

УДК 004.272.2

ОПТИМИЗИРОВАННЫЕ АЛГОРИТМЫ ПРЕОБРАЗОВАНИЙ ХААРА ДЛЯ ОТЕЧЕСТВЕННОЙ ПЛАТФОРМЫ DSP “ELCORE” И ИХ ПРИМЕНЕНИЕ В ЗАДАЧАХ ГРАФИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ

А. Н. Милов¹

Рассматриваются особенности реализации одномерного и двумерного преобразований Хаара дискретных сигналов для отечественной платформы DSP “Elcore”, а также их применение для построения огрубленных версий сеток высот местности и построения списков уменьшенных образов текстур.

1. Введение. Научно-производственный центр “Электронные вычислительно-информационные системы” (НПЦ ЭЛВИС; <http://multicore.ru/>) является разработчиком собственной IP-ядерной платформы проектирования систем на кристалле “Мультикор”, в состав которой входит библиотека оригинальных ядер цифровых сигнальных процессоров (DSP) серии “Elcore™”. Процессоры DSP используются в решении задач из таких важных областей, как радиолокация, связь, обработка, кодирование и сжатие сигналов, изображения и звука.

Процессоры серии “Мультикор” являются многоядерными микроконтроллерами с двумя типами ядер [1]: 1) RISC-ядро — является MIPS32-совместимым (в стиле R4000), выполняет функции управления и является ведущим (master) и 2) DSP-ядро “Elcore” — является высокопроизводительным 32-битным ведомым (slave) устройством и имеет типичную для многих цифровых процессоров обработки сигналов гарвардскую архитектуру с внутренним параллелизмом по потокам и форматам обрабатываемых данных. Это ядро предназначено для высокоскоростной обработки информации в форматах с фиксированной и с плавающей точкой. Существенным достоинством DSP-ядра для задач обработки компьютерной графики является то, что система его инструкций обеспечивает одновременное выполнение в течение одного командного цикла до двух вычислительных операций и до двух пересылок. Достигается это наличием многопортового регистрового файла и оригинальной архитектуры арифметико-логического устройства (АЛУ).

Все вычислительные операции, поддерживаемые “Elcore”, подразделяются на два типа в зависимости от операционных исполняющих устройств:

- 1) операции типа OP1 — это операции, исполняемые АЛУ;
- 2) операции типа OP2 — это операции, исполняемые при помощи умножителя-сдвигателя (УС).

Параллельное выполнение двух вычислительных операций DSP-ядром “Elcore” возможно только при условии, что они выполняются на разных операционных устройствах (более подробно см. [2]).

Кроме того, DSP “Elcore” позволяет, оставаясь в рамках одной системы инструкций, наращивать количество вычислительных секций (по SIMD-типу), увеличивая тем самым общую производительность системы.

DSP-ядро “Elcore” имеет два типа встроенной памяти, условно обозначаемые X и Y. X-память используется как для записи, так и для чтения; операции обмена с этой памятью могут осуществляться для любого регистра DSP. Y-память является подмножеством X-памяти и может быть использована только для чтения входных данных в регистр R0. Вся встроенная память “Elcore” является двухпортовой. Это означает, что обращения к ней со стороны RISC-ядра или DMA-канала памяти происходят без приостановки конвейера “Elcore”.

Преобразование Хаара является простейшим вейвлет-преобразованием, выделяющим высокочастотную (ВЧ) и низкочастотную (НЧ) составляющие сигнала. Рекурсивные формулы одномерного преобразования Хаара для сигнала $x = \{x_j\}_{j \in Z}$ записываются следующим образом [3]:

$$x_j^{(i)} = \frac{1}{2} (x_{2j}^{(i+1)} + x_{2j+1}^{(i+1)}), \quad y_j^{(i)} = \frac{1}{2} (x_{2j}^{(i+1)} - x_{2j+1}^{(i+1)}),$$

¹ Научно-производственный центр “Электронные вычислительно-информационные системы” (НПЦ ЭЛВИС), Южная промышленная зона, проезд 4922, Технопарк “Зеленоград”, строение 2, 124460, Зеленоград, Москва; e-mail: quattro@elvees.com



где i — уровень разрешения сигнала (для рекурсивной свертки), j — индекс соответствующего отсчета сигнала x или y , y — высокочастотная составляющая сигнала x .

В работе [3] было показано, что двумерное преобразование Хаара можно реализовывать не по общей схеме композиций одномерных преобразований, а по следующим формулам:

$$\begin{aligned}
 x_{jk}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)} + x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}), \\
 y_{jk}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)} - x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}), \\
 y'_{jk}{}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)} + x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}), \\
 y''_{jk}{}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)} - x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}).
 \end{aligned}
 \tag{1}$$

В настоящей статье показан способ реализации этих преобразований на практике и дана оценка производительности для отечественной платформы DSP “Elcore”.

