

О. С. Вітер, І. І. Лагун, Я. А. Лагун  
Національний університет “Львівська політехніка”,  
кафедра комп’ютеризованих систем автоматики

## ОСОБЛИВОСТІ ПОБУДОВИ МІКРОКОНТРОЛЕРНИХ СИСТЕМ ЦИФРОВОГО ОПРАЦЮВАННЯ СИГНАЛІВ

© Вітер О. С., Лагун І. І., Лагун Я. А., 2019

Досліджено можливість створення недорогих та широкодоступних пристроїв для опрацювання сигналів у режимі реального часу для вирішення наукових та технічних завдань на основі мікроконтролера, які є альтернативою використання спеціалізованих цифрових сигнальних процесорів. Показано, що розрядність, тактова частота, швидкість виконання інструкцій є основними параметрами, які необхідні для реалізації алгоритмів цифрового опрацювання сигналів на основі мікроконтролера. Запропоновано використання алгоритмів дискретного вейвлет-перетворення для побудови систем моніторингу, діагностики та управління на основі мікроконтролера.

**Ключові слова:** обробка сигналів, мікроконтролер, вейвлет-перетворення.

The possibility of creating inexpensive and widely available real-time signal processing devices to solve scientific and technical tasks based on a microcontroller, which is an alternative to the use of specialized digital signal processors, has been investigated. It is shown that the bit rate, clock speed, instruction speed are the basic parameters that are necessary for the implementation of algorithms for digital signal processing based on the microcontroller. The use of discrete wavelet transform algorithms for the construction of monitoring, diagnostics, and control systems based on the microcontroller is proposed.

**Keywords:** signal processing, microcontroller, wavelet transform.

### Вступ

Цифрове опрацювання сигналів (ЦОС) – область науки та техніки, яка стрімко розвивалася останні півстоліття. Цей швидкий розвиток є результатом значного прогресу в цифровій комп’ютерній техніці та виробництві інтегральних мікросхем. Протягом останнього десятиліття з’явилися нові області застосування технологій ЦОС такі, як комп’ютерний зір, приладобудування та управління, компресія даних, розпізнавання та синтез мовлення, цифрове аудіо та відео, мобільна телефонія та мобільний інтернет. Паралельно із зростанням областей застосування методів ЦОС зростала обчислювальна потужність процесорів для реалізації алгоритмів ЦОС та доступу до великої кількості бібліотечних кодів, прикладних проектів та високопродуктивних процесорів.

### Загальний аналіз методів та засобів побудови систем ЦОС

Як відомо, існує три способи реалізації алгоритмів ЦОС [2]: апаратний; програмний і програмно-апаратний.

При апаратній реалізації використовуються різноманітні функціональні блоки, такі, як регістри, суматори, шифратори та дешифратори, лічильники, помножувачі тощо. Сукупність функціональних блоків та зв’язків між ними визначає алгоритм, що реалізується. Загалом апаратна реалізація орієнтована на вирішення вузькоспеціалізованих завдань та створення систем із жорсткою логікою. Перевагою апаратної реалізації є висока швидкодія, яка дає змогу опрацьовувати

сигнали при частоті дискретизації порядку десятків мегагерц. Недолік – значне споживання енергії та висока вартість проектних рішень.

Програмна реалізація представляє алгоритм ЦОС у вигляді програми, яку послідовно виконують один або кілька блоків. Програмування здійснюється тими самими мовами, що й інші наукові та інженерні завдання: С, BASIC та асемблер. Потужність і універсальність мови С робить її мовою комп'ютерних наук та професійних програмістів. Простота мови BASIC робить її ідеальною для вчених та інженерів, які не є фахівцями з програмування. Незалежно від використовуваної мови, більшість важливих проблем програмної реалізації алгоритмів ЦОС локалізовані на нижньому рівні одиниць та нулів: числа представлені бітовими шаблонами, похибка округлення в комп'ютерній арифметиці, швидкість обчислення різних процесорів тощо. Перевагами програмної реалізації є незмінна структура системи при різних алгоритмах і областях застосування, гнучкість, що дає змогу доволі легко змінювати алгоритми роботи системи корегуванням або зміною програми, здешевлення проектування, виготовлення та налагодження системи, оскільки замість приладу розробляється програма. Недоліком програмної реалізації є відносно низька швидкодія через послідовне виконання операцій програми в одному процесорі.

