

КОМПЬЮТЕРНЫЕ СИСТЕМЫ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ COMPUTER SCIENCE

doi: 10.17586/2226-1494-2023-23-1-79-87

УДК 004.932

Вычислительные методы повышения быстродействия дискретного вейвлет-преобразования на базе FPGA

Сергей Владимирович Сай¹✉, Алексей Владимирович Зинкевич²

^{1,2} Тихоокеанский государственный университет, Хабаровск, 680035, Российская Федерация

¹ sai1111@rambler.ru✉, <https://orcid.org/0000-0002-4506-1857>

² zinkevich1985@mail.ru, <https://orcid.org/0000-0003-3231-8569>

Аннотация

Предмет исследования. Рассмотрены вычислительные методы и особенности построения сложно-функционального блока реализации дискретного вейвлет-преобразования Добеши 9/7 в системах цифровой обработки сигналов изображений на базе Field-Programmable Gate Array (FPGA). **Метод.** Предложена математическая модель и алгоритмы реализации методов параллельной и последовательно-конвейерной обработок сигналов для вычисления коэффициентов дискретного биортогонального вейвлета Добеши 9/7 с учетом архитектуры используемой FPGA. В основе модели использованы методы факторизации вейвлет-преобразований с применением лифтинговых схем. В отличие от обычных лифтинговых схем, предлагаемый способ и алгоритмы позволяют повысить быстродействие вычислений дискретного вейвлет-преобразования при упрощении аппаратной реализации. В качестве среды разработки использованы средства автоматизированного проектирования Quartus II и ModelSim. Поведенческая модель написана на языке Verilog HDL. Для апробации использовалась матрица FPGA Altera Cyclone® IV 4CE115. На основе полученной поведенческой модели разработана тестирующая модель и осуществлена симуляция цифровой схемы в среде ModelSim. **Основные результаты.** Предложена формула оценки количества тактов выполнения прямого и обратного дискретного вейвлет-преобразований от числа входных элементов блока и характеристик FPGA. В результате экспериментов получены зависимости количества тактов расчета дискретного вейвлет-преобразования в зависимости от размера стороны квадратного изображения при разных вариантах количества блоков параллельной обработки. Показано, что параллельная работа нескольких независимых модулей позволяет проводить одновременно обработку нескольких входных столбцов (строк) из исходного двумерного массива. Унификация модуля умножителя-сумматора повысила эффективность вычисления и сократила объем занимаемых аппаратных ресурсов. Конвейерная структура дискретного вейвлет-преобразования отличается меньшими аппаратными затратами при реализации блока вычисления и распределения памяти. В результате тестирования цифровой схемы получено, что разработанная структура блока повысила быстродействие дискретного вейвлет-преобразования, а также сократила стоимость системы на кристалле. **Практическая значимость.** Предложенная реализация блока двумерного прямого и обратного вейвлет-преобразований для банка фильтров Добеши 9/7 является законченным модулем и может быть использована как готовый сложно-функциональный блок для последующей разработки систем передачи изображений высокого качества в реальном времени.

Ключевые слова

вейвлет-преобразование, лифтинг-схема, FPGA, сложно-функциональный блок, цифровая обработка сигналов

Благодарности

Исследование выполнено при финансовой поддержке Российского научного фонда в рамках научного проекта № 22-21-00394 «Развитие нейросетевых методов повышения качества передачи цифровых изображений в интеллектуальных видеосистемах».

Ссылка для цитирования: Сай С.В., Зинкевич А.В. Вычислительные методы повышения быстродействия дискретного вейвлет-преобразования на базе FPGA // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23, № 1. С. 79–87. doi: 10.17586/2226-1494-2023-23-1-79-87

Computational methods to increase the speed of FPGA-based discrete wavelet transforms

Sergey V. Sai¹✉, Alexey V. Zinkevich²

^{1,2} Pacific National University, Khabarovsk, 680035, Russian Federation

¹ sai1111@rambler.ru✉, <https://orcid.org/0000-0002-4506-1857>

² zinkevich1985@mail.ru, <https://orcid.org/0000-0003-3231-8569>

Abstract

The article considers the computational methods and features of the construction of a complex functional block for the implementation of the discrete wavelet transform (DWT) Dobeshie 9/7 in digital image signal processing systems based on FPGA. We proposed a mathematical model and algorithms for the implementation of parallel and series-convector methods of signal processing to calculate the coefficients of a discrete bi-orthogonal Dobeshie wavelet 9/7 taking into account the architecture of used FPGA. The model is based on wavelet transform factorization methods using lifting schemes. In contrast to conventional lifting schemes, the proposed method and algorithms can increase the speed of FPGA calculations with simplified hardware implementation. CAD Quartus II and ModelSim are used as a development environment. The behavioral model is written in Verilog HDL. Altera Cyclone® IV 4CE115 was used as FPGA. On the basis of the obtained behavioral model the testing module was developed and the simulation of digital circuit in the ModelSim environment was carried out. The formula for estimating the number of clock cycles of the forward and reverse DWT has been proposed; on its basis the estimate of the number of parallel computations depending on the number of input elements and the characteristics of the FPGA was obtained. As a result of experiments the dependences of the number of cycles for DWT computation depending on the size of the side of a square image with different variants of the number of parallel processing blocks were obtained. It is shown that parallel work of several independent modules gives a possibility to conduct concurrent processing of several input columns (rows) from input 2D array, and unification of the multiplier-summing module allows to increase efficiency of calculations and to reduce volume of occupied hardware resources. Conveyor based DWT structure is characterized by less hardware costs in terms of implementation of the calculator unit and memory allocation. As a result of testing the digital circuit, it was found that the developed block structure can significantly increase the DWT speed as well as reduce the cost of the system on a chip. The proposed realization of the block of two-dimensional forward and reverse wavelet transform for the Dobeshie 9/7 filter bank forms a complete module and can be used as a ready-made complex functional block for further development of high quality image transmission systems in real time.

