дискретизації; Д) немає правильної відповіді
5. Перетворення частоти дискретизації з нецілим кроком забезпечує:
А) спрощення обчислень; Б) зменшення апаратних затрат; В) отримання
точніших характеристик вихідного сигналу; Г) суттєвого зменшення кількості
точок дискретизації; Д) немає правильної відповіді.

Правильні відповіді до завдань:

- 1). Б; 2) А 3) В; 4) Г; 5) В.

РОЗДІЛ 5. ПРОЕКТУВАННЯ ЦИФРОВИХ ФІЛЬТРІВ

5.1. Лінійні дискретні системи обробки сигналів: опис в часовій області

Системою обробки сигналів (далі – системою) називається об’єкт, що здійснює необхідне перетворення вхідного сигналу у вихідний. При цьому вхідний сигнал називається “впливом” вихідний – “відгуком” або “реакцією”. Взаємозв’язок між вхідними і вихідними сигналами – співвідношення вхід/вихід – описується рівнянням $Y = F(X)$, де X, Y – вектори, елементами яких є вхідні і вихідні функції часу, а F – оператор, що задає математичне перетворення. Для систем з одним входом і одним виходом, рівняння має вид $y = F(x)$. За визначенням, системою можна називати як фізичний пристрій, так і математичне перетворення F .

Дамо деякі необхідні означення.

1. Система називається лінійною, якщо для неї виконуються дві умови:

- відгук на суму вхідних сигналів дорівнює сумі відгуків на кожен з цих сигналів окремо – це властивість адитивності чи принцип суперпозиції:

$$F(x_1 + x_2 + \dots) = F(x_1) + F(x_2) + \dots$$

- відгук на вхідний сигнал, посилений у певну кількість разів, буде посилений у ту ж кількість разів – ця властивість називається однорідністю або гомогенністю:

$$F(\alpha x) = \alpha F(x).$$

Співвідношення вхід/вихід лінійної системи описують лінійним рівнянням.

2. Система називається дискретною, якщо вона перетворює вхідний дискретний сигнал $x(n)$ у вихідний дискретний сигнал $y(n)$. Дані сигнали можуть бути як дійсними, так і комплексними.

3. Дискретна система називається стаціонарною, якщо її відгук інваріантний (незалежний) до початку відліку часу – властивість інваріантності в часі. Тобто

якщо маємо відгуки $y(n)$ та $y_1(n) = y(n - m)$, то для довільного цілого m справедлива рівність:

$$y(n) = y_1(n + m)$$

Параметри стаціонарної системи незалежні від часу. В стаціонарних системах затримка вхідних сигналів $x(n - m)$ на час m приводить до затримки реакції $y(n - m)$ на той же час. Тому такі системи ще називають інваріантними до зсуву. Властивості таких систем можна досліджувати в будь-які довільні моменти часу.

4. Початкові умови в дискретних системах можуть бути нульовими, або ненульовими. Ознакою нульових початкових умов є відсутність відгуку при відсутності впливу, тобто $y(n) = 0$, якщо $x(n) = 0$. Ознакою ненульових початкових умов є ненульовий відгук (вільні коливання) при відсутності вхідних сигналів.

5. Дискретна система називається такою, що може бути фізично реалізованою (або просто фізично реалізованою), якщо за нульових початкових умов відгук не може виникнути раніше ніж вплив. Значення реакції в кожен момент часу залежать від поточного значення входу $x(n)$ і попередніх значень вхідних впливів $x(n - m), m > 0$, але не залежать від наступних вхідних значень $x(n + m), m \geq 1$. Умови фізичної реалізованості відображають причинно-наслідковий зв'язок відгуку системи з її входом, або принцип причинності, казуальності.

Лінійні дискретні системи (ЛДС) володіють двома особливо важливими властивостями: постійний сигнал на вході переводиться довільною ЛДС в постійний сигнал; при проходженні через ЛДС гармонійний сигнал не міняє частоту, може змінитися лише амплітуда та фаза.

Опис лінійних дискретних систем в часовій області.

Основною характеристикою ЛДС в часовій області є імпульсна характеристика. Імпульсною характеристикою $h(n)$ лінійної дискретної системи називається її відгук (реакція) на цифровий одиничний імпульс $u_0(n)$ за нульових початкових умов. Імпульсна характеристика дозволяє визначити реакцію системи на довільний вхідний сигнал.

Перехідною характеристикою $g(n)$ лінійної дискретної системи називається її відгук на цифровий одиничний стрибок $u_1(n)$ за нульових початкових умов.



Перехідна характеристика ЛДС пов'язана з її імпульсною характеристикою таким співвідношенням:

$$g(n) = \sum_{m=0}^n h(m)$$

Співвідношення вхід/вихід відображає зв'язок між вхідним $x(n)$ і вихідним $y(n)$ сигналами ЛДС, тобто її відгук на довільний вплив. В часовій області це співвідношення описується лінійними рівняннями двох видів:

- формулою згортки (згорткою) – якщо використовується імпульсна характеристика;
- різницеvim рівнянням – якщо використовуються параметри ЛДС.

Застосування згортки для знаходження відгуку ЛДС.

Нагадаємо, що згортка описується виразом:

$$y(n) = x(n) \otimes h(n) = \sum_{k=-\infty}^{\infty} x(k) \cdot h(n-k) \text{ або } y(n) = h(n) * x(n) = \sum_{k=-\infty}^{\infty} x(n-k) \cdot h(k) \quad (5.1)$$

З точки зору ЛДС, в якості ядра згортки виступає імпульсна характеристика системи, тобто:

$h(n)$ – відгук системи на одиничний імпульс;

$x(n)$ – довільний вплив на систему;

$y(n)$ – реакція системи на вхідний вплив.

Якщо тривалість вхідного сигналу або імпульсної характеристики безмежна, то відгук системи теж буде безмежний. Якщо ж тривалість вхідного сигналу становить N відліків, а імпульсної характеристики – M , то довжина відгуку становитиме $L=N+M-1$ відлік, а у формулі згортки (5.1) межі підсумовування зміняться:

$$y(n) = x(n) \otimes h(n) = \sum_{k=0}^{L-1} x(k) \cdot h(n-k) = \sum_{k=0}^{L-1} x(n-k) \cdot h(k) \quad (5.2)$$

ЛДС, для якої вхід/вихід описується формулою згортки, задовольняє умови фізичної реалізованості за нульових початкових умов:

$$\begin{cases} x(n-m)|_{n-m < 0} = 0 \\ h(n-m)|_{n-m < 0} = 0 \end{cases}$$

Це означає, що відгук не може виникнути раніше, ніж вплив. Значення реакції в кожен момент часу залежать від поточного і попередніх значень вхідних впливів, але не залежать від наступних вхідних значень.

Лінійні рівняння згортки вирішуються методом прямої підстановки [2,22], тобто вони безпосередньо визначають алгоритм обчислення відгуку за відомим вхідним сигналом та відомою імпульсною характеристикою системи.

Застосування різницевих рівнянь для знаходження відгуку ЛДС.

Поряд з використанням згортки, для визначення відгуку ЛДС на заданий вхідний сигнал, використовуються лінійні різницеві рівняння (схеми). У загальному випадку такі рівняння записують так:

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) - \sum_{k=1}^{M-1} a_k y(n-k) \quad (5.3)$$

де b_i, a_k – коефіцієнти (дійсні константи);

$x(n), y(n)$ – вхідний вплив та відгук (дійсні або комплексні числа);

i, k – значення затримки для входу і відгуку, відповідно;

N, M – довжини послідовностей, константи;

$x(n-i), y(n-k)$ – вхідний сигнал і відгук системи, затримані на i та k відліків (періодів дискретизації), відповідно.

Коефіцієнти b_i, a_k називають внутрішніми параметрами або просто параметрами ЛДС. ЛДС, для якої вхід/вихід описується різницевою схемою, задовольняє умови фізичної реалізованості за нульових початкових умов.

Різницеві рівняння, як і рівняння згортки, вирішуються методом прямої підстановки, тобто вони безпосередньо визначають алгоритм обчислення відгуку за відомим вхідним сигналом та відомими параметрами ЛДС. Такий метод є

точним і не вносить жодної математичної похибки. (На відміну від аналогового випадку, при якому вхід/вихід системи описується диференціальним рівнянням. А такі рівняння розв'язуються за допомогою одного з методів чисельного інтегрування, які завжди є наближеними.)

Якщо в якості вхідного сигналу $x(n)$ розглянути цифровий одиничний імпульс $u_0(n)$, то отримаємо імпульсну характеристику системи.

Рекурсивні та нерекурсивні лінійні дискретні системи.

ЛДС називається рекурсивною, якщо хоча б один з коефіцієнтів a_k ($k=1,2,\dots,M$) різницевого рівняння не дорівнює нулю. Порядок рекурсивної ЛДС дорівнює порядку різницевого рівняння, тобто $\max\{(M-1),(N-1)\}$.

За визначенням, відгук рекурсивної ЛДС визначається:

- поточним відліком вхідного сигналу (впливу);
- передісторією впливів $x(n-i)$ ($i=1,2,\dots,N-1$);
- передісторією відгуків $y(n-k)$ ($k=1,2,\dots,M-1$).

ЛДС називається нерекурсивною, якщо всі коефіцієнти a_k ($k=1,2,\dots,M$) різницевого рівняння дорівнюють нулю. Для нерекурсивних систем, різницеві рівняння набувають виду:

$$y(n) = \sum_{i=0}^{N-1} b_i x(n-i) \quad (5.4)$$

Порядок такої системи дорівнює $N-1$.

За визначенням, відгук нерекурсивної ЛДС визначається:

- поточним відліком вхідного сигналу (впливу);
- передісторією впливів $x(n-i)$ ($i=1,2,\dots,N-1$);

Системи зі скінченною та нескінченною імпульсною характеристикою.

Розглянемо, особливості рекурсивних та нерекурсивних ЛДС.

Приклад 1. Знайти імпульсну характеристику нерекурсивної ЛДС другого порядку, заданої різницевою схемою:

$$y(n) = b_0 x(n) + b_1 x(n-1) + b_2 x(n-2).$$

Оскільки, імпульсна характеристика – це відгук системи на одиничний імпульс, то очевидно, що $h(n) = y(n)$ якщо $x(n) = u_0(n)$. Тому різницеве рівняння можна записати:

$$h(n) = b_0 u_0(n) + b_1 u_0(n-1) + b_2 u_0(n-2).$$

Використовуючи метод прямої підстановки за нульових початкових умов, отримаємо такі значення:

$$h(0) = b_0 u_0(0) + b_1 u_0(-1) + b_2 u_0(-2) = b_0 \cdot 1 + b_1 \cdot 0 + b_2 \cdot 0 = b_0;$$

$$h(1) = b_0 u_0(1) + b_1 u_0(0) + b_2 u_0(-1) = b_0 \cdot 0 + b_1 \cdot 1 + b_2 \cdot 0 = b_1;$$

$$h(2) = b_0 u_0(2) + b_1 u_0(1) + b_2 u_0(0) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 1 = b_2;$$

$$h(3) = b_0 u_0(3) + b_1 u_0(2) + b_2 u_0(1) = b_0 \cdot 0 + b_1 \cdot 0 + b_2 \cdot 0 = 0;$$

Очевидно, що для $n > 3$ відліки дорівнюють нулю.

Узагальнюючи отримані результати для нерекурсивних ЛДС, маємо:

- імпульсна характеристика нерекурсивної ЛДС має скінчену тривалість (є скінченною);
- значення відліків імпульсної характеристики нерекурсивної ЛДС відповідають коефіцієнтам різницевого рівняння:

$$h(n) = b_i, \quad n = 0, 1, \dots, N-1.$$

Тому нерекурсивні ЛДС називають системами зі скінченною імпульсною характеристикою (СІХ).

Приклад 2. Знайти імпульсну характеристику рекурсивної ЛДС першого порядку, задану різницевою схемою:

$$y(n) = b_0 x(n) - a_1 y(n-1)$$

Оскільки $h(n) = y(n)$ при $x(n) = u_0(n)$, то маємо $h(n) = b_0 u_0(n) - a_1 u_0(n-1)$ і за нульових початкових умов отримаємо такі значення:

$$h(0) = b_0 u_0(0) - a_1 h_0(-1) = b_0;$$

$$h(1) = b_0 u_0(1) - a_1 h_0(0) = -a_1 b_0;$$

$$h(2) = b_0 u_0(2) - a_1 h_0(1) = a_1^2 b_0;$$

$$h(3) = b_0 u_0(3) - a_1 h_0(2) = -a_1^3 b_0;$$

Тобто, справедлива така загальна формула: $h(n) = (-1)^n a_1^n b_0$.

Узагальнивши отримані результати для рекурсивних ЛДС, робимо висновок, що імпульсна характеристика рекурсивних ЛДС має нескінчену тривалість. Тому рекурсивні ЛДС називають системами з нескінченною імпульсною характеристикою (НІХ).

Властивості лінійних дискретних систем

Загальні властивості всіх стаціонарних лінійних дискретних систем:

- адитивність (принцип суперпозиції);
- однорідність (гомогенність);
- інваріантність у часі (інваріантність до зсуву);
- пам'ять;
- стійкість.

Пам'ять ЛДС. Властивістю пам'яті системи є її здатність зберігати (пам'ятати) передісторію, тобто попередні відліки вхідних сигналів (впливів) при знаходженні відгуків в поточний момент часу. Довжина передісторії, тобто кількість попередніх відліків вхідних впливів, визначає глибину (довжину, тривалість) пам'яті.

У нерекурсивних ЛДС, для визначення відгуку в певний момент часу, використовують $(N-1)$ попередній відлік вхідного сигналу. Отже, нерекурсивна ЛДС володіє властивістю пам'яті, її глибина скінченна і становить $(N-1)$. Для обчислення поточного відліку реакції використовуються попередні відліки реакції, котрі в свою чергу можна виразити через відліки вхідного впливу:

З цього випливає, при обчисленні відліку реакції в поточний момент часу система пам'ятає всю передісторію впливу, а отже рекурсивна ЛДС володіє властивістю пам'яті і її довжина в загальному випадку нескінченна. Це пояснюється присутністю зворотного зв'язку у формулі різницевого рівняння для рекурсивних ЛДС, завдяки чому довільний ненульовий відлік впливу нескінченно (постійно) "циркулює" в системі. (в реальних системах він поступово загасає, проте теоретично – присутній завжди).

Стійкість ЛДС. Система називається стійкою, якщо довільний обмежений вхідний сигнал (вплив) породжує обмежений вихідний сигнал (відгук) за довільних обмежених початкових умов. Математично це можна описати так: якщо $\max_n |x(n)| \leq R_x$ то $\max_n |y(n)| \leq R_y$, де $R_x, R_y \neq \infty$.

Існують два критерії стійкості ЛДС [23]. Один з них дозволяє оцінити стійкість системи за її характеристикою у часовій області, а інший по z -відображенню цієї характеристики. *Критерій стійкості ЛДС у часовій області* базується на оцінюванні імпульсної характеристики цієї системи і формулюється так.

Лінійна дискретна система стійка тоді і тільки тоді, коли її імпульсна характеристика повністю сумована, тобто коли виконується умова абсолютної збіжності ряду [22,23]:

$$S = \sum_{n=0}^{\infty} |h(n)| < \infty \quad (5.5)$$

Дана умова є необхідною і достатньою для стійкості системи. Доведення ґрунтується на обмеженості вхідного впливу і використанні формули згортки для обчислення імпульсної характеристики.

Критерій стійкості дозволяє стверджувати, що нерекурсивні ЛДС (СІХ-системи) завжди є стійкими, оскільки їх імпульсна характеристика скінченна. Що стосується рекурсивних лінійних систем, то вони можуть бути як стійкими, так і нестійкими, тому потребують перевірки (дослідження) на стійкість. Імпульсна характеристика стійкої рекурсивної системи завжди матиме характер загасаючої в часі функції.

5.2. Лінійні дискретні системи обробки сигналів: опис в частотній області

Опис лінійних дискретних систем в z -області. Як самостійний довільний сигнал, так і систему обробки сигналів можна представляти не лише в часовій області, але й у комплексній z -області. Це дозволить при описі систем перейти від

різницевих рівнянь до алгебраїчних, спростити аналіз систем на стійкість та забезпечить автоматичний простий перехід до частотних характеристик систем.

В z -області основною характеристикою ЛДС є *функція передачі (передатна функція, transfer function)*, яка визначається як z -перетворення імпульсної характеристики $h(n)$:

$$H(z) = Z\{h(n)\} = \sum_{n=0}^{\infty} h(n)z^{-n}. \quad (5.6)$$

Маючи функцію передачі, імпульсну характеристику можна знайти за допомогою оберненого z -перетворення: $h(n) = Z^{-1}\{H(z)\}$.

Згортці в z -області відповідає добуток z -відображень (за аналогією до спектрів – теорема про згортку):

якщо $x(n)$ – вхідний сигнал ЛДС,
 $h(n)$ – імпульсна характеристика ЛДС,
 $y(n)$ – вихідний сигнал ЛДС;
 $X(z), H(z), Y(z)$ – z -відображення відповідних сигналів,

то в z -області будемо мати:

$$Y(z) = H(z)X(z)$$

Це дозволяє записати функцію передачі ще одним способом, а саме:

$$H(z) = \frac{Y(z)}{X(z)} \quad (5.7)$$

Тобто, функцію передачі $H(z)$ лінійної дискретної системи отримуємо як відношення z -відображення відгуку (реакції) до z -відображення вхідного впливу за нульових початкових умов.

Різницевій схемі в z -області відповідає рівняння, отримане прямою підстановкою в формулу z -перетворення, використанням властивості лінійності та зведенням подібних доданків:

$$Y(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} X(z).$$

Поділивши обидві частини цього рівняння на $X(z)$, отримуємо функцію

передачі, що не залежить ні від входу, ні від відгуку, а виражена виключно через внутрішні параметри ЛДС:

$$H(z) = \frac{\sum_{i=0}^{N-1} b_i z^{-i}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}} \quad (5.8)$$

Таким чином, в загальному випадку, функція передачі – це дробово-раціональна функція, чисельник і знаменник якої є многочленами відносно z^{-1} , порядку $(N-1)$ та $(M-1)$ з дійсними коефіцієнтами b_i та a_k , відповідно. Порядок передатної функції становить $\max\{(N-1), (M-1)\}$. Як правило, вважається, що порядок многочлена чисельника не більший порядку многочлена знаменника: $(N-1) \leq (M-1)$.

Передатна функція першого порядку має вигляд $H(z) = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1}}$, а передатна функція другого порядку – $H(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$. Найпростіші ЛДС, що описуються передатними функціями першого та другого порядку, називаються ланками першого і другого порядку. Ланку називають базовою, якщо чисельник її передатної функції дорівнює одиниці.

Нулями (zero) називають значення z , при яких передатна функція дорівнює нулю. *Полюсами* (pole) (особливими точками) називають значення z , при яких знаменник передатної функції дорівнює нулю. Сукупність нулів та полюсів функції передачі, графічно зображена на z -площині, називається *картою нулів та полюсів*. Така карта – одна з найважливіших графічних характеристик ЛДС. Якщо чисельник передатної функції є многочленом нульового степеня, то використовують термін *передатна функція полюсного виду*.

Критерій стійкості ЛДС в z -області. Для того, щоб ЛДС була стійкою необхідно і достатньо, щоб всі полюси її передатної функції знаходились всередині одиничного кола (кола одиничного радіуса) комплексної z -площини, тобто $|p_k| < 1, k = 1, 2, \dots, M-1$.

Даний критерій найчастіше використовується при дослідженні ЛДС, оскільки він найзручніший у використанні, та вносить певне обмеження. Можливі ситуації, коли за безпосередньою перевіркою в часовій області система є стійкою, а за критерієм функції передачі – ні. В таких випадках вважається, що система стійка лише тоді, якщо всі її полюси знаходяться в межах одиничного кола (навіть ті, що скорочуються в процесі розв'язку рівнянь).

Опис лінійних дискретних систем в частотній області.

Опис в частотній області дозволяє визначити поняття частотної характеристики ЛДС (саме до неї висуваються вимоги при побудові реальних систем) та визначити реакцію ЛДС на гармонійне коливання та на довільний сигнал, що можна подати як лінійну комбінацію гармонійних коливань.

В частотній області основною характеристикою ЛДС є Фур'є-відображення імпульсної характеристики $h(n)$, яке визначається за допомогою перетворення Фур'є (приведено для нормованих частоти та часу):

$$H(\hat{\omega}) = \sum_{n=0}^{\infty} h(n)e^{-j\hat{\omega}n} \quad (5.9)$$

Це називається *комплексною частотною характеристикою* або просто *частотною характеристикою* (ЧХ). За відомою ЧХ, імпульсна характеристика знаходиться за допомогою оберненого перетворення Фур'є.

Комплексну функцію $H(\hat{\omega})$ можна виразити через її модуль та аргумент:

$$H(\hat{\omega}) = |H(\hat{\omega})| e^{j\arg\{H(\hat{\omega})\}} = A(\hat{\omega})e^{j\varphi(\hat{\omega})}$$

Модуль частотної характеристики називають *амплітудно-частотною характеристикою* (АЧХ), а аргумент – *фазо-частотною характеристикою* (ФЧХ):

$$A(\hat{\omega}) = |H(\hat{\omega})| \quad \varphi(\hat{\omega}) = \arg\{H(\hat{\omega})\}.$$

Якщо вхідним сигналом ЛДС є дискретне комплексне гармонійне коливання $x(n) = C_x e^{j\varphi_x(\hat{\omega})}$ з амплітудою та фазою відповідно $C_x = const$; $\varphi_x(\hat{\omega}) = \hat{\omega}n$, то реакція ЛДС $y(n)$ буде визначатися так:

$$y(n) = x(n)H(\hat{\omega}).$$

Підставивши значення вхідного сигналу та ЧХ, маємо:

$$y(n) = C_x e^{j\varphi_x(\hat{\omega})} \cdot A(\hat{\omega}) e^{j\varphi(\hat{\omega})} = C_x A(\hat{\omega}) e^{j[\varphi_x(\hat{\omega}) + \varphi(\hat{\omega})]} = C_y e^{j\varphi_y(\hat{\omega})}.$$

Тобто, реакція ЛДС на комплексний гармонійний сигнал є комплексним гармонійним сигналом тієї ж частоти, що й вплив, з частотно залежними амплітудою та фазою:

$$C_y = C_x A(\hat{\omega}); \quad \varphi_y = \varphi_x(\hat{\omega}) + \varphi(\hat{\omega}).$$

Враховавши цей факт, частотну характеристику ЛДС можна подати як відношення двох гармонійних сигналів – впливу та відгуку:

$$H(\hat{\omega}) = \left. \frac{y(n)}{x(n)} \right|_{x(n)=C_x e^{j\hat{\omega}n}} = \frac{C_y}{C_x} e^{j[\varphi_y(\hat{\omega}) - \varphi_x(\hat{\omega})]}. \quad (5.10)$$

Тобто, *частотною характеристикою* $H(\hat{\omega})$ лінійної дискретної системи називається частотна залежність відношення відгуку (реакції) до дискретного гармонійного сигналу (впливу) в усталеному режимі.

Важливим є те, що для лінійних систем відношення двох функцій часу (відгук та вплив) дає функцію, що від часу не залежить. Щодо поняття “усталений режим”, то він передбачає, що ЛДС працює вже після проходження перехідних процесів, які виникають при подачі вхідного сигналу, тривають деякий час n_0 і не можуть вважатися періодичними.

АЧХ $A(\hat{\omega})$ лінійної дискретної системи називається частотна залежність відношення амплітуди відгуку (реакції) до амплітуди дискретного гармонійного сигналу (впливу) в усталеному режимі:

$$A(\hat{\omega}) = \frac{C_y}{C_x} \quad (5.11)$$

ФЧХ $\varphi(\hat{\omega})$ лінійної дискретної системи називається частотна залежність різниці фаз відгуку і вхідного дискретного гармонійного сигналу (впливу) в усталеному режимі:

$$\varphi(\hat{\omega}) = \varphi_y(\hat{\omega}) - \varphi_x(\hat{\omega}) \quad (5.12)$$

Зв'язок частотної характеристики з передатною функцією.

Оскільки між перетворенням Фур'є та z -перетворенням існує однозначний зв'язок, то, маючи передатну функцію, можна автоматично отримати частотну характеристику шляхом підстановки $z = e^{j\hat{\omega}}$:

$$H(\hat{\omega}) = \frac{\sum_{i=0}^{N-1} b_i e^{-ji\hat{\omega}}}{1 + \sum_{k=1}^{M-1} a_k e^{-jk\hat{\omega}}} \quad (5.13)$$

Важливим є те, що частотна характеристика як і функція передачі залежить виключно від внутрішніх параметрів ЛДС і не залежить від впливу та від відгуку.

Оскільки співвідношення вхід/вихід в z -області дорівнює $H(z) = \frac{Y(z)}{X(z)}$, то в частотній області після відповідної підстановки маємо:

$$H(\hat{\omega}) = \frac{Y(\hat{\omega})}{X(\hat{\omega})},$$

де $X(\hat{\omega})$ – Фур'є-образ впливу, $Y(\hat{\omega})$ – Фур'є-образ відгуку. Тобто, частотну характеристику ЛДС можна визначити як відношення Фур'є-образів відгуку і впливу за нульових початкових умов.

5.3. Основні операції фільтрації та типи фільтрів

Цифрова фільтрація даних (сигналів) є однією з основних і найпоширеніших задач цифрової обробки інформації [2,10]. Під фільтрацією будемо розуміти будь-яке перетворення інформації, в нашому випадку – сигналів, при якому у вхідній послідовності оброблюваних даних цілеспрямовано змінюються певні співвідношення (динамічні або частотні) між різними компонентами цих даних.

Як відомо, перетворення динаміки сигналів (і даних, які несуть ці сигнали) здійснюється в системах. Системи, що вибірково міняють форму сигналів (амплітудно-частотну або фазочастотну характеристику) називають фільтрами. Відповідно, фільтри з будь-яким цільовим призначенням є частковим випадком систем перетворення сигналів, а у вужчому розгляді – лінійних дискретних

систем.

До основних операцій фільтрації інформації відносять:

- згладжування;
- прогнозування;
- диференціювання;
- інтегрування;
- поділ на певні складові;
- виділення інформаційних (корисних) сигналів;
- придушення шумів (завад).

У загальному випадку терміном *цифровий фільтр* (ЦФ) називають апаратну або програмну реалізацію математичного алгоритму, входом якого є цифровий сигнал, а виходом – інший цифровий сигнал з певним чином модифікованою формою і/або амплітудною і фазовою характеристикою. Класифікація цифрових фільтрів звичайно базується на функціональних ознаках алгоритмів цифрової фільтрації, відповідно до якого ЦФ підрозділяються на 4 групи:

- фільтри частотної селекції;
- оптимальні (квазіоптимальні);
- адаптивні;
- евристичні.

Найбільш вивченими і випробуваними на практиці [10,23] є ЦФ частотної селекції. Оскільки фільтрація фактично є операцією згортки в часовій області, то в частотній області їй відповідає процедура множення спектру вхідного сигналу на частотну характеристику фільтра. У цьому разі довжина ядра згортки – це розмір фільтра.

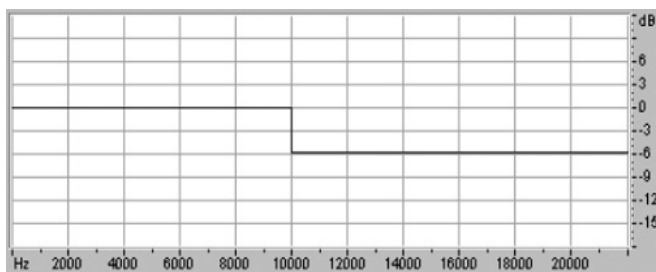
Зміна спектру в задачах звукозапису дозволяє очищати звук від шумів, міняти тембри інструментів, компенсувати перекручування сигналу, породжені, наприклад, мікрофоном. При обробці зображень, фільтрація дозволяє виділяти границі об'єктів, розмивати зображення, добиватися ефекту стиснення, вирівнювати освітлення на фотознімках і т.д.

Оскільки, фільтри є частковим випадком ЛДС, то на них поширюються визначення та властивості цих систем. Так, розрізняють два класи цифрових фільтрів: *нерекурсивні цифрові фільтри*, які також називають *трансверсальними* (від англійського transversal – поперечний), а також *СІХ-фільтрами* та *рекурсивні цифрові фільтри або НІХ-фільтри*. При цьому поняття рекурсивні/нерекурсивні відповідають поняттям ЛДС. Для кожного з цих класів існують свої, принципово різні методи побудови і кожен з них має свої переваги та недоліки. Так, *нерекурсивні фільтри* завжди стійкі, відповідно їх легше проектувати, проте, для забезпечення задовільної АЧХ, вони повинні мати високий порядок (кілька сотень або навіть тисяч).

При частотній селекції фільтр змінює у спектрі вхідного сигналу амплітуду гармонік та їх фази. Фільтри, що не змінюють фазу, називаються *фільтрами з лінійною фазою*. Це означає, що якщо вони змінюють фазу сигналу, то роблять це так, що всі гармоніки (тобто частотні складові) сигналу зсуваються у часі на однакову величину. Ядро згортки такого фільтру (імпульсна характеристика) симетричне відносно своєї центральної точки.

Довільний фільтр описують його АЧХ та ФЧХ. Ці характеристики показують як впливає фільтр на амплітуду і фазу різних гармонік вхідного сигналу. Якщо фільтр має лінійну фазу (наприклад СІХ-фільтри, як правило мають лінійну ФЧХ), то розглядається лише його АЧХ. Зазвичай, вона зображається графіком залежності амплітуди від частоти (в децибелах) (рис. 5.1). Наприклад, якщо фільтр пропускає всі сигнали у смузі 0..10кГц без змін, а всі сигнали в смузі вище 10кГц послаблює у 2 рази (тобто на 6 дБ), то ця характеристика буде описуватися так:

$$A(f) = \begin{cases} 0\text{дБ}, f < 10\text{кГц} \\ -6\text{дБ}, f > 10\text{кГц} \end{cases}$$



5.1. Амплітудно-частотна характеристика заданого фільтру.

Частотна характеристика в 0 дБ для певних частот, показує, що ці частоти фільтр не міняє. Ті частоти, амплітуда яких повинна послаблюватися у 2 рази, повинні мати на 6 дБ меншу амплітуду (тому -6дБ). Якщо ж фільтр посилює певні частотні складові, то його частотна характеристика була б додатною на цих частотах.

Залежно від призначення фільтру, а отже і загального виду його частотної характеристики, виділяють такі основні, найбільш розповсюджені, типи фільтрів (вибіркові фільтри):

- *НЧ-фільтри (low-pass filters)* – фільтр низьких частот – пропускає всі частоти нижче заданої;
- *ВЧ-фільтри (high-pass filters)* – фільтр високих частот;
- *смугові фільтри (band-pass filters)* – пропускають певну смугу частот;
- *режекторні фільтри (band-reject filters)* – пригашають смугу частот.

Як правило, в задачах фільтрації для фільтру задається потрібна частотна характеристика, а метою є побудувати фільтр, що відповідає заданим вимогам та провести фільтрацію. Оскільки ідеальний фільтр реалізувати фізично неможливо, то будують фільтр, близький за характеристиками до заданого. При цьому слід задавати вимоги до реальних вибірових фільтрів. Такі вимоги передбачають:

- вказати частоту дискретизації f_d (Гц) та тип вибіровості (НЧ, ВЧ і т. д.),
- здати вимоги до АЧХ або до характеристики спаду (тобто до логарифмічної АЧХ), при цьому вимагається:

- привести всі граничні частоти фільтру в основну смугу частот $0 \leq f \leq f_d / 2$;
- вимоги в перехідних смугах вибіркового фільтрів не задаються;
- вимоги формулюються у виді допустимих відхилень від нормованої АЧХ в смугах пропускання і затримки;

- АЧХ є безрозмірною величиною і виражається в абсолютних одиницях;
- зв'язок між характеристиками послаблення (спаду) і затухання $a(f)$ та нормованою АЧХ $\hat{A}(f)$ задається логарифмічним рівнянням $a(f) = 20 \lg \hat{A}(f)$.

Розмірність послаблення виражається в децибелах(дБ).

Наприклад, ідеальний ФНЧ повинен мати такі характеристики:

- f_d – частота дискретизації;
- f_λ – частота зрізу (або гранична частота пропускання);
- δ – максимально допустиме відхилення від 1;
- δ_0 – максимально допустиме відхилення від 0;
- $\Delta f = f_k - f_\lambda$ – ширина перехідної смуги;
- a_{\max} – максимально допустиме послаблення в смузі пропускання, тобто $a_{\max} = 20 \lg(1 - \delta)$, відповідно максимально допустиме загасання в смузі пропускання становить $a_{\max} = -20 \lg(1 - \delta)$;
- a_0 – максимально допустиме послаблення в смузі затримки $a_0 = 20 \lg \delta_0 < 0$, відповідно максимально допустиме загасання в смузі затримки становить $a_0 = a_{\min} = -20 \lg \delta_0 > 0$.

5.4. Розрахунок і побудова цифрових СІХ-фільтрів з частотною вибіркою.

Методи розрахунку. СІХ-фільтр характеризується наступними рівностями: вхідний і вихідний сигнали зв'язані операцією згортки:

$$y(m) = \sum_{k=0}^{N-1} h(n)x(m-n). \quad (5.14)$$

та

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}.$$

На практиці розрахунок вказаних фільтрів полягає в знаходженні коефіцієнтів, які відповідають заданим параметрам (частота чи смуга пропускання фільтра, дискретизація вхідного сигналу, порядок фільтра, послаблення пульсацій і т.д.). Розрахунок коефіцієнтів СІХ-фільтрів виконують за допомогою таких методів:

- метод зважування (вирізання);
- оптимізаційні методи;
- метод частотної вибірки і т.д.

Кожен із запропонованих методів знайшов своє використання залежно від заданих параметрів.

Розрахунок коефіцієнтів СІХ-фільтра за допомогою методу зважування

В даному методі використовується факт, що частотна характеристика фільтра $H_D(\omega)$ і відповідна імпульсна характеристика $h_D(n)$ зв'язані оберненим перетворенням Фур'є:

$$h_D(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_D(\omega) e^{i\omega n} d\omega. \quad (5.15)$$

Якщо $H_D(\omega)$ відома, то $h_D(n)$ можна отримати, застосувавши перетворення Фур'є до обох частин рівності (5.15). Припустимо, що необхідно розробити ФНЧ. Розглянемо імпульсну характеристику ідеального ФНЧ, зображену на рис.5.2 (б). Припустимо, що характеристика змінюється від $-\omega_c$ до ω_c . Тоді, спрощуючи інтегрування, отримаємо наступну імпульсну характеристику:

$$h_D(n) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{i\omega n} d\omega = \begin{cases} \frac{2f_c \sin(n\omega_c)}{n\omega_c}, & n \neq 0 \\ 2f_c, & n = 0 \end{cases}. \quad (5.16)$$

Імпульсні характеристики ідеальних фільтрів верхніх частот, смугових та загорожувальних фільтрів знаходяться на основі рівності (5.16) і всі вони наведені в табл. 5.1.

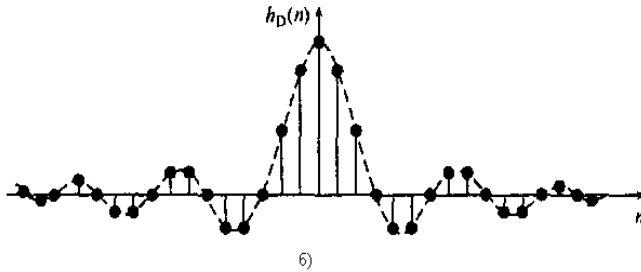
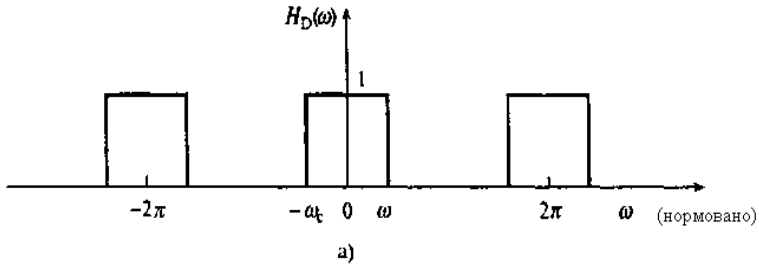


Рис. 5.2. Частотна характеристика (а) та імпульсна характеристика (б) ідеального ФНЧ.

Оскільки імпульсна характеристика $h_D(n)$ зменшується при віддалені від точки $n = 0$, то теоретично вона іде до $n = \pm\infty$. Відповідно, отриманий фільтр не є СІХ-фільтром. Для вирішення цієї ситуації пропонується обрізати ідеальну імпульсну характеристику, взявши $h_D(n) = 0$ для n , більших, ніж певне M . При цьому виникають небажана нерівномірність і викиди – має місце так званий ефект Гібса. Чим більше коефіцієнтів залишилось, тим ближчий спектр фільтра до ідеальної характеристики. Обрізання характеристики $h_D(n)$ рівнозначне множенню ідеальної імпульсної характеристики на прямокутну вагову функцію:

$$w(n) = \begin{cases} 1, & |n| \leq 0, 1, \dots, (M-1)/2 \\ 0 & \end{cases}$$

В частотній області це еквівалентно згортці $H_d(\omega)$ з $W(\omega)$, де $W(\omega)$ – Фур'є-перетворення $w(n)$. На практиці ідеальна частотна характеристика h_D множиться на відповідну вагову функцію $w(n)$.

Таблиця 5.1. Імпульсні характеристики ідеальних фільтрів

Тип фільтра	$h_d(n), n \neq 0$	$h_d(n), n=0$
ФНЧ	$\frac{2f_c \sin(n\omega_c)}{n\omega_c}$	$2f_c$
ФВЧ	$-\frac{2f_c \sin(n\omega_c)}{n\omega_c}$	$1-2f_c$
Смуговий фільтр	$\frac{2f_2 \sin(n\omega_2)}{n\omega_2} - \frac{2f_1 \sin(n\omega_1)}{n\omega_1}$	$2(f_2 - f_1)$
Загороджувальний фільтр	$\frac{2f_1 \sin(n\omega_1)}{n\omega_1} - \frac{2f_2 \sin(n\omega_2)}{n\omega_2}$	$1-2(f_2 - f_1)$

Зв'язок ширини переходу (від смуги пропускання до смуги послаблення) фільтра, побудованого на основі певної вагової функції, з довжиною фільтра виражається залежністю:

$$\Delta f = k / N, \quad (5.17)$$

де N – довжина фільтра, k – коефіцієнт, який розрахований для кожного типу вікна, а Δf – нормована ширина смуги переходу.

Для якісної розробки фільтрів необхідно задати певні специфікації відносно ФЧХ та АЧХ. При розгляді ФЧХ достатньо вказати, яка необхідна симетрія – парна чи непарна (мається на увазі, що ФЧХ лінійна). АЧХ для СІХ-фільтра часто задається у вигляді схеми допусків. Така схема для ФНЧ зображена на рис. 5.3. Подібну схему можна легко отримати й для інших частотно-вибіркових фільтрів. Виходячи із рис. 5.3, особливий інтерес представляють такі параметри:

δ_p – відхилення в смузі пропускання (нерівномірність);

δ_s – відхилення в смузі послаблення;

f_p – гранична частота смуги пропускання;

f_s – гранична частота смуги послаблення;

F_s – частота дискретизації.

