УДК 519.725 **https://doi.org/10.33082/td.2017.1-1.01**

ШВИДКІ АЛГОРИТМИ: НАУКА, МИСТЕЦТВО, РЕМЕСЛО
БЫСТРЫЕ АЛГОРИТМЫ: НАУКА, ИСКУССТВО, РЕМЕСЛО
FAST ALGORITHMS: SCIENCE, ART, CRAFT

А.П. ЦАРЁВ, докт.техн.наук
Западно-поморский технологический университет в Щецине, Польша

Наведено результати дослідження специфіки, можливостей і переваг швидких алгоритмів. Наведено опис розробленого підходу до розробки швидких алгоритмів, що використовують векторно-матричні операції.

***Ключові слова**: швидкі алгоритми, обчислювальна інформатика, структури обчислень.*

Приведены результаты исследования специфики, возможностей и преимуществ быстрых алгоритмов. Приведено описание разработанного подхода к разработке быстрых алгоритмов, использующих векторно-матричные операции.

***Ключевые слова:** быстрые алгоритмы, вычислительная информатика, структуры вычислений*

State contains the results of a study of the specifics, possibilities and advantages of fast algorithms. This paper focuses on the description of the proposed approach for the development of fast algorithms using vector-matrix operations.

***Keywords**: fast algorithms, computational informatics, computational structures.*

Введение. Быстрые алгоритмы − это область информатики, которая изучает алгоритмы реализации различного рода вычислительных задач с использованием как можно меньшего числа математических (и прочих) операций.

Развитие теории и практики конструирования быстых алгоритмов издавно находилось в непосредственной зависимости от прогресса в области проектирования и производства средств электронной вычислительной техники.

Цель работы. Так или иначе, разработка быстрого алгоритма требует от разработчика глубокого понимания особенностей решаемой задачи, а также широких теоретических знаний.

Такое положение дел может вызвать трудности у инженерно-технического персонала и специалистов, имеющих богатый практический опыт, но не обладающих достаточным уровнем теоретической подготовки, а в некоторых случаях − даже инспирировать нежелание самостоятельно разрабатывать такие алгоритмы. Тем не менее следует признать, что процесс создания быстрого алгоритма является делом необычайно интересным и творческим.

Многое здесь зависит не только от глубины знаний и уровня теоретической подготовки разработчика, но и от его интуиции и смекалки.

Не последню роль играет также накопленный опыт и наличие навыков решения подобного рода задач. Поэтому можно с полной уверенностью утверждать, что проектирование быстрых алгоритмов это и наука, и искусство, и ремесло.

Основной материал. Можно смело утверждать, что именно несовершенство вычислительных машин первого, второго и третьего поколений способствовало появлению на свет быстрых алгоритмов. Справедливости ради необходимо отметить, что система команд компьютеров первых поколений содержала весь необходимый набор команд, требующийся для реализации математических вычислений.

Однако, если такие операции как сложение и вычитание выполнялись в течение одного машинного цикла, то, к примеру, команда умножения требовала реализации довольно длинной последовательности операций сложения и сдвига в соответствии с правилами умножения двоичных чисел. Эта последовательность операций обычно «прошивалась» на ферритовых кольцах в блоке постоянной памяти ЭВМ и хранилась в виде микропрограммы.

Ясно, что реализация такой микропрограммы требовала значительно большего времени, чем выполнение операции сложения или операции обращения к памяти.

Таким образом, оказалось, что время реализации умножения стало главным фактором, ограничивающим скорость решения прикладных задач.

Этот факт стимулировал поиск и развитие способов и методов, позволяющих сократить число операций умножения при реализации тех или иных численных методов. Именно в рамках этого направления разработаны и применяются быстрые алгоритмы цифровой обработки данных [1] .

Прародителями быстрых вычислений можно с некоторой степенью условности считать немецкого математика К. Рунге и К. Гаусса, которые занимались поиском способов сокращения количества арифметичекких операций при проведении различного рода математических расчётов.

