

УДК 004.931; 004.932

А.С. Поляков, В.Е. Самсонов

ОЦЕНКА ХАРАКТЕРИСТИК АППАРАТНОЙ РЕАЛИЗАЦИИ НЕКОТОРЫХ АЛГОРИТМОВ ФИЛЬТРАЦИИ ИЗОБРАЖЕНИЙ

Приводятся данные о быстродействии и требуемых объемах оборудования при аппаратной реализации в базисе микросхем типа FPGA двух часто применяемых алгоритмов фильтрации изображений. Рассматривается возможность использования алгоритмов для обработки стандартных кадров видеопоследовательностей.

Введение

В настоящее время задачи отслеживания динамических объектов на видеопоследовательностях, снятых в видимом и инфракрасном диапазонах, являются объектом интереса огромного числа специалистов ведущих исследовательских центров и IT-компаний. Одной из основных трудностей, возникающих при решении указанной задачи, является малое время (0,03–0,04 с), отведенное на обработку каждого кадра видеопоследовательности стандартного размера 768x576 пикселей. Наиболее сложные известные алгоритмы, предназначенные для обработки видеопоследовательностей, непригодны для работы в режиме реального времени [1–3]. В этих условиях сокращение времени выполнения алгоритмов является одной из актуальных проблем в области обработки изображений, особенно при обработке видеопоследовательностей в реальном масштабе времени. Следует отметить, что изображения, полученные видеокамерами, как правило, содержат шум и поэтому нуждаются в предварительной обработке, улучшающей их качество. Поэтому одним из основных требований к алгоритмам обнаружения, отслеживания и распознавания объектов на полутоновых и цветных видеопоследовательностях является высокое быстродействие, достаточное для обработки каждого кадра получаемой цифровой видеoinформации в режиме реального времени при обеспечении должного качества.

Для улучшения качества изображений, получаемых видеокамерами в режиме реального времени, используются оконные линейные фильтры сверточного типа и медианные фильтры, которые считаются одними из лучших в области обработки изображений. Однако их применение при обработке видеопоследовательностей затруднено из-за значительной вычислительной сложности, приводящей к недопустимо большим временным затратам [2–4].

В связи с необходимостью выполнения сложных алгоритмов отслеживания динамических объектов в реальном масштабе времени при сохранении высокого качества видеопоследовательности актуальным является решение задачи повышения скорости выполнения алгоритмов фильтрации. Для увеличения их быстродействия наряду с оптимизацией применяются многопроцессорные системы или аппаратная реализация [5], которая является одним из самых эффективных способов увеличения производительности алгоритмов. К сожалению, вопросу обработки изображений путем их аппаратной реализации на основе современных микроэлектронных средств в отечественной литературе уделяется мало внимания. Настоящая статья посвящена частичному устранению пробела в этой области. В ней представлены данные о характеристиках аппаратной реализации в базисе микросхем типа FPGA [5–7] для двух часто применяемых на практике алгоритмов фильтрации изображений в терминах объемно-временных характеристик, которые используются при оценке результатов проектирования во всех САПР для FPGA и под которыми подразумеваются затраты оборудования, необходимые для реализации алгоритмов, и время их выполнения (производительность, быстродействие).

1. Краткое описание исследуемых алгоритмов

Фильтр «медиана». Медианой дискретной последовательности n элементов при нечетном n называется элемент, для которого существует не менее $(n-1)/2$ элементов, меньших или рав-

ных ему, и не менее $(n-1)/2$ элементов, больших или равных ему по величине. При рассмотрении элементарной матрицы размером 3×3 элемента для нахождения медианы требуется произвести не менее 19 операций сравнения [3], которые при программной реализации выполняются последовательно, т. е. требуется не менее 19 тактов вычислительного процесса. С учетом возможностей распараллеливания операций при аппаратной реализации алгоритма можно обойтись меньшим количеством тактов.

Рассмотрим элементарные матрицы \mathbf{M} , \mathbf{M}_1 , \mathbf{M}_2 размером 3×3 элемента (рис. 1). Видно, что при фильтрации изображений производится переычисление только одного элемента, расположенного на пересечении диагоналей матрицы.