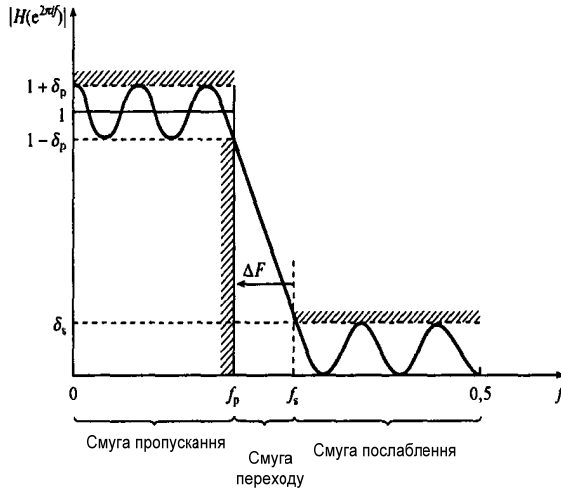


Рис. 5.3. Схема допуску для ФНЧ.

Приклад обчислення параметрів фільтра.

Обчислення параметрів фільтра здійснюється за вхідними даними, які наведені в табл. 5.2.

Таблиця 5.2. Вхідні дані для обчислення характеристик СІХ-фільтра

Вхідний сигнал, $t=0..1000$	Частоти вхідного сигналу, кГц, ω_n	Фільтр	Частоти фільтру, кГц	Ширина смуги переходу, кГц	Частота дискретизації, кГц	Загасання в смугі послаблення, дБ
$\sum_{n=1}^3 \sin(\pi \omega_n t)$	1,5 5,3 6,8	ФНЧ	5,2	0,5	16	-21,1

Використовуючи таблицю 5.1 та враховуючи (5.16), вибираємо $h_D(n)$ для ФНЧ. Із таблиці 5.2 випливає, що вимоги до загасання в смугі послаблення задовільняють функції прямокутного вікна. У цьому разі $\Delta f^2=0,5/16=0,03125$. Для

$N=256$ смуга пропускання рівна 0,0053. Тоді згідно формули (5.17) отримуємо значення коефіцієнта $k=N \cdot \Delta f = 256 \cdot 0,0053 = 1,35$. Знаходимо значення N при $\Delta f = 3,3$: $N = k / \Delta f = 1,35 / 0,03125 = 43,176 \approx 43$. Таким чином, коефіцієнти будуть рівні:

$$h_D(n)w(n) = 1, -21 \leq n \leq 21,$$

де $h_D(n)$ описується виразом (5.16), а $w(n) = 1, -21 \leq n \leq 21$.

Внаслідок ефекту змазування характеристики фільтра, що вноситься ваговою функцією, частота зрізу отриманого фільтра буде відрізнятися від заданої в специфікації. Щоб врахувати цей ефект, використаємо f_c – центр смуги переходу:

$$f_c' = f_c + \Delta f / 2 = (5,2 + 0,25) = 5,75 \text{ кГц}$$

і зауважимо, що $5,75/16 = 0,3593$.

Обчислюємо значення $h_D(n)$ згідно з наведеним раніше виразом (5.16). Оскільки $h_D(n)$ – симетрична функція, то досить врахувати її значення лише на проміжку $0 \leq n \leq 21$:

$n = 0$:

$$h_D(0) = 2f_c = 2 \cdot 0,3593 = 0,7186$$

$$\omega(0) = 1$$

$$h(0) = h_D(0) \cdot \omega(0) = 0,7186 \cdot 1 = 0,7186$$

$n = 1$:

$$h_D(1) = 2f_c \sin(\omega_c) / \omega_c = 2 \cdot 0,3593 \cdot \sin(2\pi \cdot 0,3593) / (2\pi \cdot 0,3593) = \\ = \sin(360^\circ \cdot 0,3593) / \pi = 0,2462$$

$$\omega(1) = 1$$

$$h(1) = h_D(1) \cdot \omega(1) = 0,2462 \cdot 1 = 0,2462$$

$n = 2$:

$$h_D(2) = 2f_c \sin(2\omega_c) / (2\omega_c) = 2 \cdot 0,3593 \cdot \sin(2 \cdot 2\pi \cdot 0,3593) / (2 \cdot 2\pi \cdot 0,3593) = \\ = \sin(360^\circ \cdot 0,3593 \cdot 2) / (2\pi) = -0,1559$$

$$\omega(2) = 1$$

$$h(2) = h_D(2) \cdot \omega(2) = -0,1559 \cdot 1 = -0,1559$$

•••

$n = 21$:

$$h_D(21) = 2f_c \sin(21\omega_c) / (21\omega_c) = 2 \cdot 0,3593 \cdot \sin(21 \cdot 2\pi \cdot 0,3593) / (21 \cdot 2\pi \cdot 0,3593) = \\ = \sin(360^\circ \cdot 0,3593 \cdot 21) / (21\pi) = -0,0042$$

$$\omega(21) = 1$$

$$h(21) = h_D(21) \cdot \omega(21) = -0,0042 \cdot 1 = -0,0042$$

Обчисливши всі коефіцієнти $0 \leq n \leq 21$, решта коефіцієнтів знаходимо за правилом симетрії $h(-n) = h(n)$. Тоді отримані коефіцієнти підставляємо у вираз:

$$y(m) = \sum_{n=0}^{N-1} h(n)x(m-n),$$

$$y(m) = \sum_{n=0}^{42} h(n)x(m-n) = \sum_{n=0}^{42} h(n-N/2)x(m-n),$$

$$y(1) = h(0-21)x(1-0) + h(1-21)x(1-1) + \dots + h(42-21)x(1-42),$$

$$y(2) = h(0-21)x(2-0) + h(1-21)x(2-1) + \dots + h(42-21)x(2-42)$$

•••

$$y(1000) = h(0-21)x(1000-0) + h(1-21)x(1000-1) + \dots + h(42-21)x(1000-42).$$

5.5. Завдання для самоперевірки

1. Імпульсна характеристика системи це:

А) відгук системи на одиничний імпульс; Б) відгук системи на гармонійне коливання; В) результат перетворення Фур'є; Г) результат перетворення Лапласа; Д) немає правильної відповіді/

2. Система називається рекурсивною якщо:

А) її вихід залежить хоча б від одного попереднього відгуку; Б) її вихід не залежить від жодного попереднього відгуку; В) її вихід не залежить від жодного попереднього впливу; Г) її вихід залежить лише від одного попереднього впливу; Д) немає правильної відповіді.

3. Лінійна дискретна система є стійкою якщо для її імпульсної характеристики $h(n)$ виконується умова:

А) $S = \sum_{n=-\infty}^{\infty} |h(n)| = -1$; Б) $S = \sum_{n=-\infty}^{\infty} |h(n)| < \infty$; В) $S = \sum_{n=-\infty}^{\infty} |h(n)| = 0$; Г) завжди; Д) немає

правильної відповіді

4. Яка ланка задається такою передатною функцією $H(z) = \frac{1}{1 + 0.2z^{-1} - 0.35z^{-2}}$?

А) небазова ланка другого порядку; Б) фазова ланка першого порядку; В) базова ланка першого порядку; Г) базова ланка другого порядку; Д) немає правильної відповіді

5. Режекторний фільтр:

А) пропускає всі частоти вище заданої; Б) пропускає всі частоти нижче заданої; В) пригашає визначену смугу частот; Г) пропускає визначену смугу частот; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1). А; 2) А 3) Б; 4) Г; 5) В.

РОЗДІЛ 6. ОПРАЦЮВАННЯ МОВНИХ СИГНАЛІВ

Багато напрямків мовних технологій (опрацювання мовних сигналів з певною метою: стиск мовних сигналів, синтез мови, зміна темпу мовлення, розпізнавання або визначення емоційного стану людини за голосом, діагностика степені певних захворювань, розпізнавання мови) на сьогодні інтенсивно розвиваються та знаходять усе більше застосування в різноманітних сферах. Розпізнавання мови є одним з найскладніших напрямків мовних технологій, який можна застосувати в багатьох областях.

Важливим моментом при опрацюванні мовних сигналів у цифровому вигляді є вибір частоти дискретизації та розрядності відліків у бітах при переході за допомогою аналого-цифрового перетворювача від неперервного до дискретного мовного сигналу. Загалом вважається, що для звукових сигналів (спів людини, музика, мова, інші звукові сигнали, скажімо дзенькіт кришталю) гранична частота не перевищує 22 КГц. Тому для дискретизації звукових сигналів беруть стандартні частоти 44,1 КГц або 48 КГц. Розрядність відліків цифрового звукового сигналу – 16 біт.

Проте мовні сигнали зокрема мають звужений діапазон частот – від 0 до 8 КГц. Тому при опрацюванні мови досить дискретизувати неперервні сигнали з частотою дискретизації 16 КГц та брати 16-бітові відліки. У деяких часткових випадках основні спектральні складові сигналів знаходяться в ще вужчому діапазоні, й замість частоти дискретизації 16 КГц можна взяти частоту дискретизації 8 КГц.

6.1. Мовні технології

Виділяють такі напрямки мовних технологій.

1. *Стиск (кодування) мови.* Високого ступеня стиску досягаємо використанням дискретних косинусних перетворень.

2. *Синтез мови.* Завдання полягає в тому, що треба озвучити наявний текст. На сьогодні є два принципово різних підходи до вирішення цієї задачі.

а) компілятивний синтез. Виділяються певні одиниці мови, наприклад звуки мови, після цього з них утворюють слова, речення.

б) формантний синтез. Будується математична модель голосового тракту людини, щоб мати змогу утворювати різні одиниці мови.

3. *Розпізнавання диктора за голосом.* Є два варіанти цієї задачі. Перший варіант, так звана верифікація, полягає в з'ясуванні факту: належить записаний мовний зразок заданій особі чи ні. Інший, ускладнений варіант ідентифікації, полягає у визначенні якій із наперед заданих осіб належить записаний мовний зразок.

4. *Визначення емоційного стану людини за голосом.*

5. *Визначення стану хворого за голосом* (наприклад, при лікуванні мовних розладів).

6. *Зміна темпу мовлення.*

7. *Розпізнавання мови.* Цей напрямок є одним з найскладніших напрямків мовних технологій, який можна застосувати в багатьох областях [27]. Задача полягає в тому, що маючи мовний сигнал, треба утворити відповідний текст (так звану транскрипцію).

Підхід до розпізнавання мовних команд вкладається у загальну схему розпізнавання образів (крім мови це можуть бути двовимірні або тривимірні зображення, багатовимірні сигнали різної природи тощо). Згідно з цією схемою слід виконувати такі кроки:

- на фазі навчання попередньо записати певну множину мовних образів-еталонів;

- під час розпізнавання порівнювати невідомий образ (мовний сигнал), який хочемо розпізнати, з кожним із записаних мовних образів-еталонів;

- найближчий (у сенсі вибраної відстані між образами) до невідомого образу, для якого хочемо здійснити розпізнавання, образ-еталон і буде відповіддю системи розпізнавання.

Ідея лежить на поверхні – проблема в тому, як підійти до її реалізації. Коли ми хочемо реалізувати описану схему для випадку розпізнавання мовних команд виникає багато конкретних труднощів. У якому вигляді готувати й тримати мовні сигнали-еталони ? Використовувати часове, спектральне чи якесь комбіноване зображення сигналів ? Якою міркою порівнювати відстань між двома мовними сигналами ?

Розглянемо найпростіший алгоритм розпізнавання окремих слів із обмеженого словника з навчанням на одного диктора.

6.2. Найпростіший розпізнавач мови

Спочатку опишемо як виконують спрощене попереднє опрацювання мовного сигналу $x(n)$. Пропускаємо його через набір смугопрпусних цифрових фільтрів (кількість фільтрів для прикладу взята рівною 16). Тобто, частотний діапазон сигналу від 0 до 8 кГц (для мовних сигналів) розбиваємо на 16 рівних смуг. Частотні характеристики фільтрів показані на рис. 6.1.

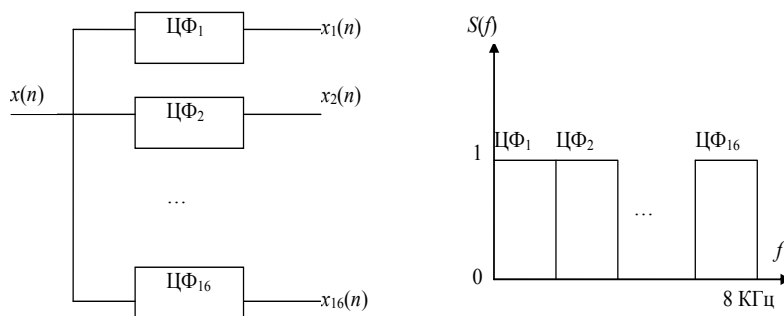


Рис. 6.1. Частотні характеристики фільтрів.

В результаті на виходах цифрових фільтрів $\text{ЦФ}_1, \text{ЦФ}_2, \dots, \text{ЦФ}_{16}$ отримуємо 16 цифрових сигналів $x_1(n), x_2(n), \dots, x_{16}(n)$.

Ділимо число N відліків кожного з цих сигналів на 16 рівних відрізків, доповнюючи сигнал при потребі нульовими відліками, щоб N було кратне 16. Для відліків кожного із шістнадцяти відрізків знаходимо середнє значення. Тобто, перше значення – це середнє відліків сигналу від першого до $N/16$ -го; друге значення – це середнє відліків сигналу від $(N/16+1)$ -го до $2N/16$ -го;...; шістнадцяте значення – це середнє відліків сигналу від $(15N/16+1)$ -го до N -го. Отже, для кожного з шістнадцяти сигналів $x_1(n), x_2(n), \dots, x_{16}(n)$ отримуємо 16 середніх значень.

У результаті для сигналу $x(n)$ маємо загалом вектор із $16 \cdot 16 = 256$ значень.

На фазі навчання диктор проводить навчання системи розпізнавання, повторюючи по три рази (для надійності) слова із вибраного невеликого словника. Наприклад, це можуть бути назви десяткових цифр, тобто словник складається з таких 10 слів: нуль, один, два, три, чотири, п'ять, шість, сім, вісім, дев'ять. Виконуючи описане попереднє опрацювання цих мовних сигналів, отримуємо тридцять векторів довжиною 256 значень кожен. Це мовні еталони системи розпізнавання.

На фазі розпізнавання диктор вимовляє будь-яке невідоме слово з обумовленого словника і записуємо відповідний мовний сигнал. Після його попереднього опрацювання отримуємо вектор довжиною 256 значень. Підраховуємо по черзі відстань між цим вектором і кожним із 30 еталонних векторів.

Відстань між двома векторами $\overset{L}{A} = (a_1, a_2, \dots, a_{256})$ та $\overset{L}{B} = (b_1, b_2, \dots, b_{256})$ задають як евклідову відстань $d(A, B) = \sqrt{\sum_{i=1}^{256} (a_i - b_i)^2}$. Мінімальна із отриманих відстаней визначає відповідь системи розпізнавання. Точність (якість) розпізнавання оцінюємо процентом правильно розпізнаних слів.

Експерименти показують, що описаний найпростіший розпізнавач мови має точність розпізнавання приблизно 85%. Це не можна вважати задовільним показником.

Що потрібно вдосконалити в розглянутому алгоритмі розпізнавання, щоб покращити точність розпізнавання? Зауважимо, що два рази абсолютно однаково вимовити якесь слово неможливо. Неодмінно є відмінності в темпі мови (хоча би незначні) та певні спектральні відмінності.

Тому вдосконалений алгоритм розпізнавання слів із невеликого словника повинен задовільняти таким вимогам:

- порівнювати мовні сигнали з урахуванням часових (темпоральних) відмінностей, при потребі вирівнюючи тривалості мовних сигналів;
- враховувати спектральні зміни в мовному сигналі при зміні цього сигналу в часі. Для цього слід, зокрема, виконувати точне попереднє опрацювання мовних сигналів.

Таким вимогам задовольняє алгоритм розпізнавання мовних команд (ізолюваних слів), який називають алгоритмом DTW (dynamic time warping – динамічне часове вирівнювання) [11, 27].

6.3. Попереднє опрацювання мовних сигналів

Результатом попереднього опрацювання мовних сигналів є отримання множини спектральних векторів, які характеризують цей сигнал і використовуються для подальшого розпізнавання [27, 31].

Принципове припущення, яке робиться в сучасних розпізнавачах є те, що мовний сигнал розглядається як стаціонарний (тобто його спектральні характеристики відносно постійні) на інтервалі в кілька десятків мілісекунд. Тому основною функцією попереднього опрацювання є розбити вхідну мовний сигнал на інтервали і для кожного інтервалу отримати згладжені спектральні оцінки.

Типова величина одного інтервалу – 25,6 мс. Сусідні інтервали беруться зі зсувом відносно попереднього інтервалу. Вживана величина перекриття інтервалів рівна 10 мс. У результаті попереднього опрацювання кожного з вказаних інтервалів отримуємо вектор з кількох десятків спектральних значень.

Кроки, які слід виконати для попереднього опрацювання кожного інтервалу мовного сигналу, детально описані далі.

Як приклад розглядаємо мовні зразки, дискретизовані з частотою 16 КГц та з розрядністю 16 біт. Дискретизований мовний сигнал розбиваємо на інтервали тривалістю 25,6 мс, тобто 409 відліків. Інтервали перекриваються зі зсувом на 10 мс (160 відліків).

Далі даються етапи попереднього опрацювання мовних сигналів.

1. Оцифрований (дискретизований у часі та квантований за рівнем) мовний сигнал розбиваємо на блоки по 25,6 мс із зсувом кожні 10 мс, тобто, блоки по 409 відліків кожен блок, із зсувом на 160 відліків.

2. Як правило, застосовують високочастотне підсилення, щоб компенсувати послаблення, спричинене розсіюванням від губ. Для цього блоки сигналу пропускають через фільтр першого порядку

$$S(1) = 0; S(n) = y(n) - y(n-1), n=2 \dots 409,$$

де $y(k)$ позначає k -й відлік у блоці.

3. Для опрацювань такого типу до кожного блоку застосовують функцію вікна. У даному випадку береться вікно Гемінга згідно з таким виразом:

$$D(n) = (0,54 - 0,46 \cdot \cos(2\pi(n-1)/408)) \cdot S(n) \text{ для } n = 1, \dots, 409.$$

4. Щоб отримати спектральні оцінки використовується дискретне перетворення Фур'є. У цьому разі збільшуємо довжину блоку до 512 елементів за рахунок доповнення його справа потрібною кількістю нулів. Після цього застосовуємо швидке перетворення Фур'є довжиною 512 точок і отримуємо 512 спектральних комплексних значень. Оскільки 512 значень, до яких застосовуємо перетворення Фур'є, є дійсними, то отримані спектральні комплексні значення попарно спряжені: друге значення з 512-м, третє – з 511-м і т.д. Тому останні 256 комплексних значень перетворення ігноруємо, бо вони комплексно спряжені з попередніми й не несуть нової інформації.

5. Для перших 256 комплексних спектральних значень знаходимо їх амплітуди. Амплітудний спектр Фур'є згладжується (усереднюється) додаванням амплітуд спектральних коефіцієнтів у межах “трикутних” частотних смуг

розташованих на нелінійній (подібній до логарифмічної) Mel-шкалі. Для граничної частоти мови рівної 16 КГц беруть 24 таких частотних смуги. Mel-шкала введена для наближення частотного розділення людського вуха, яке є лінійним до 1000 Гц та логарифмічним понад 1000 Гц.

Перший амплітудний коефіцієнт (постійну складову спектру) ігноруємо, а амплітуди решти 255 спектральних значень усереднюємо. Усереднення реалізуємо як 24 трикутні смугопропускні фільтри. Нижня, середня та верхня частоти таких смуг подані в табл. 6.1.

Кожний трикутний фільтр знаходить зважене середнє тих амплітудних спектральних значень, які відповідають частотам у межах між нижньою та верхньою частотою для даного фільтра. Якщо амплітуда відповідає точно середній частоті смуги, то вона множиться на коефіцієнт рівний одиниці. При пересуванні відповідної амплітудному значенню частоти від середини до нижньої чи верхньої межі коефіцієнт зменшується від одиниці до нуля. Отримані добутки амплітуд на коефіцієнти додаються й діляться на число амплітудних значень. У результаті знаходимо зважене середнє для даної смуги частот.

256 амплітудам відповідають частоти від 0 Гц до 8000 Гц, тобто крок пересування дорівнює $8000/256=31,25$ Гц. Це означає, що першій амплітуді відповідає частота 0 Гц, другій – 31,25 Гц, третій – 62,5 Гц і т.д. Наприклад (див. рис. 6.2), для першої смуги частот Mel-шкали: нижня частота – 0 Гц, середня частота – 74,24 Гц, верхня частота – 156,4 Гц. Отже, в першу смугу частот потрапляють перша (0 Гц), друга (31,25 Гц), третя (62,5 Гц), четверта (93,75 Гц), п'ята (125 Гц) та шоста (156,25 Гц) амплітуди. Тоді третій амплітуді відповідає коефіцієнт рівний $62,5/74,24 \approx 0,84$; а п'ятій амплітуді – коефіцієнт рівний $(156,4-125)/(156,4-74,24) \approx 0,38$.

У результаті описаних дій отримуємо 24-елементний спектральний (акустичний) вектор. На закінчення виконуємо нормалізацію акустичних векторів у межах одного мовного зразка. Для цього знаходимо найбільшу довжину вектора та значення всіх векторів множимо на величину, обернену до цієї довжини.

Таблица. 6.1. Мел-шкала частот

Смуга	Нижня частота	Середня частота	Верхня частота
1	0,000 Гц	74,24 Гц	156,4 Гц
2	74,24 Гц	156,4 Гц	247,2 Гц
3	156,4 Гц	247,2 Гц	347,6 Гц
4	247,2 Гц	347,6 Гц	458,7 Гц
5	347,6 Гц	458,7 Гц	581,6 Гц
6	458,7 Гц	581,6 Гц	717,5 Гц
7	581,6 Гц	717,5 Гц	867,9 Гц
8	717,5 Гц	867,9 Гц	1034 Гц
9	867,9 Гц	1034 Гц	1218 Гц
10	1034 Гц	1218 Гц	1422 Гц
11	1218 Гц	1422 Гц	1647 Гц
12	1422 Гц	1647 Гц	1895 Гц
13	1647 Гц	1895 Гц	2171 Гц
14	1895 Гц	2171 Гц	2475 Гц
15	2171 Гц	2475 Гц	2812 Гц
16	2475 Гц	2812 Гц	3184 Гц
17	2812 Гц	3184 Гц	3596 Гц
18	3184 Гц	3596 Гц	4052 Гц
19	3596 Гц	4052 Гц	4556 Гц
20	4052 Гц	4556 Гц	5113 Гц
21	4556 Гц	5113 Гц	5730 Гц
22	5113 Гц	5730 Гц	6412 Гц
23	5730 Гц	6412 Гц	7166 Гц
24	6412 Гц	7166 Гц	8000 Гц

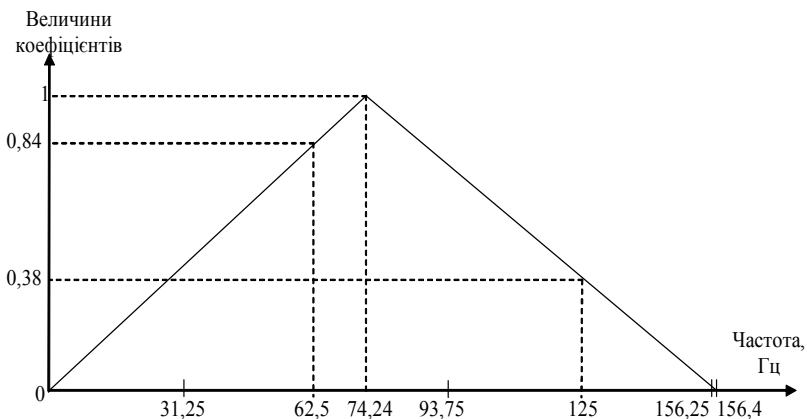


Рис. 6.2. Усереднення амплітудних спектральних значень в межах однієї смуги.

Далі проілюстровано моделювання алгоритму попереднього опрацювання мовних сигналів в середовищі MATLAB. Графічне вікно середовища MATLAB, у якому відображено етапи попереднього опрацювання мовного сигналу показано на рис. 6.3.

На першій ілюстрації показано мовний сигнал `example.wav`, дискретизований з частотою 16 КГц та розрядністю 16 розрядів. На другій ілюстрації маємо один блок (інтервал) вказаного мовного сигналу тривалістю 25,6 мс. Такому блоку відповідає 409 відліків. На третій ілюстрації бачимо один блок мовного сигналу після опрацювання його фільтром першого порядку. Четверта ілюстрація показує нам один блок після застосування вікна Гемінга.

П'ята ілюстрація дає нам 512 амплітудних значень швидкого перетворення Фур'є цього одного блоку. Оскільки ці амплітудні значення швидкого перетворення Фур'є попарно співпадають (бо відповідні комплексні значення швидкого перетворення Фур'є є попарно комплексно спряжені), то можна взяти лише 256 перших амплітудних значень. Ці 256 амплітудних значень відображені на шостій ілюстрації.

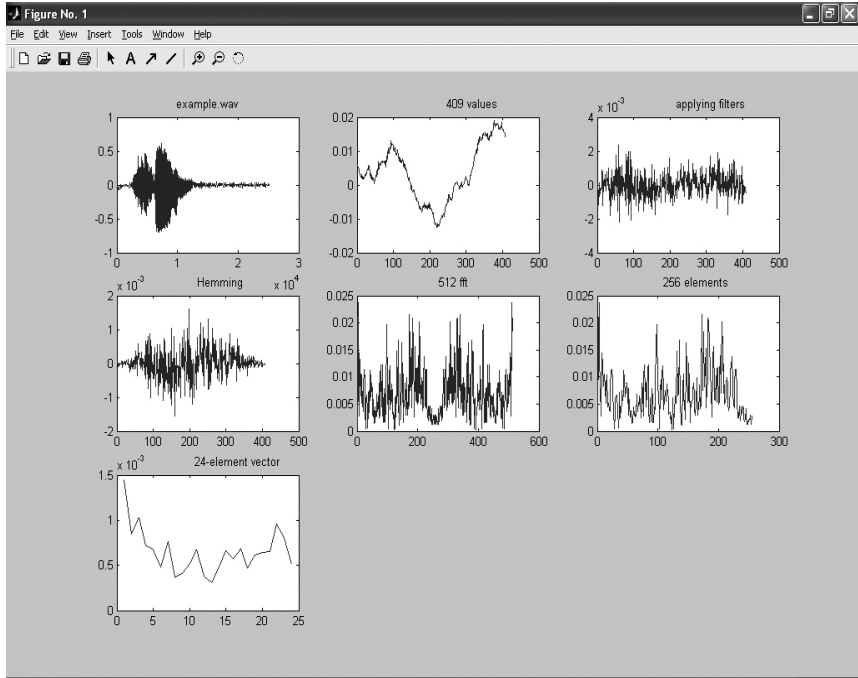


Рис. 6.3. Етапи попереднього опрацювання мовного сигналу.

Сьома ілюстрація дає значення 24-елементного вектора, компоненти якого отримані після усереднення 256 амплітудних значень у межах 24 “трикутних” частотних смуг.

6.4. Алгоритм динамічного часового вирівнювання для розпізнавання слів із невеликого словника

На фазі навчання як мовні еталони записуємо якнайкоротше вимовлені диктором слова із заданого невеликого словника.

Сигнал, який розпізнаємо, та сигнали-еталони параметризуємо – перетворюємо в послідовність спектральних векторів. Для цього розбиваємо сигнал на послідовні інтервали по 25,6 мс з перекриттям на 10 мс. У результаті

попереднього опрацювання кожного з інтервалів отримуємо вектор з 24 спектральних значень. Кроки, які слід виконати для попереднього опрацювання кожного інтервалу мовного сигналу, детально описані в підрозділі 6.3.

Кількість отриманих у результаті попереднього опрацювання спектральних векторів може бути різною. Вона залежить від тривалості в часі мовного сигналу, який опрацьовуємо.

До послідовності векторів для кожного з еталонів додаємо один вектор зі всіма нульовими компоненти на початку послідовності й такий же ж один вектор у кінці послідовності. Потреба в додаванні таких векторів полягає в необхідності моделювати паузи довільної тривалості на початку й в кінці еталону. Адже записаний сигнал, який розпізнаємо, завжди має такі паузи, і спроба автоматично усунути ці паузи не дає задовільного результату.

вектори сигналу	a_1	d_{11}	d_{12}	...	d_{1J}	
	
	a_2	d_{21}	d_{22}	...	d_{2J}	
	a_I	d_{I1}	d_{I2}	...	d_{IJ}	
		b_1	b_2	...	b_J	вектори еталону

Рис. 6.4. Матриця відстаней для сигналу й одного еталону.

На фазі розпізнавання будуємо матриці відстаней між векторами того сигналу, який хочемо розпізнати, і векторами кожного з сигналів-еталонів [11, 18, 27]. Побудова матриці відстаней для сигналу, який розпізнаємо, й одного сигналу-еталону проілюстровано на рис. 6.4.

Наведені в матриці відстані d_{ij} між векторами сигналу й векторами еталону – це евклідові відстані між відповідними векторами. Їх значення задають так:

$$d_{ij} = d(\vec{a}_i, \vec{b}_j) = \sqrt{\sum_{l=1}^{24} [a_i^{(l)} - b_j^{(l)}]^2}, \quad i=1,2,\dots,I; j=1,2,\dots,J.$$

Базуючись на отриманій матриці відстаней, будуємо матрицю (граф) шляхів. Побудова матриці шляхів для сигналу й одного еталону наведена на рис. 6.5.

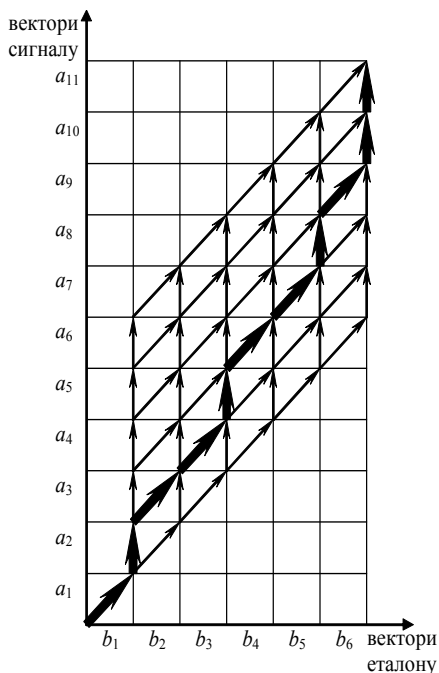


Рис. 6.5. Матриця шляхів для сигналу й одного еталону.

Мета побудови матриці шляхів – встановити найкращу відповідність між векторами сигналу й векторами еталону. Кожен вектор еталону можемо повторити, а можемо перейти до наступного вектора. Але повторювати вектор еталону два рази підряд забороняється. Винятками є початковий та кінцевий вектори еталону, повторення яких імітують паузи довільної тривалості на початку й укінці сигналу. Їх дозволяється повторити будь-яке число разів. По суті, з основних еталонів будуємо похідні еталони за рахунок повторення векторів в основних еталонах.

Такий підхід до встановлення відповідності між векторами сигналу й векторами еталону визначається відповідною моделлю утворення слів, наведеною

на рис. 6.6. Кружечки на рисунку зображають вектори еталону. У цій моделі дозволяється повторити вектор еталону (петля біля кружечка) або перейти до наступного вектора еталону (стрілка до чергового кружечка).

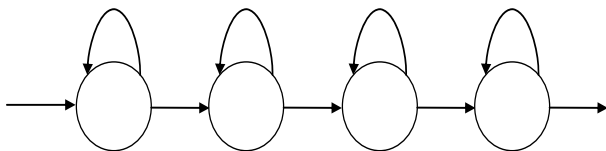


Рис. 6.6. Модель утворення слова.

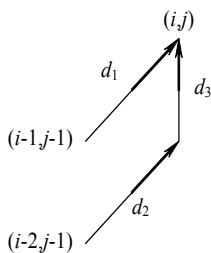
Виходячи з цієї моделі, будемо шляхи, які складаються з діагональних або вертикальних відрізків (див. рис. 6.5). Діагональний відрізок означає повторення вектора еталону, а вертикальний – повторення вектора еталону. Вибір певного шляху полягає у встановленні якоїсь відповідності між векторами сигналу, який хочемо розпізнати, і векторами чергового сигналу-еталону, до якого приміряємо наш сигнал. Стосовно наведеного в матриці шляхів конкретного шляху ця відповідність виглядає так:

- вектору сигналу a_1 ставимо у відповідність вектор b_1 еталону (відстань між цими векторами дорівнює d_{11} , її беремо з раніше підготовленої матриці відстаней);
- вектору сигналу a_2 ставимо у відповідність вектор b_1 еталону;
- вектору сигналу a_3 ставимо у відповідність вектор b_2 еталону;
- вектору сигналу a_4 ставимо у відповідність вектор b_3 еталону;
- вектору сигналу a_5 ставимо у відповідність вектор b_3 еталону;
- вектору сигналу a_6 ставимо у відповідність вектор b_4 еталону;
- вектору сигналу a_7 ставимо у відповідність вектор b_5 еталону;
- вектору сигналу a_8 ставимо у відповідність вектор b_5 еталону;
- вектору сигналу a_9 ставимо у відповідність вектор b_6 еталону;
- вектору сигналу a_{10} ставимо у відповідність вектор b_6 еталону;
- вектору сигналу a_{11} ставимо у відповідність вектор b_6 еталону.

За такими правилами будуюмо всі можливі шляхи. Для кожного шляху підраховуємо сумарну відстань як суму евклідових відстаней між парами векторів, пов'язаних з відрізками шляху.

Знаходимо в матриці шлях із найменшою сумарною відстанню. Сумарна відстань вздовж цього шляху є відстанню між сигналом і даним еталоном. Для якого з усіх можливих етalonів відстань буде мінімальною, тому еталону, ми вважаємо, відповідає невідомий сигнал.

Формально рух по матриці шляхів описуємо наступним чином (далі наведено первинний фрагмент матриці):



Найменша сумарна відстань $S(i, j)$ від точки $(0, 0)$ до точки (i, j) дорівнює

$$S(i, j) = \min \begin{cases} S(i-1, j-1) + d_1 \\ S(i-2, j-1) + d_2 + d_3 \end{cases}$$

де $S(i-1, j-1)$ – найменша сумарна відстань від точки $(0, 0)$ до точки $(i-1, j-1)$;

$S(i-2, j-1)$ – найменша сумарна відстань від точки $(0, 0)$ до точки $(i-2, j-1)$;

d_1 – відстань між векторами, які відповідають відрізку, що з'єднує точки $(i-1, j-1)$ та (i, j) ;

d_2 – відстань між векторами, які відповідають відрізку, що з'єднує точки $(i-2, j-1)$ та $(i-1, j)$;

d_3 – відстань між векторами, які відповідають відрізку, що з'єднує точки $(i-1, j)$ та (i, j) ;

Для отримання наведеного виразу застосовуємо так званий принцип оптимальності Белмана. Згідно із цим принципом (основним принципом динамічного програмування) будь-який підшлях оптимального шляху має бути

оптимальним. Виходячи з вказаного виразу всі шляхи будуються рекурсивно (зліва вправо) та одночасно. У цьому разі кращі (в сенсі меншої сумарної відстані) шляхи залишаються, а гірші відкидаються.

Модель утворення слова можна ускладнити: деякі вектори еталону дозволяється перестрибувати (пропускати, але не більше одного), як показано на рис. 6.7.

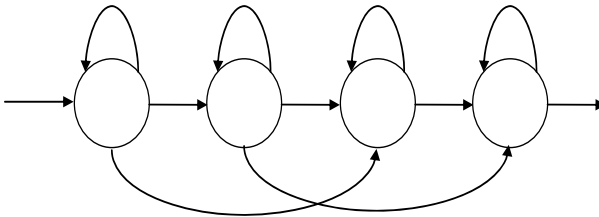


Рис. 6.7. Ускладнена модель утворення слова.

Тоді шляхи складатимуться з діагональних, вертикальних або горизонтальних відрізків. Горизонтальний відрізок означає пропускання вектора еталону. Перевага такого варіанту: при записування мовних сигналів-еталонів немає необхідності вимовляти їх якнайкоротше.

Отримавши матриці відстаней, будемо матрицю (граф) шляхів користуючись викладеними далі практичними прийомами.

Спочатку значення всіх елементів матриці шляхів покладемо рівними

$$S(u,v)=10000, u =1,.., time; v =1,.., time(k),$$

де $time$ – кількість спектральних векторів у мовному сигналі, що розпізнаємо, $time(k)$ - кількість спектральних векторів у k -му мовному сигналі-еталоні ($k=1, 2, \dots, chyslo$); $chyslo$ – кількість еталонів.

Значення 10000 є практичним заміником теоретичного нескінченно великого додатнього значення. З точки зору побудови шляху з мінімальним значенням сумарної відстані вздовж цього шляху, такі значення є нейтральними.

Далі обраховуємо $t_{delta} = time - time(k)$. Обчислене значення є більшим від нуля, бо всі еталони вимовляються коротко, і тому їх довжина завжди менша за довжину сигналу: $time > time(k)$ для $k = 1, 2, \dots, chyslo$.

Значення відстані $S(1,1)$ в точці $(1,1)$ матриці шляхів береться рівним відстані між першим вектором сигналу й першим вектором зразка: $S(1,1) = d(1,1)$. Оскільки попереднього вектора в еталоні немає, то немає що повторювати, і це єдиний можливий варіант.

При $j = 2, \dots, time(k)$ беремо значення діагональних елементів матриці шляхів рівними $S(j,j) = S(j-1,j-1) + d(j,j)$. Дійсно, оскільки рух по горизонталі заборонений, то дістатися до діагонального елемента (j,j) можна лише від діагонального елемента на місці $(j-1,j-1)$.

Для $j = 2, \dots, t_{delta} + 1$ беремо значення елементів першого стовпця матриці шляхів рівними $S(i,1) = S(i-1,1) + d(i,1)$. Справді, оскільки перед першим стовпцем немає стовпців (бо він перший), дійти до елемента $(i,1)$ першого стовпця можна лише з елемента $(i-1,1)$ першого стовпця. Як бачимо, у першому стовпці не забороняється повторення першого вектора еталону більше, ніж один раз. Зрозуміло, чому так. Адже перший вектор еталону – це вектор паузи.

Від другого до передостаннього стовпця ($j = 2, \dots, time(k) - 1$) матриці шляхів виконуємо такі дії.

Починаючи з елемента відповідного стовпця вище від діагоналі і до елемента в рядку $t_{delta} + j$ виконуємо обчислення сумарної мінімальної відстані. У цьому разі до елемента (i,j) можна дістатися двома можливими шляхами. Перший можливий шлях – від елемента $(i-1,j-1)$ по діагоналі до елемента (i,j) . У цьому разі до сумарної відстані $S(i-1,j-1)$ в точці $(i-1,j-1)$ додаємо відстань $d(i,j)$ і отримуємо число c_1 . Другий можливий шлях розбивається на два кроки. Спочатку від елемента $(i-2,j-1)$ рухаємось по діагоналі до елемента $(i-1,j)$. Потім від елемента $(i-1,j)$ рухаємось по вертикалі до елемента (i,j) .

Під час першого кроку до сумарної відстані $S(i-2,j-1)$ в точці $(i-2,j-1)$ додаємо відстань $d(i-1,j)$. Під час другого кроку до отриманої на попередньому кроці відстані додаємо $d(i,j)$. У результаті отримуємо: $c_2 = S(i-2,j-1) + d(i-1,j) +$

$d(i,j)$. Значення матриці шляхів у точці (i,j) беремо рівним меншому з чисел c_1, c_2 : $S(i,j)=\min(c_1, c_2)$. Вектор еталону від третього до передостаннього стовпця не можна повторювати більше одного разу підряд.

Рух по останньому стовпцю матриці шляхів є специфічним. Дозволяється прийти до елемента в цьому стовпці по діагоналі або по вертикалі. Тобто, в кінці матриці шляхів дозволяється багатократно повторювати останній вектор еталону, який є вектором паузи.

Алгоритм розпізнавання слів із невеликого словника з використанням динамічного часового вирівнювання включає в себе такі блоки.

