

## АРХІТЕКТУРА КОНВЕЕРНОГО ПРОЦЕСОРА ДИСКРЕТНИХ ВЕЙВЛІТНИХ ПЕРЕТВОРЕНЬ

© Ерметов Ю.О., Мельник А.О., Палій С.О., 2001

**Розглянуто алгоритм дискретного вейвлітного перетворення, основні проблеми, що виникають при його реалізації, а також запропонована архітектура процесора дискретних вейвлітних перетворень.**

**The algorithm of discrete wavelet transform and main problems that arise during its applications are considered. The architecture of discrete wavelet transform processor is proposed.**

### Вступ

Останнім часом для задач цифрової обробки сигналів ефективно використовують вейвлітне (малохвильове) перетворення та його дискретний аналог – дискретне вейвлітне перетворення (надалі ДВП) [2]. Реалізація ДВП зручна тим, що не потрібно проводити обчислення над комплексними числами і залежно від використаних фільтрів його можна використовувати як для аналізу сигналу, так і для стиску. Розглядаючи літературу, можна знайти багато різних архітектур процесора ДВП, які мають свої переваги та недоліки. Але ці архітектури є суто теоретичними, тобто жодних даних про їх апаратну реалізацію немає. В даній статті запропонована архітектура, яка спроектована з врахуванням в першу чергу можливості її апаратної реалізації.

### 1. Основний алгоритм

Дискретні вейвлітні перетворення можна описати такими формулами:

**Пряме перетворення (декомпозиція):**

$$W_L^j(n) = \sum_{m=0}^M W_L^{j-1}(2n+m) \cdot g(m) \quad (1)$$

$$W_H^j(n) = \sum_{m=0}^M W_L^{j-1}(2n+m) \cdot h(m)$$

**Обернене перетворення (реконструкція):**

$$W_L^j(n) = \sum_{m=0}^M W_L^{*j+1}(n+m) \cdot g'(m) + \sum_{m=0}^M W_H^{*j+1}(n+m) \cdot h'(m) \quad (2)$$

$$W_L^{*j}(2k) = W_L^j(k); \quad W_L^{*j}(2k+1) = 0; \quad W_H^{*j}(2k) = W_H^j(k); \quad W_H^{*j}(2k+1) = 0;$$

де –  $W_L^0(n) = X(n)$ ;  $n = 0 \dots \frac{N}{2^j} - 1$ ;  $k = 0 \dots \frac{N}{2^{j+1}} - 1$ ;  $j = \log_2 N$ ,  $W_L^j$  – апроксимаційні коефіцієнти  $j$ -го ступеня;  $W_H^j$  – деталізуючі коефіцієнти  $j$ -го ступеня;  $N$  – довжина вхідного сигналу;  $g(n)$ ,  $h(n)$  – фільтруючі послідовності прямого перетворення;  $g'(n)$ ,  $h'(n)$  – фільтруючі послідовності оберненого ДВП;  $M$  – довжина фільтруючих послідовностей  $g(n)$  та  $h(n)$  ( $g'(n)$ ,  $h'(n)$ ) відповідно;  $j$  – номер ступеня перетворення [1].

Відповідно до наведених вище формул структурна схема прямого ДВП складається з двох базових операцій – фільтрації та прорідження. Прорідження передбачає вилучення кожного другого відліку сигналу.

