

МОДЕЛИ И АРХИТЕКТУРА ВЕЙВЛЕТ-ПРЕОБРАЗОВАНИЙ ДЛЯ СТАНДАРТА JPEG 2000

Предлагается технология создания архитектуры и моделей конвейерного вычислительного устройства системного уровня описания лифтинговой схемы 9/7, которая существенно уменьшает цикл проектирования изделий цифровой обработки сигналов. Приводится практическая реализация фильтра, подтверждающая эффективность разработанного маршрута проектирования системного уровня на основе использования программы Simulink среды Matlab.

1. Введение

До настоящего времени работы, посвященные DWT-преобразованию, представляли собой теоретические разработки по алгоритмам кодирования, сжатия, преобразования информации, оценки потери качества изображения. При этом не учитывались особенности аппаратной реализации этих алгоритмов. В лучшем случае рассматривалась только программная реализация преобразователей. Исследовалось использование операции свертки или лифтинговой архитектуры для реализации DWT. Работы Добеши и Свелденса (Daubechies and W. Sweldens) [1,2] по применению факторизации в целях выполнения вейвлет-преобразований и предложенные в ней лифтинговые схемы интересны тем, что они позволяют значительно упростить аппаратную реализацию DWT.

В исследованиях, посвященных аппаратной реализации DWT-преобразований, различают архитектуры, базирующиеся на операции свертки или лифтинговых схемах. Модели, использующие свертку для выполнения DWT-преобразования, разделяются на последовательные и параллельные архитектуры. В других работах выполнена классификация устройств по способу использования внешней и внутренней памяти для хранения исходных данных, промежуточных и окончательных результатов. По способу взаимодействия с внешней памятью архитектуры, реализующие вейвлет-преобразования, делятся на три группы [3] level-by-level, block-based, line-based. Имеются предложения по модификации архитектур для упрощения вычислений за счет уменьшения количества иррациональных коэффициентов (flipping-архитектуры) [4]. При этом чаще всего используется десятичная арифметика. Некоторые из предлагаемых моделей архитектур ориентированы на реализацию только одного вида лифтинг-структуры, а другие позволяют выполнять DWT-преобразования с двухступенчатыми и четырехступенчатыми схемами лифтинга [5, 6-9].

Однако в существующих источниках не представлены: 1) Анализ архитектур с точки зрения возможности реализации на программируемой логике (FPGA). 2) Анализ реализации памяти на программируемой логике. 3) Выполнимость системных моделей архитектур в Simulink, в других подобных редакторах системного уровня.

Цель исследования состоит в разработке системной архитектуры реализации фильтра 9/7 на программируемой логике.

Для достижения поставленной цели решаются **задачи**: 1) Разработка архитектурной модели вейвлет-преобразования системного уровня. 2) Реализация и верификация системной модели с помощью пакета Simulink вычислительной среды Matlab.

2. Модель блока DWT-преобразования 9/7

Предложенная архитектура (рис. 1) относится к классу line-based. Она спроектирована для использования ее в качестве подкомпонента в устройстве JPEG2000 [9]. Предполагается, что данные в устройство подаются последовательно, по одному m -разрядному слову. Для реализации нескольких уровней преобразований используется внутренняя буферная память, хранящая LL-поддиапазон предыдущего уровня преобразования. Входные и выходные порты блока, реализующего DWT-преобразование, представлены в табл. 1, где m – разрядность входных данных, а n – разрядность полученных результатов. Оба параметра (m и n) определяют точность вычисления результатов DWT-преобразования.

Таблица 1. Порты блока одноуровневого преобразования

Имя	Описание	Размер
Порты входа		
Data	Вход данных из входного буфера	m
LL-in	LL-коэффициенты предыдущего уровня преобразования	n
Enable	Блокирует и разблокирует работу устройства	1
Rst	Сброс	1
Clk	Синхронизация	1
Порты выхода		
LL	Кoeffициенты LL-поддиапазона	n
LH	Кoeffициенты LH-поддиапазона	n
HL	Кoeffициенты HL-поддиапазона	n
HH	Кoeffициенты HH-поддиапазона	n
Done	Сигнализирует о готовности данных на выходах поддиапазонов	4
Read_data	Готовность устройства к приему входных данных	1
Read_LL	Готовность устройства к приему данных из LL поддиапазона	1
Size_tile	Размер tile (64, 128, 256, 512, 1024)	3
Level	Текущий уровень преобразования	3

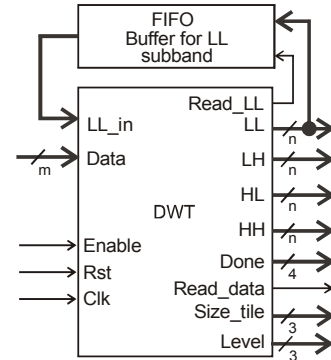


Рис. 1. Интерфейс блока одноуровневого преобразования DWT с буфером памяти LL поддиапазона

Длина буфера памяти для LL поддиапазона определяется выражением $Q_{LL} = S_{mem_FIFO} \times n$, где n – размер коэффициентов поддиапазонов, а S_{mem_FIFO} – вычисляется по формуле:

$$S_{mem_FIFO} = \frac{Tile^2}{4} - Lenght_{DWTpipeline},$$

где $Tile$ – размерность обрабатываемого блока, а параметр $Lenght_{DWTpipeline}$ – представляет собой длину конвейера DTW-преобразования.