Введення диктором мовних зразків із вибраного словника. У цьому блоці диктор вводить мовні зразки тих слів української мови, які потім система розпізнавання зможе розпізнати. Тобто, тут задаються можливі варіанти слів, з яких система буде вибирати при розпізнаванні.

Попереднє опрацювання k -го сигналу-еталону. Виконується згідно з описом, наведеним раніше.

Додавання звукових векторів паузи до і після звукових векторів сигналу-еталона. Мовний сигнал, що буде розпізнаватися, фізично неможливо записати без пауз перед і після команди. Разом з тим відомі прийоми автоматичного викидання цих пауз не дають задовільних результатів. Тому для реалізації розпізнавання мовних сигналів з паузами до отриманих сигналів-еталонів спочатку й в кінці додаємо по одному вектору, які моделюють нам паузи. Усі значення цих векторів дорівнюють нулю.

Запис мовного сигналу в звуковому редакторі. Використовуємо вікно звукового редактора для запису сказаного на мікрофон мовного сигналу, що потім повинен розпізнаватися. Після запису мовного сигналу з відповідною частотою дискретизації та розрядністю без усякого редакування він дається на розпізнавання.

Попереднє опрацювання мовного сигналу, що буде розпізнаватися: формування звукових векторів. Виконується згідно з описом, наведеним раніше.

Побудова матриці відстаней між векторами сигналу та k -го зразка. Обчислюються евклідові відстані між усерединними парами векторів, де перший елемент пари – це довільний вектор сигналу, що розпізнається, а другий елемент пари – це довільний вектор k -го сигналу-еталона. Розмір матриці відстаней дорівнює $time \times time(k)$, де $time$ – кількість векторів мовної команди, що розпізнається, а $time(k)$ – кількість векторів k -го мовного сигналу-еталона.

Задання початкових умов для матриці шляхів. Усі елементи матриці шляхів на початку покладаються рівними достатньо великому числу 10000. Оскільки подальші кроки пов'язані із знаходженням мінімумів, то такі присвоєння не вплинуть на отримання правильного результату.

Обчислення діагональних елементів матриці шляхів. Діагональні елементи матриці шляхів обчислюються за правилом, що відрізняється від правила обчислення решта елементів матриці, як описано раніше.

Обчислення першого стовпця матриці шляхів. Елементи першого стовпця матриці шляхів обчислюються за правилом, що відрізняється від правила обчислення решта елементів матриці шляхів і від правила обчислення діагональних елементів. У цьому разі дозволяється багатократно повторювати перший вектор сигналу-еталона (він відповідає паузі). Це також описано раніше.

Обчислення решти стовпців матриці шляхів. Елементи від третього до передостаннього включно стовпців матриці шляхів обчислюються з урахуванням двох можливих підшляхів на матриці й вибору коротшого (в сенсі евклідової відстані) з них. Такі варіанти описані на початку цього підрозділу.

Окреме обчислення останнього стовпця. Елементи останнього стовпця матриці шляхів обчислюються за правилом, що відрізняється від правила обчислення решта елементів матриці шляхів і від правила обчислення діагональних елементів. У цьому разі дозволяється багатократно повторювати останній вектор сигналу-еталона (він відповідає паузі). Це також описано раніше.

Отримання відстані між сигналом і k-м зразком. Користуючись збудованою матрицею шляхів, знаходимо відстань між сигналом, що розпізнається і k-м сигналом-еталоном.

Визначення відповіді системи розпізнавання. Порівнюються знайдені відстані між мовним сигналом, що розпізнається, і всіма сигналами-еталонами. Найменша з цих відстаней визначає відповідь системи розпізнавання. На екран виводиться відповідне слово із визначеного на фазі навчання словника.

Для моделювання цього алгоритму (його реалізації та виконання експериментів для з'ясування точності розпізнавання) було використано середовище MATLAB. Для роботи із звуками (у даному випадку для запису та редагування мовних сигналів-еталонів та довільних мовних сигналів) використовували професійний звуковий редактор.

Виконувалися експерименти з розпізнавання з використанням алгоритму динамічного часового вирівнювання 10 ізольованих слів (назви цифр від 0 до 9: ноль, один, два, три, чотири, п'ять, шість, сім, вісім, дев'ять). Коротко (як тільки можливо) вимовлені ці слова слугували еталонами для розпізнавання. Приклади отриманих після навчання на конкретного диктора сигналів-еталонів наведені на рис. 6.8-6.10. Для цих еталонів виконано попереднє опрацювання. Кількість акустичних векторів у цих еталонах наведена в табл. 6.2.

Далі вимовлялися вказані слова в довільному порядку. Кількість акустичних векторів у мовних сигналах, які фігурували в експериментах, наведена в табл. 6.3.

Сигнал на рис. 6.8 записаний в звуковому професійному редакторі. У цьому разі слово “ноль” вимовлялося диктором якомога коротше (у швидкому темпі). Частота дискретизації сигналу-еталона – 16 КГц. Розрядність сигналу-еталона – 16 розрядів.

Таблиця 6.2. Кількість звукових векторів для еталонів слів

Слово	Кількість векторів
ноль	22

один	38
два	22
три	22
чотири	44
п'ять	33
шість	46
сім	38
вісім	39
дев'ять	46

Таблиця. 6.3. Кількість звукових векторів для мовних сигналів слів

Слово	Кількість векторів
ноль	30-40
один	48-58
два	29-39
три	28-38
чотири	55-65
п'ять	38-48
шість	55-65
сім	52-62
вісім	53-63
дев'ять	55-65

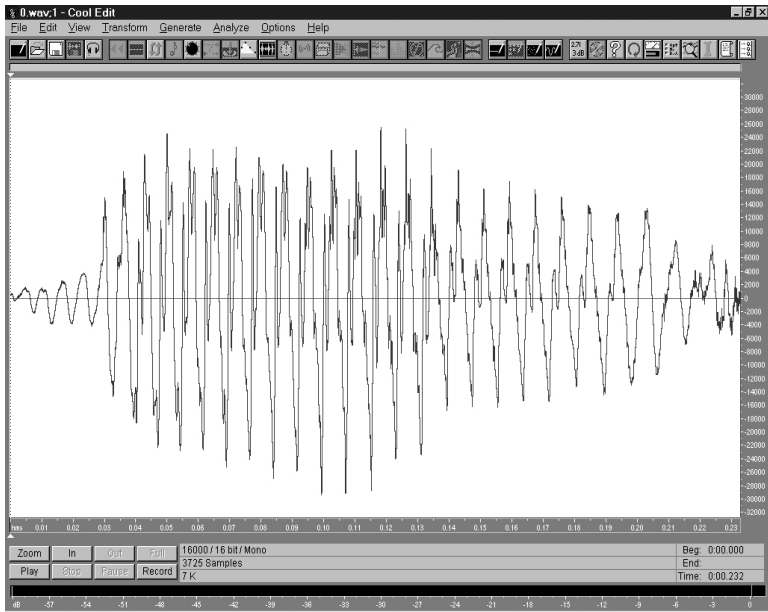


Рис. 6.8. Сигнал-эталон слова “ноль” (3725 відліків).

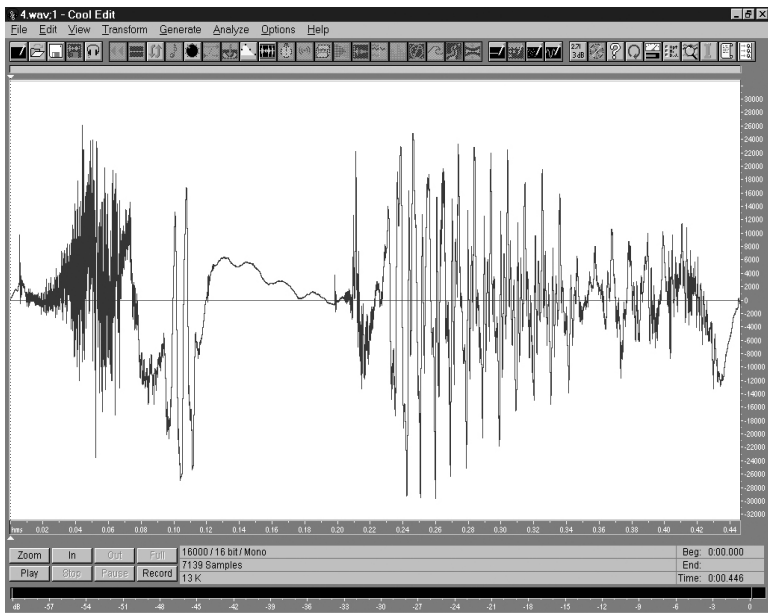


Рис. 6.9. Сигнал-эталон слова “чотири” (7139 відліків).

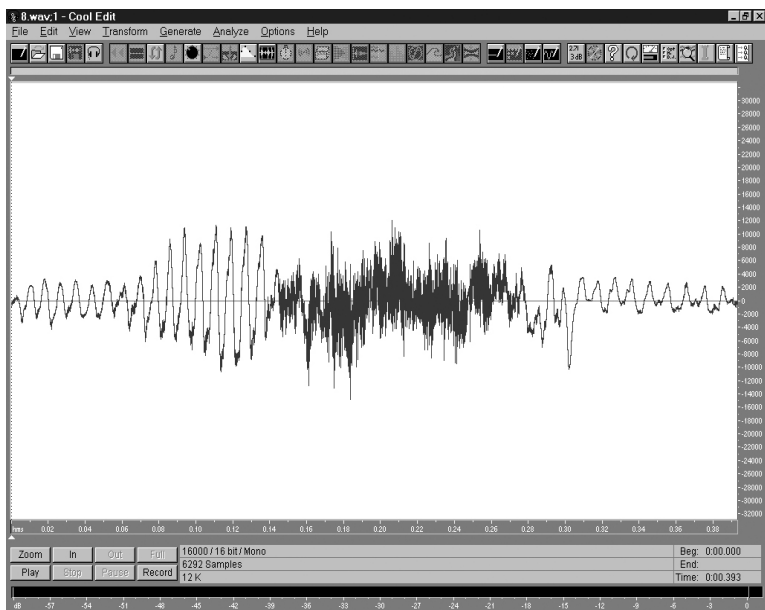


Рис. 6.10. Сигнал-еталон слова “вісім” (6292 відліків)

Після того, як сигнал був записаний, у ньому вручну на початку і вкінці викинуто паузи. Ці паузи обов’язково присутні в мові людини. З точки зору проведення подальшого розпізнавання їх слід викинути. Кількість відліків для цього сигналу-еталона дорівнює 3725 відліків.

Приклад сигналу, який розпізнавали, наведено на рис. 6.11. Як бачимо, на початку і в кінці цього реального сигналу, який розпізнається, є паузи. Пауза на початку сигналу рівна приблизно 0,3 с, а пауза вкінці сигналу рівна наближено 0,25 с. Система розпізнавання дала відповідь, що сигнал, який розпізнавався, це слово “вісім”. Можна порівняти рис. 6.11 з рис. 6.10 (сигнал-еталон слова “вісім”). Кількість відліків у сигналі суттєво більша порівняно з сигналом-еталоном і дорівнює 19538 відліків.

Зауважимо, що слова вимовлялися в різному темпі та з різною гучністю. Отримано практично 100% точність розпізнавання. Потім був змінений диктор: також отримано практично 100% точність розпізнавання. Отже, система розпізнавання з невеликим словником та для одного диктора працює добре.

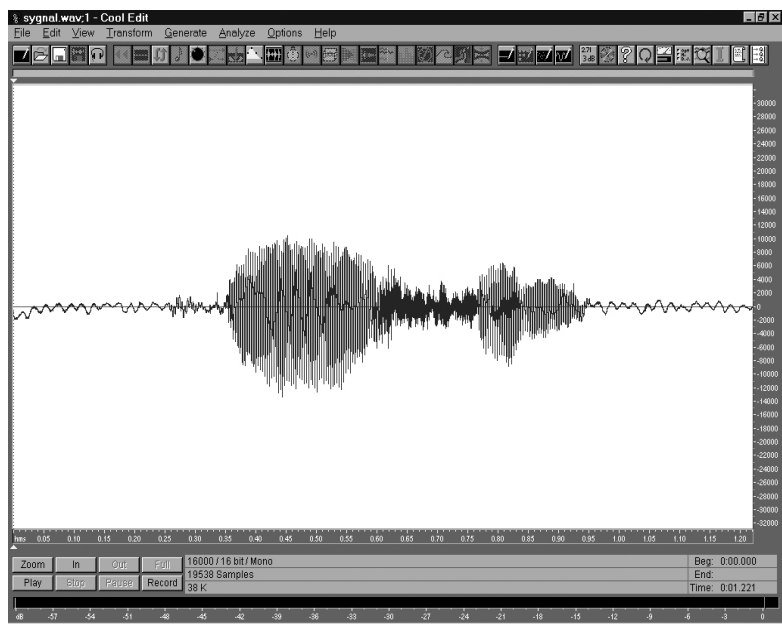


Рис. 6.11. Сигнал, що розпізнавався.

Якщо брати зразки від одного диктора, а потім намагатися розпізнавати слова іншого диктора, то система починає робити помилки. Точність розпізнавання падає до 75-85%. Така точність уже є неприйнятною для більшості практичних застосувань. Щоб покращити точність розпізнавання у цьому разі треба було записувати зразки від кількох дикторів і потім всі отримані зразки використовувати при розпізнаванні.

Інша проблема, яка виникає при розпізнаванні, це забезпечення великого словника при розпізнаванні, поступаючись при цьому точністю розпізнавання. Зрозуміло, що буквально записати велику кількість еталонів практично нереально. У цьому разі слід використовувати інший підхід, який розглянемо далі.

6.5. Розпізнавання злитної мови з великим словником

Сучасні системи для розпізнавання злитної мови з великим словником ґрунтуються на принципах статистичного розпізнавання образів [25, 31]. На першому етапі мовний сигнал перетворюється звуковий препроцесором на послідовність спектральних векторів $Y=y_1, y_2, \dots, y_T$, як описано раніше. Кожен вектор є стислим поданням короткочасного мовного спектру на інтервалі, як правило, близько 25 мс зі зсувом інтервалів на 10 мс. Типова фраза з десяти слів по 6-7 звуків у кожному може мати тривалість біля 3 с і зображатися послідовністю з $T=300$ спектральних векторів.

У загальному, фраза складається з послідовності слів $W = w_1, w_2, \dots, w_n$. Робота системи розпізнавання полягає у визначенні найбільш імовірної послідовності слів \hat{W} , маючи мовний сигнал Y . Для цього використовується правило Байєса:

$$\hat{W} = \arg \max_W P(W | Y) = \arg \max_W \frac{P(W)P(Y|W)}{P(Y)}.$$

Рівність показує, що для знаходження найбільш правдоподібної послідовності слів W , повинна бути знайдена послідовність, що робить максимальним добуток $P(W)$ та $P(Y|W)$. Так як знаменник $P(Y)$ не залежить від W , то його при розпізнаванні ігнорують. Перший із співмножників представляє апіорну ймовірність спостереження W незалежно від спостереження мовного сигналу. Ця ймовірність визначається моделлю мови. Другий співмножник представляє ймовірність спостереження послідовності векторів Y при заданій послідовності слів W . Ця ймовірність визначається звуковою моделлю.

На рис. 6.12 показана структура типової системи статистичного розпізнавання. Вона включає чотири складових [19, 25, 31]. Розглянемо кожен з цих чотирьох складових більш детально.

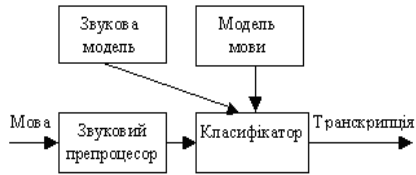


Рис. 6.12. Структура системи розпізнавання злитної мови.

Звуковий препроцесор. Потрібен початковий етап обробки, на якому з мовного сигналу вибирається вся необхідна звукова інформація в компактному вигляді. Дії, які виконуються для попереднього опрацювання мовних сигналів описані раніше.

Звукова модель. Мета звукової моделі – дати метод обчислення правдоподібності $P(Y|W)$ будь-якої послідовності звукових векторів Y при заданій послідовності слів W . У принципі потрібний розподіл імовірностей звукових векторів можна було б знайти, маючи багато зразків кожної послідовності слів та збираючи статистику відповідних послідовностей векторів. Проте це нереально для систем розпізнавання з великим словником.

Замість цього послідовності слів розбиваються на базові “будівельні” блоки (звуки), які називають фонемами. Кожна фонема зображається прихованою моделлю за Марківим (англійська назва – hidden Markov model (НММ)). НММ-модель складається з низки станів, з’єднаних дугами. НММ-модель фонем, як правило, має три породжуючих стани (рис. 6.13). Формальні вхідний і вихідний стани дозволяють об’єднувати моделі фонем, щоб утворювати слова й об’єднувати слова з метою отримати фрази.

НММ простіше зрозуміти як генератор послідовностей спектральних векторів. Це автомат зі скінченим числом станів, який змінює свій стан у кожний дискретний момент часу. Коли в момент часу t він переходить у стан j , утворюється спектральний вектор y_t з щільністю ймовірності $b_j(y_t)$. Більш того, перехід із стану i в стан j також імовірнісний і задається дискретною ймовірністю

a_{ij} . Рис. 6.13 дає приклад цього процесу, коли модель проходить послідовність станів $X=1,2,2,3,4,4,5$ і утворює послідовність векторів $Y=y_1, y_2, y_3, y_4, y_5$.

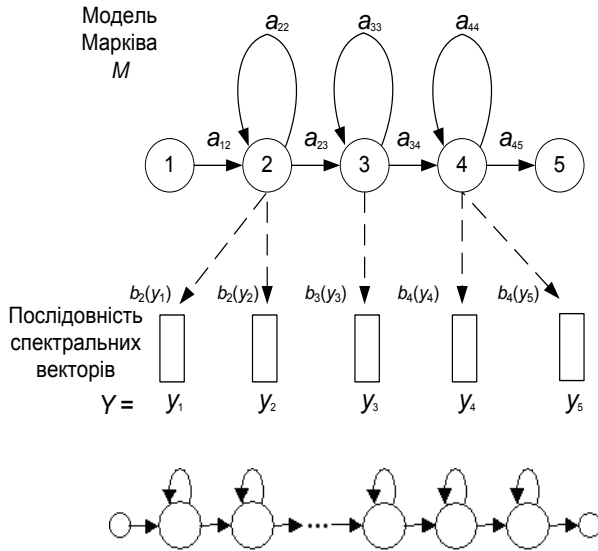


Рис. 6.13. HMM-модель фонему.

Сумісна ймовірність послідовності векторів Y та послідовності станів X при певній заданій моделі M обчислюється просто як добуток ймовірностей переходів та ймовірностей виходів. Тобто, для послідовності станів X на рис. 6.13 маємо

$$P(Y, X | M) = a_{12} \cdot b_2(y_1) \cdot a_{22} \cdot b_2(y_2) \cdot a_{23} \cdot b_3(y_3) \cdot a_{34} \cdot b_4(y_4) \cdot a_{44} \cdot b_4(y_5) \cdot a_{45}.$$

Більш загально, сумісна ймовірність послідовності векторів Y та деякої послідовності станів $X=x(0), x(1), x(2), \dots, x(T), x(T+1)$ дорівнює

$$P(Y, X | M) = a_{x(0)x(1)} \prod_{t=1}^T b_{x(t)}(y_t) a_{x(t)x(t+1)},$$

де $x(0)$ вхідний, а $x(T+1)$ вихідний стан моделі.

На практиці, звичайно, відома лише послідовність Y , а послідовність X невідома. Ось звідки взялася назва прихована за Марківим модель. Проте, потрібну ймовірність $P(Y|M)$ легко знаходимо додаючи величини $P(Y, X|M)$ для всіх можливих послідовностей станів X . Розроблено ефективний рекурсивний

метод виконання цих обчислень, який називається алгоритмом руху вперед-назад. Суттєвою особливістю цього алгоритму є те, що він дозволяє обчислити ймовірність перебування в заданому стані моделі в заданий момент часу. З нього випливає дуже проста й ефективна процедура Баума-Велча знаходження за критерієм максимальної правдоподібності оцінок параметрів a та b моделі. Слід зауважити, що поява процедури Баума-Велча стала ключовим фактором домінування НММ в звуковій моделі.

Вважається, що наведене рівняння треба переписати в логарифмічній формі та розділити члени a і b :

$$\log P(Y, X | M) = \sum_{t=0}^T \log a_{x(t)x(t+1)} + \sum_{t=1}^T \log b_{x(t)}(y_t).$$

Ймовірності переходу $a_{x(t)x(t+1)}$ моделюють темпоральну структуру даних. Кожен доданок у першій сумі може розглядатися як ціна руху з одного стану в інший. У цьому разі маємо дуже бідну модель зміни реальної мови, але це не є критичним, оскільки на практиці в наведеному виразі домінують ймовірності $b_{x(t)}(y_t)$. Кожен стан НММ забезпечує спектральний вектор-прототип, а функція логарифму ймовірності виходу дає метрику відстані і, значить, можливість порівняти поточні спектральні вектори з прототипом.

Вибір функції вихідного розподілу є критичним, бо вона мусить моделювати притаманні спектральні варіації в реальній мові диктора. Ранні НММ системи використовували дискретні функції розподілу. Такий підхід дуже ефективний з обчислювальної точки зору, але квантування вносить шум, що обмежує точність розпізнавання, яку можна отримати. Тому сучасні системи використовують неперервні параметричні розподіли вихідних щільностей, які прямо моделюють спектральні вектори. У найбільш загальному випадку розподіл беруть як лінійну комбінацію багатовимірних розподілів Гауса

$$b_j(y_t) = \sum_{v=1}^V c_{jv} G(y_t, \mu_{jv}, \Sigma_{jv}),$$

де c_{jv} – вага v -го доданка для j -го стану; $G(y_t, \mu_{jv}, \Sigma_{jv})$ позначає багатовимірний розподіл Гауса з середнім вектором μ_{jv} , та коваріаційною матрицею Σ_{jv} рівний

$$G(y_t, \mu_{jv}, \Sigma_{jv}) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_{jv}|}} e^{-\frac{1}{2}(y_t - \mu_{jv})\Sigma_{jv}^{-1}(y_t - \mu_{jv})^T},$$

де n означає розмірність спектральних векторів, $|\Sigma_{jv}|$ – визначник матриці, T – транспонування вектора. Приблизно 10 компонент лінійної комбінації дають добрі показники для системи розпізнавання.

На ранній стадії досліджень вважалося, що вимагається лише одна НММ для кожної фонем. Число фонем в українській мові рівне 38. Здавалось би, потрібно здійснити навчання лише 38 НММ-моделей. На практиці, проте, контекстні ефекти спричиняють значні зміни у способі утворення звуків (так зване явище коартикуляції). Тому, щоб досягти доброго фонетичного розрізнення, треба навчати різні НММ для різних контекстів.

Найбільш загальним є підхід з використанням трифонів, коли кожна фонема має окрему НММ-модель для кожної індивідуальної пари сусідів зліва та справа. В останніх публікаціях згадується і про використання квінфонів (сукупностей 5 сусідніх звуків: два сусіди зліва й два сусіди справа).

Наприклад, нехай позначення x - y + z представляє фонему y , що трапилась після x і перед z . Тоді фраза “Цей комп’ютер” подається послідовністю фонем \cup ц е й к о м п й у т е р \cup , де \cup позначає паузу. Якщо використовуються НММ-моделі трифонів, то фраза буде описуватися так:

\cup -ц+е ц-е+й е-й+к й-к+о к-о+м о-м+п м-п+й п-й+у й-у+т у-т+е т-е+р е-р+ \cup .

При 38 фонемах є $38^3=54872$ можливих трифони, проте не всі трапляються через обмеження української мови. Загальне число трифонів, необхідних для практичного вжитку, залежить від вибраної множини фонем, словника та граматичних обмежень.

Перед тим, як НММ може бути застосована, повинні бути визначені її параметри. Цей процес називають навчанням [25, 31]. Він вимагає три елементи.

1. Навчальну базу даних, у якій є мовні записи та відповідні їм тексти.

Для англійської мови існує багато загальнодоступних баз даних мовних зразків (ISOLET, CONNEX, Resource Management Database, Wall Street Journal Database та інші). Створення таких баз даних для української мови є завданням на часі. Як для української, так і для англійської мов, завданням є якнайкраще розбиття мовного запису на фонему відповідно до тексту.

2 Цільову функцію, яка разом з навчальною базою даних може бути використана, щоб поміряти “відповідність” НММ. Найбільш широко вживаною є цільова функція максимальної правдоподібності:

$$\log P_{\lambda}(O) = \sum_{u=1}^U \log P_{\lambda_u}(O_u | W_u),$$

де O – повна множина спектральних векторів усіх навчальних речень;

O_u – спектральні вектори u -го навчального речення;

W_u – синтезована НММ-модель для u -го навчального речення

λ – множина параметрів НММ-моделей усіх навчальних речень;

λ_u – множина параметрів НММ-моделі для u -го навчального речення.

3. Процедуру оптимізації, яку можемо використати, щоб максимізувати цільову функцію.

При цьому найчастіше використовують рекурсивний у часі алгоритм Баума-Велча для повторного оцінювання (перерахунку) параметрів моделей. Теоретично доведено, що при такому перерахунку значення максимальної правдоподібності навчальних даних є неспадною величиною, тобто збільшується або залишається таким самим.

Таким чином, навчання звукової моделі зводиться до послідовним ітерацій, на кожній з яких перераховуємо всі параметри НММ-моделей усіх навчальних речень. Ітерації припиняємо, коли різниця двох послідовних значень цільової функції стає меншою від наперед заданої величини.

Звідки взяти початкові значення (до початку навчання) параметрів моделей усіх навчальних речень? Один можливий варіант: вручну розбити невелику кількість навчальних речень на окремі фонему та їх стани, і за рахунок цього

отримати певні початкові оцінки параметрів (виходячи з 38 різних звуків). Інший, ширше вживаний сьогодні варіант, так званого плоского старту: взяти як початкові значення параметрів будь-які теоретично можливі їх значення.

Ключовим у використанні звукової моделі є зменшення числа параметрів, які треба оцінити. Це необхідно, бо надто велике число оцінюваних параметрів приводить при обмежених навчальних даних до нереальних оцінок. Крім того, зменшення числа параметрів зменшує обчислювальну складність.

Ця проблема настільки серйозна, що використовуються додаткові евристичні обмеження:

1. Розподіли ймовірностей появи звукових векторів для різних станів моделей можуть бути зв'язані, тобто користуватися тими самими параметрами. Звичайно це корисно лише тоді, коли вони представляють подібні звукові ситуації.
2. Коваріаційна матриця розподілів припускається діагональною.
3. Число розподілів Гаусса, сума яких моделює розподіл імовірностей для стану, може змінюватися, щоб досягти найкращого балансу між гнучкістю моделювання і складністю.

Ідея полягає в тому, щоб зв'язати стани, які акустично не відрізняються. Це дозволяє всі дані, які відповідають кожному індивідуальному станові, об'єднати і за допомогою цього дати більш робастні оцінки параметрів зв'язаного стану. Після зв'язування ряд станів використовують один і той самий розподіл.

Наприклад, трифон м-о+р в слові море та трифон н-о+л в слові ноль можна вважати такими, що описують подібні звукові ситуації: як у першому, так і в другому випадку середній звук о має сусіда зліва квазіперіодичний звук (м або н) та сусіда справа квазіперіодичний звук (р або л).

Вибір того, які стани зв'язувати, здійснюється за допомогою фонетичних вирішуючих дерев. Це передбачає побудову бінарного дерева для кожного стану кожної фонемі. У кожному вузлі такого дерева ставиться питання, на яке треба відповісти "так" або "ні". Питання відносяться до фонетичного оточення (контексту) безпосередньо зліва і справа. Одне дерево будується для кожного стану кожної фонемі, щоб розбити на підмножини всі відповідні стани всіх

відповідних трифонів. У результаті такого зв'язування замість $38^3=54872$ трифонів отримуємо 2000-8000 зв'язаних трифонів.

На етапі підготовки навчальних даних для звукової моделі диктор вільно начитує запропонований текст в мікрофон, і в результаті маємо відповідний цьому тексту (множині речень) мовний сигнал.

Модель мови. Метою моделі мови є дати метод обчислення апіорної ймовірності послідовності слів незалежно від спостереження мовного сигналу. Для цього треба забезпечити механізм оцінки ймовірності певного слова w_k у фразі, якщо знаємо попередні слова $W_1^{k-1} = w_1 \dots w_{k-1}$.

Простий, але ефективний шлях зробити це – використати N -ки слів, у яких приймається, що дане слово w_k залежить лише від попередніх $N-1$ слів, тобто

$$P(w_k | W_1^{k-1}) = P(w_k | W_{k-N+1}^{k-1}).$$

N -ки слів одночасно мають у собі граматику, смисл і предметну область та зосереджуються на локальних залежностях. Більш того, розподіли ймовірностей для N -ок можна обчислити прямо з текстових навчальних даних. Тому не потрібно мати такі точні лінгвістичні правила, як формальна граMATика мови.

У принципі, N -ки можна оцінити простим підрахунком частоти повторюваності слова в навчальних текстах і зберігати ці дані в таблицях. Наприклад, для випадку трійок слів ($N=3$)

$$P(w_k | w_{k-1}, w_{k-2}) = \frac{t(w_k, w_{k-1}, w_{k-2})}{b(w_{k-1}, w_{k-2})},$$

де $t(w_k, w_{k-1}, w_{k-2})$ число появ трійок w_k, w_{k-1}, w_{k-2} в навчальних даних та $b(w_{k-1}, w_{k-2})$ число появ двійок слів w_{k-1}, w_{k-2} .

Проблема полягає в тому, що при словнику V слів є V^3 можливих трійок. Навіть для помірною словника в 10000 слів це дуже велике число. Тому багато трійок не з'явиться в навчальних даних, а багато інших з'явиться лише раз чи двічі. Внаслідок цього отримані для них згідно з наведеним виразом оцінки будуть нереальними. Підхід до вирішення цієї проблеми полягає в тому, що

оцінки трійок, які найчастіше з'являються, зменшуються, а отримана залишкова ймовірнісна маса розподіляється між трійками, що рідко зустрічаються.

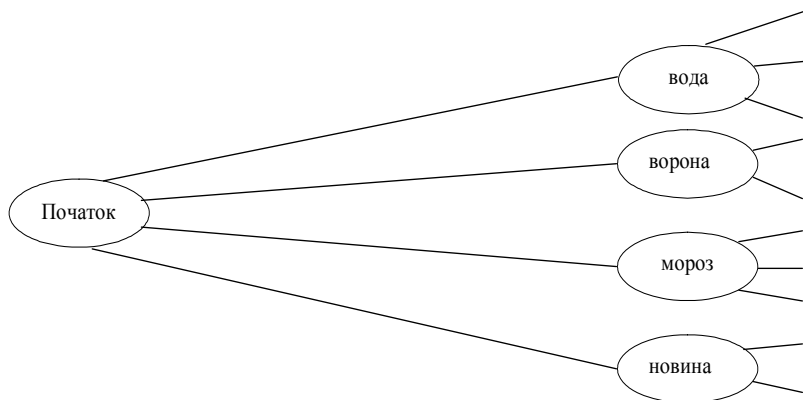
Класифікатор (декодер). Ця складова системи зводить воедино дані від трьох раніше описаних компонент і знаходить найбільш імовірний текст (транскрипцію).

Як правило, усі можливі гіпотези відслідковуються паралельно. Цей підхід спирається на принцип оптимальності Белмана (динамічного програмування) і його часто називають алгоритмом Вітербі.

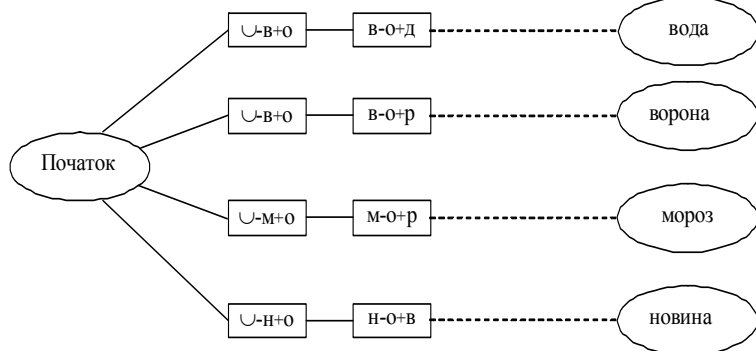
Для розуміння проблеми декодування, уявимо конструювання такого дерева, де на початку є розгалуження до будь-якого можливого початкового слова в реченні. Усі перші слова потім з'єднуються зі всіма можливими черговими словами і так далі. Це проілюстровано на рис. 6.14(а). Зрозуміло, що таке дерево буде дуже великим, але якщо брати його достатньої глибини, воно в принципі зобразило б усі можливі речення.

Замінімо кожне слово в цьому дереві послідовністю моделей згідно з його вимовою. Якщо є кілька можливих вимов, то можна дати паралельне з'єднання моделей в межах слова. Рис. 6.14(б) показує фрагмент дерева, деталізованого введенням моделей. Нарешті, зіллемо моделі однакових фонем або фонем, які за рахунок зв'язування потрапили в одну групу. Це проілюстровано на рис. 6.14(в). Отримуємо дерево з вершин НММ станів, які зв'язані ребрами переходів між станами, та вершин закінчення слів, зв'язаних ребрами переходів між словами. Кожен маршрут від початкової вершини до деякої вершини в дереві можна оцінити, додаючи всі логарифми ймовірностей переходів між станами, всі логарифми ймовірностей виходів станів та логарифми ймовірностей моделі мови. Такий шлях можна зобразити рухомою міткою, розташованою у вершині в кінці шляху. Мітка має рахунок, який є сумою логарифмів ймовірностей до даної вершини, та історію, в якій зберігається послідовність вершин закінчення слів, через які пройшла мітка. Кожен шлях можна доповнити, рухаючи мітку з поточної до сусідньої вершини та змінюючи її рахунок згідно з поточними

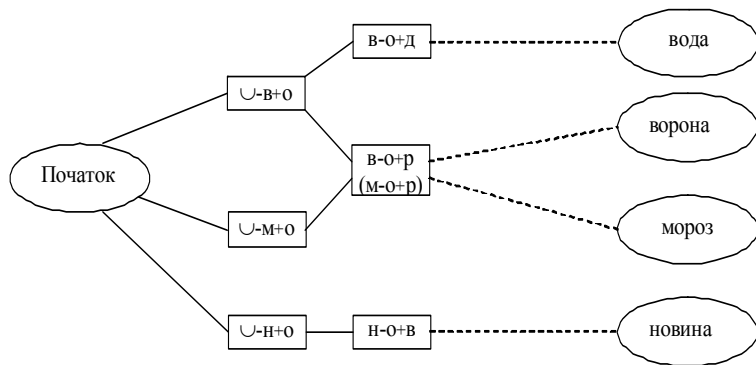
імовірністю переходу між станами, імовірністю виходу стану та імовірністю моделі мови, якщо остання має місце.



(а)



(б)



(в)

Рис. 6.14. Дерево декодування.

Проблему пошуку тепер можна переформулювати у вигляді алгоритму руху міток. На початку одну мітку ставлять у початкову вершину дерева. При появі чергового спектрального вектора кожен мітку копіюють у всі сусідні вершини і змінюють рахунки. Якщо більш ніж одна мітка опиняється у вершині, слід зберегти лише мітку з найбільшим рахунком, бо згідно з принципом оптимальності Белмана всі інші мітки не дають під шляхів оптимального шляху. Коли опрацьовані всі спектральні вектори, переглядаємо вершини закінчень слів: мітка з найбільшим рахунком дає найімовірніший маршрут, а, значить найбільш правдоподібну послідовність слів.

Цей базовий алгоритм руху міток гарантує знаходження найкращого можливого маршруту, проте, на жаль, його реалізація потребувала б занадто багато часу й пам'яті. Щоб уможливити реалізацію алгоритму, використовуємо відсікання можливих варіантів. На кожному часовому інтервалі, фіксуємо найбільший для всіх міток рахунок, і всі мітки, чий рахунок відрізняється за модулем від найбільшого рахунку більше, ніж на певну наперед встановлену величину (ширину пучка), знищуємо. Оскільки в пам'яті тепер треба зберігати лише активні мітки, які потрапляють у пучок, в кожен момент часу потрібен лише фрагмент описаного раніше дерева. Коли мітки рухаються вперед, пред ними створюємо нову структуру дерева, а стару структуру позаду знищуємо.

6.6. Завдання для самоперевірки

1. Найпростіший розпізнавач мови має точність розпізнавання:
А) 35%; Б) практично 100%; В) 50%; Г) приблизно 85%; Д) немає правильної відповіді.
2. Результатом попереднього опрацювання мовних сигналів є:
А) отримання множини відліків сигналу; Б) отримання значень дискретного косинусного перетворення сигналу; В) отримання часового опису сигналу; Г) отримання множини спектральних векторів; Д) немає правильної відповіді.

3. Мета побудови матриці шляхів у методі динамічного часового вирівнювання:

А) встановити взаємно-однозначну відповідність між векторами сигналу й векторами еталону; Б) встановити найкращу відповідність між векторами сигналу й векторами еталону; В) встановити які вектори сигналу й вектори еталону відрізняються між собою; Г) встановити які вектори сигналу й вектори еталону співпадають; Д) немає правильної відповіді.

4. Кількість зв'язаних трифонів, яку використовують у сучасних системах розпізнавання мови:

А) 2000-8000; Б) 50000-60000; В) 100000-120000; Г) 30-50; Д) немає правильної відповіді.

5. Щоб уможливити реалізацію алгоритму розпізнавання злитної мови з великим словником, використовуємо:

А) порівняння можливих варіантів; Б) побудову можливих варіантів; В) відсікання можливих варіантів; Г) перебирання можливих варіантів; Д) немає правильної відповіді.

6. Априорну ймовірність спостереження послідовності слів незалежно від спостереження мовного сигналу визначає:

А) класифікатор; Б) передпроцесор; В) модель мови; Г) звукова модель; Д) немає правильної відповіді.

7. Імовірність спостереження послідовності спектральних векторів при заданій послідовності слів визначає:

А) звукова модель; Б) модель мови; В) класифікатор; Г) перед процесор; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1) А; 2) Г; 3) Б; 4) А; 5) Б; 6) Б; 7) А.

РОЗДІЛ 7. ОБРОБКА НЕРУХОМИХ ЗОБРАЖЕНЬ

Стиск зображень (двовимірних сигналів) є одним з найважливіших прийомів при їх обробці. В результаті стиску зменшується розмір зображення, що зменшує час передачі зображення по мережі й економить простір для зберігання. Стиск зображень розділяють на стиск із втратами якості та стиск без втрат. Безвтратні методи стиску не забезпечують потрібного у багатьох випадках ступеня стиску. У цьому разі необхідно застосовувати методи стиску з втратою інформації. Останні, як правило, включають як складові частини безвтратні методи.

7.1. Стиск нерухомих зображень з використанням дискретних косинусних перетворень

Одним із найбільш поширених підходів є стиск зображень з використанням дискретних косинусних перетворень [26]. Він лежить в основі стиску зображень у форматі JPEG-9, який дозволяє стискати окремі (незмінні, still picture) зображення. Стиск у цьому форматі можна умовно розбити на три етапи: перетворення та субдискретизація кольорової інформації; поблокові дискретні косинусні перетворення; квантування та кодування значень дискретного косинусного перетворення.

