

І. О. Процько, Р. В. Рикмас

Національний університет “Львівська політехніка”, ШПТ,
кафедра інформаційних систем і технологій

АВТОМАТИЧНЕ ГЕНЕРУВАННЯ ЕФЕКТИВНИХ АЛГОРИТМІВ ДКП-II НА ОСНОВІ ЦИКЛІЧНИХ ЗГОРТОК

© Процько І. О., Рикмас Р. В., 2017

Розглянуто програмну реалізацію дискретного косинусного перетворення другого типу ДКП-II на основі циклічних згорток. Визначено етапи автоматичного генерування ефективних алгоритмів для обчислення ДКП-II довільного обсягу N . Алгоритм ДКП-II має швидшу програмну реалізацію для коротких обсягів порівняно з відомою бібліотекою FFTW.

Ключові слова: дискретне косинусне перетворення (ДКП), блочно-циклічна структура, твірний масив, циклічна згортка, автоматичне генерування алгоритмів.

I. Protsko, R. Rykmas,

Lviv Polytechnic National University
Department of Informational Systems and Technology

AUTOMATIC GENERATION OF THE EFFICIENT ALGORITHMS OF DCT-II BASED ON CYCLIC CONVOLUTIONS

© Protsko I., Rykmas R., 2017

Program of efficient implementation the discrete cosine transform of type-II using cyclic convolutions have been considered. The stages of automatic code generation the algorithms for the computation of DCT-II an arbitrary size N have been determined. The algorithm of DCT-II presents a better program performance for short sizes of transform, than known FFTW library.

Key words: discrete cosine transform (DCT), block-circular structure, hashing array, cyclic convolution, program implementation, automatic code generation.

Вступ

Алгоритми швидкого перетворення класу Фур'є є одним з найважливіших досягнень застосування ефективного обчислення для науково-технічних задач. Багато прикладних задач потребують, щоб трансформація з часової області в частотну і навпаки була обчислена якомога швидше. Опубліковано багато робіт про дослідження і розроблення швидких алгоритмів класу Фур'є. Деякі з них описують методи та їх модифікації для розроблення базових швидкодіючих алгоритмів, щоб зробити їх ефективнішими або пристосованішими для впровадження у сучасні багатопроесорні комп'ютерні системи.

Аналіз літературних джерел

Одним з підходів, серед багатьох, у теорії швидких алгоритмів цифрової обробки сигналів є структурування базисних матриць перетворень у формі блочно-циклічних структур [1]. У 1968 р. Радер показав, як ДПФ для обсягу N , де N – просте число, переформулювати в циклічну згортку обсягу $(N-1)$ [2]. Подальший розвиток підходу Радера представлено алгоритмом Вінограда

перетворення Фур'є (АВПФ) для перетворення обсягу $N=p^n$ цілого степеня простого числа [3]. Вагомий внесок в аналіз та розвиток цього підходу, що приводить алгоритм дискретного перетворення класу Фур'є до обчислення через циклічні згортки, зробили Р. Блейхут та Г. Ньюбаумер у книгах [4, 5].

Подальші дослідження оброблення сигналів стосувались автоматичного генерування алгоритмів класу Фур'є, оптимізації програмного коду та його адаптації до обчислювальної платформи. Розроблено стандартні бібліотеки ефективних алгоритмів дискретного перетворення класу Фур'є для використання у різноманітних програмах. Широко застосовується вільнодоступна бібліотека програм FFTW (Fastest Fourier Transform in the West), яку розробили С. Г. Джонсон і М. Фріго [6]. FFTW обчислює ДПФ для вхідних комплексних даних, дійсних даних, парних або непарних симетричних дійсних даних (ці симетричні перетворення відомі як дискретне косинус-або синус-перетворення, відповідно) і дискретне перетворення Хартлі (ДПХ) дійсних вхідних даних. Сучасна реалізація FFTW виконується відповідно до так званого "плану", що розробляють за багатьма критеріями послідовності програмного коду мовою С. Багато інших бібліотек, такі як Spiral FFT [7], Nukada FFT [8] і CUDA FFT (CUFFT) [9] виконання дискретного перетворення класу Фур'є, дотримуються аналогічного підходу для генерування алгоритмів, оптимізації та їх виконання. У 2003 р. Intel випустила свою комерційну бібліотеку Math Kernel [10] виконання ДПФ, яка здійснює підлаштування під апаратно-програмні особливості конкретної платформи. Цілями цих проектів є автоматичне генерування, оптимізація та адаптація алгоритмів дискретного перетворення класу Фур'є для їх ефективної реалізації у комп'ютерних системах.