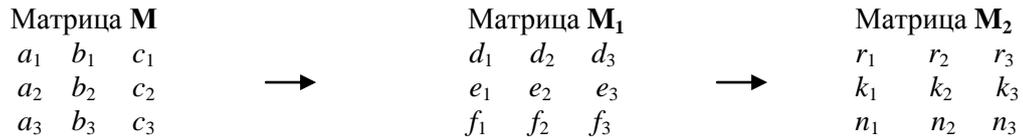


Рис. 1. Матрицы \mathbf{M} , \mathbf{M}_1 и \mathbf{M}_2

Для нахождения медианы необходимо упорядочить элементы матрицы по возрастанию (точнее, по неубыванию) значений. Пятый элемент упорядоченной таким образом последовательности из элементов $a_1, a_2, a_3, b_1, b_2, b_3, c_1, c_2, c_3$ является медианой рассматриваемого множества элементов. Для ее получения используем алгоритм, ориентированный на аппаратную реализацию.

1. Упорядочим элементы в столбцах матрицы по возрастанию (точнее, по неубыванию) значений. Это означает, что в каждом из столбцов матрицы предыдущий элемент равен или меньше следующего. Получим матрицу \mathbf{M}_1 , элементы которой находятся в отношениях $d_1 \leq e_1 \leq f_1, d_2 \leq e_2 \leq f_2, d_3 \leq e_3 \leq f_3$.

2. Упорядочим по неубыванию значений элементы в строках матрицы \mathbf{M}_1 . Получим матрицу \mathbf{M}_2 , в которой $r_1 \leq r_2 \leq r_3, k_1 \leq k_2 \leq k_3, n_1 \leq n_2 \leq n_3$.

3. Поскольку упорядочивание элементов матриц \mathbf{M}_1 и \mathbf{M}_2 производилось по неубыванию значений сверху вниз и слева направо, значения элементов в матрице \mathbf{M}_2 возрастают слева направо и сверху вниз, поэтому элементы r_1, r_2 и k_1 занимают первые три места в упорядоченной по неубыванию значений последовательности элементов матрицы \mathbf{M} . Следовательно, медианой матрицы является тот из элементов n_1, k_2, r_3 , который имеет среднее значение среди них.

С целью минимизации затрат для определения порядка следования сравниваемых элементов в столбцах и строках матриц \mathbf{M}_1 и \mathbf{M}_2 соответственно воспользуемся следующим приемом. Результат отношения неубывания будем описывать с помощью логических переменных p_1, p_2, p_3 :

$$\begin{aligned}
 p_1 &= 1, \text{ если } a_1 \leq a_2; \quad p_1 = 0, \text{ если } a_1 > a_2; \\
 p_2 &= 1, \text{ если } a_1 \leq a_3; \quad p_2 = 0, \text{ если } a_1 > a_3; \\
 p_3 &= 1, \text{ если } a_2 \leq a_3; \quad p_3 = 0, \text{ если } a_2 > a_3.
 \end{aligned} \tag{1}$$

На основании анализа значений переменных p_1, p_2 и p_3 можно установить порядок расположения элементов a_1, a_2, a_3 в последовательности по неубыванию значений. Результаты анализа представлены в табл. 1.

Таблица 1

Упорядочивание элементов (все отношения)

p_1	p_2	p_3	Порядок следования элементов по неубыванию значений
0	0	0	$a_1 \leq a_2 \leq a_3$
0	0	1	$a_1 \leq a_3 \leq a_2$
0	1	0	условия противоречивые
0	1	1	$a_3 \leq a_1 \leq a_2$
1	0	0	$a_2 \leq a_1 \leq a_3$
1	0	1	условия противоречивые
1	1	0	$a_2 \leq a_3 \leq a_1$
1	1	1	$a_3 \leq a_2 \leq a_1$

После удаления строк, содержащих противоречивые условия, получаем табл. 2.