Перший етап полягає в такому. Кожна точка зображення, представлена 3 байтами в системі *RGB*, переводиться в систему *YUV* (яскравість, кольорова насиченість, кольоровий тон) згідно виразів:

$$Y=77/256 \cdot R+150/256 \cdot G+29/256 \cdot B$$

$$U=131/256 \cdot R-110/256 \cdot G-21/256 \cdot B+128$$

$$V=-44/256 \cdot R-87/256 \cdot G+131/256 \cdot B+128$$

або в матричному вигляді

$$\begin{pmatrix} Y \\ U-128 \\ V-128 \end{pmatrix} = \begin{pmatrix} 0,301 & 0,586 & 0,113 \\ 0,512 & -0,430 & -0,082 \\ -0,172 & -0,340 & 0,512 \end{pmatrix} \begin{pmatrix} R \\ G \\ B \end{pmatrix}.$$

Зворотнє перетворення з системи YUV в систему RGB виконується за формулами:

$$R=Y+1,37 \cdot (U-128)$$

$$G=Y-0,698 \cdot (U-128) - 0,336 \cdot (V-128)$$

$$B=Y+1,73 \cdot (V-128)$$

або в матричному вигляді

$$\begin{pmatrix} R+175,4 \\ G-132,4 \\ B+221,4 \end{pmatrix} = \begin{pmatrix} 1 & 1,370 & 0 \\ 1 & -0,698 & -0,336 \\ 1 & 0 & 1,730 \end{pmatrix} \begin{pmatrix} Y \\ U \\ V \end{pmatrix}.$$

Далі значення компоненти Y залишаються без змін, а кількість значень компонент U і V зменшується (так звана субдискретизація: компоненти U і V можна закругити без суттєвої втрати якості зображення). Можливі різні варіанти субдискретизації: просте викидання частини з сусідніх точок чи заміна значень сусідніх точок зображення на їх середні. При цьому можливі кілька варіантів об'єднання точок: дві по горизонталі, дві по вертикалі, квадрат з чотирьох сусідніх точок. Найчастіше використовується наступний варіант: число точок зменшується вдвоє, причому значення точок обчислюються згідно з виразом $y(n) = \frac{1}{4}x(n-2) + \frac{1}{2}x(n-1) + \frac{1}{4}x(n)$. У цьому разі блок розміру 8×16 значень компоненти U або V перетворюється в блок 8×8 значень.

При відтворенні інформації для покращення якості проміжні точки рекомендується отримувати не простим повторенням, а шляхом інтерполяції між сусідніми точками. Найчастіше використовується наступний спосіб: при відтворенні зображення блок 8×8 точок реконструюється в блок 8×16 точок за формулою $x(n) = \frac{y(2n-1) + y(2n)}{2}$.

Якщо використовується описаний варіант субдискретизації, то досягається стиск зображення в 1,5 рази. Дійсно 1 байт компоненти яскравості залишається без змін, а кожні 2 байти компонент U і V замінюються на 1 байт. Отже, замість 6 байт на кожних 2 точки зображення тепер припадає 4 байти.

Коли ж використовується варіант об'єднання чотирьох сусідніх точок, то досягається стиск зображення в 2 рази. Адже 1 байт компоненти яскравості залишається без змін, а кожні 4 байти компонент U і V замінюються на 1 байт. Тобто, замість 12 байт на кожних 4 точки зображення тепер припадає 6 байт.

Мінімальний фрагмент інформації для обробки – це блок початкового RGB зображення розміру 8×16 елементів. У результаті обробки такого фрагменту на першому етапі отримуємо чотири блоки розміру 8×8 : два блоки розміру 8×8 для компоненти яскравості Y та по одному блоку розміру 8×8 для компонент U і V (рис. 7.1).

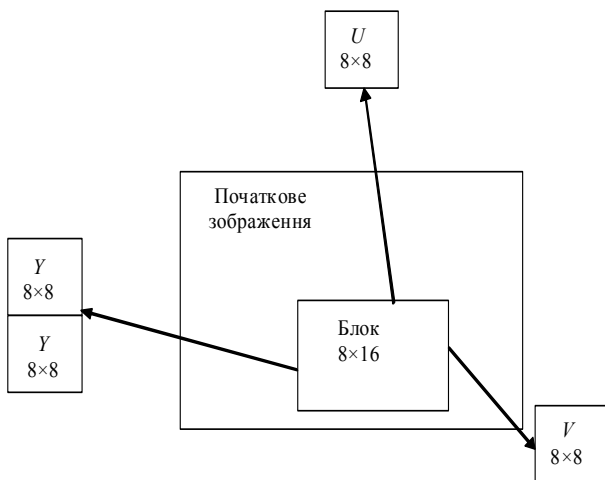


Рис. 7.1. Обробка фрагменту зображення на першому етапі.

Другий етап стиску має в своїй основі дискретне косинусне перетворення. Кожна з компонент Y, U, V зображення на цьому етапі розглядається як окреме монохромне (однокольорове) зображення та її стиск проводиться окремо. Монохромне зображення розбивається на блоки 8×8 елементів. До кожного блоку

P застосовується двовимірне ДКП $P_{\text{ДКП}}=APA^T$, де A – матриця двовимірного ДКП, $P_{\text{ДКП}}$ – матриця значень ДКП фрагменту зображення.

Двовимірне ДКП має ту властивість, що воно зосереджує найбільші за абсолютною величиною значення переважно у верхньому лівому куті матриці значень перетворення. Типовий розподіл ДКП коефіцієнтів показано в наведеній далі матриці:

1347,4	168,4	-15,5	-35,7	34,1	20,8	23,6	0
223,9	-0,4	-126,1	99,3	-14,5	18,2	0,5	-0,2
48,4	-134,8	89,4	-12,2	18,9	0,1	0,2	-0,1
-11,3	21,8	11,9	-0,8	24,5	-35,8	0,2	-0,2
0,1	0,5	-0,5	-0,1	-48,1	0,5	-0,8	-0,1
0,4	-0,1	0,5	0,2	1,1	-0,3	-0,1	-0,2
-0,4	-0,1	0,7	0	-0,4	0,1	0,8	0,5
-0,2	-0,2	0,4	-0,2	-0,4	-0,3	0,4	0

Ці значення треба взяти більш точно, а решта значень ДКП можна суттєво закругити. Це і є основою стиску зображень з використанням дискретних косинусних перетворень. Дії, які реалізують описану ідею, відбуваються на наступному третьому етапі.

На третьому етапі стиску виконуємо квантування та кодування значень дискретного косинусного перетворення. Спочатку виконується квантування значень ДКП. Для цього формується матриця Q дільників з елементами $q(i,j)=1+(1+i+j) \cdot r$, $i,j=0,1,\dots,7$; r – параметр, який визначає степінь стиску та одночасно якість відтвореного зображення. Для компоненти Y рекомендується брати $r=2$, а для компонент U,V значення r може бути більшим. Величина $q(i,j)$ – це не що інше як крок квантування, який залежно від позиції змінюється. При русі від верхнього лівого кута до правого нижнього кута крок квантування збільшується, тобто виконується грубіше квантування. Такі значення кроку квантування узгоджуються з властивістю ДКП: найбільші за абсолютною величиною значення є переважно у верхньому лівому куті матриці значень перетворення. При $r=2$ матриця дільників має вигляд:

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{3,0}$			
$a_{3,0}$	$a_{3,0}$	$a_{3,0}$	$a_{3,0}$				
$a_{4,0}$	$a_{4,1}$	$a_{4,2}$					
$a_{5,0}$	$a_{5,1}$						
$a_{6,0}$	$a_{6,1}$						
$a_{7,0}$	$a_{7,1}$						

Рис. 7.2. Порядок проходження елементів при кодуванні

До коефіцієнтів ДКП, які знаходяться на першому місці в послідовних матрицях розміру 8×8 для кожної з компонент Y, U, V , застосовується дельта імпульсно-кодова модуляція: $B(i) = A(i) - A(i-1)$, де $i=2,3,4,\dots$. Тобто, перший у послідовності цих коефіцієнтів залишається без зміни, а кожен наступний замінюється на його різницю з попереднім. При такому стиску можна однозначно відтворити початкові значення.

До послідовності із 63 коефіцієнтів, які знаходяться на місцях крім першого в кожній матриці розміру 8×8 , застосовується описане нижче кодування серій нулів.

Кодування серій нулів полягає в такому. Підраховуємо число нульових елементів від початку послідовності до першого ненульового елемента. Це число записуємо в утворювану нову послідовність. Далі записуємо ненульовий елемент, до якого ми дійшли. Тоді знову підраховуємо число нульових елементів до чергового ненульового елемента і записуємо це число в утворювану

послідовність. Таким чином, у новій послідовності чергуються число нульових елементів до чергового ненульового елемента та черговий ненульовий елемент. З такої послідовності можна однозначно відтворити початкову послідовність.

Іншими словами, при кодуванні серій нулів отримуємо пари типу (x, y) , де x означає лічильник числа пройдених нулів, а y – чергове ненульове значення. Так, послідовність

34, 45, 7, 0, -2, -4, -14, -15, -1, 0, 2, 8, 9, 3, 2, -1, -1, 1, 0, 0, 0, 0, 0, 1, 1, 2, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, -3, -2, 0,

буде згорнута в пари

$(0,34) (0,45) (0,7) (1,-2) (0,-4) (0,-14) (0,-15) (0,-1) (1,2) (0,8) (0,9) (0,3) (0,2) (0,-1) (0,-1) (0,1) (6,1) (0,1) (0,2) (3,1) (7,-3) (0,-2)$.

Зауважимо, що в кінці послідовності після 23 нульових елементів вже немає ненульового елемента. Тому при згортанні останню пару, яка була б неповною, не запишемо. Оскільки в початковій послідовності зустрічаються досить довгі серії нулів, то описане кодування дозволяє стиснути цю послідовність.

Останнім кроком є застосування до отриманої послідовності кодування Хафмена. Вважається, що поява нульового елемента у даній послідовності є найбільш імовірною, а із збільшенням абсолютної величини елемента ймовірність його появи зменшується. Таблиця кодування може фіксуватися перед початком обробки або бути адаптивною, тобто мінятися з використанням статистики появи даних у попередніх блоках.

Результатом кодування є послідовність кодових слів коду Хафмена. Відомо, що ніякі два слова в цьому кодї не мають однакового початку, тобто слова в послідовності можуть іти підряд без маркерів розділення. Певні маркери мають бути лише для того, щоб вказати, де починається послідовність кодових слів, отриманих при обробці блоку 8×8 елементів.

При відтворенні зображення виконуються наступні дії, зворотні до дій при стиску: декодування Хафмена та декодування серій нулів; деквантування (множення на відповідні значення елементів матриці дільників); обернені двовимірні ДКП блоків 8×8 елементів; обернена кольорова субдискретизація

(відтворення проміжних точок для компонент U, V шляхом інтерполяції між сусідніми точками); перехід від системи YUV до системи RGB .

Етапи стиску та відтворення даних при використанні методу на основі дискретних косинусних перетворень показані на рис. 7.3. Перетворення YUV , поблокові двовимірні ДКП, кодування серій нулів і Хафмена та обернені до них етапи є етапами без втрат інформації. На інших етапах є втрати інформації. Приблизний ступінь стиску при використанні описаного методу – від 16:1 до 25:1 без помітної втрати якості.



Рис. 7.3. Етапи стиску та відтворення зображення.

7.2. Поняття хвилькового перетворення

Дискретне хвилькове перетворення (discrete wavelet transform, DWT) принципово відрізняється від спектральних перетворень.

На рис. 7.4 показано структуру для смугового аналізу, яка ґрунтується на використанні набору цифрових фільтрів, смуги пропускання яких примикають одна до одної, з частотними характеристиками $H_i(j\omega)$, $i=1, \dots, M$. Послідовність $y_1(n), y_2(n), \dots, y_M(n)$ маємо право прорідити за теоремою Найквіста-Котельникова. Кожен смуговий фільтр має вихід $y_i(n)$, який отримується в результаті згортки вхідних даних $x(n)$ з імпульсною характеристикою $h_i(n)$ для i -го цифрового фільтра. Якщо смуги пропускання фільтрів вибрати так, що вони покривають увесь спектр вхідного сигналу, і якщо, крім того, фільтри побудовані дуже

ретельно, то сума $y(n)$ всіх вихідних сигналів фільтрів буде доброю апроксимацією вхідного сигналу (показана пунктирними лініями на рис. 7.4).

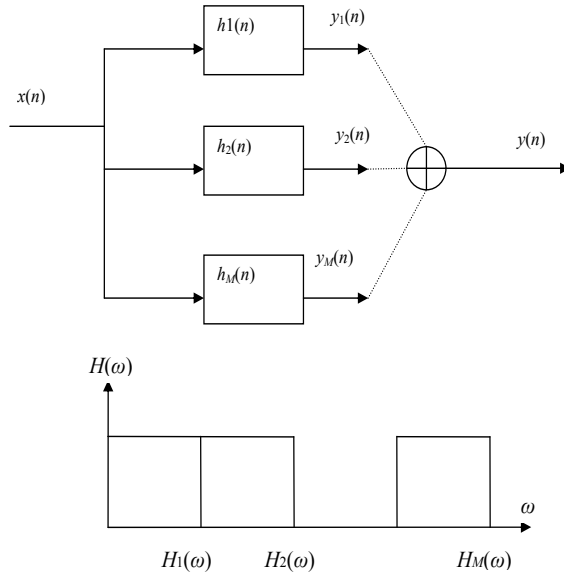


Рис. 7.4. Набір цифрових фільтрів для смугового аналізу.

Розглянуту структуру можна реалізувати, використовуючи лише один цифровий фільтр низьких частот і зсуви в частотній області спектру дискретного сигналу шляхом множення вхідних даних на відповідні експоненційні множники. Дійсно, коли до спектру $X(j\omega)$ дискретного сигналу $x(n)$ застосувати зсув частоти на величину ω_i , щоб отримати $X(j\omega - j\omega_i)$, то одержаний спектр буде відповідати новому дискретному сигналу вигляду $x_i(n) = x(n)e^{jn\omega_i T}$ (T – інтервал дискретизації, $T=2\pi/\omega_s$).

Нехай ω_M – максимальна частота сигналу (тобто аналоговий сигнал $x(t)$ має спектр, обмежений величиною ω_M). Тоді потрібно виконати зсуви частоти сигналу $\omega_i = i(\omega_M/M)$, $i=1, \dots, M-1$, щоб отримати аналіз M смуг, використовуючи один фільтр низьких частот з частотою зрізу $\omega_s = \omega_M/M$. Звідси випливає, що всі

частотні смуги початкового спектру будуть зсунуті в смугу низьких частот $[0, \omega_M/2]$.

Якщо використати в якості дискретних значень частоти величини $\omega_i = \omega_s/2$, де $\omega_s = 2\pi/T = 2\pi f_s$, то можна перевірити, що тепер співвідношення приймає дуже простий вигляд: $x_i(n) = x(n)(-1)^n$, тобто просто використовується зміна знаку, що чергується. Коли, крім того, $\omega_s = 2\omega_M$ та $\omega_s = \omega_M/2$, в результаті отримемо описану далі картину.

Низькочастотна фільтрація початкового сигналу $x(n)$ дає дискретний сигнал $x_1(n)$, який відповідає смузі $[0, \omega_M/2]$ початкового спектру, а фільтрація низьких частот сигналу $x(n)(-1)^n$ дає дискретний сигнал $x_2(n)$, що відповідає смузі $[\omega_M/2, \omega_M]$ початкового спектру, зсунутій і переставленій у смугу частот $[0, \omega_M/2]$.

Описаний крок можна застосувати знову до сигналів $x_1(n)$ і $x_2(n)$, розглядаючи тільки один з двох послідовних відліків (тобто виконуючи децимацію – проріджування відліків), і попереднє співвідношення буде надалі вірним, якщо використовується зсув частоти, рівний $\omega_M/2 = \omega_s/4$. Тоді можна отримати 4 смуги. Таким чином, цей метод може розділити спектр сигналу на швидко зростаюче число смуг. На рис. 7.5 показана деревоподібна структура цього методу, де $f_m = \omega_M/2\pi$. Якщо виконуємо r послідовних кроків, то отримуємо 2^r смуг.

Хвилькове перетворення одновимірного сигналу $x(n)$ – це описаний вище розклад його на компоненти, кожна з яких відповідає за певну частотну смугу в спектрі початкового сигналу. Кожна з компонент, яка також є одновимірним сигналом у дискретному часі, може розкладатися далі або ж розклад може бути припинено.

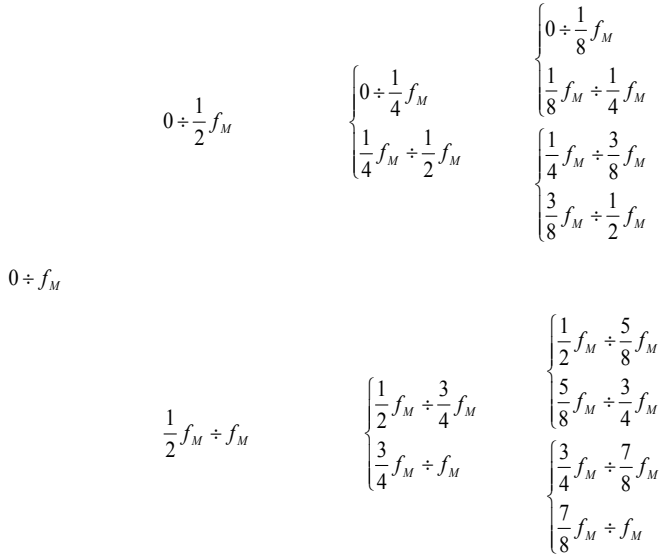
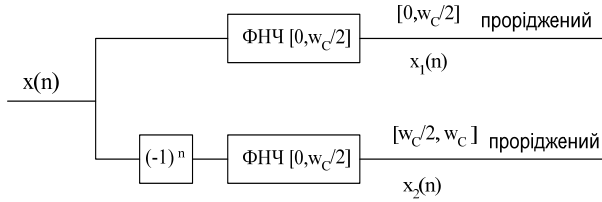


Рис. 7.5. Смуговий аналіз з використанням зсуву спектру та одного цифрового фільтра нижніх частот.

Аналогічно хвилькове перетворення для двовимірного сигналу (зображення) $x(n_1, n_2)$ – це розклад його на двовимірні компоненти, кожна з яких відповідає за певну частотну область в спектрі початкового зображення. У цьому разі для реалізації хвилькового перетворення слід використовувати двовимірні цифрові фільтри. Кожна з компонент, яка є двовимірним сигналом, може розкладатися далі або ж розклад можна зупинити.

Якщо виконуємо двовимірне хвилькове перетворення зображення $M \times N$ точок, то отримуємо рівно MN коефіцієнтів хвилькового перетворення – те ж

саме число, що й число точок початкового зображення. Тобто, хвилькове перетворення не змінює наявного обсягу інформації.

На рис. 7.6 показано варіант трирівневого хвилькового перетворення популярного тестового зображення Lena [30]. При такому перетворенні на кожному рівні перетворення, починаючи з другого, далі розкладаємо лише низькочастотну компоненту. Рівень 1 (розміру 512×512) – однорівневий хвильковий розклад початкового зображення Lena з рівня 0. Рівень 2 (розміру 256×256) показує однорівневе хвилькове перетворення низькочастотної компоненти з рівня 1. Рівень 3 (розміру 128×128) дає однорівневе хвилькове перетворення низькочастотної компоненти з рівня 2. На рис. 7.7 показано зведені разом усі 10 компонент описаного трирівневого хвилькового перетворення [30]. Отримана матриця має таке саме число відліків, як і початкове зображення.

Як базовий інструмент для розкладу сигналів, хвилькові перетворення можуть розглядатися як дуальні до більш звичних методів аналізу Фур'є. Аналіз Фур'є пов'язаний з інтуїтивним інженерним поняттям „спектру” або „частотного змісту” сигналу. Аналіз на основі хвилькових перетворень, на противагу, пов'язаний з інтуїтивним поняттям „роздільної здатності” або „масштабування” сигналу. На функціональному рівні, Фур'є-аналіз співвідноситься з хвильковим аналізом як спектральні аналізатори у порівнянні з мікроскопами.

Хвилькові перетворення витіснили більш звичний Фур'є-базовий метод у формі дискретного косинусного перетворення, що використовується у форматі стиску зображень JPEG-9. Новіший стандарт стиску JPEG-2000 базується на хвилькових перетвореннях. На згаданих перетвореннях ґрунтуються й методи стиску зображень відбитків пальців.

Візьмемо низькочастотну компоненту з верхнього рівня (рівень 3) трирівневого хвилькового перетворення з рис. 7.6. Ця компонента – проріджена (з коефіцієнтом $8^2=64$) та згладжена версія початкового зображення. Дуже простий шлях стиску – залишити вказану компоненту й викинути решта 9 компонент. Отримаємо степінь стиску даних $64:1$.

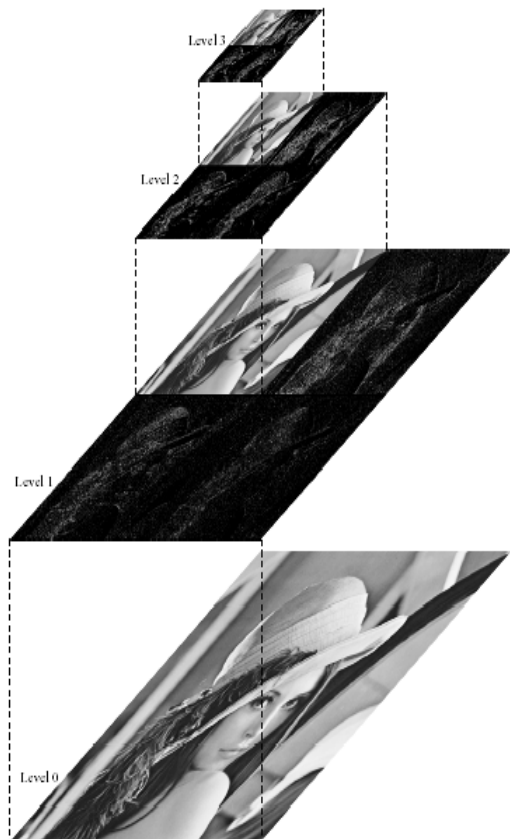


Рис. 7.6. Три рівні хвилькового перетворення кольорового зображення 512x512.



Рис.7.7 Отримані 10 компонент хвилькового перетворення.

Зауважимо, що коли потрібно наближення до початкового зображення повного розміру, слід виконати інтерполяцію низькочастотної компоненти з коефіцієнтом 64. Це можна ефективно зробити за допомогою триступеневого інтерполюючого цифрового фільтра. Користуючись іншими з решта 9 компонент можна покращити якість відтвореного зображення, як показано на рис. 7.8.

Таким чином, хвилькове перетворення природно узгоджується зі схемою передачі, яка прогресивно покращує якість отриманого зображення. Дійсно, дуже зручно мати потік бітів, який може бути зупинений у будь-якій точці, щоб забезпечити прийнятну якість відтвореного зображення. Це називається прогресивною передачею або вкладеним потоком бітів [30].

Така властивість ідеально підходить, наприклад, для передачі зображень у мережі Інтернет. Як відомо, Інтернет – це неоднорідна суміш, стосовно числа користувачів та їх обчислювальних можливостей і фактичних пропускних здатностей. Хвилькові перетворення забезпечують природний шлях задовільнити усіх користувачів: користувачам з обмеженими ресурсами можна передати зображення нижчої якості, в той час як користувачі з потужнішими ресурсами можуть використати їх, щоб отримати зображення кращої якості. Ця властивість також дуже корисна для Веб-браузерів, де отримання зображення нижчої якості за короткий час очікування може бути більш прийнятним, ніж отримання якіснішого зображення після тривалого часу очікування.



Рис.7.8 . Відтворення зображення різної якості.

7.3. Стиск зображень з використанням методу кодування областей хвилькового перетворення

У цьому методі розглядаються області коефіцієнтів хвилькового перетворення зображення, які мають різні розміри. Ідея полягає в тому, щоб коефіцієнти в різних областях опрацювати (квантувати та кодувати) по-різному, бо кожна смуга може розглядатися як окремий випадковий процес.

На рис. 7.9 показано результат типового розподілу бітів для різних областей при загальній оцінці стиску 1 біт на піксель. Компоненти кодуються по-різному залежно від частотної області, за яку відповідає конкретна компонента. Низькочастотні компоненти в лівому верхньому куті несуть більше інформації, ніж високочастотні компоненти в нижньому правому куті, тому вони кодуються більшою кількістю біт. Це нерівномірне квантування. У результаті, після квантування отримуємо такий степінь стиску :

$$\{(8+6+5+5)/4+2+2+2\}/4+1+0+0\}/4=1 \text{ біт/піксель.}$$

8	6	2	1
5	5		
2		2	
0		0	

Рис. 7.9. Стиск з використанням кодування областей хвилькового перетворення.

7.4. Стиск зображень з використанням методу дерев нулів хвилькового перетворення

Хвильковий розклад зображення можемо мислити собі як просторову множину коефіцієнтів, яка складається з дерев. Дерево коефіцієнтів хвилькового

перетворення означається як множина коефіцієнтів із різних областей, що описують ту саму просторову ділянку в зображенні.

На рис. 7.10 показано трирівневий хвильковий розклад зображення Lena та структуру дерева хвилькових коефіцієнтів, які відповідають за ділянку ока в зображенні. Стрілки на рис. 7.10 (б) показують співвідношення між батьківськими та дочірніми вершинами в дереві; Н позначає високу частоту, L – низьку частоту, 1,2,3 – рівні частот. Найбільш низькочастотна компонента хвилькового розкладу зображається кореневими вершинами дерева, які є у верхньому лівому куті. Найбільш високочастотна компонента хвилькового розкладу представляється кінцевими вершинами дерева, які знаходяться у нижньому правому куті. За винятком кореневої вершини, яка має лише три дочірніх вершини, кожна батьківська вершина має чотири дочірніх вершини, а саме ділянку розміру 2×2 з тим самим просторовим розташуванням у більш високочастотній компоненті.

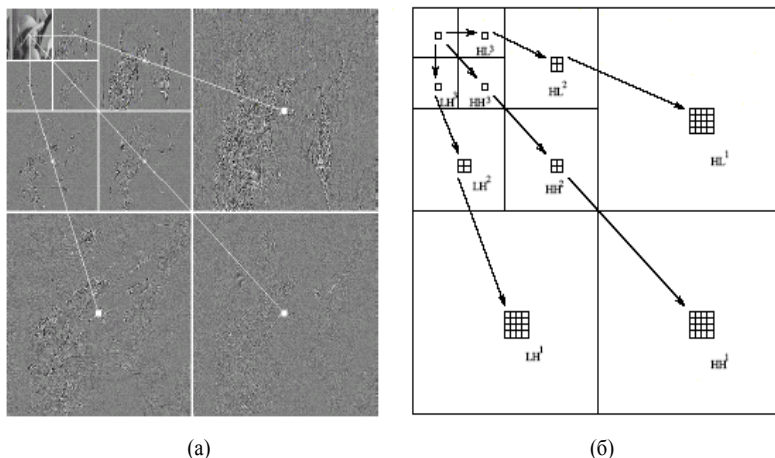


Рис. 7.10 (а) Трирівневе хвилькове перетворення зображення. (б) Дерево коефіцієнтів

Методи стиску зображень з використанням дерев нулів (EZW, SPIHT) ґрунтуються на ідеї використання багатьох проходів для кодування дерев нулів, щоб передати найбільші хвилькові коефіцієнти спочатку.

Дерева нулів переглядаємо, користуючись змінними порогами. Множина коефіцієнтів дерева називається суттєвою, якщо значення найбільшого коефіцієнта в цій множині не менше від певного порога (наприклад, поріг може бути степенем двійки). В іншому випадку ця множина є несуттєвою. Аналогічно, коефіцієнт називається суттєвим, якщо його значення не менше від певного порога. В іншому випадку цей коефіцієнт є несуттєвим.

У кожному проході суттєвість більшої множини в дереві аналізується спочатку. Якщо множина несуттєва, вона кодується відповідним символом, і при відтворенні всі множина заповнюється нулями. В іншому випадку множина розбивається на підмножини (дочірні множини) для подальшого аналізу на суттєвість. Після того, як в поточному проході проаналізовано всі коефіцієнти, поріг перед черговим проходом зменшується вдвоє.

Основне припущення при кодуванні за допомогою дерев нулів полягає в тому, що більшість зображень можуть бути змодельовані як такі, що мають спадаючі спектральні значення. Це конкретно означає: якщо батьківська вершина дерева коефіцієнтів хвилькового перетворення несуттєва, то дуже висока ймовірність, що її дочірні вершини також будуть несуттєвими. У цьому разі дуже ефективно працює для стиску кодів символ для дерева нулів.

Далі детально розглядаємо, як приклад, алгоритм EZW на підставі дерев нулів коефіцієнтів хвилькового перетворення [28]. Метод SPIHT є розвитком методу EZW.

Позначення EZW утворене від слів Embedded Zerotree Wavelet. Пояснимо значення кожного із слів у цьому методі. Слово wavelet (хвилькове) означає, що метод працює з коефіцієнтами хвилькового перетворення. Слово embedded (вкладене) означає, що кодер стискає зображення, утворюючи потік бітів із зростаючою точністю. Тобто, коли до потоку додати більше бітів, відтворене зображення буде точнішим. Слово zerotree (дерево нулів) означає, що метод використовує поняття дерева нулів. Хвилькове перетворення дає представлення зображення, структуроване на дерева. Дерево нулів – це четвіркове (кожна вершина розгалужується на чотири дочірні вершини) дерево, усі коефіцієнти

якого менші від поточного порогу. Це дерево кодується одним символом, а декодер відтворює його як заповнене нулями четвіркове дерево.

Проходимо коефіцієнти хвилькового перетворення у порядку, який називається порядком Мортонa. Цей порядок схематично показаний нижче.

1	2	5	6	17	18	21	22
3	4	7	8	19	20	23	24
9	10	13	14	25	26	29	30
11	12	15	16	27	28	31	32
33	34	37	38	49	50	53	54
35	36	39	40	51	52	55	56
41	42	45	46	57	58	61	62
43	44	47	48	59	60	63	64

На кожному етапі цього методу виконують головний та підрядний проходи. *Головний прохід.* Якщо коефіцієнт додатній і більший від порогу, то він кодується символом p . Якщо коефіцієнт від'ємний і за абсолютною величиною більший від порогу, то він кодується символом n . Символом t кодується корінь дерева нулів. Символом z кодується ізольований нуль, тобто коефіцієнт менший за абсолютною величиною від порогу, але не є коренем дерева нулів.

Якщо коефіцієнт закодований як p або n , то ми заміняємо його для подальших проходів нулями. Крім того, вносимо його в список для підрядного проходу.

Підрядний прохід. Поріг для підрядного проходу отримується зменшенням порогу для відповідного головного проходу вдвоє. Якщо коефіцієнт більший від порогу, то він кодується значенням 1, а якщо менший – значенням 0.

Коефіцієнти кодуються в послідовних проходах. Для кожного проходу вибирається поріг, з яким порівнюються коефіцієнти. Якщо коефіцієнт більший за абсолютною величиною від порогу, то він кодується та заміняється нулем. У протилежному випадку коефіцієнт залишається для чергового проходу. Коли ми

пройшли всі коефіцієнти, поріг зменшується і ми знову проходимо коефіцієнти, щоб додати більше деталей до вже закодованого зображення.

Розглянемо для прикладу тривірневе хвилькове перетворення зображення розміру 8×8 елементів. Матриця значень цього хвилькового перетворення наведена далі.

63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Оскільки найбільше абсолютне значення коефіцієнтів у цій матриці рівне 63, ми можемо вибирати початковий поріг з проміжку $(31,5;63]$. Нехай початковий поріг дорівнює 32. Табл. 7.1 описує опрацювання коефіцієнтів на першому основному проході.

Якщо маємо несуттєвий коефіцієнт без дочірніх коефіцієнтів, то в цьому разі ситуації t чи z не розрізняються. Ми ставимо символ $t(z)$.

Далі наведені коментарі до табл. 7.1.

1) Коефіцієнт має значення 63, яке більше від порогу 32 і додатне. Тому використовується символ p . При декодуванні цього символа декодер замінить його на середнє значення проміжку $[32,64)$, тобто на 48.

2) Хоча коефіцієнт 31 є несуттєвим відносно порогу 32, він має суттєве дочірнє значення на дві рівні нижче в смузі LH1 рівне 47. Тому цей коефіцієнт заміняється на символ z ізольованого нуля.

Табл. 7.1. Опрацювання коефіцієнтів на першому основному проході

Підсмуга	Значення коефіцієнта	Кодуючий символ	Значення при відтворенні	Коментар
LL3	63	p	48	1)
HL3	-34	n	-48	
LH3	-31	z	0	2)
HH3	23	t	0	3)
HL2	49	p	48	
HL2	10	t	0	4)
HL2	14	t	0	
HL2	-13	t	0	
LH2	15	t	0	
LH2	14	z	0	5)
LH2	-9	t	0	
LH2	-7	t	0	
HL1	7	$t(z)$	0	6)
HL1	13	$t(z)$	0	
HL1	3	$t(z)$	0	
HL1	4	$t(z)$	0	
LH1	-1	$t(z)$	0	
LH1	47	p	48	7)
LH1	-3	$t(z)$	0	
LH1	-2	$t(z)$	0	

3) Величина 23 менша від 32 і всі дочірні значення (3,-12,-14,8) у смузі HH2 і всі коефіцієнти у смузі HH1 несуттєві. Коефіцієнт 23 замінюється на символ t дерева нулів, а для всіх коефіцієнтів у смугах HH2 та HH1 під час першого основного проходу не утворюються ніякі символи.

4) Величина 10 менша від 32 і всі її дочірні значення (-12,7,6,-1) також менші від порогу 32. Отже, утворюється символ t дерева нулів. Зауважимо, що в цьому

дереві порушується гіпотеза про “спадаючий спектр”, бо коефіцієнт -12 в підсмугі HL1 більший за абсолютною величиною від свого батьківського коефіцієнта 10 . Проте ціле дерево має значення менші від порогу 32 , а тому ще є деревом нулів.

5) Величина 14 несуттєва порівняно з 32 . Її дочірні значення рівні $(-1,47, -3,2)$. Оскільки її дочірні значення 47 суттєве, то утворюється символ ізольованого нуля.

6) Зауважимо, що ніякі символи не були утворені для підсмуги HH2, яка при проходженні коефіцієнтів передує підсмугі HL1. Також зауважимо, що оскільки підсмуга HL1 не має дочірніх вершин, то між символами t чи z немає різниці. Ми використовуємо символ $t(z)$.

7) Величина 47 суттєва порівняно з 32 . Зауважимо, що для подальших основних проходів цей коефіцієнт буде замінений значенням 0 ; так що при черговому основному проході з порогом 16 батьківське значення 14 цього коефіцієнта буде замінене символом кореня дерева нулів.

Під час першого основного проходу, який використовує поріг 32 , знайдено чотири суттєвих коефіцієнти. Ці коефіцієнти уточнюються під час першого підрядного проходу. Перед першим підрядним проходом проміжок для всіх суттєвих коефіцієнтів – це проміжок $[32,64)$. Перший підрядний прохід уточнить ці величини й віднесе їх до проміжку $[32,48)$, що кодується символом 0 , або до проміжку $[48,64)$, що кодується символом 1 . Тобто, межею для прийняття рішення є число 48 .

Порядок дій при першому підрядному проході ілюструється в табл. 7.2.

Перший коефіцієнт рівний 63 і розміщується у верхньому проміжку, центр якого дорівнює 56 . Другий коефіцієнт рівний 34 і розміщується в нижньому проміжку. Третій коефіцієнт 49 є у верхньому проміжку, а четвертий коефіцієнт 47 – у нижньому проміжку.

Табл. 7.2. Опрацювання коефіцієнтів на першому підрядному проході

Значення	Кодуючий	Значення при
----------	----------	--------------

коефіцієнта	символ	відтворенні
63	1	56
34	0	40
49	1	56
47	0	40

Матриця, яка може бути відтворена декодером, виходячи з даних першого основного та підрядного проходів наведена нжче.

56	-40	56	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	40	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Опрацювання продовжується на другому основному проході з новим зменшеним вдвоє порогом 16. Під час цього проходу проглядаються лише ті коефіцієнти, які ще не були ідентифіковані як суттєві на першому проході. Крім того, ті коефіцієнти, які раніше були ідентифіковані як суттєві, замінюються нулями.

Видозмінена матриця значень хвилькового перетворення перед другим основним проходом наведена далі.

0	0	0	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2
-5	9	-1	0	4	6	-2	2
3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4

Таким чином, на другому основному проході коефіцієнт -31 у підсмuzі LH3 кодується як суттєвий від'ємний, коефіцієнт 23 у підсмuzі HH3 як суттєвий додатний, три коефіцієнти у підсмuzі HL2, які раніше не були ідентифіковані як суттєві, всі кодуються як корені дерев нулів. Так само кодуються як корені дерев нулів усі чотири коефіцієнти у підсмuzі LH2 та всі чотири коефіцієнти у підсмuzі HH2. На цьому другий основний прохід завершується.

Список для другого підрядного проходу містить коефіцієнти $(63,34,49,47,31,23)$, які перед цим проходом представляли три проміжки $[48,64)$, $[32,48)$ та $[16,32)$. Кожен із наведених проміжків має однакову ширину, яка дорівнює 16 .

Опрацювання уточнить кожне значення, утворивши два нових проміжки для кожного з трьох вказаних проміжків. Використовуючи середні значення проміжків як значення для відтворення, декодер утворить такі величини: $(60,36,52,44,28,20)$.

Матриця, яка може бути відтворена декодером, виходячи з даних першого та другого основного та підрядного проходів наведена далі.

Опрацювання в методі, що розглядаємо, продовжується з чергуванням основного та підрядного проходів, і може бути зупинене в будь-який момент часу.

60	-36	52	0	0	0	0	0
-28	20	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	44	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Усі етапи кодування та декодування в методі EZW детально ілюструються нижче.

Етапи роботи EZW кодера.

поріг = 32

$D = pnztpittztittttpt$ $S = 1 \ 0 \ 1 \ 0$

поріг = 16

$D = zlnptttttt$ $S = 1 \ 0 \ 0 \ 1 \ 1 \ 0$

поріг = 8

$D = zzzzppnppntlnnptptnttttttptttttttttpttttttttt$

$S = 1 \ 0 \ 0 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 0$

поріг = 4

$D = zzzzzztztnzzzpttpptpnpntttttptpnpptttttptttptnp$

$S = 1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1$
 $1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 0 \ 0$

поріг = 2

D = zzzzzzzzzzztpzzztptttntptppttpttppntttppnnpttptppttt

S = 1 0 1 1 1 1 0 0 1 1 0 1 0 0 0 1 0 1 1 1
1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 0 0 0 0 0
1 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1

поріг = 1

D = zzztztztzttttnttt

Етапи роботи EZW декодера.

поріг = 16	рівень = 1	поріг = 8	рівень = 2
56 -40 56 0 0 0 0 0		60 -36 52 0 0 0 0 0	
0 0 0 0 0 0 0 0		-28 20 0 0 0 0 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	
0 0 0 40 0 0 0 0		0 0 0 44 0 0 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	

поріг = 4	рівень = 3	поріг = 2	рівень = 4
62 -34 50 10 0 14 -14 0		63 -35 49 11 7 13 -13 7	
-30 22 14 -14 0 0 0 0		-31 23 15 -13 0 5 7 0	
14 14 0 -14 0 0 0 10		15 15 0 -13 5 -7 0 9	
-10 0 -14 10 0 0 0 0		-9 -7 -15 9 5 0 0 0	
0 10 0 46 0 0 0 0		-5 9 0 47 5 7 0 0	
0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 5	
0 0 0 0 0 0 0 0		0 0 7 -5 0 7 0 7	
0 10 0 0 0 0 0 0		5 11 5 7 0 0 -5 5	

поріг = 1 рівень = 5

поріг = 0.5 рівень = 6

63	-34	49	10	7	13	-12	7	63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	0	-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9	15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2	-9	-7	-14	8	4	-2	3	2
-5	9	0	47	4	6	-2	2	-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4	3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6	2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4	5	11	5	6	0	3	-4	4

початкове зображення =

відтворене зображення =

63	-34	49	10	7	13	-12	7	63	-34	49	10	7	13	-12	7
-31	23	14	-13	3	4	6	-1	-31	23	14	-13	3	4	6	-1
15	14	3	-12	5	-7	3	9	15	14	3	-12	5	-7	3	9
-9	-7	-14	8	4	-2	3	2	-9	-7	-14	8	4	-2	3	2
-5	9	-1	47	4	6	-2	2	-5	9	-1	47	4	6	-2	2
3	0	-3	2	3	-2	0	4	3	0	-3	2	3	-2	0	4
2	-3	6	-4	3	6	3	6	2	-3	6	-4	3	6	3	6
5	11	5	6	0	3	-4	4	5	11	5	6	0	3	-4	4

різниця =

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Для подальшого стиску потоку даних на виході EZW кодера може бути додатково застосоване кодування Хафмена.