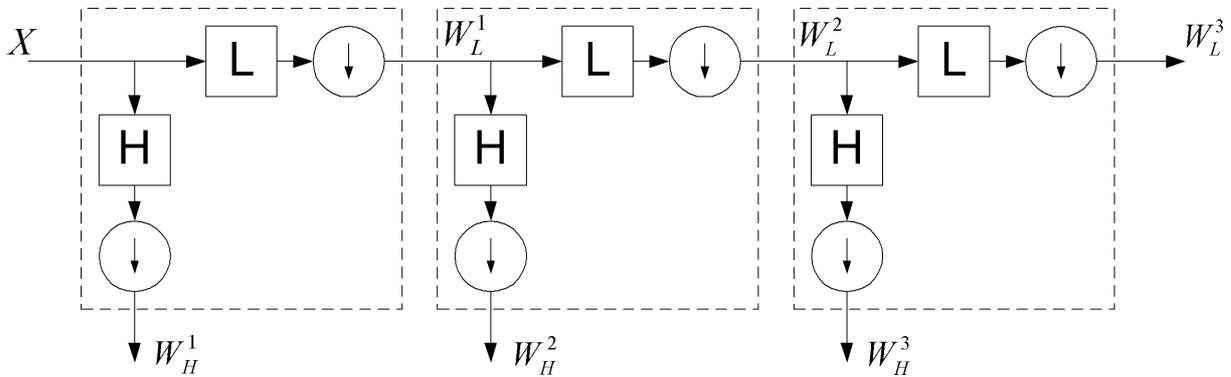


Рис. 1. Структурна схема ДВП для  $N=8$

Суть пірамідального алгоритму полягає в тому, що деталізуючі коефіцієнти ( $W_H$ ) видаються як результуючі, а апроксимаційні ( $W_L$ ) є вихідними даними для наступного етапу перетворення. Структурна схема перетворення, яке складається з трьох стадій ( $j=0\dots3$ ,  $N=8$ ), наведена на рис. 1. Аналогічна структура буде і при більшому значенні  $N$ . Тут літерою  $H$  позначено деталізуючий фільтр, літерою  $L$  – апроксимуючий фільтр. Прорідження показано стрілкою вниз.

Обернене перетворення складається з трьох базових операцій – розширення, фільтрації і підсумовування. Розширення полягає в доповненні нулями простору між всіма відліками вхідного сигналу. На рис. 2 наведено структурну схему оберненого ДВП для довжини вхідного сигналу  $N=8$  [3]. Аналогічно до попередньої схеми літерою  $H$  позначено деталізуючий фільтр, літерою  $L$  – апроксимуючий фільтр. Розширення сигналу показано стрілкою вверх.