В микросхемах программируемой логики буфер поддиапазона LL может быть реализован на выделенных блоках памяти силиконового кристалла.

На рис. 2 представлена структурная модель конвейерной схемы двумерного DWT-преобразования для банка фильтров 9/7 с четырехступенчатой лифтинговой схемой. Она позволяет оценить длину конвейера для каждого поддиапазона частот:

$$\begin{aligned} HH: & 2 + 8 + 4*Tile + 8 + 1 = 4*Tile + 19; \\ HL, LH: & 2 + 8 + 4*Tile + 11 + 1 = 4*Tile + 22; \\ LL: & 2 + 11 + 2*Tile + 11 + 1 = 4*Tile + 25. \end{aligned}$$

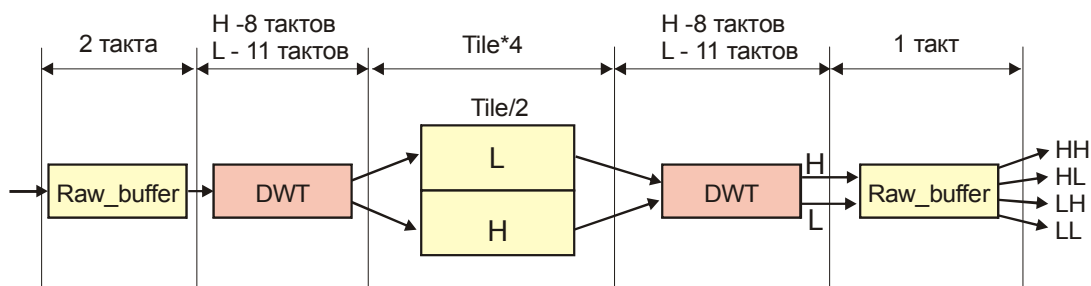


Рис. 2. Конвейерная схема обработки данных для DWT

Зависимость длины конвейера LL- поддиапазона DWT-преобразования для банка фильтров 9/7 представлена в табл. 2. Размерность буфера памяти определяется параметром $Tile$ и может быть уменьшена на длину конвейера DWT-преобразования (табл. 3).

Таблица 2. Зависимость длины конвейера DWT-преобразования для банка фильтров 9/7 от размера Tile

Tile	Length _{DWT pipeline}
64	281
128	537
256	1049
512	2073
1024	4121

Таблица 3. Зависимость размера буфера памяти от Tile

Tile	Tile ² /4, слова	Size_LL_FIFO, слова
64	1024	743
128	4096	3559
256	16384	15335
512	65536	63463
1024	262144	258023

3. Simulink модель DWT блока

В результате выполненных исследований была создана системная модель вычислительного устройства, ориентированного на вейвлет-преобразование с помощью Simulink-редактора среды Matlab. Simulink предоставляет пользователю графические средства для создания динамически функционирующих моделей, что позволяет проверить не только функцию, но и взаимодействие отдельных блоков устройства во времени. Общий вид модели представлен на рис. 3. Блок DWT реализует DWT-преобразования, FIFO memory for LL, идентифицирует блок памяти для хранения LL-поддиапазона.

Архитектура модуля преобразователя DWT блока представлена на рис. 4. Здесь блок Mux2 обозначает мультиплексор, который выбирает входные данные для последующей обработки. На первом уровне DWT-преобразования используется информация со входа data, на всех последующих уровнях – из LL_in.

Входной буфер Raw_biffer представляет собой последовательно-параллельный преобразователь, который формирует входные данные для арифметического блока Arithmetic Block for Raw. Вычислительное устройство принимает данные последовательно, с одновременной выдачей трех значений сигналов [7].