При апаратно-програмній реалізації частина функцій системи ЦОС виконується апаратно (аналого-цифрове та цифро-аналогове перетворення, множення, множення з накопиченням, приймання/передавання даних та ін.), інша частина функцій виконується програмно. Апаратно-програмна реалізація поєднала в собі переваги апаратної та програмної реалізації алгоритмів ЦОС.

Алгоритми ЦОС традиційно реалізуються за допомогою спеціальних мікросхем обробки цифрових сигнальних процесорів DSP, програмованих логічних схем FPGA або процесорів RISC [1].

Сьогодні більш поширеними є цифрові сигнальні процесори DSP. Однак, при виконанні алгоритмів ЦОС процесор повністю є завантажений, і спроба покласти на нього ще які-небудь функції призводить до різкого збільшення часу обробки. Мікросхеми FPGA характеризуються високою ємністю та швидкодією, а ступінь завантаженості мікросхеми визначається кількістю ресурсів (макромірок, DSP блоків, апаратних помножувачів і т.д.), які задіяні реалізованим у ній проектом. Тому додавання нових функцій практично ніяк не відбивається на швидкості роботи існуючих [5]. Якщо ж стоїть завдання обробки даних з плаваючою комою, перевагу знову отримують процесори DSP. Сучасні процесори здатні виконувати багато операцій з плаваючою комою за один такт, що недоступно для логічних схем FPGA. Отже, застосування FPGA вигідно при побудові систем, в яких потрібна багатоканальна обробка даних або багатоступенева фільтрація. Процесори DSP підходять для роботи зі складними алгоритмами з плаваючою комою, проектів, що не потребують великої обчислювальної потужності, а також для створення пристроїв з низьким енергоспоживанням [4].

Процесори загального призначення RISC, які використовують у персональних комп'ютерах, за своїми можливостями не відстають від процесорів DSP. Вони мають велику кількість регістрів і швидко арифметику та забезпечують апаратну підтримку основних операцій ЦОС.

Якщо для складних обчислювальних завдань ЦОС потрібен цифровий сигнальний процесор або FPGA, то для численних простих рішень із використанням вбудованих додатків доцільним буде використання мікроконтролера [5].

### **Особливості реалізації алгоритмів ЦОС на мікроконтролерах**

Мікроконтролери (MCU) – це пристрої загального призначення для управління, моніторингу та зв'язку із зовнішніми пристроями, які можуть бути адаптовані до завдань ЦОС за допомогою відповідного програмного забезпечення [5]. Вони використовують 4-, 8- або 16-бітні дані, пам'ять програм та даних, а також пристрої вводу/виводу, таймери, набір інструкцій. Також містять деякі додаткові програмовані периферійні пристрої, недоступні на процесорах DSP (наприклад, АЦП та ЦАП). Мікроконтролери є менш продуктивними, ніж DSP-процесори. Це пов'язано з невисокою швидкістю обробки та обмеженням пам'яті.

Використання мікроконтролерів загального призначення для обробки цифрових сигналів стало актуальним із появою процесорів із високою швидкістю. Оскільки більшість систем обробки сигналів складаються з хост-процесора та спеціалізованої мікросхеми DSP, використання одного мікроконтролера для виконання обох цих функцій забезпечує простіше і дешевше рішення. Крім того, однокристальна конструкція буде споживати менше енергії, що дасть змогу створювати системи з акумуляторним живленням [8].

Загалом типова система опрацювання сигналів [3] містить АЦП, ЦАП та процесор, який виконує алгоритм опрацювання сигналу, як показано на рис. 1.