2. Особенности реализации одномерного преобразования Хаара для DSP “Elcore”. Предлагается следующий конвейерный алгоритм обработки данных одномерного преобразования Хаара для чисел в формате IEEE-754 (рис. 1).

В строках указана последовательность тактов DSP, в столбцах приведены первые шесть 32-битных регистров из регистрового файла “Elcore”. Условные обозначения:  — операция OP1;  — операция OP2; 0.5 — константное значение в регистре; R0*R10 — операция умножения регистров в формате с плавающей точкой; R2 ± R4 — операция “сложение-вычитание” регистров в формате с плавающей точкой (FAS), результаты помещаются в исходные регистры; (AT) + IT — загрузка входных данных из Y-памяти и постинкремент адресного регистра AT на величину IT (IT=1); (A0)+, (A1)+ — выгрузка обработанных данных в X-память и постинкремент соответствующего адресного регистра (A0, A1).

	R0	R2	R4	R6	R8	R10
1	(AT)+IT					0.5
2	(AT)+IT	R0*R10				0.5
3	(AT)+IT		R0*R10			0.5
4	(AT)+IT	R2±R4		R0*R10		0.5
5	(AT)+IT	(A0)+			R0*R10	0.5
6	(AT)+IT	R0*R10	(A1)+	R6±R8		0.5
7	(AT)+IT		R0*R10	(A0)+		0.5
8	(AT)+IT	R2±R4		R0*R10	(A1)+	0.5
9	(AT)+IT	(A0)+			R0*R10	0.5
10	(AT)+IT	R0*R10	(A1)+	R6±R8		0.5
11	(AT)+IT		R0*R10	(A0)+		0.5
12	(AT)+IT	R2±R4		R0*R10	(A1)+	0.5

Рис. 1. Реализация одномерного преобразования Хаара

Алгоритм использует две пары регистров (R2, R4) и (R6, R8) и чередует чтение, обработку и запись данных с одной пары на другую.

Входные данные для обработки загружаются из Y-памяти в регистр R0. Поскольку регистровый файл является многопортовым, то параллельно происходит операция деления предыдущего входного отсчета и результат помещается в регистр R2 (или R6).

Подготовка конвейера осуществляется до четвертого такта включительно. Основная стадия конвейера преобразования занимает восемь тактов (на рис. 1 она показана с 5 такта по 12 включительно). На каждой такой стадии из восьми отсчетов входного сигнала образуется по четыре отсчета НЧ и ВЧ составляющих. В течение основной стадии происходит 16 обменов с памятью данных (используется два адресных генератора — AGU и AGU-Y) и 16 арифметических операций с плавающей точкой в формате IEEE-754.

Заметим, что вместо операции FAS могли быть использованы операции FADD (сложение) и FSUB (вычитание), однако в этом случае потребовался бы как минимум еще один регистр из регистрового файла. Кроме того, операция FAS позволила освободить устройство АЛУ на 5 и 9 тактах. Как будет показано ниже, операция FAS наиболее эффективно может быть применена для конвейера двумерного преобразования Хаара.

Итак, производительность приведенного алгоритма ограничена выгрузкой результатов в X-память данных. Поэтому приведенный алгоритм является полностью оптимизированным для данного устройства.

Поскольку конвейер преобразования является довольно коротким, то особо важным моментом является минимизация затрат на выполнение подготовительных действий, связанных с организацией циклов. Для этого в DSP "Elcore" предусмотрена аппаратная поддержка программных циклов типа DO, которая позволяет избегать накладных расходов при организации простых циклов, связанных с инкрементом счетчика циклов и проверкой условия конца цикла. Необходимым требованием при использовании этого аппаратного механизма является то, чтобы число повторов цикла могло быть вычислено до вхождения в цикл. Помимо этого, это число должно находиться в пределах от 1 до 16383.

Основная стадия конвейера преобразования (5–12 такты на рис. 1), описанная на языке ассемблера "Elcore", приведена на рис. 2.