Keywords

wavelet transform, lifter circuit, FPGA, complex function block, digital signal processing

Acknowledgements

The research supported financially by the Russian Science Foundation within the scientific project no. 22-21-00394 “Development of neural network methods to improve the quality of digital image transmission in intelligent video systems”.

For citation: Sai S.V., Zinkevich A.V. Computational methods to increase the speed of FPGA-based discrete wavelet transforms. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 1, pp. 79–87 (in Russian). doi: 10.17586/2226-1494-2023-23-1-79-87

Введение

Дискретное вейвлет-преобразование (ДВП) является основой стандарта сжатия JPEG2000 [1] который имеет преимущество по сравнению со стандартом JPEG по визуальному качеству изображения при одинаковом коэффициенте компрессии. В системах видеокompresии на основе стандартов H.264 (H.265) используются алгоритмы для компрессии опорных (I) кадров видеоряда.

В работе [2] рассмотрены особенности алгоритма сжатия JPEG2000. Одна из особенностей — выполнение в кодере следующих основных процедур: преобразование цифровых RGB-сигналов исходного кадра изображения в сигналы YVU; выполнение ДВП для каждой из YVU-компонент изображения; квантование коэффициентов трансформации и обнуление малозначущих коэффициентов; энтропийное сжатие. Данные процедуры также используются в алгоритме JPEG, где вместо ДВП применено блочное дискретное косинус-преобразование.

Для двумерного сигнала изображение ДВП реализуется следующим образом. На первой итерации

одномерное ДВП выполняется последовательно по строкам и столбцам, затем формируются четыре матрицы (НН — низкочастотная фильтрация по строкам и столбцам; НВ — низкочастотная фильтрация по строкам и высокочастотная по столбцам; ВН — высокочастотная фильтрация по строкам и низкочастотная по столбцам; ВВ — высокочастотная фильтрация по строкам и столбцам) коэффициентов ДВП с размерами в два раза меньшими ширины и высоты изображения. На второй итерации происходит аналогичное преобразование для НН матрицы и т. д. Обычно достаточно 4–6 итераций. Такой процесс преобразований эквивалентен операции фильтрации сигналов изображения по частотным субдиапазонам для каждой итерации. Таким образом, в процессе итераций образуются масштабированные (уменьшенные) в два раза копии исходного изображения и его дополнение в виде высокочастотных составляющих.

Основные проблемы использования JPEG2000 связаны с большим количеством вычислений, в том числе работы блока для получения коэффициентов двумерного ДВП. Особенно эти проблемы актуальны при об-

работке изображений с высоким разрешением, например, в форматах UHD, 2K или 4K. Реализация ДВП на специализированных интегральных схемах (Application Specific Integrated Circuit, ASIC) или программируемых вентильных матрицах (Field-Programmable Gate Array, FPGA) с применением параллельных вычислений позволяет повысить эффективность и скорость кодирования изображений в реальном масштабе времени.

В работе [3] предложена реализация ДВП путем объединения вейвлет-преобразования и банка фильтров. Добеши применила ДВП к различным изображениям и предложила множество популярных вейвлет-фильтров, включая фильтры 9/7 и 5/3 [4]. В [5] представлена схема лифтинга, и показано, что любое ДВП можно разложить на последовательность простых шагов фильтрации и тем самым ускорить вычислительную способность преобразования.

Для аппаратной реализации вейвлет-преобразований известно несколько архитектур, которые можно разделить на последовательные и параллельные. Часть последовательных архитектур основана на использовании систолических матриц, либо на конвейере, более эффективно реализующем вычислительный процесс [6–17]. Работы [4, 5], использующие факторизацию для выполнения вейвлет-преобразований, и предложенные в них лифтинговые схемы, позволяют значительно упростить аппаратную реализацию ДВП. Чаще всего в подобных работах основное внимание уделяется минимизации арифметической части устройства или используемой памяти. Однако для устройств, имеющих внутреннюю память, размерность арифметического блока не так важна. При этом важно минимизировать объем используемой памяти и повысить быстродействие вычислений, учитывая особенности реализации в конкретной FPGA.

В настоящей работе рассмотрены методы и алгоритмы реализации ДВП на основе биортогонального вейвлета Добеши 9/7 с использованием лифтинговой схемы. Цель работы — сравнительная оценка быстродействия и занимаемых аппаратных ресурсов двух архитектур (параллельной и конвейерной), а также сравнение по количеству вычислительных операций, полученных в [18] при программной реализации ДВП.

Вейвлет-преобразование на основе лифтинговой схемы

На сегодняшний день существуют два подхода к вычислению ДВП, основанных на свертке и лифтинговой схеме. Традиционный подход на основе свертки представляет собой свертку вейвлет-функции с сигналом. Такая функция представляется фильтром с конечной импульсной характеристикой, состоящим из фильтров нижних и верхних частот. Затем применяется операция децимации, цель которой сократить избыточность хранимой информации. В результате получается объем данных коэффициентов ДВП, равный по размерности исходному сигналу. При этом первая половина данных представлена низкочастотными коэффициентами, а вторая половина — высокочастотными.

Второй подход, основанный на лифтинговой схеме, снижает вычислительную сложность в два раза и позволяет экономить память. Схема прямого лифтинг-преобразования реализуется с помощью четырех базовых этапов: разбиения (S), предсказания (P), обновления (U) и масштабирования (K). Функции $P(z)$ и $U(z)$ представляются через z -преобразование. Выходные коэффициенты LP и HP соответствуют низкочастотной и высокочастотной составляющей (рис. 1).