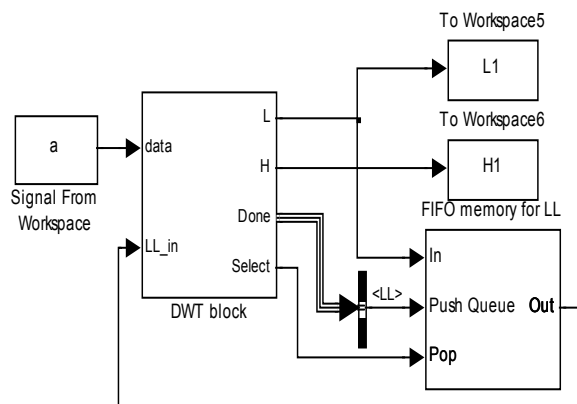


Рис. 3. Схема 2D-DWT-блока с FIFO памятью для коэффициентов LL-поддиапазона

4. Управляющий блок преобразователя

Схема управляющего блока (Subsystem, рис. 4) для 2D-DWT-преобразования представлена на рис. 5. Поскольку для реализации преобразования использовалась конвейерная архитектура, то для управления работой устройства целесообразно использовать классический автомат Мили или Мура. Управляющий блок построен на двух счетчиках, хранящих горизонтальную и вертикальную координаты обрабатываемого элемента изображения. На основе значений этих счетчиков формируются все управляющие сигналы устройства, которые разбиты на группы: C_ALU_raw и C_ALU_column – для арифметических блоков, выполняющих одномерное преобразование по строкам и по столбцам соответственно; Enable_L и Enable_H – разрешение записи информации во временный буфер памяти между обработкой сигналов по строкам и столбцам; pop_L и pop_H – чтение информации из временного буфера памяти; data_select – выбор входных данных для обработки; done – сигнал готовности данных в каждом поддиапазоне. Для такого типа управляющего блока уже технологично и просто может быть построен генератор HDL-кода [8] для последующей реализации в аппаратуру на кристалле.