Таблица 2

Упорядочивание элементов (без противоречивых условий)

p_1	p_2	p_3	Порядок следования элементов по неубыванию значений
0	0	0	$a_1 \leq a_2 \leq a_3$
0	0	1	$a_1 \leq a_3 \leq a_2$
0	1	1	$a_3 \leq a_1 \leq a_2$
1	0	0	$a_2 \leq a_1 \leq a_3$
1	1	0	$a_2 \leq a_3 \leq a_1$
1	1	1	$a_3 \leq a_2 \leq a_1$

Линейный *фильтр «свертка»* обеспечивает преобразование исходной матрицы \mathbf{M} в результирующую матрицу $\underline{\mathbf{S}}$ с помощью задаваемой пользователем матрицы \mathbf{S} поправочных коэффициентов. Рассматриваемые матрицы \mathbf{S} , \mathbf{M} и $\underline{\mathbf{S}}$ представлены ниже:

$$\mathbf{S} = \begin{vmatrix} s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,1} & s_{3,2} & s_{3,3} \end{vmatrix} \times \mathbf{M} = \begin{vmatrix} m_{1,1} & m_{1,2} & m_{1,3} \\ m_{2,1} & m_{2,2} & m_{2,3} \\ m_{3,1} & m_{3,2} & m_{3,3} \end{vmatrix} \rightarrow \underline{\mathbf{S}} = \begin{vmatrix} \underline{s}_{1,1} & \underline{s}_{1,2} & \underline{s}_{1,3} \\ \underline{s}_{2,1} & \underline{s}_{2,2} & \underline{s}_{2,3} \\ \underline{s}_{3,1} & \underline{s}_{3,2} & \underline{s}_{3,3} \end{vmatrix}$$

Значение элемента $m_{2,2}$ матрицы \mathbf{M} преобразуется в значение элемента $\underline{s}_{2,2}$ матрицы $\underline{\mathbf{S}}$ следующим образом:

– вычисляется

$$\begin{aligned} K = & s_{1,1} * m_{1,1} + s_{1,2} * m_{1,2} + s_{1,3} * m_{1,3} + s_{2,1} * m_{2,1} + s_{2,2} * m_{2,2} + \\ & + s_{2,3} * m_{2,3} + s_{3,1} * m_{3,1} + s_{3,2} * m_{3,2} + s_{3,3} * m_{3,3}; \end{aligned} \quad (2)$$

– в зависимости от величины K определяется значение $\underline{s}_{2,2}$ – результат «свертки» матрицы \mathbf{M} : $\underline{s}_{2,2} = 255$, если $K \geq 255$; $\underline{s}_{2,2} = 0$, если $K \leq 0$; $\underline{s}_{2,2} = K$ в остальных случаях.

2. Аппаратная реализация алгоритмов

Аппаратную реализацию алгоритмов можно выполнить в виде синхронной последовательностной схемы либо асинхронной комбинационной схемы. Комбинационные схемы обеспечивают большее быстродействие, поэтому с помощью системы проектирования фирмы Xilinx Ise 9.2i были разработаны проекты реализации рассматриваемых алгоритмов фильтрации изображений в виде комбинационных схем.

В алгоритме «медиана» основными являются операции сравнения. Поскольку при вычислении значений средних элементов исходных матриц порядок и очередность выполнения операций сравнения не имеют значения, при аппаратной реализации все операции сравнения могут выполняться параллельно, что легко увидеть на структурной схеме (рис. 2).

Обозначения переменных на рис. 2–4 соответствуют обозначениям, используемым в рассматриваемых выше матрицах \mathbf{M} , \mathbf{M}_1 и \mathbf{M}_2 . Показанные блоки comp1–comp4 и median реализуют функции сравнения трех величин по отношению неубывания и их упорядочивания в соответствии с табл. 2. Все блоки аналогичны представленному на рис. 2 блоку comp1, но в отличие от comp1 блоки comp2a–comp4 и median вычисляют только одно значение из трех возможных: блок comp2a – наибольшее значение, comp3 и median – среднее, comp4 – минимальное значение из входных переменных. Результат выполнения алгоритма представлен на выходе блока median в виде вектора **med** разрядностью 8 бит.