3. Реализация двумерного преобразования Хаара для DSP "Elcore". Для того чтобы столь же эффективно реализовать двумерное преобразование Хаара, необходимо представить формулы (1) в другом виде:

$$\begin{aligned}
 x_{jk}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)}) + \frac{1}{4} (x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}), \\
 y_{jk}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} + x_{2j+1,2k}^{(i+1)}) - \frac{1}{4} (x_{2j,2k+1}^{(i+1)} + x_{2j+1,2k+1}^{(i+1)}), \\
 y'_{jk}{}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)}) + \frac{1}{4} (x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}), \\
 y''_{jk}{}^{(i)} &= \frac{1}{4} (x_{2j,2k}^{(i+1)} - x_{2j+1,2k}^{(i+1)}) - \frac{1}{4} (x_{2j,2k+1}^{(i+1)} - x_{2j+1,2k+1}^{(i+1)}).
 \end{aligned} \tag{2}$$

Заметим, что все комбинации сумм/разностей операндов в (2) можно получить с использованием всего четырех операций FAS. Обозначим

$$\text{FAS}(n, m)_+ = n + m, \quad \text{FAS}(n, m)_- = n - m, \quad \begin{pmatrix} N \\ M \end{pmatrix} = \text{FAS}(n, m) \longleftrightarrow \begin{cases} N = \text{FAS}(n, m)_+, \\ M = \text{FAS}(n, m)_-. \end{cases}$$

Тогда двумерное преобразование Хаара может быть записано следующим образом:

$$\begin{aligned}
 \begin{pmatrix} x_{jk}^{(i)} \\ y_{jk}^{(i)} \end{pmatrix} &= \text{FAS} \left(\text{FAS} \left(\frac{1}{4} x_{2j,2k}^{(i+1)}, \frac{1}{4} x_{2j+1,2k}^{(i+1)} \right)_+, \text{FAS} \left(\frac{1}{4} x_{2j,2k+1}^{(i+1)}, \frac{1}{4} x_{2j+1,2k+1}^{(i+1)} \right)_+ \right), \\
 \begin{pmatrix} y'_{jk}{}^{(i)} \\ y''_{jk}{}^{(i)} \end{pmatrix} &= \text{FAS} \left(\text{FAS} \left(\frac{1}{4} x_{2j,2k}^{(i+1)}, \frac{1}{4} x_{2j+1,2k}^{(i+1)} \right)_-, \text{FAS} \left(\frac{1}{4} x_{2j,2k+1}^{(i+1)}, \frac{1}{4} x_{2j+1,2k+1}^{(i+1)} \right)_- \right).
 \end{aligned}$$

В соответствии с предложенной формой записи, алгоритм двумерного преобразования Хаара примет вид, изображенный на рис. 3. Условные обозначения на этом рисунке: (AT) ± IT, (AT) ± DT — загрузка входных данных из Y-памяти и постинкремент (декремент) адресного регистра AT на величину регистра смещения IT или DT (IT=1, DT=N), N — размер матрицы входного двумерного сигнала (предполагается, что матрица имеет N × N элементов).

Подготовка конвейера алгоритма занимает 8 тактов, основная стадия конвейера также занимает 8 тактов, в течение которых происходит 16 обращений к памяти и 24 арифметические операции с плавающей точкой. Основная стадия соответствует паре преобразований (2), на входе которых по четыре отсчета двумерного сигнала, на выходе — также по четыре отсчета одного шага двумерного преобразования Хаара.

DO Nd2, LastLabel:			
FMPY R0, R10, R8		R2.L, (A0) +	(AT) + IT, R0.L
FMPY R0, R10, R2	FAS R6, R8	R4.L, (A1) +	(AT) + IT, R0.L
FMPY R0, R10, R4		R6.L, (A0) +	(AT) + IT, R0.L
FMPY R0, R10, R6	FAS R2, R6	R8.L, (A1) +	(AT) + IT, R0.L
FMPY R0, R10, R8		R2.L, (A0) +	(AT) + IT, R0.L
FMPY R0, R10, R2	FAS R6, R8	R4.L, (A1) +	(AT) + IT, R0.L
FMPY R0, R10, R4		R6.L, (A0) +	(AT) + IT, R0.L
LastLabel:			
FMPY R0, R18, R6	FAS R4, R8	R2.L, (A0) +	(AT) + IT, R0.L

