

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
ДОНЕЦЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ

Кафедра електронної техніки

Конспект лекцій

з дисципліни

«Комп'ютерна обробка сигналів»

для студентів спеціальності 123 Комп'ютерна інженерія

усіх форм навчання

Луцьк, 2023

УДК 004.383(07)

К 64

Конспект лекцій з дисципліни «Комп'ютерна обробка сигналів» для студентів спеціальності 123 Комп'ютерна інженерія усіх форм навчання [Електронний ресурс] / уклад. С.О. Ковальов, В.В. Шамаєв. – Луцьк : ДонНТУ, 2023. – 124 с.

Конспект лекцій містить основні теоретичні положення, терміни і визначення з дисципліни «Комп'ютерна обробка сигналів» студентами бакалаврату ДонНТУ, що навчаються за спеціальністю 123 Комп'ютерна інженерія усіх форм навчання. Наведено основи теорії, методи, алгоритми цифрової обробки сигналів; виділено операційний базис комп'ютерних систем цифрової обробки сигналів. Проаналізовано сучасну елементну базу, структурну організацію пам'яті, інтерфейсів та значну увагу приділено базовим структурам комп'ютерних систем цифрової обробки сигналів. Розкрито принципи побудови та методи оцінки основних параметрів комп'ютерних систем цифрової обробки сигналів.

Конспект лекцій з дисципліни «Комп'ютерна обробка сигналів» розроблено за участю О.В. Самощенко. Укладачі щиро вдячні йому за співпрацю та надані матеріали.

Укладачі: С.О. Ковальов, доц., к.т.н., доц. каф. ЕТ.

В.В. Шамаєв, доц., к.т.н., доц. каф. ЕТ.

Рецензент: Н.О. Маслова, доц., к.т.н., доц. каф. ПМІ.

Відповідальний за випуск: С.О. Ковальов, доц., к.т.н., зав. каф. ЕТ.

Затверджено навчально-методичним відділом ДонНТУ

Протокол № 4 від 26.12.2023 р.

Розглянуто на засіданні кафедри електронної техніки

Протокол № 5 від 11.12.2023 р.

ЗМІСТ:

Вступ	4
1. Структура комп'ютерної обробки сигналів.	6
1.1 Структура та компоненти системи цифрової обробки сигналів.	6
2. Цифрова фільтрація.	10
2.1 Нерекурсивний і рекурсивний цифрові фільтри.	11
2.2 Медіанна фільтрація	13
3. Дискретне перетворення Фур'є	14
4. Швидкі алгоритми ортогональних тригонометричних перетворень	16
4.1 Алгоритми швидкого перетворення Фур'є комплексної послідовності.	16
4.2 Швидкі алгоритми дискретного перетворення Хартлі.	22
4.3 Швидкі алгоритми дискретних косинус і синус-перетворень Фур'є.	26
4.4 Алгоритми швидкого косинусного та синусного перетворень	29
4.5 Ортогональні швидкі хвильові перетворення.	31
4.6 Швидкі алгоритми двовимірних ортогональних тригонометричних перетворень.	34
5. Алгоритми функціонування, програмування і навчання штучних нейронних мереж паралельної обробки сигналів.	36
6. Кореляція.	38
7. Сортвання.	40
8. Операційний базис комп'ютерних систем обробки сигналів	41
9. Елементна база комп'ютерних систем цифрової обробки сигналів.	44
9.1 Мікропроцесори ЦОС.	44
9.2 Трансп'ютери та однорідні обчислювальні середовища.	52
9.3 Спеціалізовані НВІС.	57
10. Особливості структурної організації пам'яті комп'ютерних систем цифрової обробки сигналів.	60
10.1 Багатомодульна пам'ять.	60
10.2 Пам'ять з паралельним доступом.	64
10.3 Кеш-пам'ять.	65

10.4 Особливості доступу до пам'яті.	67
10.5 Організація пам'яті комп'ютерних систем цифрової обробки сигналів.	70
10.6 Принципи побудови пам'яті комп'ютерних систем ЦОС.	74
11. Інтерфейси комп'ютерних систем ЦОС.	76
12. Узагальнена структура та шляхи вдосконалення характеристик комп'ютерних систем обробки сигналів.	80
13. Засоби реалізації алгоритмів і підходи до побудови комп'ютерних систем обробки сигналів.	89
14. Базові структури комп'ютерних систем цифрової обробки сигналів.	94
14.1 Універсальна комп'ютерна система ЦОС.	94
14.2 Спеціалізовані комп'ютерні системи ЦОС.	95
14.3 Реконфігуровані комп'ютерні системи ЦОС.	101
15. Базові структури процесорів ЦОС.	104
15.1 Алгоритмічні процесори ЦОС.	104
15.2 Програмовані процесори цифрової обробки сигналів.	107
15.3 Процесори швидких косинус- і синус-перетворень Фур'є.	109
16. Задачі і основні принцип проектування комп'ютерних систем цифрової обробки сигналів.	115
17. Оцінка основних параметрів комп'ютерних систем ЦОС.	120
17.1 Оцінка витрат обладнання.	120
17.2 Продуктивність комп'ютерних систем обробки сигналів.	121
17.3 Ефективність використання обладнання.	122
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.	124

ВСТУП

Розвиток комп'ютерних систем цифрової обробки сигналів (ЦОС) характеризується розширенням галузей їх застосування, у значній частині з яких вимагається обробка у реальному часі інтенсивних потоків даних за складними алгоритмами на апаратних засобах, що задовольняють жорстким умовам експлуатації та обмеженням у частині габаритів і споживаної потужності.

Комп'ютерні системи ЦОС – це системи реального часу, які призначені для розв'язання задач приймання, обробки, зменшення надлишковості та передавання інформації в реальному часі. Сучасний етап розвитку теорії ЦОС тісно пов'язаний з інтенсивним впровадженням методів цифрових сигналів, які орієнтовані на застосування мікропроцесорів ЦОС, архітектурно перепрограмованих надвеликих інтегральних схем (НВІС) на базі програмованих логічних інтегральних схем (ПЛІС) і багатопроцесорних систем, побудованих на їх основі. ПЛІС, завдяки великій ємності логічних вентилів на кристалі і високій робочій тактовій частоті, міцно займають свою нішу між спеціалізованими НВІС і універсальними мікропроцесорами ЦОС.

Комп'ютерні системи ЦОС мають високу продуктивність, забезпечують високу гнучкість за рахунок архітектурної адаптації перепрограмованих НВІС до структури алгоритмів розв'язання задачі. В тих випадках, коли комп'ютерна система орієнтована на розв'язання складних задач з значною кількістю переходів і логічних операцій над вхідними потоками з різною інтенсивністю надходження даних, найбільша ефективність використання обладнання досягається при спільному використанні ПЛІС і мікропроцесорів ЦОС.

Сучасна концепція побудови комп'ютерних систем ЦОС базується на використанні потенційних можливостей сучасної елементної бази (мікропроцесорів ЦОС, спеціалізованих і перепрограмованих НВІС) і використанні методів розпаралелювання та конвеєризації при розв'язанні конкретних задач.

У паралельно-конвеєрних комп'ютерних системах ЦОС висока продуктивність та ефективність використання обладнання досягається тільки в тих випадках, коли їх архітектура адаптується до інтенсивності надходження потоків даних і адекватно відображає структуру алгоритму розв'язання задачі.

Паралелізм обробки даних у комп'ютерних системах ЦОС висуває свої вимоги до організації пам'яті, які в першу чергу пов'язані з необхідністю забезпечення паралельного доступу до множини даних і підтримкою швидкісного обміну з операційними пристроями, процесорними елементами та багатоканальними пристроями введення-виведення, тобто пам'ять повинна бути швидкодіючою та паралельною. У паралельній пам'яті висока швидкодія та ефективність використання обладнання досягається шляхом адаптації її архітектури до структур даних і алгоритмів розв'язання задачі.

Зменшення енергоспоживання, габаритів, часу та вартості розробки комп'ютерних систем ЦОС досягається у випадку коли архітектура є регулярною, модульною та орієнтованою на НВІС-реалізацію. Для створення високоефективних комп'ютерних системи ЦОС доцільно використовувати інтегрований підхід, який охоплює архітектуру, сучасну елементну базу, методи та алгоритми ЦОС та враховує інтенсивності надходження даних і вимоги конкретних застосувань.

Метою викладання дисципліни «Комп'ютерна обробка сигналів» – є вивчення основ методології: методів і алгоритмів ЦОС, елементної бази, архітектури та методів проектування комп'ютерних систем ЦОС у реальному часі.

1. Структура комп'ютерної обробки сигналів

1.1 Структура та компоненти системи цифрової обробки сигналів.

1.1 Структура та компоненти системи цифрової обробки сигналів

Поведінка деякої фізичної величини в часі (інтенсивність звуку, інтенсивність світла, температура, тиск та ін.) сприймається як сигнал, який є неперервною функцією часу. Сигнали, визначені у всі моменти часу, називаються аналоговими (рис.1.1а), які за допомогою датчиків перетворюються в електричні, наприклад, мікрофон для перетворення звуку.

Процес перетворення аналогового сигналу в послідовність відліків називається дискретизацією, а результат такого перетворення – дискретним сигналом (рис.1.1б). Як правило, відліки беруться через рівні проміжки часу T_d , називаються періодом або кроком дискретизації. Величина, обернена періоду дискретизації, називається частотою дискретизації – $f_d = 1/T_d$. Представлення сигналу набором дискретних відліків веде до втрати інформації в проміжках між відліками.

При обробці сигналів в комп'ютерних системах його відліки представляються у вигляді двійкових чисел, що мають обмежене число розрядів. Процес перетворення відліків сигналу в числа називається квантуванням по рівню, а виникаючі при цьому помилки округлення – шумами квантування. Сигнал, дискретний в часі і квантований по рівню, називається цифровим сигналом (рис.1.1в).

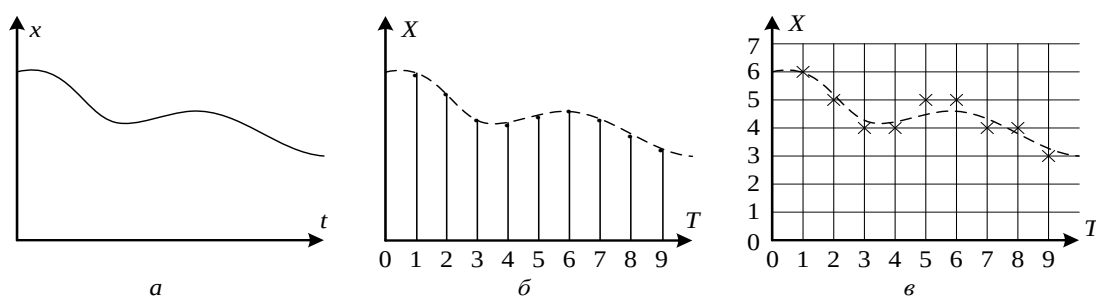


Рис.1.1 Сигнали: а) аналоговий; б) дискретний; в) цифровий

Метою обробки сигналу є представлення його в зручному для сприйняття вигляді, формування сигналу з заданими параметрами, виділення з сигналу та внесення у нього корисної інформації.

В залежності від використовуваних апаратних засобів обробка сигналів і зображень може бути аналоговою або цифровою. Добре вивчені та широко розповсюджені методи і засоби аналогової обробки сигналів і зображень, які підвищенням вимог до складності, точності та стабільності обробки у же не можуть конкурувати з комп'ютерними засобами обробки сигналів і зображень. Комп'ютерні засоби ЦОС, в порівнянні з аналоговими, мають наступні переваги:

- характеристики комп'ютерних засобів абсолютно стабільні і не змінюються при зміні зовнішніх умов;
- дозволяють реалізацію ряду операцій і алгоритмів, що принципово не реалізуються з допомогою аналогових елементів; забезпечують швидку (в основному – програмним шляхом) модифікацію або заміну алгоритмів обробки;
- мають високу надійність, малі габарити і просто повторяються.

Для обробки аналогових сигналів цифровими засобами необхідні допоміжні перетворення на вході та виході системи. На вході аналоговий сигнал попередньо перетворюється в цифровий з допомогою аналого-цифрових перетворювачів (АЦП), а на виході цифровий сигнал перетворюється в аналоговий з допомогою цифро-аналогових перетворювачів (ЦАП). В цій останній процедурі найчастіше немає необхідності, якщо метою ЦОС є отримання інформації про сигнал. Розширення використання методів і засобів ЦОС пов'язано як із вдосконаленням основних метрологічних характеристик АЦП і ЦАП, так і розвитком НВІС-технології.

Структура системи цифрової обробки сигналів і зображень наведена на рис.1.2.

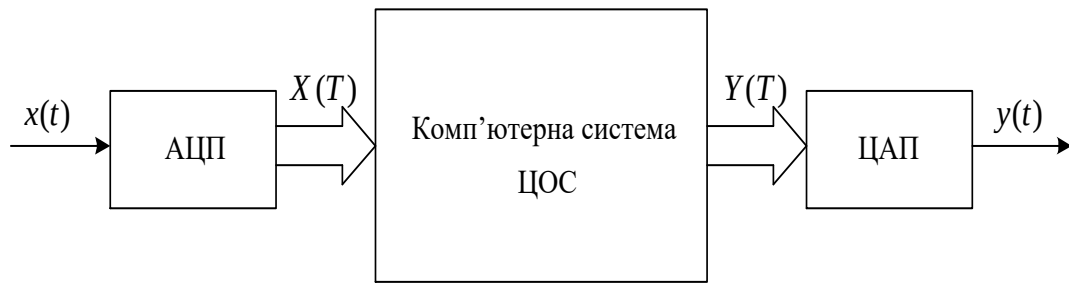


Рис.1.2 Структура системи цифрової обробки сигналів

На вхід системи ЦОС надходить аналоговий сигналів $x(t)$, який з допомогою АЦП перетворюється послідовність двійкових відліків $X(T)$, які надходять в комп'ютерну систему, де за оператором F виконується обробка. Результати обробки $Y(T) = F[X\{T\}]$ комп'ютерної системи є нова послідовність чисел, які з допомогою ЦАП перетворюються в аналоговий сигнал $y(t)$.

Успіхи в теорії ЦОС та інтенсивний розвиток НВІС-технології забезпечили створення високопродуктивних малогабаритних комп'ютерних систем обробки сигналів, які істотно розширили області застосування технологій обробки сигналів [1-10]. Основними компонентами систем ЦОС є апаратні, алгоритмічні, програмні та технологічні засоби.

Апаратні засоби систем ЦОС забезпечують перетворення, приймання, зберігання, обробку вхідної інформації та вивід результатів обробки. Вони складаються АЦП, комп'ютерної системи ЦОС і ЦАП.

АЦП призначені для перетворення аналогової інформації (у вигляді напруги) у цифровий код. Основними параметрами АЦП є:

- число розрядів p вихідного коду;
- роздільна здатність h – мінімальний квант вхідної напруги, за якої вихідний код змінюється на одиницю молодшого розряду;
- час перетворення $t_{пр}$ – інтервал від моменту перетворення до появи на виході сталого коду.

У сучасних АЦП застосовуються такі методи перетворення: послідовної лічби; порозрядного кодування; паралельної дії; комбіновані паралельно-послідовні.

ЦАП призначені для перетворення цифрової інформації в аналогову у вигляді напруги(струму). Структура ЦАП вміщує: резистивну або транзисторну матрицю для формування еталонних струмів; ключі для комутації еталонних струмів згідно з вхідним кодом до спільної точки підсумовування; операційний підсилювач для перетворення струму у вихідну напругу; допоміжні схеми для узгодження з вхідними рівнями сигналів; стабільне джерело опорної напруги.

Основними параметрами ЦАП є:

- число розрядів вхідного цифрового коду;
- похибки перетворення;
- діапазон зміни вихідної напруги;
- час установлення, який визначає швидкодію.

АЦП і ЦАП реалізуються у вигляді гібридних або напівпровідникових НВІС.

Комп'ютерна система ЦОС складається із процесорного ядра та підсистем зберігання і введення-виведення. Структура процесорного ядра комп'ютерних систем обробки сигналів повинна забезпечувати реалізацію операційного базису і враховувати особливості алгоритмів ЦОС. До складу процесорного ядра комп'ютерних систем обробки сигналів можуть входити як програмовані процесори обробки сигналів, так і алгоритмічні процесори, архітектура яких відображає структуру алгоритму розв'язання окремих підзадач. Для зменшення часу та вартості розробки апаратних засобів комп'ютерних систем обробки сигналів їх архітектура повинна бути модульною, регулярною та орієнтованою на НВІС-реалізації [9].

Алгоритмічне забезпечення комп'ютерних систем обробки сигналів охоплює методи та алгоритми обробки сигналів і зображень. Зокрема, це методи і алгоритми кореляції, цифрової фільтрації, спектрального аналізу сигналів на базі дискретних ортогональних тригонометричних перетворень та нейромережевих технологій.

На сьогодні розроблено досить потужний математичний апарат технологій цифрової обробки сигналів і зображень, який постійно вдосконалюється, розширюється і орієнтується на апаратні НВІС-реалізації.

Програмне забезпечення комп'ютерних систем обробки сигналів ділиться на системне і спеціальне та охоплює набори програм і даних. До системного програмного забезпечення відносяться програми управління ресурсами та процесами розв'язання задач. Спеціальне програмне забезпечення реалізує базові алгоритми і задачі обробки сигналів та орієнтовано на роботу у реальному часі.

Апаратні та програмні засоби, які забезпечують розробку, налагодження та перевірку функціонування на робочих частотах як апаратури, так і програмного забезпечення комп'ютерних систем обробки сигналів, називаються технологічними. Вартість і час розробки комп'ютерних систем обробки сигналів у значній мірі визначається технологічними засобами.

Основними галузями використання технологій ЦОС є радіолокація (контроль повітряного простору, синтез радіозображень аеродинамічних об'єктів в польоті, класифікація об'єктів за їх радіозображенням), гідроакустика (визначення шумових паспортів кораблів, підводна сейсмологія), зв'язок (підвищення надійності, пропускну здатності, скритності, підвищення завадостійкості), геофізика (пошук нафтоносних (водоносних) шарів), біомедицина (візуалізація органів, діагностика, покращення якості і розпізнавання зображень), контроль оточуючого середовища (льодова обстановка, стан атмосфери і рослинності, метеорологія), радіоастрономія (дослідження планет і випромінювання сонця) та ядерна фізика (покращення якості і розпізнавання зображень).

Література : [1], [3], [6], [12].

2. Цифрова фільтрація

2.1 Нерекурсивний і рекурсивний цифрові фільтри

2.2 Медіанна фільтрація

Цифрова фільтрація – процедура, що використовується для приглушення завад зі спектром, що не перетинається зі спектром сигналу. Фільтрація є загальним випадком лінійного перетворення [1]. Вона дозволяє виділити в чистому вигляді функцію, що лежить в основі спостережуваного явища.

Цифрова система, яка використовується для фільтрації цифрових сигналів називається цифровим фільтром. Такі фільтри можуть реалізовуватися програмним або апаратним шляхом. Цифрові фільтри в порівнянні з аналоговими мають такі переваги:

- висока точність, передатна функція не залежить від дрейфу характеристик елементів;
- гнучкість налаштування та легкість зміни;
- компактність – аналоговий фільтр на дуже низьку частоту вимагає конденсаторів великої ємності або індуктивності.

Недоліки цифрових фільтрів у порівнянні з аналоговими такі:

- складність роботи з високочастотними сигналами;
- трудність забезпечення роботи у реальному часі.

2.1 Нерекурсивний і рекурсивний цифрові фільтри

Основними видами цифрових фільтрів є: нерекурсивний і рекурсивний фільтри.

Нерекурсивний цифровий фільтр використовує лише пряму передачу і задається таким математичним виразом

$$Y_n = \sum_{i=0}^{P-1} a_i X_{n-i}$$

де X_n - вихідні відліки; P – довжина вхідної послідовності, a_i – коефіцієнти, які визначають амплітудно-частотні характеристики фільтра.

Рекурсивний цифровий фільтр використовує як пряму, так і зворотну передачу даних і задається таким математичним виразом

$$Y_n = \sum_{i=0}^{P-1} a_i X_{n-i} + \sum_{j=1}^M b_j Y_{n-j}$$

де a_j і b_j - коефіцієнти, які визначають амплітудно-частотні характеристики фільтра; P і M - довжини послідовностей, які приймають участь як в прямій, так і зворотній передачі даних. Зворотній зв'язок, закладений у виразі обумовлює назву "рекурсивний" цифровий фільтр.

Структури некурсивних і рекурсивних фільтрів наведені на рис.2.1, де Z^{-1} - елемент затримки, який відповідає часовій затримці, рівній періоду дискретизації, Σ - суматор, $\times$ - пристрій множення.

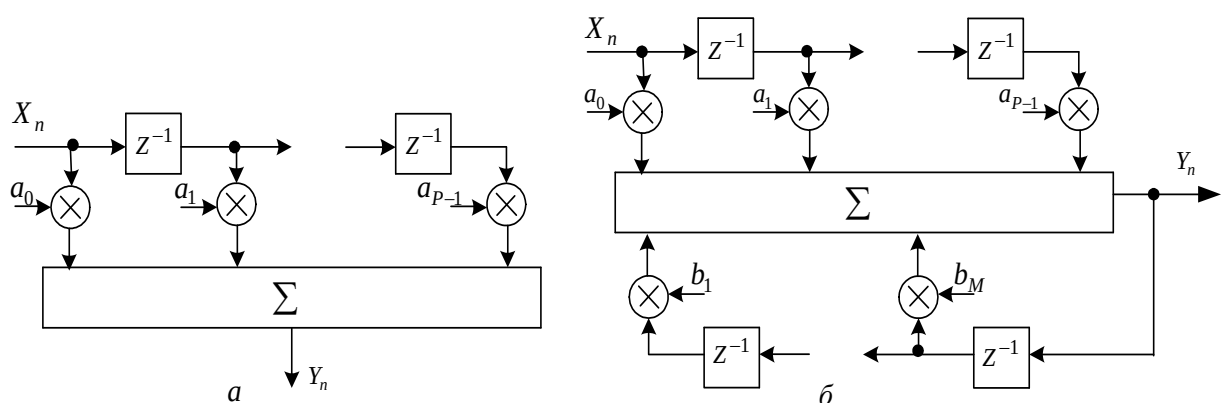


Рис.2.1 Структури цифрових фільтрів: а) некурсивний; б) рекурсивний

Можлива інша класифікація цифрових фільтрів: з скінченною (некурсивний) і нескінченною (рекурсивний) імпульсною

характеристикою. Для збільшення точності та швидкості цифрової фільтрації необхідно використовувати спеціалізовані НВІС або потужний мікропроцесор, які доповнюються швидкісними та багаторозрядними АЦП і ЦАП.

2.2 Медіанна фільтрація

Медіанний фільтр — один із видів цифрових фільтрів, які широко використовуються при ЦОС і зображень для приглушення нерегулярних завад. Медіанна фільтрація є нелінійним способом обробки одномірних і двомірних послідовностей вибірок [9]. В порівнянні з лінійною медіанна фільтрація має важливі переваги: зберігає різкі перепади сигналу; добре згладжує імпульсний шум. Алгоритми медіанної фільтрації базуються на повному або частковому сортуванні чисел-елементів зображень у «вікні» розміром $N=2m+1$ та виділенні у просортованій послідовності центрального елемента, тобто елемента з номером $m+1$.

3. Дискретне перетворення Фур'є

Реальні сигнали є комбінацією багатьох частот, які відображають частотний спектр сигналу. Сигнали представляється у вигляді суми гармонічних функцій або комплексних експонент з частотами, які утворюють арифметичну прогресію. Ряд Фур'є в синусно - косинусні формі має такий вигляд.

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} [a_k \cos(kw_1 t) + b_k \sin(kw_1 t)]$$

де $w_1 = 2\pi / T$ - кругова частота, яка відповідає періоду повторення сигналу.

Частоти kw_1 , які входять у формулу, називаються гармоніками. Вони нумеруються в відповідності з індексом k , а частота $w_k = kw_1$ називається k -й гармонікою сигналу. Коли $x(t)$ є парною функцією, то всі b_k рівні нулю і в формулі ряду Фур'є присутні тільки косинусні доданки. У випадку коли $x(t)$ є непарною функцією, то всі a_k рівні нулю і в формулі ряду Фур'є присутні тільки синусні доданки. Дійсна форма запису ряду Фур'є має такий вигляд

$$x(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty} A_k \cos(kw_1 t + \varphi_k)$$

Комплексна форма запису ряду Фур'є записується так

$$x(t) = \sum_{k=1}^{\infty} C_k e^{-jkw_1 t}$$

де $e^{-jkw_1 t} = \cos \frac{2\pi kt}{T} + j \sin \frac{2\pi kt}{T}$, C_k - комплексні коефіцієнти ряду.

Які зв'язані з амплітудами і фазами таким чином

$$C_k = \frac{a_k}{2} - j \frac{b_k}{2}, \quad \varphi_k = \arctg \frac{b_k}{a_k}$$

Сукупність амплітуд C_k гармонік ряду Фур'є називають амплітудним спектром, сукупність їх фаз φ_k – фазовий спектр.

Для отримання дійсної і уявної частин комплексної огибаючої сигналу потрібно вхідний сигнал помножити на $\cos w_0 t$ і $\sin w_0 t$, де w_0 - несуча частота, та пропустити через фільтри нижніх частот. Для отримання дійсної та уявної частин огибаючої сигналу використовується квадратурна дискретизація, структура якої наведена на рис 3.1, де ФНЧ - фільтр низьких частот, АЦП – аналого-цифровий перетворювач.

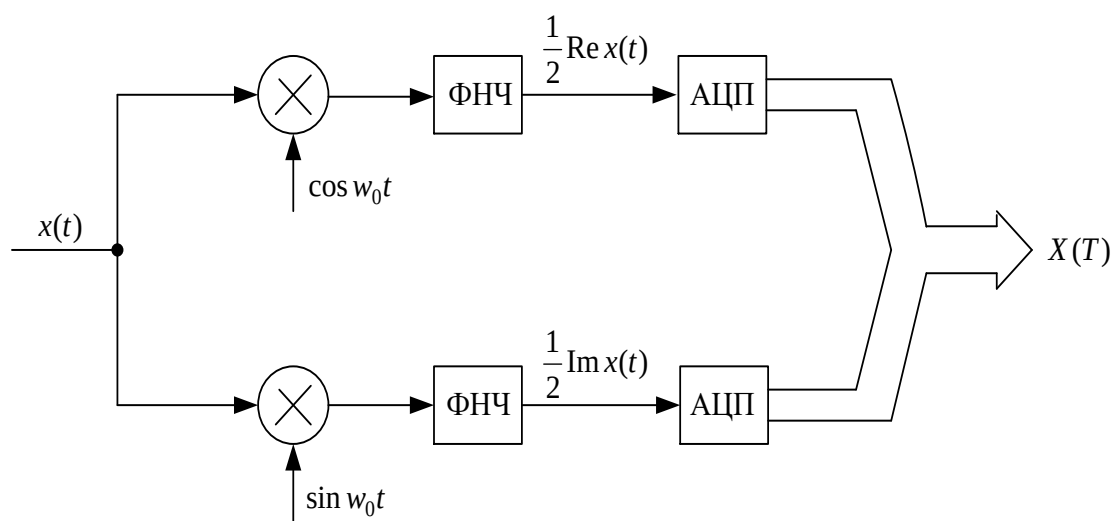


Рис. 3.1 Квадратурна дискретизація

Дискретне перетворення Фур'є (ДПФ) в загальному випадку комплексної послідовності $x(n)$, $n=0,1,\dots,N-1$ визначається виразом

$$X(k) = \sum_{n=0}^{N-1} x(n) W_N^{kn}, \quad (3.1)$$

де $k=0,1,\dots,N-1$, $W_N^{kn} = \cos \frac{2\pi kn}{N} + j \sin \frac{2\pi kn}{N}$ - повертаючий множник.

Послідовність $x(n)$, як правило, визначає N послідовних відліків $x(nT)$ неперервного сигналу $x(t)$, а послідовність $X(k)$ представляє його в частотній області $X(kf)$. ДПФ є лінійним перетворенням, перетворюючий вектор часових відліків в вектор такої самої довжини, що містить спектральні відліки.

Таке перетворення реалізується як множення квадратної матриці (повертаючи множників) перетворення на вхідний вектор-стовпець

$$X = \begin{bmatrix} 1 & 1 & 1 & 1 & \dots & 1 \\ 1 & e^{-j\frac{2\pi}{N}} & e^{-j\frac{4\pi}{N}} & e^{-j\frac{6\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}(N-1)} \\ 1 & e^{-j\frac{4\pi}{N}} & e^{-j\frac{8\pi}{N}} & e^{-j\frac{12\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}2(N-1)} \\ 1 & e^{-j\frac{6\pi}{N}} & e^{-j\frac{12\pi}{N}} & e^{-j\frac{18\pi}{N}} & \dots & e^{-j\frac{2\pi}{N}3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & e^{-j\frac{2\pi}{N}(N-1)} & e^{-j\frac{2\pi}{N}2(N-1)} & e^{-j\frac{2\pi}{N}3(N-1)} & \dots & e^{-j\frac{2\pi}{N}(N-1)^2} \end{bmatrix} \times \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \\ \vdots \\ x(N-1) \end{bmatrix}$$

Обчислення ДПФ шляхом множення матриці на вектор повністю відповідає формулі (3.1).

Результат ДПФ є спектр дискретного періодичного сигналу, який легко можна використати для відновлення неперервного періодичного сигналу. Для даного перетворення використовується обернене ДПФ, яке має вигляд

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) W_N^{-kn}$$

Безпосереднє обчислення ДПФ за формулою (3.1) потребує N^2 множень і $N(N-1)$ додавань комплексних чисел.

Література : [1], [2], [3], [5], [6], [9].

4. Швидкі алгоритми ортогональних тригонометричних перетворень

4.1 Алгоритми швидкого перетворення Фур'є комплексної послідовності.

4.2 Швидкі алгоритми дискретного перетворення Хартлі.

4.3 Швидкі алгоритми дискретних косинус і синус-перетворень Фур'є.

4.4 Алгоритми швидкого косинусного та синусного перетворень.

4.5 Ортогональні швидкі хвильові перетворення.

4.6 Швидкі алгоритми двовимірних ортогональних тригонометричних перетворень.

4.1 Алгоритми швидкого перетворення Фур'є комплексної послідовності

Основна ідея алгоритмів швидкого перетворення Фур'є комплексної послідовності (ШПФк) полягає в збалансованому рекурсивному використанні методики зведення однієї задачі більшої розмірності до задач меншої розмірності. В найпростішому випадку, коли $N=2^m$, де $m=1,2,\dots$, N -точкове дискретне перетворення Фур'є (ДПФ) зводиться до двох $N/2$ -точкових, кожне з яких в свою чергу замінюється двома $N/4$ -точковими і т.д. до одержання двоточкових ДПФ. На даний час розроблені різні методи побудови алгоритмів ШПФ [4]. Нижче розглянемо найбільш поширені при апаратній реалізації.

Алгоритми ШПФ комплексної послідовності за основою два. Існують два варіанти формул переходу до двох ДПФ меншої розмірності, які отримали назву формул розкладу (ФР) алгоритму ШПФ. В першому з них, що отримав назву алгоритму ШПФк за основою два з часовим прорідженням (ШПФк2t), ФР задається виразом.

$$X(k)=X_1(k)+X_2(k), \quad k=0,1,\dots,N-1,$$

$$\text{де } X_1(k)=\text{ДПФ}_{N/2}\{x(n)\}, \quad X_2(k)=\text{ДПФ}_{N/2}\{x(2n+1)\}.$$

В другому варіанті, що отримав назву частотного (ШПФк2f), N -точкове ДПФ замінюється двома $N/2$ точковими ДПФ наступним чином[4]:

$$X(2k)=\sum_{n=0}^{N/2-1} x_1(n)W_{N/2}^{nk}; \quad X(2k+1)=\sum_{n=0}^{N/2-1} x_2(n)W_{N/2}^{nk},$$

$$\text{де } x_1(n)=x(n)+x(n+N/2), \quad x_2(n)=[x(n)-x(n+N/2)].$$

Обидві форми алгоритмів еквівалентні з точки зору кількості обчислень, вони відзначаються простотою структури.

Обчислення за даними алгоритмами вимагає виконання $m = \log_2 N$ етапів, кожен з яких має $N/2$ базових операцій (БО). Графи БО алгоритмів ШПФк2t і ШПФк2f наведені відповідно на рис.4.1а і 4.1б.