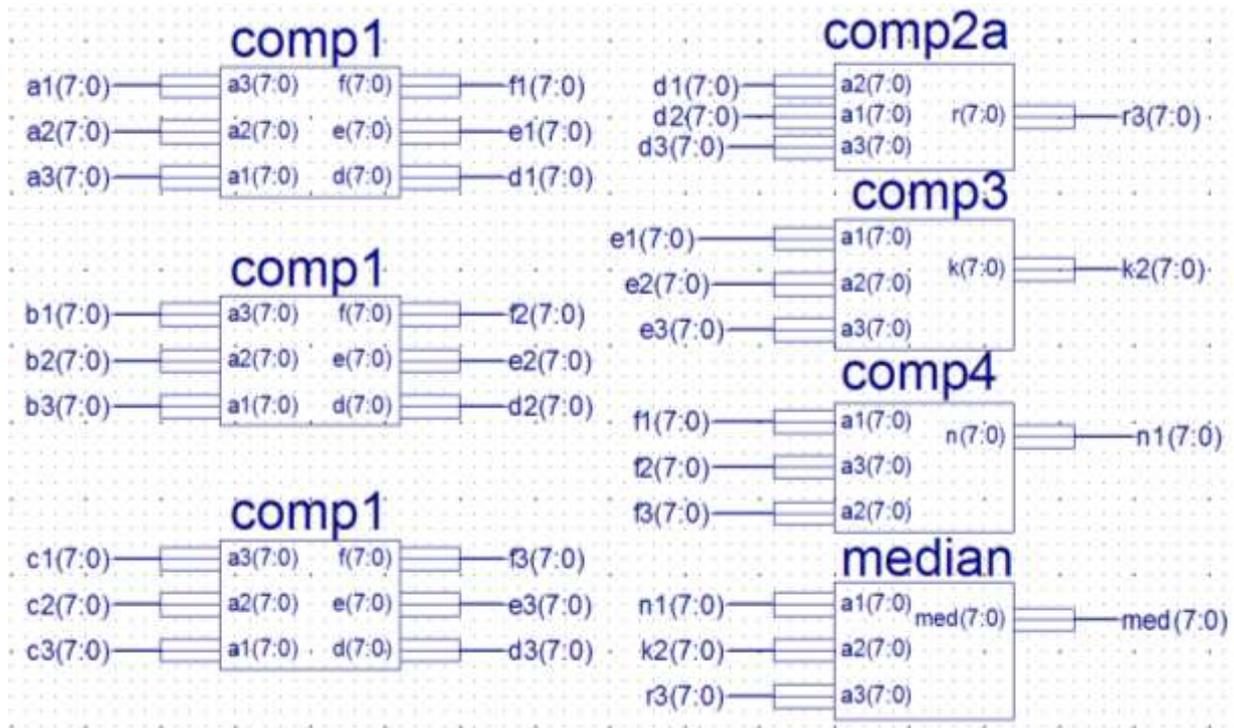
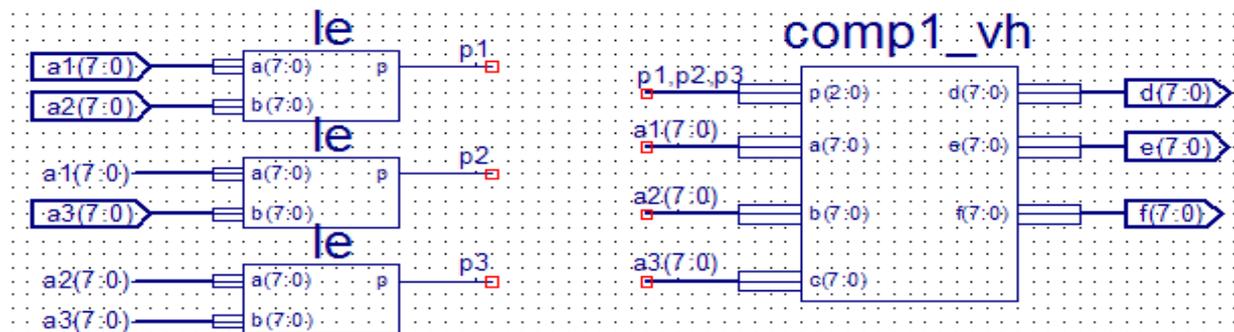


Рис. 2. Комбинационная схема, реализующая алгоритм «медиана»

Рис. 3. Блок comp1, реализующий упорядочивание элементов матрицы \mathbf{M}

Представленный на рис. 3 блок **le** производит сравнение двух величин по отношению убывания и присваивает значение переменной p в соответствии с уравнением 1; блок **comp1_vh** в зависимости от значений p_1, p_2, p_3 в соответствии с табл. 2 определяет порядок следования переменных a_1, a_2, a_3 по отношению убывания значений.