Рис. 2. Алгоритм основной стадии конвейера преобразования Хаара

	R0	R2	R4	R6	R8	R10	R12	R14	R16	R18	DT
1	(AT)+IT									0.25	N
2	(AT)+DT	R0*R18								0.25	N
3	(AT)-IT		R0*R18							0.25	N
4	(AT)+DT			R0*R18						0.25	N
5	(AT)+IT	R2±R4			R0*R18					0.25	N
6	(AT)+DT			R6±R8		R0*R18				0.25	N
7	(AT)-IT	R2+R6		R2-R6			R0*R18			0.25	N
8	(AT)+DT	(A0)+	R4+R8		R4-R8			R0*R18		0.25	N
9	(AT)+IT		(A1)+			R10±R12			R0*R18	0.25	N
10	(AT)+DT	R0*R18		(A2)+				R14±R16		0.25	N
11	(AT)-IT		R0*R18		(A3)+	R10+R14		R10-R14		0.25	N
12	(AT)+DT			R0*R18		(A0)+	R12+R16		R12-R16	0.25	N
13	(AT)+IT	R2±R4			R0*R18		(A1)+			0.25	N
14	(AT)+DT			R6±R8		R0*R18		(A2)+		0.25	N
15	(AT)-IT	R2+R6		R2-R6			R0*R18		(A3)+	0.25	N
16	(AT)+DT	(A0)+	R4+R8		R4-R8			R0*R18		0.25	N
17	(AT)+IT		(A1)+			R10±R12			R0*R18	0.25	N

Рис. 3. Реализация двумерного преобразования Хаара

Заметим, что результаты преобразования могут быть выданы в различные диапазоны X-памяти. Для этого необходимо настроить соответствующие адресные регистры A0 – A3.

Основная стадия конвейера предлагаемого алгоритма на языке ассемблера “Elcore” приведена на рис. 4.

Как видно из рисунка, на каждом такте происходит по два обмена с памятью данных, одна операция умножения и одна операция “сложение–вычитание”. Поскольку выходные данные могут быть записаны только в X-память, предложенный алгоритм также является полностью оптимизированным для данного устройства.

4. Замечания по масштабированию производительности предложенных алгоритмов. Масштабируемая архитектура операционных блоков DSP-ядер серии “Elcore” включает два режима обработки данных:

- один поток инструкций, один поток данных (Single Instruction Single Data — SISD),
- один поток инструкций, множественные потоки данных (Single Instruction Multiple Data — SIMD).

Отличие SIMD-режима от SISD-режима состоит в том, что за один командный цикл одна арифметическая инструкция выполняется в нескольких одинаковых операционных блоках одновременно (а именно, наборах ALU и UC), причем с разными данными. Таким образом, применение SIMD-масштабирования в DSP-разработке, выполненной в НПЦ ЭЛВИС, способно ускорить выполнение математической задачи в несколько раз. DSP-ядро “Elcore-24” микросхемы 1892BM2T “MC-24” обладает двумя операционными SIMD-секциями, т.е. вычисления в нем могут быть ускорены в два раза.

Преобразование Хаара локально, т.е. затрагивает только ту область сигнала, к которой оно применяется. Поэтому каждый шаг преобразования Хаара любого одномерного или двумерного сигнала можно осуществить, разложив исходный сигнал на несколько составляющих. Например, если необходимо осуществить один шаг двумерного преобразования Хаара для сигнала, заданного двумерной матрицей

DO Nd2, LastLabel:

```

FMPY R0,R18,R16 FAS R10,R12 R4.L, (A1)+ (AT)+DT,R0.L
FMPY R0,R18,R2 FAS R14,R16 R6.L, (A2)+ (AT)+IT,R0.L
FMPY R0,R18,R4 FAS R10,R14 R8.L, (A3)+ (AT)-DT,R0.L
FMPY R0,R18,R6 FAS R12,R16 R10.L, (A0)+ (AT)+IT,R0.L
FMPY R0,R18,R8 FAS R2,R4 R12.L, (A1)+ (AT)+DT,R0.L
FMPY R0,R18,R10 FAS R6,R8 R14.L, (A2)+ (AT)+IT,R0.L
FMPY R0,R18,R12 FAS R2,R6 R16.L, (A3)+ (AT)-DT,R0.L
LastLabel:
FMPY R0,R18,R14 FAS R4,R8 R2.L, (A0)+ (AT)+IT,R0.L
    
```

Рис. 4. Основная стадия алгоритма двумерного преобразования Хаара

Характеристики предложенных алгоритмов

Преобразование	DSP-ядро	Ограничения вх. сигналов, отсчетов	² Время преобразования вх. сигнала максимально возможного размера, мкс
Одномерное в формате IEEE-754	Elcore-14(SISD)	12 288	153,6
	Elcore-24(SIMD)	8 192	51,2
Двумерное в формате IEEE-754	Elcore-14(SISD)	12 100 (110 × 110)	151,25
	Elcore-24(SIMD)	8 100 (90 × 90)	50,625
³ Двумерное в формате R8G8B8X8	Elcore-14(SISD)	12 100 (110 × 110)	302,5
	Elcore-24(SIMD)	8 100 (90 × 90)	101,25

размером 128×128 , то такое преобразование эквивалентно четырем преобразованиям Хаара для составляющих сигнала с размерами 64×64 или к двум преобразованиям для составляющих сигнала с размерами 128×64 или 64×128 .