Хорошо известен, к примеру, алгоритмический трюк Гаусса, позволяющий вычислить произведение двух комплексных чисел с помощью всего лишь трёх умножений и пяти сложений действительных чисел [2].

Однако началом эпохи наиболее заметных достижений в области быстрых вычислений можно считать разработку в 1960 году Анатолием Алексеевичем Карацубой метода «разделяй и властвуй», продемонстрированного им, в частности, на примере синтеза нового эффективного алгоритма быстрого умножения больших целых чисел [3; 4].

Следующим революционным событием в научном мире стала разработка и публикация в 1965 году алгоритма быстрого преобразования Фурье (БПФ) авторства Дж. Кули и Дж. Тьюки, полученного по сути дела также с применением метода «разделяй и властвуй».

Появление этого алгоритма стало переломным пунктом развития теории и практики цифровой обработки сигналов и изображений, а также целого ряда других областей науки и техники, поскольку позволяло радикально сократить количество арифметических операций при вычислении дискретного преобразования Фурье [5].

Позднее появились многочисленные «быстрые» алгоритмы, вычисления свёрток и корреляций цифровых последовательностей, дискретных преобразований в различных ортогональных базисах и многие другие [6-8].

Среди прочих следует выделить ставшие «классикой» быстрых вычислений алгоритмы умножения матриц Штрассена, Винограда, алгоритмы умножения больших целых чисел Тоома-Кука, Фюрера и многие другие [9-11].

Главным преимуществом всех «быстрых» алгоритмов было радикальное сокращение операций умножения (снижение мультипликативной сложности) по сравнению с «наивными» алгоритмами.

Однако, в ряде случаев снижение количества опреаций умножения приводило к увеличению (иногда существенному) количества сложений (аддитивной сложности) и почти всегда – к увеличению сложности управления процессом вычислений, а также к росту операций пересылки данных, на которые тогда никто не обращал особого внимания в силу незначительного, по сравнению с умножением, времени их выполнения.

С развитием технологии производства элементной базы электронных вычислительных машин, появлением СБИС, содержащих встроенные аппаратные умножители, позволяющих выполнить команду умножения в течении одного машинного цикла, значение быстрых алгоритмов несколько приуменьшилось.

Неожиданно оказалось, что сокращение умножений в быстрых алгоритмах вызывающее рост операций сложения и операций переадресации данных в условиях, когда время выполнения этих операций является сравнимым, может возыметь и негативный эффект.

Практика показала, что, по крайней мере, в ряде случаев, «наивные» подходы, основанные на трудоёмких с точки зрения количества выполняемых арифметических операций, но более простых с точки зрения организации вычислений и реализации механизмов адресации данных в алгоритмах, могут оказаться эффективнее их «быстрых» модификаций.

Это позволило всякого рода дилетантам и скептикам утверждать о дальнейшей нецелесообразности поиска и применения алгоритмических решений, позволяющих снизить вычислительную сложность математических расчётов.

Необходимо отметить, что действительно, в случае, когда компьютер или иное вычислительное устройство уже содержит встроенный аппаратный умножитель, сокращение числа операций умножения за счёт непропорционально большого роста сложений может привести к негативным последствиям.

Тем не менее, при проектировании специализированных процессоров, особенно процессоров с распараллеливанием вычислений, в которых предполагается наличие целого ряда параллельно работающих блоков умножения, проблема минимизации количества этих блоков остаётся по-прежнему актуальной.

Это объясняется тем, что если аппаратная сложность сумматора зависит линейно от размера операндов, то аппаратная сложность блока умножения – квадратично.

Умножитель по сравнению с сумматором занимает на кристалле значительно больше места, потребляет значительно больше энергии и выделяет значительно больше тепла.

Ясно, что разработчик такого процесора будет стремиться к тому, чтобы его структура содержала как можно меньше блоков умножения.