Аппаратная реализация алгоритма «свертка» имеет более сложную структуру (рис. 4), в которой сначала выполняются операции умножения элементов матрицы \mathbf{M} на соответствующие элементы матрицы \mathbf{S} с помощью умножителей **mult**, а затем с помощью четырехступенчатой схемы сумматоров и блока **res** вычисляется величина \mathbf{K} , в зависимости от значения которой в соответствии с уравнением 2 определяется значение переменной **svert**, представляющей собой результат выполнения алгоритма «свертка» для рассматриваемой матрицы.

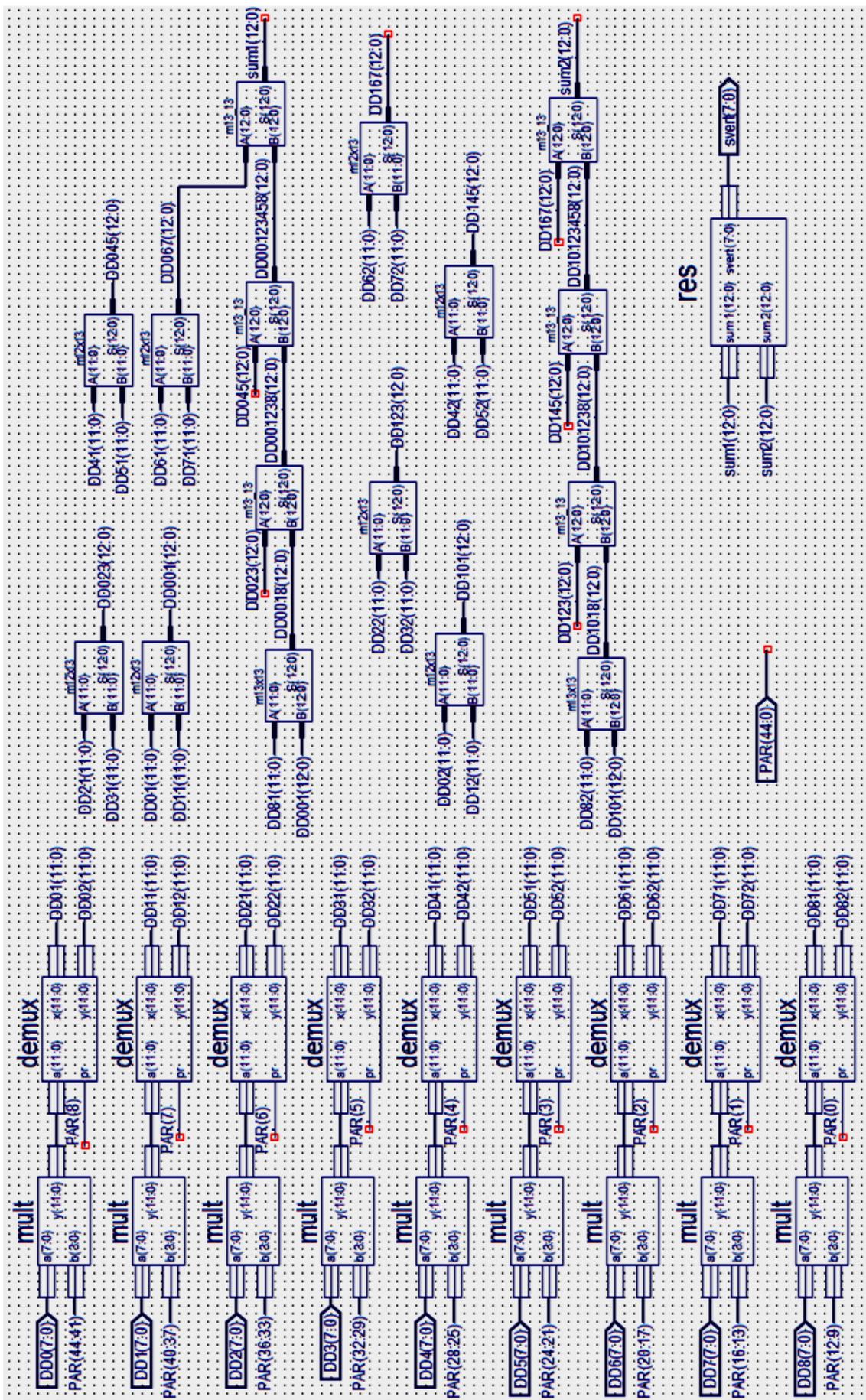


Рис. 4. Комбинационная схема, реализующая алгоритм «свертка»

На рис. 4 используются следующие обозначения: demux – демультимплексоры; m12x12, m12x13, m13x13 – сумматоры; логические переменные PAR(0)– PAR(8) представляют знаки (+ или –) элементов матрицы **S**; PAR(9)–PAR(44) – значения элементов матрицы **S**; блок res вычисляет значение *svert* в зависимости от значений *sum1* (положительные значения) и *sum2* (отрицательные значения) в соответствии с уравнением 2.

3. Объемно-временные характеристики аппаратной реализации алгоритмов

Оценка характеристик аппаратной реализации исследуемых алгоритмов фильтрации изображений производилась с использованием результатов выполнения разработанных проектов в системе проектирования фирмы Xilinx ISE 9.2i (этапы Synthesize, Translate, Map) и логического проектирования с помощью системы моделирования ModelSim 6.2f.

Затраты оборудования на реализацию проекта, предусматривающего обработку одной элементарной матрицы размером 3x3 элемента, т. е. на вычисление значения среднего элемента матрицы, приведены в табл. 3 и 4. Значения затрат оборудования приведены в процентах от ресурса, имеющегося в микросхемах.

Таблица 3