Этап разбиения делит входной сигнал на четные (s) и нечетные (d) элементы:

$$s_i^{(0)} = x_{2i}; d_i^{(0)} = x_{2i+1},$$

где x_{2i} и x_{2i+1} — четные и нечетные элементы входного сигнала.

Значения элементов на каждом этапе преобразования вычисляются по формулам:

$$d_i^{(n)} = d_i^{(n-1)} + \sum_k P_n(k) s_k^{(n-1)}, n \in [1, 2, \dots, N];$$

$$s_i^{(n)} = s_i^{(n-1)} + \sum_k U_n(k) d_k^{(n)}, n \in [1, 2, \dots, N],$$

где i и k — индексы вычисляемых элементов; P_n и U_n — функции предсказания и обновления на n -ом уровне.

Для банка фильтров Добеши 9/7 количество уровней $N = 2$, и используется четыре этапа для вычисления вейвлет-преобразования. Выход последнего этапа предсказания умножается на коэффициент K_1 , а последнего этапа обновления — на K_0 .

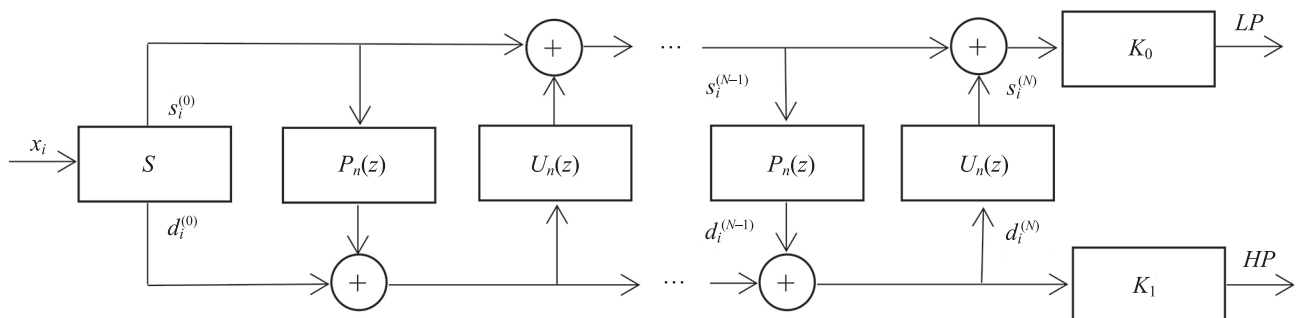


Рис. 1. Лифтинговая схема прямого вейвлет-преобразования

Fig. 1. Lifting scheme of direct wavelet transform

Алгоритм работы лифтинговой схемы прямого вейвлет-преобразования можно представить в виде выражения:

$$\begin{bmatrix} LP(z) \\ HP(z) \end{bmatrix} = \begin{bmatrix} s_m(z) \\ d_m(z) \end{bmatrix} = \tilde{\mathbf{P}}(z) \begin{bmatrix} s(z) \\ d(z) \end{bmatrix}, \quad (1)$$

где

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \left\{ \prod_{k=m} \begin{bmatrix} 1 & Q_{2k}(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ Q_{2k-1}(z) & 1 \end{bmatrix} \right\},$$

$LP(z)$ и $HP(z)$ — низкочастотные и высокочастотные коэффициенты сигнала $X(z)$; $Q_{2k}(z)$ и $Q_{2k-1}(z)$ — многочлены Лорана, полученные из факторизации многофазной матрицы, представляющей фильтр по алгоритму Евклида [5, 6]; K и $1/K$ — ненулевые коэффициенты масштабирования.

Применяя факторизацию лифтинговой схемы (1) для банка фильтров Добеши 9/7 получим:

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} K & 0 \\ 0 & \frac{1}{K} \end{bmatrix} \begin{bmatrix} 1 & d(1+z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ c(1+z^{-1}) & 1 \end{bmatrix} \times \\ \times \begin{bmatrix} 1 & b(1+z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a(1+z^{-1}) & 1 \end{bmatrix}, \quad (2)$$

где $a = -1,586134342$; $b = -0,0529801185$; $c = 0,882911076$; $d = 0,443506852$; $K = 1,149604398$.

В выражении (2) и схеме на рис. 1 этап масштабирования представляет собой операцию умножения на ненулевые коэффициенты K и $1/K$. Так как умножители обычно — дорогостоящие логические ресурсы, то желательно минимизировать их количество в схеме. Схему вычисления можно изменить за счет внесения коэффициентов K и $1/K$ следующим образом:

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} K & dK^2(1+z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ \frac{c}{K}(1+z^{-1}) & \frac{1}{K} \end{bmatrix} \times \\ \times \begin{bmatrix} 1 & b(1+z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ a(1+z^{-1}) & 1 \end{bmatrix}, \quad (3)$$

где $dK^2 = 0,586134341$; $c/K = 0,7680129595$.

В результате такого преобразования этап масштабирования (K) отдельно не выполняется. Это позволяет унифицировать процесс вычисления для этапов предсказания и обновления, который будет включать три операции умножения и две — сложения.

Схема обратного лифтинг-преобразования реализуется также с помощью четырех базовых этапов и напоминает зеркальное отражение схемы прямого вейвлет-анализа, где этап разбиения (S) заменяется этапом — объединения (M). Получим выражение для вычисления коэффициентов обратного вейвлет-преобразования:

$$\tilde{\mathbf{P}}(z) = \begin{bmatrix} 1 & 0 \\ -a(1+z^{-1}) & 1 \end{bmatrix} \begin{bmatrix} 1 & -b(1+z) \\ 0 & 1 \end{bmatrix} \times \\ \times \begin{bmatrix} 1 & 0 \\ -c(1+z^{-1}) & K \end{bmatrix} \begin{bmatrix} \frac{1}{K} & -dK(1+z) \\ 0 & 1 \end{bmatrix}, \quad (4)$$

где $dK = 0,5098574276$.