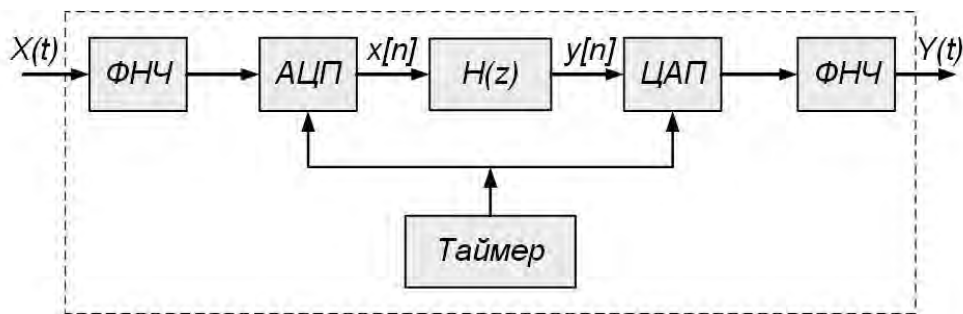


Рис. 1. Типова система опрацювання сигналів

Вхідний сигнал  $x(t)$  спочатку подається на вхідний низькочастотний фільтр (ФНЧ), функцією якого є обмеження сигналу частотою Найквіста (половина частоти дискретизації) для запобігання спотворенням. Потім сигнал дискретизується АЦП зі швидкістю, визначеною тактовою частотою вибірки, для отримання вхідної дискретної послідовності  $x(n)$ . Передавальна функція системи  $H(z)$  реалізує алгоритм ЦОС, тобто ставить у відповідність кожному значенню  $x(n)$  значення  $y(n)$ . Вихідна дискретна вибірка  $y(n)$  перетворюється на сигнал безперервного часу  $y(t)$  за допомогою ЦАП і вихідного фільтра низьких частот (ФНЧ).

Операції дискретизація сигналу, обчислення вихідної послідовності та вихідного аналогового сигналу повинні бути завершені протягом періоду дискретизації. Швидкість, з якою це можна зробити, визначає максимальну пропускну здатність, якої можна досягти в системі. Відносно низька швидкість більшості мікроконтролерів є головним обмеженням для використання їх у системах ЦОС, але висока швидкість виконання інструкцій мікроконтролером може забезпечити продуктивність, необхідну для реалізації систем із низькою пропускну здатністю. Крім того, вбудовані модулі АЦП та ЦАП забезпечують усі функції, необхідні для реалізації системи ЦОС на мікроконтролері.

Вбудований у мікроконтролер АЦП у більшості випадків використовує схему вибірки та зберігання, яка утримує вхідну напругу на сталому рівні до закінчення процесу перетворення сигналу на цифрову форму [6]. АЦП працює за принципом послідовного наближення, порівнюючи опорні рівні напруги із вхідним сигналом для отримання дискретної послідовності. Якщо потрібна підвищена швидкість перетворення, то можна знехтувати точністю, зчитуючи тільки старші біти дискретної послідовності та відкидаючи молодші біти перетворення. Перетворення опрацьованої дискретної послідовності вхідного сигналу на аналогову форму відбувається з використанням вбудованого в мікроконтролер блоку формування широтно-імпульсної модуляції (ШІМ). Вихід ШІМ разом із зовнішнім згладжувальним фільтром забезпечує цифро-аналогове перетворення. Форма ШІМ кодує потрібну амплітуду сигналу за допомогою співвідношення тривалостей імпульсів логічного «0» і логічної «1» свого виходу [8].

В основу усіх алгоритмів ЦОС покладено такі основні операції [1], як згортка, фільтрація, перетворення та модуляція. Необхідно зазначити, що для усіх основних операцій ЦОС необхідним є виконання тільки базових арифметичних дій, а саме множення, ділення, додавання, віднімання та

операції зсуву. Розрахунок вихідної вибірки  $y(n)$  згідно з алгоритмами ЦОС потребує операції множення з накопиченням (MAC). Це, як правило, одноциклова інструкція для процесорів DSP, але для виконання стандартним мікроконтролером може знадобитися більше циклів, оскільки вона повинна бути реалізована в програмному коді. Це суттєво обмежує сферу використання мікроконтролерів додатками, що потребують меншої потужності обробки (з погляду кількості операцій множення з накопиченням). Також для конкретного додатка можна оптимізувати операцію множення, підвищуючи так ефективність обробки. Наприклад, множення між змінною та константою часто може бути реалізовано швидше, ніж множення між двома змінними. Крім того, результат множення може становити лише декілька бітів точності, що зменшує кількість необхідних інструкцій.