В этом случае поиск алгоритмических решений, приводяших к снижению аппаратных и связанных с ними затрат является необычайно актуальным. С этой точки зрения разработка быстрых алгоритмов является экономически обоснованной и технически целесообразной.

Необходимо отметить, что до сих пор не существует универсальной методики проектирования быстрых алгоритмов.

Наиболее известные и интересные решения были получены, скорее всего, именно на основе учёта частных свойств и уникальных особенностей конкретных задач.

Так, например, алгоритм БПФ был разработан благодаря учёту свойств переодичности и мультипликативности дискретных экспоненциальных функций, алгоритм быстрой циклической свёртки − благодаря доказательству того, что свёртка двух последовательностей может быть вычислена как произведение коэффициентов БПФ этих последовательностей.

Предлагается простой и не требующий специальных знаний подход [12-13] к разработке быстрых алгоритмов, использующих векторно-матричные операции.

Главное внимание сосредоточено именно на этом типе операций, поскольку необходимость быстрого вычисления векторно-матричных произведений с различными матричными ядрами возникает при решении огромного количества прикладных задач, связанных с цифровой обработ-

кой данных в радио- и гидролокации, навигации, телекоммуникации, распознавании образов, анализе сцен, машинной графике и т.д..

Не претендуя на полную универсальность, предлагаемый подход всё-таки обладает достаточным набором свойств, позволяющих унифицировать, формализовать и даже автоматизировать в интерактивном режиме разработку быстрых алгоритмов [14].

Выводы. С помощью развиваемого подхода был разработан целый ряд эффективных алгоритмических решений, позволяющих уменьшить время выполнения вычислений при решении различных прикладных задач и/или упростить структуры вычислителей [15-28].

ЛИТЕРАТУРА

1. *Гашков С.Б. Занимательная компьютерная арифметика. Быстрые алгоритмы операций с числами и многочленами / Гашков С.Б. – М.: Книжный дом «ЛИБРОКОМ», 2012. – 224 с.*
2. *Блейхут Р. Быстрые алгоритмы цифровой обработки сигналов: Пер. с англ. – М.: Мир, 1989. – 448 с.*
3. *Карацуба А., Офман Ю. Умножение многозначных чисел на автоматах // Доклады Академии Наук СССР, 1962. – Т. 145. − № 2.*
4. *Гриценко С.А. Научные достижения Анатолия Алексеевича Карацубы / С.А. Гриценко, Е.А. Карацуба, М.А. Королёв, И.С. Резвякова, Д.И. Толев, М.Е. Чанга // Совр. пробл. математики: Математика и информатика, 1. − 2012. – Т. 16. – С. 7-30.*
5. *Кули Льюис Уэлч. Исторические замечания относительно быстрого преобразования Фурье // ТИИЭР. − 1967. – Т. 55. − № 10. – С. 18-21.*
6. *Нуссбаумер Г. Быстрое преобразование Фурье и алгоритмы вычисления сверток: Пер. с англ. − М.: Радио и связь, 1985. – 248 с.*
7. *Хуанг Т.С., Эклунд Дж. О., Нуссбаумер Г. Быстрые алгоритмы в цифровой обработке изображений. − М.: Радио и связь, 1984. –220 с.*
8. *Макклеллан Дж.Г., Рейдер Ч.М. Применение теории чисел в цифровой обработке сигналов. − М.: Радио и связь, 1983. – 264 с.*
9. *Рабин М.О., Виноград Ш. Быстрое вычисление многочленов с предварительной рациональной обработкой коэффициентов // Математика. − 1974. – Т. 18. − Вып. 4. – С. 98-120.*
10. *Strassen V., Gaussian Elimination is not Optimal // Numer. Math — Springer Science+Business Media. − 1969. – Vol. 13. − № 4. – P. 354-356.*