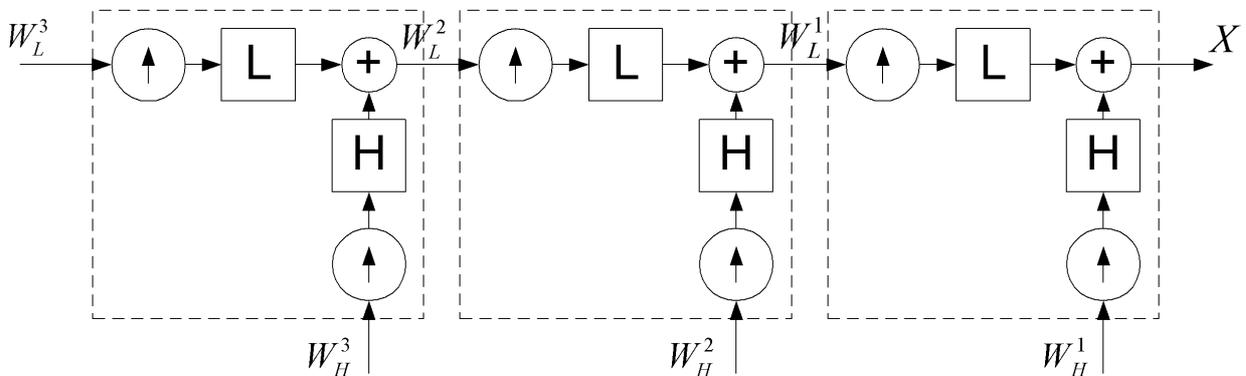


Рис. 2. Структурна схема оберненого ДВП для  $N=8$

## 2. Огляд існуючих архітектур процесорів ДВП

Детальний огляд існуючих архітектур здійснено в праці [3]. Зупинимось на найбільш вдалим варіантах архітектур. Архітектура Lang [3] має часову затримку  $\approx 2N=O(N)$ , добру масштабованість, але вона розрахована на нескінченний потік вхідних даних. Архітектура Chuang [6] – це систолічний масив фільтрів. Вона має типову для систолічних масивів регулярність та масштабованість. Недоліками є прив'язаність до конкретних значень коефіцієнтів і довжини фільтра, а також невисокий коефіцієнт використання апаратури. Так, фільтр першого етапу використовується на 50 %, другого – на 25 %, третього – на 12,5 % і так далі. Ще один оригінальний варіант – архітектура Rioul and Duhamel [7]. Її характеризує використання операції швидкої конволюції для зменшення кількості множень. Основним недоліком є те, що дана архітектура розрахована на довгі фільтруючі послідовності, тоді як на практиці застосовуються короткі (4-12 елементів) послідовності. Доволі цікава напівсистолична архітектура Vishwanath [8, 9]. Ця архітектура має два окремі фільтри H і L. Позитивними якостями такої архітектури є її гнучкість та масштабованість, але недоліком є невисокий коефіцієнт завантаженості апаратури. Aware [10] – єдина архітектура, про яку відомо, що вона була апаратно реалізована. Ця архітектура передбачає можливість занесення коефіцієнтів фільтра, що є дуже хорошим параметром. Особливості побудови цієї архітектури не розголошуються через комерційні причини.

## 3. Пропонована архітектура процесора ДВП

### 3.1. Часові схеми реалізації перетворення

Одним з основних питань при побудові конвеєрного процесора ДВП є порядок обчислення коефіцієнтів. Існує дві часові схеми обчислень. Спочатку розглянемо схему обчислень за готовністю даних. Зміст цієї схеми полягає в тому, що коефіцієнт наступного ступеня обчислюється зразу після того, як готові коефіцієнти попереднього ступеня, які є вихідними для його обчислення. Часова структура такого обчислення наведена на рис. 3.

Як приклад використано 8-точкове ДВП ( $N=8$ ) з довжиною фільтруючої послідовності 4 елементи ( $M,S=3$ ). Доповнення сигналу полягає в тому, що сигнал вважається умовно періодичним і для ліквідації краєвих ефектів використовуються перші відліки сигналу. Показано опрацювання лише одного пакету даних. Насправді опрацювання даних відбувається потоково. Тобто після закінчення першої вхідної послідовності зразу починається друга, за другою третя і так далі. Така часова схема є зручною через невелику складність пристрою керування. Стратегія керування полягає в тому, що кожен другий відлік виконується обчислення коефіцієнта першого рівня, кожен четвертий відлік – обчислення коефіцієнта другого рівня, кожен восьмий – третього рівня і так далі. Ще однією позитивною рисою такої схеми є відсутність потреби у великому об'ємі пам'яті для зберігання проміжних коефіцієнтів (необхідно зберігати  $2(M-1)\log_2 N$  елементів). Але є потреба впорядковувати вихідні коефіцієнти, тобто потрібно використовувати сортувальну пам'ять [5]. Недоліком також є велика затримка від першого вхідного до першого вихідного коефіцієнтів, яка становить  $2N+\log_2 N-1$  тактів.