Затраты оборудования на реализацию алгоритмов в микросхемах серии Spartan 3, %

Показатель	Тип микросхемы			
	xc3s200-5ft256		xc3s400-5fg456	
	медиана	свертка	медиана	свертка
Slices	14,5	16,2	7,8	8,7
4 Input LUTs	11,0	12,0	6,0	6,5
Flip Flops	5,5	3,0	3,0	1,6

Таблица 4

Затраты оборудования на реализацию алгоритмов в микросхемах серии Virtex 4

Показатель	Тип микросхемы					
	xc4vlx15-12sf363		xc4vlx25-12sf363		xc4vlx40-12ff668	
	медиана	свертка	медиана	свертка	медиана	свертка
Slices	4,5	7,1	2,6	4,2	1,5	2,4
4 Input LUTs	3,4	5,7	2,5	3,2	1,1	1,9
Flip Flops	1,8	0,9	1,0	0,5	0,6	0,3

Примечание к табл. 3 и 4: LUT (look-up table) – логическая таблица, представляющая собой однобитовое ОЗУ на 16 ячеек; Slice – единица оборудования, состоящая из двух триггеров и двух LUT, Flip-Flop – триггер.

Из табл. 3 и 4 видно, что затраты оборудования на реализацию алгоритмов для обработки одной элементарной матрицы составляют незначительную часть от возможностей микросхем. Это позволяет разработать проекты, предусматривающие загрузку и параллельную обработку матриц большего размера, что существенно в случае фильтрации видеопоследовательностей. Подробнее возможность аппаратной обработки видеопоследовательностей будет рассмотрена ниже.

Поскольку аппаратные реализации алгоритмов «медиана» и «свертка» выполнены в виде комбинационных схем, оценку их временных характеристик можно произвести путем вычисления времени прохождения сигнала от входа до выхода комбинационной схемы, соответствующее времени обработки исходной матрицы **M**. Для этой цели были использованы материалы отчетов о результатах выполнения проектов в системе проектирования Xilinx, а также справочные материалы о характеристиках микросхем серий Spartan 3 и Virtex 4 [5–7]. Результаты представлены в табл. 5.