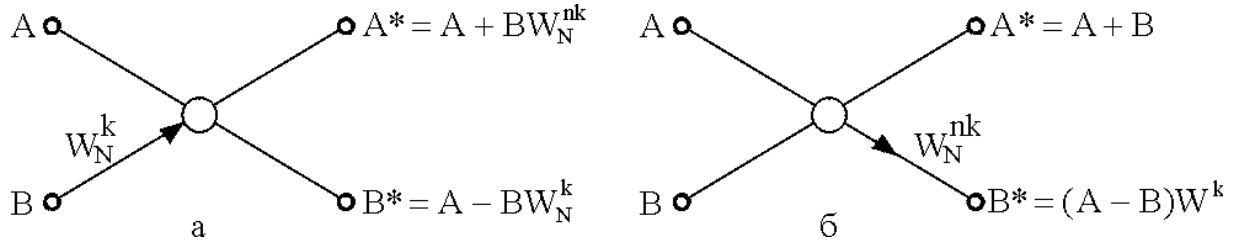


Рис.4.1 Базові операції алгоритмів ШПФк за основою два: а) з часовим прорідженням; б) з частотним прорідженням

Зауважимо, що в загальному випадку БО складаються з однієї операції множення і двох операцій додавання комплексних чисел, тобто вимагається чотири множення і шість додавань дійсних чисел.

Алгоритми ШПФ комплексної послідовності за основою чотири. З точки зору кількості необхідних операцій ефективнішими є алгоритми ШПФк за основою чотири (ШПФк4), коли N - точкове ДПФ одразу за один етап розбивається на чотири $N/4$ - точкові.

Нехай $N=4^m$, $m=1,2,\dots$, тоді загальна ФР алгоритму ШПФк4 з часовим прорідженням (ШПФк4t) задається виразом

$$X(k) = \sum_{p=0}^3 W_N^{pk} \sum_{n=0}^{N/4-1} x(4n+p) W_{N/4}^{nk} \quad (4.1)$$

На основі ФР (4.1) будемо обчислювальну процедуру

$$\left. \begin{aligned} a_0 &= X_0(k); a_p = W_N^{pk} X_p(k), p = 1,2,3; \\ X(k) &= (a_0 + a_2) + (a_1 + a_3); X(k + N/4) = (a_0 - a_2) - j(a_1 - a_3); \\ X(k + N/2) &= (a_0 + a_2) - (a_1 + a_3); X(k + 3N/4) = (a_0 - a_2) + j(a_1 - a_3); \end{aligned} \right\} \quad (4.2)$$

де $k=0,1,\dots,N/4-1$; $X_p(k) = \sum_{n=0}^{N/4-1} x(4n+p) W_{N/4}^{nk}$, $p=0,1,2,3$.

Рекурсивно продовжуючи за формулою (4.1) і процедурою (4.2) розбиття меншої розмірності до чотири-точкових, синтезуємо алгоритм ШПФк4t. Граф БО алгоритму ШПФк4t показаний на рис.4.2.

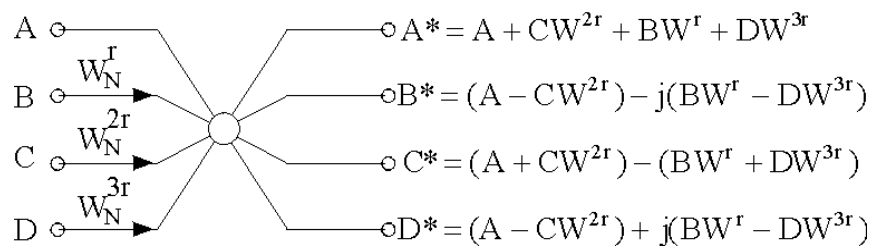


Рис.4.2 Базова операція алгоритму ШПФк4t

Як і в алгоритмі за основою два, для одержання алгоритму ШПФк4 з частотним прорідженням (ШПФк4f) достатньо розглянути граф ШПФк4t у зворотному напрямку. Граф БО алгоритму ШПФк4f показаний на рис.4.3.

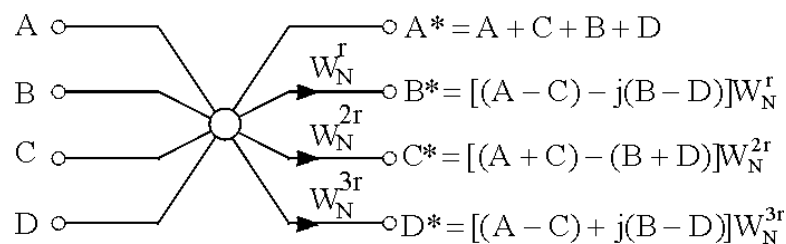


Рис.4.3 Базова операція алгоритму ШПФк4f

Алгоритми ШПФк4 дозволяють на 25% скоротити обчислювальні затрати порівняно з алгоритмами ШПФк2 [4]. Подібним чином будуються алгоритми за основою 8, 16, а також алгоритми за змішаною основою 2-4, 2-4-8 і т.д. У загальному випадку збільшення основи алгоритму веде до зменшення обчислювальних витрат, але при цьому ускладнюється реалізація алгоритму.

Алгоритми ШПФ комплексної послідовності за розщепленою основою два-чотири. Важливим етапом в теорії швидких алгоритмів є розробка алгоритму ШПФк за розщепленою основою два-чотири (ШПФк2-4). В них ФР одержується комбінацією ФР алгоритмів ШПФк2 та ШПФк4 [4]. Так для алгоритму ШПФк2-4 з часовим прорідженням (ШПФк2-4t), ФР має вигляд

$$X(k) = X_0(k) + X_1(k)W_N^k + X_3(k)W_N^{3k}, \quad (4.3)$$

де $X(k) = \text{ДПФ}_N \{x(n)\}$; $X_p(k) = \text{ДПФ}_{N/4} \{x(4n+p)\}$, $p=1,3$; $X_0(k) = \text{ДПФ}_{N/2} \{x(2n+1)\}$.

За формулою (4.3) N - точкове ДПФ розбивається на одне $N/2$ - точкове і два $N/4$ - точкові перетворення.

На основі (4.3) використовуючи періодичність фазових множників W_N^r та перетворень $X_r(k)$ будується обчислювальна процедура

$$\left. \begin{aligned} a &= X_1(k)W_N^k; b = X_3(k)W_N^{3k}; a_1 = a + b; b_1 = -j(a - b) \\ X(k) &= X_0(k) + a_1; X(k + N/2) = X_0(k) - a_1 \\ X(k + N/4) &= X_0(k + N/4) + b_1; X(k + 3N/4) = X_0(k + N/4) - b_1 \end{aligned} \right\} \quad (4.4)$$

Рекурсивно використовуючи ФР (4.3) та процедуру (4.4) до перетворень меншої розмірності, синтезуємо алгоритм ШПФк2-4t. Граф БО алгоритму ШПФк2-4t показаний на рис.4.4.

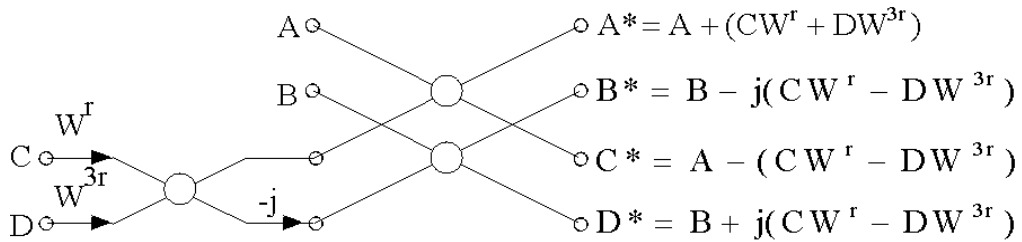


Рис.4.4 Базова операція алгоритму ШПФк2-4t

Для отримання алгоритму ШПФк2-4 з частотним прорідженням (ШПФк2-4f) використовують ФР алгоритмів ШПФк2f і ШПФк4f. ФР для алгоритму ШПФк2-4f має вигляд

$$\begin{aligned} X(2k) &= \text{ДПФ}_{N/2} \{x_0(n)\} \\ X(4k + p) &= \text{ДПФ}_{N/2} \{x_p(n)\}; p = 1, 3 \end{aligned} \quad (4.5)$$

де $x_0(n) = x(n) + X(n + N/2)$; $a_n = x(n) - x(n + N/2)$, $n = 0, 1, \dots, N/2 - 1$; $x_1(n) = (a_n - j a_{n+N/4})$,

$x_3(n) = (a_n + j a_{n+N/4}) W_N^{3n}$, $n = 0, 1, \dots, N/4 - 1$.

Процедура переходу до перетворень меншої розмірності наступна:

$$\begin{aligned} x_0(n) &= x(n) + x(n + N); x_0(n + N/4) = x(n + N/4) + x(n + 3N/4); \\ a &= x(n) - x(n + N/2); b = -j[x(n + N/4) - x(n + 3N/4)]; \\ x_1(n) &= (a + b)W_N^n; x_3(n) = (a - b)W_N^{3n}, n = 0, \dots, N/4 - 1. \end{aligned} \quad (4.6)$$

Шляхом рекурсивного продовження на основі ФР (4.5) та процедури (4.6) розбиття перетворень меншої розмірності синтезується алгоритм ШПФк2-4f. Граф БО такого алгоритму наведений на рис.4.5.

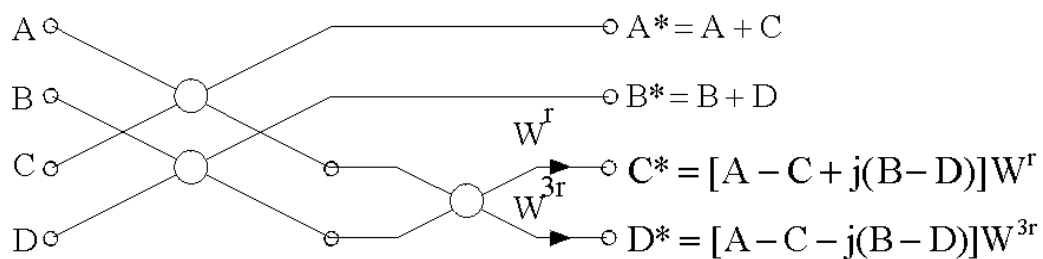


Рис.4.5 Базова операція алгоритму ШПФк2-4f

Алгоритми ШПФк2-4 вдало поєднують простоту структури алгоритмів за основою два з ефективністю алгоритмів з високою основою. Зберігаючи в основному структуру алгоритмів ШПФк2, вони мають найменші обчислювальні затрати серед розглянутого класу алгоритмів.

Алгоритми ШПФк Рейдера-Бреннера. Суттєвий недолік вищерозглянутих алгоритмів ШПФк полягає у використанні комплексних фазових множників W_N^r . Алгоритми ШПФк Рейдера-Бреннера (ШПФкРБ) позбавлені цього недоліку, вони використовують фазові множники, що мають структуру дійсних чисел [4]. ФР для алгоритмів ШПФкРБ з косинусними (ШПФкРБс) фазовими множниками записуються наступним чином

$$X(k) = X_0(k) + 2 \cos(2\pi k / N) D(k), k = 0, 1, \dots, N-1, k \neq N/4, N/3,$$

$$X(N/4) = X_0(N/4) + N/2Q; X(3N/4) = X_0(N/4) - N/2Q,$$

де $X(n) = \text{ДПФ}_N \{x(2n)\}$, $D(k) = \text{ДПФ}_N \{d(n)\}$, елементи $d(n)$ число Q відповідно обчислюються за виразами:

$$d(0)=0$$

$$d(n+1)=x(2n+1)-d(n)-(-1)^n Q, n=1,2,\dots,N/2-1$$

$$Q = 2/N \sum_{n=0}^{N/2-1} (-1)^n x(2n+1)$$

За аналогічною схемою будуються алгоритми ШПФкРБс з виключно уявними фазовими множниками:

$$X(k) = X_0(k) + 2j \sin(2\pi k / N) B(k), k = 0, 1, \dots, N-1, k \neq N/2,$$

$$X(0) = X_0(0) + N/2Q; X(N/2) = X_0(0) - N/2Q,$$

$$Q = \sum_{n=0}^{N/2-1} x(2n+1), X_0(0) = ДПФ_{N/2}\{x(n)\};$$

де

$$B(k) = ДПФ_{N/2}\{b(n)\}, b(0) = 0, b(n+1) = b(n) - x(2n+1) + Q.$$

Необхідно зауважити, що число Q вибирається з умови забезпечення періодичності послідовностей $d(n)$ і $b(n)$.

4.2 Швидкі алгоритми дискретного перетворення Хартлі

Дискретне перетворення Хартлі. Пара взаємно-однозначних перетворень:

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

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

де $H(k)$ і $x(n)$ дійсні послідовності, $C_N^r = \cos(2\pi r / N)$, $S_N^r = \sin(2\pi r / N)$, називається відповідно прямим та зворотнім дискретним перетворенням Хартлі [4,9].

Алгоритми швидкого перетворення Хартлі за основою два.

Для алгоритму швидкого перетворення Хартлі (ШПХ) за основою два з часовим прорідженням (ШПХ2t) ФР має вигляд:

$$H(k) = H_1(k) + H_2(k)C_N^k + H_2(N-k)S_N^k, k = 0, 1, \dots, N-1,$$

де $H(k) = \text{ДПФ}_N\{x(n)\}$ і $H_p(k) = \text{ДПФ}_{N/2}\{x(2n+p-1)\}$, $p=1, 2$.

Процедура обчислення ШПХ2t задається виразами:

$$H(k) = H_1(k) + a; H(k+N/2) = H_1(k) - a; \quad (4.7)$$

$$H(N/2-k) = H_1(N/2-k) + b; H(N-k) = H_1(N/2-k) - b; \quad (4.8)$$

$$a = H_2(k)C_N^k + H_2(N/2-k)S_N^k; b = H_2(k)S_N^k - H_2(N/2-k)C_N^k, \quad (4.9)$$

де $k=0, 1, \dots, N/4-1$. ШПХ2t реалізується на основі простих і складних БО.

Граф простої БО, що реалізує вираз (4.7) чи вираз (4.8), наведений на рис.1.6а, а граф складної БО, що реалізує вираз (4.9), - на рис.4.6б

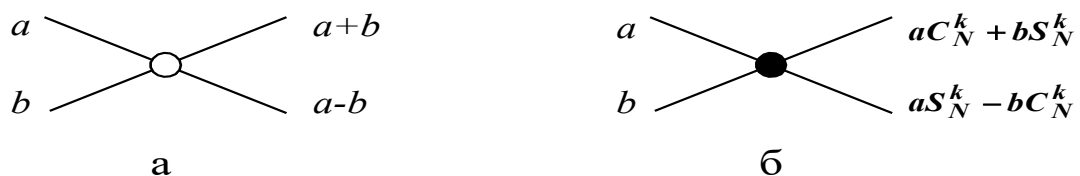


Рис.4.6. Базові операції алгоритму ШПХ2t

Граф алгоритму ШПХ2 з частотним прорідженням (ШПХ2f) є інваріантним графу алгоритму ШПХ2t. Обчислення ШПХ2f здійснюється справа наліво за графом алгоритму ШПХ2t.

Алгоритми швидкого перетворення Хартлі за основою чотири. Для алгоритму ШПХ за основою чотири з часовим прорідженням (ШПХ4t) ФР має вигляд.

$$H(k) = H_0(k) + \sum_{l=0}^3 (C_N^{pk} H_p(k) + S_N^{pk} H_p(N/4-k)),$$

де $H_p(k) = \text{ДПХ}_{N/4}\{x(n)\}$; $p=0, 1, 2, 3$; $n=0, 1, \dots, N/4-1$; $x_p(4n) = x(4n+p)$.

Процедура обчислення ШПХ4t задається виразами

$$\left. \begin{aligned} c_{02} &= H_0(0) + H_2(0); c_{20} = H_0(0) - H_2(0); \\ c_{13} &= H_1(0) + H_3(0); c_{31} = H_1(0) - H_3(0); \end{aligned} \right\} \quad (4.10)$$

$$\left. \begin{aligned} d_{02} &= H_0(N/8) + H_2(N/8); d_{20} = H_0(N/8) - H_2(N/8); \\ d_1 &= \sqrt{2}H_1(N/8); d_3 = \sqrt{2}H_3(N/8); H(0) = c_{02} + c_{13}; \\ H(N/8) &= d_{02} + d_1; H(N/4) = c_{20} + c_{31}; H(3N/8) = d_{20} + d_3; \\ H(N/2) &= c_{02} - c_{13}; H(5N/8) = d_{02} - d_1; \\ H(3N/4) &= c_{20} - c_{31}; H(7N/8) = d_{20} - d_3; \end{aligned} \right\} \quad (4.11)$$

$$\left. \begin{aligned} a_0 &= H_0(k), a_p = C_N^{pk} H_p(k) + S_N^{pk} H_p(N/4 - k), b_0 = H_0(N/4 - k), \\ b_p &= S_N^{pk} H_p(k) - C_N^{pk} H_p(N/4 - k), p = 1, 2, 3; \end{aligned} \right\} \quad (4.12)$$

$$\left. \begin{aligned} a_{02} &= a_0 + a_2; a_{20} = a_0 - a_2; a_{13} = a_1 + a_3; a_{31} = a_1 - a_3; \\ b_{02} &= b_0 + b_2; b_{20} = b_0 - b_2; b_{13} = b_1 + b_3; b_{31} = b_1 - b_3; \end{aligned} \right\} \quad (4.13)$$

$$\left. \begin{aligned} H(k) &= a_{02} + a_{13}; H(k + N/4) = a_{20} - b_{31}; \\ H(k + N/2) &= a_{02} + a_{13}; H(k + 3N/4) = a_{20} - b_{31}; \\ H(N/4 - k) &= b_{02} + a_{13}; H(N/2 - k) = b_{20} + b_{31}; \\ H(3N/4 - k) &= b_{02} - a_{13}; H(N - k) = b_{20} - b_{31}; k = 1, 2, \dots, N/8 - 1. \end{aligned} \right\} \quad (4.14)$$

На рис.4.7а і рис.4.7б наведені графи перетворення, що реалізують обчислення за виразами (4.10), (4.11), (4.12), (4.13), (4.14).

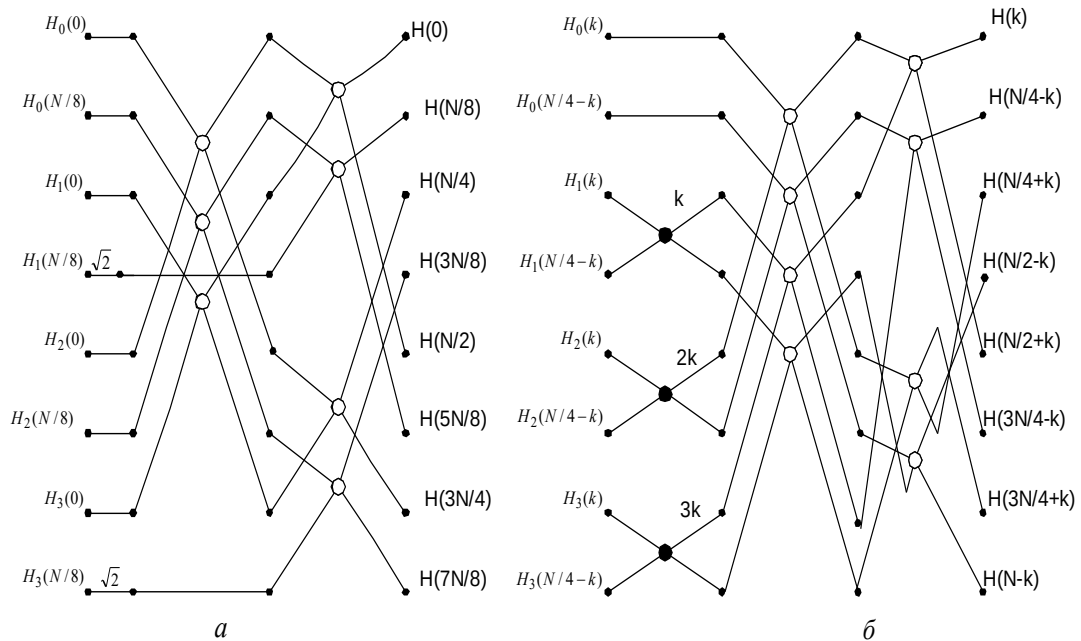


Рис.4.7 Восьмиточкові БО алгоритму ШПХ4t : а – спрощена, б загальна

Граф алгоритму ШПХ4 з частотним прорідженням (ШПХ2f) є інваріантним графу алгоритму ШПХ4t. Обчислення ШПХ4f здійснюється справа наліво за графом алгоритму ШПХ4t.

Алгоритми швидкого перетворення Хартлі за розщепленою основою два-чотири. Важливим етапом в теорії швидких алгоритмів є розробка алгоритму ШПХ за розщепленою основою два-чотири (ШПХ2-4). В них ФР одержується комбінацією ФР алгоритмів ШПХ2 та ШПХ4 [4]. Так для алгоритму ШПХ24 з часовим прорідженням (ШПХ2-4t), ФР має вигляд.

$$H(k) = H_0(k) + \sum_{p=1,3} C_N^{pk} H_p(k) + S_N^{pk} H_p(N/4 - k)$$

де $H_0(k) = \text{ДПХ}_{N/2}\{x(2n)\}$, $H_p(k) = \text{ДПХ}_{N/4}\{x_p(n)\}$, $x_p(n) = x(4n+p)$, $p=1,3$.

При переході до перетворень меншої розмірності використовується процедура.

$$\left. \begin{aligned} a_{13} &= H_1(0) + H_3(0); a_{31} = H_1(0) - H_3(0); d_1 = \sqrt{2}H_1(N/8); \\ d_3 &= \sqrt{2}H_3(N/8); H(0) = H_0(0) + a_{13}; H(N/2) = H_0(0) - a_{13}; \\ H(N/4) &= H_0(N/4) + a_{13}; H(3N/4) = H_0(N/4) - a_{31}; \\ H(N/8) &= H_0(N/8) + d_1; H(5N/8) = H_0(N/8) - d_1; \\ H(3N/8) &= H_0(3N/8) + d_3; H(7N/8) = H_0(3N/8) - d_3; \end{aligned} \right\} \quad (4.15)$$

$$\left. \begin{aligned} a_p &= C_N^{pk} H_p(k) + S_N^{pk} H_p(N/4 - k); \\ b_p &= S_N^{pk} H_p(k) - C_N^{pk} H_p(N/4 - k), p = 1,3 \end{aligned} \right\} \quad (4.16)$$

$$\left. \begin{aligned} a_{13} &= a_1 + a_3; a_{31} = a_1 - a_3; b_{13} = b_1 + b_3; b_{31} = b_3 - b_1; H(k) = \\ &= H_0(k) + a_{13}; H(N/4 - k) = H_0(N/4 - k) + a_{31}; H(k + N/4) = \\ &= H_0(k + N/4) + b_{31}; H(N/2 - k) = H_0(N/2 - k) + b_{13}; \end{aligned} \right\} \quad (4.17)$$

$$\left. \begin{aligned} H(k + N/2) &= H_0(k) - a_{13}; H(3N/4 - k) = H_0(N/4 - k) - a_{31}; \\ H(k + 3N/4) &= H_0(k + N/4) - b_{31}; H(N - k) = H_0(N/2 - k) - b_{13}, \\ k &= 1, 2, \dots, N/8 - 1. \end{aligned} \right\} \quad (4.18)$$

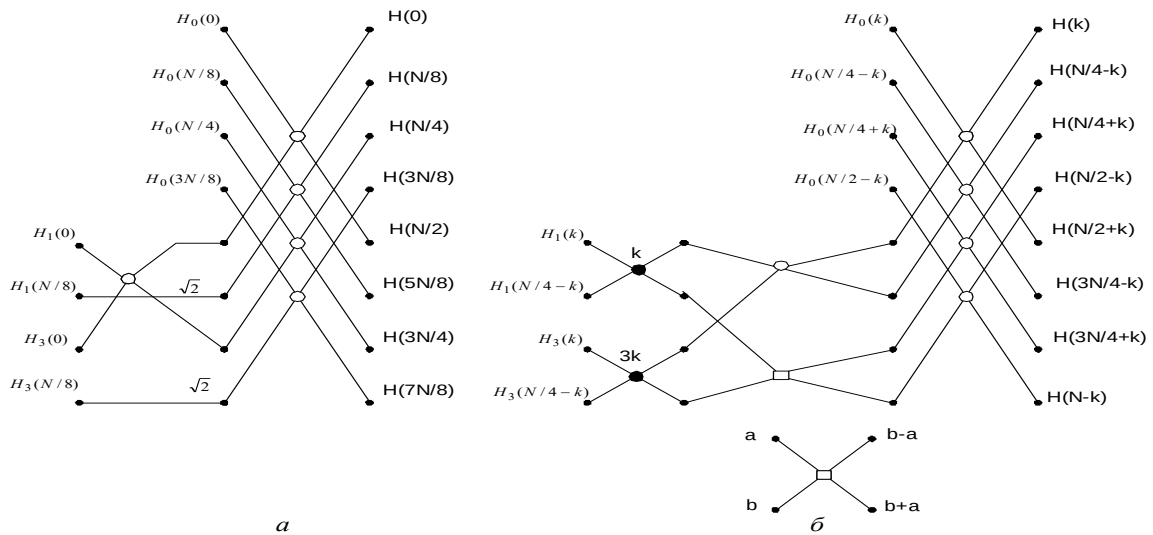


Рис. 4.8 Восьмиточкові БО алгоритму ШПХ2-4t : а – спрощена, б загальна

На рис.4.8а і 1.8б показані восьмиточкові БО алгоритму ШПХ2-4t графа, що реалізують відповідно (4.15) і (4.16), (4.17), (4.18). Алгоритм ШПХ за розщепленою основою два-чотири з частотним прорідженням (ШПХ2-4f) можна отримати з графу алгоритму ШПХ2-4t: шляхом зміни напрямку виконувати перетворення справа-наліво, починаючи з останнього етапу. Обчислювальні витрати на реалізацію алгоритмів ШПХ2-4t і ШПХ2-4f співпадають.

4.3 Швидкі алгоритми дискретних косинус і синус-перетворень Фур'є

Дискретне косинус-перетворення Фур'є (ДКПФ) і дискретне синус-перетворення Фур'є (ДСПФ), а також відповідні швидкі алгоритми (ШКПФ і ШСПФ) забезпечують безнадлишкове виконання дискретного перетворення Фур'є (ДПФ) і Хартлі (ДПХ) комплексної, дійсної, комплексно-спряженої, парної або непарної послідовності [4]. При цьому розпаралелюється обробка дійсної, комплексно-спряженої і комплексної послідовностей.

Нехай $x(n)$, $n=0,1,\dots,N-1, N=2^m$, $m=1,2,3,\dots$ - дійсна послідовність, $x_c(n)$ і $x_s(n)$ – відповідно її парна (4.19) і непарна (4.20) частини

$$x_c(n)=x_c(N-n)=\{x(n)+x(N-n)\}/2, n=0,1,\dots,N/2; \quad (4.19)$$

$$x_s(n)=-x_s(N-n)=\{x(n)-x(N-n)\}/2, n=0,1,\dots,N/2; \quad (4.20)$$

так, що $x(n)=x_c(n)+x_s(n)$, $n=0,1,\dots,N-1$, і нехай $H_C(k)=\text{ДКПФ}_N\{x_c(n)\}$,

$$H_S(k)=\text{ДСПФ}_N\{x_s(n)\}, H_C(k)=\sum_{n=0}^{N-1} x_c(n) C_N^{kn}, H_S(k)=\sum_{n=0}^{N-1} x_s(n) S_N^{kn},$$

де $C_N^r=\cos(2\pi r/N)$, $S_N^r=\sin(2\pi r/N)$, $x_c(n)=\text{ДКПФ}_N^{-1}\{H_C(k)\} = N \text{ДКПФ}\{H_C(k)\}$, $x_s(n) = \text{ДСПФ}_N^{-1}\{H_S(k)\} = \text{ДСПФ}\{H_S(k)\}$.

Серед відомих [4] алгоритмів ШКПФ і ШСПФ простотою реалізації відрізняються алгоритми, побудовані за методом Рейдера-Бреннера (ШКПФРБ і ШСПФРБ). Їх графи загалом зберігають схему зв'язків алгоритмів за основою два, але за кількістю множень відповідають алгоритмам за розщепленою основою і мають дуже просту базову операцію “дійсний метелик”. Тому віддамо їм перевагу при реалізації ШКПФ і ШСПФ на потокових обчислювальних структурах.

Формули розкладу (ФР) алгоритмів ШКПФ і ШСПФ відповідно задаються виразами

$$H_C(2k) = \text{ДКПФ}_{N/2} \{x_c(n) + x_c(N/2 + n)\};$$

$$H_C(1) = A(0), H_C(2k+1) = 2A(k) - H_C(2k-1), k=1,2,\dots,N/2-2; \quad (4.21)$$

$$H_S(2k) = \text{ДСПФ}_{N/2} \{x_s(n) + x_s(N/2 + n)\};$$

$$H_S(1) = B(0), H_S(2k+1) = 2B(k) + H_S(2k-1), k=1,2,\dots,N/2-2, \quad (4.22)$$

де $A(k)=\text{ДКПФ}_{N/2}\{[x_c(n)-x_c(N/2+n)]C_N^n\}$, $B(k)=\text{ДСПФ}_{N/2}\{[x_s(n)-x_s(N/2+n)]S_N^n\}$.

На їх основі N -точкове ДКПФ розбивається на два $N/2$ -точкові ДКПФ, а N -точкове ДСПФ - на $\text{ДСПФ}_{N/2}$ і $\text{ДКПФ}_{N/2}$.

Рекурсивно застосовуючи ФР (4.21) і (4.22) до перетворень меншої розмірності, отримуємо алгоритми ШКПФРБ і ШСПФРБ [4].

Ці алгоритми мають близькі схеми реалізації обчислень, які достатньо просто реалізуються на одному і тому ж пристрої. Так, на рис. 4.9 наведений узагальнений граф алгоритмів ШКПФРБ і ШСПФРБ для $N=16$.

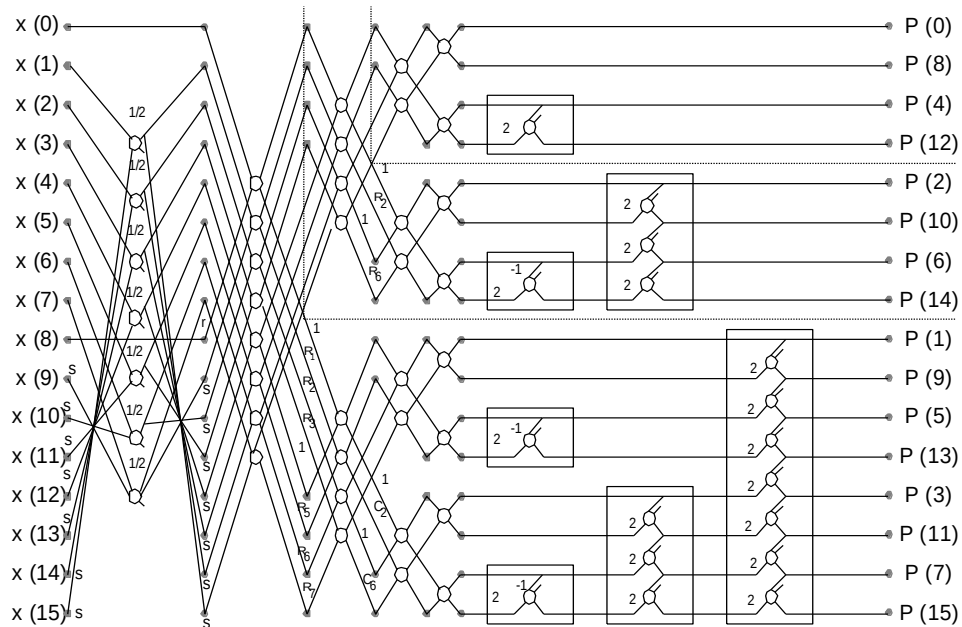


Рис 4.9. Узагальнений граф алгоритму ШКПФРБ і ШСПФРБ для $N=16$

Для першого з них використовуються множники $r=s=1$, $R_n=C_{16}^n$, на виході маємо $P(k)=H_C(k)$, а для другого $r=0$, $S=-1$, $R_n=S_{16}^n$, $P(k)=H_S(k)$.

Порівняльний аналіз приведеного та відомого графа алгоритму комплексного ШПФ Кулі-Тьюкі за основою два (ШПФк2) з частотним прорідженням [4], показує співпадіння їх структури на $m=\log_2 N$ основних етапах переходу до перетворень меншої розмірності. При цьому використовується суттєво спрощена базова операція, що виконується над дійсними даними при дійсних фазових множниках (рис.4.10).

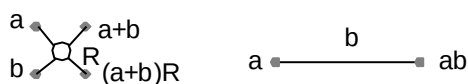


Рис.4.10 Базові операції алгоритмів ШКПФ-ШСПФ

На двох останніх етапах фазові множники приймають тривіальне значення, і тому на графі не показані. Крім того, в алгоритмах ШКПФРБ-ШСПФРБ додатково введений етап виділення парної за (4.19) чи непарної за (4.20) складових вхідної послідовності, а також $m=\log_2 N-1$ етапів додавань, на яких здійснюється перехід від проміжних перетворень $A(k)$ чи $B(k)$ до необхідних $H_C(k)$ і $H_S(k)$.