Мікроконтролери в переважній більшості є основою реалізації сучасних систем моніторингу, діагностики та управління в таких областях, як радіолокація, технічна та медична діагностика, телеметрія та ін. Як правило, тут мають місце нестационарні випадкові сигнали. Такі системи можуть не тільки зчитувати та представляти сигнали у вигляді цифрових кодів, але й забезпечувати якісну обробку значень сигналів та здійснювати обмін інформацією через мережеві канали.

Відомо [2], що для аналізу та опрацювання нестационарних сигналів ефективним є метод дискретного вейвлет-перетворення (ДВП). В основу методу ДВП покладено багаторівневий розклад дискретної вибірки сигналу  $x[n]$  на  $j$  октавах, які мають значення  $j=1, \mathbf{L}, J$  згідно з виразом

$$x[n] = \sum_{k_1=1}^{\infty} \sum_{k_2=1}^{\infty} c_{j,k} h_j \hat{e}^{n - 2^j k_1} + \sum_{k_1=1}^{\infty} \sum_{k_2=1}^{\infty} b_{j,k} g_j \hat{e}^{n - 2^j k_1}, \quad (1)$$

де  $b_j$  – коефіцієнти апроксимації;  $c_j$  – деталізуючі коефіцієнти;  $g(k)$  та  $h(k)$  – низькочастотний та високочастотний фільтри відповідно.

Алгоритм Маллата або пірамідальний алгоритм – це ефективний метод реалізації дискретного вейвлет-перетворення, який найчастіше використовують як основу для апаратних реалізацій. Класична структура такого алгоритму передбачає рекурсивну реконструкцію сигналу в частотній області та представляє ітераційний алгоритм, який використовує фільтрацію, причому кількість ітерацій може бути будь-якою. За допомогою пірамідального алгоритму обробляється скінченний набір вхідних даних, які складаються з  $N$  вибірок, де  $N$  – кратне двом. Таке представлення розглядається як розмір вхідного блока. Вхідні дані обробляються за допомогою двох фільтруючих функцій, кожна з яких створює вихідний потік довжиною вдвічі меншою за довжину вхідного потоку. Перша половина вихідних даних перетворюється функціональною залежністю низькочастотного фільтра  $g(k)$  та описується таким виразом

$$y[n] = (x * g)[n] = \sum_{k=-\infty}^{\infty} x[k] g[2n - k], \quad (2)$$

У результаті отримують коефіцієнти апроксимації  $b_j$ . Одночасно вхідні дані перетворюються функціональною залежністю високочастотного фільтра  $h(k)$ , результатом чого будуть деталізуючі коефіцієнти  $c_j$ . Результати низькочастотних фільтрувань мають найбільший “інформаційний вміст” оригінального вхідного сигналу, а результати високочастотних фільтрувань містять різницю між справжнім вхідним сигналом та реконструйованим сигналом, якщо це відновлення базувалося лише на інформації, яку представляють елементи низькочастотного фільтрування. Згідно з теоремою Котельникова, відліки сигналів можна прорідити вдвічі. Якщо кількість вхідних вибірок  $N = 2^d$ , то можна здійснити максимум  $d$  рівнів перетворення. Однак обчислення повної згортки із подальшим проріджуванням є надлишковою тратою обчислювальних ресурсів, тому доцільніше є застосовувати ліфтингову схему.

Ліфтингова схема не потребує складних математичних обчислень, які необхідні за традиційним методом ДВП. Окрім значного зменшення потреб пам'яті і обчислювальної складності ДВП, ліфтингова схема дозволяє обчислення вейвлет-коефіцієнтів шляхом перезаписування

значень вхідної вибірки в комірках пам'яті. Завданням ліфтингової схеми є пошук представлення сигналу меншим числом коефіцієнтів, що еквівалентно збільшенню інтервалу дискретизації. Оскільки точного представлення сигналу досягти важко, то достатньо апроксимувати його з допустимою похибкою. Це означає, що різниця між вихідним сигналом і його апроксимацією повинна бути якомога меншою.

Перетворення сигналу складається з трьох фаз: розділення (S), прогнозування (P) та оновлення (U). На кожному етапі дискретне представлення сигналу  $x[n]$  поділяється на дві множини значень з парними і непарними номерами відліків аргументу  $n$ .

Побудову вейвлетів за ліфтинговою схемою [2] зображено на рис. 2.