11. *Окулов С.М. Алгоритмы компьютерной арифметики // С.М. Окулов, А.В. Лялин, О.А. Пестов, Е.В. Разова. − 2-е изд. (эл.). − М.: Лаборатория знаний, 2015. – 288 с.*
12. *Cariow A. Strategies for the synthesis of fast algorithms for the computation of the matrix-vector products // Journal of Signal Processing Theory and Applications. − 2014. – Vol. 3. – № 1. – P. 1-19.*
13. *Cariow A. Algorytmiczne aspekty racjonalizacji obliczeń w cyfrowym przetwarzaniu sygnałów. Wydawnictwo Uczelniane ZUT / PPH ZAPOL Dmochowski Sobczyk Spółka Jawna, 2011. – 230 c.*
14. *Andreatto B., Cariow A., Automatic generation of fast algorithms for matrix-vector multiplication // International Journal of Computer Mathematics. − 2017. – P. 1-19.*
15. *Gliszczyński M., Cariow A. Szybki algorytm splotu kołowego dla $N = 2^m$ // Pomiary Automatyka Kontrola. − 2009. – 55. – № 8. – P. 566-568.*
16. *Cariow A., Cariowa G. Aspekty algorytmiczne redukcji liczby bloków mnożących w układzie do obliczania iloczynu dwóch kwaternionów // Pomiary, Automatyka, Kontrola. 2010. – P. 688-690.*
17. *Cariow A., Cariowa G. Aspekty algorytmiczne organizacji jednostki procesorowej do mnożenia liczb Cayleya // Elektronika: konstrukcje, technologie, zastosowania. − 2010. – 51. – № 11. – S. 104-108.*
18. *Царёв А.П., Царёва Г.Д. Алгоритм умножения октонионов: Известия Вузов // Радиоэлектроника. − 2012. – Т. 55. – № 10. – С. 44-54.*
19. *Cariow A., Cariowa G. An algorithm for complex-valued vector-matrix multiplication // Electrical Review. − 2012. – P. 88. – № 10 b. – P. 213-216.*
20. *Majorkowska-Mech D., Cariow A. An algorithm for discrete fractional Hadamad transform with reduced arithmetical complexity // Electrical Review. − 2012. – R 88. – № 11 a. – P. 70-76.*
21. *Cariow A., Gliszczyński M. Fast algorithms to compute matrix-vector products for Toeplitz and Hankel matrices // Electrical Review. − 2012. – R 88. – № 8. – P. 166-171.*
22. *Cariow A., Cariowa G. An algorithm for fast multiplication of sedenions // Information Processing Letters. − 2013. – P. 324-331.*
23. *Cariow A., Cariowa G. An algorithm for multiplication of Dirac numbers // Journal of Theoretical and Applied Computer Science. − 2013. – № 4. – P. 26-34.*
24. *Cariow A., Cariowa G. Algorithmic tricks for reducing the complexity of FDWT/IDWT basic operations implementation // Inter-*

national Journal of Image, Graphics and Signal Processing. − 2014. – № 10. – P.1-9.

25. *Cariow A., Cariowa G. An algorithm for fast multiplication of Pauli numbers. Advances in Applied Clifford Algebras, 2015. – P. 1-11.*
26. *Cariow A., Majorkowska-Mech D. Fast algorithm for discrete fractional Hadamard transform. Numerical Algorithms, 2015. – Vol. 68. – № 3. – P. 585-600.*
27. *Cariow A., Cariowa G. On the Multiplication of Biquaternions, Soft Computing in Computer and Information Science: Advances in Intelligent Systems and Computing, 2015. – Vol. 342. – P. 423-434.*
28. *Cariow A., Cariowa G., Witczak M. A FPGA-Oriented Fully Parallel Algorithm for multiplying dual quaternions, Measurement Automation Monitoring, Jul. 2015. – Vol. 61. – № 07. – P. 370-372.*

Стаття надійшла до редакції 14.09.2017