Рис. 7.11 показує початкові загальновідомі тестові зображення Lena та Barbara. Вони були стиснуті до 0,25 бітів/піксель (коефіцієнт стиску 32:1) за допомогою методів JPEG (з використанням дискретних косинусних перетворень) та EZW. Порівнюємо якість відповідних відтворених зображень.

Для цього слід використати певні міри спотворення, які характеризують відхилення відтвореного зображення від початкового зображення. Переважно в області опрацювання зображень використовують такі дві рівносильні міри спотворення:

- 1) середньоквадратична похибка (mean square error, MSE)
- 2) відношення максимальний рівень сигналу до шуму (peak signal to noise ration).

Одиницею вимірювання для міри PSNR є децибели: $PSNR = 10 \lg \frac{255}{MSE}$.



Рис. 7.11. Тестові зображення Lena та Barbara.

Відтворені після стиску методом JPEG зображення мають такі характеристики: PSNR=31,6 дБ для зображення Lena; PSNR=25,2 дБ для зображення Barbara. Разом з тим відтворені після стиску методом EZW зображення: PSNR=34,1 дБ для зображення Lena; PSNR=27,6 дБ для зображення Barbara. Як бачимо, метод EZW дає кращі характеристики для відтворених зображень.

7.5. Адаптивні хвилькові перетворення

Слід зауважити, що традиційний підхід використання хвилькових перетворень з фіксованою частотною роздільною здатністю (розглянуте раніше хвилькове перетворення) є добрий лише в загальному для типового статистичного класу зображень. Цей клас добре узгоджується з характеристиками вибраного фіксованого перетворення. Виникає природне питання: чи можна отримати кращі результати, підбираючи перетворення для особливого зображення, яке може не належати до типового ансамблю.

Більш точно, розглянуте раніше хвилькове перетворення добре підходить для типових натуральних зображень, що мають експоненційно спадні спектральні значення разом зі стійкими низькочастотними компонентами (наприклад, фон зображення) і інформаційно важливими короткотривалими високочастотними компонентами (наприклад, гострі грані на зображенні).

Але є важливі класи зображень, чії властивості не узгоджуються з загальноприйнятим хвильковим розкладом, наприклад, зображення, що мають стійкі високочастотні компоненти. Добрим прикладом слугують текстури штанів, шалика чи скатертини в зображенні Barbara.

Іншим особливим класом зображень, для яких не підходить логарифмічне хвилькове перетворення, є клас зображень відбитків пальців (див. типовий приклад на рис. 7.12), які мають періодичні високочастотні ребристі фрагменти. Цим зображенням краще відповідають хвилькові розклади, що мають добру частотну локалізацію на високих частотах.



Рис. 7.12. Зображення відбитку пальця.

Це спричинило пошук альтернативних хвилькових перетворень, які більш адаптивні у їх представленні і більш стійкі для великого класу зображень з

невідомими або невідповідними просторово-частотними характеристиками. Слід зауважити, що задача знаходження оптимального розкладу для кожного окремого зображення є некоректно поставленою. Проте, ситуація стає більш визначеною, якщо розглядаємо велику, але скінчену бібліотеку бажаних перетворень, і підбираємо найкраще перетворення з цієї бібліотеки, адаптуючи до конкретного зображення. Щоб це було допустимим, є дві вимоги. Перша вимога: бібліотека має містити репрезентативну множину перетворень (наприклад, було б добре, якби вона містила загальноприйнятий хвильковий розклад). Друга вимога: суттєво, щоб був швидкий спосіб пошуку в бібліотеці для знаходження найкращого для конкретного зображення перетворення.

Ці дві вимоги дають логічне узагальнення загальноприйнятого хвилькового перетворення під назвою пакетний хвильковий розклад [30].

Хвильковий розклад, який відповідає структурі логарифмічного дерева, є найбільш відомим членом хвилькового пакету. Тоді як хвилькові розклади найкраще узгоджуються з сигналами, що мають спадний енергетичний спектр, хвилькові пакети узгоджуються з сигналами, що мають майже довільні спектральні особливості.

Переваги опрацювання, пов'язані з хвильковими пакетами, звичайно залежать від конкретного зображення. Для таких складних зображень як Barbara, показаний на рис. 7.13 хвильковий розклад з пакету дає кращий стиск, ніж звичний хвильковий розклад. Білі лінії на цьому рисунку представляють границі областей коефіцієнтів хвилькового перетворення. Характеристики стиску: стіпень стиску 0,1825 бітів/піксель, міра спотворення PSNR=27,6 дБ.

Важливим практичним застосуванням хвилькових пакетів є метод для стиску зображень відбитків пальців.

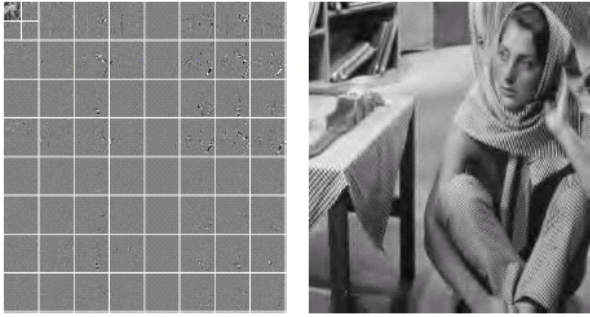


Рис. 7.13. Хвильковий розклад з пакету для зображення Barbara.

Через складність, пов'язану з адаптивними перетвореннями хвилькового пакету, цей метод використовує фіксоване хвилькове перетворення. На рис. 7.14 показано структуру цього перетворення згідно з вказаним методом. Воно було розроблене для зображень відбитків пальців з роздільною здатністю 200 точок на сантиметр з використанням спектрального аналізу та методу проб і помилок. 64 компоненти отримуємо в результаті п'ятирівневого хвилькового перетворення.

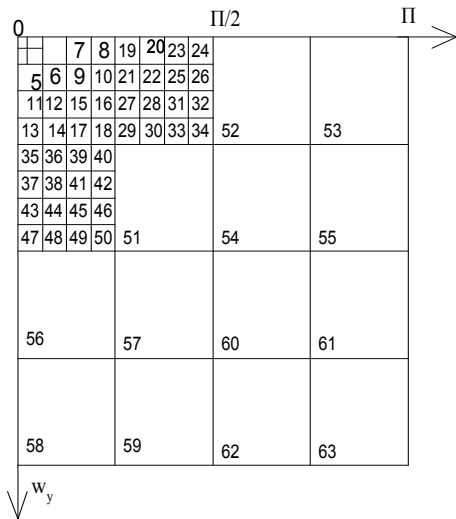


Рис. 7.14. П'ятирівневе хвилькове перетворення для відбитків пальців.

Узагальнюючи адаптивні хвилькові пакети, можна ввести розбиття зображення на фрагменти і для кожного з фрагментів або групи фрагментів підбирати відповідне перетворення з пакету. Це показано на рис. 7.15. Зліва маємо розбиття на фрагменти зображення, яке відділяє небо у фоні від висотного будинку та ставка на передньому плані. Справа маємо проілюстровані найкращі хвилькові перетворення з пакету для кожного фрагменту зображення. Чорні лінії представляють фрагменти зображення, а білі лінії – частотні компоненти вибраних хвилькових перетворень.

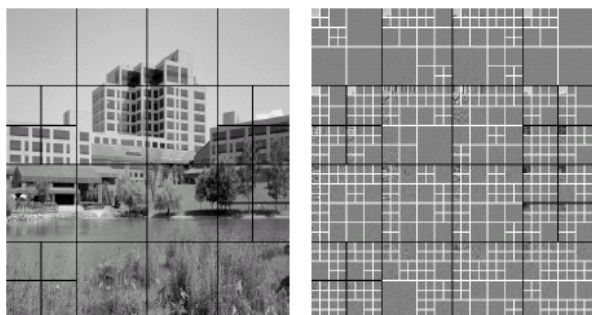


Рис. 7.15. Підбирання найкращих перетворень для фрагментів зображення.

7.6. Завдання для самоперевірки.

1. Двовимірне дискретне косинусне перетворення має ту властивість, що:

- А) зосереджує найбільші за абсолютною величиною значення переважно у верхньому лівому куті (низькі частоти) матриці значень перетворення;
- Б) зосереджує найбільші за абсолютною величиною значення переважно у нижньому правому куті (високі частоти) матриці значень перетворення;
- В) зосереджує найбільші за абсолютною величиною значення переважно у нижньому лівому куті (середні частоти) матриці значень перетворення;
- Г) зосереджує найбільші за абсолютною величиною значення переважно у верхньому правому куті (середні частоти) матриці значень перетворення;

Д) немає правильної відповіді.

2. Дерево коефіцієнтів хвилькового перетворення означається як:

А) множина коефіцієнтів з однієї області, що описують різні просторові ділянки в зображенні; Б) множина коефіцієнтів з верхнього правого кута, що описують ту саму просторову ділянку в зображенні; В) множина коефіцієнтів з нижнього лівого кута, що описують різні просторові ділянки в зображенні; Г) множина коефіцієнтів із різних областей, що описують ту саму просторову ділянку в зображенні; Д) немає правильної відповіді.

3. Методи стиску зображень з використанням дерев нулів ґрунтуються на:

А) ідеї використання багатьох проходів для кодування дерев нулів, щоб передати найменші хвилькові коефіцієнти спочатку;
Б) ідеї використання багатьох проходів для кодування дерев нулів, щоб передати найбільші хвилькові коефіцієнти спочатку;
В) ідеї використання багатьох проходів для кодування дерев нулів, щоб передати нульові хвилькові коефіцієнти спочатку;
Г) ідеї використання багатьох проходів для кодування дерев нулів, щоб передати попарно рівні хвилькові коефіцієнти спочатку;

Д) немає правильної відповіді.

4. Підрядний прохід в методі EZW може дати такі значення:

А) 0,1; Б) p,n ; В) t,z ; Г) -1,1; Д) 0,-1.

5. Для стиску зображень відбитків пальців використовують:

А) адаптивні перетворення хвилькового пакету; Б) фіксоване хвилькове перетворення; В) дискретне косинусне перетворення; Г) дискретне перетворення Фур'є; Д) немає правильної відповіді.

6. Хвилькове перетворення сигналу це:

А) розклад його на спектральні компоненти; Б) розклад його на часові компоненти, кожна з яких відповідає за певну частотну смугу; В) перетворення сигналу в спектральну область; Г) перетворення сигналу в часову область; Д) немає правильної відповіді.

7. Двовимірне хвилькове перетворення зображення $M \times N$ точок дає:

А) MN коефіцієнтів; Б) менше MN коефіцієнтів; В) M^2N коефіцієнтів; Г) MN^2 коефіцієнтів; Д) $(MN)^2$ коефіцієнтів.

8. Прогресивною передачею зображення називають:

А) потік бітів, який слід прийняти до кінця, щоб забезпечити прийнятну якість відтвореного зображення; Б) потік бітів, який можна зупинити тільки у фіксованих точках, щоб забезпечити прийнятну якість відтвореного зображення; В) потік бітів, який можна зупинити у будь-якій точці, щоб забезпечити прийнятну якість відтвореного зображення; Г) потік бітів, який можна зупинити лише посередині, щоб забезпечити прийнятну якість відтвореного зображення; Д) немає правильної відповіді.

9. Метод кодування областей хвилькового перетворення полягає в такому:

А) коефіцієнти в сусідніх областях опрацьовують по-різному; Б) коефіцієнти в однакових за розміром областях опрацьовують однаково; В) коефіцієнти в різних областях опрацьовують по-різному; Г) коефіцієнти в усіх областях опрацьовують однаково; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1) А; 2) Г; 3) Б; 4) А; 5) Б; 6) Б; 7) А; 8) В; 9) В.

РОЗДІЛ 8. ОРГАНІЗАЦІЯ ЦИФРОВИХ СИГНАЛЬНИХ ПРОЦЕСОРІВ

Для опрацювання сигналів та зображень найчастіше використовують цифрові сигнальні процесори. Розглянемо підходи до їх реалізації на базі обчислення алгоритму ШПФ.

В загальному випадку, вимоги до пам'яті при реалізації N -точкового ШПФ такі [3]: необхідно N комірок для дійсних даних, N комірок для уявних даних і N комірок для синусоїдальних базисних функцій (коефіцієнти повертання). Додаткові комірки пам'яті необхідні у випадку використання зважування з використанням віконних функцій (windowing). Якщо прийняті вимоги по пам'яті задоволені, процесор повинен виконати необхідні обчислення за необхідний час. Виробники ЦСП або проводять тест продуктивності для зазначеної розмірності ШПФ, або визначають час обчислення для базової операції "метелик". При порівнянні характеристик ШПФ важливо упевнитися, що у всіх випадках використовується однаковий тип ШПФ. Наприклад, тест 1024-точкового ШПФ на одному ЦСП, отриманому за допомогою алгоритму ШПФ за основою 2, не слід порівнювати з тестом алгоритму ШПФ за основою 4 для іншого ЦСП.

Інша проблема полягає у виборі процесора з фіксованою чи рухомою крапкою. Результат обчислення "метелика", може бути більшим ніж розрядна сітка ЦСП з фіксованою крапкою, що створює проблему при реалізації ШПФ на процесорах такого типу. Для запобігання переповнення, дані потрібно масштабувати, заздалегідь залишаючи достатню кількість додаткових розрядів для збільшення значень оброблюваних даних. Альтернативний метод полягає в тому, що дані можуть масштабуватися після кожного каскаду обчислення ШПФ. Метод масштабування даних після кожного проходу ШПФ відомий як блокова рухома крапка, (block floating point). Блок масштабується таким чином, щоб відносні співвідношення між даними залишилися попередніми. Наприклад, якщо кожне слово даних зсунене вправо на один розряд (поділене на 2), то абсолютні

значення змінюються, але відносно один одного співвідношення даних залишаються попередніми.

У 16-розрядному ЦСП з фіксованою крапкою після множення формується 32-розрядне слово. Тому до складу процесора входять регістри-акумулятори більшої розрядності, наприклад, в сімействі процесорів ADSP21xx (фірма Analog Devices) операції множення з накопиченням виконуються за допомогою внутрішнього 40-розрядного регістра-акумулятора, що забезпечує розширення динамічного діапазону.

Використання ЦСП з рухомою крапкою усуває потребу в масштабуванні даних і тому приводить до більш простої реалізації алгоритму ШПФ, але наслідком цього спрощення є збільшення часу опрацювання, що потрібно для складних арифметичних обчислень з рухомою крапкою. Крім того, 32-х розрядний ЦСП із рухомою крапкою має менший рівень шумів заокруглення, ніж 16-ти розрядний ЦСП з фіксованою крапкою.

8.1. Вимоги до процесорів при опрацюванні сигналів

Є два основних способи опрацювання сигналів у режимі реального часу (РРЧ) [1]: опрацювання одного відліку в кожен момент часу (безперервне опрацювання) і опрацювання одного пакета даних у кожен момент часу (пакетне опрацювання). Системи, засновані на безперервному опрацюванні, такі як цифровий фільтр, одержують дані у виді одного відліку в кожен момент часу. У кожному такті новий відлік надходить у систему, а оброблений відлік передається на вихід. Системи, засновані на пакетному опрацюванні, такі як побудований на ШПФ цифровий аналізатор спектру, одержують дані у виді цілого пакета відліків. Відбувається опрацювання всього пакета вихідних даних, результатом якого є пакет перетворених вихідних даних.

Для забезпечення функціонування в РРЧ обчислення ШПФ слід виконувати в проміжку, що відповідає часу накопичення одного пакету даних. Передбачається, що, поки обчислюється ШПФ поточного пакета даних, ЦСП

накопичує дані для наступного пакету. При накопиченні даних важливу роль відіграють спеціальні архітектурні особливості ЦСП. Безупинне отримання даних полегшується, завдяки можливостям гнучкої адресації даних у ЦСП у сполученні, наприклад, з використанням різних каналів прямого доступу до пам'яті (DMA).

Приклад (порядок) обчислення ШПФ у РРЧ.

1. Припустимо, що час виконання 1024-точкового алгоритму ШПФ за основою 2 дорівнює 69 мкс (процесор типу TigerSHARC, 32-розрядний режим).

2. Визначаємо максимальну частоту дискретизації:

$$f_s (\text{maximum}) < 1024 \text{ відліків} / 69 \text{ мкс} = 14,8 \text{ MSPS.}$$

3. Визначаємо ширину смуги вхідного сигналу, для даного прикладу менша 7,4 МГц.

4. При розрахунку не приймалися до уваги додаткові операції, пов'язані з реалізацією ШПФ і передачею вхідних/вихідних даних

Наведений приклад дає оцінку максимальної ширини смуги сигналу, що може бути опрацьована даним ЦСП з врахуванням характеристик реалізованого на ньому ШПФ.

Інший підхід полягає в тому, щоб, задаючи ширину смуги сигналу, розробити вимоги до ЦСП для опрацювання сигналу в розглянутій смузі. Якщо ширина смуги частот сигналу відома, необхідна частота дискретизації може бути визначена шляхом її множення на коефіцієнт 2-2,5 (збільшення частоти дискретизації може вимагатися для ослаблення вимог до попереднього АЦП фільтра низьких частот, який усуває ефект накладення спектру). На наступному кроці визначається число точок ШПФ, необхідне для досягнення бажаної роздільної здатності за частотою. Значення роздільної здатності за частотою отримуємо діленням частоти дискретизації f_s на кількість N точок ШПФ. Ці й інші підходи стосовно реалізації ШПФ наведені далі.

Порядок реалізації ШПФ у РМЧ такий. Визначаються такі показники [1]:

- ширина смуги сигналу;
- частота дискретизації, f_s ;
- кількість точок ШПФ, N ;

- роздільна здатність за частотою, (f_s / N) ;
- максимальний час обчислення N -точкового ШПФ, (N / f_s) ;
- тип процесора (фіксована чи рухома крапка);
- час виконання алгоритму ШПФ за основою 2 у порівнянні з ШПФ за основою 4;
- виграш ШПФ у відношенні сигнал/шум, $[10 \log_{10}(N / 2)]$;
- вимоги зважування з використанням віконної функції.

Кількість точок ШПФ також визначає мінімальний рівень шуму ШПФ щодо рівня широкопasmового шуму, що також повинно бути враховане при виборі кількості точок ШПФ.

Необхідно звернути увагу, що виграш у відношенні сигнал/шум ШПФ визначається кількістю точок ШПФ. ШПФ діє подібно до аналогового аналізатору спектра із шириною смуги розгортки f_s/N . Збільшення кількості точок підвищує роздільну здатність ШПФ і звужує смугу частот, що пропускаються ним, скорочуючи, таким чином, мінімальний рівень шуму. У цьому аналізі ми зневажили шумом, викликаним помилкою заокруглення при реалізації ШПФ. На практиці, в АЦП, що використовується для оцифрування сигналу, формується шум квантування, що є домінуючим шумовим джерелом у системі.

Щоб уявити собі, при яких умовах здійснюється обробка сигналів у РРЧ, необхідно дослідити характеристики реально існуючих ЦСП і час реалізації ШПФ на цих процесорах. Це означає, що ШПФ повинно бути обчислене протягом часу накопичення пакета даних, рівного N/f_s . Також доцільно розглянути такі характеристики, як використання процесора з фіксованою крапкою в порівнянні з процесором з рухомою крапкою, використання алгоритму за основою 2 у порівнянні з алгоритмом за основою 4, споживана процесором потужність і вартісні показники.

8.2. Типова структура цифрового сигнального процесора

На рис. 8.1 наведена типова схема взаємодії процесора ADSP-2189M з зовнішніми вузлами при використанні повномасштабної моделі пам'яті [32].

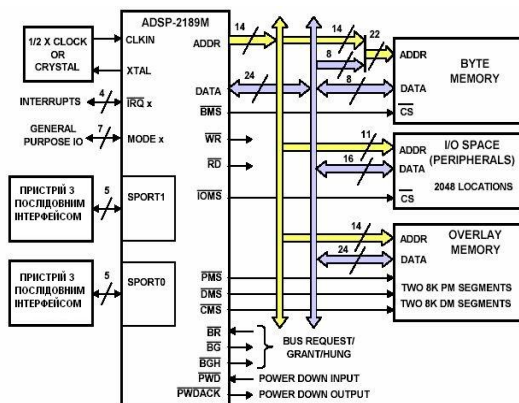


Рис. 8.1 Типова схема взаємодії процесора ADSP-2189M з зовнішніми вузлами.

Процесор підтримує чотири зовнішні переривання, сім універсальних сигналів вводу-виводу, під'єданий до 8-ми розрядної EPROM, зовнішньої пам'яті програм і даних, взаємодіє з двома зовнішніми вузлами через через послідовні інтерфейси. Один з послідовних портів може бути зконфігурований як джерело двох додаткових сигналів переривання, один універсальний вхід і один універсальний вихідний сигнал, що дає в сумі шість зовнішніх сигналів переривання, дев'ять каналів вводу-виводу загального призначення при збереженні одного повнофункціонального послідовного порту. Процесор може працювати в режимі доступу до хост-пам'яті (host memory mode), який дозволяє організувати доступ по всій ширині зовнішньої шини даних. Додаткові периферійні пристрої можуть бути під'єднані в режимі host memory mode при використанні зовнішніх апаратних засобів для генерації і фіксації додаткових адресних сигналів.

Можливість програмної генерації циклів очікування дозволяє під'єднувати швидкий процесор до більш повільних периферійних пристроїв.

8.3. Інтерфейси сигнальних процесорів

Ефективність роботи ЦСП в системі обробки сигналів залежить від організації каналів вводу-виводу [1,10]. До складу сучасних ЦСП (наприклад, ADSP-21ESP202) входять інтегровані АЦП/ЦАП, що знімає проблему організації інтерфейсу між окремими компонентами. АЦП і ЦАП оснащуються інтерфейсами, які призначені для зв'язку з ЦСП, що мінімізує і/чи усуває необхідність зовнішньої підтримки інтерфейсу або застосування інтерфейсної логіки. Високопродуктивні сігма-дельта-АЦП і ЦАП в даний час випускаються в одному корпусі (комбіноване вирішення називається КОдер/ДЕКОдер або КОДЕК), наприклад, AD73311, AD73322. Дані пристрої також розроблені з урахуванням мінімальних вимог до інтерфейсної логіки при роботі з найпоширенішими ЦСП.

Аналіз паралельного інтерфейсу з ЦСП: читання даних з АЦП, що під'єднаний до адресного простору пам'яті. Під'єднання АЦП або ЦАП через паралельний інтерфейс до ЦСП вимагає розуміння специфіки процесів читання/запису даних процесором з/в периферійних пристроїв при їх під'єднанні до адресного простору пам'яті. Спрощена блок-схема типового паралельного інтерфейсу ЦСП із зовнішнім АЦП наведена на рис. 8.2.

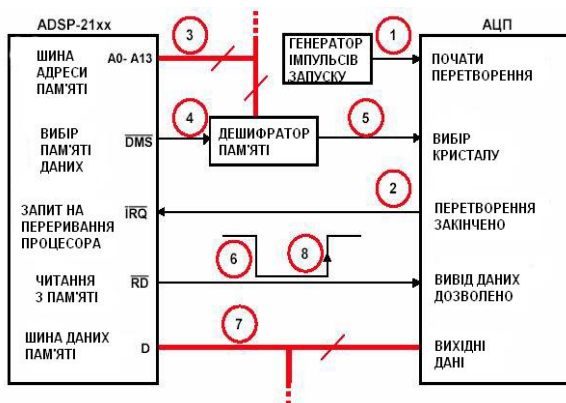


Рис.8.2. Блок-схема типового паралельного інтерфейсу ЦСП.

Вважається, що АЦП проводить вибірку з постійною частотою, яка задається зовнішнім тактовим генератором, асинхронно по відношенню до внутрішньої тактової синхронізації процесора. Використання окремого задаючого генератора для АЦП є перевагою, оскільки сигнал внутрішнього генератора процесора може мати високий рівень завад і фазовий шум (jitter), який в процесі аналого-цифрового перетворення веде до збільшення рівня шумів АЦП.

Тактовий імпульс задаючого генератора на вході “старт перетворення” (convert start) АЦП ініціює процес перетворення вхідних даних (крок 1). По передньому фронту цього імпульсу внутрішня схема вибірки-зберігання АЦП перемикається з режиму вибірки в режим зберігання, і, таким чином, починається процес перетворення. Після виконання перетворення на виході АЦП виставляється строб “перетворення виконано” (крок 2). Коли цей сигнал приходить на вхід “запиту переривання процесора” (IRQ), починається процес читання даних з АЦП. Далі процесор формує адресу периферійного пристрою, (крок 3). Одночасно процесор переводить в активний стан сигнал доступу до пам’яті (DMS) (крок N 4). Дві внутрішні шини адреси в процесорі ADSP-21xx (шина адреси пам’яті програм і шина адреси пам’яті даних) спільно використовують зовнішню шину адреси, а дві внутрішні шини даних (шина даних пам’яті програм і шина даних пам’яті даних) спільно використовують одну зовнішню шину даних. Сигнали “вибору пам’яті початкового завантаження” (BMS), “вибору пам’яті даних” (DMS), “вибору пам’яті програм” (PMS) і “вибору пам’яті пристроїв вводу-виводу” (IOMS) вказують, для якої пам’яті в даний момент використовуються зовнішні шини. Ці сигнали використовуються для зовнішньої дешифрації адреси (див. рис. 8.2). Вихідний сигнал дешифратора адреси подається на вхід “chip select” вибору периферійного пристрою (крок 5).

Часова діаграма циклу читання для процесорів ADSP-21xx наведена на рис. 8.3. Сигнал читання пам’яті (“memory read”, RD) виставляється через проміжок часу t_{ASR} після активації сигналу DMS (крок 6). Щоб повністю використовувати перевагу високої швидкодії ЦСП, сума часу затримки дешифрації адреси і часу вмикання периферійного пристрою після подачі сигналу

вибору (“chip select”) не повинна перевищувати час t_{ASR} . Сигнал RD залишається активним (низький логічний рівень) протягом часу t_{RP} . Цей сигнал використовується для переведення в активний стан паралельного виходу даних периферійного пристрою (крок 7). Сигнал RD звичайно під’єднується до відповідного виводу периферійного пристрою, що зветься сигналом “дозволу виходу” або “читання” (“output enable” або “read”). Висхідний (задній) фронт сигналу RD використовується для введення даних з шини в DSP-процесор (крок 8). Після появи висхідного фронту сигналу RD дані на шині повинні утримуватися периферійним пристроєм протягом часу t_{RDH} , званого “часом утримання даних”. Для більшості процесорів сімейства ADSP-21xx цей час рівний нулю.

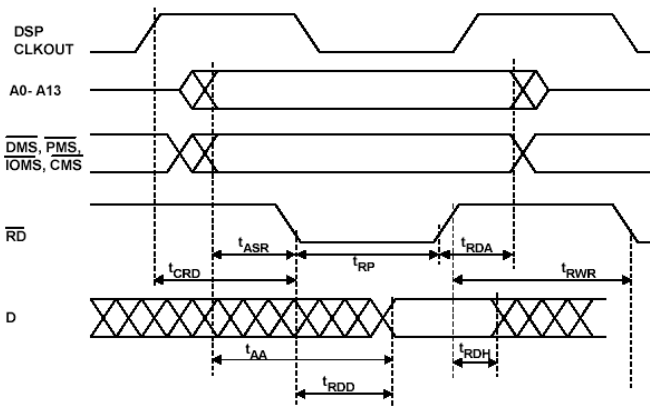


Рис. 8.3. Часова діаграма циклу читання для процесорів сімейства ADSP-21xx

t_{CK} - період тактового сигналу процесора; t_{ASR} - час установки процесором адреси і сигналу вибору пам'яті до спадаючого (переднього) фронту сигналу читання = $0.25 \cdot t_{CK}$; t_{RDD} - час від спадаючого (переднього) фронту сигналу читання до моменту достовірної установки даних - $0.5 \cdot t_{CK} + (\text{число циклів очікування}) \times t_{CK}$ (максимум); t_{RDH} - час утримання даних після висхідного (заднього) фронту сигналу читання; t_{RP} - тривалість імпульсу сигналу читання = $0.5 \cdot t_{CK} + (\text{число циклів очікування}) \times t_{CK}$ (мінімум).

Шина даних периферійного пристрою повинна підтримувати високоімпедансний Z-стан.

Час декодування адреси і час вмикання периферійного пристрою не повинен перевищувати час tASR встановлення процесором адреси і сигналу вибору пам'яті. Для здійснення доступу без режиму очікування, час від спадаючого (переднього) фронту сигналу RD до моменту достовірного встановлення даних не повинен перевищувати tRDD, інакше необхідно програмно забезпечити режим очікування або понизити частоту роботи процесора

На виході АЦП повинні підтримуватися достовірні дані протягом часу tRDH після висхідного (заднього) фронту сигналу читання RD.

Периферійний пристрій повинен працювати при якомога меншій тривалості строба tRP, інакше необхідно програмно забезпечити режим очікування або понизити частоту роботи процесора .

Параметр tRDD визначає час, необхідний для доступу до даних периферійного пристрою. Якщо бажаний час доступу до периферійного пристрою більший, то потрібно використовувати цикли очікування або зменшити тактову частоту процесора. Дані характеристики залежать від тактової частоти ЦСП, який використовуємо.

Є три спеціальні регістри для управління процесом очікування: для пам'яті початкового завантаження, для пам'яті програм і для пам'яті даних та до простору вводу-виводу. Програміст може задати від 0 до 15 тактів очікування для кожного паралельного інтерфейсу пам'яті. Кожний такт очікування збільшує час доступу до зовнішньої пам'яті на величину, рівну по тривалості одному такту генератора тактових імпульсів процесора. В даному прикладі сигнали DMS і RD утримуються незмінними протягом додаткового часу, що визначається тривалістю тактів очікування.

Спрощена схема інтерфейсу між АЦП (типу AD7854/AD7854L) і ЦСП наведена на рис. 8.4.

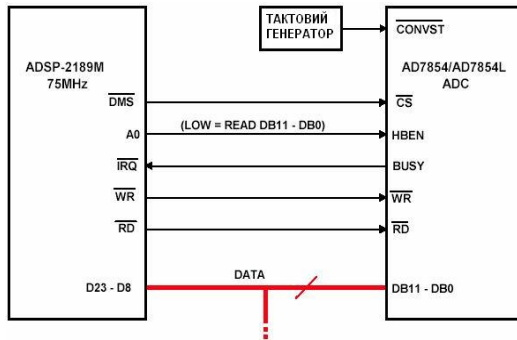


Рис. 8.4. Схема інтерфейсу між АЦП (типу AD7854/AD7854L) і ЦСП.

Як сигнал закінчення перетворення від AD7854/AD7854L використовується сигнал BUSY. Наведена конфігурація дозволяє ЦСП записувати дані в регістр управління паралельним інтерфейсом AD7854/AD7854L. Це необхідно для встановлення різних опцій в AD7854/AD7854L і виконання процесу калібрування. Проте, в звичайному режимі читання даних з AD7854/AD7854L здійснюється відповідно до наведеного вище опису.

Аналіз паралельного інтерейсу з ЦСП: запис даних в ЦАП, що під'єднаний до адресного простору пам'яті. Спрощена блок-схема інтерфейсу між ЦСП і, наприклад, ЦАП наведена на рис. 8.5. Діаграми циклу запису в пам'ять для семейства ADSP-21xx наведені на рис. 8.6.

В системах реального часу ЦАП здійснюють подвійну буферизацію даних і функціонують безперервно з постійною тактовою частотою.

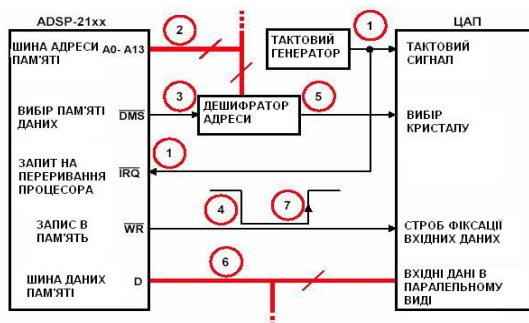


Рис. 8.5. Схема інтерфейсу між ЦСП і ЦАП.

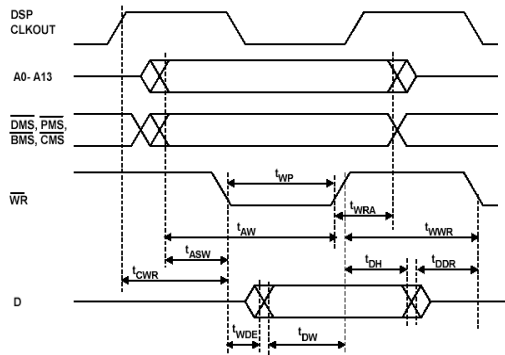


Рис. 8.6. Часова діаграма циклу запису в пам'ять.

t_{CK} - період тактового сигналу процесора; t_{ASW} - час установки процесором адреси і сигналу вибору пам'яті до спадаючого фронту сигналу запису - $0.25 \cdot t_{CK}$; t_{DW} - час від моменту установки даних до висхідного фронту сигналу запису - $0.5 \cdot t_{CK} + (\text{число циклів очікування}) \times t_{CK}$; t_{DH} - час затримки даних після висхідного фронту сигналу запису - $0.25 \cdot t_{CK}$; t_{WP} - тривалість імпульсу сигналу запису WR - $0.5 \cdot t_{CK} + (\text{число циклів очікування}) \times t_{CK}$ (мінімум).

Процес запису ініціюється ЦАП за допомогою встановлення сигналу запиту переривання процесора, який вказує, що периферійний пристрій готовий до прийому нових даних (крок 1). Далі процесор виставляє адресу ЦАП на адресній шині (крок 2) і переводить у активний стан сигнал DMS (крок 3). Це приводить до того, що дешифратор адреси видає сигнал вибору (chip select) на периферійний пристрій (крок 5). Після спадаючого фронту сигналу DMS через проміжок часу t_{ASW} процесор переводить в активний нульовий стан сигнал запису WR (крок 4). Тривалість імпульсу WR складає t_{WP} нс. Дані поступають на шину даних (D) і утримуються протягом часу t_{DW} , до переходу сигналу WR в неактивний одиничний стан (крок 6). Висхідний фронт сигналу WR використовується для фіксації присутніх на шині даних (D) в зовнішню паралельну пам'ять (крок 7). Дані на шині залишаються достовірними ще протягом часу t_{DH} після проходження позитивного фронту сигналу WR.

Спрощена схема інтерфейсу між ADSP-2189M і ЦАП AD5340 наведена на рис. 8.7. Зауважимо, що вона передбачає 2 програмних цикли очікування.

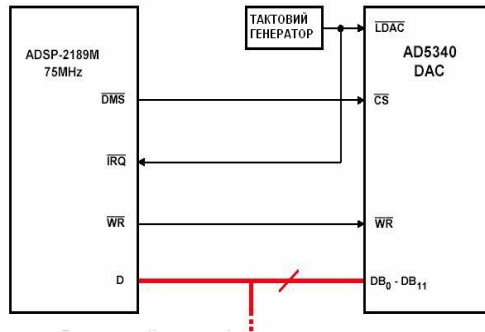


Рис. 8.7. Схема інтерфейсу між ADSP-2189M і ЦАП AD5340.

Аналіз послідовного інтерфейсу з ЦСП. Наявність послідовного порту усуває необхідність використання паралельних шин для підключення АЦП і ЦАП до ЦСП.

Структурна схема одного з двох послідовних портів процесора сімейства ADSP-21xx наведена на рис. 8.8. Передаючий (TX) і приймаючий (RX) регістри послідовного порту визначені на рівні синтаксису мови асемблера ЦСП і не відображаються в пам'яті процесора.

Характеристики послідовного порту: окремі секції прийому і передачі даних для кожного порту; подвійна буферизація регістрів прийому і передачі даних; тактові імпульси для послідовного обміну можуть генеруватися як всередині процесора, так і поступати ззовні; сигнали синхронізації фреймів можуть бути внутрішніми або зовнішніми; довжина передаваних слів може складати від 3 до 16 біт; автоматична генерація переривань. Апаратний компандер звільняє ресурс ядра процесора.

В приймальній частині послідовного порту сигнал фреймової синхронізації прийому (RFS) ініціює прийом даних. Послідовний потік даних (DR), що приймаються, від зовнішнього пристрою (АЦП), побітно записується в приймальний регістр зсуву. Для бітової синхронізації використовуються спадаючі фронти тактових імпульсів сигналу SCLK. Після завершення прийому чергового слова, воно записується в регістр RX, і послідовний порт генерує запит

переривання, по якому ядро процесора має можливість прочитати ухвалене слово з регістра RX.



Рис. 8.8. Структурна схема послідовного порту процесора сімейства ADSP-21xx

Запис в регістр TX готує послідовний порт до передачі даних. Початок передачі даних супроводиться сигналом фреймової синхронізації передачі (TFS). Потім слово з регістра TX записується у внутрішній передаючий регістр зсуву. Дані з передаючого регістра зсуву побітно посилаються на ЦАП. Для синхронізації послідовно передаючих на зовнішній пристрій даних (DT) використовуються зростаючі фронти імпульсів тактового сигналу SCLK. Після передачі першого біта послідовний порт генерує запит переривання, по якому ядро процесора може записати в регістр передачі даних нове слово, не дивлячись на те, що передача попередніх даних ще не завершена.

При нормальному режимі фреймової синхронізації сигнал RFS або TFS перевіряється по низхідному фронту тактового сигналу SCLK. Якщо у цей момент сигнал фреймової синхронізації активний, то дані доступні (в режимі передачі) або дані фіксуються в приймальному регістрі зсуву (в режимі прийому) по низхідному фронту наступного тактового імпульсу сигналу SCLK. Сигнал фреймової синхронізації не перевіряється далі до закінчення передачі або прийому всього слова, що залишилося. При альтернативному режимі фреймової синхронізації сигнал фреймової синхронізації встановлюється в тому ж самому

такті сигналу SCLK, що і перший біт слова. Біти даних фіксуються по низхідному фронту сигналу SCLK, але сигнал фреймової синхронізації перевіряється тільки в такті, що відповідає першому біту. Генерований всередині сигнал фреймової синхронізації залишається в активному стані на весь час прийому або передачі послідовного слова.

Послідовні порти процесорів сімейства ADSP-21xx універсальні. Сигнали TFS, RFS або SCLK можуть або генеруватися вбудованим генератором процесора сімейства ADSP-21xx (режим master), або поступати від зовнішнього джерела (режим slave). Полярність цих сигналів може бути змінена програмно, ще більш підвищуючи таким чином гнучкість інтерфейсу. Порт також містить апаратні засоби стиску з μ -і А-характеристикою для голосових телекомунікаційних додатків.

Часові діаграми роботи послідовного порту процесора ADSP-2189M, працюючого в режимі прийому (альтернативний режим фреймової синхронізації), наведені на рис. 8.9. Перший (низхідний) фронт сигналу SCLK, наступний після негативного (низхідного) фронту сигналу RFS, синхронізує фіксацію старшого біта даних (MSB) від АЦП у вхідному регістрі звуку процесора. Процес продовжується до тих пір, поки всі послідовні біти не будуть по черзі прийняті у вхідному регістрі звуку. Основні часові характеристики, на які потрібно звернути увагу, це час встановлення послідовних даних (t_{SCS}) і час їхнього утримання (t_{SCH}) по відношенню до фронтів сигналу SCLK.