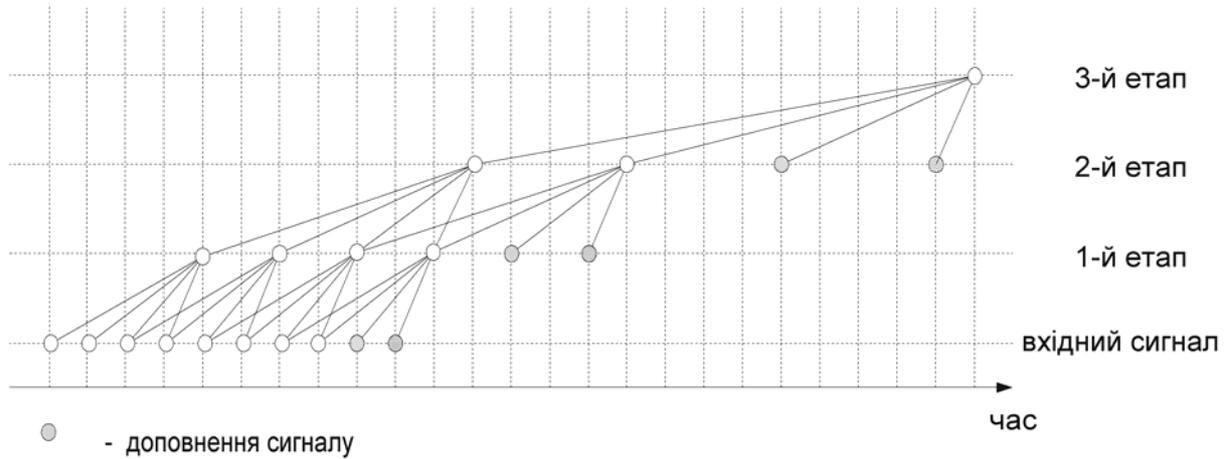


Рис. 3. Схема обчислення ДВП за готовністю даних

Іншим варіантом є поетапна часова схема обчислень, наведена на рис. 4. За цією схемою коефіцієнти наступного етапу обчислюються після того, як всі елементи попереднього етапу є обчисленими. Такий підхід значно зменшує затримку від першого вхідного до першого вихідного коефіцієнта (затримка становить  $N$  тактів). Оскільки коефіцієнти обчислюються в порядку видачі, то не потрібно використовувати сортувальну пам'ять. Така схема вимагає додаткової пам'яті для зберігання сигналу та коефіцієнтів в процесі обчислення (потрібно зберігати  $N+N/2$  елементів). Через те, що за такою схемою затримка значно менша, надалі розглядаємо реалізацію саме такої часової схеми.