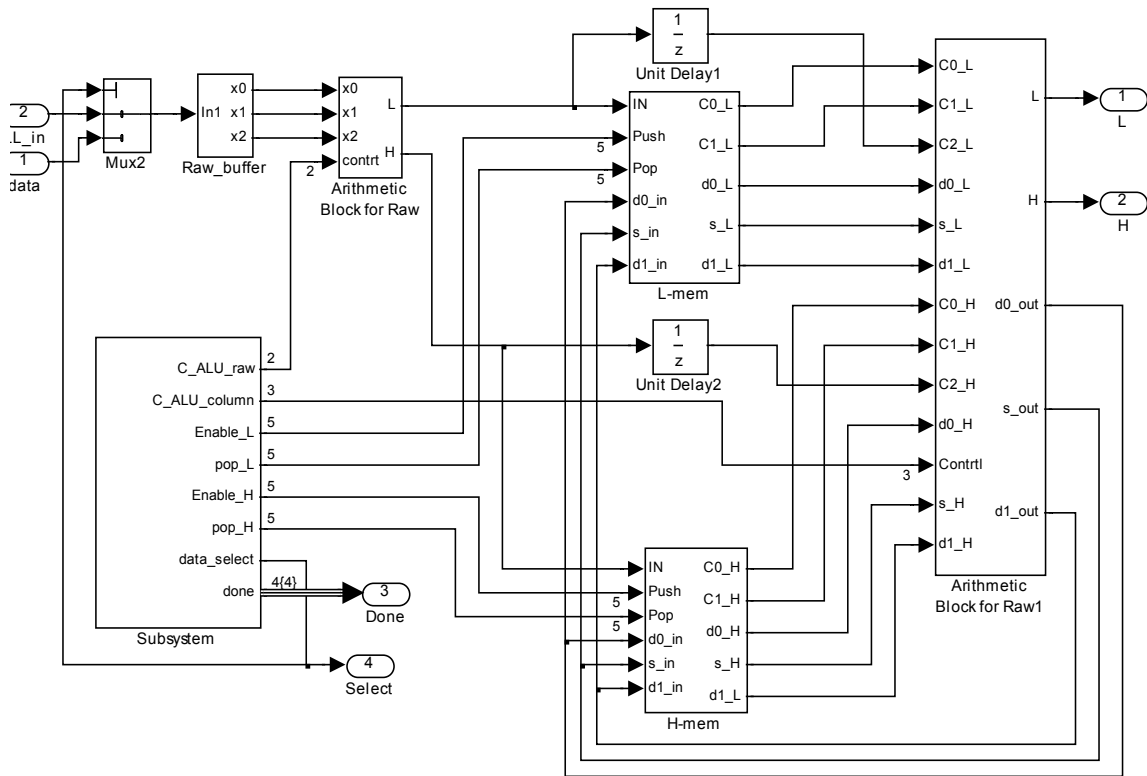


Рис. 4. Схема одноуровневого 2D-DWT 9/7 устройства

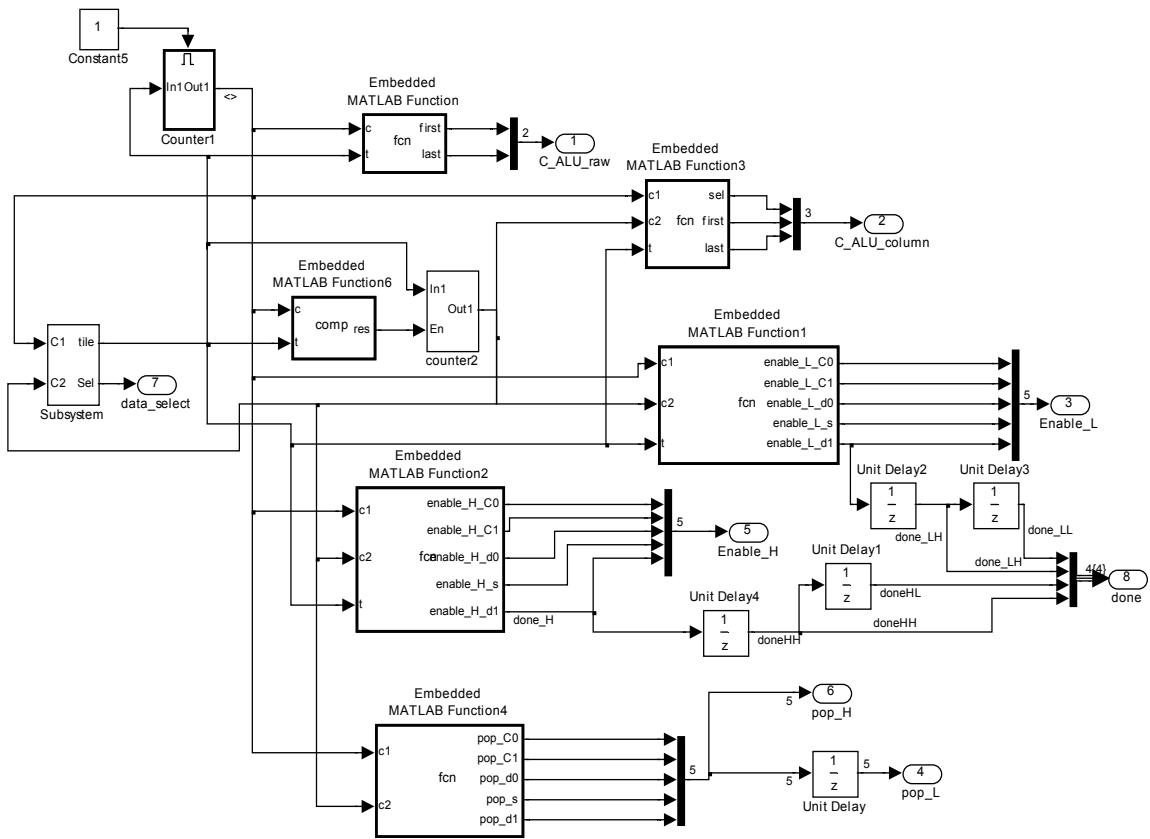


Рис. 5. Управляющий блок для 2D-DWT-преобразования

5. Арифметические блоки обработки строк и столбцов

Рис. 6 представляет собой модель арифметического блока для выполнения DWT Добеши преобразования по строкам, которая содержит управляющие сигналы для вычисления коэффициентов крайних элементов. Сигнал *first* используется в начале вычисления последовательности, а *last* – при завершении.

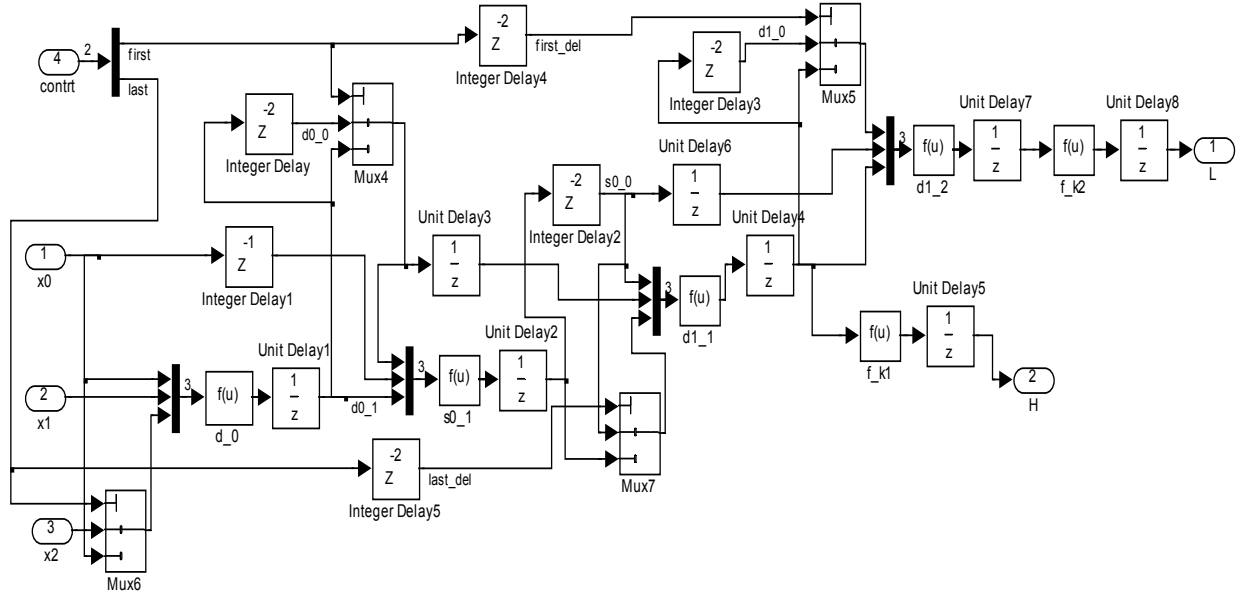


Рис. 6. Блок обработки данных по строкам для 4-ступенчатой лифтинговой схемы

Реализация математического дискретного вейвлет-преобразования DWT основывается на лифтинговой схеме (рис. 7), которая базируется на четырех основных операциях: разбиение (S), предсказание (P), обновление (U) и масштабирование (K).