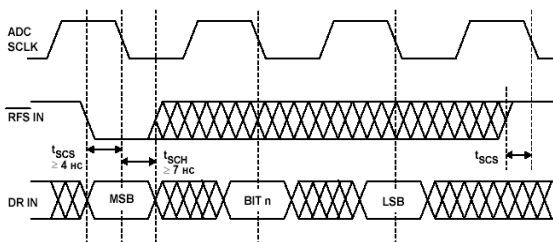


Рис. 8.9. Часові діаграми роботи послідовного порту процесора ADSP-2189M

Часова діаграма запису в пам'ять процесора наведена на рис. 8.10.

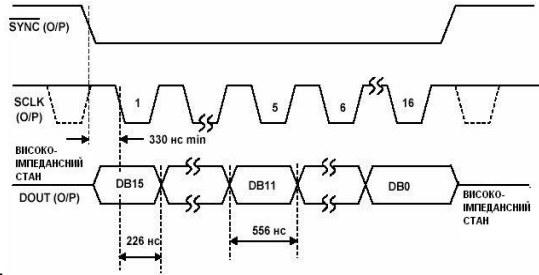


Рис. 8.10. Часова діаграма запису в пам'ять процесора ADSP-2189M

Схема послідовного інтерфейсу між АЦП AD7853/AD7853L і ADSP-2189M наведена на рис. 8.11. Схема функціонує в режимі передачі даних від АЦП до ЦСП (альтернативний режим фреймової синхронізації, АЦП працює в режимі "майстер"). Мікросхема AD7853/AD7853L має внутрішні регістри, які доступні для запису з сторони ЦСП через послідовний порт. Ці регістри використовуються для встановлення різних режимів роботи АЦП AD7853/AD7853L, а також для ініціалізації процесу калібрування.

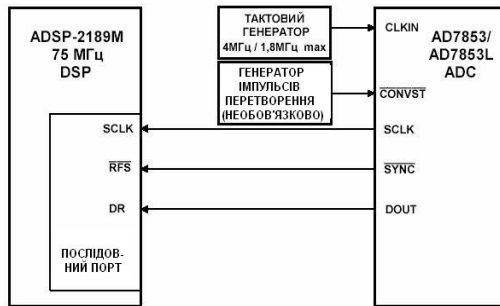


Рис. 8.11. Схема послідовного інтерфейсу між АЦП AD7853/AD7853L і ADSP-2189M

Організація стандартного інтерфейсу між процесором ADSP-2189M і ЦАП AD5322 наведена на рис. 8.12.

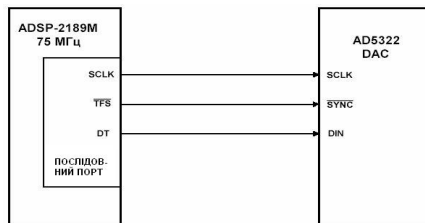


Рис. 8.12 Схема інтерфейсу між процесором ADSP-2189M і ЦАП AD5322

Зауважимо, що сигнал синхронізації для AD5322 задається генератором тактового сигналу процесора ADSP-2189M. Також існує можливість зовнішньої по відношенню до AD5322 генерації сигналів SCLK і SYNC і їхнього використання для синхронізації процесора ADSP-2189M. Послідовний інтерфейс AD5322 недостатньо швидкий для роботи з ADSP-2189M на максимальній швидкості, яку може забезпечити процесор. Проте частота сигналу синхронізації послідовного порту процесора може бути запрограмована на відповідну швидкість, підтримувану швидкими або повільними ЦАП.

Вхідний регістр зсуву в мікросхемі AD5322 має розрядність 16 біт. 16-розрядне слово складається з чотирьох бітів управління, за якими йде 12 біт даних для ЦАП.

Перший відісланий біт визначає, для якого з двох наявних ЦАП (А або В) призначені дані. Другий біт визначає використання буферизованого або небуферизованого режиму роботи входу опорної напруги. Наступні два біти керують режимами роботи ЦАП. На рис. 8.13 наведена схема інтерфейсу між АЦП AD9201, ЦАП AD9761 і процесором ADSP-21065L

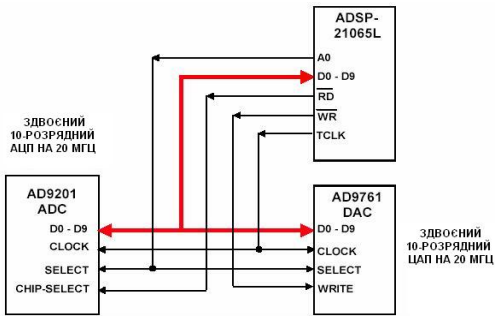


Рис. 8.13. Схема інтерфейсу між АЦП AD9201, ЦАП AD9761 і процесором ADSP-21065L.

8.4. Завдання для самоперевірки

1. Обчислення алгоритму ШПФ в режимі реального часу виконується:

А) на протязі періоду нагромадження одного пакету даних; Б) при поступленні $N/2$ даних; В) без врахування операцій вводу/виводу даних; Г) на протязі часу опрацювання даних на одному метелику; Д) немає правильної відповіді.

2. Ширина смуги сигналу – 10МГц. Розмірність ШПФ – 1024 точки. Час виконання алгоритму ШПФ становить:

А) 1024 мкс.; Б) 102,4 мкс.; В) 512 мкс.; Г) 51,2 мкс; Д) немає правильної відповіді.

3. Ширина смуги сигналу – 12,8 МГц. Час виконання алгоритму ШПФ дорівнює 10 мкс. Роздільна здатність за частотою становить:

А) 1,28 МГц; Б) 2,56 МГц; В) 1МГц; Г) 0,1 МГц; Д) немає правильної відповіді

4. Послідовність взаємодії між процесором ЦСП і АЦП засобами паралельного інтерфейсу така:

А) завершення перетворення АЦП – формування адреси і сигналу вибірки АЦП процесором – формування сигналу “Запис” процесором – надходження даних з АЦП на процесор; Б) завершення перетворення АЦП – формування адреси і сигналу вибірки АЦП процесором – формування сигналу “Читання”

процесором – надходження даних з АЦП на процесор; В) завершення перетворення АЦП – формування адреси АЦП процесором – надходження даних з АЦП на процесор; Г) формування адреси АЦП процесором – формування сигналу “Запис” процесором – надходження даних в АЦП; Д) немає правильної відповіді.

5. В послідовному інтерфейсі ЦСП тактові сигнали: А) генеруються всередині або поступають ззовні; Б) поступають тільки ззовні; В) генеруються тільки всередині; Г) можуть бути відсутніми; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1). А; 2) Г 3) Г; 4) Б; 5) А

РОЗДІЛ 9. ПРОЕКТУВАННЯ ПРОЦЕСОРА ШПФ НА ЦИФРОВИХ СИГНАЛЬНИХ ПРОЦЕСОРАХ

9.1. Процедури визначення параметрів процесора

Показники, що використовуються для реалізації ШПФ, наведені в п. 8.1 даного посібника. При заданих всіх достатніх загальних характеристик (параметри сигналу, необхідна кількість точок ШПФ, час обчислення) необхідно вибрати основу ШПФ та тип процесора.

З погляду кількості необхідних операцій ефективнішими є алгоритми ШПФ за більшою та за розщепленою основами. У загальному випадку збільшення основи алгоритму веде до зменшення обчислювальних витрат, але при цьому ускладнюється реалізація алгоритму. В [24] наведені характеристики стосовно кількості операцій для різного типу основ.

Розглянемо реалізацію алгоритму ШПФ за основою 4. Граф 64-точкового ШПФ за основою 4 з прорідженням за часом наведений на рис. 9.1. Граф базової операції (метелик) алгоритму ШПФ з прорідженням за часом наведений на рис. 9.2. Граф базової операції (метелик) алгоритму ШПФ з прорідженням за частотою наведений на рис. 9.3.

Алгоритми ШПФ за основою 4 дозволяють на 25% скоротити обчислювальні затрати порівняно з алгоритмами ШПФ за основою 2. Подібним чином будуються алгоритми за основою 8, 16, а також алгоритми за змішаною основою 2-4, 2-4-8 і т.д.

Розглянемо більш детально алгоритми ШПФ за розщепленою основою двачотири. В них граф одержується комбінацією графів алгоритмів ШПФ за основою 2 та ШПФ за основою 4, де N -точкове ДПФ розбивається на одне $N/2$ -точкове і два $N/4$ -точкові перетворення.

Граф базової операції (метелик) алгоритму ШПФ за основою 2-4 з прорідженням за часом наведений на рис. 9.4. Граф базової операції (метелик)

алгоритму ШПФ за основою 2-4 з прорідженням за частотою наведений на рис. 9.5.

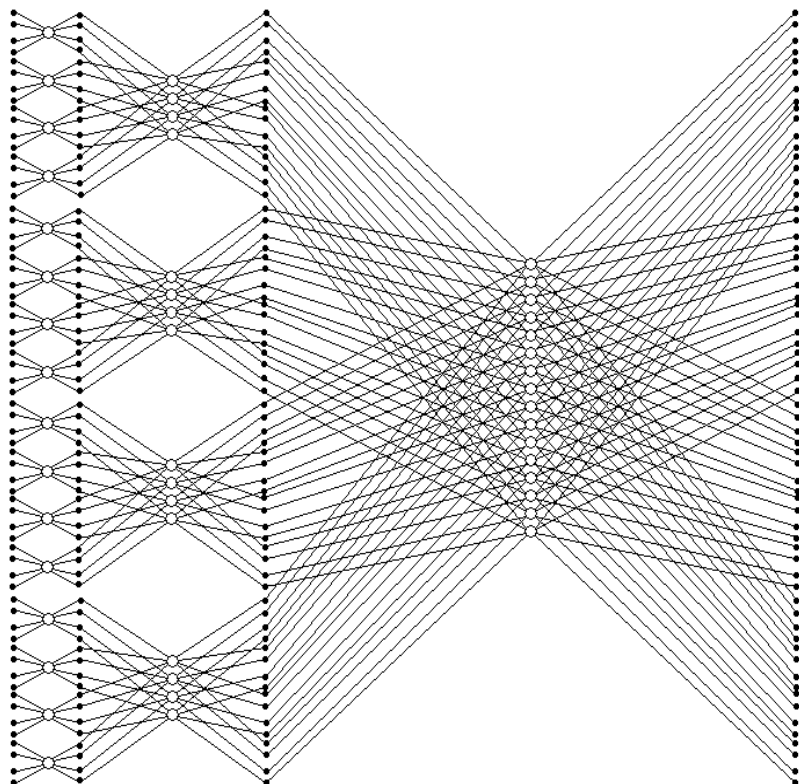


Рис. 9.1. Граф 64-точкового ШПФ за основою 4 з прорідженням за часом.

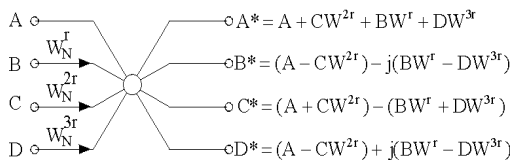


Рис. 9.2. Граф базової операції (метелик) алгоритму ШПФ з прорідженням за часом.

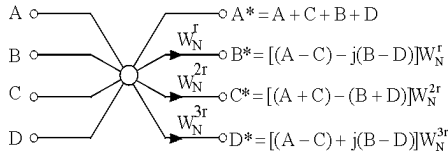


Рис. 9.3. Граф базової операції (метелик) алгоритму ШПФ з прорідженням за частотою.

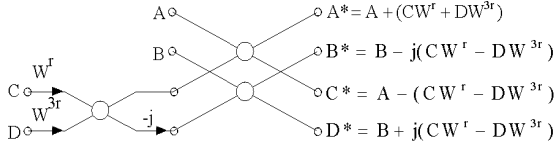


Рис. 9.4. Граф базової операції (метелик) алгоритму ШПФ за основою 2-4 з прорідженням за часом.

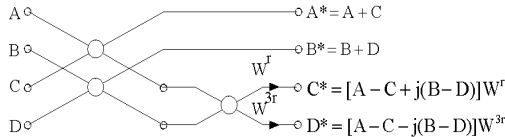


Рис. 9.5. Граф базової операції (метелик) алгоритму ШПФ за основою 2-4 з прорідженням за частотою.

Алгоритми ШПФ за основою 2-4 вдало поєднують простоту структури алгоритмів за основою два з ефективністю алгоритмів з високою основою. Зберігаючи в основному структуру алгоритмів ШПФ за основою 2, вони мають найменші обчислювальні затрати серед розглянутого класу алгоритмів.

Недоліком даних класів алгоритмів є використання комплексних фазових множників. В алгоритмі ШПФ Рейдера-Бреннера використовуються фазові множники, що мають структуру дійсних чисел.

Відзначимо, що відліки вхідної послідовності переходять у відповідні ДПФ нульового рівня відповідно до інверсії їхній двійкових номерів (операція називається перестановкою вхідних відліків, біт-інверсія). Наприклад, десятковий

номер $4|_{10}$ у двійковому вигляді запишеться як $100|_2$. Інверсія числа $100|_2$ (у прочитанні справа наліво) запишеться як $001|_2 = 1|_{10}$. Таким чином, вхідний відлік під номером 4 співпадає з першим ДПФ $X_{0,1}(0)$. Перестановку для всіх відліків можна показати стрілками переходу:

$$0 \rightarrow 0, 1 \rightarrow 4, 2 \rightarrow 2, 3 \rightarrow 6, 4 \rightarrow 1, 5 \rightarrow 5, 6 \rightarrow 3, 7 \rightarrow 7.$$

9.2. Основи вибору сигнальних процесорів

Вибір ЦСП залежить, насамперед, від сфери використання, наприклад:

- для таких портативних пристроїв, як мобільні телефони, портативні цифрові плеєри, вартість, ступінь інтеграції і споживана потужність є першорядними, а максимальна продуктивність часто не потрібна (тому що зазвичай тягне за собою значне підвищення споживаної потужності);

- для гідроакустичних або радіолокаційних систем визначальними параметрами є швидкість роботи, наявність високошвидкісних інтерфейсів і зручна система розробки, а вартість є другорядним критерієм.

Крім того, у багатьох випадках, потрібно враховувати місце на ринку, займане постачальником процесора, та наявність інструментальних, діагностичних засобів підтримки розробки.

При виборі процесора враховуються такі параметри [1,32].

Формат даних і розрядність. Вибираючи формат даних, необхідно мати на увазі наступне: ЦСП з фіксованою крапкою дешевші і точніші при рівній розрядності (на мантиссу в 32-бітному процесорі з фіксованою крапкою відводяться всі 32 біти, а в процесорі з рухомою крапкою – тільки 24).

Крім того, при виборі ЦСП необхідно ретельно аналізувати алгоритм обробки і параметри вхідних сигналів для правильного вибору розрядності і типу арифметики. Іноді при неможливості підібрати відповідний процесор з рухомою крапкою (через більшу його вартості або енергоспоживання) використовують ЦСП з фіксованою крапкою і стиснення динамічного діапазону оброблюваних

сигналів (компресію), однак це призводить до збільшення складності алгоритму обробки сигналу і підвищує вимоги до швидкодії.

Швидкість. Одна з найчастіших помилок розробника – ототожнення тактової частоти і швидкодії, що в більшості випадків неправильно. Дуже часто швидкість роботи вказують в MIPS (мільйонів інструкцій у секунду).

Однак проблема порівняння швидкості різних ЦСП полягає в тому, що процесори мають різні системи команд, і для виконання одного і того ж алгоритму різними процесорами потрібна різна кількість цих команд. Крім того, іноді для виконання різних команд одним процесором потрібна різна кількість тактів синхронізації. В результаті процесор зі швидкістю 1000 MIPS цілком може виявитися в рази повільніше процесора зі швидкістю 300 MIPS, особливо при різній їх розрядності.

Одне з рішень цієї проблеми – порівнювати процесори за швидкістю виконання певних операцій, наприклад, операції множення з накопиченням (MAC). Найбільш точною є оцінка швидкості виконання певних алгоритмів – наприклад, КІХ- і НІХ-фільтрації, однак це вимагає розробки відповідних програм і ретельного аналізу результатів тестування.

Організація пам'яті. Організація системи пам'яті процесора впливає на продуктивність. Це пов'язано з тим, що ключові команди ЦСП є багатооперандними і прискорення їх роботи вимагає одночасного читання декількох комірок пам'яті. Більшість ЦСП з фіксованою крапкою, що застосовуються у вбудованих додатках, мають малий обсяг внутрішньої пам'яті, зазвичай від 4 до 256 Кбайт, і невисоку розрядність зовнішніх шин даних.

ЦСП з рухомою крапкою, зазвичай, опрацьовують великі масиви даних і складні алгоритми, тому мають або вбудовану пам'ять великого обсягу, або велику розрядність адресних шин для підключення зовнішньої пам'яті (а іноді і те й інше).

Зручність розробки додатків. Зручність для розробника (зазвичай пов'язується з використанням при програмуванні ЦСП мов високого рівня) в більшості випадків обертається отриманням менш компактного і швидкого коду і

приводить до необхідності використання більш потужних і дорогих процесорів. З іншого боку, в сучасних умовах, швидкість розробки (і, отже, виходу нового виробу на ринок) може принести більше вигод, ніж витрати часу на оптимізацію коду при написанні програми на асемблері.

Крім того, слід пам'ятати, що безпомилкових програм не буває, тому кошти налагодження і можливість корекції програм в готовому пристрої дуже часто мають першорядне значення. У той же час при виборі ЦСП і засобів розробки необхідно враховувати деякі особливості архітектури процесорів.

Майже всі виробники пропонують широкий набір так званих “стартер-кітів” і “оціночних модулів”, забезпечують розробників симуляторами й емуляторами своїх ЦСП.

Енергоспоживання. Для зниження енергоспоживання використовуються різні методи, зокрема, зменшення напруги живлення і введення функцій управління споживанням, наприклад, динамічної зміни тактової частоти, перемикання в сплячий або режим очікування чи відмикання невикористовуваної в даний момент периферії. Такі заходи суттєво впливають на швидкість роботи процесора.

Вартість. При виборі процесора за критерієм вартості необхідно брати до уваги вартість зовнішніх компонентів (наприклад, ЦПС з вбудованою пам'яттю достатнього об'єму коштує дорожче від аналогічного без вбудованої пам'яті, але ціна пристрою в цілому на його основі може бути значно нижчою через відсутність інших компонентів і меншого розміру друкованої плати). Дуже значущим фактором, що впливає на вартість ЦСП, є тип корпусу: Інтегральні схеми в керамічних корпусах, розраховані на промислові або спеціальні умови експлуатації, коштують значно дорожче від таких же схем, які працюють в комерційному діапазоні температур.

Методологія вибору процесора. Правильний вибір ЦСП сильно залежить від програми: процесор може добре підходити для одних додатків, але абсолютно не підходити для інших. При виборі процесора потрібно визначити найважливіші в конкретному випадку характеристики і розставити їх за ступенем важливості.

Потім відповідно до цих критеріїв відібрати можливих кандидатів і, нарешті, вибрати з відповідних кращий, звертаючи увагу на додаткові, не критичні характеристики. При цьому доцільно скористатися оцінкою характеристик процесорів, виробленої будь-якої авторитетною компанією (наприклад, VTDI).

Пріоритети. Якщо необхідна висока швидкість і низька ціна доцільно використовувати ЦСП фірми Texas Instruments. При необхідності низького енергоспоживання, при готовності пожертвувати швидкістю, використовуються процесори фірми Analog Devices, зокрема, сімейства TigerSHARC.

9.3. Проектування процесора ШПФ

Розглянемо процедуру проектування процесора ШПФ для вхідних даних, які наведені далі:

<i>Тип процесора</i>	<i>ADSP-21060</i>
<i>Кількість точок</i>	<i>256</i>
<i>Основа ШПФ</i>	<i>4</i>
<i>Прорідження</i>	<i>часове</i>
<i>Розрядність вхідних даних</i>	<i>32</i>
<i>Такт поступлення вхідних даних</i>	<i>20 нс</i>
<i>Час опрацювання</i>	<i>0,5 мс</i>

ADSP-21060 SHARC – 32-х розрядний процесор опрацювання сигналів з супергарвардською архітектурою, є системою на кристалі, оскільки базується на ядрі сімейства ADSP-21000, має двохпортову пам'ять на чипі SRAM, інтегровані зовнішні пристрої вводу-виводу з підтримкою розподілених шин вводу-виводу. Завдяки кеш інструкцій процесор може виконувати практично кожну команду за один цикл. Чотири незалежних шини для даних, команд та шин вводу-виводу плюс перехресні переключення між зв'язками пам'яті, яка може бути сконфігурована максимум з 128 Кслів по 32 біти, з 256 Кслів по 16 біт для даних, та 80 Кслів по 48 біт для інструкцій, чи з різних комбінацій довжин слів, які

поміщаються у 4М біт. Але до пам'яті можна звертатися лише по шинах, які мають 16, 32, чи 48 біт.

У режимі мультиопрацювання дозволяється підключення шістьох процесорів ADSP-21060 і паралельне їх функціонування. Єдиний адресний простір дозволяє прямі міжпроцесорні доступи до внутрішньої пам'яті кожного ADSP-21060. Розподілена логіка арбітражу шини керує даним режимом і розподіляє ресурси процесорів між собою. Арбітраж надає шині вибраний чи встановлений пріоритет. Підтримується стандарт IEEE1149.1 Joint Test Action Group (JTAG) для відлагодження системи.

Алгоритм базової операції ШПФ за основою 4 і проріджуванням за часом (див.рис. 9.2) можна представити так:

$$A'_1 = A_1 + A_2W_1 + A_3W_2 + A_4W_3 = (A_1 + A_3W_2) + (A_2W_1 + A_4W_3),$$

$$A'_2 = A_1 \Delta jA_2W_1 - A_3W_2 \pm jA_4W_3 = (A_1 - A_3W_2) \Delta j(A_2W_1 - A_4W_3),$$

$$A'_3 = A_1 - A_2W_1 + A_3W_2 - A_4W_3 = (A_1 + A_3W_2) - (A_2W_1 + A_4W_3),$$

$$A'_4 = A_1 \pm jA_2W_1 - A_3W_2 \Delta jA_4W_3 = (A_1 - A_3W_2) \pm j(A_2W_1 - A_4W_3),$$

де A'_1, A'_2, A'_3, A'_4 – результати базової операції; A_1, A_2, A_3, A_4 – вхідні відліки; W_1, W_2, W_3 – комплексні коефіцієнти; j – уявна одиниця, верхній знак перед j відповідає прямому, нижній – оберненому ШПФ.

$$ReA'_1 = [ReA_1 + (ReA_2*ReW_2 - ImA_3*ImW_2)] + [(ReA_2*ReW_1 - ImA_2*ImW_1) + (ReA_4*ReW_3 - ImA_4*ImW_3)],$$

$$ImA'_1 = [ImA_1 + (ReA_3*ImW_2 + ImA_3*ReW_2)] + [(ReA_2*ImW_1 + ImA_2*ReW_1) + (ReA_4*ImW_3 + ImA_4*ReW_3)],$$

$$ReA'_2 = [ReA_1 - (ReA_3*ReW_2 - ImA_3*ImW_2)] \pm [(ReA_2*ImW_1 + ImA_2*ReW_1) - (ReA_4*ImW_3 + ImA_4*ReW_3)],$$

$$ImA'_2 = [ImA_1 - (ReA_3*ImW_2 + ImA_3*ReW_2)] \Delta [(ReA_2*ReW_1 - ImA_2*ImW_1) - (ReA_4*ReW_3 - ImA_4*ImW_3)],$$

$$ReA'_3 = [ReA_1 + (ReA_3*ReW_2 - ImA_3*ImW_2)] - [(ReA_2*ReW_1 - ImA_2*ImW_1) + (ReA_4*ReW_3 - ImA_4*ImW_3)],$$

$$ImA'_3 = [ImA_1 + (ReA_3*ImW_2 + ImA_3*ReW_2)] - [(ReA_2*ImW_1 + ImA_2*ReW_1) + (ReA_4*ImW_3 + ImA_4*ReW_3)],$$

$$ReA'_4 = [ReA_1 - (ReA_3 * ReW_2 - ImA_3 * ImW_2)] \text{ \& } [(ReA_2 * ImW_1 + ImA_2 * ReW_1) - (ReA_4 * ReW_3 - ImA_4 * ReW_3)],$$

$$ImA'_4 = [ImA_3 - (ReA_3 * ImW_2 + ImA_3 * ReW_2)] \pm [(ReA_2 * ReW_1 - ImA_2 * ImW_1) - (ReA_4 * ReW_3 - ImA_4 * ImW_3)].$$

Для виконання базової операції слід виконати 12 операцій множення і 22 додавання. В табл. 9.1 наведено порядок проходження відліків в ярусах.

Табл. 9.1. Порядок проходження відліків в ярусах

I	II	III	IV
0	0	0	0
64	16	4	1
128	32	8	2
192	48	12	3
1	1	1	4
65	17	5	5
129	33	9	6
193	49	13	7
...			
62	206	242	248
126	222	246	249
190	238	250	250
254	254	254	251
63	207	243	252
127	223	247	253
191	239	251	254
255	255	255	255

Розрахунок основних параметрів.

Частота роботи процесора: $f = 40\text{МГц}$, звідси цикл виконання команди:

$$T_{в.к.} = \frac{1}{f} = \frac{1}{40} \cdot 10^{-6} = 0.025 \cdot 10^{-6} = 25 \cdot 10^{-9} (с) = 25 (нс).$$

Визначимо кількість “метеликів”, де

- $base$ – основа базової операції “метелик”, $base=4$;
- N – кількість точок вхідного перетворення, $N=256$;
- $N_{ет} = \log_{base} N$ – кількість етапів перетворення;
- $N_{на1ет} = \frac{N}{base}$ – кількість базових операцій “метелик” на одному етапі;
- $N_{заг} = \frac{N}{base} \cdot \log_{base} N$ – кількість базових операцій у всьому перетворенні.

$$N_{ет} = \log_{base} N = \log_4 256 = 4$$

$$N_{на1ет} = \frac{N}{base} = \frac{256}{4} = 64$$

$$N_{заг} = \frac{N}{base} \cdot \log_{base} N = \frac{256}{4} \cdot \log_4 256 = 64 \cdot 4 = 256$$

Для виконання базової операції “метелик” необхідно:

- 12 операцій множення;
- 22 операцій додавання;
- 14 операцій читання з пам’яті:
 - $4 \cdot 2 = 8$ (операцій для читання дійсної та уявної частини вхідних відділів);
 - $3 \cdot 2 = 6$ (операцій для читання дійсної та уявної частини комплексних коефіцієнтів);
- 8 операцій запису:
 - $4 \cdot 2 = 8$ (операцій для запису дійсної та уявної частини вхідних відділів).

В результаті на одну базову операцію алгоритму, який розглядаємо, припадає 56 операцій: $N_{на1мет} = 56$ (оп).

Тривалість виконання обчислення ШПФ:

$$T_{за:ШПФ} = N_{заг} \cdot N_{на1мет} \cdot T_{в.к.} = 256 \cdot 56 \cdot 25 \cdot 10^{-9} = \cdot 10^{-9} = 358400 \cdot 10^{-9} = 0,3584 \cdot 10^{-3} = 0,3584 \text{ мс}$$

Тривалість надходження даних у процесор для обробки:

$$T_{надх} = 20 \text{ нс} - \text{такт надходження даних.}$$

Тривалість надходження даних у процесор та тривалість обчислення ШПФ описуються таким виразом:

$$T = T_{\text{заг.надх}} + T_{\text{заг.ШПФ}} = 0,3584 + 0,00512 = 0,36352\text{мс}$$

Ця величина менша за заданий час обробки: $T < 0,5\text{мс}$, тобто для виконання обчислення достатньо одного процесора. Часові параметри роботи процесора наведені на рис. 9.6.

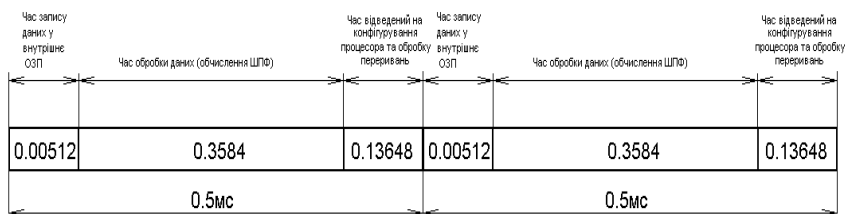


Рис. 9.6. Часові параметри роботи процесора.

Об'єм внутрішнього ОЗП процесора ADSP-21060 4 Мбіта, тобто пам'ять може бути сконфігурована таким чином: 2^{18} слів по 16 розрядів ($4\text{ Мб} = 2^{22}$ біт). Для розв'язання поставленої задачі необхідно 512 слів по 16 розрядів. $512 \times 16 = 2^9 \times 2^4 = 2^{13}$. Тобто внутрішнього ОЗП цілком вистачає. Для зберігання повертаючих множників необхідно ПЗП об'ємом: 1536×16 , оскільки на 1 операцію метелик необхідно 3 повертаючих множники, кожен з яких містить дійсну та уявну частину; всього є 256 базових операцій, розрядність операндів – 16. Тому $3 \times 2 \times 256 = 1536$. Окремо використовуємо завантажувальну пам'ять, де буде зберігатися лістинг програми. Необхідна зовнішня ОЗП розміром 512×16 для постійного приймання даних, що надходять.

9.4. Завдання для самоперевірки

1. Кількість операцій дійсних чисел при виконанні одного “метелика” в алгоритмі ШПФ за основою 2 становить:

А) 1 операція множення + 1 операція додавання/віднімання; Б) 2 операції множення + 2 операції додавання/віднімання; В) 4 операції множення + 3 операції додавання/віднімання; Г) 4 операції множення + 6 операцій додавання/віднімання; Д) немає правильної відповіді.

2. Для обчислення алгоритму ШПФ з кількістю точок 1024 за основою 4 необхідна така кількість “метеликів”:

А) 1024; Б) 1280; В) 512; Г) 640; Д) немає правильної відповіді.

3. В операції “метелик” кількість повертаючих множників при виконанні алгоритмів ШПФ за основами 2 і 4 є:

А) однаковою; Б) більшою за основою 4 на 3; В) більшою за основою 4 на 2; Г) більшою за основою 2; Д) немає правильної відповіді

4. При виконанні алгоритму ШПФ десятковий номер вхідної послідовності дорівнює 230. Значення біт інверсії є:

А) 11100110; Б) 11001101; В) 10011011; Г) 01100111; Д) немає правильної відповіді.

5. На вхід процесора ШПФ по послідовному інтерфейсі поступають 16-ти розрядні дані з тактом 4 нс. Розмірність ШПФ $N=1024$. Тривалість надходження даних становить: А) 16384; Б) 8192; В) 4096; Г) 2048; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1). Г; 2) Б 3) В; 4) Г; 5) А.

РОЗДІЛ 10. ПРОЕКТУВАННЯ ЗАСОБІВ ОПРАЦЮВАННЯ СИГНАЛІВ НА ПЛІС

10.1. Однокристална реалізація на ПЛІС алгоритму ШПФ

Для реалізації високошвидкісних пристроїв ЦОС використовують різні типи ПЛІС [34], зокрема, Virtex, Virtex-E, XC4000XL/XLA/XV, Spartan/XL та інших сімейств. Привабливою рисою ПЛІС для реалізації алгоритмів ЦОС є наявність внутрішнього швидкодіючого розподіленого ОЗП, вбудованих вузлів обчислення ШПФ тощо.

На рис. 10.1 наведена структурна схема вузла реалізації алгоритму ШПФ на ПЛІС. Вхідний ОЗП використовується для завантаження вхідної послідовності, збереження результатів проміжних обчислень і вивантаження результатів перетворення, буферний ОЗП – для збереження результатів проміжних обчислень; у ПЗП (ROM) зберігаються значення коефіцієнтів W . Застосування двох ОЗП дозволяє одночасно виконувати операції читання і запису та забезпечити коректність опрацювання даних. Блок “метелик” виконує обчислювальні дії відповідно до схеми, наведеної в розділі 2 (див. рис. 2.4). Кількість помножувачів у загальному випадку може бути різною – від 1 до 4. Блок керування формує сигнали синхронізації і необхідні сигнали керування.

Наприклад, якщо обчислення алгоритму ШПФ при $N = 256$ відбувається за 8 кроків, то при апаратній реалізації на ПЛІС потрібно додати ще 2 кроки – крок завантаження вхідних даних і крок вивантаження результатів перетворення. Таким чином, повне перетворення відбувається за 10 кроків:

- крок 1 – запис вхідної послідовності у вхідний ОЗП відповідно до двійкової інверсії номерів;

- крок 2 – перша ступінь перетворення. Дані зчитуються з вхідного ОЗП, перетворюються і записуються в буферний ОЗП;

- крок 3 – друга ступінь перетворення. Дані зчитуються з буферного ОЗП, перетворюються і записуються у вхідний ОЗП;
- виконання кроків 4, 6, 8 налогічно виконанню кроку 2;
- виконання кроків 5, 7, 9 аналогічно виконанню кроку 3;
- крок 10 – вивантаження отриманого перетворення з вхідного ОЗП.

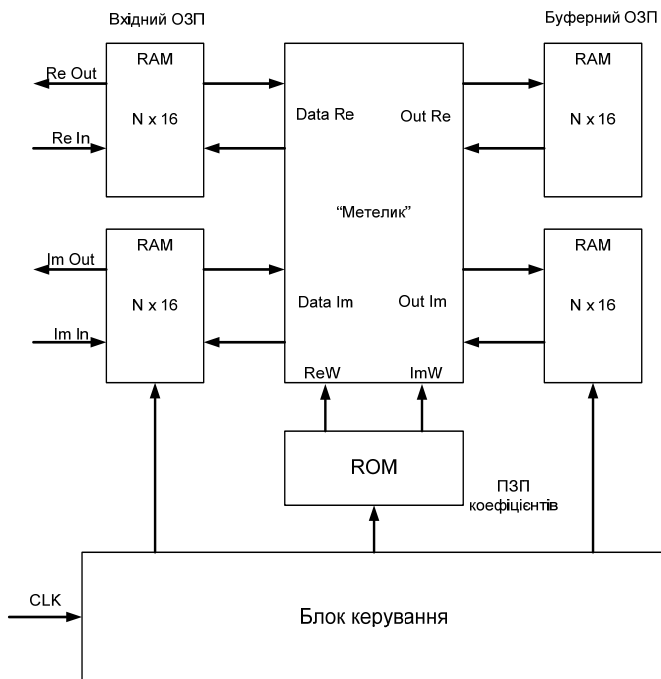


Рис. 10.1. Узагальнена структурна схема виконання алгоритму ШПФ на ПЛІС.

10.2. Оцінка продуктивності вузла реалізації алгоритму ШПФ на ПЛІС

Швидкодія виконання алгоритму ШПФ на ПЛІС визначається в N_{MAC} (кількість операцій типу множення-накопичення) за такою формулою:

$$N_{MAC} = \frac{N_{ми} * f_{відл}}{N_{точ}}$$

де N_{MAC} – кількість операцій типу множення-накопичення, c^{-1} ;

- N_{mn} – кількість множень, необхідних для обчислення перетворення;
- $f_{вхід}$ – частота надходження вхідних даних, Гц;
- $N_{точ}$ – розмірність перетворення.

Тоді, наприклад, для обчислення ШПФ розмірністю 256 точок за основою 2 з комплексними вхідними даними потрібно приблизно 3 тис. множень і 5,5 тис. додавань дійсних чисел; для 1024-точкового ШПФ за основою 2 – приблизно 16 тис. множень і 28,5 тис. додавань. При надходженні вхідних даних з частотою 40 МГц продуктивність вузла обчислення ШПФ розмірністю 256 точок повинна складати не менш як 460 млн МАС/сек., вузла обчислення ШПФ розмірністю 1024 точки – не менш 620 млн МАС/сек.

При реалізації ковзного ШПФ для обчислень на кожному кроці використовується окремий закінчений вузол, забезпечується конвеєризація в межах не тільки одного кроку, але і всього модуля. При цьому час перетворення буде рівним часу обчислень на одному кроці.

10.3. Приклад розробки процесора ШПФ на ПЛІС

Завдання полягає в такому:

Спроекувати процесор для обчислення 16-ти точкового ШПФ за основою 2 та з прорідженням за частотою на ПЛІС фірми Xilinx. Розрядність вхідних даних – 32 (по 16 розрядів дійсна й уявна частини).

При апаратній реалізації графу ШПФ виникають незручності через неспівпадіння адрес комірок пам'яті з яких потрібно вчитувати елементи на кожному ярусі. Тому на рис. 10.2. наведений граф, де для кожної базової операції на будь-якому ярусі дані беруться і записуються за тими самими адресами, що й на попередньому ярусі (із заміщенням).

Спрощена проекція потокового графу ШПФ на вертикальну площину показана на рис. 10.3, де використані такі позначення:

- *Memory array* – масив елементів пам'яті для зберігання проміжних результатів опрацювання;

- *MUX* – вхідні мультиплексори для вибору вхідних даних на першому циклі, та проміжних даних не решті циклах роботи пристрою;

- $F1, F2, \dots, F8$ – вузли обробки (двохточкові ШПФ)

На рис. 10.3 не наведені повертаючі множники та виходи пристрою (проілюстровано лише потік даних при обробці).

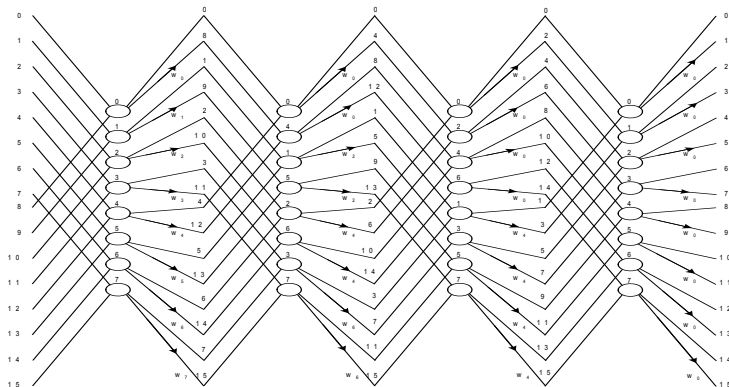


Рис. 10.2. Модифікований граф 16-ти точкового ШПФ за основою 2 з прорідженням за частотою.

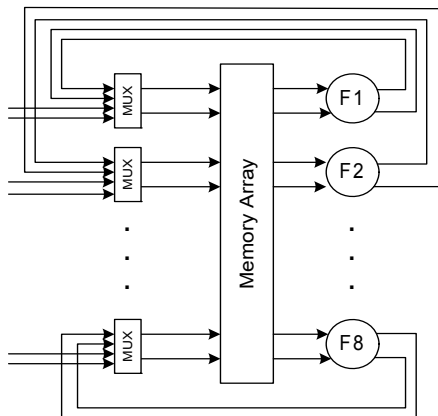


Рис. 10.3. Спрощена проекція графа на вертикальну площину.

Формат вхідних та вихідних даних.

Вхідні та вихідні дані є комплексними числами виду $A = a_r + jb_i$, де a_r – дійсна частина, b_i – уявна частина, j – уявна одиниця. Для їх кодування використовуються 32 розряди, причому 16 старших розрядів для кодування дійсної частини, 16 молодших для кодування уявної частини. На рис. 10.4 наведений формат вхідних та вихідних даних.

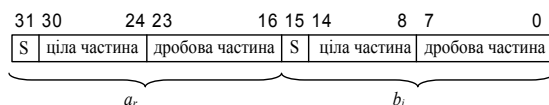
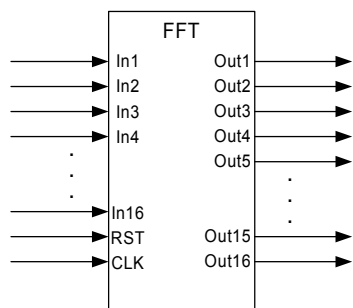


Рис. 10.4. Формат вхідних та вихідних даних.