4.4 Алгоритми швидкого косинусного та синусного перетворень

Пряме та обернене дискретне косинусне перетворення (ДКП) послідовності дійсних чисел $x(n)$, $n=0,1,\dots,N-1$, відповідно описуються виразами

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

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

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

Оскільки множення на P_k (у загальному випадку на степінь двійки) в обчислювальному аспекті не має принципового значення, то його можна опустити.

Пряме та обернене дискретні синусні перетворення (ДСП) відповідно мають вигляд

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

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

Між ДСП і ДКП існує проста математична залежність

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

котра дозволяє звести перше до обчислення другого шляхом тривіального множення на -1 і перестановки елементів послідовності. Тому надалі обмежимося тільки розглядом алгоритмів обчислення ДКП.

На рис.4.11 показані основні етапи графа 16-точкового алгоритму швидкого косинусного перетворення (ШКП) [4].

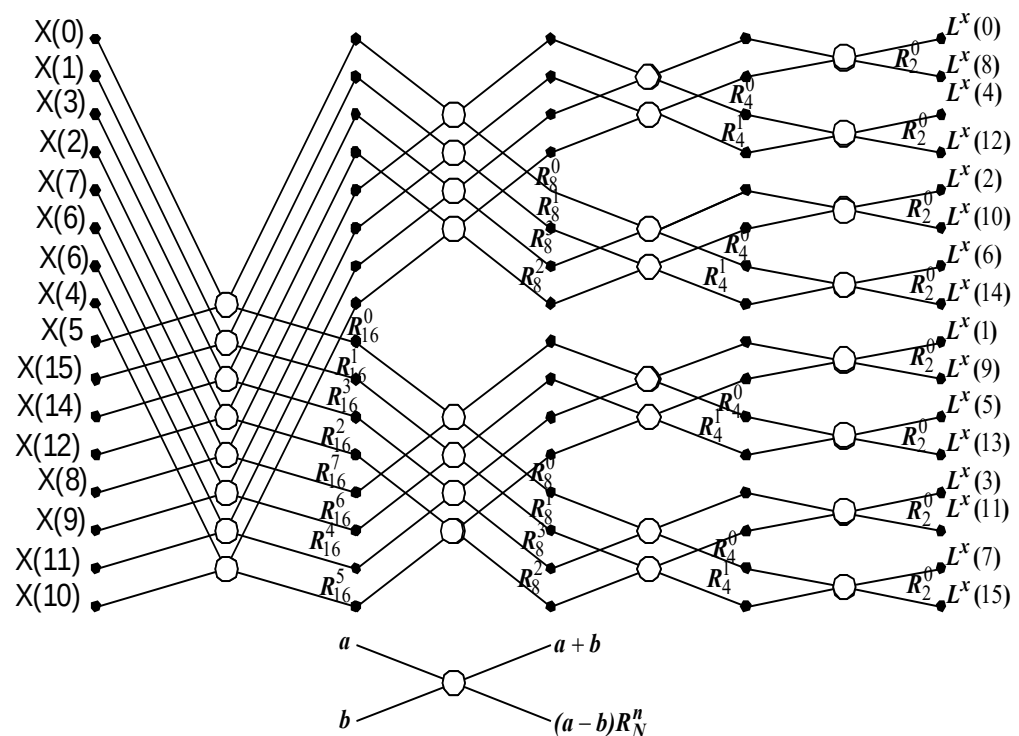


Рис. 4.11 Направлений граф алгоритму прямого ШКП для N=16

Етапи додаткових операцій підсумовування тут не показані. Структура графа алгоритму ШКП повторює структуру графа алгоритму ШПФк за основою 2, але базові операції виконуються над дійсними числами з дійсними фазовими множниками.

4.5 Ортогональні швидкі хвильові перетворення

За останні роки широкого розвитку набули методи, які базуються на дискретних хвильових перетвореннях (ДХП). Ці перетворення також називаються вейвлетними (wavelets) або хвильовими [4,9].

Використання ДХП забезпечує ефективне вирішення багатьох існуючих задач обробки сигналів. Оскільки швидкі алгоритми ДХП (ШХП) потребують виконання тільки $O(N)$ арифметичних операцій, де N – довжина перетворення, то їх доцільно використовувати при обмежених апаратних або обчислювальних ресурсах. Крім того, ДХП добре представляють нестационарні сигнали, а також локально стаціонарні сигнали, тому ДХП особливо часто використовуються в задачах кодування (стиску) даних та задачах фільтрації.

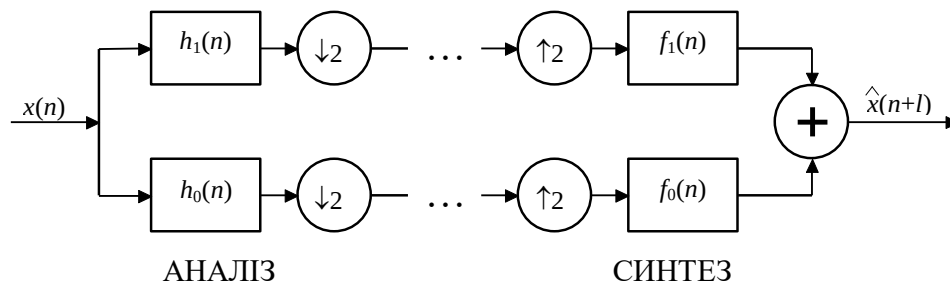


Рис. 4.12 Загальний вигляд одного етапу ДХП

Нехай $x(n)$, $n = 0, 1, \dots, N-1$, вхідний дискретний сигнал. Загальний вигляд одного етапу прямого (аналіз) та оберненого (синтез) дискретного хвильового перетворення (ДХП) графічно показаний на рис.4.12 [].

Тут $h_0(n)$, $f_0(n)$, $h_1(n)$, $f_1(n)$ імпульсні характеристики СІХ-фільтрів, а квадрати, де вони вказані, відповідають операції згортки

$$y(n) = x(n) \bullet h(n) = \sum_{k=0}^{N-1} h(k) x(n-k), n = 0, 1, \dots, N-1. \quad (4.23)$$

Вихідний сигнал $\hat{x}(n)$ запізнюється на l відліків стосовно вхідного сигналу $x(n)$. Оператор $\downarrow 2$ відповідає операції прорідження сигналу – із нього вилучається кожен другий відлік, починаючи з першого, а оператор $\uparrow 2$ відповідає операції розширення сигналу – після кожного відліку вставляється нульовий відлік, за рахунок чого загальна кількість відліків збільшується в два рази.

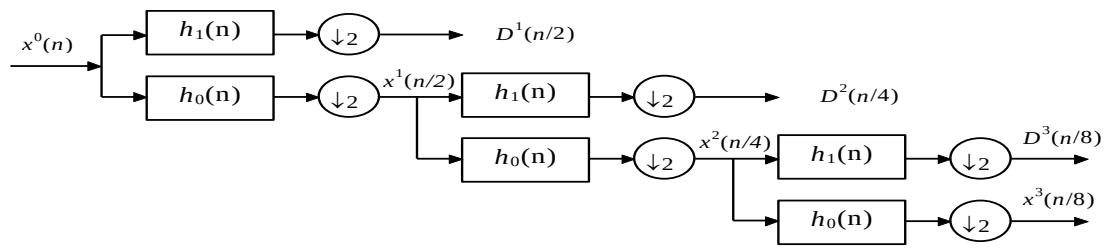


Рис.4.13 Пірамідальна схема продовження

Класичною схемою продовження одноетапного ДХП є пірамідальна (рис.4.13.), що відповідає змішаному частотно–часовому поданню сигналу. Якщо $N = 2^m$, то пірамідальна схема складається з m послідовних одноетапних 2^k –точкових ДХП, де $k = m, m-1, \dots, 1$ при прямому та $k = 1, \dots, m$ при оберненому перетвореннях. При цьому в ортогональних фільтрах з скінченною імпульсною характеристикою (СІХ-фільтри) імпульсні характеристики мають однакову довжину, позначимо її літерою q [4].

Приведемо оцінку обчислювальних затрат такого алгоритму ШХП. Нехай $\mu_N^{(p)}$ і $\alpha_N^{(p)}$ відповідно задають кількість множень та додавань, що використовуються в N –точковому p –етапному прямому ДХП. Обчислення одного відліку згортки (1) при q –точковій імпульсній характеристиці потребує q множень та $q-1$ додавань.. Відомо, що в загальному випадку ортогональних ШХП, обчислювальні затрати добре оцінюються виразами

$$\mu_N^{(m)} \approx 2Nq; \quad \alpha_N^{(m)} \approx 2N(q-1)$$

Тут враховано те, що операція прорідження вилучає з вихідних послідовностей $N/2 + N/2 = N$ відліків, відповідно обчислення яких опускається. Таким чином, на один відлік вихідного сигналу потрібно $2q$ множень і $2(q-1)$ додавань або сумарно $2(2q-1) \approx 4q$ арифметичних операцій.

4.6 Швидкі алгоритми двовимірних ортогональних тригонометричних перетворень

На даний час існують дві методики побудови швидких алгоритмів двовимірних ортогональних тригонометричних перетворень (ОТП): рядково стовчикове перетворення (РСП) та за векторною основою.

Методика РСП полягає в тому, що над вхідними даними поетапно (спочатку над рядками, а потім над стовпцями) виконуються базове одновимірне перетворення з використанням відповідного швидкого алгоритму. Узагальнену формулу розкладу для ОТП для методу РСП можна записати у вигляді

$$X(p, q) = \sum_{n=0}^{N-1} \left[\sum_{m=0}^{M-1} x(n, m) \cdot RW_M^{mq} \right] \cdot RW_N^{np}$$

де RW_R^r - комплексні або дійсні повертаючі множники, що визначаються видом ОТП, $N \times M$ – розмірність перетворення.

Додатковою операцією при використанні РСП є транспонування матриці вхідних даних після кожного з етапів. Методика РСП може використовуватися лише тоді, коли ОТП є такими що діляться.

Швидкі алгоритми двовимірних перетворень за векторною основою будуються на основі властивостей матриць відповідних перетворень. На практиці найчастіше використовуються швидкі алгоритми, що побудовані за методом Кулі-Тюкі, тому що вони мають найпростішу структуру. Між собою алгоритми одного класу різняться основою розбиття та способом прорідження. Спосіб прорідження в основному не впливає на ефективність швидких алгоритмів. Алгоритми, побудовані за високою основою, мають кращі обчислювальні характеристики, але й складнішу формулу розкладу та базову операцію.

Як приклад розглянемо ФР для швидких алгоритмів ДПФк за векторною основою два на два та розщепленою векторною основою два на чотири. Для ДПФк формула розкладу швидкого алгоритму з часовим прорідженням за векторною основою 2×2 (ШПФк $2 \times 2t$) запишеться

$$X(p, q) = X_{00}(p, q) + X_{10}(p, q)W_N^p + X_{01}(p, q)W_M^q + X_{11}(p, q)W_N^p W_M^q, \left. \begin{array}{l} p = 0, 1, \dots, N-1, \\ q = 0, 1, \dots, M-1, \end{array} \right\}$$

$$\text{де } X(p, q) = \text{ДПФ}_{N \times M} \{x(n, m)\}, \quad X_{kk}(p, q) = \text{ДПФ}_{N/2 \times M/2} \{x(2n+k, 2m+k)\}$$

Вона зводить одне $N \times M$ –точкове перетворення до чотирьох $N/2 \times M/2$ –точкових.

Формула розкладу швидкого алгоритму ДПФк за векторною розщепленою векторною основою 2×4 зводить одне $N \times M$ –точкове перетворення до одного $N/2 \times M/2$ –точкового та дванадцяти $N/4 \times M/4$ –точкових і при прорідженні в часі записується

$$X(p, q) = X_{00}(p, q) + X_{10}^{00}(p, q)W_N^p + X_{10}^{10}(p, q)W_N^{3p} + X_{10}^{01}(p, q)W_{N,M}^{p,2q} + X_{10}^{11}(p, q)W_{N,M}^{3p,2q} + \left. \begin{array}{l} + X_{01}^{00}(p, q)W_M^q + X_{01}^{10}(p, q)W_{N,M}^{2p,q} + X_{01}^{01}(p, q)W_M^{3q} + X_{01}^{11}(p, q)W_{N,M}^{2p,3q} + \\ + X_{11}^{00}(p, q)W_{N,M}^{p,q} + X_{11}^{10}(p, q)W_{N,M}^{3p,q} + X_{11}^{01}(p, q)W_{N,M}^{p,3q} + X_{11}^{11}(p, q)W_{N,M}^{3p,3q}, \\ p = 0, 1, \dots, N-1, \quad q = 0, 1, \dots, M-1, \end{array} \right\}$$

$$\text{де } X_{00}(p, q) = \text{ДПФ}_{N/2 \times M/2} \{x(2n, 2m)\}$$

$$X_{kk}^{rr}(p, q) = \text{ДПФ}_{N/4 \times M/4} \{x[2(2n+r)+k, 2(2m+r)+k]\}, \quad X(p, q) = \text{ДПФ}_{N \times M} \{x(n, m)\}$$

Формула розкладу швидких алгоритмів із прорідженням за частотою можна отримати з наведених вище, якщо розглянути їх у зворотному порядку.

Формули розкладу швидких алгоритмів інших ОТП можна отримувати на основі, розглянутих вище, ФР алгоритмів ДПФк, врахувавши особливості відповідних перетворень.

Література : [1], [2], [3], [7], [9], [12].

5. Алгоритми функціонування, програмування і навчання штучних нейронних мереж паралельної обробки сигналів

Штучні нейронні мережі (ШНМ) довільної структури можна представити відповідним графом, кожна вершина якого відповідає окремому нейронному елементу (НЕ).

Всі нейронні елементи об'єднуються дугами - синаптичними зв'язками, що можуть починатися з виходу довільного НЕ, а також від джерела вхідного сигналу. Задаємо систему повних зв'язків так, що вхід кожної вершини графу з'єднаний з виходами всіх вершин та з усіма джерелами вхідних сигналів синаптичними зв'язками з певними заданими вагами, величини яких встановлюються для заданої структури та при навчанні ШНМ. Відсутність певного зв'язку можна задати його нульовою вагою. Позначаючи кількість входів ШНМ через n_x , а число НЕ - n , маємо $n_x + n$ входів для кожного НЕ. Проходження сигналу з входу на вихід вершини графу передбачає обчислення зваженої суми та перетворення останньої у відповідності з перехідною функцією НЕ. В цьому випадку вичерпним описом навченої ШНМ є представлення довільного НЕ (рис.5.1).

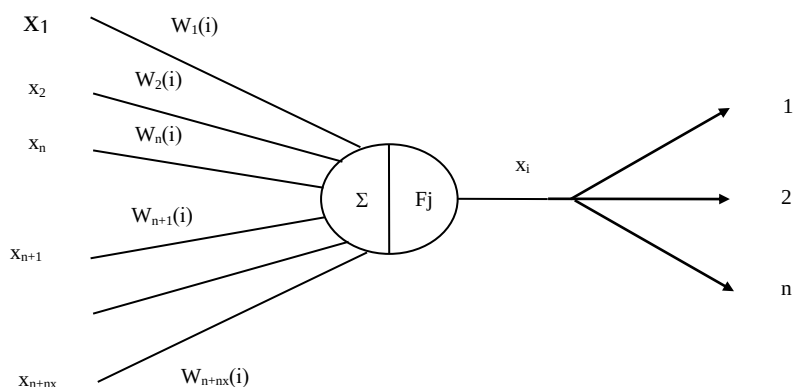


Рис.5.1. Нейронний елемент

де $x_1, \dots, x_n$ - вихідні сигнали НЕ; $x_{n+1}, \dots, x_{n+n_x}$ - вхідні сигнали ШНМ; $w_1(i), \dots, w_{n+n_x}(i)$ - вагові коефіцієнти синаптичних зв'язків i -го НЕ [9].

Вихідний сигнал i -го нейронного елемента визначається через сигнали його входів

$$x_i = F_i \left[\sum_{j=1}^{n+n_x} x_j \cdot W_j(i) \right] \quad (5.1)$$

Так як для кожного НЕ в процесі задання структури та навчання ШНМ встановлюється свій набір синаптичних ваг вважаємо, що вони є функціями від дискретного аргумента i - номера НЕ. Крім того кожному НЕ відповідає певна перехідна функція F_i , що є однотипною для гомогенних ШНМ та індивідуальною для гетерогенних.

Якщо набір векторів-реалізацій, що подаються на входи ШНМ при навчанні, контролі, використанні пронумерувати довільним чином $N=1, 2, \dots, N_m, \dots$, то кожному номеру реалізації N буде відповідати певний сигнал на виході кожного НЕ, який пов'язується з номером реалізації функціонально, тобто

$$x_i = F_i^{(1)}(N) \quad (5.2)$$

де $i=1, \dots, n$; $N=1, 2, \dots, N_m$.

Назвемо функції вказаного типу, які аргументом мають довільно задані номери реалізацій, вихідними на відміну від перехідних функцій F_i , аргументами яких є зважені суми по відповідних входах. Метою налаштування гетерогенних ШНМ є встановлення перехідних функцій НЕ, однак останні можуть бути задані на основі відомих вихідних.

Таким чином, навчена ШНМ повністю задається множиною функцій синаптичних ваг $\{W_j(i)\}$ та множиною вихідних функцій $\{F_i^1(N)\}$ де $j=1, \dots, n+n_x$. При використанні ШНМ на входи останньої подається вектор вхідних сигналів $(x_{n+1}, x_{n+2}, \dots, x_{n+n_x})$, який можна трактувати як функцію від номера входу $\{F_N^2(i)\}$, яка є загалом різною для кожної з реалізацій.

Таким чином, заданим множинам функцій $\{W_j(i)\}$, $\{F_i^1(N)\}$ та функції $\{F_N^2(i)\}$ відповідає число або вектор, що продукується ШНМ

$$Q[\{W_j(I)\}, \{f_i(N)\}, \varphi_N(I)] = x, \quad (5.3)$$

де Q - оператор ШНМ, що є не чим іншим, як функціоналом від відповідних аргументних функцій табличного типу.

В зв'язку з цим пропонується назва розробленої моделі - “Функціонал на множині табличних функцій”(ФТФ). За мету навчання ШНМ ставиться визначення множин аргументних функцій синаптичних ваг, вихідних функцій синаптичних ваг, вихідних функцій НЕ, що забезпечують виконання рівності (1.26) (точне або приблизне) для множини функцій входу $\{F_N^2(i)\}$, $N=1, 2, \dots$

Запропонована модель (5.3) достатньо загального виду, однак вона стала основою для розробки принципово нових парадигм навчання ШНМ, в тому числі і для нових структур типу Feed Forward.

Література : [1], [4], [6], [7], [8], [12].

6. Кореляція

Коли необхідно визначити подібність між сигналами в різні моменти часу або виділити сигнал на фоні шуму, то здійснюють кореляційну обробку, важливе місце в якій займає обчислення функцій взаємної кореляції. Взаємна кореляційна функція двох часових послідовностей X і Y , кожна з яких містить N відліків записується у вигляді

$$R_{xy} = \frac{1}{N-r} \sum_{i=1}^{N-r} X_i Y_{i+r}$$

Згідно цього виразу взаємна кореляційна функція двох сигналів обчислюється з відносною затримкою r одного сигналу по відношенню до другого.

Кореляційна функція записується у вигляді

$$R_{xx} = \frac{1}{N-r} \sum_{i=1}^{N-r} X_i X_{i+r}$$

Обчислення кореляційної і взаємнокореляційної функції двох сигналів складається з трьох основних операцій: часової затримки, множення і підсумовування.

Література : [1], [3], [5], [6], [10], [12].

7. Сортування

Серед всієї сукупності алгоритмів, що реалізують логічну обробку сигналів в системах ЦОС, найчастіше використовуються алгоритми сортування. Задача сортування формулюється наступним чином: для заданої послідовності $\{x(i)\}$ необхідно отримати нову послідовність $\{m(i)\}$, яка складається із елементів $\{x(i)\}$, переставлених в необхідному порядку. В основі реалізації алгоритмів сортування лежать дві операції: порівняння і пересилання даних.

Для організації сортування в програмованих процесорах є в наявності достатньо велике число алгоритмів, кожний з яких має свої переваги і недоліки. Виконання сортування програмним шляхом є часомістким. З розвитком технології НВІС в системах ЦОС для реалізації сортування все більше використовуються спеціалізовані апаратні засоби, які дозволяють суттєво зменшити час виконання даної операції.

Література : [1], [2], [4], [13].

8. Операційний базис комп'ютерних систем обробки сигналів

Аналіз задач, методів і алгоритмів ЦОС і зображень дозволив виділити наступні характерні особливості:

- великий об'єм обчислень з перевагою обчислювальних операцій над логічними;
- регулярність і рекурсивність алгоритмів;
- структура даних дозволяє застосувати векторну обробку з використанням обох видів паралелізму (просторового і часового);
- велику інтенсивність і постійність потоків даних;
- широкий динамічний і частотний діапазон сигналів, що обробляються;
- багатоканальне введення та виведення даних з виконанням функцій переставлення і затримки даних на необхідну кількість тактів;
- можливості розпаралелювання як в часі, так і в просторі;
- розв'язання поряд з прямою, оберненою задачею;
- постійне ускладнення нових алгоритмів і підвищення вимог до точності результатів.

Для визначення операційного базису комп'ютерних систем обробки сигналів і зображень необхідно виділити базові операції алгоритмів ЦОС і зображень [9]. Операції затримки, додавання, віднімання, множення та обчислення сум парних добутоків є БО для алгоритмів цифрової фільтрації, обчислення кореляційної і взаємнокореляційної функцій. Складнішими є БО швидких алгоритмів ортогональних тригонометричних перетворень дійсної та комплексної послідовності, які зводяться до табличного обчислення коефіцієнтів $W_N^{kn}, C_N^{kn}, S_N^{kn}$ та виконання послідовності операцій множення, додавання, віднімання дійсних і комплексних. Для ефективної реалізації швидких алгоритмів ортогональних тригонометричних перетворень дійсної та комплексної послідовності в склад комп'ютерних систем обробки сигналів

і зображень доцільно включити багатооперандні операційні пристрої для обчислення БО та малоточкові процесори швидких дискретних тригонометричних перетворень.

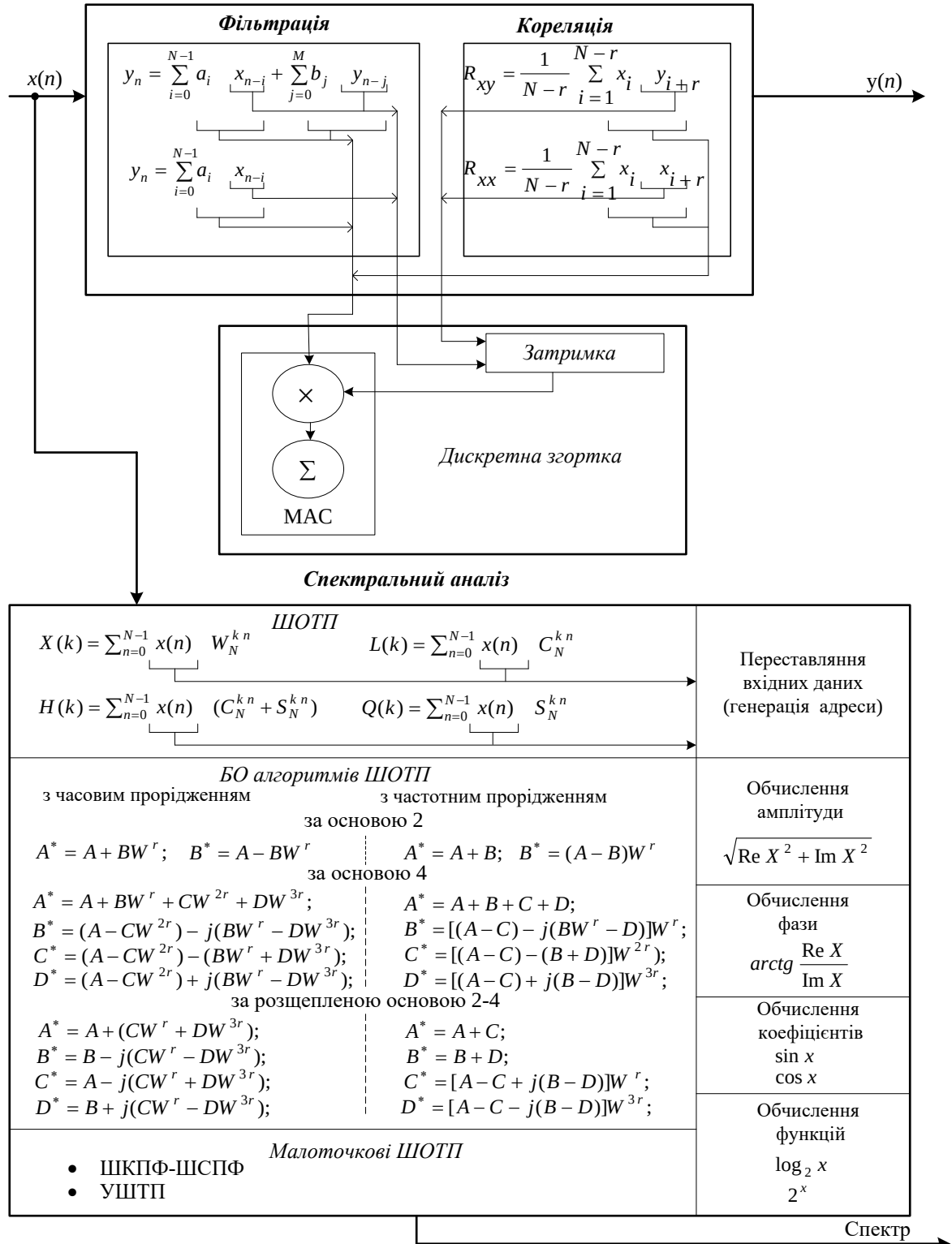


Рис.8.1 Операційний базис комп'ютерних систем обробки сигналів

Окрім перерахованих БО, при розв'язанні задач ЦОС і зображень великий об'єм обчислень займають операції обчислення тригонометричних функцій, добування квадратного кореня, піднесення до степені, ділення, сортування, переставляння елементів вхідних, вихідних масивів даних і генерації необхідних послідовностей адрес при звертанні до пам'яті.

Окрім перерахованих БО, при розв'язанні задач ЦОС і зображень великий об'єм обчислень займають операції обчислення тригонометричних функцій, добування квадратного кореня, піднесення до степені, ділення, сортування, переставляння елементів вхідних, вихідних масивів даних і генерації необхідних послідовностей адрес при звертанні до пам'яті.

На рис.8.1 наведений операційний базис комп'ютерних систем обробки сигналів і зображень, де МАС – множення з підсумовуванням; ШОТП – швидкі ортогональні тригонометричні перетворення; ШКПФ–ШСПФ – швидке косинус-синус перетворення Фур'є; УШТП – універсальні швидкі тригонометричні перетворення, які забезпечують реалізацію Фур'є, Хартлі, косинусного та синусного перетворень дійсної послідовності. Необхідно зауважити, що операційні можливості сучасних мікропроцесорів, на базі яких реалізуються комп'ютерні системи обробки сигналів і зображень, обмежені в частині команд обробки. Більшість сучасних мікропроцесорів апаратно реалізують тільки операції множення. Складніші операції реалізуються програмним способом, який є відносно повільним і вимагає значної кількості пересилань між операційними пристроями та пам'яттю. Для досягнення високої швидкодії в комп'ютерних систем обробки сигналів і зображень пропонується виділені БО і малоточкові ШОТП реалізувати апаратно.

Література : [1], [4], [5], [6], [9].

9. Елементна база комп'ютерних систем цифрової обробки сигналів

9.1 Мікропроцесори ЦОС.

9.2 Трансп'ютери та однорідні обчислювальні середовища.

9.3 Спеціалізовані НВІС.

Швидкий розвиток технологій ЦОС обумовлений не тільки успіхом в розробці нових алгоритмів, але і стрімким прогресом технології виготовлення надвеликих інтегральних схем (НВІС). Сучасні дослідження в теорії ЦОС та НВІС-технології суттєво змінили склад і архітектуру комп'ютерних систем ЦОС. В теперішній час для реалізації алгоритмів ЦОС застосовуються комп'ютерні системи з нетрадиційною архітектурою, що складаються з великої кількості спеціалізованих і функціонально орієнтованих процесорів. Подальший розвиток архітектури комп'ютерних систем ЦОС тісно пов'язаний з розвитком НВІС-технології, яка дозволяє зменшити енергоспоживання, масогабаритні характеристики, підвищити швидкодію та стимулює розвиток одних підходів проектування і забороняє інші.

Створення ефективних комп'ютерних систем ЦОС пов'язано з розвитком елементної бази, яка розвивається в трьох напрямках [2,3,5,7,9]:

- одно кристальні програмовані мікропроцесори та мікро-ЕОМ з архітектурою орієнтованою на розв'язання задач ЦОС;
- трансп'ютери та НВІС однорідних обчислювальних середовищ, які орієнтовані на масово-паралельні обчислення;
- замовні та напівзамовні спеціалізовані НВІС, які апаратно реалізують БО алгоритмів ЦОС та малоточкові ШОТП.

9.1 Мікропроцесори ЦОС

Мікропроцесори ЦОС (МЦОС) мають високу ступінь спеціалізації. В них широко використовуються методи скорочення тривалості командного циклу, характерні і для універсальних RISC-процесорів, такі як конвеєризація на рівні окремих мікроінструкцій та інструкцій, розміщення операндів більшості команд у регістрах, використання тінювих регістрів для зберігання

стану обчислень при переключенні контексту, поділ шин команд і даних (гарвардська архітектура). Звернемо увагу на те, що для МЦОС характерною є наявність апаратного перемножувача, що дозволяє виконувати множення двох чисел за один командний такт. В універсальних мікропроцесорах множення звичайно реалізується за декілька тактів, як послідовність операцій зсуву і додавання. Іншою особливістю МЦОС є включення в систему команд таких операцій, як множення з підсумовуванням МАС ($C := A \times B + C$) з зазначеним у команді числом виконань у циклі та з правилом зміни індексів використовуваних елементів масивів A і B), інверсія біту адреси, різноманітні бітові операції. У МЦОС реалізується апаратна підтримка програмних циклів, кільцевих буферів і вибір з пам'яті за цикл виконання команди декількох операндів.

За обробкою даних МЦОС діляться на два класи: з фіксованою та з плаваючою крапкою. Використання МЦОС з плаваючою крапкою обумовлено декількома причинами. Для багатьох задач, пов'язаних із виконанням інтегральних і диференціальних перетворень, особливе значення має точність обчислень, забезпечити яку дозволяє експоненційний формат представлення даних. Алгоритми компресії, декомпресії, адаптивної фільтрації в ЦОС пов'язані з визначенням логарифмічних залежностей і дуже чутливі до точності представлення даних у широкому динамічному діапазоні.

Робота з даними у форматі з плаваючою крапкою істотно спрощує і прискорює обробку, підвищує надійність програми, оскільки не потребує виконання операцій округлення і нормалізації даних, відслідковування ситуацій втрати значимості і переповнення. Платою за ці додаткові "комфорт і швидкість" є висока складність функціональних пристроїв, що виконують обробку даних у форматі з плаваючою крапкою, необхідність використання більш складних технологій виробництва мікросхем, більший відсоток відбракування виробів і як наслідок висока ціна мікропроцесорів.

Найпоширеніші МЦОС виготовляють такі фірми як Motorola (56002, 96002), Intel (i960), Texas Instruments (TMS320Cxx), Analog Devices (21xx, 210xx) [2,6,7]. Порівняльний аналіз основних характеристик МЦОС

одного покоління різних фірм показав відсутність суттєвих відмінностей, що пояснюється близькістю архітектури та технології виготовлення. За повнотою сімейства, наявністю інструментальних технологічних засобів і розробленого програмного забезпечення МЦОС із сімейства TMS320 переважають МЦОС інших фірм. Принципи проектування та особливості архітектури МЦОС із сімейства TMS320 є характерними і для всіх інших МЦОС. Тому архітектуру мікропроцесорів TMS320, можна розглядати як базову. В основу проектування мікропроцесорів TMS320 покладені наступні принципи: застосування модифікованої гарвардської архітектури, широке використання конвеєрного режиму роботи, наявність спеціалізованого пристрою множення, існування спеціальних команд ЦОС, короткий командний цикл.

Для мікропроцесорів серії TMS320 характерне використання апаратної реалізації ряду функцій, які звичайно виконуються програмно. Наприклад, апаратний перемножувач, який забезпечує множення двох чисел за один командний цикл. Є також апаратні паралельні зсувачі, які здійснюють зсув даних і результатів обчислень, індексні регістри, які забезпечують непряму адресацію даних в ОЗП та режим автоінкременту (декременту) при одноциклових маніпуляціях з таблицями даних.

Системи команд сімейства мікропроцесорів TMS320 є комплексними та сумісними знизу до верху. Вони містять інструкції різного функціонального призначення: універсальні - арифметичні, логічні і керуючі команди; спеціальні - призначені для розв'язання задач ЦОС. Наявність комплексної системи команд розширює області застосування МЦОС і спрощує розробку програм.