Постановка проблеми

Інтенсивний розвиток інформаційних технологій задає вищі вимоги щодо функціональності, швидкодії та економічності реалізації алгоритмічних, програмно-апаратних засобів дискретних перетворень класу Фур'є. Для розв'язання цих задач застосовують системи автоматичної генерації швидких алгоритмів дискретних перетворень класу Фур'є. Ці системи охоплюють весь цикл автоматизації, що починається від задання виду та обсягу дискретного перетворення, обчислювальної платформи і до отримання результатів адаптованої імплементації [11].

Етапи автоматичної генерації алгоритмів ДКП-II на основі циклічних згорток

Швидкі алгоритми дискретного перетворення класу Фур'є широко досліджуються для дійсних перетворень з косинусними, синусовими, косинусними-базисними функціями. Особливо широко використовують дискретні косинусні перетворення (ДКП). Найпоширенішим варіантом дискретного косинусного перетворення є другий тип ДКП-II, який часто називають просто "ДКП", зворотним перетворенням якого є третій тип ДКП-III. Ефективні алгоритми косинусних перетворень застосовують у кодуванні сигналів, розпізнаванні образів, цифровій інтерполяції, стисненні даних, адаптивній інтерполяції сигналів.

Дискретне косинусне перетворення ДКП-II подано в матричній формі:

$$X = W^* x, \quad (1)$$

де $W (N \times N)$ – базисна квадратна матриця; $x (N)$ і $X (N)$ – стовпці вхідних і вихідних даних; N – обсяг перетворення. Елементи ДКП-II базисної квадратної матриці W дорівнюють:

$$W(k, n) = c(n) x(n) \cos[(2k+1)n\pi/2N], \quad (2)$$

де $c(0) = (1/N)^{1/2}$, $c(n) = (2/N)^{1/2}$, $k, n = 0, 1, \dots, N-1$.

Успішно використовують ДКП завдяки існуванню так званих швидких перетворень (ШКП). Узагальнену схему автоматичного генерування ефективних алгоритмів ДКП-II з використанням циклічних згорток описано в [12, 13].

Відповідно до узагальненої схеми генерація автоматичного коду містить вісім основних етапів (див. рисунок). Спочатку необхідно задати обсяг N дискретного перетворення ДКП-II.

Після цього визначається період повторення T базисних функцій ДКП для значення N обсягу перетворення, що дорівнює