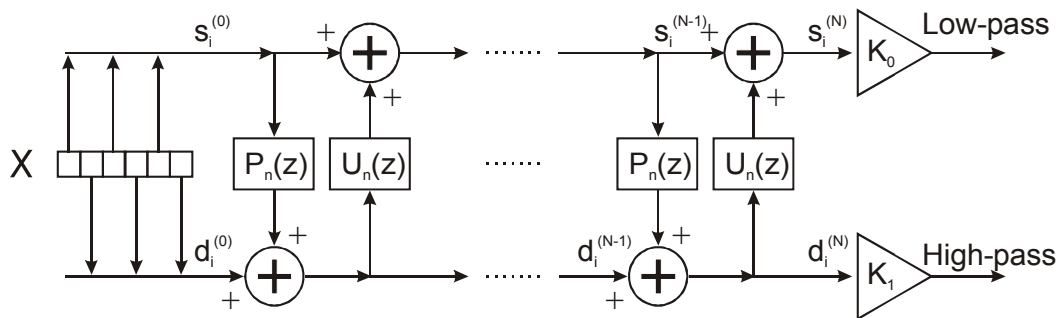


Рис. 7. Лифтинговая схема вейвлет-анализа: разбиение, предсказание и обновление

Значения элементов на каждом этапе преобразования вычисляется по формулам:

$$d_i^{(n)} = d_i^{(n-1)} + \sum_j P_n(j) s_j^{(n-1)}, n \in [1, 2, \dots, N],$$

$$s_i^{(n)} = s_i^{(n-1)} + \sum_j U_n(j) d_j^{(n)}, n \in [1, 2, \dots, N].$$

Для банка фильтров Добеши 9/7 $N=2$ используется четыре этапа построения вейвлет-преобразования. Выход последнего этапа предсказания умножается на масштабирующий коэффициент K_1 , а последнего этапа обновления – на величину K_0 .

Значения коэффициентов для преобразования Добеши имеют следующий вид:

$$P_1 = -1.586134342_059924(1+z);$$

$$U_1 = -0.052980118572961(1+z^{-1});$$

$$P_2 = 0.882911075530934(1+z);$$

$$U_2 = 0.443506852043971(1+z^{-1}).$$

При этом масштабирующие коэффициенты определяются в следующем виде:

$$K_0 = \frac{1}{K}, \quad K_1 = \frac{K}{2}, \quad \text{где } K = 1,230174104914001.$$

Для вычисления коэффициентов граничных элементов, при условии четного числа входных элементов, используется зеркальное отображение входных значений данных, как это изображено на рис. 8, где представлено формирование результатов DWT-преобразования для первых элементов входной последовательности, и на рис. 9, где проиллюстрировано формирование результатов DWT-преобразования последних элементов входной последовательности данных.

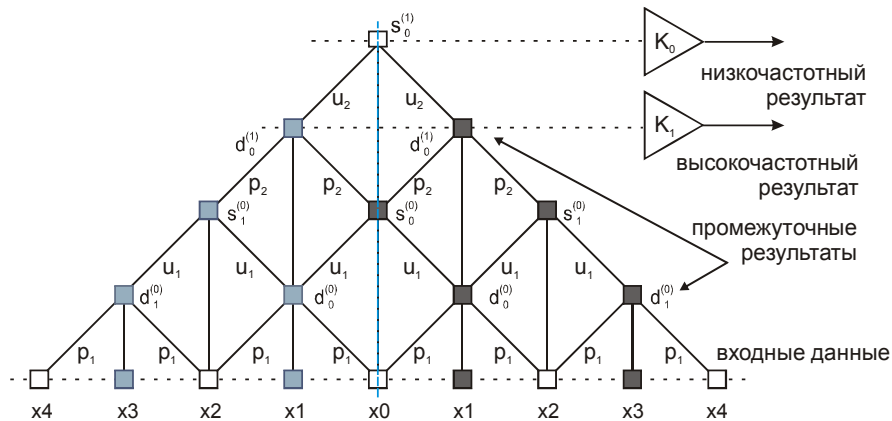


Рис. 8. Формирование результатов DWT-преобразования для первых элементов входной последовательности

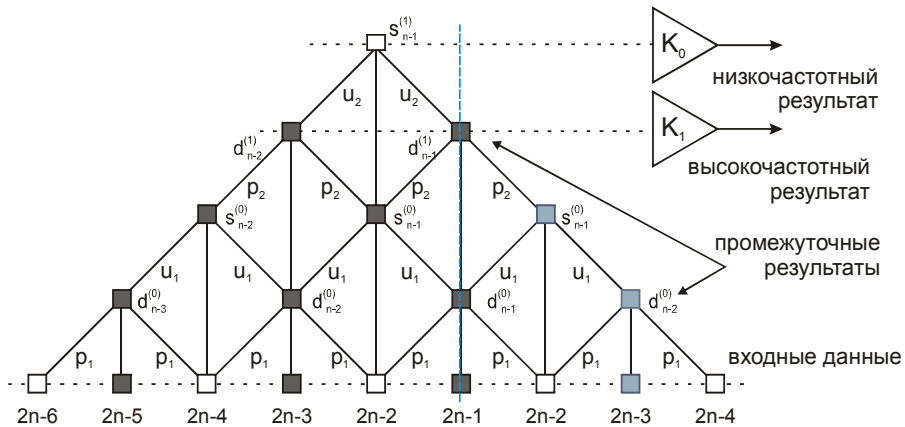


Рис. 9. Формирование результатов DWT-преобразования последних элементов входной последовательности

В табл. 4 и 5 представлены примеры для иллюстрации вычисления коэффициентов DWT-преобразования для граничных элементов.