Дійсна і уявна частина є знаковими числами з фіксованою крапкою. Розряди 31 та 15 використовуються для кодування знаків, 30-24 та 14-8 для кодування цілих частин, 23-16 та 7-0 для кодування дробових частин дійсної та уявної частини даних.

Інтерфейс пристрою.

Пристрій має 16 паралельних вхідних портів для прийому даних та два



вхідні порти для керуючих сигналів – сигнали RST (Reset – початковий скид) та CLK (Clock – синхронізація). Результати обробки вхідних даних видаються по 16-х паралельних вихідних портах. Всі порти, крім RST і CLK є 32-х розрядні. Група сигналів наведена на рис. 10.5.

Рис.10.5. Група сигналів.

Розробка структурної схеми процесора.

Дані надходять по 16-ти паралельних портах і, проходячи через мультиплексори, записуються у відповідні регістри. Мультиплексорами управляє керуючий автомат (Control Unit) по лінії SEL. При $SEL=0$, в регістри записуються

вхідні дані, при $SEL=1$ – проміжні дані. Дані з регістрів надходять на входи двохточкових ШПФ та на наступному циклі знову записуються до відповідних регістрів. На останньому циклі, керуючий автомат генерує сигнал *Open*, по якому дані записуються в регістри та поступають на вихідні порти у біт-інверсному порядку.

В ПЗП (ROM) зберігаються повертаючі множники, що є константами, і подаються на двохточкові ШПФ через комутуючу мережу. Роботою комутуючої мережі (Communication Network) управляє керуючий пристрій по лініях DC [3:0].

Далі детальніше розглянемо кожний з основних вузлів, на базі яких побудована структура процесора.

Двохточкові ШПФ (fft0 – fft7).

Схема вузла для обчислення двохточкового ШПФ наведена на рис. 10.6. На входи пристрою поступають два 32-х розрядних числа A і B та два 16-ти розрядних числа W_r і W_i . На виході отримуємо два 32-х розрядних числа $X=A+B$ і $Y=(A-B)W_N$.

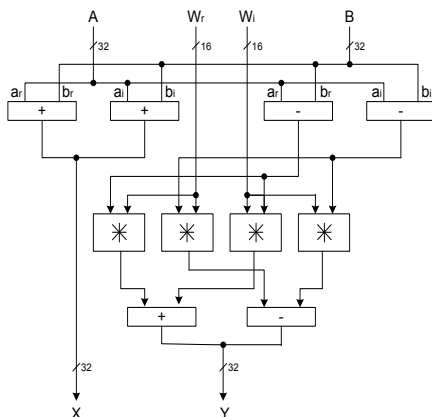


Рис. 10.6. Внутрішня структура двохточкового ШПФ.

Комутаційна мережа (Communication Network).

Комутаційна мережа використовується для комутації двохточкового ШПФ з потрібним номером ПЗП у відповідні моменти часу (на відповідних робочих

циклах пристрою). Вона складається із набору з'єднаних буферів, якими управляє керуючий автомат по лініях $DC[3:0]$. Встановлюючи потрібні значення сигналів на лініях керування він дозволяє або забороняє видачу даних на виході кожного з буферів. В момент часу на операційний пристрій можуть поступати дані тільки з одного буфера. На рис. 10.7 зображено підключення одного з операційних пристроїв до ПЗП.

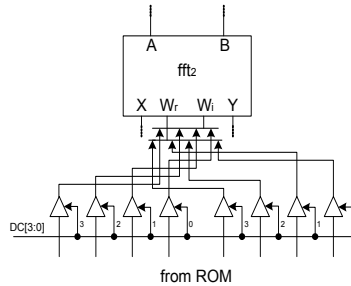


Рис. 10.7. Підключення двохточкового ШПФ до ПЗП через комутаційну мережу.

Вибір кількості і об'єму ПЗП.

Для вибору кількості і об'єму ПЗП необхідно провести аналіз кількості і значень повертаючих множників.

В алгоритмі 16-ти точкового ШПФ за основою 2 необхідно зберігати в пам'яті 8 значень повертаючих множників W_N . $W_N = w_r + jw_i$ є комплексними числами (косинусна складова – w_r ; синусна складова – w_i). На рис. 10.8 показані значення функцій $\cos(t)$ та $\sin(t)$, а на рис. 10.9 – граф керуючого автомата.

З рис. 10.8 видно, що деякі з коефіцієнтів співпадають за значенням між собою, а саме: $Wr0 = Wi4$, $Wr1 = Wi3 = Wi5$, $Wr2 = Wi2 = Wi6$, $Wr3 = Wi1 = Wi7$, $Wr4 = Wi0$. Залишаються $Wr5$, $Wr6$, $Wr7$. Провівши такий аналіз, приходимо до висновку – необхідно вісім 16-ти розрядних ПЗП.

Керуючий автомат (Control Unit).

Керуючий автомат здійснює управління роботою процесора. Отримуючи сигнали RST і CLK він формує всі необхідні сигнали управління, а саме:

- SEL – для управління роботою мультиплексорів;

- DC[3:0] – чотирихрозрядний сигнал для управління комутаційною мережею;

- OEN – сигнал дозволу видачі вихідних даних.

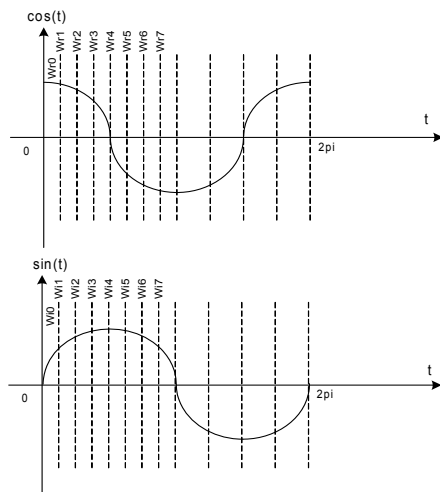


Рис. 10.8. Значення функцій \cos і \sin на періоді 2π .

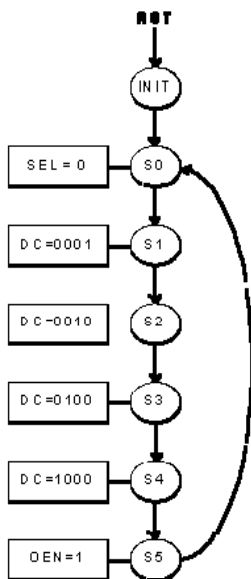


Рис. 10.9. Граф керуючого автомата.

При надходженні сигналу $RST = 1$ керуючий автомат переходить в початковий стан INIT. При встановленні RST в нуль автомат переходить з початкового стану в стан S0, далі в S1, і так далі. З стану S5 автомат повертається в стан S0. Перехід із стану в стан здійснюється по зростаючому фронту сигналу CLK. На відповідних станах керуючий автомат формує необхідні керуючі сигнали.

На рис. 10.10 наведена симуляційна часова діаграма функціонування керуючого автомата.

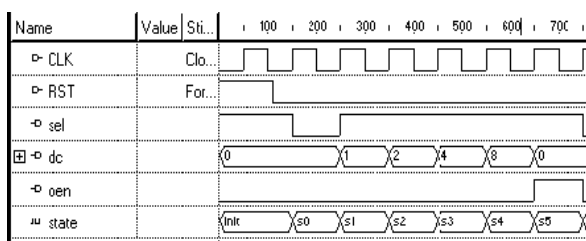


Рис. 10.10. Симуляційна часова діаграма функціонування керуючого автомата.

Тестування пристрою.

Для тестування пристрою формується тестова послідовність такого типу:

Модельюються сигнали $Acos(kt)$ і $Asin(kt)$. Далі беруться значення цих сигналів у відповідних точках: ділянка на якій розглядається сигнал розбивається на 16 відліків і в кожній точці отримуються необхідні для проведення тестів значення. Значення отримані в результаті обчислення $Acos(kt)$ є дійсними складовими вхідних даних, а $Asin(kt)$ – уявними.

Для побудови даної тестової послідовності необхідно розробити програму для обчислення виразів $sin(kt)$ і $cos(kt)$ при $1 \leq k \leq 16$. Програма складається з процедури переводу числа з десяткової системи числення в двійкову та тіла програми де в циклі обчислюються значення $sin(kt)$ і $cos(kt)$, які передаються як параметри процедури для переведення в двійкову систему.

За результатами отриманими в результаті виконання даної програми буде створено набір тестових послідовностей та формується script – файл для тестування пристрою засобами пакету Active HDL.

Структурна схема процесора наведена на рис. 10.11.

10.4. Завдання для самоперевірки

1. При виконанні N-точкового ШПФ на ПЛІС кількість етапів опрацювання даних така:

А) 10; Б) 11; В) 12; Г) 13 ; Д) немає правильної відповіді.

2. Продуктивність процесора ШПФ на ПЛІС обчислюється в:

А) MIPS; Б) MFLOPS; В) MCPS; Г) MAC; Д) немає правильної відповіді.

3. При виконанні алгоритму ШПФ на ПЛІС в схемі вузла “метелик” буферне ОЗП використовується для:

А) зберігання результатів проміжних обчислень і вивантаження результатів; Б) зберігання результатів проміжних обчислень; В) вводу вхідних даних; Г) виводу даних; Д) немає правильної відповіді

4. Комутаційна мережа в процесорі ШПФ на ПЛІС використовується для:

А) комутації двохточкових ШПФ і ПЗП коефіцієнтів у відповідні моменти часу; Б) постійної комутації двохточкових ШПФ і ПЗП коефіцієнтів; В) комутації виходів ПЗП і виходів керуючого автомата; Г) комутації ПЗП коефіцієнтів і ПЗП тестів; Д) немає правильної відповіді.

5. Виділений об’єм пам’яті для повертаючих множників процесора ШПФ за основою 4 на ПЛІС 384 комірки, в котрі записуються дійсна чи уявна складові. Розмірність ШПФ $N=256$. Даний об’єм пам’яті є:

А) оптимальним; Б) не вистачає 64 комірки; В) надлишок – 32 комірки; Г) не вистачає 16 комірок Д) немає правильної відповіді.

Правильні відповіді до завдань:

1) В; 2) Г; 3) Б; 4) А; 5) А

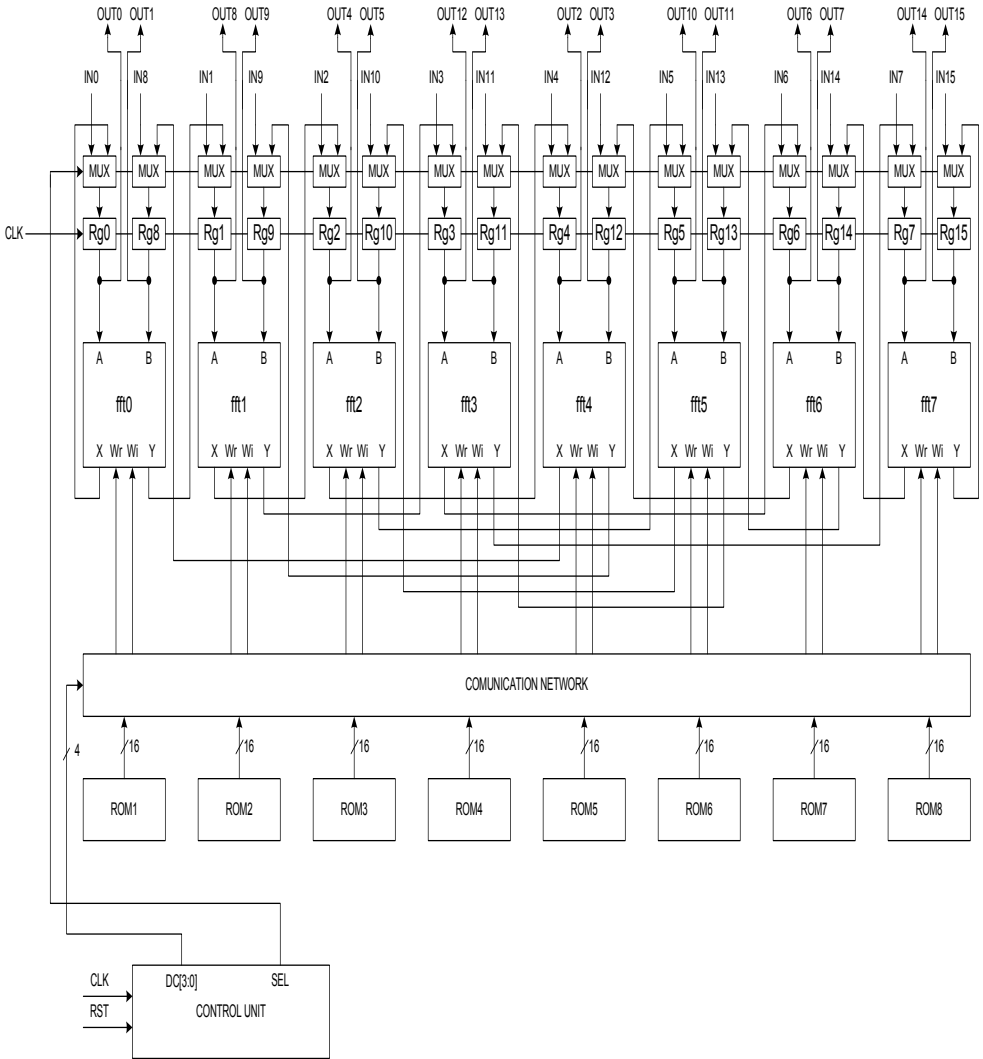


Рис. 10.11. Структурна схема процесора обчислення 16-ти точкового ШПФ за основою 2.

РОЗДІЛ 11. РЕАЛІЗАЦІЯ АЛГОРИТМІВ ОПРАЦЮВАННЯ СИГНАЛІВ НА НЕЙРОПРОЦЕСОРАХ

11.1. Основні характеристики нейрокомп'ютерів

Нейрокомп'ютер – це обчислювальна система з MSIMD архітектурою, тобто з паралельними потоками однакових команд і множинним потоком даних. На сьогодні можна виділити три основних напрямки розвитку обчислювальних систем з масовим паралелізмом (ОСМП) (див.табл. 11.1).

Таблиця 11.1 Напрямки розвитку обчислювальних систем з масовим паралелізмом

№	Найменування напрямку	Опис
1	ОСМП на базі каскадного з'єднання універсальних SISD, SIMD, MISD мікропроцесорів.	Елементна база – універсальні RISC чи CISC процесори: Intel, AMD, Sparc, Alpha, Power PC, MIPS і т.п.
2	На базі процесорів з розпаралелюванням на апаратному рівні.	Елементна база – ЦСП: TMS, ADSP, Motorola, ПЛІС.
3	ОСМП на спеціалізованій елементній базі	Елементна база: від спеціалізованих однобітових процесорів до нейрочипів.

Нейромережеві системи, реалізовані на апаратних платформах першого напрямку (нехай і мультипроцесорних) відносяться до нейроемулаторів – тобто систем, що реалізують типові нейрооперації (зважене підсумовування і нелінійне перетворення) на програмному рівні. Нейромережеві системи, реалізовані на апаратних платформах другого і третього напрямку у вигляді плат розширення стандартних обчислювальних систем (1-го напрямку), називаються нейроприскорювачами, і системи, реалізовані на апаратній платформі третього напрямку у вигляді функціонально закінчених обчислювальних пристроїв, відносяться до нейрокомп'ютерів (всі операції виконуються в нейромережевому логічному базисі). Нейроприскорювачі діляться на два класи: “віртуальні”

(вставляються в слот розширення стандартного PC) і “зовнішні” (з’єднуються з керуючим Host комп’ютером по конкретному інтерфейсу). Нейрозасоби проектуєть на базі ПЛІС, каскадного з’єднання ЦСП типу ADSP2106х, TMS320C4х,8х, DSP96002 і ін., спеціалізованих процесорів, наприклад, Л1879ВМ1 (NM6403) [10, 33].

Процесор Л1879ВМ1 – високопродуктивний спеціалізований мікропроцесор, що об’єднує в собі риси двох сучасних архітектур: VLIW (Very Low Instruction Word) і SIMD (Single Instruction Multiple Data) (рис. 11.1).

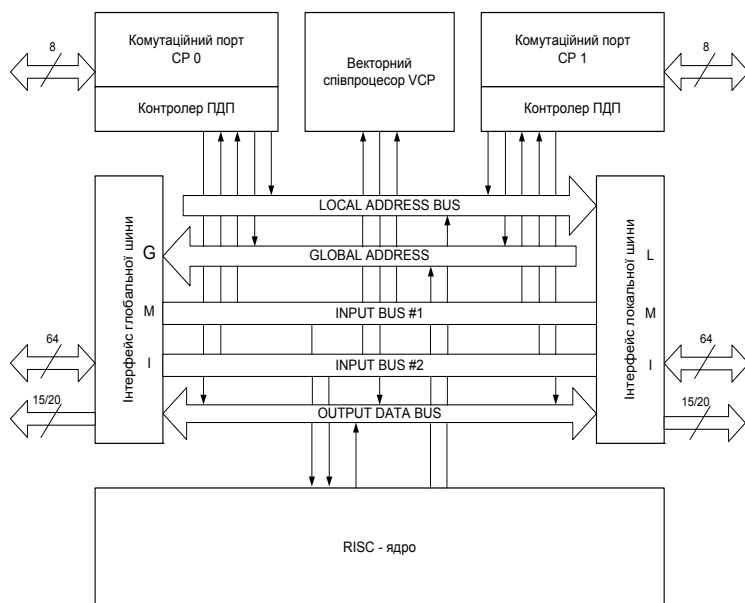


Рис. 11.1. Архітектура процесора NM6403.

Основні обчислювальні вузли процесора – керуюче RISC-ядро і векторний співпроцесор. RISC-ядро – центральний процесорний вузол, що виконує всі основні функції по керуванню роботою кристала. Крім того, RISC-процесор виконує арифметико-логічні і операції зсуву над 32-розрядними скалярними даними і формує 32-розрядні адреси команд і даних при звертаннях до зовнішньої

пам'яті. Довжина команди – 32 і 64 розряди (зазвичай в команді виконуються дві операції). Процесор реалізує п'ятиступінчатий 32-розрядний конвеєр. Адресний простір – 16 Гбайт, два адресних генератори, вісім регістрів загального призначення і вісім адресних регістрів. Будь-яка інструкція виконується за один такт. Векторний співпроцесор призначений для арифметичних і логічних операцій над 64-розрядними скалярними даними програмувальної розрядності. Обмін даними між основними вузлами процесора відбувається по трьох внутрішніх шинах, двох вхідним і одній вихідній.

GMI і LMI – два однакових блоки програмованого інтерфейсу з локальною і глобальною 64-розрядними зовнішніми шинами. До кожної з них може бути під'єднана зовнішня пам'ять, що містить до 2^{31} 32-розрядних комірок. Обмін даними з зовнішньою пам'яттю здійснюється як 32-, так і 64-розрядними словами (NM6403 одночасно вибирає дві сусідні комірки пам'яті). Кожен блок програмувального інтерфейсу дозволяє працювати з двома банками зовнішньої пам'яті різного обсягу, типу (DRAM, SRAM, Flash ROM, EDO DRAM і т.д.) і швидкодії без додаткового устаткування. Передбачено апаратну підтримку режиму розподіленої пам'яті для різних мультипроцесорних конфігурацій зовнішніх шин.

CP1 і CP2 – ідентичні комунікаційні порти, що забезпечують інформаційний обмін по двонаправленій восьмиразрядній шині, призначені для побудови високопродуктивних мультипроцесорних систем і сумісні з комунікаційними портами процесора TM320C4x. Кожен комунікаційний порт має вбудований контролер прямого доступу до пам'яті (ПДП, DMA), що дозволяє обмінюватися 64-розрядними даними з зовнішньою пам'яттю.

Векторний співпроцесор – основний функціональний елемент L1879BM1. Структурно він являє собою матрично-векторний операційний пристрій і набір регістрів різного призначення.

Операційний пристрій (ОП) – регулярна матрична структура розміром 64x64 комірки (рис. 11.2). Матриця може бути довільно розділена на стовпці і

рядки. В макрокомірці завантажуються вагові коефіцієнти W_{ij} . На вхід матриці подається вектор вхідних даних $\vec{X} = (X_1, X_2, \dots, X_n)$, кожному елементу якого відповідає рядок матриці. Ширина рядка (у бітах) – розрядність даного елемента вхідних даних. У макрокомірках відбувається множення елемента вектора вхідних даних на ваговий коефіцієнт і додавання зі значенням верхньої комірки (або значень входів U). Таким чином, для кожного стовпця обчислюється скалярний вираз $y_i = U_i + \sum_j W_{ij} X_j$. Для зменшення розрядності вихідних даних і захисту від арифметичного переповнення використовується програмована функція насичення (рис. 11.2).

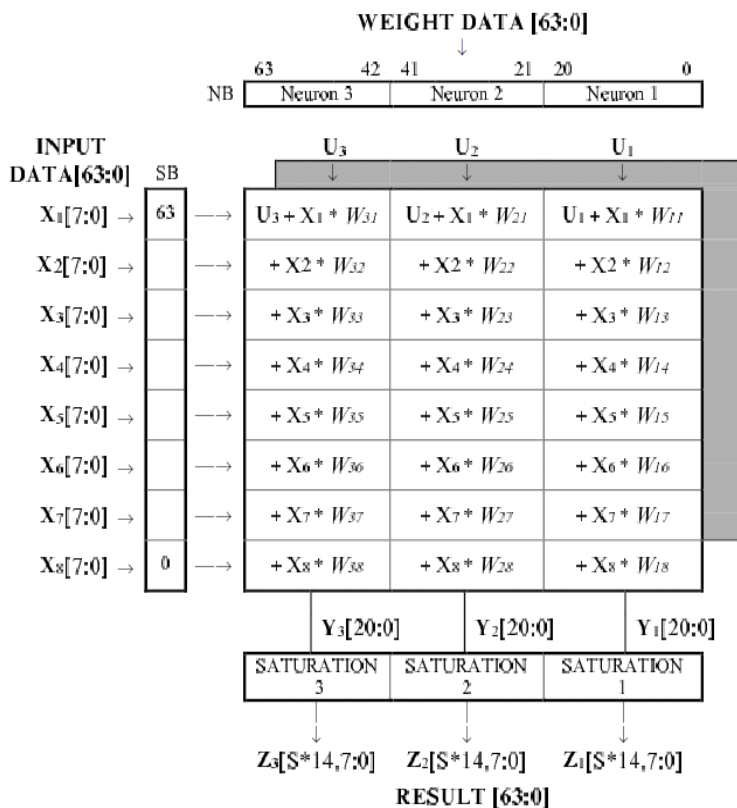


Рис. 11.2. Програмована функція насичення.

Операнди і вихідні значення упаковуються в 64-х розрядне слово. Всі операції в матриці ОП виконує паралельно, за один такт. Завантаження вагових коефіцієнтів відбувається за 32 такти. У векторному співпроцесорі є “тіньова” матриця, в яку вагові коефіцієнти можна завантажувати у фоновому режимі. Перемикання “тіньової” і робочої матриць здійснюється за один такт. Найважливіша особливість векторного співпроцесора – робота з операндами довільної довжини у діапазоні 1-64 біт. Цим досягається оптимальне співвідношення між швидкістю і точністю обчислень: при однобітових операндах на тактовій частоті 40 МГц продуктивність складає 11 520 ММАС (мільйонів операцій множення з сумуванням) чи 40 000 MOPS (мільйонів логічних операцій/секунду), при 32-бітових операндах і 64-бітовому результаті вона стане номінальною – 40 ММАС. Динамічна, в процесі обчислень, зміна розрядності операндів дозволяє підвищити продуктивність.

11.2. Визначення продуктивності нейрокомп'ютерів

Для обчислення продуктивності використовуються такий вираз:

$$MCPS = \left\lfloor \frac{64}{N_X} \right\rfloor * \left\lfloor \frac{64}{N_X + N_W + \left\lceil \log_2 \frac{64}{N_X} \right\rceil} \right\rfloor * F ,$$

де MCPS – мільйон з'єднань у секунду (еквівалентно ММАС), 64 – розрядність слова даних, N_X – розрядність синапсів, N_W – розрядність ваг, F – тактова частота ($F = 50$ МГц).

У випадку $N_X \neq 1$ і $N_W = 1$ маємо рівність:

$$MCPS = \left\lfloor \frac{64}{N_X} \right\rfloor * \left\lfloor \frac{64}{N_X + \left\lceil \log_2 \frac{64}{N_X} \right\rceil} \right\rfloor * F .$$

У випадку $N_X = 1$ і $N_W \neq 1$ виконується співвідношення:

$$MCPS = 32 * \left\lfloor \frac{64}{N_W + 5} \right\rfloor * F .$$

Якщо $N_X = N_W = 1$, то отримуємо вираз:

$$MCPS = 1024 * F$$

Якщо ж $N_X = N_W = 32$, то маємо співпадіння:

$$MCPS = F$$

11.3. Організація паралельних обчислень алгоритмів ШПФ на нейрокомп'ютері NM6403

При виконанні алгоритму ШПФ кількість одночасно виконуваних множень головним чином визначає продуктивність процесора. Тому виникає питання про розпаралелювання обчислень і реалізацію алгоритмів ШПФ із більш високими основами і їхніми можливими комбінаціями. В залежності від вибору основи змінюється кількість арифметичних операцій і кількість шарів графа. В табл. 11.2 наведені показники обчислювальної складності ШПФ для різних розмірностей і основ, де ККМ, ККД, КШГ, КО, відповідно, кількість комплексних множень, кількість комплексних додавань, кількість шарів в графі, комбінація основ.

Таблиця 11.2. Обчислювальна складність ШПФ

	Пряме обчислення ШПФ (основа N)			Обчислення ШПФ за основою 2			Обчислення ШПФ з комбінованими основами 2, 16, 32			
	ККМ	ККД	КШГ	ККМ	ККД	КШГ	ККМ	ККД	КО	КШГ
N	N^2	$N^2 - N$		$(N/2) * \log_2 N$	$N \log_2 N$					
256	65536	65280	1	1024	2048	8	8192	7680	16-16	2
512	262144	261632	1	2304	4608	9	16384	16384	2-16-16	3
1024	1048576	1047552	1	5120	10240	10	49152	49152	32-32	2
2048	4194304	4194304	1	11264	22528	11	131072	131072	2-32-32	3

В алгоритмах ШПФ за основою 2 кількість таких шарів максимальна, тому при поетапному надходженні результатів обчислень від шару до шару відбувається більше нагромадження помилок заокруглення, ніж в алгоритмах з

більш високими основами. Це особливо критично у випадках, коли обчислення проводяться в цілочисельній арифметиці (з фіксованою крапкою). У цьому випадку для запобігання переповнення проміжні результати після кожного чи після групи етапів множення (шарів графа) необхідно додатково нормалізувати, застосовуючи операцію зсуву вправо. Нормалізація крім зсуву може містити в собі процедуру округлення, що також вносить додаткові обчислювальні витрати. Можливим компромісним рішенням може виступати підхід, оснований на збільшенні основи в алгоритмах ШПФ. Для прикладу розглянемо варіант виконання ШПФ-256 за основою 16, який забезпечує можливість організації паралельних обчислень і зменшення кількості шарів графа до двох.

Дискретне 256-точкове перетворення Фур'є визначається формулою:

$$Y(k) = \sum_{n=0}^{255} W_{256}^{kn} \cdot X(n), k = 0..255, \quad \text{де} \quad W_{256}^k = \exp\left(-\frac{2\pi \cdot i \cdot k}{256}\right)$$

Дана формула після тотожних перетворень приймає вид, що є опорним для побудови ШПФ-256 за основою 16:

$$Y(k) = \sum_{n=0}^{15} W_{256}^{kn} \cdot \sum_{i=0}^{15} W_{256}^{16ki} \cdot X(16 \cdot i + n); k = 0..255$$

Структура графу обчислення даного ШПФ-256 наведена на рис. 11.3.

Кожен блок графа має 16 комплексних входів і виходів. Як показано на рис. 11.3(б) кожен блок графа є 16-точковим ДПФ і відрізняється від інших блоків тільки комплексними коефіцієнтами W . Таким чином, розпаралелювання алгоритму ШПФ фактично зводиться до реалізації ефективного обчислення ДПФ-16, тобто до знаходження 16 скалярних добутків різних векторів $[W]$ з одним вектором $[x]$, що еквівалентно множенню матриці коефіцієнтів перетворення Фур'є - $[W]$ розмірністю 16×16 на вхідний вектор $[x]$.

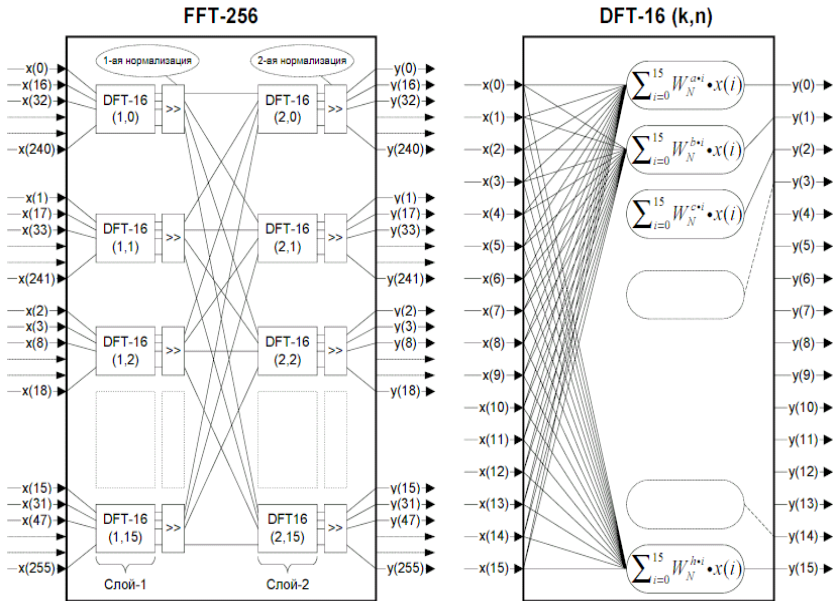


Рис. 11.3 Узагальнений граф ШПФ-256 (а) Розгорнута схема блоку ДПФ за основою 16 (б).

Уявні і дійсні частини коефіцієнтів W зберігаються в упакованому виді, але в різних 64-ти розрядних словах. Усі коефіцієнти W обчислюються заздалегідь і тому зберігаються усередині масиву в порядку зручному для наступних обчислень, що показано на рис. 11.4.

63					32 31			0	
Imag					Real				
63	56					15	8	7	0
W(7).re	W(6).re	W(5).re	W(4).re	W(3).re	W(2).re	W(1).re	W(0).re		
W(7).im	W(6).im	W(5).im	W(4).im	W(3).im	W(2).im	W(1).im	W(0).im		

Рис. 11.4 Формат збереження вхідних даних і коефіцієнтів перетворення.

Векторний помножувач працює в двох конфігураціях згідно з схемами, які наведені на рис. 11.5 та рис. 11.6.

Перша схема виконує 16 множень з накопиченням за такт і служить для знаходження сум попарних добутків уявних і дійсних частин, друга виконує 4 множення з накопиченням за такт, але фактично служить тільки для остаточного додавання отриманих часткових сум.

Весь процес обчислення скалярного добутку $y(k) = \sum_{i=0}^{15} W_{256}^{ai} X(i)$ складається з

трьох етапів:

1. В матрицю вагових коефіцієнтів завантажуються 8 комплексних чисел $x(0)..x(7)$. На вхід помножувача X по черзі подається спочатку вектор з 8-ми дійсних частин комплексних коефіцієнтів $W(0)..W(7)$ (тут $W(i) = \sum_{a=0}^{15} W_{256}^{ai}$), а потім вектор з 8-ми уявних частин. Множення виконуються у відповідності з схемою на рис. 11.5.

2. Далі з виходу помножувача результат добутку у виді двох 64-ти розрядних слів безпосередньо поступає на підсумовуючий Y -вхід помножувача. При цьому в матрицю вагових коефіцієнтів завантажуються числа $x(8)..x(15)$, а на вхід X помножувача аналогічно поступають і збільшуються нові коефіцієнти $W(8)..W(15)$.

3. Для отримання кінцевого результату $y(k)$, суми в лівих і правих частинах двох останніх результатів ("3-rd product" і "4-th product") необхідно перехресно додати (з врахуванням знака "–"). Для цього в матрицю вагових коефіцієнтів завантажуються числа 0,1 і -1, а самі суми подаються на вхід X і Y , і векторний помножувача видає кінцевий результат для $y(k)$.

В результаті, з врахуванням часу передачі даних, кожен крок множення практично займає один процесорний такт, що досягається за рахунок одночасного використання двох шин даних – підкачування вхідних даних $x(i)$ по одній шині суміщається з завантаженням коефіцієнтів $W(i)$ чи вивантаженням результатів множення $y(i)$ по іншій. Таким чином, реально вся процедура скалярного множення двох комплексних 16-ти вимірних векторів у середньому по всьому ШПФ-256 складає 7 процесорних тактів.

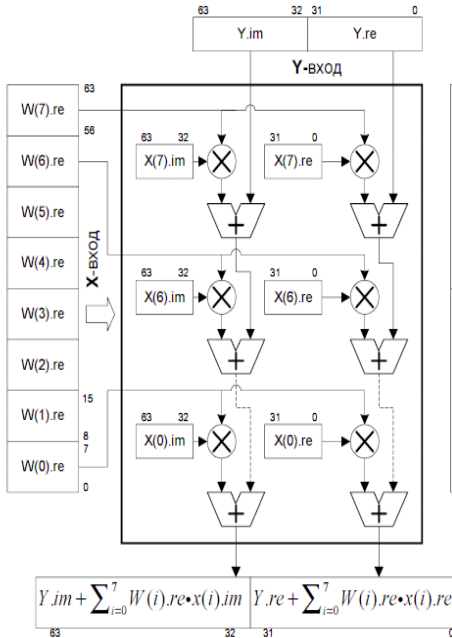


Рис. 11.5. Еквівалентна схема помножувача векторного співпроцесора NM6403 при розбитті матриці вагових коефіцієнтів – (2x32біти)/(8x8біт)

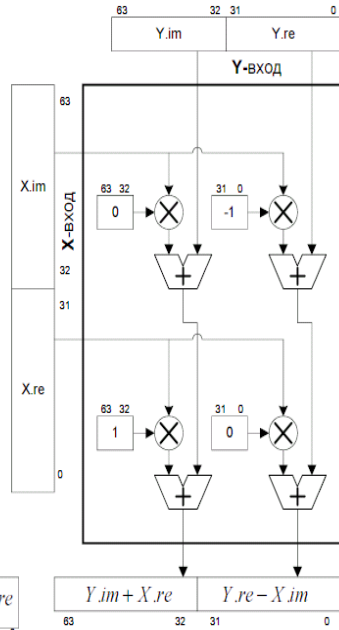


Рис. 11.6. Еквівалентна схема помножувача векторного співпроцесора NM6403 при розбитті матриці вагових коефіцієнтів – (2x32біти)/(2x32біти)

Схема послідовності обчислення скалярного добутку $y(k) = \sum_{i=0}^{15} W_{256}^{ai} X(i)$ наведена на рис. 11.7.

Продуктивність і точність обчислень.

Точність обчислень визначається кількістю біт, що відводяться для представлення коефіцієнтів W. Є два способи представлення значень косинусів і синусів у 8 розрядній сітці:

1. $W = \text{round}(64.0 * \cos(x))$ – (умовно 6 біт на одиницю).
2. $W = \text{round}(127.0 * \cos(x))$ – (умовно 7 біт на одиницю).

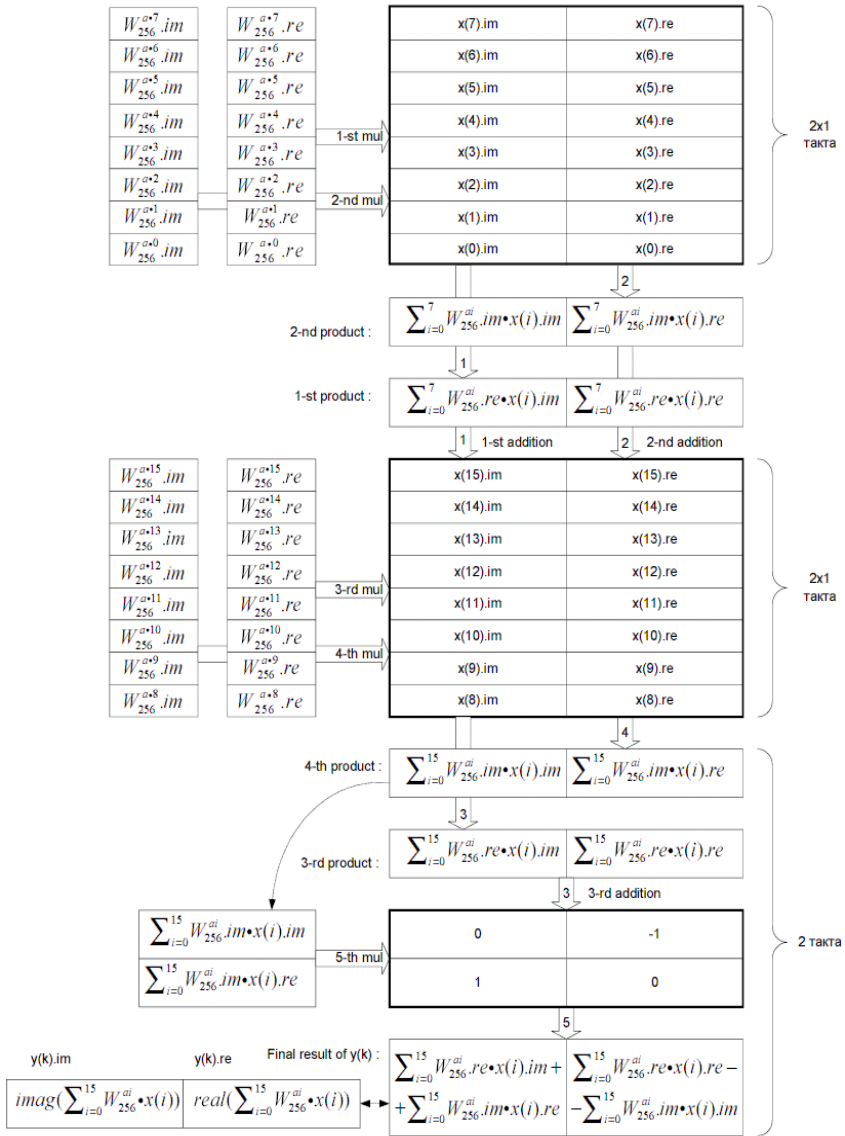


Рис. 11.7. Схема послідовності обчислення скалярного добутку $y(k) = \sum_{i=0}^{15} W_{256}^{ai} X(i)$.

Перший спосіб має 65 градацій косинуса в діапазоні 0..1, а другий – 128 градацій, і, отже, більшу точність. Однак, виконуючи в ході обчислень лише

операції цілочисельного множення і додавання, наприкінці необхідно провести нормалізацію результатів, тобто кожен елемент вихідного масиву потрібно поділити на відповідний масштабуючий коефіцієнт. Для першого випадку він рівний 642, а для другого – 1272. В обох випадках розподіл доцільно замінити зсувом вправо, але, на відміну від першого способу, заміна розподілу на 1272 зсувом на 14 біт вправо привносить невелику систематичну похибку. Нормалізацію можна проводити або один раз наприкінці множень, або двічі – після кожного етапу множення (проміжна нормалізація служить для запобігання переповнення в ході обчислень). Кількість необхідних етапів нормалізації визначається в залежності від діапазону вхідних даних згідно з табл. 11.3.

Для оцінки точності довільний сигнал оброблявся прямим і оберненим перетворенням Фур'є, після чого вихідний x порівнювався з відновленим сигналом x' . Зокрема, знаходилося математичне сподівання M і середнє квадратичне відхилення σ відносних похибок δ : $\delta(i) = \frac{x'(i).re - x(i).re}{x(i).re}, i = 0..255$.