Таким образом, производительность приведенных алгоритмов преобразования Хаара может быть увеличена в два раза включением SIMD-режима и разложением исходного сигнала на составляющие.

5. Результаты. В таблице представлены характеристики производительности предложенных алгоритмов для различных DSP-ядер серии "Elcore". Ввиду того, что внутренняя память DSP-ядер реализована непосредственно на кристалле (а следовательно, существенно ограничена), в таблице также представлены граничные размеры входных сигналов, для которых возможно преобразование без разделения на составляющие.



Рис. 5. Список уменьшенных копий текстур, полученный с помощью преобразования Хаара

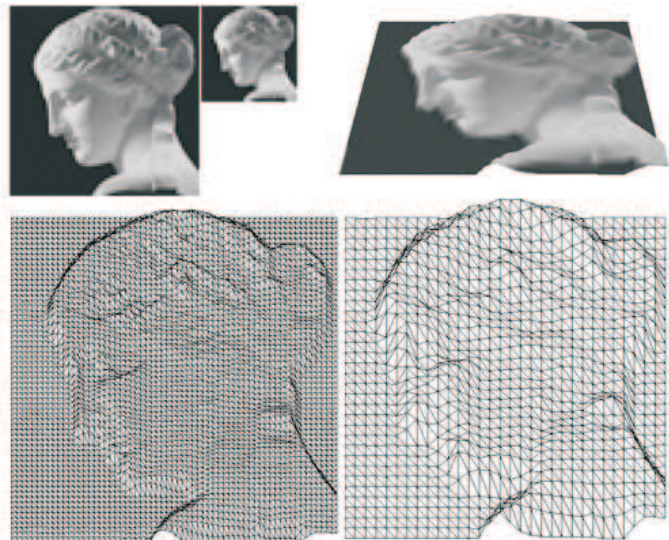


Рис. 6. Один шаг двумерного преобразования Хаара для заданной сетки высот

Для достижения указанной пиковой производительности при обработке непрерывных потоков данных обычно используют блочную обработку. Суть ее состоит в разделении сигнала на составляющие

² Время преобразования приведено для интегральных микросхем 1892BM3T (DSP-ядро Elcore-14) и 1892BM2T (DSP-ядро Elcore-24), работающих на частоте 80 МГц.

³ Широко используется для обработки растровых изображений и текстур. Этот формат известен также как 32-битный формат растровых изображений.

блоки такого размера, чтобы одновременно в памяти входных данных DSP могло содержаться два таких блока. Таким образом, в то время пока DSP производит обработку одного из блоков, DMA-канал производит выгрузку обработанных данных предыдущего и загрузку входных данных следующего блока.

Для эффективного разделения двумерных сигналов на составляющие и последующей их загрузки в Y-память в микроконтроллерах серии “Мультикор” предусмотрен режим двумерной DMA-передачи. Таким образом, нет необходимости в предварительном построении двумерных массивов данных или организации цепочек DMA-передач.

6. Применение преобразований Хаара. Преобразования Хаара применяются для построения списков уменьшенных копий текстур с размерами $2^n \times 2^n$, $n \in N$, при осуществлении пространственно-аккуратного текстурирования (mipmapping), см. рис. 5. Более подробно см., например, [4].

В работе [3] также было показано, что преобразование Хаара может применяться для сжатия сеток высот поверхностей, параметризуемых на плоскости. На рис. 6 приведены результаты применения одного шага двумерного преобразования Хаара к заданной сетке высот. Слева направо, сверху вниз: монотонные изображения сеток высот размерами 64×64 и 32×32 соответственно, перспективное представление поверхности, заданной сеткой высот, триангуляция исходной сетки высот 64×64 , триангуляция сетки высот 32×32 , полученной с помощью двумерного преобразования Хаара. Все изображения получены с использованием графической библиотеки с интерфейсом OpenGL для микроконтроллеров серии “Мультикор”.

СПИСОК ЛИТЕРАТУРЫ

1. Микросхема интегральная 1892ВМЗТ. Руководство пользователя. М.: ГУП НПЦ “ЭЛВИС”, 2005.
2. DSP-ядро ELcore-x4. Система инструкций. М.: ГУП НПЦ “ЭЛВИС”, 2006.
3. *Переберин А.В.* Многомасштабные методы синтеза и анализа изображений: Дисс. ... кандидата физико-математических наук. М.: ИПМ им. М.В. Келдыша, 2002.
4. *Williams L.* Pyramidal parametrics // *Computer graphics*. 17, N 3. 1983. 1–11.

Поступила в редакцию
02.03.2007