Аппаратная реализация ДВП на базе FPGA

Отметим, что микросхемы FPGA известны своими возможностями по цифровой обработке сигналов (Digital Signal Processor, DSP) в формате фиксированной точки. Однако вычисляемые коэффициенты ДВП имеют дробный формат, что требует представления и обработки чисел в формате с плавающей точкой (IEEE 754¹). Использование такого формата усложняет схему и приводит к затратам большего количества ресурсов FPGA. Чтобы этого избежать, необходимо представить коэффициенты целыми значениями, умножив их на 2^k , где k — число двоичных разрядов для сдвига данных влево.

Входными данными изображения являются значения сигналов яркости (Y) или цветности (U , V) в диапазоне от 0 до 255, что соответствует стандартному значению в 8 бит на пиксел. В результате преобразования диапазон будет расширен, а также станет знаковым. Чтобы хранить целую часть промежуточных и выходных коэффициентов потребуется диапазон от минус 1024 до 1024, что будет занимать в дополнительном коде 11 бит. Для хранения дробной части потребуется увеличить число двоичных разрядов, задав общее количество бит, исходя из разрядности типовых умножителей FPGA равной 18, либо кратной 18.

Алгоритм прямого вейвлет-преобразования по выражению (3) для сигнала яркости приведен на рис. 2. «Входная строка» и «Выходная строка» — данные, хранящиеся во внутренней или внешней памяти FPGA. Значения считываются из памяти, проходят все этапы вычислений и записываются обратно. В отдельных блоках памяти (на рис. 2 отмечены одним цветом), хранятся промежуточные результаты четных и нечетных элементов. Обозначения «Е» и «О» являются четными и нечетными позициями, а их индексы соответствуют порядку вычисления. Коэффициенты ($a_1 - a_4$, k_1 , k_2) задаются согласно коэффициентам, указанным в выражении (3), где $a_1 = a$, $a_2 = b$, $a_3 = c/K$, $a_4 = dK^2$, $k_1 = 1/K$, $k_2 = K$. Также имеются единичные коэффициенты на первом этапе предсказания и обновления.

Процесс симметричного расширения — часть основного алгоритма преобразования и обеспечивает вычисление на границах входных данных. Для его эффективной реализации используется внутренняя двухпортовая память, в результате необходимые для расчета операции — чтение и запись, происходят одновременно и независимо по двум шинам. На этапах «Предсказание» и «Обновление» выполняются операции с одинаковым числом умножителей и сумматоров. Поскольку этапы исполняются последовательно, и на каждом происходит три умножения и два сложения, то для расчета используется один модуль. В результате выполнения этапов получим коэффициенты одномерного уровня прямого ДВП. Рассчитанные коэффициенты записываются обратно в исходную строку и заменяют в памяти предыдущие данные. В процессе формирования выходной строки происходит группировка низкоча-

¹ [Электронный ресурс]. Режим доступа: <https://ieeexplore.ieee.org/document/4610935> (дата обращения: 19.12.2022).