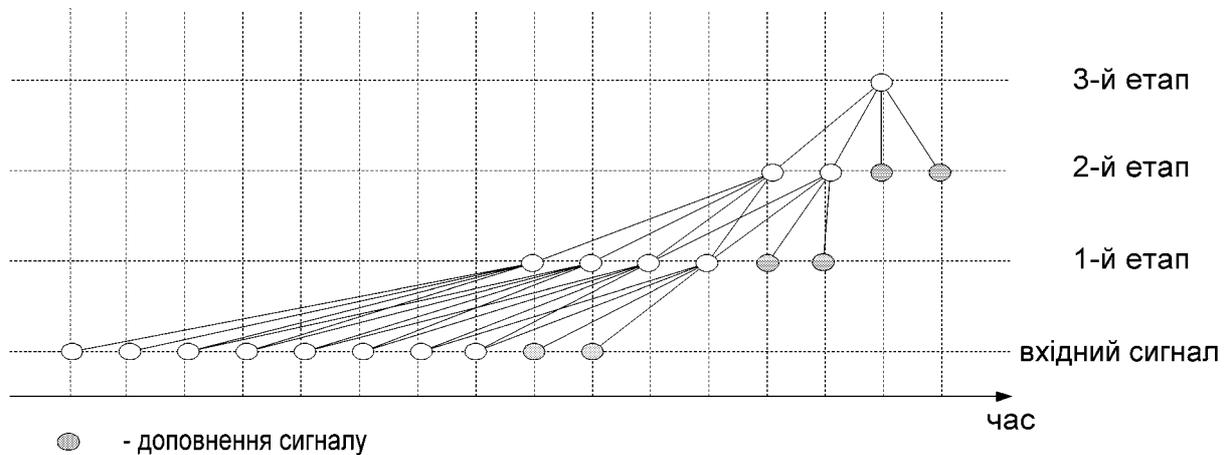


Рис. 4. Схема поетапного обчислення ДВП

### 3.2. Пропонована архітектура

Виходячи з вищевказаних міркувань, можна запропонувати архітектуру інформаційного тракту процесора прямого ДВП, наведену на рис. 5. Інформаційний тракт функціонує так: спочатку вхідний сигнал записується в пам'ять MEM1 (для цього потрібно  $N$  тактів). Водночас протягом  $N/2$  тактів з пам'яті MEM2 здійснюється читання. Зчитані дані надходять на фільтри  $H$  та  $L$ . Пройшовши через деталізуючий фільтр ( $H$ ), дані видаються назовні. Пройшовши апроксимуючий фільтр ( $L$ ), дані записуються до пам'яті MEM3. Тоді за  $N/4$  тактів дані зчитуються з MEM3, проходять через фільтри і записуються до MEM2. Це

продовжується доти, доки не буде досягнуто кінця перетворення. Після того дані з MEM3 (1 слово –  $W_L^K$ ) через MUX2 видаються назовні. Цей процес займе  $N$  тактів ( $N/2+N/4+N/8+\dots+2+1+1=N$ ), – рівно стільки триває запис вхідного сигналу до MEM1. Коли MEM1 є заповненою повністю, вхідні дані починають записуватись до MEM2, а зчитування починає відбуватись з MEM1 (аналогічно вищеописаній послідовності, але замість MEM2 використовується MEM1). Комутатор та мультиплексор забезпечують розділення в часі читання та запису до однієї пам'яті, мультиплексор на виході (MUX2) призначений для подачі на вихід останнього апроксимаційного коефіцієнта (в даному випадку  $W_L^K$ ).

Основними перевагами такої архітектури є те, що легко можна змінити елементи фільтрів і це не викличе необхідності перебудовувати всю архітектуру. Також така архітектура може бути легко реалізована для різних довжин вхідної послідовності.