МЦОС з фіксованою крапкою. МЦОС TMS320C1x, C2x, C5x призначені для обробки чисел у форматі з фіксованою крапкою. Перше покоління мікропроцесорів TMS320C1x побудоване на базовій архітектурі мікропроцесора TMS320C10 [2,3,9], структура якого наведена на рис.9.1.

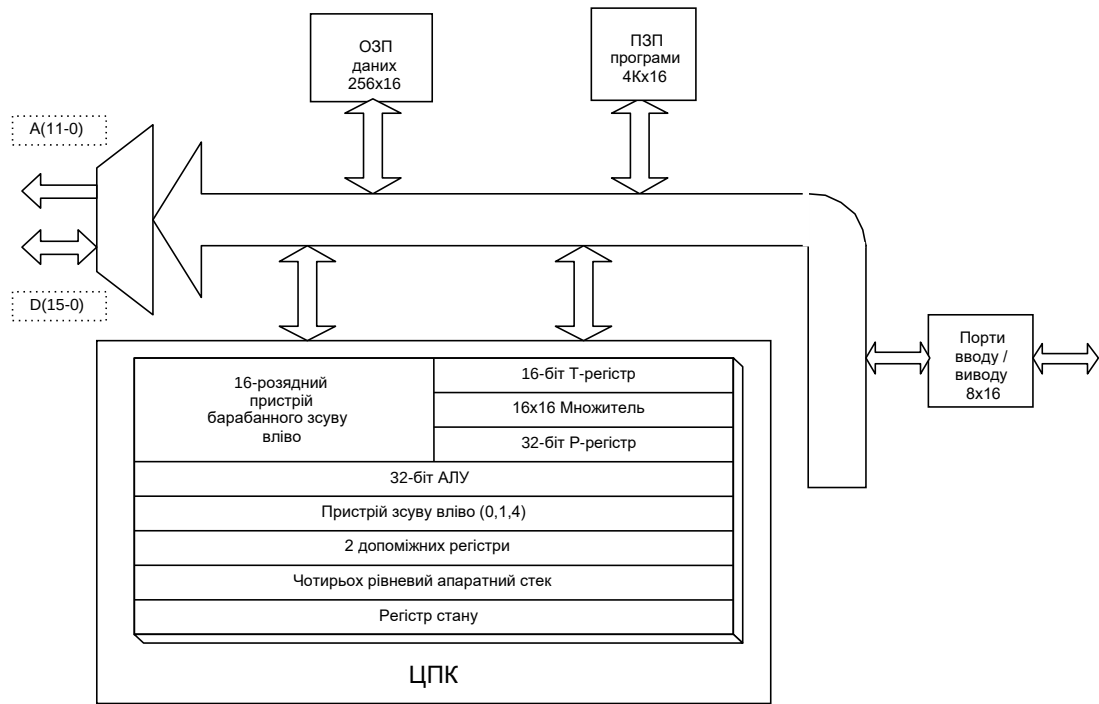


Рис.9.1 Структура мікропроцесора TMS320C10

Його адресний простір складає $4K \times 16$ -розрядних слів пам'яті програм і 144×16 -розрядних слів пам'яті даних. Тривалість командного такту процесора складає 160-200 нс. Арифметичні функції в процесорі реалізовані апаратно. З зовнішніми пристроями процесор взаємодіє через 8-м 16-розрядних портів вводу/виводу. Передбачено можливість обробки зовнішнього переривання.

Інші мікропроцесори даного сімейства (C14 – C17) мають аналогічну архітектуру і відрізняються тривалістю командного такту, конфігурацією пам'яті, наявністю (або відсутністю) додаткових периферійних пристроїв.

Мікропроцесори сімейства TMS320C2x мають підвищену продуктивність і ширші функціональні можливості. Всі мікропроцесори сімейства можуть використовувати пам'яті програм і даних ємністю 64K слів, послідовний порт та шістнадцять паралельних портів введення/виведення. Мікропроцесори сімейства TMS320C2x мають можливість використання зовнішнього контролера прямого доступу до пам'яті (ПДП). Пристрої множення мікропроцесорів, крім операцій множення, дозволяє виконувати за

один такт піднесення до квадрату. У мікропроцесори включена апаратна підтримка кратного виконання команди, реалізований режим двійкової інверсно-непрямой адресації, призначений для ефективної реалізації швидкого перетворення Фур'є.

Мікропроцесори сімейства TMS320C5x, забезпечуючи сумісність за системою команд і наслідуючи загальні архітектурні особливості побудови мікропроцесорів попереднього покоління, відрізняються більшими функціональними можливостями, підвищеною тактовою частотою, меншим енергоспоживанням. У МЦОС реалізована апаратна підтримка кільцевих буферів, є можливість одночасного створення в пам'яті даних двох незалежних кільцевих буферів. Існує можливість кратного виконання блока програми. Мікропроцесор містить одинадцять тіньових регістрів, що використовуються для швидкого зберігання/відновлення стану основних регістрів у випадку виникнення програмних або апаратних переривань. *Паралельний логічний пристрій мікропроцесора дозволяє виконувати бітові і логічні операції над операндами, що утримуються в пам'яті і різних регістрах.*

Мікропроцесор TMS320C5x може використовувати 244К слів пам'яті, у тому числі: 64К – пам'ять програм, 64К – пам'ять даних, 64К – 16-розрядні порти вводу/виводу, 32К – глобальна пам'ять. Для можливості роботи з повільною пам'яттю в мікропроцесор включений програмований генератор тактів чекання. При створенні мультипроцесорних систем виникає необхідність спільного використання єдиної області пам'яті. Для цього в процесорі передбачені сигнали запиту і готовності при звертанні до глобальної пам'яті, доступ до якої регулює спеціальний арбітр пам'яті. Різниця між мікропроцесорами – представниками сімейства TMS320C5x полягають, в основному, у конфігурації внутрішньокристалльної пам'яті.

Крім 16-розрядних портів вводу/виводу, мікропроцесори сімейства мають 2 послідовних порти (у TMS320C52 – один), таймер, інтерфейс тестування і налагодження JTAG.

МЦОС з плаваючою крапкою. Першим представником класу мікропроцесорів із плаваючою крапкою є TMS320C30, він має гнучку систему команд, апаратну підтримку операцій із плаваючою крапкою, потужну систему адресації, розширений адресний простір і підтримку мови високого рівня – Сі [3]. Мікропроцесор виготовляється за 0,7 мікронною КМОН технологією з 3-ма рівнями металізації. Всі операції в мікропроцесорі виконуються за один такт. При тривалості такту 60 нс процесор TMS320C30 має швидкодію біля 33 млн. операцій із плаваючою крапкою в секунду. Висока продуктивність мікропроцесора на алгоритмах ЦОС забезпечується завдяки апаратному виконанню ряду специфічних функцій, що в інших мікропроцесорах реалізуються програмно або мікропрограмно. Мікропроцесор має конвеєрну регістро-орієнтовану архітектуру і може паралельно виконувати в одному такті множення і арифметико-логічні операції з числами у форматі з фіксованою або плаваючою крапкою. Структура мікропроцесора TMS320C30 наведена на рис.9.2.

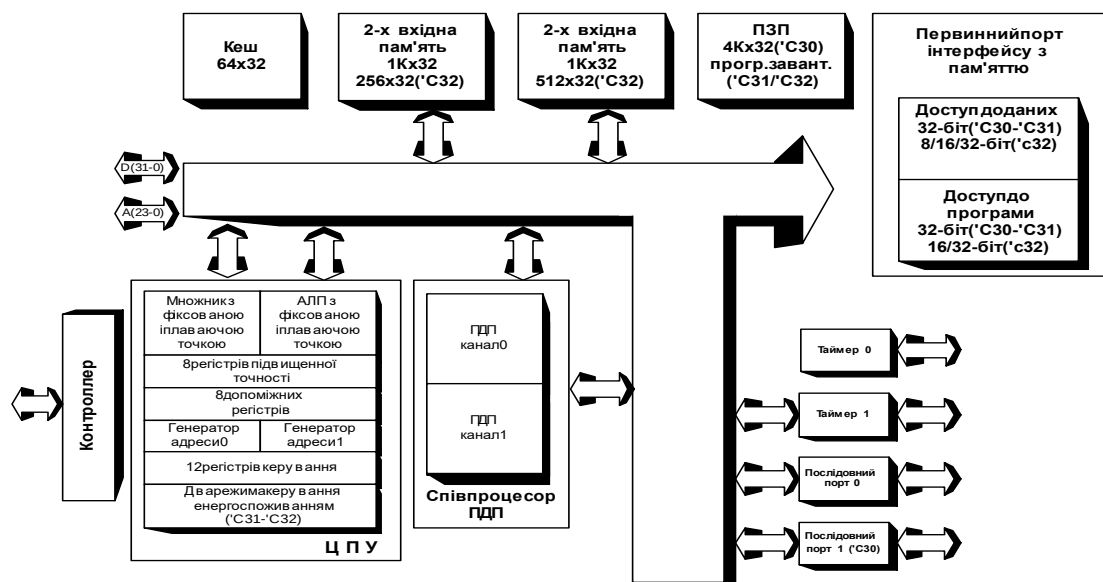


Рис.9.2 Структура мікропроцесора TMS320C30

В склад мікропроцесора TMS320C30 входить: 32-розрядна шина команд і даних; 24-розрядна шина адреси; 2-а блоки ОЗП; 32-розрядний перемножувач із плаваючою крапкою; кеш-пам'ять команд ємністю 64-и

слова; 8-м регістрів для операцій із підвищеною точністю; два генератори адреси; регістровий файл; 40-розрядний АЛП, який працює як із цілими числами, так і з числами у форматі з плаваючою крапкою. Вмонтований контролер ПДП дозволяє поєднувати в часі виконання обмінів даними з пам'яттю і обчислення. Наявність у TMS320C30 мультипроцесорного інтерфейсу, двох зовнішніх інтерфейсних портів, двох послідовних портів, розширеної системи переривань спрощує конструювання систем на його основі. Завдяки своїй високій продуктивності і простоті використання в обчислювальних системах TMS320C30 може застосовуватися як у якості головного процесора, так і в якості спеціалізованого співпроцесора.

Наступними представниками МЦОС із плаваючою точкою є мікропроцесори сімейства TMS320C4x. Завдяки своїй унікальній архітектурі мікропроцесори TMS320C40 одержали широке поширення в мультипроцесорних системах і практично витіснили раніше пануюче в цій технологічній ніші сімейство трансп'ютерів. Процесори TMS320C4x сумісні по системі команд із TMS320C3x, проте мають більшу продуктивність і кращі комунікаційні можливості.

У сімейство TMS320C4x входять процесори TMS320C40, TMS320C44, TMS320LC40. TMS320C40 має продуктивність 30MIPS/60MFLOPS і максимальну пропускну спроможність підсистеми вводу/виводу 384 Мбайт/с. TMS320C40 містить на кристалі шість високошвидкісних (20 Мбайт/с) комунікаційних портів і шість каналів ПДП, 2К слів пам'яті, 128-м слів програмного кеша. Дві зовнішні шини забезпечують доступ до 4Г слів.

Висока продуктивність і хороші комунікаційні властивості процесора дозволяють ефективно використовувати його для розв'язання задач обробки зображень, моделювання віртуальної реальності, розпізнавання мови, моніторингу.

Мультипроцесорні МЦОС. Подальшим розвитком сімейства МЦОС Texas Instruments є мікропроцесори принципово нової архітектури – TMS320C8x [9].

Друга назва процесорів – MVP (Multimedia Video Processor) – характеризує їх високу ефективність на задачах обробки зображень, 2- і 3-вимірній графіці, у системах віртуальної реальності, компресії і декомпресії відео- і аудіоданих і ін.

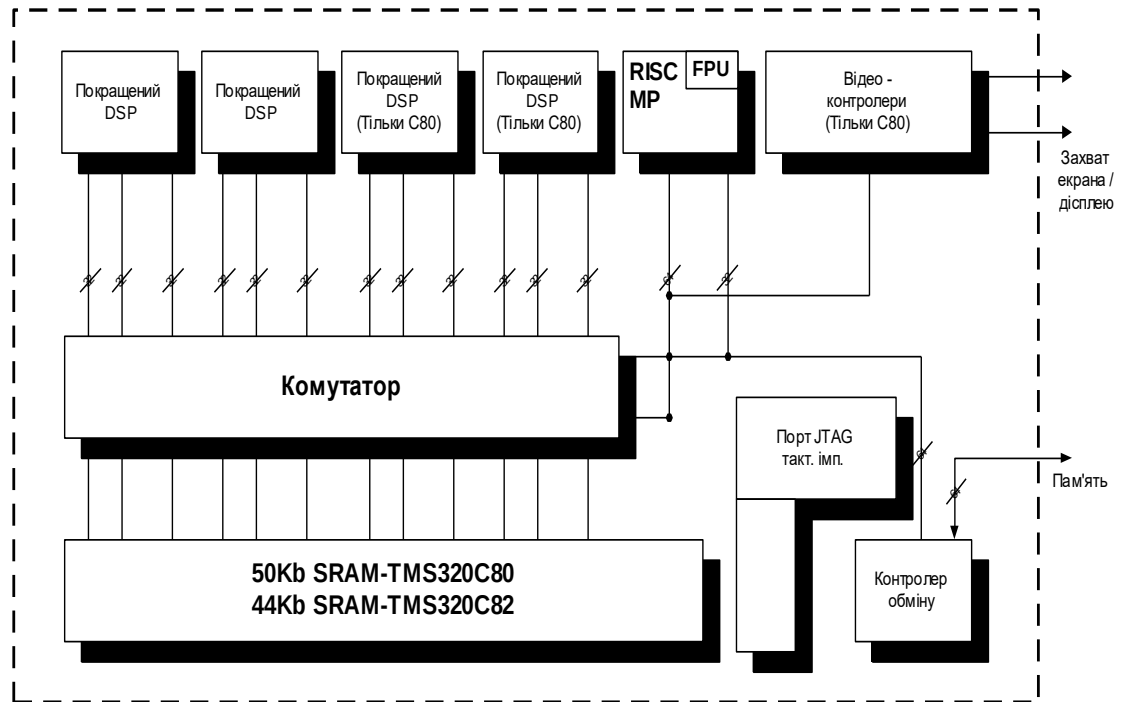


Рис.9.3 Структура мікропроцесора TMS320C80

Базовим МЦОС сімейства TMS320C8x є TMS320C80 (рис. 9.3), який об'єднує в одній мікросхемі п'ять повнофункціональних процесорів, чотири з яких – покращені процесори ЦОС (Advanced Digital Signal Processor).

Кожний із ADSP дозволяє виконати за один командний такт декілька RISC-подібних операцій. П'ятий процесор, головний (Master Processor (MP)), являє собою 32-розрядний RISC-процесор із високопродуктивним обчислювачем із плаваючою крапкою, сумісним із стандартом IEEE 754. Крім процесорного ядра на кристалі розміщені: контролер обміну (Transfer Controller (TC)) – інтелектуальний контролер ПДП, що підтримує інтерфейс із DRAM і SRAM, відеоконтролер (Video Controller (VC)), система контролю і налагодження – порт JTAG (IEEE 1149.1), 50Кб SRAM.

Випускається також спрощений варіант мікропроцесора TMS320C82, що відрізняється меншим обсягом пам'яті, кількістю сигнальних процесорів ADSP (2), відсутністю відеоконтролера і відповідно меншою вартістю.

Сумарна продуктивність TMS320C80 на регістрових операціях досягає 2 млрд. RISC-подібних команд в секунду. Завдяки високій продуктивності TMS320C80 може замінити при реалізації ряду ужитків більш 10 високопродуктивних МЦОС або мікропроцесорів загального призначення. Пропускна здатність шини TMS320C80 досягає 2,4 Гбайт/с – у потоці даних і 1,8 Гбайт/с – у потоці інструкцій.

TMS320C80 забезпечує високу ступінь гнучкості й адаптивності системи, побудованої на його базі, що досягається за рахунок наявності на кристалі паралельно функціонуючих процесорів ЦОС і головного RISC-процесора. Архітектура процесора TMS320C80 відноситься до класу MIMD-множинний потік даних, множинний потік команд. Процесори, що входять до складу TMS320C80 програмується незалежно один від іншого і можуть виконувати як різні, так і одну загальну задачу. Обмін даними між процесорами здійснюється через спільну внутрішньокристалічну пам'ять. Доступ до розподіленої внутрішньокристалічної пам'яті забезпечує матричний комутатор (Crossbar), що виконує також функції монітора при звертанні декількох процесорів до одного сегмента пам'яті.

9.2 Трансп'ютери та однорідні обчислювальні середовища

Трансп'ютери відносяться до класу RISC-процесорів. Сімейство трансп'ютерних HBIC, включає ряд типів HBIC трансп'ютерів, а також зовнішніх запам'ятовуючих пристроїв (ЗП), призначених для підключення до трансп'ютерів. Основними типами HBIC трансп'ютерів є: трансп'ютер зі статичним ОЗП; трансп'ютер-зв'язний адаптер; трансп'ютер-контролер графічних пристроїв, трансп'ютер-контролер масової пам'яті. Кожний з цих типів може включати ряд HBIC, що різняться типом процесора, типом і конфігурацією внутрішнього ЗП, швидкодією та іншими параметрами [9].

Структура трансп'ютера IMS T424 є характерною для більшості трансп'ютерів (рис. 9.4).

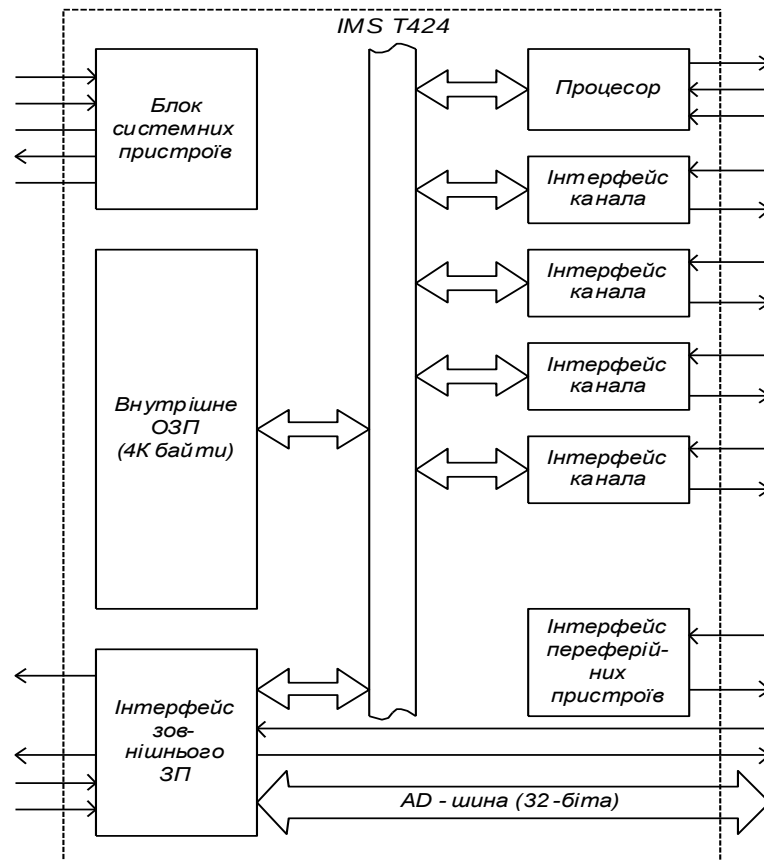


Рис.9.4 Структура трансп'ютера IMS T424

Основними елементами такої структури є 32-розрядний процесор, внутрішнє ОЗП ємністю 4К байт, блок системних пристроїв, чотири блоки інтерфейсів каналів зв'язку, блок інтерфейсу зовнішнього ЗП і блок інтерфейсу периферійних пристроїв.

Трансп'ютер можна використовувати як в якості автономного однокристального обчислювального пристрою (мікро-ЕОМ), так і процесорного модуля (ПМ) для побудови високопродуктивних паралельних обчислювальних систем. При використанні трансп'ютера в якості однокристальної мікро-ЕОМ можна використовувати стандартні алгоритмічні мови високого рівня. Але трансп'ютер спеціально спроектовано таким чином, щоб забезпечити ефективну реалізацію конкретної, спеціально розробленої для нього мови паралельного програмування Оссам.

Мова Оссам дозволяє в явному вигляді програмувати функціонування паралельних систем. З одного боку, вона як асемблер забезпечує ефективне використання пам'яті і високу продуктивність при виконанні програм, а з іншого боку, подібно мовам високого рівня – велику ефективність при розробці програмного забезпечення і його високу надійність. Мова Оссам є основою методології проектування паралельних систем на базі трансп'ютерів подібно тому, як булева алгебра, - основою методології проектування сучасних електронних систем на базі логічних елементів.

Задача розробки трансп'ютерних систем полегшується завдяки архітектурній відповідності між мовою Оссам і трансп'ютером. Програма, яка виконується одним трансп'ютером, формально еквівалентна одному процесу мови Оссам. З цього випливає, що систему, яка складається з деякої сукупності трансп'ютерів, можна безпосередньо описати за допомогою програми на даній мові. Основне поняття, що використовується в мові Оссам, - це поняття процесу. Кожний процес розглядається як деякий об'єкт (чорна скринька), що характеризується своїм внутрішнім станом. Процес може змінювати свій внутрішній стан і взаємодіяти з іншими процесами шляхом обміну повідомленнями, використовуючи для цього двоточкові канали зв'язку. Під каналом розуміється мовна конструкція, що забезпечує можливість організації зв'язку між процесами, а не фізичний канал (link), що використовується для зв'язку між процесорними модулями.

Повідомлення, що відсилаються процесом, залежать від його внутрішнього стану. Цей стан, в свою чергу, змінюється при прийомі даним процесом повідомлень від інших процесів. Таке поняття процесу можна використовувати для опису поведінки найрізноманітніших об'єктів, наприклад логічного елементу, мікропроцесора, станка, закладу і так далі. Процеси є скінчені, тобто кожен процес починає своє виконання, виконує ряд дій і завершується. Кожна дія може представляти собою або множину послідовних процесів, що виконуються один за одним, або множину паралельних процесів, що виконуються одночасно, або множину альтернативних процесів, тобто таких, з множини яких виконується тільки

один, що є вибраний у відповідності до заданого критерію. Кожен процес може складатися з інших процесів, причому деякі з цих процесів можуть бути паралельними, тому процес може мати будь-який рівень внутрішньої паралельності і цей рівень може змінюватися в залежності від моментів початку та завершення процесів.

Всі процеси складаються з наступних примітивних (основних) процесів: присвоєння, вводу, виводу, пропуску, зупинки. Процес присвоєння обчислює значення деякого виразу і присвоює це значення деякій змінній. Процеси вводу і виводу використовуються для організації взаємодії між більш складними процесами.

Два паралельних процеси можуть підтримувати зв'язок один з одним використовуючи однонаправлений канал, який зв'язує два дані процеси. При виконанні такого зв'язку один процес посилає (виводить) повідомлення в канал, а інший процес приймає (вводить) це повідомлення з каналу. Ключова концепція мови Оссам є в тому, що цей зв'язок є синхронним. Він виконується тільки в тому випадку, коли перший процес готовий виконати вивід, а другий процес – ввід. Якщо один процес готовий до виконання відповідної дії раніше ніж другий, то він повинен чекати готовності другого процесу. Таким чином, спосіб взаємодії між процесами подібний до методу “рукопотиснення”, що використовується для організації зв'язку в апаратних засобах. Якщо процес має внутрішню паралельність, то він може використовувати для зв'язку з іншими процесами одночасно декілька каналів, так як наявні в ньому паралельні процеси можуть одночасно виконувати введення і виведення для декількох каналів.

Мова Оссам є для трансп'ютера мовою того ж рівня, що і асемблер. Трансп'ютер має спеціальні засоби для реалізації концепції паралельності процесів, що є прийняті в даній мові. Цю мову можна використовувати для написання програм, що є призначені як для окремого трансп'ютера, так і для систем, що складаються із декількох трансп'ютерів, тобто мультитрансп'ютерних систем.

Однорідне обчислювальне середовище (ООС) (рис.9.5) - це двовимірна регулярна матриця процесорних елементів (ПЕ), кожен з яких фізично зв'язаний за входом-виходом з чотирма сусідами – зверху, знизу, зліва та справа (комутація).

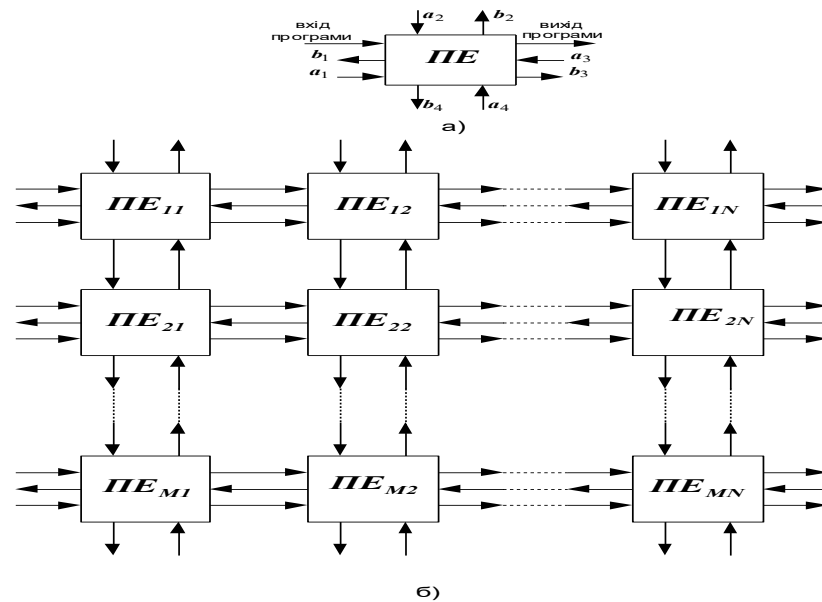


Рис.9.5 Структури: а)- процесорного елемента, б)- ООС на основі ПЕ.

Кожний ПЕ може виконувати набір бітових операцій перетворення інформації з вхідних каналів у вихідні. ООС є універсальною системою, тобто в ньому можливо реалізувати довільну обчислювальну функцію. Бітовий рівень ПЕ та повна система комутації дозволяють реалізувати паралелізм та конвеєрність обчислень на найнижчому бітовому рівні. Це є суттєвою перевагою ООС при реалізації програмно налаштованих спеціалізованих паралельних комп'ютерних систем для задач ЦОС. В багатьох випадках при реалізації обчислень на ООС вимагається організація взаємодії такої системи з однорідним запам'ятовуючим середовищем, яке є матрицею регістрів зсуву, довжина котрих програмується. Промисловістю випускається ООС з 20-92 ПЕ на кристалі при продуктивності одного ПЕ біля 10млн. одnobітових операцій в секунду.

9.3 Спеціалізовані НВІС

За способом проектування і виготовленням, тобто налаштуванням на виконання конкретного алгоритму, спеціалізовані НВІС поділяються на два класи: замовні і напівзамовні (рис.9.6).

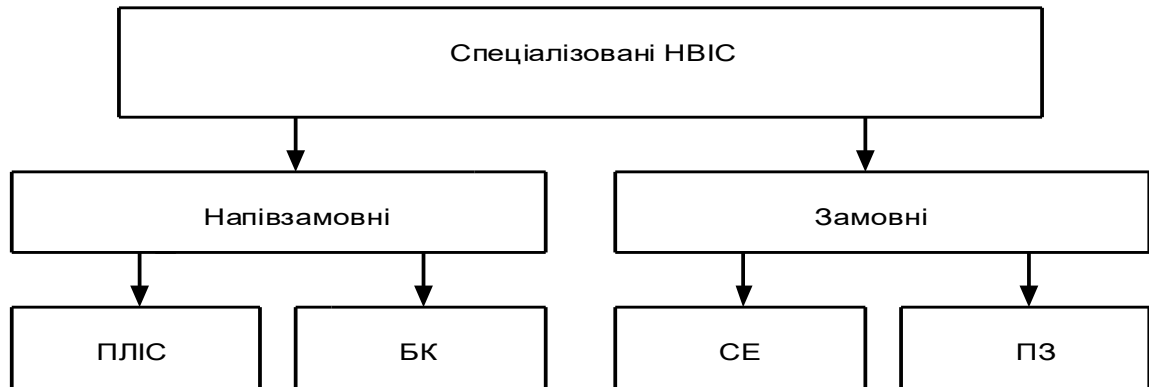


Рис.9.6 Класифікація спеціалізованих НВІС

Замовні НВІС - це мікросхеми, розроблені на основі стандартних або спеціально створених елементів і вузлів за схемою замовника. Всі топологічні шари замовних НВІС проектується і виготовляються індивідуально. Існують повністю замовні (ПЗ) НВІС, схеми яких оптимізовані на рівні окремих компонентів, та НВІС побудовані на основі стандартних елементів (СЕ), які вибираються із раніше спроектованої і перевіреної бібліотеки елементів. В склад бібліотеки можуть входити прості логічні елементи типу І-НЕ, АБО-НЕ, тригери, а також складніші типу суматори, регістри, комутатори та інші. Особливістю замовних НВІС є оптимізація елементів і зв'язків на реалізацію алгоритмів, що дозволяє досягнути граничних значень параметрів для кожного рівня технології.

Напівзамовні НВІС - це мікросхеми, що складаються з двох частин: наперед спроектованої постійної та змінної - замовної, структура якої визначається замовником. До напівзамовних НВІС відносяться мікросхеми на основі базових кристалів (БК) та програмовані користувачем логічні інтегральні схеми (ПЛІС).

Проектування пристроїв на БК здійснюється за рахунок нанесення відповідних шарів з'єднань. Основними елементами БК є базові комірки, що складаються з набору некомутованих елементів-транзисторів і резисторів. На базі таких елементів реалізуються функціонально завершені вузли, які виконують елементарні функції.

В порівнянні з БК, технологія ПЛІС забезпечує рекордно малий проектно-технологічний цикл (від декількох годин до декількох днів), мінімальні витрати на проектування, максимальну гнучкість при необхідності модифікації апаратури.

На даний час на світовому ринку можна виділити декілька компаній-виробників – ПЛІС-XILINX, ALTERA, LATTICE, AT&T, INTEL, які виготовляють мікросхеми з архітектурою EPLD (EPROM technology based complex Programmable Logic Device) – з можливістю багаторазового перепрограмування, і FPGA (Field Programmable Gate Array) – з можливістю багаторазового реконфігурування.

В якості пам'яті для зберігання конфігурації в ПЛІС EPLD використовується ППЗП з ультрафіолетовим стиранням, а у ПЛІС FPGA – статичний ОЗП.

Мікросхема FPGA являє собою матрицю логічних комірок, що з'єднані між собою логічними ключами. Статична пам'ять, яка є в мікросхемах FPGA при заповненні деякою бітовою послідовністю діє на логічні комірки і з'єднує їх через ключі, що дозволяє отримати необхідні електричні схеми (реєстри, лічильники, логічні схеми і ін., що з'єднані в необхідній послідовності). Кожна мікросхема FPGA має вхід для запису бітової послідовності, яка заповнює статичну пам'ять, а також елементи “вхід/вихід” для зв'язку з іншими мікросхемами. Таким чином, на основі однієї чи декількох мікросхем FPGA можна створювати реконфігурований процесор з перевагами спеціалізованого процесора на “жорсткій” логіці, але з можливістю шляхом зміни вмісту статичної пам'яті.

Відмінною особливістю ПЛІС архітектури FPGA різновидів XILINX XC3000, XC3100, XC4000 є наявність поля логічних блоків і блоків введення/виведення, які зв'язані між собою через комутаційні блоки. Логічні блоки, блоки введення/виведення і комутаційні поля конфігуруються при завантаженні в ПЛІС бітової послідовності, що отримана в результаті розробки схеми. В залежності від різновиду ПЛІС логічні блоки, блоки введення/виведення, комутаційні блоки мають різну ступінь складності і володіють різними функціональними можливостями.

Логічний блок – один з базових елементів архітектури ПЛІС FPGA, може виконувати будь-яку логічну функцію в залежності від заданої бітової послідовності. Шляхом завантаження іншої бітової послідовності можна змінювати виконувану функцію необмежену кількість разів. Блок введення/виведення, так само як і логічний блок, може бути налаштований на виконання будь-якого електричного з'єднання реалізованої в середині ПЛІС схеми з зовнішніми пристроями через відповідний контакт мікросхеми.

Література : [1], [2], [4], [6], [8, [9], [11].

10. Особливості структурної організації пам'яті комп'ютерних систем цифрової обробки сигналів

10.1 Багатомодульна пам'ять.

10.2 Пам'ять з паралельним доступом.

10.3 Кеш-пам'ять.

10.4 Особливості доступу до пам'яті.

10.5 Організація пам'яті комп'ютерних систем ЦОС.

10.6 Принципи побудови пам'яті комп'ютерних систем ЦОС.