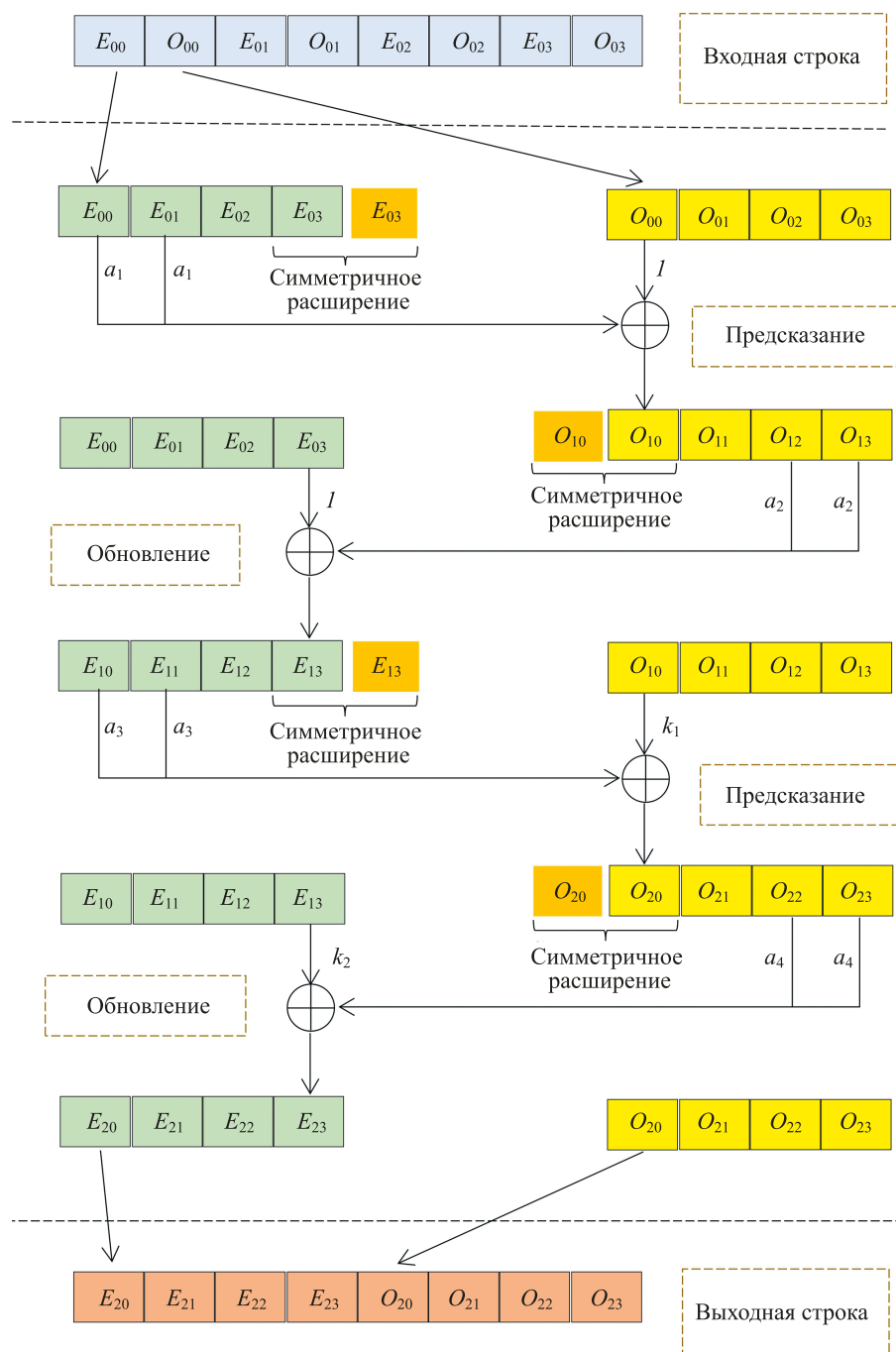


Рис. 2. Алгоритм прямого вейвлет-преобразования

Fig. 2. Direct wavelet transform algorithm

стотных ($E_{20}-E_{23}$) и высокочастотных ($O_{20}-O_{23}$) коэффициентов. После этого одномерное ДВП повторяется для столбцов и получаются итоговые коэффициенты двумерного преобразования.

Схема обратного ДВП может быть получена из выражения (4) и имеет зеркальную структуру схемы прямого вейвлет-преобразования (рис. 2) со следующими значениями коэффициентов: $a_1 = 1,58613$; $a_2 = 0,05298$; $a_3 = -0,88291$; $a_4 = -0,50985$; $k_1 = 1,14960$; $k_2 = 0,86986$.

Для известного количества входных элементов по предложенному алгоритму можно вычислить количе-

ство тактов для одномерного прямого или обратного вейвлет-преобразований:

$$N = 4(w/2 + 1 + p), \quad (5)$$

где w — количество входных элементов; l и p — количество тактов задержки в модуле умножения-сложения (1 такт) и чтения из памяти (1 такт).

Для одного уровня двумерного ДВП количество тактов будет удвоено.

Параллельное и конвейерное вычисления ДВП

Параллельное выполнение ДВП позволяет разделить обработку независимых данных на несколько блоков вычисления. Поскольку процесс двумерного преобразования состоит из последовательного расчета сначала по столбцам, а затем по строкам, то выполнить эти два процесса параллельно невозможно. Таким образом, представляется возможным выполнять параллельно вычисления по строкам или столбцам. В этом случае происходит считывание столбца (строки), его обработка и запись обратно. Во время обработки текущего столбца возможно считывание и обработка следующего и т. д. Если взять количество тактов для расчета одномерного ДВП из выражения (5), то можно рассчитать количество параллельных вычислений по формуле:

$$L = 2(N + w/2 + 1)/w. \quad (6)$$

Так, для столбца (строки) длиной $w = 8$, по выражению (5) получим $N = 4(8/2 + 1 + 1) = 24$ и по выражению (6) — $L = 2(24 + 8/2 + 1)/8 = 7,25$. Отсюда следует, что при использовании семи параллельно работающих вычислительных блоков расчет двумерного ДВП будет выполнен максимально быстро. Однако, если увеличить длину строки, например, $w = 2048$, то $N = 4(2048/2 + 1 + 1) = 4104$ и $L = 2(4104 + 2048/2 + 1)/2048 \approx 5$. Такое изменение связано со временем простоя блока вычисления в моментах, когда осуществляется доступ к памяти.

На рис. 3 приведена зависимость количества тактов для вычисления ДВП в зависимости от размера стороны квадратного изображения при некоторых вариантах количества блоков параллельной обработки.

Выполним оценку корректности рассмотренного алгоритма. Для этого запишем код блока вычисления на языке Verilog HDL, разработаем Testbench (тестирующий модуль) и осуществим симуляцию цифровой схемы в среде ModelSim. Для параллельного расчета ДВП реализован следующий алгоритм. Процесс начинается с чтения из двумерного массива двухпортовой памяти первого столбца. Считывание четного и нечетного элементов столбца происходит одновременно с их записью в память первого блока вычисления. Когда

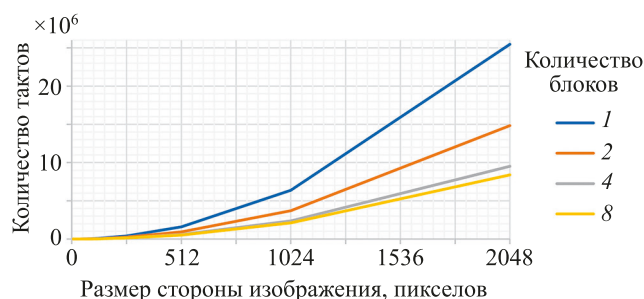


Рис. 3. Зависимость количества тактов для вычисления двумерного дискретного вейвлет-преобразования при различном количестве блоков параллельной обработки

Fig. 3. Dependence of the number of cycles for calculating a two-dimensional DWT with a different number of parallel processing units

данные первого столбца скопированы в память первого блока вычисления, он начинает процесс расчета одномерного ДВП, а элементы из второго столбца двумерного массива аналогично копируются во второй блок вычисления. Когда восьмой блок вычисления получил входные данные, первый уже закончил процесс расчета, и начинается последовательное считывание обратно в двумерный массив. Если все столбцы скопированы, вычисление аналогично повторяется для всех строк двумерного массива. Вычисление последующего уровня происходит тогда, когда получены низкочастотные коэффициенты текущего уровня. При такой организации расчета ДВП занимаемые ресурсы FPGA в основном определяются количеством умножителей и размером памяти для хранения двумерного массива.