$$T = 4N. \quad (3)$$

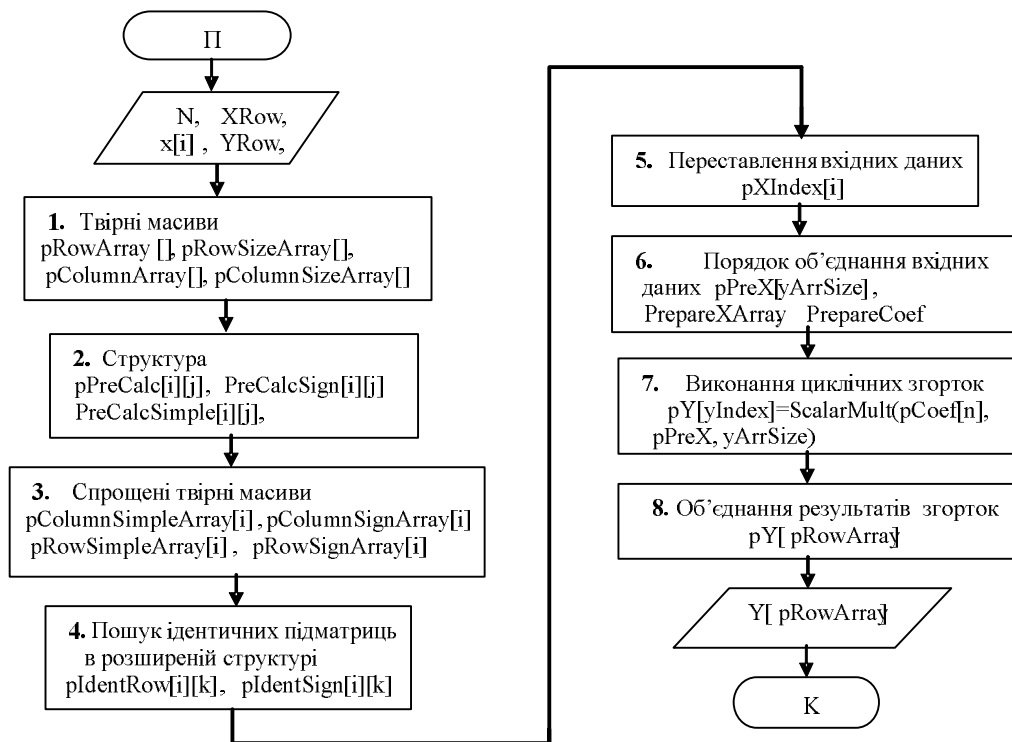


Рис. 1. Етапи автоматичного генерування алгоритмів ДКП-II

Першим етапом схеми є визначення твірних масиву $Hr(n)$, $Hc(n)$ рядка та стовпця, де n – це номер рядка та стовпця масивів, що не належать до множини канонічного розкладу величини обсягу N перетворення. Елементи твірних масиву – цілі значення від 1 до $(4N-1)$. Твірні масиви $Hr(n)$, $Hc(n)$ можна отримати, використовуючи підстановку першого і відповідного n -рядка/стовпця для матриці $c_{k,n}$ аргументів базисної функції ДКП-II

$$c_{k,n} = ((2n + 1) k) \bmod 4N, k=0,1,\dots, 4N-1. \quad (4)$$

У результаті визначаємо підстановки, отримані у формі циклічного розкладу підстановок у вигляді цілочислових масивів $pRowArray$ [], $pColumnArray$ [], які відповідають $Hr(n)$, $Hc(n)$. Твірні масиви $Hr(n)$, $Hc(n)$ стисло описують циклічні підматриці у структурі базисної квадратної матриці ДКП-II, що приводить до обчислення циклічних згорток над значеннями підматриць базисної матриці W та вхідними даними x .

Основним етапом автоматичного генерування алгоритму є аналіз структури базисної матриці ДКП-II. Визначають ідентичні підматриці під час аналізу за відповідними параметрами $Hr(n)$ та $Hc(n)$, порівнюючи знаки та значення перших спрощених елементів $s_{i,j}$ $\underline{c}_{i,j}$ циклічних підматриць. Спрощений елемент $\underline{c}_{i,j}$, відповідно до властивості симетрії базису ДКП-II, визначається за послідовними виконаннями:

$$c_{k,n} = 4N - [(c_{k,n}) \bmod 4N], \text{ якщо } [(c_{k,n}) \bmod 4N] > 2N;$$

$$\underline{c}_{k,n} = 2N - \{4N - [(c_{k,n}) \bmod 4N]\}, \text{ якщо } \{4N - [(c_{k,n}) \bmod 4N] \text{ or } c_{k,n}\} > N, \quad (5)$$

інакше $\underline{c}_{k,n} = c_{k,n}$.

Спрощені матриці аргументів доповнюють знаки $s_{i,j}$ косинуса, визначені нерівностями:

$$s_{i,j} = \begin{cases} +1, & \text{якщо } 3T < c_{i,j} < T, \\ 0, & \text{якщо } c_{i,j} = T, 3T, \\ -1, & \text{якщо } T < c_{i,j} < 3T. \end{cases} \quad (6)$$

У результаті аналізу структури базисної матриці ДКП-II використовує перші спрощені елементи підматриць $s_{i,j}$ $\underline{c}_{i,j}$, що описують розташування підматриць у базисній матриці, визначає подальшу автоматичну генерацію алгоритму обчислення ДКП-II довільного обсягу N . Отже, подальша частина програми містить такі 6, 7, 8 (рис. 1) три послідовні етапи:

- об'єднання вхідних даних $x(i)$ за однаковими горизонтально розміщеними підматрицями;
- обчислення мінімальної кількості циклічних згорток з використанням швидких алгоритмів згортки;

- об'єднання отриманих результатів циклічних згорток для обчислення вихідних даних.