10.1 Багатомодульна пам'ять

Одномодульна структура пам'яті з однією шиною адреси і даних характерна для процесорів з архітектурою фон Неймана. Особливістю даної структури є можливість забезпечення тільки одного звертання до пам'яті протягом одного циклу команди. Збільшити кількість звертань до пам'яті на протязі одного циклу виконання команди дозволяє багатомодульна структура пам'яті з двома або більше шинами адреси і даних. Така організація пам'яті характерна для більшості ПЦОС [2,3,6,7]. Основною перевагою даної архітектури є можливість виконання двох звертань до пам'яті протягом одного циклу виконання команди. Крім цього, таке розділення пам'яті дозволяє сумістити в часі виклик команд і їх виконання.

Подальшим розвитком даної архітектури є модифікована гарвардська архітектура, яка дозволяє обмін інформацією між пам'яттю даних і пам'яттю команд. Ця модифікація забезпечує можливість зчитування в пам'ять даних коефіцієнтів і констант, записаних в пам'яті програм і тим самим виключає використання спеціальних постійних запам'ятовуючих пристроїв. Вона також забезпечує можливість використання команд з безпосередньою адресацією та звертання до підпрограм за результатами обчислень. З метою скорочення довжини командного циклу ПЦОС гарвардська архітектура доповнюється конвеєрним режимом роботи. В залежності від типу ПЦОС конвеєр може мати від двох до чотирьох сходинок. Це означає, що процесор може одночасно опрацьовувати від двох до чотирьох команд, при чому кожна з команд буде знаходитись на різних етапах виконання. Прикладом двохмодульної організації пам'яті є гарвардська архітектура (рис. 10.1), де пам'ять розділена на два незалежні блоки (пам'ять програм і пам'ять даних), які під'єднані до процесора за допомогою двох пар шин.

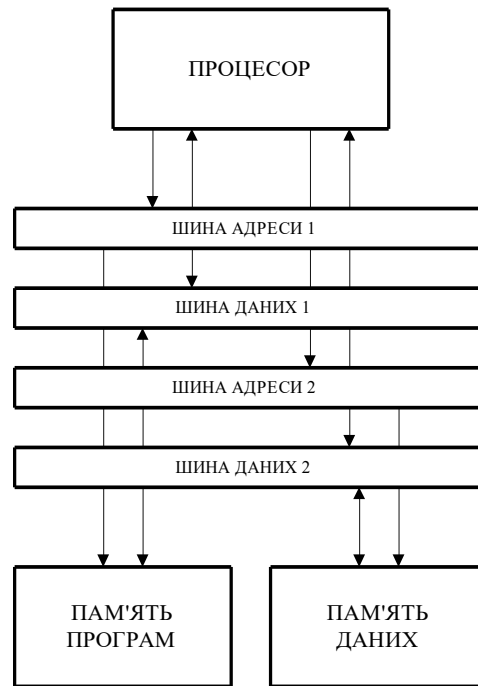


Рис. 10.1 Структура пам'яті процесора з гарвардською архітектурою

Модифікована гарвардська архітектура з двома модулями пам'яті використовується в ПЦОС фірми Texas Instruments (TMS320C1x), Analog Devices (ADSP-21xx) і AT&T (DSP16xx).

Подальшого збільшення кількості звертань до пам'яті можна досягнути шляхом використання тримодульної модифікованої гарвардської архітектури. В ПЦОС з такою архітектурою пам'ять розділена на три незалежні модулі пам'яті, причому кожен з модулів має свою власну множину шин. Один з модулів пам'яті можна використовувати для зберігання команд і даних, а два інші лише для зберігання даних. Архітектура з тримодульною пам'яттю дозволяє процесору виконувати три незалежні звертання до пам'яті за один цикл команди. Одне звертання до модуля пам'яті для отримання команди, а два інші модулі - для отримання даних. Для забезпечення високої швидкодії потоки даних і команд в такій архітектурі розділені. Модифікована гарвардська архітектура з трьохмодульною пам'яттю використовується у швидкодіючих ПЦОС фірм Zilog (Z893xx), Motorola (DSP560xx, 563xx і 96008), Texas Instruments (TMS320C2x-TMS320C5x).

Велику роль в забезпеченні високої швидкодії перерахованих вище структур ПЦОС відіграє кількість внутрішніх шин і їх зв'язок з модулями пам'яті, розміщеними на кристалі і поза ним. Розподіл внутрішніх шин в ПЦОС дозволяє здійснювати паралельні виклики програм, доступ до даних і прямий доступ до пам'яті. Розглянемо організацію внутрішньокристалічної пам'яті і зв'язок з шинами на прикладі ПЦОС TMS320C40 (рис. 10.2).

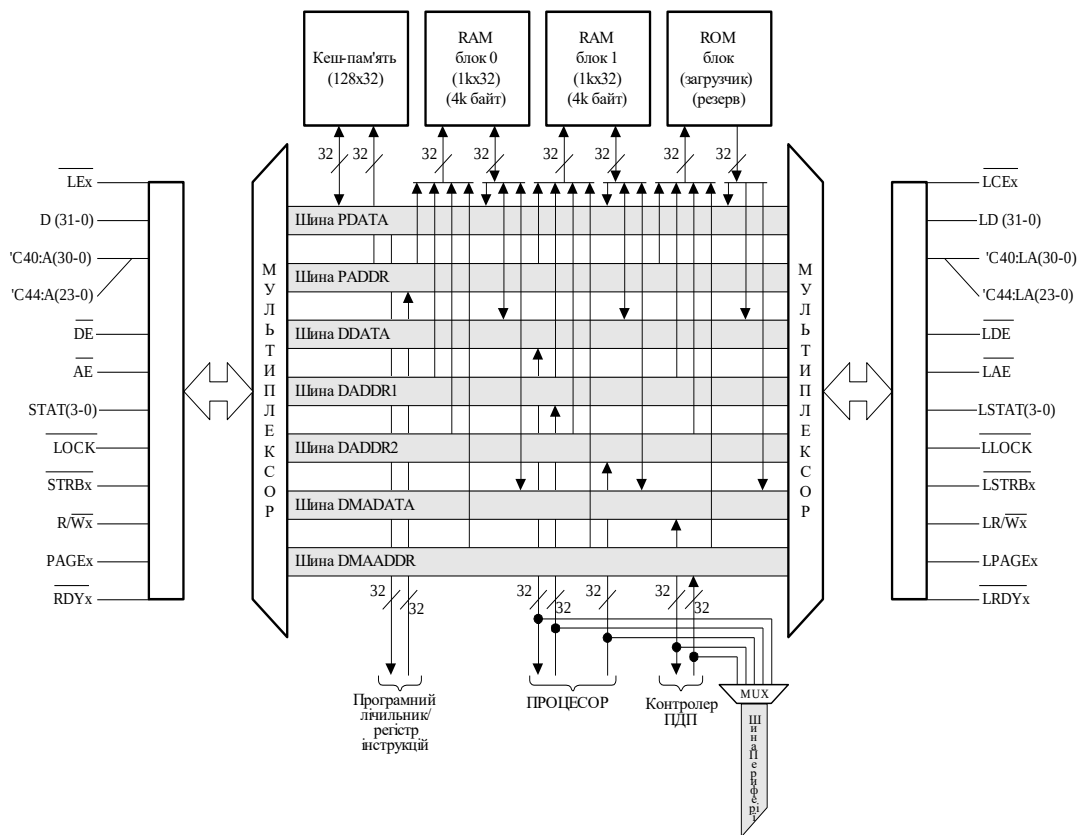


Рис. 10.2 Організація пам'яті процесора TMS320C40

Даний процесор має сім шин, одна з яких є шиною команд (PDATA), дві - шинами даних (DDATA, DMADATA) і чотири - шинами адрес (PADDR, DADDR1, DADDR2 і DMAADDR). Шина адреси пам'яті програм PADDR пов'язана з лічильником адреси програм, а шина команд PDATA - з регістром команд. Ці шини дозволяють вибирати одне слово команди в кожному циклі. Шини адреси - DADDR1 і DADDR2 та шина даних DDATA можуть забезпечувати два доступи до пам'яті в кожному циклі роботи. Прямий доступ до пам'яті підтримується адресною шиною DMAADDR і шиною даних DMADATA. Ці шини дозволяють здійснювати прямий доступ до пам'яті одночасно з доступом, що може здійснюватись з шин програм і даних.

Особливістю ПЦОС з двома і трьома модулями пам'яті на кристалі є відносно мала ємність їхньої пам'яті. Розширення ємності пам'яті однокристальних ПЦОС досягається шляхом під'єднання зовнішньої пам'яті. Оскільки розміщення декількох множин шин поза кристалом вимагає великої кількості виводів і додаткових апаратних витрат, то більшість однокристальних ПЦОС використовують одну множину шин (адреси, даних і управління) поза кристалом. Зовнішня пам'ять, під'єднана до однієї множини, дозволяє лише одне звертання протягом одного командного циклу. Тому команди, які вимагають декількох звертань до зовнішньої пам'яті, виконують за декілька циклів, що значно зменшує швидкодію процесора.

Серед інших шляхів збільшення кількості звертань до пам'яті, є використання швидкодіючої пам'яті, яка підтримує декілька послідовних звертань за один командний цикл. Використання швидкої пам'яті в ПЦОС з гарвардською архітектурою дозволяє отримати більшу швидкодію ніж при використанні її окремо. Прикладом такого використання є процесор ZR3800х фірми Zoran, який поряд з модулем пам'яті з одиночним доступом використовує модуль швидкодіючої пам'яті з подвійним доступом. В даному процесорі за один командний цикл може виконуватися одне звертання до пам'яті програм і два звертання до пам'яті даних.

В ПЦОС з гарвардською архітектурою збільшення кількості звертань до пам'яті за один командний цикл можливе при використанні в якості пам'яті програм і даних двох модулів швидкодіючої пам'яті. Така пам'ять дозволяє виконання чотирьох звертань за один командний цикл при умові, що звертання впорядковані таким чином, що кожен модуль обслуговує два звертання. Широке використання швидкодіючої пам'яті на кристалі стримується значним зростанням потужності споживання та його вартості. При переміщенні швидкої пам'яті або її частини за межі кристалу, необхідно вирішити питання мінімізації затримок та усунення завад, що виникатимуть при такому обміні.

10.2 Пам'ять з паралельним доступом

Радикальним шляхом збільшення кількості звертань до пам'яті за один цикл виконання команди є використання багатопортової пам'яті. Особливістю багатопортової пам'яті є незалежні множини шин адреси і даних, кількість яких відповідає кількості портів. Така пам'ять підтримує одночасний доступ до неї з всіх портів. Найбільш популярною багатопортовою пам'яттю є двохпортова пам'ять, яка дозволяє одночасно два звертання. Рідше в процесорах ЦОС використовується трьох і чотирьох-портова пам'ять. Особливістю багатопортової пам'яті є паралелізм доступу до множини даних і вирішення всіх конфліктів, що пов'язані з паралельністю доступу. В той же час, основним недоліком багатопортової пам'яті є великі апаратні витрати, які в першу чергу залежать від паралельності доступу, а в другу чергу - від ємності пам'яті.

В ПЦОС багатопортова пам'ять переважно використовується в парі з однопортовою пам'яттю. Прикладом такого поєднання служить процесор фірми Motorola DSP561xx (рис. 10.3). В даному процесорі за один командний цикл може виконуватись одне звертання до пам'яті програм і два звертання до пам'яті даних. Переміщення багатопортової пам'яті за межі кристалу викликає потребу у додатковому виведенні назовні декількох шин адреси і даних. Це в свою чергу вимагає збільшення кількості контактів, розмірів кристалу і відповідно його ціни.

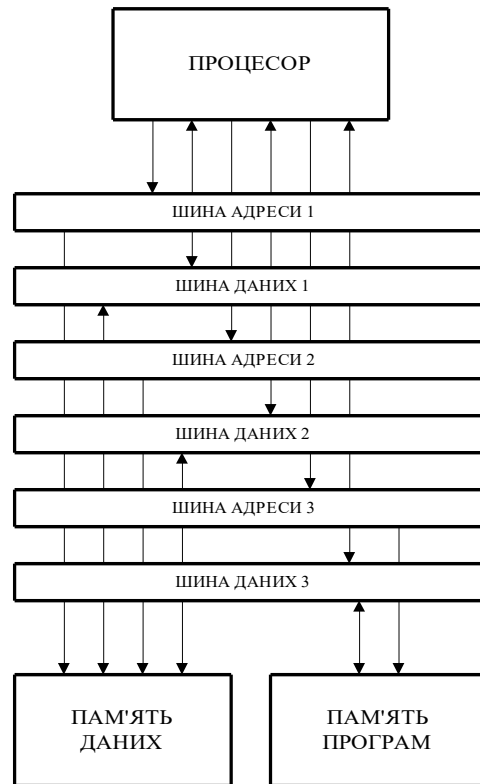


Рис.10.3 Система пам'яті з використанням модуля багатопортової пам'яті

Поряд із пам'яттю загального призначення існують спеціалізовані мікросхеми пам'яті, оптимізовані для використання в пристроях визначеного класу. У комп'ютерній індустрії найчастіше використовується спеціалізована пам'ять у відеоконтролерах.

10.3 Кеш-пам'ять

Використання в складі ПЦОС швидкодіючої кеш-пам'яті програм дозволяє підвищити його швидкодію шляхом уникнення звертань до менш швидкодіючої пам'яті програм при зчитуванні даних і команд. В ПЦОС використовуються простіші і менші за ємністю кеш-пам'яті ніж в універсальних процесорах. Типи кеш-пам'яті, що використовуються в ПЦОС відрізняються за функціонуванням і ємністю.

Найпростішим типом кеш-пам'яті є буфер повторення однієї інструкції. Це є кеш на одне слово команди з використанням спеціальної команди повторення. Команда, яка буде виконуватись декілька разів завантажується в буфер перед її першим виконанням. Виконання даної команди здійснюється

шляхом її зчитування з кеш-пам'яті, що дозволяє звільнити пам'ять програм для доступу до даних. Даний тип кеш-пам'яті використовується в процесорах TMS320C2x і TMS320C5x [], де він дозволяє при повторному виконанні команди забезпечити продуктивність, яка може бути досягнута при трьох звертаннях до пам'яті за один командний цикл. Очевидним недоліком використання буферу повторення є те, що він працює тільки з однією командою. Якщо для деяких алгоритмів цей метод використання кеш-пам'яті є ефективним, то для алгоритмів з повторенням блоків команд він є недоцільним.

Концепція буферу повторення може бути вдосконалена шляхом його розширення на запам'ятовування кодів декількох команд. Прикладом такої конфігурації кеш-пам'яті може бути буфер повторення на 16 команд процесора DSP16xx (ATT&T). В даному процесорі при першому виконанні блоку команд, він копіюється в буфер, а при кожному його повторі, команди читаються з буфера, звільняючи таким чином процесор від додаткових звертань до пам'яті. Такий тип кеш-пам'яті є ефективним при виконанні алгоритмів ЦОС, так як більшість з них використовують цикли з декількох команд.

Завдання узгодження буферу повторення декількох команд вирішує проста односекторна кеш-пам'ять команд. Ця кеш-пам'ять дозволяє зберігати деяку кількість команд, які найчастіше використовуються. Односекторна кеш-пам'ять команд завантажується кожною командою, що виконується і відстежує адрес команд в кеші. Коли процесор звертається за командою, то при наявності її в кеш-пам'яті, вона зчитується, а при її відсутності - вміст кеш-пам'яті стає недійсним. Дана кеш-пам'ять є ефективною при доступі до єдиної неперервної ділянки пам'яті програм. Прикладом ПЦОС з використання односекторної кеш-пам'яті є процесор ZR3800x фірми Zoran. Як і буфери повторення для декількох команд, так і односекторна кеш-пам'ять використовується для вирішення задач повторного виконання невеликих груп команд.

Більш гнучкою структурою є кеш-пам'ять з декількома незалежними секторами. Кеш такого типу функціонує подібно до простої односекторної кеш-пам'яті, але може зберігати дві або і більше незалежні ділянки пам'яті програм. Такий тип кеш-пам'яті використовується в високопродуктивних ПЦОС. Так в процесорі TMS320C3x [] ця пам'ять складається з двох секторів ємністю 32 слова. Кожен сектор може використовуватися для зберігання команд з незалежних ділянок пам'яті програм об'ємом 32 слова. Дана кеш-пам'ять працює наступним чином: при попаданні адреси в один з секторів кеш-пам'яті (співпадіння кешу), слово зчитується з кешу, а при не співпадінні адреси - генерується сигнал промаху сектору. В цьому випадку весь вміст одного з секторів оновлюється за алгоритмом LRU (least recently used), за яким вибирається той сектор, до якого довгий час не було звертань.

В деяких ПЦОС для управління кеш-пам'яттю використовуються спеціальні команди або біти управління, які дозволяють програмісту заблокувати вміст кешу в деякій точці виконання програми або заборонити кеш-пам'ять взагалі. Ці можливості дозволяють програмісту здійснювати ручний контроль за кеш-пам'яттю, який в деяких випадках дозволяє підвищити продуктивність ПЦОС.

10.4 Особливості доступу до пам'яті

Доступ до пам'яті з використанням станів очікування. Пам'ять ПЦОС є багаторівневою, що передбачає спільне використання різних за швидкодією, ємністю і вартістю пристроїв пам'яті [2,3]. Ємність пам'яті на кожному із рівнів збільшується в напрямку від процесора, а швидкодія - в протилежному напрямку. Для організації ефективного обміну з повільною зовнішньою пам'яттю в ПЦОС використовуються генератор станів очікування, який керується за допомогою відповідних маніпуляцій з регістрами управління шинами. В даних регістрах є два поля, одне з яких використовується для вибору режиму генерації стану очікування, а друге - для завантаження внутрішнього таймера, який формує внутрішній сигнал готовності. В

процесорі ЦОС TMS320C3x можуть використовуватись чотири режими генерації стану очікування: зовнішній сигнал готовності – RDY, внутрішній сигнал готовності – RDYwtcnt, логічне “І” сигналів - RDY і – RDYwtcnt, логічне “АБО” сигналів - RDY і-RDYwtcnt.

Всі чотири режими можуть використовуватись для генерації внутрішнього сигналу RDYint, який керує доступом. До того часу доки цей сигнал дорівнює “1”, біжучий зовнішній доступ затримується, а при встановленні його в “0”, біжучий доступ завершується.

Окрім звертання до повільної пам'яті стани очікування можуть виникати при спільному використанні шин, при спробі процесора здійснити одночасний доступ до пам'яті, яка не підтримує множинний доступ та при реалізації конвеєра команд. Тривалість стану очікування відносно тривалості командного циклу може змінюватись в діапазоні від четвертини командного циклу до декількох командних циклів. Менша тривалість стану очікування дає змогу мінімізувати час доступу до зовнішньої пам'яті.

Спеціалізовані команди доступу до пам'яті. В більшості ПЦОС використовуються спеціалізовані команди запису даних в пам'ять паралельно до читання команд і даних. Для виконання такого доступу в процесорах ЦОС використовується внутрішній паралелізм, що забезпечується декількома множинами шин адреси і даних. Внутрішній паралелізм підтримується відповідним набором команд, які характерні для алгоритмів ЦОС. До даного набору команд можна віднести такі: одночасне читання пам'яті даних і пам'яті програм, обчислення з читанням з пам'яті, обчислення з записом в пам'ять, обчислення з одночасним читанням даних і команд. Однією з найчастіше використовуваних операцій в ЦОС є операція обчислення суми добутоків, що виконується таким чином: завантажити два операнди, перемножити операнди і додати добуток до суми попередніх добутоків.

Прямий доступ до пам'яті. Прямий доступ до пам'яті (ПДП) - це режим вводу-виводу даних в пам'ять без участі процесора. Для здійснення операцій передачі даних між пам'яттю і пристроями вводу-виводу на кристалі у

більшості ПЦОС розташований контролер ПДП, який дозволяє виконувати дані операції без зниження продуктивності процесора. Переважно контролер ПДП процесорів ЦОС дозволяє звертання до будь-якої комірки адресного простору процесора. Крім цього, такі контролери характеризуються високою швидкодією. Наприклад, контролер ПДП процесора TMS320C4x в комбінації з внутрішньокристаліною пам'яттю дозволяє виконувати на протязі командного циклу одне звертання до пам'яті.

Подальший розвиток ПДП-контролерів процесорів ЦОС направлений на забезпечення декількох паралельних передач. Такі контролери ПДП повинні мати декілька каналів, кожний з яких може забезпечувати обмін і з повільною зовнішньою пам'яттю та пристроями вводу-виводу. Для забезпечення багатоканального обміну в склад контролера повинні входити генератори адреси, регістри управління та лічильники передач. Кількість каналів, які може обслуговувати контролер ПДП в процесорах різних фірм є різною. Наприклад, контролер ПДП процесора TMS320C40 обслуговує 6 каналів, а процесор ADSP2106x - 10 каналів, причому кожен з каналів може використовуватися для обміну типу пам'ять-пам'ять і пам'ять-периферія.

Режими доступу до пам'яті. Для доступу до комірки пам'яті, необхідно спочатку на шині адреси виставити адресу необхідної комірки. Ядро пам'яті являє собою прямокутний масив комірок, кожна з яких визначається своїми координатами у цій прямокутній матриці - рядком і стовпцем. При виборі необхідного рядка і стовпця здійснюється доступ до комірки, що знаходиться на їхньому перетині.

Адреси рядка і стовпця передаються по шині адреси паралельно і повинні бути встановлені до приходу синхроімпульсу. Як тільки він надходить, адреса запам'ятовується у внутрішньому регістрі мікросхеми і декодується, розділяючись на адреси стовпця і рядки. Дані, зчитані з комірки, передаються у вихідний регістр і встановлюються на шині даних, після чого можуть бути зчитані контролером.

Основна відмінність між мікросхемами пам'яті полягає в тому, яким способом дані передаються з комірки на шину даних. Існують чотири фундаментальних режими доступу до пам'яті: послідовний (Flow thru), конвеєрний (Pipeline), регістровий (Register to Latch), пакетний (Burst). За винятком останнього режиму, головна відмінність між цими типами доступу полягає в моменті появи даних на виході стосовно синхроімпульсу. Пакетний режим може використовуватися разом із послідовним і конвеєрним.

10.5 Організація пам'яті комп'ютерних систем цифрової обробки сигналів

Параметри комп'ютерних систем ЦОС в значній мірі визначаються параметрами пам'яті - ємністю, швидкодією та шириною доступу до даних. Задача опису всіх можливих структур є нерозв'язною. Тому доцільним є виділення і дослідження узагальнених базових структур, на основі яких можуть бути синтезовані системи пам'яті для конкретних застосувань. За організацією пам'яті комп'ютерні системи ЦОС можна розділити на три класи: системи із розподіленою локальною пам'яттю; системи із зосередженою спільною пам'яттю; системи із зосереджено-розподіленою пам'яттю [5,9]. Особливістю комп'ютерних систем ЦОС з розподіленою пам'яттю є те, що кожний процесор такої системи має свою локальну пам'ять, доступ до якої є швидким і простим. У таких комп'ютерних системах взаємодія між процесорами є повільною та здійснюється за допомогою передачі повідомлень. В комп'ютерних системах ЦОС з зосередженою загальною пам'яттю взаємодія між процесорами та пам'яттю є складною і здійснюється через систему жорстких або програмованих каналів зв'язку. Найефективнішими за організацією обмінів є комп'ютерні системи ЦОС із зосереджено-розподіленою пам'яттю.

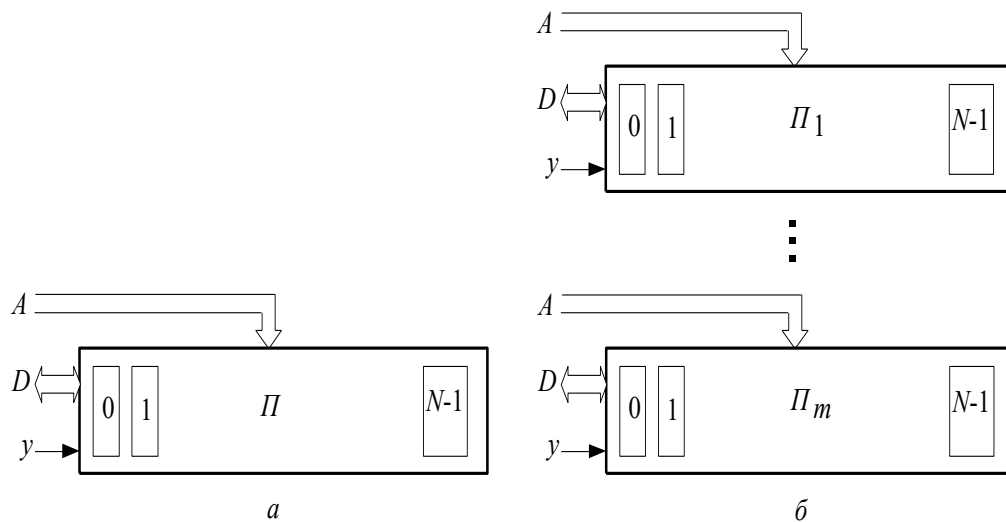


Рис. 10.4. Базові структури пам'яті: а) з послідовним доступом до даних;
б) з незалежним доступом до t елементів даних

Аналіз ширини і часових стратегій доступу та зв'язків пам'яті з процесорами, операційними пристроями і пристроями вводу-виводу дозволив виділити наступні базові структури пам'яті [9]:

- на основі одновимірного з лінійною адресацією блока пам'яті, в якій забезпечується послідовний доступ до елементів даних (рис.10.4 а);
- основою якої є t окремих одновимірних з лінійною адресацією блоків пам'яті, в якій реалізується незалежний доступ до t елементів даних (рис.10.4б);
- на основі одновимірного з лінійною адресацією блока пам'яті, в якій реалізується паралельна адресація і доступ до даних (рис.10.5 а);
- з паралельною адресацією і доступом до даних із загального поля пам'яті, реалізованого на основі t одновимірних з лінійною адресацією блоків пам'яті (рис.10.5 б).

На рис.10.4 і 10.5 наведені базові структури пам'яті комп'ютерних систем ЦОС, де – A , D і y є відповідно шини адреси, даних і управління; Π – пам'ять. В основу організації першої базової структури пам'яті (рис.2.16 а) покладено принципи одновимірності та лінійності адресного простору пам'яті. Для даної пам'яті при звертанні мінімальним неподільним елементом даних є n - розрядне слово.

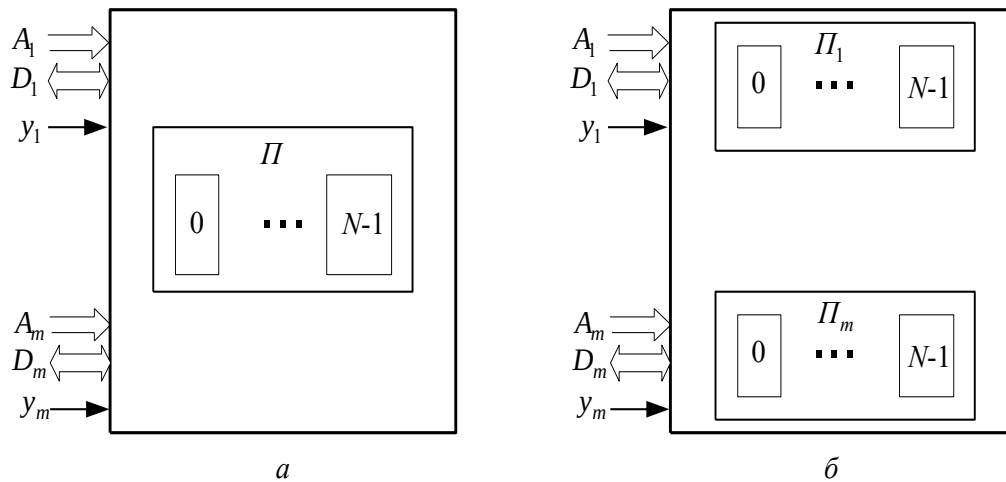


Рис. 10.5 Базові структури пам'яті з паралельним доступом
а) на основі одновимірної пам'яті; б) на основі двовимірної пам'яті.

Друга базова структура пам'яті (рис.10.4 б) реалізується на t окремих блоках пам'яті, кожен з яких має власне запам'ятовуюче середовище з незалежною адресацією. Така структура пам'яті дозволяє реалізувати одну із наступних часових стратегій звертання до блоків: послідовну, з перекриттям у часі та паралельну.

Основою третьої базової структури пам'яті (рис.10.5а) є спільне одновимірне поле пам'яті з лінійною адресацією. Паралельний доступ до множини елементів даних у такій структурі забезпечується шляхом паралельно-послідовного масштабування доступу та часового розподілу ресурсів спільного поля пам'яті.

Особливістю четвертої базові структури пам'яті (рис.10.5б) є двовимірне спільне поля пам'яті, яке реалізується на основі t блоків пам'яті з незалежною адресацією. В такій структурі всі питання, що пов'язані із встановленням фізичного зв'язку та забезпеченням паралельного доступу, вирішуються в самій пам'яті за допомогою комутуючої мережі.

Третя і четверта базові структури пам'яті відповідають вимогам, що висуваються до паралельної пам'яті, оскільки забезпечують паралельний доступ до множини даних.

В більшості випадків вхідні дані в задачах ЦОС мають просторово-часову організацію, яка визначається природою їх отримання. Для вводу і обробки даних з просторово-часовою організацією використовується як пам'ять з послідовним, так і паралельним доступом. Аналіз структур даних, алгоритмів розв'язання задач і архітектур процесорів та систем дозволив сформулювати наступні вимоги до пам'яті КС реального часу [2,3,9]:

- бути адаптованою до інтенсивності надходження потоків даних, структури і алгоритмів обробки даних;
- забезпечувати можливість зміни ширини та часу доступу до пам'яті;
- забезпечувати можливість зміни величини затримки на необхідне число тактів та частоти переключення каналів видачі даних;
- забезпечувати можливість налаштування генераторів адрес на формування необхідної послідовності адрес.

Один з напрямків підвищення швидкодії КС є широке використання паралельної пам'яті на різних рівнях обробки. Створення пам'яті, у якій принципи паралелізму реалізовані на всіх рівнях ієрархії, є недоцільним, що пояснюється кількома причинами. Перша - висока вартість і великі апаратні затрати. Друга причина - відсутність зовнішньої пам'яті паралельного типу. Крім цього, велику роль відіграє і те, що вартість паралельної пам'яті в більшій мірі залежить від паралельності доступу, ніж від ємності.

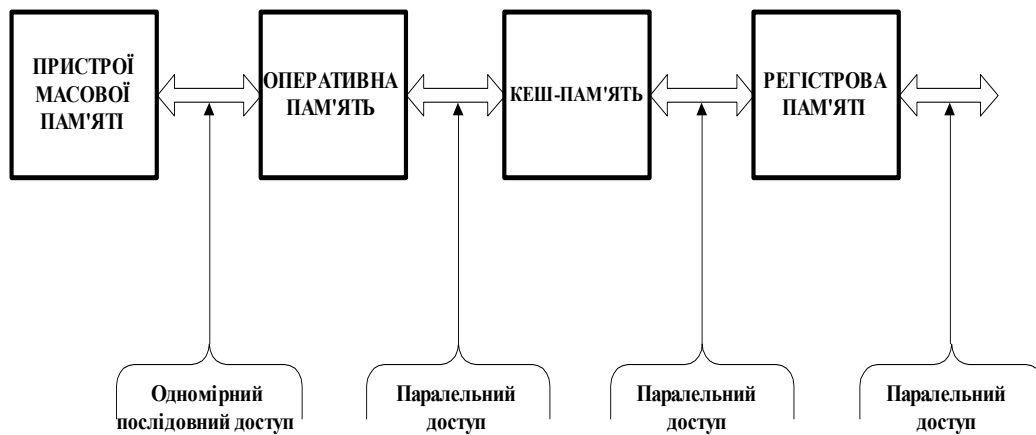


Рис.10.6. Ієрархічна багаторівнева структура пам'яті комп'ютерних систем ЦОС

Раціональним на теперішній час бачиться організація пам'яті систем цифрової обробки сигналів ієрархічною багаторівневою на базі пристроїв з паралельним і послідовним доступом (рис.10.6), причому оперативну пам'ять, кеш і регістрову пам'ять доцільно реалізовувати у вигляді паралельної пам'яті.

10.6 Принципи побудови пам'яті комп'ютерних систем ЦОС

Об'єкти, процеси і явища реального світу, які аналізуються, моделюються або синтезуються на комп'ютерних системах ЦОС у своїй дискретній моделі практично завжди описуються певним чином організованою сукупністю даних. В більшості випадків вхідні дані мають просторово-часову організацію, яка визначається природою їх отримання.

Пам'ять комп'ютерних систем ЦОС повинна відповідати таким вимогам:

- мати організацію, що орієнтована на роботу з векторами і матрицями даних;
- забезпечити одночасне введення вхідного масиву даних і виведення раніше прийнятого масиву, дані в якому розміщені у відповідності до алгоритму розв'язання задачі;

- забезпечувати введення і виведення даних як з однаковою, так і з різною тактовою частотою;
- виконувати функції переставляння та затримки даних на необхідне число тактів;
- забезпечувати багатоканальне введення та виведення даних, причому число входів і виходів може бути різним;
- бути адаптованою до структури даних і специфіки задач, які розв'язуються.