Спочатку обчислюють звичайне вейвлет-перетворення (проста вейвлет-функція або багатофазне перетворення) над множиною вхідних даних  $b_{j-1}$ . Ця операція називається стадією розбиття або розділення. У результаті стадії розділення формуються парні та непарні вибірки даних, які поділяються відповідно на дві підмножини  $b_j$  і  $c_j$ . Після багатофазного фільтрування застосовується послідовність перетворення та подвійного перетворення. Спочатку прогнозується непарна підмножина  $c_j$  з парної підмножини  $b_j$  за допомогою перетворюючої послідовності  $p$ , тобто стадія «прогнозування». Передбачені значення віднімаються від фактичних значень непарних вибірок  $c_j$ . Результат цієї операції записується знову в непарну підмножину. Розраховують високочастотні вейвлет-коефіцієнти так

$$c_j(n) = c_{j-1}(n) - \sum_k p(k) b_{j-1}(n-k). \quad (3)$$

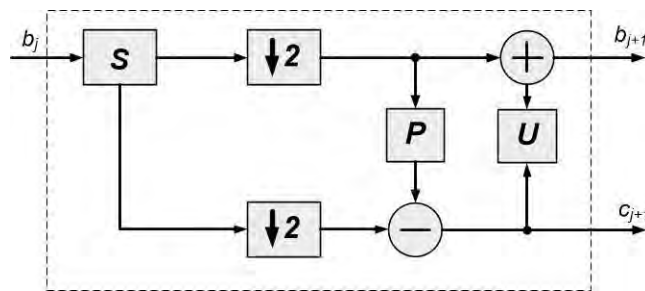


Рис. 2. Ліфтингова схема реалізації ДВП

Оскільки сумарна енергія, що міститься в сигналі, повинна бути однаковою до і після перетворення, застосовують корекцію. З цією метою множини парних нумерованих даних  $b_j$  корегують з використанням високочастотного набору даних  $c_j$ . У результаті множина даних  $b_j$  міститиме низькочастотні вейвлет-коефіцієнти

$$b_j(n) = b_{j-1}(n) + \sum_k u(k) c_{j-1}(n-k). \quad (4)$$

Результатом стандартної схеми перетворення є ненормалізоване перетворення. Для нормалізації результуючих даних використовують коефіцієнт перемноження  $K$ :

$$\begin{aligned} b_j(n) &= b_j(n) / K \\ c_j(n) &= K \times c_j(n) \end{aligned} \quad (5)$$

Для пірамідального алгоритму вейвлет-перетворення загальне число операцій множення визначають за виразом [2]

$$S_{II} = M \times N \sum_{i=1}^n \frac{1}{2^{i-1}}, \quad (6)$$

де  $M$  – порядок фільтра;  $N$  – кількість вибірок;  $n$  – кількість рівнів розкладу.

Для ліфтингової схеми загальне число операцій множення для  $n$  рівнів розкладу визначають як

$$S_{лсн} = (M_1 + M_2) N \prod_{i=1}^n \frac{1}{2^i}, \quad (7)$$

де  $M_1$  і  $M_2$  – кількість коефіцієнтів на етапах та оновлення відповідно.

Швидкодія ліфтингової схеми удвічі вища, ніж швидкодія пірамідального алгоритму. Це пояснюється тим, що в пірамідальному алгоритмі прорідження виконується в процесі фільтрації, а в алгоритмі за ліфтинговою схемою спочатку виконується прорідження, а потім фільтрація. Тобто, вектор вхідних даних є менший удвічі.

Часто сигнали від сенсорів містять високочастотні шуми або потребують для представлення значного обсягу дискретних значень, що передбачає використання алгоритмів дискретного вейвлет-перетворення для знешумлення або компресії сигналів. Програмна реалізація алгоритму дискретного вейвлет-перетворення передбачає такі кроки, які необхідно виконати перед впровадженням у мікроконтролер:

1. Визначення виду застосування: декомпозиція та/або реконструкція, фільтрація, компресія. Чим складніший алгоритм ЦОС, тим більше необхідно апаратних ресурсів, пам'яті, швидкості.
2. Встановлення рівня декомпозиції.
3. Визначення оптимальної вейвлет-функції.

Ефективне вирішення цих кроків шляхом попереднього моделювання розроблюваної мікроконтролерної системи дасть змогу забезпечити простоту реалізації та ефективність її роботи.