Отримано коефіцієнти ДКП-II у послідовності, заданій у `pRowArray`, відповідно до послідовності елементів твірного масиву $Hr(n)$.

Аналіз виконання програми DCT-II (CC)

Реалізація програми DCT-II (CC) мовою C++ для автоматичного генерування і виконання ефективних алгоритмів ДКП-II з використанням циклічних згорток підтверджує методологічні концепції запропонованого в [14, 15] підходу. Тому основні частини реалізації програми DCT-II (CC) (рис. 2) спочатку потребують введення обсягу N та вхідних даних $x[0,1,\dots,N-1]$ перетворення, номер $Xrow$ рядка для визначення твірних масивів $Hr(n)$, $Hc(n)$ вводять для задання версії ефективного алгоритму.

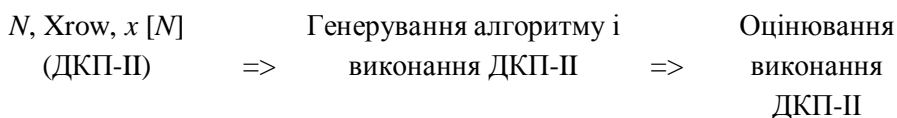


Рис. 2. Частина реалізації програми DCT-II

Між етапами автоматичного генерування ефективного алгоритму послідовно виконується обчислення ДКП-II для заданого обсягу N перетворення.

Для порівняння результатів оцінювання розробленої програми DCT-II(CC) з використанням циклічної згортки вибрано FFTW, що поширюється як вільне програмне забезпечення. Бібліотека FFTW генерує алгоритми для обчислення дійсних дискретних косинусних перетворень і дискретних синусних перетворень типів I-IV. Програмне забезпечення FFTW дає змогу використовувати його в більшості комп'ютерних архітектур без змін. У FFTW значну частину критично важливого коду автоматично створює спеціальний компілятор, який називається *genfft*. FFTW використовує різні підходи для автоматичного генерування програмного коду реалізації ДКП. Цей код оптимізовано для роботи з більшістю наявних компіляторів і платформ і не налаштовано на якусь конкретну платформу. Програма DCT-II (CC) передбачає оцінювання швидкості виконання згенерованої програми, яка складається з усередненого значення часу виконання (мкс) (див. таблицю). Для порівняння оцінок продуктивності застосовується обчислювальна платформа: *Intel (R) Core (TM) i-7 CPU 2600 @ 4,2 ГГц*, на якій отримано результати тестування (див. таблицю).

В результаті програма DCT-II(CC) виконує обчислення ДКП-II швидше, ніж FFTW_DCT для коротких обсягів. Бібліотека програм FFTW ґрунтується на найефективніших алгоритмах ДПФ і добре оптимізована для обсягів $N=2^n$.

ДКП для великих обсягів залежить від можливості розкладу обсягу N на прості множники і, відповідно, подальший розвиток застосування програми DCT-II (CC) – в її використанні на основі коротких обсягів. Тобто алгоритми перетворення ДКП коротких обсягів є будівельними блоками для багатьох складених обсягів N . Для ДКП великих обсягів бібліотека FFTW має вбудовану свободу вибору відповідних рекурсивних обчислень, для яких використовує пошук та налаштування коду відповідно до ієрархії пам'яті в обчислювальній платформі. FFTW використовує оптимізацію програмного коду за чотири етапи [16] на різних рівнях для генерування ефективного з погляду швидкодії виконання алгоритму.