Для забезпечення вищевказаних вимог при проектуванні пам'яті комп'ютерних систем ЦОС необхідно спиратися на наступні принципи:

- ієрархічності і багаторівневості побудови пам'яті великої ємності з широким використанням кешування і механізмів управління сигналами очікування;
 - просторового і функціонального розділення внутрішньої пам'яті з широким використанням різних за швидкодією і шириною доступу модулів пам'яті;
 - багатоканального доступу до пам'яті з метою забезпечення декількох звертань на протязі одного командного циклу;
 - апаратної реалізації складних алгоритмів генерації послідовності адрес з використанням модульної арифметики;
 - гармонічного поєднання можливостей інтегральної технології з розширенням функцій пам'яті, збільшення кількості внутрішніх і зовнішніх каналів доступу та з покращенням параметрів модулів пам'яті (ємності, швидкодії і т.д.).
- Література : [1], [3], [4], [6], [11], [12].

11. Інтерфейси комп'ютерних систем ЦОС

Інтерфейси комп'ютерних систем ЦОС за функціональним призначенням діляться на зовнішні, призначені для введення-виведення даних і зв'язку з ведучим комп'ютером та внутрішні - для з'єднань між процесорними елементами (ПЕ), процесорами і пам'яттю у середині системи. У більшості випадків для зв'язку ПЦОС з ведучим комп'ютером використовується послідовні інтерфейси RS-232C, RS-422A, RS-423A, RS-485 або магістралі (шини) VME, FUSTBUS, Futurebus, SCSI, ISA, PCI, FireWire []. Причому послідовні інтерфейси в основному використовується для зв'язку комп'ютера з простими процесорними модулями (Starter Kit), а шини ISA, PCI - для зв'язку ПЦОС з персональним комп'ютером. Для введення-виведення даних в ПЦОС використовуються АЦП, ЦАП, паралельні, послідовні порти введення-виведення та канали ПДП [2,3,9].

В системах ЦОС для обміну між елементами використовуються мережі обміну, які за топологією зв'язків діляться на три типи: шинні, статичні та динамічні. Мережі з шинною організацією в основному використовуються при реалізації багатопроцесорних систем. Статичні та динамічні мережі забезпечують локальні та глобальні зв'язки між елементами системи. Особливістю статичних мереж є жорстко фіксовані з'єднання, а динамічних – можливість підключення різних виходів до одного і того ж входу.

В систолічних процесорах часто для забезпечення ефективного обміну між процесорними елементами використовуються лінки (комутаційні порти), в яких управління передачею даних здійснюється шляхом запису відповідних кодів в регістри стану і керування. При цьому у процесі передачі використовується буферизація даних, які передаються і приймаються.

Ефективність систем ЦОС в значній мірі визначається інтерфейсами зовнішньої пам'яті ПЦОС, які можна розділити за трьома основними ознаками: кількістю портів пам'яті, інтелектуальністю і гнучкістю інтерфейсу та часовими параметрами. Більшість ПЦОС використовують один

зовнішній порт пам'яті, який складається з шин: адреси, даних і управління. Інтерфейс зовнішньої пам'яті для ПЦОС з багатошинною і багатомодульною пам'яттю повинен забезпечувати їх зовнішнє розширення. Приклад типового інтерфейсу зовнішньої пам'яті для ПЦОС з трьома незалежними множинами шин зображений на рис.11.1. Даний інтерфейс дозволяє нарощувати пам'ять програм і даних шляхом підключення до ПЦОС зовнішніх модулів пам'яті та комутацією на його лінії однієї із трьох внутрішніх множин шин.

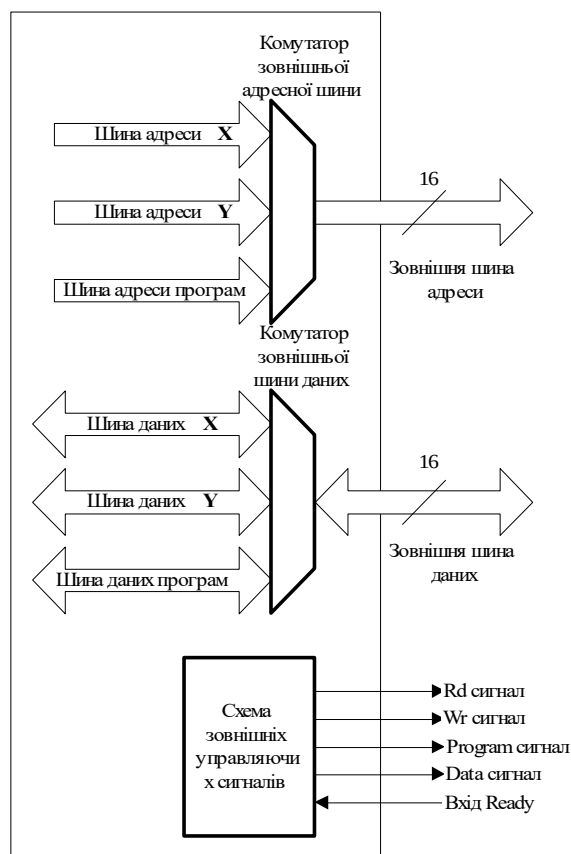


Рис.11.1 Типовий інтерфейс зовнішньої пам'яті

Недоліком такого інтерфейсу є неможливість множинного доступу до модулів зовнішньої пам'яті на протязі одного командного циклу, що веде до зменшення продуктивності ПЦОС при виконанні програм з доступом до зовнішньої пам'яті. Забезпечення паралельного доступу до зовнішньої пам'яті протягом одного командного циклу можливе шляхом збільшення кількості портів зовнішньої пам'яті.

Збільшення кількості портів зовнішньої пам'яті вимагає збільшення вартості ПЦОС. Прикладом процесорів ЦОС з двома зовнішніми портами є процесори ADSP-21020 (Analog Devices), DSP 96002 (Motorola), TMS 320C30 і TMS 320C40 (Texas Instruments), де один із портів є портом зовнішньої глобальної, а другий - локальної пам'яті. Шини кожного з портів можуть використовуватися для звертання до зовнішньої пам'яті програм і даних з синхронним або асинхронним доступом.

Гнучкість та інтелектуальність інтерфейсів зовнішньої пам'яті визначається кількістю і функціями сигналів шини управління, які забезпечують можливість роботи з широким діапазоном пристроїв зовнішньої пам'яті без спеціального обладнання. Деякими характерними властивостями інтерфейсів зовнішньої пам'яті є можливість програмування станів очікування, та підтримка сторінкового режиму динамічної пам'яті і її регенерації.

Часові характеристики інтерфейсів зовнішньої пам'яті значно впливають на загальну продуктивність та вартість системи. Порівняння часових характеристик інтерфейсів зовнішньої пам'яті ПЦОС необхідно здійснювати з врахуванням тривалості командного циклу. При однаковій тривалості командного циклу перевагу необхідно віддавати ПЦОС з більшим часом доступу до зовнішньої пам'яті. Використання в системах повільної зовнішньої пам'яті робить їх дешевшими.

В більшості сучасних ПЦОС в інтерфейсах зовнішньої пам'яті закладено спеціальні можливості для підтримки функції багатопроцесорності складних обчислювальних систем. До них в першу чергу слід віднести наявність двох портів зовнішньої пам'яті. В багатопроцесорних застосуваннях один з портів доцільно використовувати для підключення до локальної пам'яті процесора, а другий - для доступу до глобальної пам'яті, яка спільно використовується всіма процесорами системи. Можливі випадки коли декільком процесорам одночасно необхідно звернутися до глобальної пам'яті. Але магістраль глобальної пам'яті є одна і використовувати її в

кожний окремий момент часу може тільки один процесор. Тому виникає необхідність в механізмі управління використанням магістралі процесорами, тобто в функції арбітражу, при якому вибраному процесорові надається право доступу до магістралі.

Існують два основних способи арбітражу: паралельний і послідовний. Паралельний спосіб вимагає використання додаткових засобів для арбітражу магістралі, послідовний реалізується простим з'єднанням відповідних виводів процесорів без використання спеціального програмного або апаратного забезпечення. Для підтримки систем з спільною магістраллю в більшості ПЦОС є два контакти які можуть бути зконфігуровані як сигнали запиту і підтвердження надання магістралі. Коли один з процесорів хоче мати доступ до магістралі, то він формує сигнал запиту магістралі і передає їх арбітрові. Останній за певним алгоритмом вибирає процесор, якому надається право доступу до магістралі, про що він повідомляється сигналом дозволу на заняття магістралі.

Крім того, деякі ПЦОС, наприклад TMS320C50, дозволяють зовнішньому пристрою звертатись до внутрішньокристалльної пам'яті. Для забезпечення такого звертання використовуються дві пари сигналів: перша пара - HOLD (запит управління магістраллю) і HOLDA (підтвердження надання магістралі), друга пара - BR (запит на доступ до внутрішньокристалльної пам'яті) і IAQ (дозвіл доступу до внутрішньокристалльної пам'яті). При цьому для доступу до внутрішньокристалльної пам'яті необхідно забезпечити спершу доступ до зовнішньої магістралі за допомогою сигналів HOLD і HOLDA, а потім за допомогою сигналів BR і IAQ отримати доступ до внутрішньокристалльної пам'яті. Використання механізму доступу до внутрішньокристалльної пам'яті в багатопроцесорних системах дозволяє мінімізувати час міжпроцесорного обміну.

Ще один механізм підтримки функції багатопроцесорності, що спрощує використання спільних змінних в глобальній пам'яті закладено в

ПЦОС. Даний механізм пов'язаний з блокуванням шини (bus locking) на час читання, зміни та запису нового значення в пам'ять. Іноді цю можливість називають елементарною операцією перевірки-встановлення (test-and-set). В процесорах TMS320C3x і TMS320C4x для виконання даної процедури використовуються апаратна підтримка і спеціальні команди.

Інтерфейс процесора фірми Analog Devices (ADSP-2106x) орієнтований на побудову багатопроцесорних систем з спільною магістраллю. В даних ПЦОС арбітр магістралі реалізований на кристалі і дозволяє об'єднувати в систему до шести процесорів без додаткових зовнішніх апаратних засобів і спеціального програмного забезпечення. Крім того, процесор ADSP-2106x забезпечує доступ до внутрішньокристалічної пам'яті. Це дозволяє виконувати безпосередньо міжпроцесорний обмін без додаткових операцій.

Для побудови багатопроцесорних систем з різною конфігурацією зв'язків в архітектуру процесора TMS320C4x і ADSP-2106x введені спеціальні комунікаційні порти.

Для налагодження й емуляції систем ЦОС використовується інтерфейс JTAG/IEEE 1149.1 (Join Test Action Group). Цей інтерфейс, підтримуваний багатьма фірмами-виробниками систем ЦОС забезпечує покрокове налагодження як одного, так і декількох процесорів у системі з використанням спеціальних програм - відлагоджувачів і відповідного апаратного забезпечення.

Література : [2], [3], [5], [8], [9], [10].

12. Узагальнена структура та шляхи вдосконалення характеристик комп'ютерних систем обробки сигналів

Специфіка задач та алгоритмів обробки сигналів і зображень накладає певні особливості на архітектуру комп'ютерних систем, відрізняючи їх від архітектури більшості універсальних систем. Аналіз архітектури таких комп'ютерних систем [9] дозволив виділити їх основні компоненти. Такими компонентами є: процесорне ядро, підсистеми введення/виведення та зберігання. На основі цих компонентів синтезовано узагальнену структуру комп'ютерної системи обробки сигналів і зображень (рис.12.1).



Рис. 12.1 Узагальнена структура комп'ютерної системи обробки сигналів

Структура процесорного ядра відображає виділений операційний базис і враховує особливості алгоритмів ЦОС і зображень. Основними структурними одиницями процесорного ядра є: операційні пристрої, блоки керування, відлагодження та контролю. Для забезпечення обробки сигналів і зображень у реальному часі за широким набором алгоритмів різних за інтенсивністю надходження потоків даних в склад операційних пристроїв повинні входити як універсальні арифметично-логічні пристрої (АЛП), так процесорні елементи (ПЕ), алгоритмічні процесори швидких ортогональних тригонометричних перетворень і нейроакселератори. Більшість операційних пристроїв процесорного ядра працюють з числами фіксованою (ФК), а для задач, що вимагають високу точність обчислення, використовуються операційні пристрої з плаваючою крапкою (ПК).

Один з шляхів зменшення часу доступу до пам'яті є використання генераторів адреси, які генерують необхідну послідовність адрес. В комп'ютерних системах обробки сигналів і зображень для генерації адрес можуть використовуватися два незалежних генератори, які забезпечують одночасну видачу адреси для пам'яті програм і даних. Такі генератори адрес забезпечують непряму адресацію з автоматичною модифікацією адрес.

Особливістю більшості алгоритмів обробки сигналів і зображень є необхідність перестановки елементів вхідного або вихідного масивів даних. Ця процедура може бути здійснена шляхом використання генератора адреси даних. Так у мікропроцесорі TMS320C3x для зменшення часу формування послідовності адреси для алгоритмів швидкого перетворення Фур'є використовується генератор адреси даних. Цей блок дозволяє програмним шляхом налаштовуватися на генерацію біт-реверсивних адрес для різних за розміром і основою швидких перетворень Фур'є.

Одним з варіантів мінімізації часових витрат при цифровій фільтрації є реалізація лінії затримки без переміщення даних в пам'яті. Ефект зсуву даних в лінії затримки моделюється за допомогою використання модульної арифметики. В цьому випадку наступна адреса на виході генератора

обчислюється за формулою

$$A_{i+1}=(A_i+M-A_b)\text{mod}(L)+A_b$$

де A_{i+1} - наступна адреса; A_i - біжуча адреса; M - значення модифікації; A_b - базова адреса; L - тривалість лінії затримки. Використання модульної адресації дозволяє збільшити число одночасних звертань до пам'яті протягом одного командного циклу.

Включення в склад процесорного ядра синтезатора тактової частоти дає можливість регулювати продуктивність і споживану потужність.

Вартість і час проектування комп'ютерних систем обробки сигналів і зображень в значній мірі визначається апаратними засобами відлагодження і контролю. В процесорне ядро введений стандартний набір таких засобів, в який входять сторожовий таймер (WDT), апаратура для внутрішньосхемної емуляції (OnCE), тестування та відлагодження (JTAG).

Підсистема введення/виведення включає перетворювачі (АЦП і ЦАП), паралельні і послідовні порти вводу/виводу, host-ітерфейс для зв'язку з персональним комп'ютером чи іншим комп'ютером, широтно-імпульсний модулятор, таймери, контролери прямого доступу до пам'яті (ПДП) та інші пристрої. Для здійснення операцій передачі даних між пам'яттю і пристроями вводу-виводу в комп'ютерних систем обробки сигналів і зображень використовуються контролери ПДП, які дозволяють виконувати дані операції без зниження продуктивності процесорів. Такі контролери забезпечують звертання до будь-якої комірки адресного простору процесорів і характеризуються високою швидкістю.

Подальший розвиток контролерів ПДП направлений на забезпечення декількох паралельних передач як із зовнішньою пам'яттю, так пристроями вводу-виводу. Для забезпечення багатоканального обміну в склад контролера повинні входити генератори адреси, регістри управління та лічильники передач.

Організація підсистеми зберігання у комп'ютерних систем обробки сигналів і зображень відображає специфіку даних і алгоритмів обробки сигналів і зображень, забезпечує швидкісний обмін з пристроями введення/виведення та операційними. Збільшити кількість звертань до пам'яті на протязі одного циклу виконання команди дозволяє багатомодульна структура пам'яті з двома або більше шинами адреси і даних. Така організація пам'яті характерна для більшості комп'ютерних систем обробки сигналів і зображень [1]. Основною перевагою такої організації є можливість виконання двох звертань до пам'яті протягом одного циклу виконання команди. Крім цього, таке розділення пам'яті дозволяє сумістити в часі виклик команд і їх виконання.

Подальшим розвитком даної архітектури є модифікована гарвардська архітектура, яка дозволяє обмін інформацією між пам'яттю даних і пам'яттю команд. Ця модифікація забезпечує можливість зчитування в пам'ять даних коефіцієнтів і констант, записаних в пам'яті програм і тим самим виключає використання спеціальних постійних запам'ятовуючих пристроїв. Вона також забезпечує можливість використання команд з безпосередньою адресацією та звертання до підпрограм за результатами обчислень. З метою скорочення довжини командного циклу у процесорах обробки сигналів і зображень гарвардська архітектура доповнюється конвеєрним режимом роботи. В залежності від типу процесора конвеєр може мати від двох до чотирьох сходинок. Це означає, що процесор може одночасно опрацьовувати від двох до чотирьох команд, при чому кожна з команд буде знаходитись на різних етапах виконання.

Покращення характеристик пам'яті спрямовано на досягнення балансу між продуктивністю комп'ютерних систем обробки сигналів і зображень та їх вартістю. Серед множини способів покращення характеристик пам'яті в комп'ютерних систем обробки сигналів і зображень найчастіше використовуються наступні: кешування пам'яті, генерація адрес пам'яті, динамічне встановлення часу очікування, спеціалізовані операції звертання до пам'яті.

Широке використання комп'ютерних систем обробки сигналів і зображень в різних областях науки, техніки і виробництва вимагають від них високих технічних характеристик [5,9]. Однією з найбільш широко розповсюджених вимог, що ставиться до комп'ютерних систем обробки сигналів і зображень є забезпечення високої швидкодії. Така проблема, як правило, виникає при обробки сигналів і зображень у реальному часі. Це накладає обмеження на час розв'язання задачі T_p , який не повинен перевищувати часу обміну повідомленнями $T_{обм}$, тобто

$$T_p \leq T_{обм}$$

Час обміну залежить як від кількості вхідних даних N і їх розрядності n , так і від кількості K , розрядності n_k та частоти F_d надходження даних у каналах. Даний час визначається за формулою

$$T_{обм} = \frac{Nn}{F_d K n_k}$$

Для забезпечення обробки i -го потоку даних в реальному часі за алгоритмами з складністю R_i продуктивність комп'ютерної системи повинна бути

$$P_i \geq \frac{R_i F_d K_i n_{ki}}{N_i n_i}$$

де N_i -величина і n_i – розрядність i -го вхідного масиву даних; K_i – кількість каналів і їх n_{ki} – розрядність при вводі i -го потоку даних. Сумарна продуктивність апаратних засобів ЦОС і зображень, яка необхідна для обробки m потоків даних у реальному часі визначається так:

$$P = \sum_{i=1}^m P_i$$

Кількість операцій, яка необхідно для реалізації алгоритму, визначає його складність R_i . Так, наприклад, складність алгоритмів фільтрації, обчислення згорток, кореляційних функцій, дискретного перетворення Фур'є приблизно дорівнює N^2 . Нові методи обчислень, які базуються на матричних операціях, алгоритмах розв'язання систем лінійних рівнянь, апроксимації за методом найменших квадратів мають складність N^3 [9].

Застосування комп'ютерних систем в апаратурі передачі даних, в контрольно-вимірювальній техніці, в радіолокаційних і гідролокаційних системах та інших областях, де апаратура є бортовою, тобто такою, що возиться, носить, літає і плаває на відповідних механізмах, накладає жорсткі обмеження на енергоспоживання та масогабаритні характеристики. Конструктивні обмеження при наявній тенденції до росту продуктивності відображаються на архітектурній організації комп'ютерних систем, способах з'єднання з зовнішньою апаратурою та можливістю нарощування обчислювальних ресурсів.

Необхідність задоволення вимог забезпечення масогабаритних характеристик, енергоспоживання, вартості змушує проектувати комп'ютерні системи під заданий клас задач. Крім того, до апаратури ЦОС ставляться високі вимоги за надійністю та забезпеченням перевірки працездатності, швидкої локалізації і знешкодження несправності. Проблема високої надійності апаратних засобів технологій ЦОС виникає при використанні їх в системах управління особливо відповідальними об'єктами, розміщеними на великій відстані від людини або об'єктами, що мають великий зовнішній вплив. Забезпечення високої надійності та діагностики апаратних засобів технологій ЦОС досягається за рахунок НВІС-реалізацій та апаратної і програмної надлишковості.

Не дивлячись на досягнуті успіхи в області автоматизації проектування апаратних засобів ЦОС одним з найтрудомікших етапів у їх створенні є етап відлагодження. Для відлагодження засобів ЦОС вони повинні володіти властивостями керованості, спостережуваності і передбачуваності.

Вдосконалення характеристик комп'ютерних систем обробки сигналів і зображень можна досягнути як на рівні схемотехнічних, так і архітектурних рішень. На схемотехнічному рівні можна виділити наступні основні напрямки вдосконалення:

- перехід на НВІС-технологію, застосування спеціалізованих замовних і напівзамовних НВІС, нових високопродуктивних одно кристальних мікро-ЕОМ, мікропроцесорів, трансп'ютерів, мікроконтролерів ;
- інтеграція на кристалі НВІС як можна більше функцій для опрацювання та зберігання даних, а також інтерфейсу для зв'язку з іншими елементами системи;
- апаратна реалізація базових перетворень і базових операцій алгоритмів обробки сигналів і зображень;
- просторове та функціональне розділення внутрішньої пам'яті з широким використанням різних за швидкодією і шириною доступу модулів пам'яті;
- багатоканальний доступ до пам'яті з метою забезпечення декількох звертань на протязі одного командного циклу;
- апаратна реалізації складних алгоритмів генерації послідовності адрес з використанням модульної арифметики;
- гармонійне поєднання можливостей інтегральної технології з розширенням функцій пам'яті, збільшення кількості внутрішніх і зовнішніх каналів доступу та покращення параметрів модулів пам'яті (ємності, швидкодії і т.д.);
- широкого впровадження сучасних методів доступу до даних в пристроях пам'яті - конвеєрний, регістровий, пакетний режими.

На архітектурному рівні вдосконалення характеристик комп'ютерних систем необхідно проводити в наступних напрямках:

- адаптації програмно-апаратних засобів до алгоритмів розв'язання задач;
- забезпечення регулярності, модульності архітектури комп'ютерних систем і широкого використання стандартних елементів;

- локалізації і спрощені зв'язків між елементами системи та використанні багатоглибини організації;
- збільшення пропускної здатності каналів вводу-виводу і розширення їх функцій;
- скорочення кількості внутрішньосистемних пересилань команд, даних і забезпечення балансу між введенням-виведенням та обчисленнями;
- суміщення в часі процесів функціонування максимальної кількості елементів системи;
- використання системи макрокоманд і функціональних розширювачів;
- заміна програмних модулів апаратними;
- розширення використання паралельної і конвеєрної обробки та децентралізації обчислень;
- використання паралельної пам'яті для зовнішнього та внутрішнього обміну;
- адаптація пам'яті до структури даних і специфіки задач, які розв'язують, та забезпечення виконання функцій переставлення, затримки даних на необхідне число тактів.
- Література : [1], [5], [8], [9], [10].

13. Засоби реалізації алгоритмів і підходи до побудови комп'ютерних систем обробки сигналів

При комп'ютерній обробці сигналів і зображень у реальному часі можуть використовуватися програмні, мікропрограмні або апаратні засоби реалізації алгоритмів. Основними вимогами при виборі засобів реалізації алгоритмів є забезпечення реального часу при високій ефективності використання обладнання.

При програмній реалізації алгоритмів обробки сигналів і зображень обчислювальні процеси переважно розгортаються в часі з великим об'ємом пересилок інформації між оперативною пам'яттю і операційними пристроями. Програмні засоби є доступними для програміста, перед яким виникає задача мінімізації об'єму програм і часу реалізації при заданій точності обчислень. Вказані засоби характеризуються низькою швидкістю і гнучкістю з точки зору можливості модифікації та заміни алгоритмів.

Мікропрограмна реалізація алгоритмів обробки сигналів і зображень передбачає їхнє розгортання як в часі, так і в просторі. При мікропрограмуванні є доступ до системи мікропрограм процесора, що забезпечується застосуванням постійної пам'яті, програмованих логічних матриць, а також оперативних запам'ятовуючих пристроїв, які використовуються в якості пам'яті мікропрограм. Прикладом мікропрограмної реалізації є реалізація алгоритмів обробки сигналів і зображень на базі однорідних обчислювальних середовищ. Процесор на базі однорідних обчислювальних середовищ - це двовимірне регулярне поле процесорних елементів, кожен з яких фізично зв'язаний з чотирма сусідами – зверху, знизу, зліва та справа [5]. Кожний процесорний елемент може виконувати набір бітових операцій перетворення інформації з вхідних каналів у вихідні. Процесор на базі однорідних обчислювальних середовищ є універсальною системою, тобто в ньому можливо реалізувати довільну обчислювальну функцію. Бітовий рівень процесорного елемента та

повна система комутації дозволять реалізувати розпаралелення обчислень на найнижчому бітовому рівні. Це є суттєвою перевагою мікропрограмної реалізації алгоритмів обробки сигналів і зображень. Реалізація в повній мірі потенціальних можливостей мікропрограмних засобів може бути досягнута лише при глибокому вивченні як задачі, яка розв'язується, так і внутрішньої мови процесора. Мікропрограмні засобами реалізації алгоритмів обробки сигналів і зображень в порівнянні з програмними є більш швидкодіючими.

Успіхи в області інтегральної технології дозволяють все більше перекладати реалізацію алгоритмів обробки сигналів і зображень на апаратні засоби, які розгортають обчислення не тільки у часі, а і в просторі. Такі обчислення характеризуються введенням додаткового обладнання і відсутністю проміжних пересилок інформації в процесі обчислення, а також спрощенням функції місцевого управління. В основі структурної організації апаратних засобів лежить принцип адекватного апаратного відображення поточкових графів алгоритмів обробки сигналів і зображень на апаратні пристрої, які реалізують функціональні оператори та з'єднані між собою відповідно з поточковим графом алгоритмом [5,9]. Синтезовані за таким принципом структури є алгоритмічними. В алгоритмічних структурах алгоритм виконується над вхідними даними при їх одноразовому проходженні через всі апаратні пристрої. За режимами роботи такі структури діляться на синхронні та асинхронні. В асинхронних (однотактних) структурах обробка даних здійснюється без проміжних запам'ятовувань. Продуктивність таких структур визначається часом виконання найскладніших функціональних операторів Φ_{ij} алгоритму, які лежать на шляху проходження даних. Кожна однотактна структура є послідовною з точки зору реалізації функціональних операторів Φ_{ij} . Це є причиною обмеженої швидкодії та неефективного використання обладнання при обробці інтенсивних потоків даних у реальному часі. Тому для обробки потоків даних доцільно використовувати синхронні структури з конвеєрною реалізацією графів алгоритмів, яка передбачає суміщення у часі виконання

функціональних операторів алгоритму над різними даними. Апаратні засоби дозволяють з максимальною швидкістю реалізовувати алгоритми обробки сигналів і зображень, скоротити витрати на створення та експлуатацію програм і є доступними тільки розробникам архітектури апаратних засобів.

Потрібно відзначити, що всі види реалізації алгоритмів в безпосередньому вигляді зустрічаються досить рідко. На практиці в більшості випадків для реалізації алгоритмів використовуються комбіновані підходи з переважанням одного з перерахованих. Перевага того чи іншого виду засобів реалізації алгоритмів обробки сигналів і зображень визначається вимогами, які висуваються до системи за швидкістю, габаритами та споживаною потужністю.

Умови експлуатації, інтенсивність поступлення даних, складність алгоритмів, за якими здійснюється обробка, та обмеження, що висуваються до габаритів, ваги і споживаної потужності, визначають підходи до проектування комп'ютерних засобів обробки сигналів і зображень у реальному часі [9]. Із множини існуючих підходів до створення комп'ютерних засобів обробки сигналів і зображень у реальному часі розглянемо наступні :

- на основі універсальних і функціонально-орієнтованих процесорів обробки сигналів шляхом розробки спеціалізованого програмного забезпечення;
- процесорного ядра орієнтованого на задачі обробки сигналів і зображень доповненого апаратними розширювачами, які реалізують найскладніші базові операції алгоритмів;
- у вигляді спеціалізованої комп'ютерної системи, архітектура та організація обчислювального процесу в якій відображає структуру алгоритму розв'язання задачі.

Підхід до створення комп'ютерних систем обробки сигналів і зображень у реальному часі на основі універсальних і функціонально-орієнтованих процесорів обробки сигналів шляхом розробки відповідного

програмного забезпечення є доступними для широкого кола користувачів. Крім того, дуже суттєвою перевагою цього підходу є можливість використання раніше розроблених програм. При розв'язанні конкретної задачі обробки сигналів або зображень архітектура такої комп'ютерної системи буде надлишковою як у функціональному, так і структурному відношеннях. Такий підхід до створення комп'ютерних систем обробки сигналів і зображень у реальному часі дозволяє опрацьовувати потоки даних з невисокою інтенсивністю надходження.

Підхід до створення комп'ютерних систем обробки сигналів і зображень у реальному часі на базі процесорного ядра доповненого апаратними розширювачами є перспективним, так як він передбачає сполучення універсальних і спеціальних засобів. Процес взаємопроникнення універсального та спеціального, програмного і апаратного дозволяє створити комп'ютерних систем, які ефективно реалізують алгоритми обробки сигналів і зображень та мають широкий діапазон областей застосування. Цей підхід передбачає розробку, відпрацьовування та передачу конструкторської документації на процесорне ядро замовнику з наданням йому можливостей доповнення процесорного ядра необхідними додатковими функціональними вузлами. Звичайно процесорне ядро є завершеним проектом, який конструктивно може бути реалізований у вигляді модуля або спеціалізованої напівзамовної НВІС. За кордоном такий підхід дістав назву core-технологія (core - ядро, серцевина) [2,3,6,9]. В склад комп'ютерних систем, крім процесорного ядра, може входити інтерфейсна логіка, зовнішні та спеціалізовані операційні пристрої, пам'ять і т.д. При такому підході розробка комп'ютерної системи з заданими технічними параметрами зводиться до доповнення процесорного ядра необхідним обрамленням і розробки спеціалізованого програмного забезпечення для розв'язання поставленої задачі.

Спеціалізований підхід до створення комп'ютерних систем реального часу дозволяє досягнути високої продуктивності та є альтернативою вище

розглянутим підходам. Для забезпечення обробки потоків даних у реальному часі при високій ефективності використання обладнання необхідно узгодити обчислювану здатність комп'ютерної системи з інтенсивністю надходження потоків даних. Таке узгодження досягається вибором необхідної кількості трактів обробки m і їх розрядності n або частоти роботи конвеєра F_k . При створенні орієнтованих на НВІС-реалізації спеціалізованих комп'ютерних систем реального часу з високою ефективністю використання обладнання необхідно використовувати інтегрований підхід, який охоплює методи і алгоритми обробки сигналів і зображень, архітектури спеціалізованих засобів, НВІС-технологію та особливості застосувань. На сучасному етапі розвитку НВІС-технології комп'ютерні системи обробки сигналів і зображень у реальному часі доцільно реалізовувати на обчислювальних полях, які є двовимірними регулярними матрицями програмованих користувачем логічних інтегральних схем (ПЛІС), кожна з яких фізично зв'язана за входами-виходами з чотирма сусідами – зверху, знизу, зліва та справа. Використання для побудови обчислювальних полів ПЛІС з динамічним репрограмуванням відкриває нові можливості апаратної реалізації алгоритмів ШІС. Можливості оперативної реконфігурації елементів обчислювального поля дозволяє побудову комп'ютерних систем з багатофункціональним використанням апаратних ресурсів для розв'язання складних задач при їх розбитті на послідовності підзадач, які реалізуються на одному і тому обчислювальному полі, яке оперативно переналаштовується на реалізацію конкретної підзадачі.

Література : [1], [2], [3], [5], [6], [9].

14. Базові структури комп'ютерних систем цифрової обробки сигналів

14.1 Універсальна комп'ютерна система ЦОС.

14.2 Спеціалізовані комп'ютерні системи ЦОС.

14.3 Реконфігуровані комп'ютерні системи ЦОС.

Структури комп'ютерних систем ЦОС в основному є відображенням структури алгоритмів розв'язання задачі на апаратно-програмні засоби [9]. В залежності від складності алгоритмів і вимог застосування вибираються засоби реалізації та ступінь деталізації відображення алгоритмів, яка визначається як структурними одиницями інформації, так і складністю виконуваних операцій. Основними засобами реалізації функціональних операторів в комп'ютерних системах ЦОС є алгоритмічні процесори (АП) та функціонально-орієнтовані процесори (ФОП), які реалізуються на базі мікропроцесорів ЦОС. Використання буферної паралельної пам'яті (БПП) в комп'ютерних системах ЦОС зменшує час обміну між процесорами та зводить до мінімуму проблеми, що зв'язані з управлінням системою, упорядкуванням, затримкою та комутацією потоків даних. Особливістю комп'ютерних систем ЦОС з обміном через паралельну пам'ять є можливість програмувати канали зв'язку між процесорами, що з'єднанні з нею.