Основное ограничение в рассмотренном вычислении — ожидание завершения предыдущего этапа. Так, в формуле (5), умножение на коэффициент 4 показывает, что необходимо выполнить этапы предсказания и обновления и далее повторить их еще раз. При этом промежуточные данные сохраняются в память, что при последующем доступе снова вносят задержку. Заметим (как видно из рис. 2), что для начала вычисления этапа обновления не обязательно ожидать окончания вычисления этапа предсказания, достаточно только первых двух коэффициентов. Аналогично происходит и на последующих этапах — для второго этапа обновления требуется только пара коэффициентов второго этапа предсказания. В этом случае необходимость в памяти для хранения промежуточных строк (столбцов) отсутствует. Исходные данные поступают последовательно на вход блока вычислителя по одному элементу за такт, и через несколько тактов задержки, вносимой конвейером, на выходе последовательно появляются коэффициенты ДВП. Для хранения промежуточных коэффициентов различных этапов используется внутренняя регистровая память с параллельным чтением данных. На FPGA такое устройство можно реализовать с помощью триггеров, входящих в состав типовых конфигурируемых блоков.

Для реализации конвейерной схемы требуется два модуля умножителя-сумматора, аналогичные использовались при параллельном выполнении ДВП (имеется в виду модуль, содержащий три умножителя и два сумматора, используемый для параллельного выполнения ДВП), а также 11 регистров для хранения текущих коэффициентов. С учетом задержки в блоке умножителя-сумматора в один такт задержка конвейера составляет 7 тактов. Это означает, что вычисленные коэффициенты ДВП появляются на выходе схемы спустя 7 тактов. В результате для входной строки длины w вычисление займет $N_k = (w + 7)$ тактов.

В случае вычисления двумерного ДВП чтение и запись двухпортовой памяти происходит одновременно по разным адресам с разницей в 7 тактов. Зависимость количества тактов для вычисления двумерного ДВП практически не отличается от варианта с количеством блоков параллельной обработки равной 8. Однако в этом случае количество затрачиваемых ресурсов FPGA меньше. Так, например, блоков умножителей-сумматоров в четыре раза меньше, а память для хранения

Таблица. Результаты сравнения быстродействия и занимаемых ресурсов архитектур
Table. Results of comparison of performance and resources occupied architectures

Характеристика	Архитектура			
	Программная реализация [18]	Параллельная ($L = 4$)	Параллельная ($L = 8$)	Конвейерная
Количество умножителей	—	24	48	12
Количество тактов вычисления	4 456 448	148 736	131 200	131 079
Занимаемая внутренняя память, бит	—	1 198 080	1 198 080	1 179 648

промежуточных четных и нечетных коэффициентов, суммарно 4 строки (столбца), не используется.

Апробация алгоритмов на FPGA

Для апробации разработанных алгоритмов была выбрана FPGA Altera Cyclone® IV 4CE115. Выбранная FPGA имеет следующие характеристики: 114 480 логических элементов; 3888 кбит встроенной блочной памяти; 532 встроенных аппаратных умножителя разрядностью 9 бит; максимальная рабочая частота 250 МГц.

Использование внутренней памяти, в силу ее ограниченного объема, позволяет реализовать ДВП только для квадратных изображений со стороной не более 256 пикселей. Также возможны варианты эффективного параллельного вычисления, когда строки и столбцы исходного изображения будут кратны количеству блоков вычисления, определенных по формуле (6).

В ходе синтеза схемы, для достижения требуемой разрядности коэффициентов, умножители объединены попарно. Таким образом, становится возможным обрабатывать значения 18 битных чисел. При этом первые 11 бит будут хранить целую часть коэффициентов, а остальные 7 бит — дробную часть. Для перехода от чисел с плавающей точкой к числам с фиксированной точкой входные данные необходимо умножить на 2^7 путем сдвига влево на 7 разрядов, тогда все промежуточные и выходные коэффициенты будут располагаться в диапазоне от 1 до 131 072. При переводе выходных коэффициентов обратно необходимо выполнить их сдвиг вправо на 7 разрядов. Один модуль умножения-сложения в такой конфигурации использует шесть 9-битных умножителей для операций с фиксированной точкой.

Средства анализа и синтеза в среде Quartus II позволяют оценить максимальную рабочую частоту блока вычисления. Поскольку в общей схеме самой «медленной» является операция умножения, то она ограничивает максимальную тактовую частоту величиной 164 МГц. При этом задержка между вводом данных в модуль умножения-сложения и получением результата составит три такта.

В таблице приведены результаты сравнения быстродействия и занимаемых ресурсов программной [18],

параллельной и конвейерной архитектур для квадратного изображения со стороной 256 пикселей.

Заключение

Предложенные реализации параллельной и конвейерной архитектур двумерного прямого и обратного вейвлет-преобразований для банка фильтров Добеши 9/7 являются законченным решением и могут быть использованы как готовый сложно-функциональный блок для последующей разработки системы кодирования и передачи изображений.

Параллельная работа нескольких независимых модулей дает возможность проводить одновременно обработку нескольких входных столбцов (строк) из исходного двумерного массива, а унификация модуля умножителя-сумматора позволяет повысить эффективность вычисления и сократить объем занимаемых аппаратных ресурсов. Конвейерная архитектура дискретного вейвлет-преобразования, ориентированная на синтез в FPGA или в функциональность сложно-функционального блока, отличается меньшими аппаратными затратами при реализации блока вычисления и распределения памяти. В результате можно повысить быстродействие, а также уменьшить стоимость системы на кристалле.