Таблиця 11.3. Порівняльна характеристика точності відновленого сигналу після прямого і зворотного ШПФ з різними основами

Перетворення Фур'є	Систематична похибка – M		СКО – σ	
	6 біт/1.0	7 біт/1.0	6 біт/1.0	7 біт/1.0
FFT-256 за основою 2	-1%	-12%	2.0%	0.9%
FFT-256 за основою 16	-0.4%	-3%	1.2%	0.6%

Знак “-” означає, що відновлений сигнал зменшується на M відсотків порівняно з вихідним. Згідно з табл. 11.3, 7-бітне зображення коефіцієнтів для ШПФ за основою 2 вносить значну систематичну похибку і робить такий алгоритм менш ефективним. Алгоритми ШПФ з основами 16 і 32 дозволяють ефективніше використовувати наявну розрядну сітку під коефіцієнти W і мають вищу точність за рахунок у 4-5 разів меншої кількості шарів графа, що також зменшує додаткові витрати на нормалізацію і округлення. При цьому операція округлення на процесорі NM6403 виконується за допомогою векторного регістра vr разом з основними обчисленнями, не приводячи до додаткових витрат часу.

Можлива додаткова оптимізація через поєднання нормалізації (арифметичного зсуву вправо) з останнім етапом обчислення скалярного добутку (рис. 11.7).

В табл. 11.4 наведені значення продуктивності функцій прямого і оберненого ШПФ на процесорі NM6403 з тактом виконання 25 нс.

Загальна характеристика функцій ШПФ. Вхідні і вихідні дані – цілі 32-ох розрядні комплексні числа, формат збереження наведений на рис. 11.4. Діапазон вхідних даних зазначений у табл. 11.3. Розрядність коефіцієнтів перетворення – 8 біт (два варіанти представлення: 6 і 7 біт). Робота з даними – арифметика з фіксованою крапкою. Вихідні дані розташовані в правильному порядку.

Таблиця 11.4 Продуктивність функцій прямого і зворотного ШПФ на процесорі NM6403

Кільк. комплекс. відліків	Без нормалізації			З однією нормалізацією			З двома нормалізаціями		
	Тактів	Час, мс	Діапазон вх.даних	Тактів	Час, мс	Діапазон вх.даних	Тактів	Час, мс	Діапазон вх.даних
256	36621	0.092	±512 (7bit) ±2047(6bit))	3994	0.1	±512 (7bit) ±2047(6bit))	4300	0.11	±2 ¹⁸ (7bit) ±2 ¹⁸ (6bit)
512	8180	0.2	±256 (7bit) ±1023(6bit))	8766	0.22	±256 (7bit) ±1023(6bit))	9350	0.24	±2 ¹⁸ (7bit) ±2 ¹⁸ (6bit)
1024	18900	0.47	±128 (7bit) ±511 (6bit)	20041	0.5	±128 (7bit) ±511 (6bit)	21200	0.53	±2 ¹⁷ (7bit) ±2 ¹⁷ (6bit)
2048	47624	1.2	±64 (7bit) ±255 (6bit)	49800	1.25	±64 (7bit) ±255 (6bit)	52000	1.3	±2 ¹⁷ (7bit) ±2 ¹⁷ (6bit)

11.4. Завдання для самоперевірки

1. Нейропроцесори, що використовуються для обробки сигналів, реалізовані на:

А) ЦСП-процесорах; Б) ПЛІС; В) нейрочіпах; Г) універсальних RISC-процесорах ; Д) немає правильної відповіді.

2. Нейропроцесор об'єднує риси таких архітектур:

А) MIMD, SIMD; Б) SIMD, VLIW; В) NUMA, VLIW; Г) SIMD, PVP; Д) немає правильної відповіді.

3. Операційний пристрій нейропроцесора є:

А) матричною структурою; Б) конвейерним регістром; В) універсальним обчислювальним вузлом; Г) вузлом керування; Д) немає правильної відповіді

4. Продуктивність нейрокомп'ютерів обчислюється в таких одиницях/сек:

А) кількістю операцій.; Б) кількістю множень з сумуванням; В) кількістю з'єднань; Г) кількістю операцій з рухомою крапкою; Д) немає правильної відповіді.

5. Шляхами запобігання переповнення проміжних результатів обчислення алгоритму ШПФ на нейропроцесорі є:

А) зменшення основи ШПФ; Б) збільшення розрядності вхідних даних; В) збільшення основи ШПФ; Г) зсув вліво результату Д) немає правильної відповіді.

Правильні відповіді до завдань:

1) В; 2) Б; 3) А; 4) В; 5) В.

РОЗДІЛ 12. ДІАГНОСТИКА І КОНТРОЛЬ ПРОЦЕСОРІВ І СИСТЕМ ОПРАЦЮВАННЯ СИГНАЛІВ

12.1. Особливості діагностики та контролю процесорів та систем опрацювання сигналів

Для контролю і діагностики вузлів опрацювання сигналів застосовуються різні сполучення відомих методів вбудованого і зовнішнього контролю комп'ютерів або методи діагностики складних систем, що базуються на використанні тестових наборів різного типу. Проте, робота в РРЧ, застосування високочастотних НВІС, динамічний характер відмов зробили практично нереальним визначення моменту і місця виникнення несправностей у вузлах такого класу, при використанні цих методів.

Розрізняють функціональне діагностування з метою контролю і з метою пошуку несправностей. Загальну структурну схему системи діагностування апаратними засобами з метою контролю функціонування систем опрацювання сигналів (СОС) показано на рис. 12.1.



Рис.12.1. Структурна схема апаратного контролю

На згаданому рисунку прийнято такі позначення:

- $P_{очн}(t)$ – ймовірність безвідмовної роботи основної схеми (СОС);
- $P_k(t)$ – ймовірність безвідмовної роботи схеми контролю;
- $P_{сп}(t)$ – ймовірність безвідмовної роботи схеми порівняння (схеми прийняття рішення);

- $X = \{x_1 \ x_2 \ x_n\}$ – вектор входу СОС;

- $Y = \{y_1 \ y_2 \ y_n\}$ – вектор виходу СОС.

При організації такого контролю коефіцієнт надлишкового обладнання дорівнює

$$K_n = \frac{L_o + L_n + L_k + L_{cn}}{L_o + L_n}, \quad (12.1)$$

де - L_o – об'єм обладнання СОС, яка охоплена апаратним контролем;

- L_n – об'єм неконтрольованої частини СОС;

- $L = L_o + L_n$ – об'єм обладнання СОС;

- L_k – об'єм обладнання схеми контролю;

- L_{cn} – об'єм обладнання схеми порівняння.

Враховуючи, що $L_{cn} \ll L$ і $L_{cn} \ll L_k$, значенням L_{cn} можна знехтувати.

При оцінці затрат враховуються два випадки: схема контролю контролюється повністю; схема контролю не контролюється.

Тестове діагностування передбачає подачу на контролюючий пристрій спеціальних збурень (тестів), які дозволяють за значеннями вихідних сигналів пристрою, шляхом порівняння отриманих результатів з наперед відомими еталонними, виявити заданий клас несправностей. Достовірність функціонування перевіряється за певний проміжок часу

$$t = t_p + t_r, \quad (12.2),$$

де - t_p – тривалість основної роботи;

- t_r – тривалість проходження тесту.

Структурну схему тестового діагностування показано на рис. 12.2.

Відомі методи діагностування та контролю (з використанням принципів резервування, дублювання, мажоритування, коректуючих кодів, сигнатурного аналізу) не забезпечують стовідсоткову повноту перевірки схем вбудованого контролю, оскільки кінцеві вихідні каскади цих схем не контролюються.



Рис.12.2. Структурна схема тестового діагностування

Тому розроблений клас самодіагностуючих (самоперевіряючих) схем в яких довільна несправність з заданого класу при подачі вхідних робочих наборів приводить до появи значень вихідних функцій, які відрізняються від значень вихідних функцій, що видаються справною контролюючою схемою при справній основній схемі [7]. Одним з оптимальніших методів побудови самодіагностуючих схем вбудованого контролю є метод, схему якого показано на рис. 12.3.

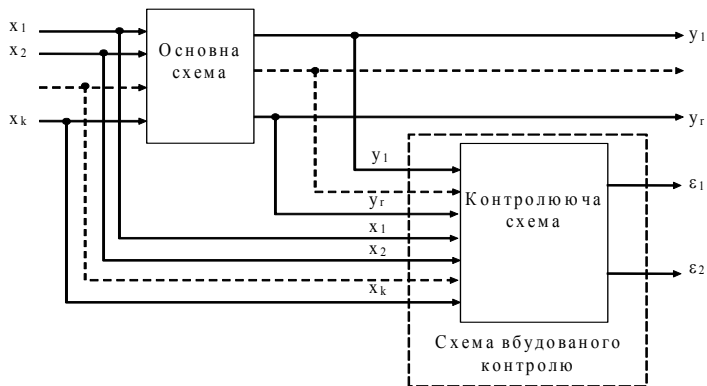


Рис.10.3. Структурна схема самодіагностуючого пристрою

Значення x_1, x_2, \dots, x_k ; y_1, y_2, \dots, y_r є, відповідно, вхідними та вихідними значеннями схеми. При справній роботі основної схеми на її виходах буде N різних вихідних сигналів (для комбінаційних схем $N \leq 2^r$, для схем з пам'яттю $N \leq 2^q \cdot 2^r$, де q – кількість елементів пам'яті). Проте, робочими вхідними наборами

схеми вбудованого контролю будуть тільки $2^k + N$ різних значень з 2^{r+q} можливих (якщо основна схема є комбінаційною) і 2^{r+q+k} (якщо основна схема з пам'яттю). Робочі вхідні значення є дозволеними вхідними значеннями для схеми вбудованого контролю, решту є забороненими і характеризують наявність несправностей певного класу в основній схемі.

Аналіз самоперевіряючих схем показує, що для повної перевірки необхідні великі додаткові апаратні затрати.

Є три основні принципи побудови діагностичних систем, які визначають організацію процесу діагностування: принцип централізованого діагностування, принцип розкручування, принцип розподіленого діагностичного ядра.

Діагностичні графи, які ілюструють різні принципи діагностування показано на рис. 12.4, де а) – централізований з послідовною процедурою; б) – централізований з послідовно-паралельною процедурою; в) – принцип розкручення; г) – принцип розподіленого ядра. Буквою i позначено час проведення тестування.

Особливості діагностики

При роботі СОС виникає певна група відмов, які не є характерними для систем і процесорів інших типів. На пошук і виявлення несправностей в таких системах впливають такі фактори [5]:

- накопичення помилок при переході від одного етапу обчислень до іншого;
- складність визначення моменту і місця виникнення несправності (зумовлені, насамперед, високою швидкістю і великим обсягом поступаючої інформації);
- складність спряження реальних потоків даних і виникнення завад в лініях зв'язку (швидкодійні системи можуть бути несправними і при працюючих складових частинах).

Хоча більшість алгоритмів опрацювання сигналів можна представити у формалізованому вигляді, сьогодні не існує однотипних підходів щодо діагностики і контролю відповідних структур. Здебільшого розглядаються задачі контролю і діагностики тільки окремих груп алгоритмів і їх апаратної реалізації.

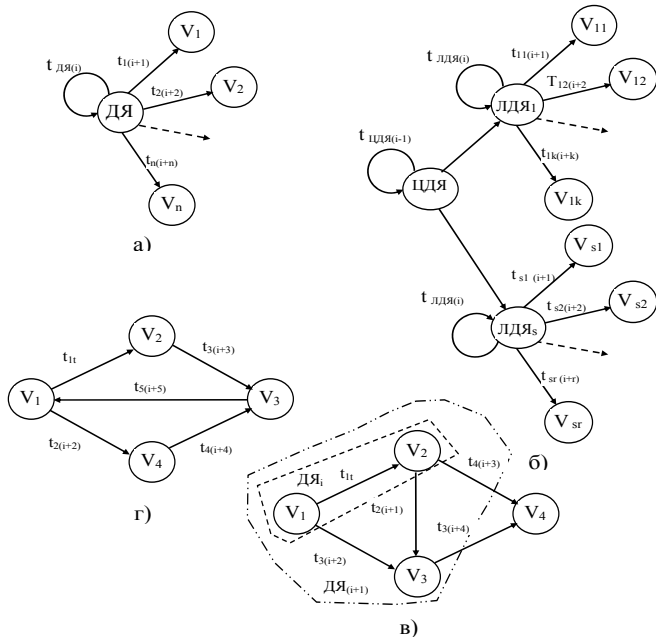


Рис.12.4. Діагностичні графи

Таким чином аналіз задач діагностики, варіантів реалізації алгоритмів, особливостей діагностики і контролю пристроїв опрацювання сигналів показує, що ефективними є такі шляхи діагностики і контролю:

- використання непрямих методів контролю;
- дослідження можливості введення мінімальної кількості резервних елементів;
- дослідження можливості перевірки процесорів і систем на виконання задач в РРЧ з використанням мінімального об'єму апаратури контролю;
- дослідження можливості використання самодіагностичних властивостей.

12.2. Ієрархічність засобів діагностики та контролю процесорів та систем опрацювання сигналів

Ієрархічність засобів діагностики відповідає ієрархічності обчислювальних засобів. Тому розглядається ієрархічність на рівні: систем, процесорів та окремих вузлів.

Використовуються такі засоби діагностики на рівні:

- систем – зовнішня контрольна- діагностична апаратура (КДА);
- процесорів – КДА + програмний тестовий контроль;
- вузлів – програмний тестовий контроль + вбудований апаратний контроль.

Головними спільними завдання при їх використанні є:

- забезпечення контролю і діагностики в режимі реального часу;
- мінімальні додаткові діагностичні і контролюючі апаратні засоби;
- забезпечення перевірки вхідних та вихідних сигналів;
- забезпечення контролю і діагностики в важких (промислових) умовах експлуатації.

Для прикладу розглянемо варіант реалізації системи опрацювання сигналів когерентно-імпульсного локатора, схема якої показано на рис. 12.5 [8].

Нехай процесор ШПФ обчислює в РРЧ 32-х або 256-ти точкове амплітудне ДПФ (залежно від режиму роботи системи) зі зважуванням над черезперіодними вибірками $S(n, i)$ (n – поточний номер елемента дальності, i – номер черезперіодної вибірки в межах одного циклу обчислень) комплексної огинаючої вхідного сигналу, що приймаються з N суміжних елементів дальності (віддалі до об'єкту), визначає ковзаюче середнє значення сигналу.

В пороговому пристрої інформація, що поступає з процесора ШПФ порівнюється з кодом порога, формуються ознаки порівняння, які перемножуються на значення режектуючої функції.

У вузлі керування та синхронізації формується набір синхроімпульсів і зондуючих сигналів, які використовуються для синхронізації роботи всіх вузлів радіолокаційного комплексу.

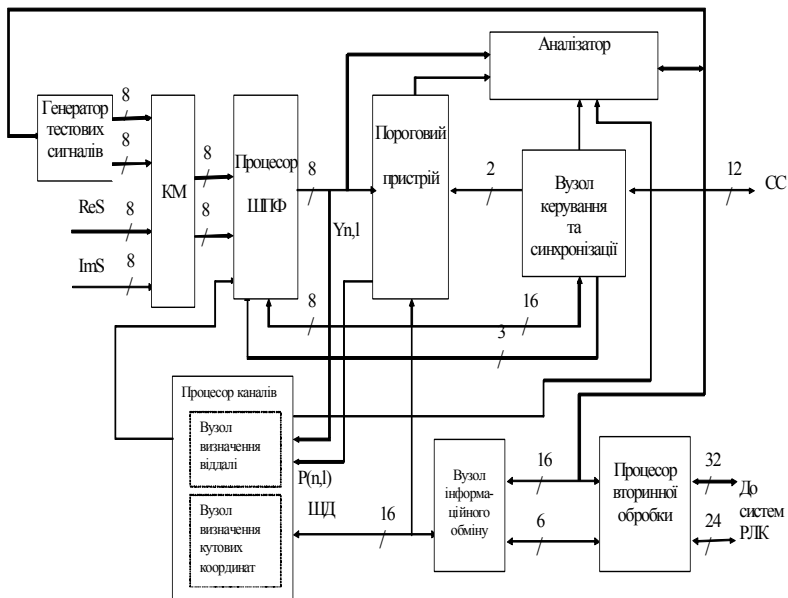


Рис. 12.5. Структурна схема системи цифрової обробки радіолокаційних сигналів із засобами діагностики

Вузол інформаційного обміну використовується для обміну між процесорами первинного і вторинного опрацювання.

На вхід системи опрацювання надходять з приймача проміжкової частоти 8-мі розрядні значення ReS і ImS , які приймаються і обробляються в стробі приймання інформації, що є одним з імпульсів слідкування, який виробляється в каналі визначення віддалі. Інформація з виходу процесора ШПФ $Y(n,l)$ (l – номер коефіцієнта Фур’є – номер гармоніки) поступає на вузол визначення віддалі (на схему визначення максимальної гармоніки). В пороговому пристрої значення $Y(n,l)$ порівнюються з еталонними значеннями (порогом). Результати порівняння $P(n,l)$ поступають на схему формування коду дальності у вузлі визначення віддалі до цілі.

Розробка стратегії діагностування та контролю

Вибір стратегії діагностування залежить від призначення пристрою, режимів експлуатації, заданих обмежень на вагогабаритні показники. В загальному випадку цифрові пристрої (системи) містять апаратно-програмні засоби діагностування, причому вбудовані засоби контролю (апаратні) виконують перевірку в процесі її основної роботи, а засоби тестового діагностування в режимі контролю, регламентних і профілактичних робіт. Причому в тестовому діагностуванні вирішується дві задачі:

- знаходження (генерація) тестової послідовності, яка може виявити заданий клас несправностей;
- моделювання несправностей заданого класу (верифікація тесту), з метою доведення, що ці несправності можуть бути виявлені розробленим тестом.

Розмірність задач і трудомісткість їх розв'язку різко збільшується з ростом складності вузлів. Для виявлення тільки однієї несправності для перевірки комбінаційних схем потрібно 2^l тестових наборів для комбінаційних схем і 2^{r+c} для схем з пам'яттю. Тому можливість тестування передбачають на етапі розробки виробів. Аналіз показує, що пристрої обчислення ШПФ в такій системі (див. рис.12.5) доцільно перевіряти на двох рівнях: в складі системи (зовнішня КДА) і методами вбудованих тестів. При використанні зовнішньої КДА достовірність роботи вузла ШПФ можна перевірити шляхом аналізу його амплітудно-частотної характеристики.

Для побудови самотестуючих архітектур ШПФ - схем доцільно використати такий алгоритм [7].

Відомо, що ШПФ-схеми мають $\log_2 N$ ступенів, індексованих через s , де $0 \leq s < \log_2 N$. Кожна ступінь має $N/2$ модулів типу "метелик". На рис. 12.6 наведена схема "метелика" з позначеними місцями можливого розташування функціональних помилок.

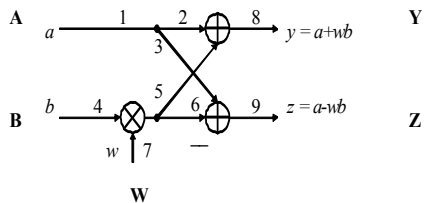


Рис.12.6. Схема „метелика” з місцями розташування функціональних помилок

На рис. 12.6 наведені входи (A, B), виходи (Y, Z), повертаючий (W) множник $W_N^{nk} = e^{-j\frac{2\pi}{N}nk}$, тестові значення a , b і w , значення виходів y , z при надходженні тестових значень для конкретного “метелика”. Тобто є дев’ять помилок, які можуть моделюватися різними способами, наприклад, неправильна робота суматорів може бути змодельована як навмисні помилки на вихідних лініях цих суматорів. Помилки розрізнені чотирма еквівалентними множинами помилок: $\{1\}$, $\{4,7\}$, $\{2,5,8\}$ та $\{3,6,9\}$, причому, дві помилки не можуть бути розрізнені, якщо вони знаходяться у тій самій еквівалентній множині.

Визначаючи “метелик”, через множину T -пар тестових входів a_j та b_j , $0 < j \leq T$, повинні бути отримані усі помилки, які можна визначити.

Визначення та оцінка основних параметрів вузлів контролю у відмовостійких системах опрацювання сигналів

Вузли контролю і діагностики в системах повинні:

- забезпечувати видачу тестових масивів з генератора на вузол, який перевіряється, в режимі реального часу;
- забезпечувати видачу інформації з вузла, який перевіряється, на аналізатор в режимі реального часу;
- час перевірки повинен бути наближеним до часу роботи в основних режимах;
- мати мінімальні додаткові затрати на вузли контролю та діагностики в порівнянні з апаратними затратами на основний вузол.

12.3. Процес формування АЧХ

Для обчислення АЧХ нерекурсивних ЦФ здебільшого застосовують метод передатних функцій [2]. Від передатної функції, яка в загальному вигляді записується як многочлен виду:

$$H(Z) = a_0 + a_1 Z^{-1} + a_2 Z^{-2} + \dots + a_{N-1} Z^{-(N-1)},$$

можна перейти до комплексної частотної характеристики, підставляючи $Z = e^{j\omega}$, що відповідає перетворенню Фур'є

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} a_n e^{-j\omega n} = |H(e^{j\omega})| e^{arg[H(e^{j\omega})]} \quad (12.3)$$

де $\{a\}$ – коефіцієнт фільтра N -го порядку; $|H(e^{j\omega})|$ – АЧХ фільтра; $arg[H(e^{j\omega})]$ – ФЧХ фільтра.

Згідно з формулою Ейлера для комплексних чисел $e^{j\omega} = \cos \omega + j \sin \omega$, перетворимо комплексну частотну характеристику ЦФ:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} a_n \cdot \cos \omega n + j \sum_{n=0}^{N-1} a_n \cdot \sin \omega n.$$

Виходячи з останнього виразу, можна визначити АЧХ.

Можливий такий варіант обчислень: визначення дискретизованих АЧХ і ФЧХ як результату вагового сумування в узгодженому фільтрі дискретизованих значень потрібної для дослідження сітки гармонічних коливань з одиничною амплітудою і постійною початковою фазою:

$$u_{si} = \sin 2\pi f_i t; \quad u_{ci} = \cos 2\pi f_i t,$$

де u_{si} і u_{ci} – ортогональні складові аналітичного сигналу $u_i = \exp(j2\pi f_i t)$.

Результат проходження складових через узгоджений з сигналом фільтр – вихідна інформація для обчислення модуля (АЧХ) і аргумента (ФЧХ) коефіцієнта передачі. Згідно з алгоритмом, $u_{c\text{ вих. }i}$ визначається як сума зважених з відповідними коефіцієнтами $\{a\}$ значень гармонійної одиничної напруги u_{ci} в моменти часу, кратні періоду дискретизації T :

$$u_{c\text{ вих. }i} = \sum_{k=0}^N a_k u_{cs}(kT) = \sum_{k=0}^N a_k \cos 2\pi f_i kT,$$

де N - порядок фільтра, $\cos 2\pi f_i kT$ – фазовий коефіцієнт одиничної косинусної напруги. Його значення детерміноване і може зберігатися в запам'ятовуючому пристрої для реального каналу, якщо відомі сітка частот, період дискретизації аналізованої вхідної послідовності.

В загальному випадку, якщо фільтр N -го порядку,

$$u_{c\text{ вих }i} = \sum_{k=0}^N a_k \cos 2\pi f_i kT.$$

Аналогічно можна визначити реакцію фільтра на синусоїдальну вхідну напругу з нульовою початковою фазою:

$$u_{s\text{ вих }i} = \sum_{k=0}^N a_k \sin 2\pi f_i kT.$$

Для будь-якої з частот f_i можна визначити значення АЧХ і ФЧХ:

$$|k(f_i)| = \sqrt{u_{ci}^2 + u_{si}^2}; \quad f_i = \arctg(u_{si} / u_{ci}) \quad (12.4)$$

За отриманими результатами визначається АЧХ.

Розглянемо процедуру діагностики схеми ШПФ на прикладі системи опрацювання сигналів когерентно-імпульсної радіолокаційної станції (рис. 12.4), де для вимірювання доплерівської частоти використовується n каналів з погодженими фільтрами. Нехай процесор виконує N -точкове амплітудне дискретне перетворення Фур'є згідно з такою формулою [5,10]:

$$Y(n,l) = \sum_{i=0}^{N-1} \dot{U}(n,i)W(i)e^{-j(2\pi/N)li}, \quad (12.5)$$

де N визначає розмірність перетворення, n – номер елемента віддалі, l – номер гармоніки, i – номер періоду повторення в межах інтервалу обчислення ДПФ, $W(i)$ – вагова функція.

Тоді на виході генератора тестових сигналів маємо сигнал $\dot{U}(n,i)$:

$$\dot{U}(n,i) = Ae^{+j2\pi(Sl+Q)i/SN}, \quad (12.6)$$

де A – амплітуда сигналу, S – кількість частотних діапазонів між сусідніми l , Q – визначає смугу перевірки АЧХ ($Q = -[\overline{S_{m-1} + S_i}] \cdot [\overline{S_{p-1} + S_i}]$, де m, p – кількість

гармонік, в діапазоні яких (відносно l) перевіряється АЧХ, $m = \overline{1, N-1}$, $p = \overline{1, N-1}$, s_i – поточне значення частотного діапазону між сусідніми l).

Алгоритм формування вхідних даних полягає у видачі на кожному етапі обчислень синфазної і квадратурної складових комплексного сигналу, фаза яких визначається на значення Q на двох сусідніх періодах, на кожному з яких вираховується одне значення $U(i)$.

Процедура діагностики відбувається таким чином. Для пристрою ШПФ задається значення гармоніки l_j . На інформаційні входи поступає вхідний сигнал $\dot{U}(n, i)$. Зміна значень $\dot{U}(n, i)$ (синфазна і квадратурна складові) на вході процесора відбувається на кожному періоді повторення (по i). Одне значення $Y(n, l)$ визначається сумуванням по i . Після того змінюється частота поступлення $\dot{U}(n, i)$, зміна задається значенням $s_i = \overline{0, S}$, і вираховується наступне значення $Y(n, l)$. Повна АЧХ, для заданого l_j , отримується після поступлення на вхід SN значень вхідного сигналу. На практиці обмежуються перевіркою АЧХ для $\pm 3l$, відносно l_j . Після перевірки амплітудно-частотних характеристик для всіх гармонік і елементів віддалі процес діагностики завершується. В ідеальному випадку характеристики всіх АЧХ повинні бути ідентичними.

Тобто, при використанні такого підходу процес діагностики пристрою ШПФ розбивається на три етапи:

- задання значень для отримання числової послідовності вхідних сигналів;
- визначення значень $Y(n, l)$ реальної АЧХ;
- порівняння значень ідеальної і реальної АЧХ в кожній точці виміру.

12.4. Визначення і дослідження виду АЧХ

Нехай задано проаналізувати АЧХ фільтра з такими параметрами сигналу:
 $l = 0, 1, \dots, 31$; $A = 1, 2, \dots, 100$; $S = 8, 16$; $Q = -64 \dots 64$; $N = 0, 1, \dots, 31$.

Згідно з виразом (12.6), формуємо масиви значень синусів та косинусів:

for ($i = 0$; $i < N$; $i++$)
 $\{$

```

sin0[i] = sin(2*pi*i/N);
cos0[i] = cos(2*pi*i/N);
}
for(i = 0; i < SN; i++)
{
sin1[i] = sin(2*pi*i/SN);
cos1[i] = cos(2*pi*i/SN);
}

```

де N – розмір перетворення, а SN – добуток N та S .

Обчислюється значення дійсної та уявної складової як результат добутку $Ae^{+j2\pi(Sl+Q)i/SN}$ та $e^{-j(2\pi/N)li}$ (див. формулу (12.5)). Значення синусів та косинусів вибираємо із масивів, отриманих на попередньому етапі:

```

for(i = 0; i < N; i++)
{
a = (abs(S*l+Qmin)*i)%SN;
b = l*i%N;
Re += A*cos1[a]*cos0[b] + A*sin1[a]*sin0[b];
Im += A*cos1[a]*sin0[b] - A*sin1[a]*cos0[b];
}

```

Обчислюємо значення координати Y для відображення результату:

$$Y[j] = \text{sqrt}(\text{pow}(\text{Re}, 2) + \text{pow}(\text{Im}, 2)).$$

При запуску програми з'являється вікно, показане на рис. 12.7.

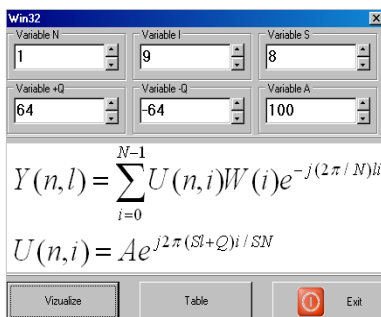


Рис.12.7. Головне вікно програми.

Зміна параметрів проводиться шляхом натиску мишкою на стрілки, що містяться біля кожного поля праворуч. Графік АЧХ показано на рис.12.8, а фрагмент значень АЧХ – в табл.12.1.

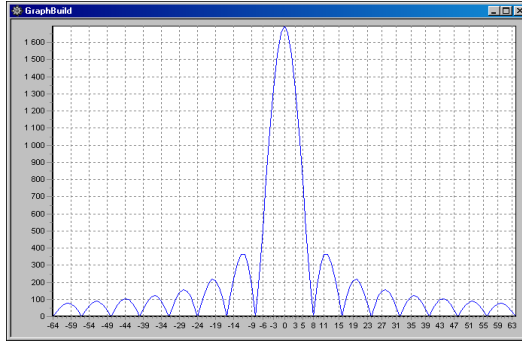


Рис.12.8. Графік АЧХ.

Таблиця 12.1. Значення АЧХ (фрагмент)

X	Y
-64	2,75764738977458E-14
-63	29,0419697584423
-62	54,3501896863475
-61	71,9446715207641
-60	78,9208447262436
-59	73,9197598843918
-58	57,3763281911878
-57	31,5023737798159
-56	4,93558757932881E-14
-55	32,4588524304012
-54	60,9147189069376
-53	80,8665934638205
-52	88,9710624140459
-51	83,5879042571257
-50	65,0853182734446

12.5. Завдання для самоперевірки

1. Яке з діагностичних ядер передбачає послідовне під'єднання процесорів системи з метою перевірки каналів зв'язку та правильності виконання комплексної задачі:

А) централізоване з послідовною процедурою; Б) централізоване з послідовно-паралельною процедурою; В) з використанням принципу розкручення; Г) розподілене; Д) немає правильної відповіді.

2. Достовірність правильної роботи забезпечується шляхом:

А) тестування виконання робочих алгоритмів окремими процесорами; Б) аналізу виконання всіх команд процесорів; В) правильності виконання комплексної задачі; Г)) правильності виконання комплексної задачі в режимі реального часу; Д) немає правильної відповіді.

3. Максимальна кількість можливих функціональних помилок при реалізації “метелика” ШПФ за основою 2 становить:

А) 4; Б) 8; В) 9; Г) 10; Д) немає правильної відповіді

4. При правильній роботі процесора ШПФ графіки АЧХ для всіх гармонік ϵ :

А) однаковими; Б) різними; В) вид залежить від номера гармоніки; Г) відрізняються для парних і непарних гармонік; Д) немає правильної відповіді.

5. Для повної перевірки процесора ШПФ необхідно проаналізувати значення АЧХ для:

А) нульової гармоніки; Б) всіх гармонік; В) непарних гармонік; Г) парних гармонік; Д) немає правильної відповіді.

Правильні відповіді до завдань:

1) В; 2) Г; 3) В; 4) А; 5) Б.

СПИСОК ЛІТЕРАТУРИ

1. Айфичер Э. Цифровая обработка сигналов: практический подход / Э. Айфичер, Б. Джервис – М.: Изд. дом “Вильямс”, 2004. – 992 с.
2. Бабак В. П. Обробка сигналів: підручник для вузів / В. П. Бабак, А. І. Хандецький, Е. Шрюфер – Київ: Либідь, 1996. – 390 с.
3. Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов / Р. Блейхут – М.: Мир, 1989. – 448 с.
4. Бондарев В. Н. Цифровая обработка сигналов, методы и средства: підручник для вузів / В. Н. Бондарев, Г. Трестер, В. С. Чернега. – Харьков: Конус, 2001. – 398 с.
5. Ваврук Є. Я. Організація контролю та діагностики ШПФ-схем у режимі реального часу в системах опрацювання сигналів / Є. Я. Ваврук // Вісник НУ “Львівська політехніка”, комп’ютерні системи та мережі. – 2004. – № 523. – С 24–29.
6. Ваврук Є. Я. Компресія зображень з використанням штучних нейронних мереж / Є. Я. Ваврук, І. Є. Ваврук // Зб. наук. праць “Моделювання та інформаційні технології”. – Інститут проблем моделювання в енергетиці НАНУ, 2017 – Вип. 78. – С. 148-154.
7. Ваврук Є. Я. Організація модулів відновлення відмовостійких систем опрацювання сигналів / Є. Я. Ваврук, В. В. Грос // Вісник НУ “Львівська політехніка”, комп’ютерні науки та інформаційні технології. – 2010. – № 672. – С 23–29.
8. Ваврук Є. Я. Моделі контролю і діагностики систем опрацювання сигналів / Є. Я. Ваврук, В. А. Коваль // Зб. наук. праць “Моделювання та інформаційні технології”. – Інститут проблем моделювання в енергетиці НАНУ, 2006 – Вип. 35. – С. 131–139.
9. Ваврук Є. Я. Швидкий алгоритм виконання каскаду тригонометричних перетворень на основі спільної базової операції / Є. Я. Ваврук, О. Л. Лашко // Вісник НУ “Львівська політехніка”, комп’ютерні системи та мережі. – 2003. – № 498. – С 183–189.
10. Ваврук Є. Я. Цифрове опрацювання сигналів та зображень, алгоритми та реалізація: навчальний посібник / Є. Я. Ваврук, Р. Б. Попович – Національний університет “Львівська політехніка”, 2008. – 147 с.

11. Винцюк Т. К. Анализ, распознавание и смысловая интерпретация речевых сигналов / Т. К. Винцюк. – Киев: Наукова думка, 1987. – 264 с.
12. Даджион Д. Цифровая обработка многомерных сигналов / Д. Даджион, Р. Мерсеро. – М.: Мир, 1988. – 488 с.
13. Дробик О. В. Цифрова обробка аудіо- та відеоінформації у мультимедійних системах: навчальний посібник / О. В. Дробик, В. В. Кідалов, В. В. Коваль, Б. Я. Костік, В. С. Лазебний, Г. М. Розорінов, Г. О. Сукач. – К.: Наукова думка, 2008. – 144 с.
14. Иванова Н. Н. Особенности реализации алгоритма БПФ на ПЛИС типа FPGA / Н. Н. Иванова, Н. А. Галанина, Д. В. Моисеев // Вестник Чувашского университета. – 2018. – № 3. – С 182–187.
15. Кестер У. Проектирование систем цифровой и смешанной обработки сигналов / У. Кестер. – М.: Техносфера, 2010. – 328 с.
16. Лайонс Р. Цифровая обработка сигналов / Р. Лайонс – М.: Бинум-Пресс, 2006. – 656 с.
17. Мороз І. В. Критерії оцінки відмовостійкості систем опрацювання сигналів / І. В. Мороз, Є. Я. Ваврук // Вісник НУ “Львівська політехніка”, комп’ютерні системи та мережі. – 2014. – № 806. – С. 175–180.
18. Попович Р. Б. Використання довжин зразків при розпізнаванні мови / Р. Б. Попович // Вісник НУ “Львівська політехніка”, комп’ютерна інженерія та інформаційні технології. – 2001. – № 433. – С. 190–193.
19. Попович Р. Б. Складові системи розпізнавання мови / Р. Б. Попович // Вісник НУ “Львівська політехніка”, комп’ютерні системи та мережі. – 2001. – № 437. – С. 121–124.
20. Рабинер Л. Теория и применение цифровой обработки сигналов / Л. Рабинер, Б. Гоулд – М.: Мир, 1978. – 848 с.
21. Сато Ю. Без Паники! Цифровая обработка сигналов / Ю. Сато – М.: Додэка-XXI, 2010. – 176 с.
22. Сергиенко А. Б. Цифровая обработка сигналов: учебник для вузов / А. Б. Сергиенко – СПб.: Питер, 2013. – 768 с.
23. Солонина А. И. Основы цифровой обработки сигналов: курс лекций / А. И. Солонина, Д. А. Улахович, С. М. Арбузов, Е. Б. Соловьева, И. И. Гук – СПб.: БХВ-Петербург, 2012. – 768 с.

24. Яцимірський М. М. Швидкі алгоритми ортогональних тригонометричних перетворень / М. М. Яцимірський // – Львів: Академічний Експрес, 1997. – 219 с.
25. Gales M. The Application of Hidden Markov Models in Speech Recognition / M. Gales, S. Young // Foundations and Trends in Signal Processing. – 2007. – Vol. 1, No. 3. – P. 195–304.
26. Hudson G. JPEG-1 standard 25 years: past, present, and future reasons for a success / G. Hudson, A. Léger, B. Niss, I. Sebestyén, J. Vaaben // Journal of Electronic Imaging. – 2018. – Vol. 27, No. 4. – 040901.
27. Rabiner L. Fundamental of Speech Recognition / L. Rabiner, B. H. Juang, B. Yegnanarayana – New Delhi, India: Pearson Education Inc., 2009. – 497 p.
28. Shapiro J. M. Embedded Image Coding Using Zerotrees of Wavelet Coefficients / J. M. Shapiro // IEEE Transactions on Signal Processing. – 1993. – Vol. 41, No. 12. – P. 3445–3462.
29. Varichenko L. V. Digital filtering of random signals using number theoretic transforms / L. V. Varichenko, R. B. Popovich, M. A. Rakov // Radioelectronics and Communications Systems. – 1987. – Vol. 30, Issue 12. – P. 36–39.
30. Xiong Z. The Essential Guide to Image Processing / Z. Xiong, K. Ramchandran (Ed. A. C. Bovik) – NY: Academic Press, 2009. – P. 463–493, Ch. 18: Wavelet Image Compression.
31. Young S. Large vocabulary continuous speech recognition / S. Young // IEEE Signal Processing Magazine. – 1996. – Vol. 13, No. 5. – P. 45–57.
32. Digital signal processor fundamentals and system design [Електронний ресурс] – Режим доступу до ресурсу: [https://core.ac.uk > download > pdf](https://core.ac.uk/download/pdf)
33. Neuroprocessor NeuroMatrix NM6403 architecture – EURASIP [Електронний ресурс] – Режим доступу до ресурсу: <https://www.urasip.org>
34. Parallel 2D FFT implementation on FPGA suitable for real-time MR image processing [Електронний ресурс] – Режим доступу до ресурсу: <https://aip.scitation.org>

Навчальне видання

Ваврук Євгеній Ярославович

Лашко Оксана Любомирівна

Попович Роман Богданович

АЛГОРИТМИ ТА ЗАСОБИ ОБРОБКИ СИГНАЛІВ

Навчальний посібник

Авторська підготовка

Відповідальний за випуск – Богдан Пелех

Підписано до друку 15.05.2021 р. Формат 60x84/16.
Папір офсетний. Друк – різнограф. Обсяг 13,95 ум.-друк. арк.
Наклад 100 прим. Зам. № 10.

Видавництво “СПОЛОМ”. 79008 Україна,
м. Львів, вул. Краківська, 9. Тел.: (380-32) 297-55-47.
E-mail: spolom_lviv@ukr.net. Свідоцтво суб'єкта видавничої діяльності:
серія ДК, № 2038 від 02.02.2005 р.

Друк ФОП Гуменецький М. В. 81630 Львівська обл.,
Миколаївський р-н, с. Гонятичі, вул. Польова, 10.
Свідоцтво фізичної особи підприємця: № 083613 від 18.08.2008 р.