В залежності від вимог застосування, частоти зміни задач, алгоритму їх розв'язання, розміру та частоти надходження вхідних даних можуть бути синтезовані різні структури комп'ютерних систем ЦОС, які відрізняються як організацією обчислень, так і технічними параметрами. Задача опису всіх можливих структур є нерозв'язною. Тому доцільним є виділення і дослідження узагальнених базових структур високопродуктивних комп'ютерних систем ЦОС, на основі яких можуть бути синтезовані комп'ютерні системи для розв'язання конкретних задач з потрібними параметрами.

14.1 Універсальна комп'ютерна система ЦОС

Комп'ютерна система ЦОС складаються із ядра, обчислюваного поля та підсистем зберігання і введення-виведення. Ядро системи складається із комп'ютера та розширювача інтерфейсу (PI), забезпечує управління обчислювальним процесом, відображення результатів обробки і при цьому

використовує стандартні апаратно-програмні засоби.

Обчислювальне поле, яке складається з множини програмованих і алгоритмічних процесорів обробки сигналів, забезпечує необхідну швидкодію та налаштування на реалізацію розв'язання конкретної задачі. Структура універсальної комп'ютерної системи ЦОС наведена на рис.14.1.

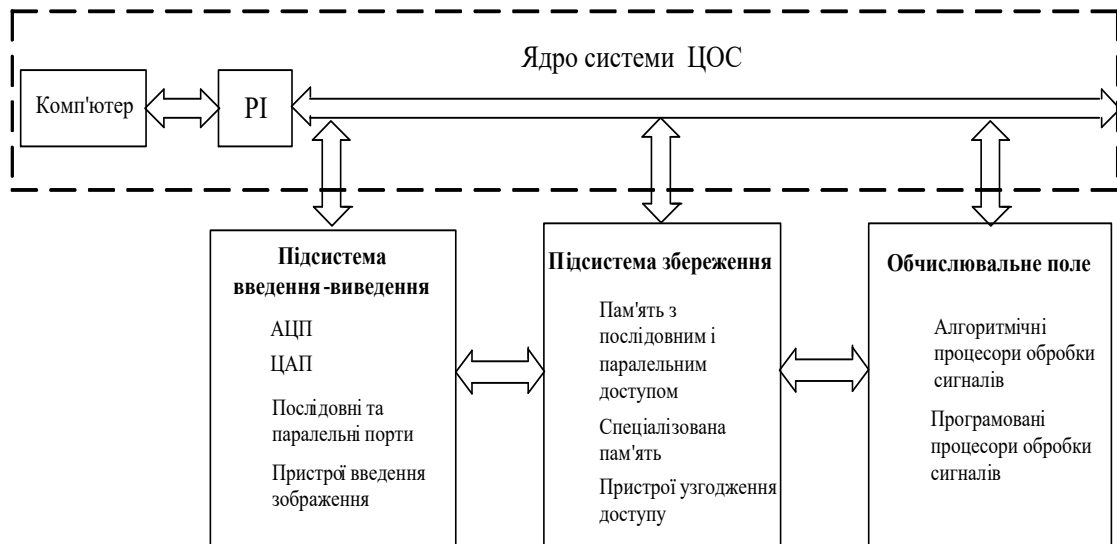


Рис 14.1 Структура універсальної комп'ютерної системи ЦОС

В універсальній комп'ютерній системі ЦОС використовуються два підходи, які полягають в застосуванні та оптимальному поєднанні універсальних та спеціалізованих засобів.

14.2 Спеціалізовані комп'ютерні системи ЦОС

Матричні асинхронні комп'ютерні системи ЦОС. В основі структурної організації комп'ютерних систем ЦОС лежить матрична мережа, яка є двовимірною сукупністю процесорів, які апаратно-програмно відображають конкретизований потоковий граф алгоритму розв'язання задачі.

Асинхронні комп'ютерні системи ЦОС це системи, в яких обмін між процесорами здійснюється за готовністю проміжних результатів обчислень, відсутня глобальна синхронізація та результати обчислень отримуються асинхронно []. Особливістю більшості асинхронних комп'ютерних систем ЦОС є обчислення, що керуються потоком даних.

Базова структура асинхронної алгоритмічної комп'ютерної системи

ЦОС на основі масиву АП= $\{АП_1, АП_2, \dots, АП_p\}$ наведена на рис.14.2, де ПК – пристрій керування; БПП_{Вх} і БПП_{Вих} – вхідна і вихідна буферна паралельна пам'ять.

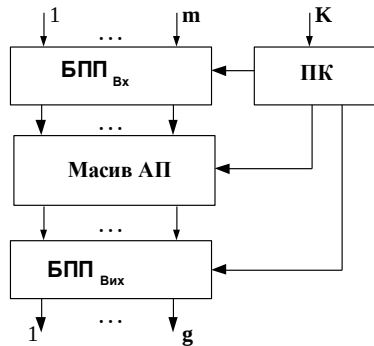


Рис.14.2 Базова структура асинхронної комп'ютерних систем ЦОС комп'ютерних систем ЦОС

Перед початком роботи такої комп'ютерної системи ЦОС здійснюється налаштування АП на виконання відповідних макрооперацій, виконується необхідна комутація між АП та проводиться програмування генераторів адрес вхідної і вихідної БПП на формування необхідної послідовності адреси.

Швидкодія асинхронної комп'ютерної системи ЦОС на базі АП визначається часом обчислення, який дорівнює

$$T_{Аскс} = \frac{N}{m} \sum_{j=1}^n \max_k t_{jk} + t_{3nBx} + t_{qmBx} + t_{3nP}$$

де N – розмір масиву даних; m- кількість трактів опрацювання; t_{jk} - час виконання функціональних макрооператорів Φ_{jk} , які лежать на найдовшому шляху виконання алгоритму; n-кількість ярусів алгоритму; t_{3nBx} , t_{qmBx} – часи відповідно запису та читання вхідних даних, t_{3nP} – час запису результатів.

Особливістю асинхронної комп'ютерні системи ЦОС на базі масиву АП є обмежений клас задач. Така комп'ютерна система ЦОС є критичною до зміни або модифікації алгоритмів розв'язання задач, вона вимагає повного їх відпрацювання.

Асинхронна комп'ютерна система ЦОС забезпечує опрацювання в реальному часі неперервного потоку даних, що надходять з інтенсивністю

$$P_d = m / \sum_{j=1}^n \max_k t_{jk}$$

Гнучкішою є асинхронна комп'ютерна система ЦОС, в якій замість масиву АП використовується масив ФОП, які з'єднані між собою у відповідності з графом алгоритмом розв'язання задачі. Для реалізації такої комп'ютерної системи ЦОС алгоритм представляється конкретизованим потоковим графом, функціональні оператори якого є складними макроопераціями, що орієнтовані на реалізацію ФОП. Перспективою є структура з множиною ФОП які з'єднані тільки з БПП_{Вх} і БПП_{Вих}. В такій комп'ютерній системі ЦОС за рахунок використання БПП можна програмувати зв'язки між процесорами відповідно з потоковим графом розв'язання конкретної задачі. Заміна алгоритму розв'язання задачі в комп'ютерній системі ЦОС на базі ФОП здійснюється шляхом перезапису пам'яті програм процесорів.

Паралельно-потоківі комп'ютерні системи ЦОС. В паралельно-потоківих комп'ютерних системах ЦОС опрацювання даних здійснюється за конвеєрним принципом. Конвеєризація комп'ютерних систем ЦОС передбачає розділення масиву процесорів на сходинки шляхом введення буферної паралельної пам'яті. При цьому кожна j-а сходинка конвеєра (СК_j) складається з двох компонент – процесорів і буферної паралельної пам'яті.

Структури паралельно-потоківих комп'ютерних систем ЦОС з обміном через буферну паралельну пам'ять на базі алгоритмічних і функціонально орієнтованих процесорів наведені відповідно на рис.14.3 а і б, де БПП – буферна паралельна пам'ять; ПрК – процесор керування; АП – алгоритмічний процесор; ФОП – функціонально-орієнтований процесор; СПП – спеціалізована паралельна пам'ять.

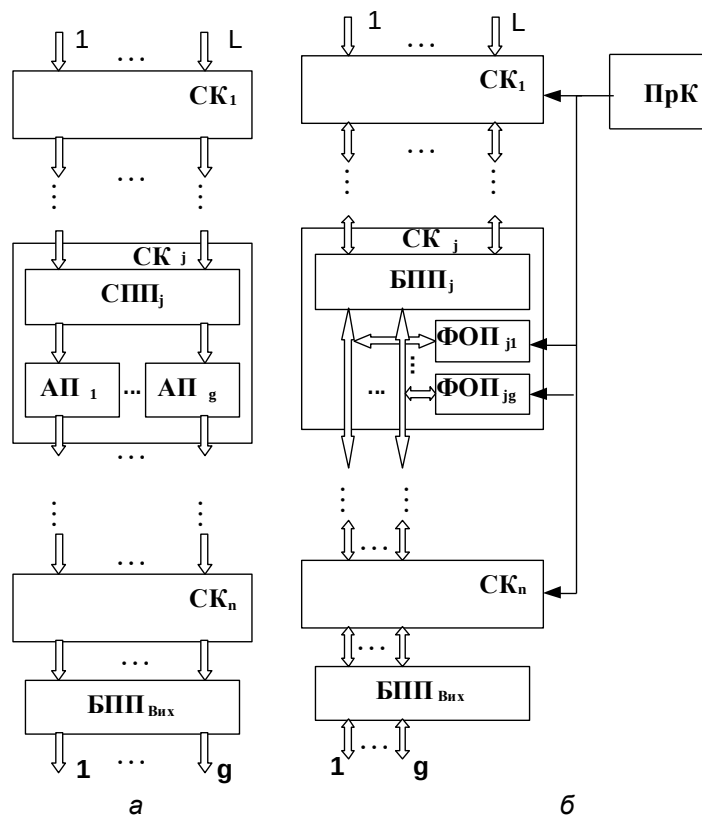


Рис.14.3 Базові структури паралельно-потоківих систем ЦОС з обміном через паралельну пам'ять: а) на базі алгоритмічних процесорів; б) на базі функціонально-орієнтованих процесорів

Особливістю паралельно-потоківих систем ЦОС на базі АП є використання СПП, структура якої адаптована до структури даних і структури алгоритмів розв'язування. Така паралельно-потоківих система ЦОС працює з макротактом, який рівний

$$T_{MT} = \frac{N}{g} t_{СПП} + t_{АП}$$

де g – кількість трактів обробки; $t_{СПП}$ – час звертання до СПП; $t_{АП}$ – час виконання найскладнішої макрооперації.

В паралельно-потоківих системах можуть використовуватися АП як з синхронним, так і асинхронним режимами роботи. Такі системи забезпечують опрацювання потоків даних в реальному часі, що надходять з інтенсивністю $P_d = gT_{кап}$, де $T_{кап}$ – такт роботи конвеєра АП. Паралельно-

потокові системи на базі АП доцільно реалізовувати при повному відпрацюванні алгоритмів розв'язування задач.

Гнучкішими є паралельно-потокові системи на базі ФОП, вони дозволяють оперативно здійснювати модифікацію алгоритмів та перехід на розв'язування нових задач. Перед початком роботи такої системи необхідно здійснити її налаштування на розв'язання заданого класу задач. Ця процедура виконується шляхом завантаження в пам'ять програм ФОП відповідних програм, які є складовими частинами загальної програми. Процес розв'язання задачі починається з завантаження першого масиву вхідних даних і інформації управління, що супроводжує цей масив, в перший БПП₁. В інформації управління вказується код задачі, яку необхідно виконати, розмірність і стан масиву. Інформація управління записується в визначенні адреси БПП₁. Ємність БПП_j сходинки конвеєра визначається розмірами масивів проміжних результатів і керуючої інформації, що їх супроводжує наступним чином:

$$Q_{\text{БПП}} = 2Nn + 2Vn$$

де N – кількість проміжних результатів; n – розрядність слова; V – кількість слів управління, які супроводжують масив в процесі його опрацювання.

Особливістю такої системи є можливість роботи в синхронному або асинхронному режимі. В синхронному режимі паралельно-потокова система синхронізується сигналами переривання, які надходять з ПрК.

Період поступлення цих сигналів рівний макротакту роботи системи, який визначається так:

$$T_{\text{ПП}} = \frac{2(N+V)t_{\text{БПП}}}{L} + t_{\text{max MO}}$$

де $t_{\text{max MO}}$ – час виконання найскладнішого макрооператора.

В асинхронному режимі роботи відсутній спільний такт роботи системи. Перехід j -ої сходинки системи від опрацювання попереднього масиву на наступний здійснюється після завершення j -ою сходинкою опрацювання попереднього масиву і завантаження в БПП $_j$ наступного масиву. Управління роботою паралельно–поточною системою в асинхронному режимі здійснюється шляхом аналізу інформації управління, яка супроводжує масив даних, що передаються між сусідніми сходинками системи.

Паралельно-потокова система забезпечує високу ефективність завантаження апаратних засобів і найбільше відповідає умовам роботи в реальному часі. Застосування паралельної пам'яті для взаємодії між процесорами та сходинками конвеєра зводить до мінімуму проблеми, що пов'язані з синхронізацією роботи процесорів та системи в цілому [].

Для підтримки високої пропускної здатності необхідно, щоб усі сходинки конвеєра мали приблизно однакові часи обчислення функціональних операторів.

У випадку, коли пропускна здатність конвеєра системи $D_k = L/T_k$ є меншою чим інтенсивність поступлення даних $P_d = n/T_d$, то для забезпечення обробки в реальному часі необхідно паралельне включення декількох таких конвеєрів, кількість яких визначається виразом:

$$S = \lceil P_d / D_k \rceil$$

де $\lceil \rceil$ - знак округлення до більшого цілого. Особливістю паралельно-потокової системи є інтерфейс, який в кожному такті роботи забезпечує одночасне введення N даних і виведення g результатів обробки.

Реалізація СКС з таким інтерфейсом вимагає значної кількості виводів і великих розмірів кристала, що при реалізації системи у вигляді НВІС, що веде до збільшення її вартості

14.3 Реконфігуровані комп'ютерні системи ЦОС

Реконфігуровані комп'ютерні системи ЦОС зводяться до двох типів: спеціалізованої та проблемно-орієнтованої [9]. Структура спеціалізованої реконфігурованої комп'ютерної системи ЦОС наведена на рис.14.4, де БПП – буферна паралельна пам'ять, ПР - пам'ять реконфігурацій.

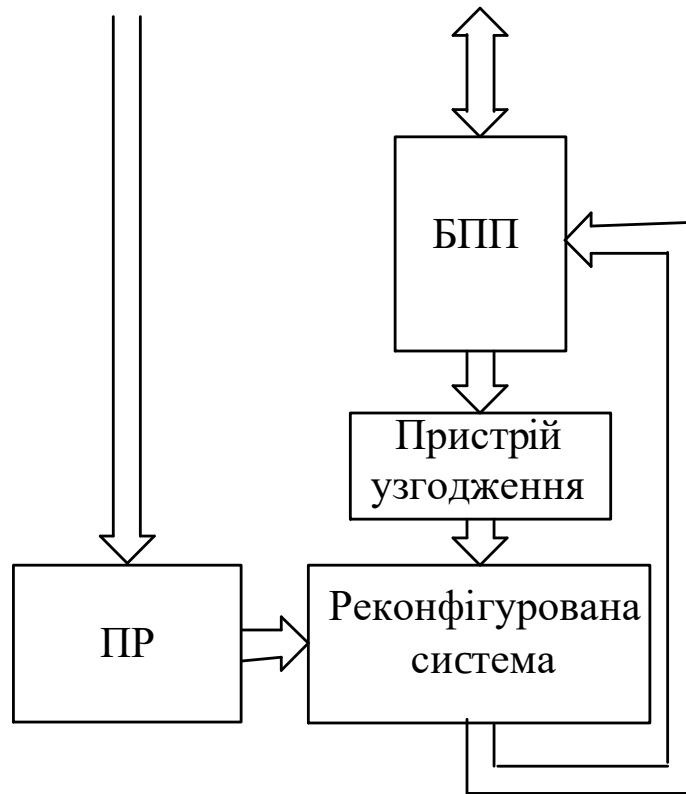


Рис.14.4.Реконфігурована комп'ютерна система ЦОС

Структура реконфігурованої комп'ютерної системи ЦОС є гнучкою, оскільки вона забезпечує оперативне переналаштування алгоритму розв'язання задачі.

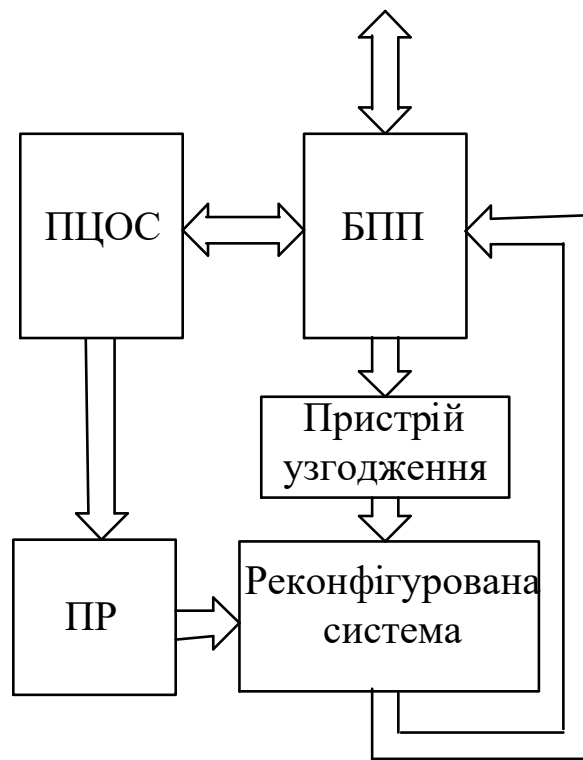


Рис.14.4 Комп'ютерна система ЦОС на базі реконфігурованих і програмованих процесорів

Структури комп'ютерної системи ЦОС на базі реконфігурованих і програмованих процесорів наведена на рис.14.4. Таку структуру доцільно використовувати для реалізації складних алгоритмів з нерегулярними зв'язками та великою кількістю логічних операцій. На ранніх етапах створення та відпрацювання алгоритмів доцільно використовувати комп'ютерну систем на базі реконфігурованих і програмованих процесорів ЦОС.

Література : [1], [2], [6], [8], [11], [12], [13].

15. Базові структури процесорів ЦОС

15.1 Алгоритмічні процесори ЦОС.

15.2 Програмовані процесори цифрової обробки сигналів.

15.3 Процесори швидких косинус- і синус-перетворень Фур'є.

15.1 Алгоритмічні процесори ЦОС.

Структури алгоритмічних процесорів (АП) ЦОС мають повною мірою використовувати можливості НВІС-технології, враховувати вартість площі кристала, а також кількість вхідних і вихідних виводів. Кількість зовнішніх виводів НВІС обмежена рівнем технології та розміром кристала. При розробці структури процесора необхідно узгодити швидкодію обробки інформації в НВІС з швидкодією введення-виведення, тобто з обчислювальною здатністю НВІС, яка визначається як кількістю, так і якістю зовнішніх виводів. Якість зовнішніх виводів визначає затримку переключення зовнішніх зв'язків, які навантажені на ці виводи. За характеристиками введення-виведення визначаються вимоги до ємності внутрішньої пам'яті НВІС. Особливо жорсткі вимоги до введення-виведення висуваються тоді, коли на процесорі розв'язується задача великої розмірності, а ємність пам'яті НВІС не дозволяє розмістити необхідний об'єм даних. В цих випадках необхідно або наростити процесор, або розбити задачу на частини і розв'язувати її послідовно. Другий підхід вимагає перелаштування процесора в процесі розв'язання задачі. З цього випливають два підходи до налаштування АП: перший ґрунтується на програмуванні задачі для процесора, другий - на програмуванні структури [9]. Другий підхід використовується при реалізації АП на базі оперативно реконфігурованих НВІС, що дозволяє міняти конфігурацію сходинок процесора в процесі роботи. Ці два підходи можуть використовуватися як окремо, так і спільно. Одночасне використання двох підходів дозволить створювати високоефективні АП реального часу.

Структури АП ґрунтуються на матричних мережах алгоритмічних операційних пристроїв (АОП), які апаратно відображають узгоджені потокові графи алгоритмів розв'язування задачі. Розв'язування задачі в АП здійснюється за один прохід даних через множину АОП, які виконують всі функціональні оператори потокового графа алгоритму. В такому процесорі вихідні результати Y однозначно визначаються вхідними даними X і функцією Φ , зашитою в його структурі $Y=\Phi(X)$. У загальному випадку в АП може здійснюватись обробка двовимірних масивів даних $X=\{X_{fj}\}$ з отриманням матриці результатів

$$Y=\{Y_{fk}\}$$

де $f=1,\dots,N$; $j=1,\dots,R$; $k=1,\dots,P$; N - кількість груп чисел; R і P – кількість входів і виходів процесора [49].

Такі процесори можуть працювати як в асинхронному, так і в синхронному режимах. Для обробки інтенсивних потоків даних найбільше підходять конвеєрні паралельно-потоків АП, які діляться на три типи: спеціалізовані, налаштовуванні та реконфігуровані.

Спеціалізовані паралельно-потоків АП синтезуються на базі спеціалізованої паралельної пам'яті (СПП) та однофункціональних алгоритмічних операційних пристроях (ОАОП). Структура спеціалізованого паралельно-потоків АП наведена на рис.15.1.а, СК – сходи́нка конвеєра. Перевагою такого АП є висока ефективність використання обладнання, а їх недоліком – неможливість зміни алгоритму роботи. Налаштовуванні АП синтезуються на базі буферної паралельної пам'яті (БПП) та багатофункціональних алгоритмічних операційних пристроях (БАОП) і забезпечують реалізацію заданого класу алгоритмів. Структура налаштовуваного паралельно-потоків АП наведена на рис.15.1.б, де ПК пристрій керування. У таких АП налаштування на виконання необхідного алгоритму здійснюється інформацією на вході K .

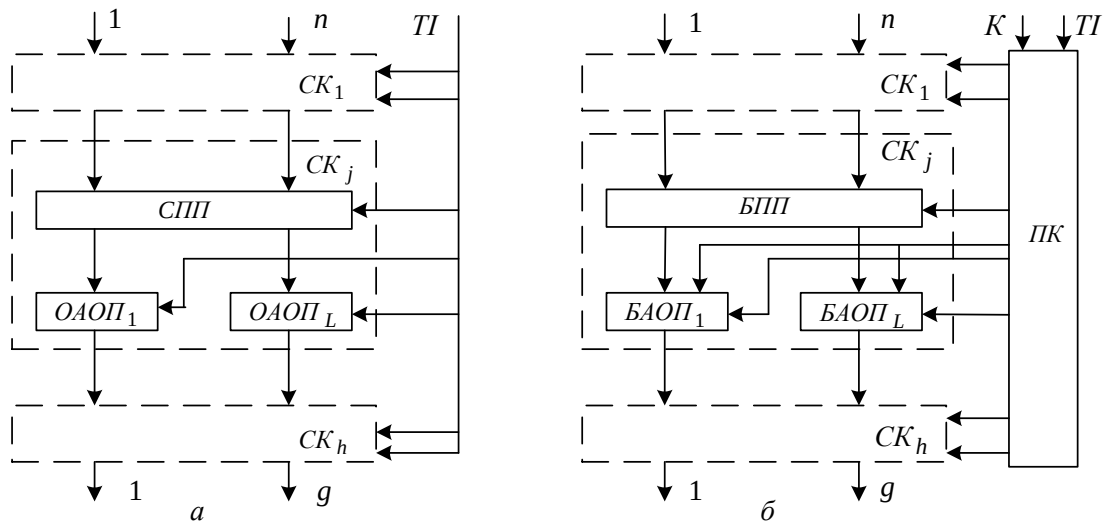


Рис.15.1 Структури паралельно-потоківих алгоритмічних процесорів:

а) спеціалізована; б) з налаштуванням

Гнучкішою є структура, яка синтезується на базі буферної паралельної пам'яті (БПП), пам'яті реконфігурації (ПР) та реконфігурованих НВІС (рис.15.2). Такі АП є апаратно надлишковими і їх доцільно використовувати на ранніх етапах створення інформаційних технологій при відпрацюванні алгоритмів функціонування.

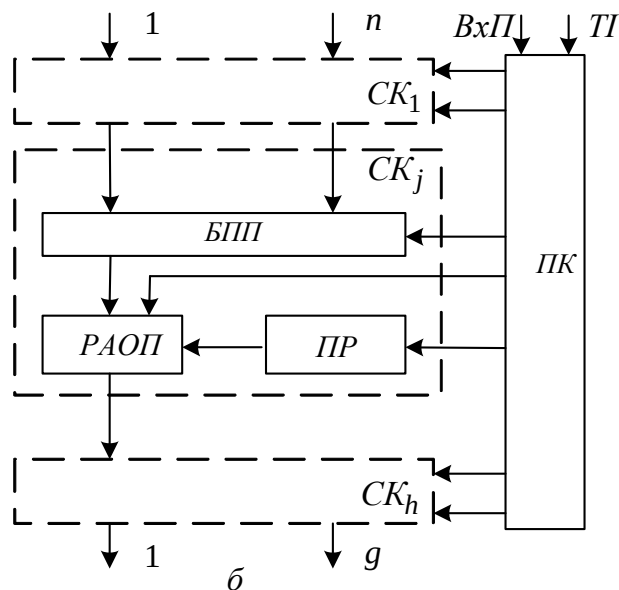


Рис.15.2 Паралельно-потоківі реконфігуровані процесори

Налаштування реконфігурованих паралельно-потоківих процесорів на реалізацію необхідного алгоритму здійснюється інформацією з ВхП, яка записується і ПР кожної сходинки конвеєра.

Опрацювання даних у АП здійснюється при подачі тактових імпульсів, які однонаправлено просувають інформацію з входу на вихід, шляхом запису проміжних результатів у спеціалізовану або буферну паралельну пам'ять. Частота надходження тактових імпульсів дорівнює

$$F_{TI} = \frac{1}{t_{БПП} + t_{\max ОП_{jk}}}$$

де $t_{БПП}$ – час звертання до буферної паралельної пам'яті; $t_{\max ОП_{jk}}$ – час виконання найскладнішого функціонального оператора Φ_{jk} .

15.2 Програмовані процесори цифрової обробки сигналів

Програмовані процесори ЦОС (ПЦОС) реалізуються на базі однокристальних мікропроцесорів ЦОС шляхом їх доповнення пам'яттю, інтерфейсними і операційними пристроями. Такі процесори мають змінний склад обладнання, який забезпечує оптимізації швидкісних і масогабаритних параметрів і розширює галузі їх застосування [2,3,9]. ПЦОС із змінним складом обладнання просто адаптовуються до вимог конкретних застосувань, забезпечують високу ефективність використання обладнання та можуть використовуватися для синтезу широкого спектру ПЦОС.

Структура ПЦОС зі змінним складом обладнання представляється у вигляді процесорного ядра і довільної кількості пристроїв, які приєднуються до процесорного ядра в залежності від вимог розв'язуваної задачі. Склад процесорного ядра для всіх застосувань є постійним з мінімальною кількістю пристроїв, які перетворюють його у незалежний ПЦОС. Процесорне ядро реалізовується у вигляді процесорного модуля (ПМ), що володіє властивістю до розширення, яка є основною підставою для побудови його на базі сімейства ПЦОС.

Основні характеристики ПМ в більшості визначаються особливостями архітектури і технічними характеристиками мікропроцесорів ЦОС. До числа таких характеристик відносяться: довжина інформаційного слова, час реалізації основних команд, обсяг внутрішньокристальної пам'яті даних і програм.

Сучасні мікропроцесори ЦОС мають добре розвинуту багатоштинну організацію, пристрої множення, зсувачі, пам'яті програм і даних. За рахунок вдалих архітектурних рішень в таких мікропроцесорах вдалось сумістити цифрову обробку з реалізацією алгоритмів управління та логічної обробки [].

Самостійно мікропроцесор ЦОС не відповідає вимогам, які ставляться до ПЦОС за ємністю пам'яті, завадостійкістю та навантажувальною здатністю інтерфейсу. Задовольнити такі вимоги можна шляхом доповнення мікропроцесора ЦОС зовнішньою оперативною та багатопортовою пам'яттю, адаптером і розширювачем інтерфейсу.

Структура ПМ на базі мікропроцесора ЦОС наведена на рис.15.3, де ПК – пристрій керування; МП – мікропроцесор; БП – багатопортова пам'ять; ОЗП - оперативний запам'ятовуючий пристрій; АІ – адаптер інтерфейсів; РІМП – розширювач інтерфейсу мікропроцесора.

Застосування БП для взаємодії між ПМ і зовнішніми пристроями зменшує час обміну та розв'язує проблеми пов'язані з синхронізацією їх роботи.

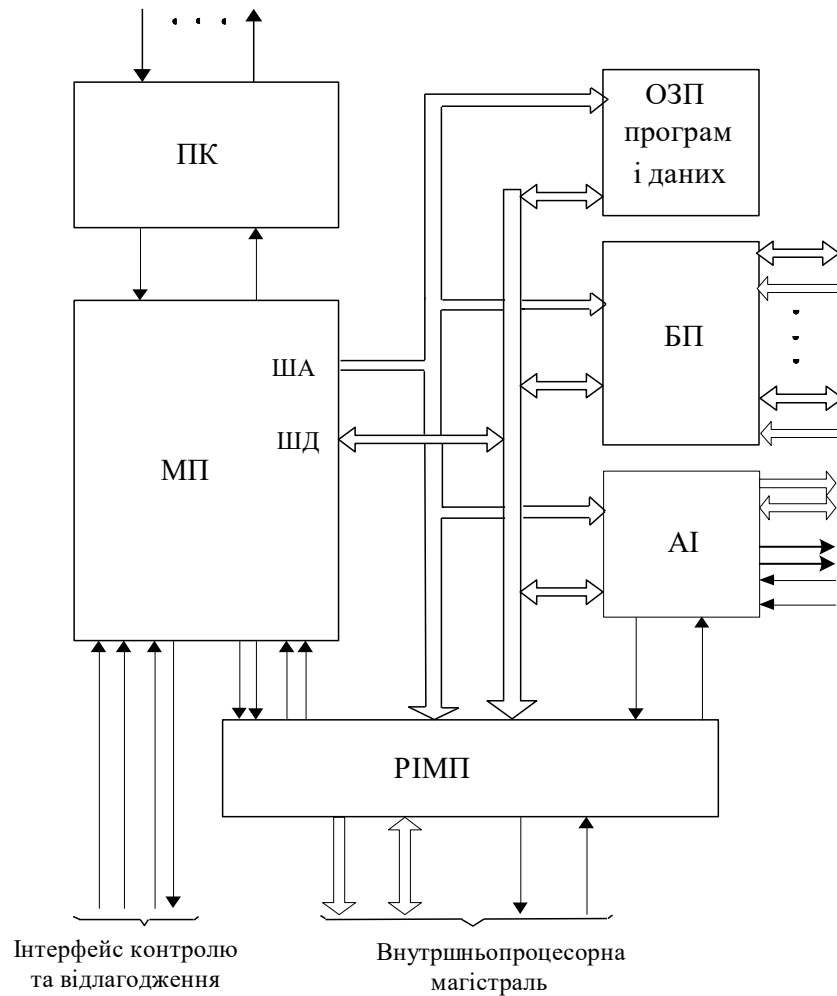


Рис.15.3 Структура процесорного модуля

Особливістю структурної організації ПМ є відділення процесорного ядра від зовнішніх пристроїв, що дозволило підвищити завадостійкість і сумістити у часі роботу процесорного ядра та зовнішніх пристроїв.

15.3 Процесори швидких косинус- і синус-перетворень Фур'є

В основі структурної організації процесорів швидких косинус- і синус-перетворень Фур'є (ШКПФ-ШСПФ) лежить матрична мережа, яка є двовимірною сукупністю ПЕ, які апаратно реалізують базові операції (БО) та з'єднані між собою відповідно до графа алгоритму ШКПФ-ШСПФ (рис.15.4).