Для хранения результатов одномерного DWT-преобразования изображения по строкам перед выполнением преобразования по столбцам используется специальная буферная память, построенная на пяти блоках памяти FIFO (рис. 10). Два блока памяти (C0 и C1) предназначены для хранения результатов DWT-преобразования двух строк, два других (d0 и s) – для хранения коэффициентов предсказания и обновления нулевого уровня, пятый (d1) – для хранения коэффициентов предсказания первого уровня.

Таблица 4. Порядок обработки данных низкочастотных коэффициентов для первых элементов входной последовательности в Excel

										L= 31,66835	
										$s^{(1)}_0$	
										38,9575	
		H= 0,62507		$d^{(1)}_0$				$d^{(1)}_0$		H= 0,62507	
		1,01623		1,01623				1,01623			
		$s^{(0)}_0$				$s^{(0)}_1$				$s^{(0)}_2$	
		49,2069				38,0560				49,2069	
		$d^{(0)}_1$				$d^{(0)}_0$				$d^{(0)}_1$	
		-97,7517				-76,0291				-97,7517	
x4	x3	x2	x1	x0	x1	x2	x3	x4			
50	45	40	35	30	35	40	45	50			

Таблица 5. Порядок обработки данных низкочастотных коэффициентов для последних элементов входной последовательности в Excel

										L= 32,35174	
										$s^{(1)}_3$	
										39,7981	
		H= 0,16871		$d^{(1)}_2$				$d^{(1)}_3$		H= 3,41277	
		0,27429		5,54845				5,54845			
		$s^{(0)}_2$				$s^{(0)}_3$				$s^{(0)}_3$	
		49,2069				37,2157				37,2157	
		$d^{(0)}_1$				$d^{(0)}_2$				$d^{(0)}_3$	
		-97,7517				-76,0291				-60,1678	
x2	x3	x4	x5	x6	x7	x6	x5	x4			
50	45	40	35	30	35	30	35	40			

Выполнение DWT-преобразования по столбцам начинается после завершения DWT-преобразования первых двух строк. Результаты преобразования этих двух строк заносятся в банки памяти (C0 и C1). В момент DWT-преобразования третьей строки параллельно считывается информация из блоков C0 и C1 и поступает на арифметический блок DWT-преобразования по столбцам. Результат преобразования третьей строки сохраняется в банке C0, а вычисленные коэффициенты предсказания и обновления нулевого уровня – в банках памяти d и s, далее они будут участвовать в вычислении коэффициентов предсказания и обновления первого уровня. При выполнении преобразования четвертой строки, информация сохраняется в банке C1. На этом этапе преобразования по столбцам не происходит. Табл. 6 представляет порядок формирования данных в памяти.

Таблица 6. Порядок обработки данных по столбцам для N=16 (N/2=8)

Строка	Память				Этапы лифтинга				
	C0	C1	d0	s	d1	$d^{(0)}$	$s^{(0)}$	$d^{(1)}$	$s^{(1)}$
0	C0								
1		C1							
2	C3		$d^{(0)}_0$	$s^{(0)}_0$		$d^{(0)}_0=C1-P_1(C0+C2)$	$s^{(0)}_0=C0-U_1(d^{(0)}_0+d^{(0)}_0)$		
3		C4							
4	C5		$d^{(0)}_1$	$s^{(0)}_1$	$d^{(1)}_0$	$d^{(0)}_1=C3-P_1(C2+C4)$	$s^{(0)}_1=C2-U_1(d^{(0)}_0+d^{(0)}_1)$	$d^{(1)}_0=d^{(0)}_0-P_2(s^{(0)}_0+s^{(0)}_1)$	$s^{(1)}_0=s^{(0)}_0-U_2(d^{(1)}_0+d^{(1)}_0)$
5		C6							
6	C7		$d^{(0)}_2$	$s^{(0)}_2$	$d^{(1)}_1$	$d^{(0)}_2=C5-P_1(C4+C6)$	$s^{(0)}_2=C4-U_1(d^{(0)}_1+d^{(0)}_2)$	$d^{(1)}_1=d^{(0)}_1-P_2(s^{(0)}_1+s^{(0)}_2)$	$s^{(1)}_1=s^{(0)}_1-U_2(d^{(1)}_0+d^{(1)}_1)$
7		C8							
8	C9		$d^{(0)}_3$	$s^{(0)}_3$	$d^{(1)}_2$	$d^{(0)}_3=C7-P_1(C6+C8)$	$s^{(0)}_3=C6-U_1(d^{(0)}_2+d^{(0)}_3)$	$d^{(1)}_2=d^{(0)}_2-P_2(s^{(0)}_2+s^{(0)}_3)$	$s^{(1)}_2=s^{(0)}_2-U_2(d^{(1)}_1+d^{(1)}_2)$
9		C10							
10
	Запись								
	Чтение и запись								

Поскольку вычисление результатов происходит через строку, то один и тот же арифметический блок может быть задействован поочередно для обработки низкочастотных и высокочастотных результатов одномерного DWT-преобразования. Архитектура обработки данных по столбцам представлена на рис. 11. Блок Input_Select используется для переключения режимов вычислительных процессов в целях получения низкочастотных и высокочастотных результатов одномерного DWT-преобразования, которые являются в данном блоке входными значениями.