## Висновки

Проаналізовано особливості реалізації системи обробки сигналів на основі мікроконтролера для побудови спеціалізованих систем моніторингу, діагностики та управління з використанням алгоритмів ДВП. Мікроконтролери можуть забезпечувати необхідну продуктивність для реалізації операцій цифрової обробки сигналів з низькою пропускнуою здатністю. Використання цих пристроїв у системах ЦОС на заміну спеціалізованим компонентам, які зазвичай виконують такі завдання, може мати перевагу завдяки економії витрат та енергії. Однак ефективність мікроконтролерів обмежується конкретним застосуванням внаслідок набору інструкцій, які запрограмовані безпосередньо на мікросхемі.

## Список літератури

1. Айфичер Э., Джервис Б. *Цифровая обработка сигналов: практический подход* / Э. Айфичер, Б. Джервис. М.: Вильямс, 2004. 992 с.
2. Наконечний А. Й. *Теорія малохвильового (wavelet) перетворення та її застосування* / А. Й. Наконечний. Львів: Фенікс, 2001. 278 с.
3. Солонина А. И., Улахович Д. А., Арбузов С. М., Соловьева Е. Б. *Основы цифровой обработки сигналов. Курс лекций: учебн. пособие* / А. И. Солонина, Д. А. Улахович, С. М. Арбузов, Е. Б. Соловьева. – СПб.: БХВ Петербург, 2005. 768 с.
4. Эдмунд Лэй. *Цифровая обработка сигналов для инженеров и технических специалистов* / Э. Лэй. – М.: Группа ИДТ, 2007. 336 с.
- 5 Lyons, R. *Understanding Digital Signal Processing*, 3 nd ed. Upper Saddle River, NJ: Prentice-Hall, 2011, p.954.
6. Pinto, S.E. & Mendoza, Luis Enrique & Florez, Elkin. *Compressive sensing in FPGA and microcontroller. International Journal of Engineering and Technology*. 2016, 7(6). Pp. 2202–2206.
7. AN219: *Digital Signal Processing with the PIC16C74*. Silicon Labs, <https://www.silabs.com/documents/public/application-notes/an219.pdf>
8. AN4841: *Digital signal processing for STM32 microcontrollers using CMSIS*. STMicroelectronics. [https://www.st.com/resource/en/application\\_note/dm00273990.pdf](https://www.st.com/resource/en/application_note/dm00273990.pdf).

### References

1. Emmanuil Ayficher, Barri Dzhervis. *Tsifrovaya obrabotka signalov: prakticheskiy podhod* / E. Ayficher, B. Dzhervis. M.: Vilyams, 2004. 992 s.
2. Nakonechnyi A. Y. *Teoriia malokhvylovoho (wavelet) peretvorennia ta yii zastosuvannia* / A. Y. Nakonechnyi. Lviv: Feniks, 2001. – 278 s.
3. A. I. Solonina, D. A. Ulahovich, S. M. Arbuzov, E. B. Soloveva. *Osnovyi tsifrovoy obrabotki signalov. Kurs lektsiy: Uchebnoe posobie* / A. I. Solonina, D. A. Ulahovich, S. M. Arbuzov, E. B. Soloveva. SPb.: BHV Peterburg, 2005. 768 s.
4. Edmund Ley. *Tsifrovaya obrabotka signalov dlya inzhenerov i tehnikeskikh spetsialistov* / E. Ley. M.: Gruppya IDT, 2007. 336 s.
- 5 Lyons, R. *Understanding Digital Signal Processing, 3 nd ed.* Upper Saddle River, NJ: Prentice-Hall, 2011, p. 954.
6. Pinto, S.E. & Mendoza, Luis Enrique & Florez, Elkin. *Compressive sensing in FPGA and microcontroller. International Journal of Engineering and Technology.* 2016, 7(6). Pp. 2202–2206.
7. AN219: *Digital Signal Processing with the PIC16C74.* Silicon Labs, <https://www.silabs.com/documents/public/application-notes/an219.pdf>
8. AN4841: *Digital signal processing for STM32 microcontrollers using CMSIS.* STMicroelectronics. [https://www.st.com/resource/en/application\\_note/dm00273990.pdf](https://www.st.com/resource/en/application_note/dm00273990.pdf).