Использование FPGA с небольшим объемом внутренней памяти ограничивает реализацию блока дискретного вейвлет-преобразования для изображений большого размера. При этом обработка может выполняться блоками, согласно стандарту JPEG2000, либо, при использовании внешней памяти и некоторого изменения в модуле, целиком для всего изображения.

Предложенные алгоритмы, реализованные в виде поведенческой модели на языке Verilog HDL, при незначительной корректировке могут быть использованы на базе отечественных аналогов, например, в программируемых логических интегральных схемах производства АО «ВЗПП-С» (г. Воронеж).

К перспективным направлениям исследования относится разработка и внедрение в видеосистему на кристалле дополнительных к блоку дискретного вейвлет-преобразования модулей: нейросетевого классификатора блоков изображения, адаптивного квантования и сжатия.

Литература

References

1. Taubman D., Marcellin M.W. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002. 774 p.
2. Воробьев В.И., Грибунин В.Г. Теория и практика вейвлет-преобразования. СПб.: Изд-во ВУС, 1999. 208 с.
3. Mallat S. A theory for multiresolution signal decomposition: The wavelet representation // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 1989. V. 11. N 7. P. 674–693. <https://doi.org/10.1109/34.192463>
4. Cohen A., Daubechies I., Feauveau J.-C. Biorthogonal bases of compactly supported wavelets // *Communications on Pure and Applied Mathematics*. 1992. V. 45. N 5. P. 485–560. <https://doi.org/10.1002/cpa.3160450502>
5. Daubechies I., Sweldens W. Factoring wavelet transforms into lifting steps // *Journal of Fourier Analysis and Applications*. 1998. V. 4. N 3. P. 247–269. <https://doi.org/10.1007/BF02476026>
6. Sameen I., Chang Y.C., Ng M.S., Goi B.-M., Ooi C.-P. A Unified FPGA-based system architecture for 2-D discrete wavelet transform // *Journal of Signal Processing Systems*. 2013. V. 71. N 2. P. 123–142. <https://doi.org/10.1007/s11265-012-0687-1>
7. Rekha K., Ravi K. Design of high speed lifting based DWT using 9/7 wavelet transform for image compression // *Proc. of the International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*. 2017. P. 132–137. <https://doi.org/10.1109/ICRAECT.2017.38>
8. Zhong X., Jiang H., Cao H., Yang R. Efficient lifting based CDF9/7 wavelet transform using fixed point // *Proc. of the 3rd International Congress on Image and Signal Processing*. 2010. P. 3094–3097. <https://doi.org/10.1109/CISP.2010.5648075>
9. Dimitroulakos G., Zervas N., Sklavos N., Goutis C. An efficient VLSI implementation for forward and inverse wavelet transform for JPEG2000 // *Proc. of the 14th International Conference on Digital Signal Processing Proceedings (DSP)*. V. 1. 2002. P. 233–236. <https://doi.org/10.1109/ICDSP.2002.1027877>
10. Grzeszczak A., Yeap T., Panchanathan S. VLSI architecture for discrete wavelet transform // *Proc. of the Canadian Conference on Electrical and Computer Engineering*. V. 2. 1994. P. 461–464. <https://doi.org/10.1109/CCECE.1994.405788>
11. Grzeszczak A., Mandal M., Panchanathan S. VLSI implementation of discrete wavelet transform // *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*. 1996. V. 4. N 4. P. 421–433. <https://doi.org/10.1109/92.544407>
12. Swami S., Mulani A. An efficient FPGA implementation of discrete wavelet transform for image compression // *Proc. of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*. 2017. P. 385–3389. <https://doi.org/10.1109/ICECDS.2017.8390088>
13. Nagaraj P., Rajasekaran M., Muneeswaran V., Sudar K.M., Gokul K. VLSI implementation of image compression using TSA optimized discrete wavelet transform techniques // *Proc. of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2020. P. 667–670. <https://doi.org/10.1109/ICSSIT48917.2020.9214220>
14. Sakthivel S. Sankar A. Real time watermarking of grayscale images using integer DWT transform // *Proc. of the International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA)*. 2016. P. 7593056. <https://doi.org/10.1109/VLSI-SATA.2016.7593056>
15. Xiong C., Tian J., Liu J. A fast VLSI architecture for two-dimensional discrete wavelet transform based on lifting scheme [image compression applications] // *Proc. of the 7th International Conference on Solid-State and Integrated Circuits Technology*. V. 3. 2004. P. 1661–1664. <https://doi.org/10.1109/ICSICT.2004.1435150>
16. Xiong C., Tian J., Liu J. Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme // *IEEE Transactions on Image Processing*. 2007. V. 16. N 3. P. 607–614. <https://doi.org/10.1109/TIP.2007.891069>
17. Zhang C., Long Y., Kurdahi F. A hierarchical pipelining architecture and FPGA implementation for lifting-based 2-D DWT // *Journal of Real-Time Image Processing*. 2007. V. 2. N 4. P. 281–291. <https://doi.org/10.1007/s11554-007-0057-6>
18. Сай С.В., Шоберг А.Г. Способ управления качеством передачи мелких структур изображений в стандарте JPEG2000 // *Компьютерная оптика*. 2020. Т. 44. № 3. С. 401–408. <https://doi.org/10.18287/2412-6179-co-616>
1. Taubman D., Marcellin M.W. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2002. 774 p.
2. Vorobev V.I., Gribunin V.G. *Principles and Practices of the Wavelet Transform*. St.Petersburg, 1999, 208 p. (in Russian)
3. Mallat S. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1989, vol. 11, no. 7, pp. 674–693. <https://doi.org/10.1109/34.192463>
4. Cohen A., Daubechies I., Feauveau J.-C. Biorthogonal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics*, 1992, vol. 45, no. 5, pp. 485–560. <https://doi.org/10.1002/cpa.3160450502>
5. Daubechies I., Sweldens W. Factoring wavelet transforms into lifting steps. *Journal of Fourier Analysis and Applications*, 1998, vol. 4, no. 3, pp. 247–269. <https://doi.org/10.1007/BF02476026>
6. Sameen I., Chang Y.C., Ng M.S., Goi B.-M., Ooi C.-P. A Unified FPGA-based system architecture for 2-D discrete wavelet transform. *Journal of Signal Processing Systems*, 2013, vol. 71, no. 2, pp. 123–142. <https://doi.org/10.1007/s11265-012-0687-1>
7. Rekha K., Ravi K. Design of high speed lifting based DWT using 9/7 wavelet transform for image compression. *Proc. of the International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, 2017, pp. 132–137. <https://doi.org/10.1109/ICRAECT.2017.38>
8. Zhong X., Jiang H., Cao H., Yang R. Efficient lifting based CDF9/7 wavelet transform using fixed point. *Proc. of the 3rd International Congress on Image and Signal Processing*, 2010, pp. 3094–3097. <https://doi.org/10.1109/CISP.2010.5648075>
9. Dimitroulakos G., Zervas N., Sklavos N., Goutis C. An efficient VLSI implementation for forward and inverse wavelet transform for JPEG2000. *Proc. of the 14th International Conference on Digital Signal Processing Proceedings (DSP)*. V. 1, 2002, pp. 233–236. <https://doi.org/10.1109/ICDSP.2002.1027877>
10. Grzeszczak A., Yeap T., Panchanathan S. VLSI architecture for discrete wavelet transform. *Proc. of the Canadian Conference on Electrical and Computer Engineering*. V. 2, 1994, pp. 461–464. <https://doi.org/10.1109/CCECE.1994.405788>
11. Grzeszczak A., Mandal M., Panchanathan S. VLSI implementation of discrete wavelet transform. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1996, vol. 4, no. 4, pp. 421–433. <https://doi.org/10.1109/92.544407>
12. Swami S., Mulani A. An efficient FPGA implementation of discrete wavelet transform for image compression. *Proc. of the International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)*, 2017, pp. 385–3389. <https://doi.org/10.1109/ICECDS.2017.8390088>
13. Nagaraj P., Rajasekaran M., Muneeswaran V., Sudar K.M., Gokul K. VLSI implementation of image compression using TSA optimized discrete wavelet transform techniques. *Proc. of the Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2020, pp. 667–670. <https://doi.org/10.1109/ICSSIT48917.2020.9214220>
14. Sakthivel S. Sankar A. Real time watermarking of grayscale images using integer DWT transform. *Proc. of the International Conference on VLSI Systems, Architectures, Technology and Applications (VLSI-SATA)*, 2016, pp. 7593056. <https://doi.org/10.1109/VLSI-SATA.2016.7593056>
15. Xiong C., Tian J., Liu J. A fast VLSI architecture for two-dimensional discrete wavelet transform based on lifting scheme [image compression applications]. *Proc. of the 7th International Conference on Solid-State and Integrated Circuits Technology*. V. 3, 2004, pp. 1661–1664. <https://doi.org/10.1109/ICSICT.2004.1435150>
16. Xiong C., Tian J., Liu J. Efficient architectures for two-dimensional discrete wavelet transform using lifting scheme. *IEEE Transactions on Image Processing*, 2007, vol. 16, no. 3, pp. 607–614. <https://doi.org/10.1109/TIP.2007.891069>
17. Zhang C., Long Y., Kurdahi F. A hierarchical pipelining architecture and FPGA implementation for lifting-based 2-D DWT. *Journal of Real-Time Image Processing*, 2007, vol. 2, no. 4, pp. 281–291. <https://doi.org/10.1007/s11554-007-0057-6>
18. Sai S.V., Shoberg A.G. Quality control method of the transmission of fine image details in the JPEG2000. *Computer Optics*, 2020, vol. 44, no. 3, pp. 401–408. (in Russian). <https://doi.org/10.18287/2412-6179-co-616>