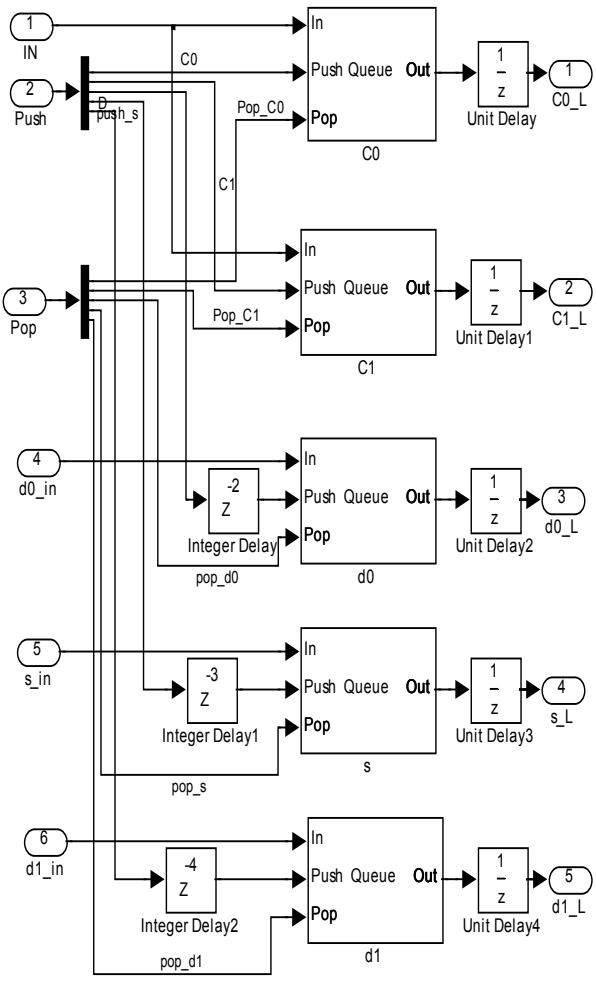


Рис. 10. Схема буфера памяти строк (L-mem)

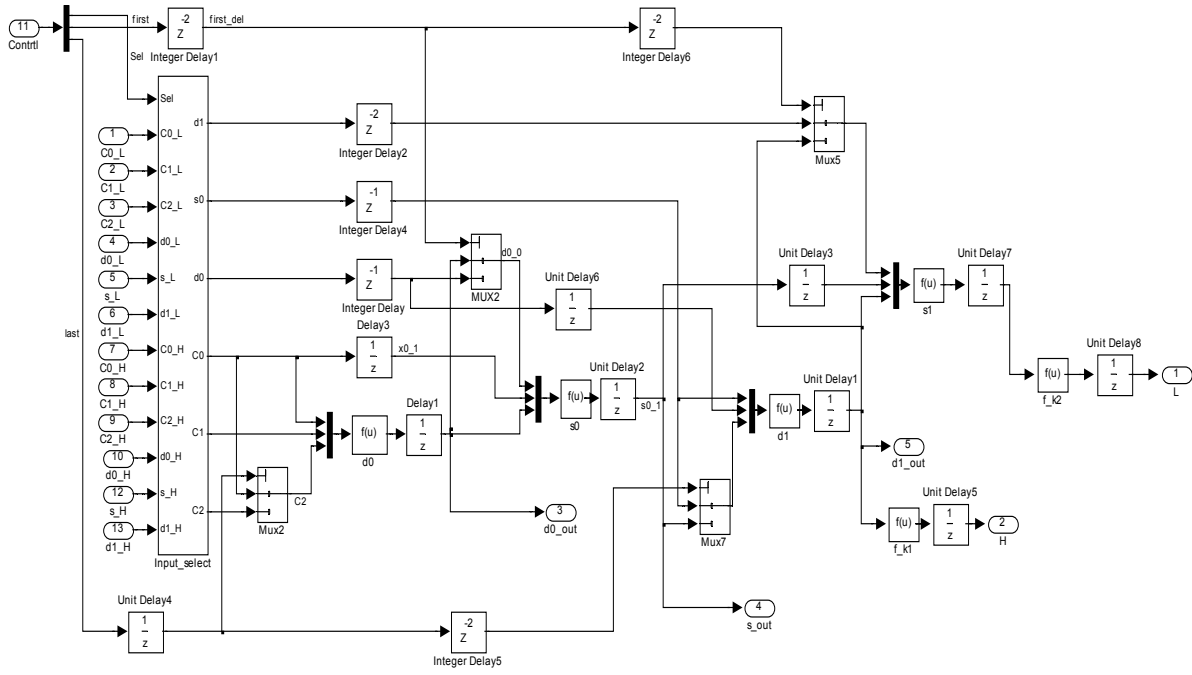


Рис. 11. Арифметический блок для обработки данных по столбцам (Arithmetic Block)

6. Заключение

Научная новизна – предложена системная модель вейвлет-преобразования 9/7 цифровой обработки данных, реализованная в Simulink. Модель позволяет однозначно трактовать работу устройства. Спроектирован специализированный вычислитель, работающий по принципу конвейера, и управляющий блок устройства. Выполнено моделирование и верификация режимов работы памяти.