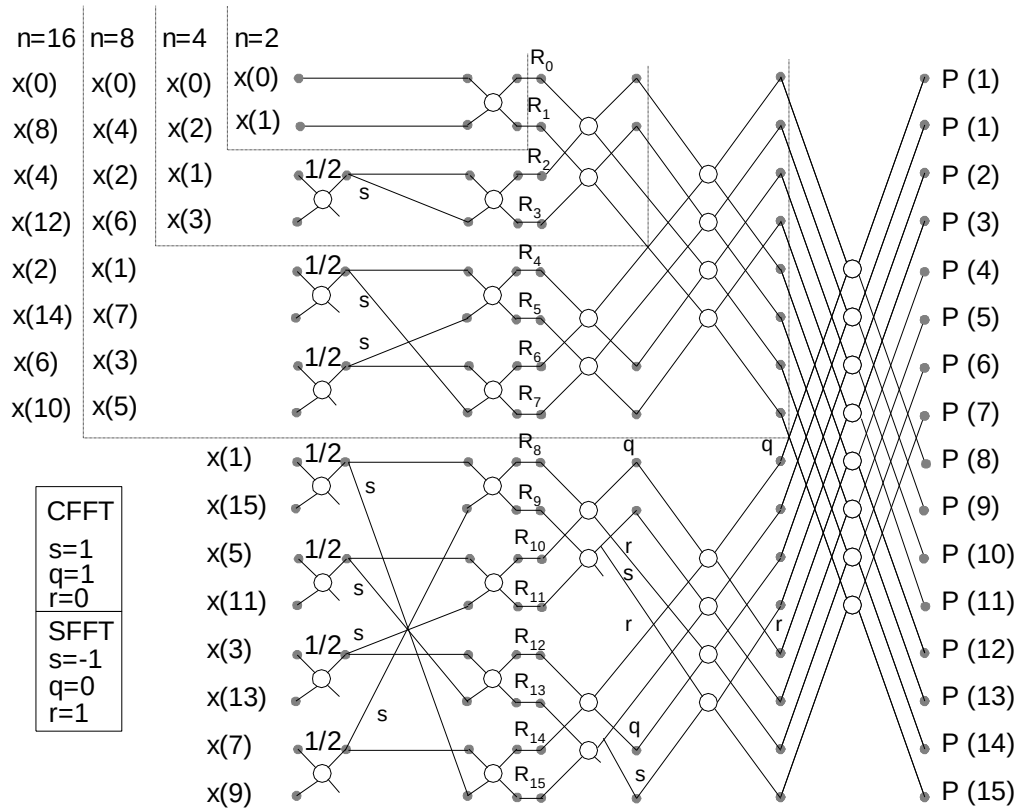


Рис.15.4 Граф алгоритму ШКПФ-ШСПФ

Апаратне відображення графа модифікованого алгоритму ШКПФ-ШСПФ на ПЕ, які складаються з операційних пристроїв і конвеєрних регістрів, дозволяє отримати матричний конвеєрний процесор. Такий процесор повинен забезпечити неперервну обробку потоку даних, що надходять з інтенсивністю $P_d = k n_d F_d$. Структура матричного процесора 16-точкового ШКПФ-ШСПФ, який працює за конвеєрним принципом, наведена на рис.15.5, де ТІ – вхід тактових імпульсів; RS – вхід скиду; СК – сходи́нка конвеєра.

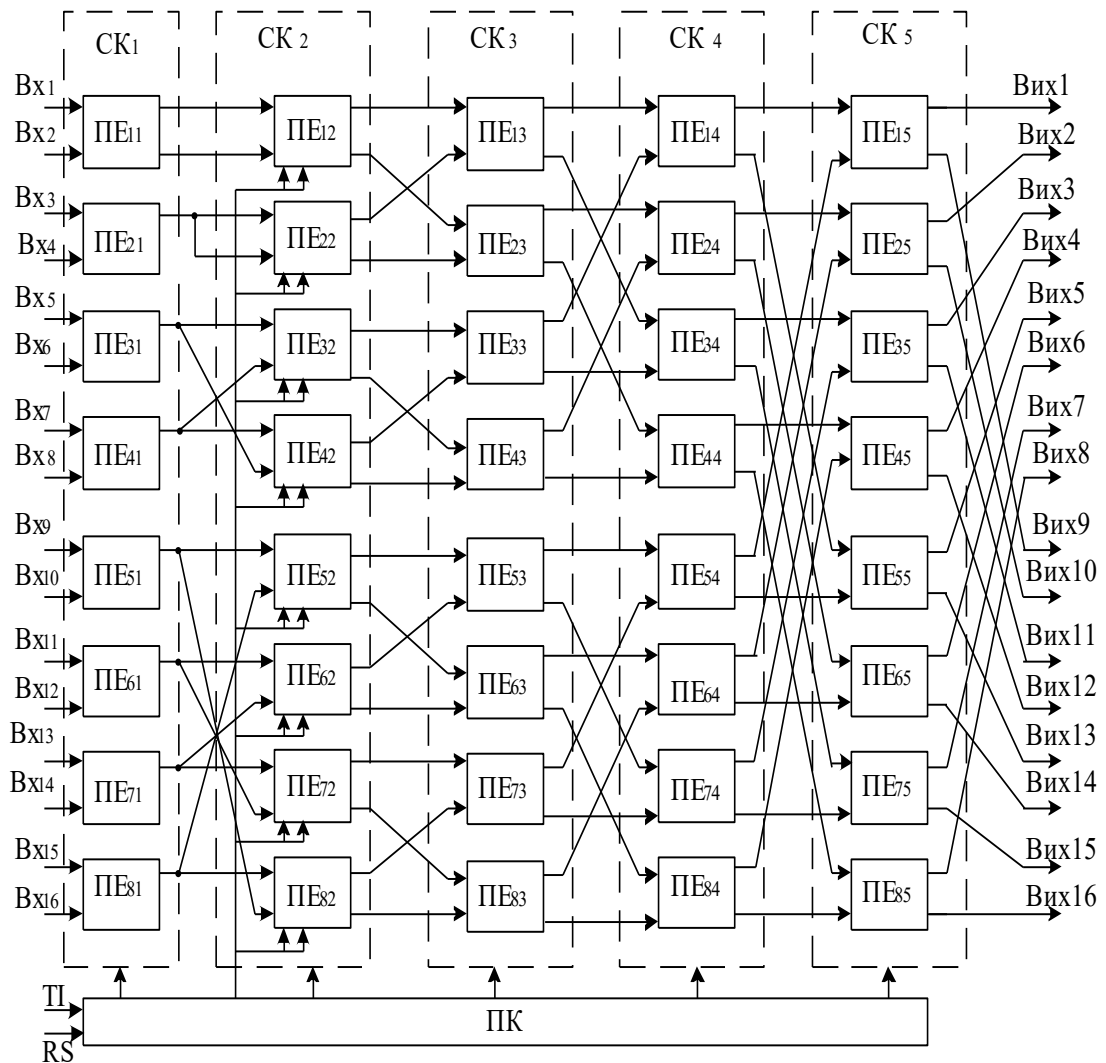


Рис.15.5 Структура матричного конвеєрного процесора 16-точкового
ШКПФ-ШСПФ

Схеми основних ПЕ, на базі яких реалізується такий процесор, наведені на рис.15.6, де См – суматор; Вд – віднімач; КПМ – конвеєрний пристрій множення, який працює з тактом рівним часу спрацювання суматора та регістра

$$T_{к1} = t_{см} + t_{пр}.$$

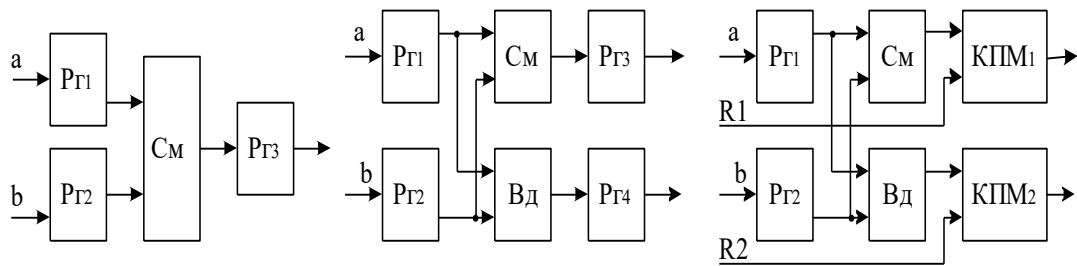


Рис.15.6. Схеми процесорних елементів

Розроблений матричний процесор 16-точкового ШКПФ-ШСПФ забезпечує наступну інтенсивність обробки

$$P_{МП1} = \frac{16n}{t_{Cm} + t_{Pz}}$$

Особливістю матричного процесора ШКПФ-ШСПФ є інтерфейс, який в кожному такті роботи забезпечує одночасне введення 16 даних і виведення 16 результатів обробки. Реалізація процесора з таким інтерфейсом вимагає значної кількості виводів і великих розмірів кристала. Апаратні витрати на реалізацію матричного процесора ШКПФ-ШСПФ обчислюються за наступною формулою

$$W_1 = 4N(1 + \log_2 N)W_{Pr} + (N/2 + N \log_2 N)W_{Cm} + W_{Pk} + NW_{KPM}$$

де N -розмірність перетворення; W_{Pr} , W_{Cm} , W_{Pk} , W_{KPM} – апаратні витрати відповідно на регістр, суматор, пристрій керування та конвеєрний пристрій множення.

Для синтезу процесора з меншою кількістю виводів використовують лінійну проекцію потокового графа модифікованого алгоритму ШКПФ-ШСПФ на вісь паралельну передачі даних. При цьому для кожного ярусу формується проекція БО - макрооператор та проекція каналів передачі даних оператор зберігання. Для відтворення потокового графу модифікованого

алгоритму ШКПФРБ-ШСПФРБ на базі макрооператорів та операторів зберігання для кожного ярусу вводяться оператори управління, які визначають послідовність операцій у ПЕ та керують обчислювальним процесом у відповідності із структурою алгоритму

При апаратному відображенні лінійної проекції графа алгоритму макрооператор і оператор зберігання кожного ярусу реалізується ПЕ, а оператори управління пристроєм керування. Структурна схема потокового процесора 2-4-8-16-точкового ШКПФ-ШСПФ наведена на рис.15.7, де ТІ – вхід тактових імпульсів; РП – вхід задання розміру перетворення; cos/sin – вхід задання виду перетворення; СПП – спеціалізована паралельна пам'ять; СВ – суматор-віднімач; КПМ – конвеєрний пристрій множення. Обробка даних в процесорі здійснюється за конвеєрним принципом з тактом

$$T_{к1}=t_{CM}+t_{PГ}+t_{KM}$$

де $t_{PГ}$, t_{CM} і t_{KM} - час спрацювання відповідно регістра, суматора і комутатора. Інтенсивність обробки даних у розробленому процесорі визначається за наступною формулою

$$P_{ппп1} = \frac{2n}{t_{CM} + t_{P2} + t_{KM}}$$

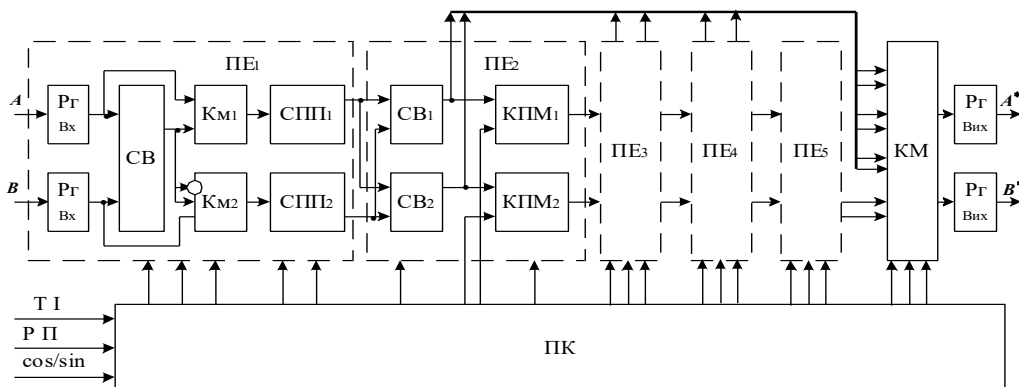


Рис.15.7 Структурна схема 2-4-8-16-точкового процесора ШКПФ ШСПР

Для реалізації 16-точкового перетворювання ШКПФ-ШСПФ використовуються п'ять послідовно з'єднаних процесорних елементів (ПЕ), перший з яких здійснює виділення парної та непарної складових, другий базову операцію виду $a'=(a+b)R_j$, $b'=(a-b)R_j$, а решта - її спрощений варіант $a'=a+b$, $b'=a-b$. До складу ПЕ₃, ПЕ₄ і ПЕ₅ входять суматор-віднімач СВ, комутатор Км і пристрій затримки, що складається з набору регістрів Рг. Схема таких ПЕ наведена рис.15.8.

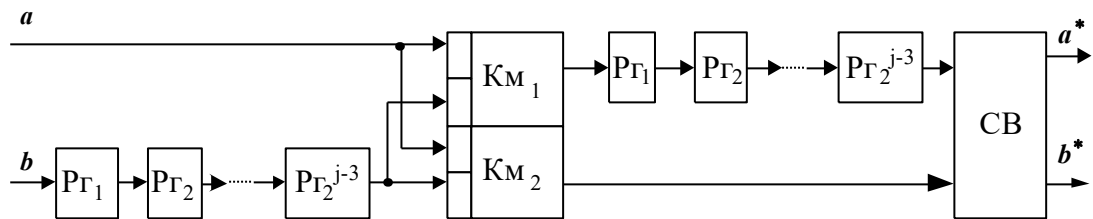


Рис.15.8. Схема процесорного елемента

Затримка інформації в ПЕ_j, де $j=3, 4, 5$, визначається його порядковим номером j та дорівнює $Z_j=2^{j-3}$. Процесор забезпечує виконання 2-4-8- чи 16-точкового ШКПФ-ШСПФ, при цьому тактова частота роботи прийому і видачі даних не залежить від розмірності перетворення. Апаратні витрати на реалізацію такого процесора рівні

$$W_2=(1+2\log_2 N)W_{\text{СВ}}+2W_{\text{КПМ}}+2W_{\text{СПП}}+(2+2\log_2 N)W_{\text{Км}}+(N+2\log_2 N)W_{\text{Рг}}+W_{\text{Км}}+W_{\text{ПК}}$$

де $W_{\text{Рг}}$, $W_{\text{СВ}}$, $W_{\text{ПК}}$, $W_{\text{КПМ}}$, $W_{\text{Км}}$, $W_{\text{4Км}}$, $W_{\text{СПП}}$ – апаратні витрати відповідно на регістр, суматор-віднімач, пристрій керування, конвеєрний пристрій множення, двовходовий комутатор, чотиривходовий комутатор і спеціалізовану паралельну пам'ять.

Література : [1], [2], [4], [6], [9], [13].

16. Задачі і основні принципи проектування комп'ютерних систем цифрової обробки сигналів

На сучасному етапі розвитку інтегральної технології основною метою проектування комп'ютерних систем ЦОС є отримання модульної та регулярної структури орієнтованої на НВІС-технологію. Вихідною інформацією для проектування високопродуктивних комп'ютерних систем ЦОС є:

- алгоритми розв'язання задачі;
- кількість вхідних даних N ;
- інтенсивність вхідного потоку даних P_d ;
- вимоги до інтерфейсу САЗ, включаючи частоту приймання та видачі інформації;
- розрядність вхідних даних і точність обчислень;
- техніко-економічні вимоги і обмеження.

При проектуванні комп'ютерних систем ЦОС необхідно забезпечити реалізацію заданих алгоритмів розв'язання задачі в реальному часі при мінімальних апаратних затратах. Перехід від алгоритму розв'язання задачі в реальному часі до структури комп'ютерних систем ЦОС формально зводиться до мінімізації апаратних затрат

$$W_{CA3} = W_{III} + W_{II} + W_{IIV} + \sum_{i=1}^n W_{II E_i} m_i$$

де W_{CA3} , W_{IIV} , W_{II} , W_{II} , $W_{II E}$ - апаратні затрати відповідно на комп'ютерну систему ЦОС, пристрої управління, інтерфейсні пристрої, пам'ять, i -тий процесорний елемент, m_i – кількість процесорних елементів i -го типу, при забезпеченні у залежності від виду надходження даних однієї з наступних умов:

$$F_d K_d \leq \frac{L}{T_k} \quad \text{або} \quad \frac{N_d}{F_d K_d} \geq T_p$$

де F_d – частота надходження вхідних даних, K_d – кількість каналів надходження вхідних даних, L – кількість каналів обробки, N_d – кількість вхідних даних; T_k – тривалість такту роботи конвеєра, T_p – час розв'язання задачі.

В загальному задачі проектування комп'ютерних систем ЦОС можна сформулювати так:

- вибрати алгоритм розв'язання задачі та представити його у вигляді конкретизованого потокового графу;
- розробити структуру комп'ютерної системи ЦОС з максимальною ефективністю використання обладнання, яка враховує всі обмеження та забезпечує обробку даних в реальному масштабі часу;
- визначити основні характеристики компонентів комп'ютерної системи ЦОС та здійснити їх синтез;
- вибрати способи обміну, визначити необхідні зв'язки та розробити систему обміну між компонентами комп'ютерної системи ЦОС;
- визначити порядок реалізації у часі алгоритму розв'язання задачі та синтезувати пристрої управління.

Процес проектування комп'ютерних систем ЦОС зводиться до виконання наступних етапів:

- вибору та відображення алгоритму у вигляді конкретизованого потокового графу ;
- вибору базової структури комп'ютерних систем ЦОС, розробки топології зв'язків і функцій синхронізації обміну між процесорними елементами;
- розробки інтерфейсу зв'язку з оточуючим середовищем, алгоритмів генерації адрес для запису вхідних даних у пам'яті, визначення переліку операцій (макрооперацій), які повинні виконувати процесорні елементи (процесори);

- синтезу структури процесорних елементів (процесорів), які реалізують визначений перелік операцій (макрооперацій), з врахуванням особливостей елементної бази, технології виготовлення та обмежень за продуктивністю.

Завершальними результатами перших трьох етапів проектування є набір адекватних алгоритму структур, що складаються з обмеженої множини процесорних елементів (процесорів), об'єднаних системою комутації та забезпечують технічні вимоги, які висуваються до комп'ютерної системи ЦОС.

Процес проектування спеціалізованих комп'ютерних систем ЦОС постійно ускладнюється за рахунок підвищення складності виконуваних задач і обмежень, які накладаються в частині габаритів, ваги і споживаної потужності. Крім цього, постійно вимагається зменшення термінів, вартості проектування та підвищення його якості. Забезпечити такі вимоги можливо при інтегрованому підході до проектування комп'ютерних систем ЦОС, який охоплює:

- дослідження та розробку теоретичних основ і методів проектування комп'ютерних систем ЦОС як окремого класу засобів комп'ютерної техніки;
- пошук нових алгоритмічних, структурних, схемотехнічних і конструктивно-технологічних рішень, орієнтованих на НВІС-технології, які забезпечать задані параметри комп'ютерних систем ЦОС на існуючій та перспективній елементній базі;
- вибір, а при необхідності проектування, швидкодіючих малоспоживаючих НВІС-архітектур, що враховують особливості виконуваних задач і вимог застосування;
- впровадження та створення засобів автоматизованого проектування комп'ютерних систем ЦОС, які забезпечать зменшення термінів і підвищать якість проектування.

Архітектури комп'ютерних систем ЦОС повинні в повній мірі використовувати можливості НВІС-технології та забезпечувати обробку інтенсивних потоків інформації в реальному часі [92]. Вартість НВІС в основному залежить від площі кристала і кількості виводів. Число зовнішніх виводів НВІС обмежене рівнем технології та розміром кристалу. При розробці спеціалізованих НВІС-структур необхідно узгодити пропускну здатність та швидкодію обробки інформації в НВІС з інтенсивністю P_d вхідних потоків даних. Пропускна здатність НВІС визначається як кількістю, так і якістю зовнішніх виводів. Якість зовнішніх виводів визначає затримку переключення зовнішніх зв'язків, які навантажені на ці виводи. За характеристиками вводу-виводу визначаються вимоги до ємності внутрішньої пам'яті НВІС. Особливо жорсткі вимоги до вводу-виводу висуваються тоді, коли за допомогою НВІС-процесора вирішується задача великої розмірності, а ємність пам'яті НВІС не дозволяє розмістити необхідний об'єм даних. В цих випадках необхідно або наростити НВІС-процесор, або розбити задачу на частини і вирішувати її послідовно. Другий варіант вимагає перестроювання НВІС-процесора в процесі вирішення задачі, що можливо у випадку коли НВІС є оперативною програмованою мікросхемою.

Для забезпечення нарощування обчислювальних потужностей, розширення виконуваних функцій, адаптації до специфіки задач, що розв'язується, необхідно комп'ютерні системи ЦОС будувати за модульним принципом, при якому основні елементи (процесори, пам'ять) реалізуються у вигляді функціонально завершених пристроїв (модулів). Модульний підхід при проектуванні САЗ сприяє стандартизації елементів, скороченню витрат на проектування та відсуває час морального старіння технічних засобів. Структури модулів повинні бути орієнтованими на застосування НВІС, а також на реалізацію окремих модулів чи їх вузлів у вигляді НВІС. Впровадження НВІС перш за все повинно сприяти зменшенню габаритів, маси, споживання, а також збільшенню надійності та швидкодії САЗ шляхом

перекладання на апаратну частину функцій програмних засобів, які найширше використовуються та є найважливішими. Апаратна реалізація таких функцій спрощується при реалізації їх на модулях з процесорною реалізацією, яка передбачає наявність пристроїв обробки, зберігання, управління та обміну.

Для найповнішого використання переваг НВІС технології при розробці архітектури комп'ютерних систем ЦОС використовується наступні принципи:

- адаптації апаратно-програмних засобів до алгоритмів розв'язання задач;
 - структурно-алгоритмічної орієнтації при розробці методів, алгоритмів і апаратних засобів;
 - перекладання на апаратну частину якомога більшого числа функцій програмного забезпечення;
 - використання оперативно програмованих НВІС та реконфігурованих архітектур;
 - забезпечення регулярності і модульності архітектури комп'ютерних систем ЦОС та широке використання стандартних елементів;
 - локалізації та спрощення зв'язків між елементами комп'ютерних систем ЦОС;
 - забезпечення балансу між вводом/виводом та обчисленнями;
 - широкого використання конвеєризації і просторового паралелізму обробки даних;
 - одноциклової реалізації більшості команд;
 - використання універсальної та спеціалізованої паралельної пам'яті для обміну та зберігання інформації;
 - мінімізації пересилок та скорочення операцій над адресами.
- Література : [1], [4], [6], [9], [10], [13].

17. Оцінка основних параметрів комп'ютерних систем ЦОС

17.1 Оцінка витрат обладнання.

17.2 Продуктивність комп'ютерних систем обробки сигналів.

17.3 Ефективність використання обладнання

17.1 Оцінка витрат обладнання

Витрати обладнання - це кількість апаратури, що необхідна для реалізації комп'ютерної системи, виражена в певних одиницях [9]. Одиницею виміру витрат обладнання може бути: кількість блоків стандартних розмірів; кількість друкованих плат; кількість однотипних НВІС або умовних корпусів, приведених до одного типу корпусу; кількість вентилів або транзисторів при реалізації компонент системи у вигляді НВІС. Для сучасних СКС, що виготовлені на базі НВІС, за одиницю виміру витрат обладнання доцільно брати вентиль, який рівний витратам на логічні елементи типу інвертор, І-НЕ, АБО-НЕ. В залежності від вимог до швидкодії, потужності споживання і стійкості до зовнішніх факторів НВІС можуть бути реалізовані за n-МОН, КМОН, ЕСЛ, ТТЛШ технологіями або на основі арсеніду галію (As Ga). В кожній з перерахованих технологій основні функціональні вузли (реєстри, суматори, комутатори, дешифратори, перемножувачі, елементи пам'яті і т.д.) систем мають різну реалізацію. Витрати обладнання на реалізацію комп'ютерних систем ЦОС складаються з витрат на її складові частини:

$$W_{СКС} = W_{БПП_{Вх}} + W_{БПП_{Вих}} + W_{ПК} + \sum_{j=1}^m W_{СКj}$$

де $W_{БПП_{Вх}}$ і $W_{БПП_{Вих}}$ - витрати обладнання на вхідну та вихідну буферну паралельну пам'ять; $W_{ПК}$ - витрати обладнання на пристрій керування; $W_{СКj}$ - витрати обладнання на j- сходинку конвеєра системи.

При реалізації комп'ютерних систем ЦОС на НВІС витрати обладнання на її реалізацію в загальному вигляді можна обчислити за формулою

$$W_{СКС} = \sum_{i=1}^n W_{\Phi y_i} m_i,$$

де $W_{\Phi y_i}$ - витрати обладнання на i -й тип функціонального вузла;
 m_i - кількість вузлів i -го типу.

17.2 Продуктивність комп'ютерних систем обробки сигналів

Однією із основних характеристик комп'ютерних систем ЦОС є продуктивність, яка залежить від множини параметрів [9], основними є:

- тактова частота приймання вхідних даних $F_1=1/T_{цзп}$, де $T_{цзп}$ - час циклу запису в БПП;
- тактова частота видачі результатів $F_2=1/T_{цчт}$, де $T_{цчт}$ - час циклу читання із БПП;
- пропускна здатність конвеєра $D_k=m/T_k$, де m – кількість каналів поступлення даних, T_k – такт роботи конвеєра;
- кількість каналів введення K_1 та виведення K_2 (ширина доступу до вхідної та вихідної БПП);
- алгоритми розв'язання задач і швидкодія елементної бази.

В залежності від розміру вхідних масивів даних для визначення продуктивності конвеєрної комп'ютерної системи ЦОС може використовуватися: конвеєрний такт T_k , час обчислення T_o або блочний конвеєрний такт $T_{кб}$.

При обробці неперервних вхідних потоків даних продуктивність конвеєрної комп'ютерної системи ЦОС визначається наступним чином.

$$P_{кск_1} = \frac{R}{T_{АОП} / n + T_{Бпп}} = \frac{R}{T_k} \quad (17.1)$$

де R – складність алгоритму обчислення, яка оцінюється кількістю операцій необхідних для реалізації алгоритму; n - кількість сходінок конвеєра; $T_{АОП}$ – час асинхронного обчислення результату; T_k – конвеєрний такт; $T_{Бпп}$ – час звертання до БПП, який визначається її швидкодією.

З формули (17.1) видно, що підвищення продуктивності комп'ютерної системи ЦОС зв'язано з зменшенням конвеєрного такту T_k , який

визначається затримкою у найповільнішому ярусі системи. Його можна подати у вигляді суми $T_k = T_{Oям} + T_{Бп}$, де $T_{Oям}$ – максимальний час обчислення в ярусі. Відмінність у часах обчислення $T_{Oя}$ в ярусах приводить до неповного завантаження обладнання. Виникає задача вирівнювання часів $T_{Oя}$, яка розв'язується двома способами: спрощенням комбінаційної схеми швидших ярусів з метою мінімізації затрат обладнання, прискоренням роботи повільних ярусів шляхом їх конвеєризації або розпаралелювання.

При розв'язанні однієї задачі з одиничним масивом вхідних даних продуктивність СКС визначається за формулою

$$P_{скс_2} = \frac{R}{N_1(1-E_1)F_1K_1 + T_0 + N_2(1-E_2)F_2K_2},$$

де T_0 - час обчислення, який дорівнює часу між початком першого обчислення і завершенням останнього обчислення при розв'язанні задачі; N_1, N_2 – розміри відповідно вхідного масиву даних і вихідного масиву результатів; E_1, E_2 – коефіцієнти, які враховують ефективність часового суміщення відповідно введення вхідних даних і виведення результатів з процесом обчислення.

Для випадку коли СКС розв'язує декілька задач важливе значення має блочний конвеєрний такт $T_{Бк}$, який дорівнює інтервалу часу між початком розв'язку двох задач, що послідовно розв'язуються системою. Продуктивність САЗ у такому режимі роботи визначається так

$$P_{скс_3} = \frac{R}{T_{Бк}} = \frac{R}{N_1(1-E_1)F_1K_1 + T_0 + N_2(1-E_2)F_2K_2 + T_H},$$

де T_H – час налаштування на розв'язання задачі.

17.3 Ефективність використання обладнання

Ефективність використання обладнання - це інтегральний параметр, який зв'язує продуктивність комп'ютерної системи ЦОС з витратами

обладнання на їх реалізацію та дає оцінку елементам (вентиліям) системи за продуктивністю [9].

Кількісна величина ефективності використання обладнання визначається наступним чином

$$E = \frac{R}{W_{CK}t},$$

де R - складність алгоритму розв'язання задачі; W_{CK} - витрати обладнання на комп'ютерної системи в вентиліях; t - час розв'язання задачі.

Ефективність використання обладнання конвеєрних комп'ютерної системи ЦОС при обробці неперервних вхідних потоків визначається наступним чином

$$E_1 = \frac{R}{T_k \sum_{i=1}^n W_{\phi y_i} m_i}$$

де T_k – такт конвеєра.

Підвищення ефективності використання обладнання комп'ютерною системою ЦОС безпосередньо зв'язане зі зменшенням часу розв'язання задачі та витратами обладнання на реалізацію функціональних вузлів.

При оцінці ефективності використання обладнання пристроїв, які реалізуються у вигляді НВІС необхідно враховувати геометричні, динамічні та інші параметри як активних елементів, так і зв'язків між ними. Це зумовлено тим, що вартість розробки, виготовлення і експлуатації НВІС у великій мірі визначається площею кристалу. Зменшення розмірів активних елементів веде до пропорційного збільшення їх швидкодії та зменшення довжини ліній зв'язку.

Література : [1], [3], [5], [6], [9], [13].

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ.

1. Тотосько О.В. Цифрова обробка сигналів та зображень [Електронний ресурс] : навч. посіб. / О.В. Тотосько, П.Д. Стухляк. - Тернопіль : ТНТУ імені Івана Пулюя, 2016. - 132 с. – Режим доступу : <http://elartu.tntu.edu.ua/handle/123456789/18295>. - Назва з екрана.
2. Digital Image Processing Using MATLAB, 2nd ed. by Rafael C. Gonzalez [Електронний ресурс]. - Режим доступу : <http://surl.li/oqbdj>. - Назва з екрана.
3. Lyons Richard G. Understanding Digital Signal Processing [Електронний ресурс] / Richard G. Lyons. – Prentice Hill, Professional Technical Reference, 2011. - Режим доступу : <http://surl.li/oqbgg>. - Назва з екрана.
4. Наконечний А. Обробка сигналів : навч. посіб. / А. Наконечний. – Львів: Растр-7, 2017. - 217 с.
5. Gonzalez Rafael C. Digital Image Processing /Rafael C. Gonzalez, Richard E. Woods [Електронний ресурс]. - 4th Edition, 2017. - Режим доступу : <http://surl.li/oqbke>. - Назва з екрана.
6. Опорний конспект лекцій з курсу «Проектування комп'ютерних систем на програмованих логічних мікросхемах» з освітньо-кваліфікаційного рівня «спеціаліст» спеціальності «Комп'ютерні системи та мережі» / А.С. Ляпандра, І.М. Майків. – Тернопіль: ТНЕУ, 2012. – 53 с.
7. Richardson Iain E. The H.264 Advanced Video Compression Standard c / Iain E. Richardson. - United Kingdom : Richardson John Wiley & Sons, Ltd, 2011. - Режим доступу : <http://surl.li/oqcgq>. - Назва з екрана.
8. Ifeakor Emmanuel C. Digital Signal Processing. A practical approach [Електронний ресурс] / Emmanuel C. Ifeakor, Barrie W. Jervis. – Prentice Hill, Pearson Education Limited, 2012. - Режим доступу : <http://surl.li/oqcrg>. - Назва з екрана.
9. Bocharnikov V. Optimal discrete fuzzy filter of UAV's flight parameters / V. Bocharnikov, I. Bocharnikov // Proceedings of the NAU, 2012. - № 2. - P. 22-29.
10. Digital signals [Електронний ресурс] : веб-сайт. - Режим доступу : <http://surl.li/oqcum>. - Назва з екрана.
11. Бочарніков В.П. Частотно-часовий аналіз сигналів на основі функцій поведінки і арифметичних рядів: дослідження працездатності методу [Електронний ресурс] / В.П. Бочарніков // *Збірник наукових праць Центру воєнно-стратегічних досліджень*, 2019. - Режим доступу : <http://znp-cvds.nuou.org.ua/article/view/179056>. - Назва з екрана.
12. Комп'ютеризовані системи цифрової обробки сигналів [Електронний ресурс] : лекція. – Львів, 2008. - Режим доступу : <https://infopedia.su/17x51fa.html>. - Назва з екрана.
13. Smith Steven W. The Scientist and Engineer's Guide to Digital Signal Processing [Електронний ресурс] / Steven W. Smith. - Режим доступу : <http://www.dspguide.com>. - Назва з екрана.

14. Signal Processing for Communications [Електронний ресурс] / Paolo Prandoni, Martin Vetterli, Steven W. Smith. - Режим доступу : <http://www.sp4comm.org>. - Назва з екрана.
15. Nair B. Somanathan DIGITAL ELECTRONICS AND LOGIC DESIGN [Електронний ресурс] / B. Somanathan Nair. - Режим доступу : <http://surl.li/oqdfh>. - Назва з екрана.
16. Khanna V. K. Digital Signal Processing [Електронний ресурс] / V. K. Khanna. - S Chand & Co Ltd, 2009. - Режим доступу : <http://surl.li/oqdit>. - Назва з екрана.
17. Smillie G. Analogue and Digital Communication Techniques [Електронний ресурс] / G. Smillie. - Режим доступу : <http://surl.li/oqdkl>. - Назва з екрана.