Оцінка продуктивності DCT-II(CC) та FFTW_DCT

\N	7	8	10	16	20	21	29	30	32	38
FFTW	0.331292	0.301174	0.391526	0.301174	0.722818	0.963757	0.692701	0.993875	0.572231	1.295049
DCT-II	0.301174	0.301174	0.301174	0.301174	0.301174	0.301174	0.301174	0.301174	0.361409	0.301174
Ratio	1.100001	1.000001	1.300001	1.000001	2.400001	3.200002	2.300001	3.300002	1.583334	4.300002

	45	48	51	64	67	83	85	91	96	103	114
	1.114344	1.596223	1.385401	1.325166	1.686575	2.138337	1.837162	1.566106	1.566106	2.891272	2.710567
	0.301174	0.451761	0.331291	0.572231	0.301174	0.511996	0.391526	0.391526	0.752935	0.301174	0.632465
	3.700002	3.533335	4.18182	2.315791	5.600003	4.176473	4.69231	4.000002	2.080001	9.600005	4.285717

Висновки

Ефективні алгоритми ДКП-II з використанням циклічних згорток можуть застосовуватись для автоматичного генерування та виконання для коротких обсягів перетворень. Ефективна реалізація ДКП-II на основі циклічних згорток разом з використанням алгоритму простих множників Гуда–Томаса (Good–Thomas DFA), алгоритмів Вінограда та іншими методами може бути застосована для виконання великих обсягів перетворень [16]. Реалізація алгоритмів ДКП-II з використанням циклічних згорток актуальна не лише в комп'ютерних системах загального призначення, але й у цифрових сигнальних процесорах, векторизованих і багатопроцесорних системах, НВІС. Окреме обчислення циклічних згорток, які структурують базисну матрицю ДПФ за твірними масивами, і об'єднання результатів згорток уможливають застосування цих алгоритмів у паралельних комп'ютерних системах.

1. Prots'ko I., Rykmas R. *Becoming of Discrete Harmonic Transform Using Cyclic Convolutions*, *American Journal of Circuits, Systems and Signal Processing*. vol. 1, no. 3, pp. 114–119, 2015.
2. Rader C. M. *Discrete Fourier Transforms When the Number of Data Samples is prime*, *Proc. IEEE*, 56, pp. 1107–1108, 1968.
3. Winograd S. *On computing the discrete Fourier transform*, in *Proc. Nat. Acad. Sci. USA*, vol. 73, no. 4, pp. 1005–1006, April 1976, *Mathematics*.
4. Blahut R. E. *Fast algorithms for signal processing*. Cambridge University Press, 2010. – 469 p.
5. Nussbaumer Henri J. *Fast Fourier Transform and Convolution Algorithms*, by Springer-Verlag, Berlin, Heidelberg, 1982.
6. FFTW Homepage. URL: <http://fftw.org>
7. Spiral Homepage. URL: <http://spiral.net/index.html>
8. Nukada A., Nukada FFT library. Web site. 2011. URL: <http://matsuwww.is.titech.ac.jp/~nukada/nufft/>
9. NVIDIA. CUFFT libray. Fetched March 23, 2012. 2701 San Tomas Expressway, Santa Clara, CA 95050, 2012.
10. Math Kernel Library. Accessed May 14, 2012. URL: <http://software.intel.com/en-us/articles/intel-mkl/>
11. Gacic A., Voronenko Y., Chen K., Johnson R. W., Rizzolo N. SPIRAL: Code generation for DSP transforms // *Proceedings of the IEEE*, – 2005. – Vol. 93. – No. 2. – P. 232–275, special issue on "Program Generation, Optimization".
12. Prots'ko I. *The generalized technique of computation the discrete harmonic transforms*. *Proceedings of the IVth International Conference (MEMSTECH'2008)*. – Polyana, 21–24 may, 2008. – P. 101–102.
13. Prots'ko I. *Ukraine Patent 96540, G06F 17/16 (2006.01), H03M 7/30 (2006.01)*.
14. Prots'ko I. *Algorithm of Efficient Computation of DCT I-IV Using Cyclic Convolutions*, *International Journal of Circuits, Systems and Signal Processing*, vol. 7, issue 1, pp. 1–9, 2013.
15. Prots'ko I., Rykmas R., Teslyuk V. *The program implementation of the synthesis the efficient algorithms for computation of DCT-II via cyclic convolutions*. *Proceedings of the IXth International Scientific and Technical Conference (CSIT'2014)*. – Lviv, 18–22 november, 2014. – P. 116–118.
16. Frigo M., Johnson S. G. *The design and implementation of FFTW3* *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.