Технология, заключающаяся в предварительной системной разработке структуры DWT-вычислителя, существенно упрощает последующий процесс проектирования RTL-модели устройства на VHDL и Verilog.

Практическая значимость – существенное уменьшение (на 30-50%) цикла проектирования конвейерного устройства цифровой обработки сигналов за счет предварительной разработки и верификации архитектурной модели системного уровня по заданной спецификации.

Преимущества предложенной технологии создания иерархических системных моделей проектирования:

1) Наглядность описания модели устройства в виде системной архитектуры, возможность моделирования и верификации функциональности изделия до создания RTL-модели на языках HDL.

2) Технологичность и инвариантность предложенных системных архитектурных моделей относительно написания VHDL -кода, наглядность процедур тестирования, диагностирования и коррекции ошибок передачи данных или формирования управляющих сигналов, что значительно упрощает весь процесс проектирования и на 30% сокращает время разработки конвейерных вычислительных устройств.

3) Дальнейшие исследования связаны с анализом и определением зависимости качества изображения от разрядности коэффициентов лифтинга, аппаратурной сложности, промежуточных и конечных результатов. Интересным является направление, связанное с получением оптимального решения в отношениях «качество изображения – размер аппаратуры» для реализации устройства. Также актуальными на сегодня следует считать аппаратную реализацию в кристалле программируемой логики, полную верификацию HDL-модели, автоматическое генерирование управляющего блока конвейерных архитектур.

Список литературы: 1. *Daubechies I. and Sweldens W.* Factoring wavelet transforms into lifting steps. Bell Laboratories, Lucent Technologies. 1996. 368p. 2. *Daubechies I. and Sweldens W.* Factoring wavelet transforms into lifting schemes. J. Fourier Anal. Appl. Vol. 4. 1998. P. 247-269. 3. *Kotteri K.A., Barua S., Bell A.E., and Carletta J.E.* A comparison of hardware Implementations of the biorthogonal 9/7 DWT: Convolution Versus Lifting – IEEE Transactions on circuits and systems. II: Express Briefs, Vol. 52, No. 5, May 2005. P. 256-260. 4. *Liao H., Mandal M.K., and Cockburn B.F.* Novel Architectures for Lifting-based Discrete Wavelet Transform. *Electronics Letters*. Vol. 38. 2002. Issue 18. P. 1010-1012. 5. *Dimitroulakos G., Zervas N.D., Sklavos N., Goutis C.E.* An efficient VLSI implementation for forward and inverse wavelet transform for JPEG2000. Proc. of 14th IEEE International Conference on Digital Signal Processing (DSP'02). Greece. 2002. P. 233-236. 6. *Abdullah Al Muhit, Md. Shabiul Islam, Masuri Othman.* VLSI Implementation of Discrete Wavelet Transform (DWT) for Image Compression. 2nd International Conference on Autonomous Robots and Agents. 2004. New Zealand. P. 391-395. 7. *Irina Hahanova, Irina Pobegenko, Wade Ghribi, Hassan Kteiman.* Lifting structure implementation of discrete wavelet transform for filter bank 5/3. TCSET'06. 2006. P. 293 -296. 8. *Hahanova I., Miroshnychenko Y., Pobegenko I., Savvutin O.* Design Models of Pipelined Units for Digital Signal Progressing. Proceedings of the IX International Conference The Experience of Designing and CAD Systems in Microelectronics CADSM'07. Ukraine. 2007. P. 87-91. 9. *Tinku Acharya, Ping-Sing Tsai* JPEG2000 Standard for Image Compression. Concepts, Algorithms and VLSI Architectures. J.Wiley & Sons. New Jersey. 2005. 292 p.

Поступила в редколлегию 19.06.2007

Хаханова Ирина Витальевна, докторантка кафедры АПВТ ХНУРЭ. Научные интересы: проектирование цифровых систем на кристаллах. Увлечения: английский язык, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanova@mail.ru

Побеженко Ирина Александровна, аспирантка кафедры АПВТ ХНУРЭ. Научные интересы: проектирование цифровых систем на кристаллах. Увлечения: английский язык, музыка. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: irina_pob@ukr.net

Хаханов Владимир Иванович, декан факультета КИУ, д-р техн. наук, профессор кафедры АПВТ ХНУРЭ. Научные интересы: проектирование и тестирование цифровых систем. Увлечения: футбол, горные лыжи, путешествия. Адрес: Украина, 61166, Харьков, пр. Ленина, 14, тел. 70-21-326. E-mail: hahanov@kture.kharkov.ua