Авторы

Сай Сергей Владимирович — доктор технических наук, профессор, заведующий кафедрой, Тихоокеанский государственный университет, Хабаровск, 680035, Российская Федерация, [sc 14042260800](https://orcid.org/0000-0002-4506-1857), <https://orcid.org/0000-0002-4506-1857>, sai1111@rambler.ru

Зинкевич Алексей Владимирович — кандидат технических наук, доцент, Тихоокеанский государственный университет, Хабаровск, 680035, Российская Федерация, <https://orcid.org/0000-0003-3231-8569>, zinkevich1985@mail.ru

Authors

Sergey V. Sai — D.Sc., Professor, Head of Department, Pacific National University, Khabarovsk, 680035, Russian Federation, [sc 14042260800](https://orcid.org/0000-0002-4506-1857), <https://orcid.org/0000-0002-4506-1857>, sai1111@rambler.ru

Alexey V. Zinkevich — PhD, Associate Professor, Pacific National University, Khabarovsk, 680035, Russian Federation, <https://orcid.org/0000-0003-3231-8569>, zinkevich1985@mail.ru

Статья поступила в редакцию 22.08.2022
Одобрена после рецензирования 10.11.2022
Принята к печати 15.01.2023

Received 22.08.2022
Approved after reviewing 10.11.2022
Accepted 15.01.2023



Работа доступна по лицензии
Creative Commons
«Attribution-NonCommercial»