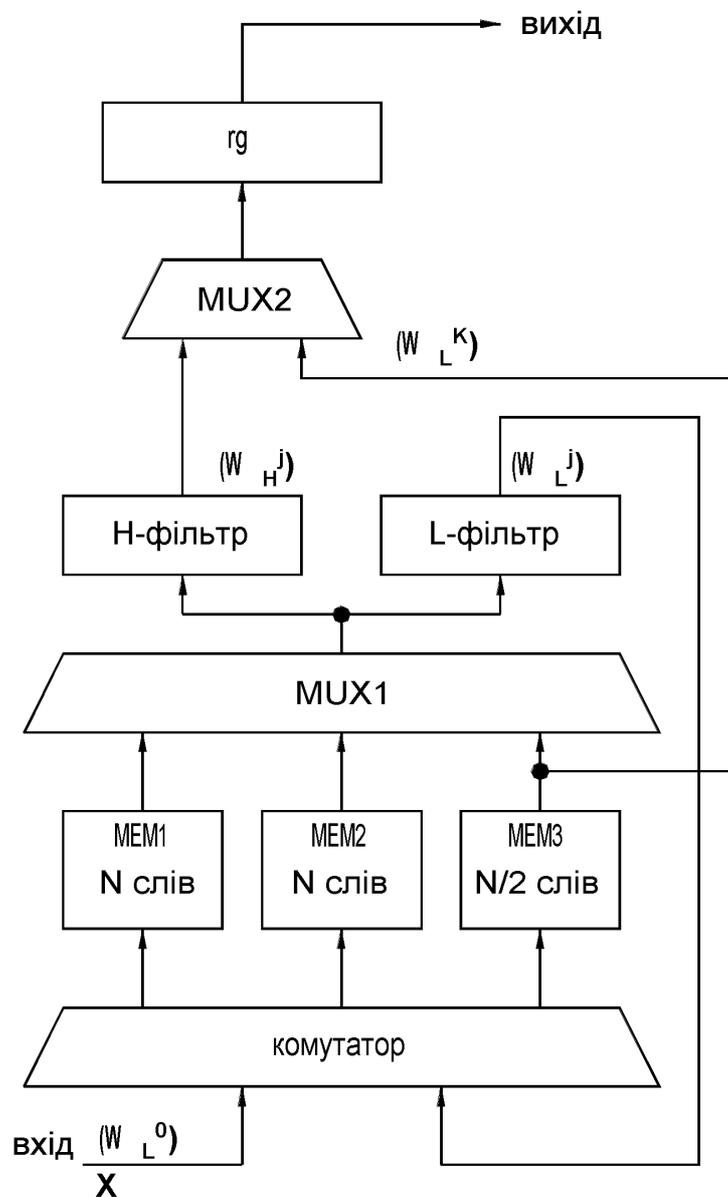


Рис. 6. Архітектура процесора прямого ДВП

### Висновки

Оскільки популярність вейвлітного перетворення для обробки сигналів постійно зростає, виникає потреба в розробці апаратних засобів, що реалізують дане перетворення. В літературі є багато пропозицій щодо побудови їх архітектури. Але вони мало враховують реальну можливість побудови апаратних засобів і є швидше науковою основою для апаратної реалізації ДВП. Розглянуто алгоритм вейвлітних перетворень, проаналізовано деякі питання, що стосуються побудови архітектури процесора ДВП і запропоновано архітектуру, придатну для апаратної реалізації. Пропонована архітектура передбачає можливість використання різних фільтруючих послідовностей, що дає змогу застосовувати її для різних потреб (стиску чи аналізу) – для цього не потрібно змінювати архітектуру цілого процесора, необхідно лише змінити деталізуючий та апроксимуючий фільтри. Також така архітектура може бути реалізована для різних довжин вхідної послідовності, при цьому змінюються лише розміри пам'яті (MEM1, MEM2, MEM3).

1. Ronald A. DeVore and Bradley J. Lucier. *Wavelets*. Cambridge University Press, v. 1 (1992), P. 1-56  
 2. Наконечний А.Й., Наконечний Р.А. Пірамідальний алгоритм як основа дискретного малохвильового (wavelet) перетворення сигналів // *Вимірювальна техніка та метрологія*, № 53, 1998. – С.26 – 32  
 3. Robert Lawrence Lang. *Parallel VLSI architectures for one – two-, three-dimensional discrete wavelet transforms*. A thesis submitted in partial fulfillment of the requirement for the degree of Doctor of Philosophy. The University of Newcastle, March 1996.  
 4. Edwards T. *Discrete wavelet transforms: Theory and implementation*. Technical report, Information Systems Laboratory, Stanford University, September 1992.  
 5. Мельник А.О. *Спеціалізовані комп'ютерні системи реального часу*. Львів, 1996. – С.6.  
 6. Henry Y.H. Chuang, Hyung Kim, and Ching-Chung Li. *Systolic architecture discrete wavelet transforms: From single chip implementations to mapping SIMD array computers*. *IEEE Transactions on Signal Processing*, 43(3):759 – 771, 1992.  
 7. Olivier Rioul and Pierre Duhamel. *Fast algorithm for discrete and continuous wavelet transforms*. *IEEE Transactions on Information Theory*, 38(2):569 – 568. – March 1992.  
 8. Vishwanath Mohan, Owens Robert M. and Irwin M. Jane. *An efficient systolic architecture for QMF filter bank trees*. In *VLSI Signal Processing V*, Oct 1992. – P. 175 – 184.  
 9. Mohan Vishwanath. *The recursive pyramid algorithm for the discrete wavelet transform*. *IEEE Transactions on Sygnal Processing*, 42(3):637 – 676, March 1994.  
 10. Aware Inc. *Wavelet transform processor chip: User's guide*, p/n 22500. Technical report AD940216, Aware Inc. – 1994.