Таблица 5

Время выполнения алгоритмов (задержка прохождения сигнала через схему), нс

Серия и тип микросхемы	Алгоритм		Максимальная частота, МГц
	«медиана»	«свертка»	
Spartan 3 xc3s200-5ft256 xc3s400-5fg456	32,317 32,317	25,280 25,280	270
Virtex 4 xc4vlx15-12sf363 xc4vlx25-12sf363 xc4vlx40-12ff668	17,407 17,407 17,407	13,493 13,493 13,493	584

Следует учитывать, что ко времени выполнения алгоритма, указанному в табл. 5, нужно добавить еще время одного такта, необходимого для записи результатов обработки элементарной матрицы в выходной регистр.

Используя данные табл. 5, можно определить количество тактов, необходимое для выполнения алгоритма. Например, если выбрать микросхему серии Spartan 3, работающую на частоте 100 МГц (такт = 10 нс), то для обработки одной элементарной матрицы алгоритмом «медиана» потребуются 32,317 нс: $10 \text{ нс} = 4 + 1 = 5$ тактов, а алгоритмом «свертка» – четыре такта (25,280 нс: $10 \text{ нс} = 3 + 1$). Аналогично определяем, что при использовании микросхемы серии Virtex 4, работающей на частоте 100 МГц, для выполнения алгоритмов «медиана» и «свертка» потребуются три такта.

4. Обработка видеопоследовательностей в режиме реального времени

На основании данных табл. 3 и 4 можно рассчитать возможности аппаратной обработки видеопоследовательностей в режиме реального времени. Поскольку затраты оборудования на обработку одной элементарной матрицы составляют небольшую часть объема микросхемы, можно загрузить в микросхему и одновременно обработать некоторую часть (блок) матрицы изображений. Загружая и обрабатывая поочередно блоки матрицы изображений, можно получить результат фильтрации всей матрицы.

Рассмотрим возможность обработки матрицы изображений размером 768 x 576 пикселей (байтов) с помощью алгоритма «медиана». Например, при обработке изображений с помощью микросхемы xc4vlx40-12ff668 можно одновременно загрузить блок матрицы, состоящий из 60 элементарных матриц, т. е. из трех строк длиной 60 элементов каждая. При этом будет задействовано $1,5 \% \times 60 = 90 \%$ оборудования микросхемы. Оставшаяся часть аппаратуры может быть использована для организации операций ввода данных и вывода результатов обработки.

Рассматриваемый блок данных содержит $3 \times 60 = 180$ элементов (байтов). В результате обработки такого блока будут получены новые значения 60 элементов средней строки. Затраты времени на обработку такого блока будут составлять: 180 тактов для ввода данных из ПЭВМ; 60 тактов для вывода результатов обработки; 3 такта для обработки данных; всего 243 такта. Если принять рабочую частоту микросхемы в 100 МГц (Clock, такт = 10 нс), то на обработку блока данных потребуется $243 \text{ такта} \times 10 \text{ нс} = 2430 \text{ нс} = 2,43 \text{ мкс}$.

Для обработки одной строки рассматриваемой матрицы изображений необходимо обработать $768 : 60 = 12,8$, т. е. 13, блоков данных, а для обработки всей матрицы $13 \times 574 = 7462$ блока (из 576 строк матрицы изображений значения первой и последней строк остаются без изменения). Следовательно, обработка матрицы размером 768 x 576 пикселей будет произведена за время, равное $2,43 \text{ мкс} \times 7462 \text{ блока} = 18132,66 \text{ мкс} = 18,2 \text{ мс}$, что значительно меньше 40 мс, через которые происходит смена кадров изображений при обработке видеопоследовательностей.

Заключение

Представленные результаты объемно-временных характеристик аппаратной реализации двух алгоритмов фильтрации изображений позволяют сформулировать ее преимущества перед программной реализацией:

1. Обеспечивается высокая скорость выполнения алгоритмов, позволяющая производить обработку видеопоследовательностей в реальном масштабе времени.
2. Максимальный размер обрабатываемой матрицы изображений определяется только логическими возможностями используемой микросхемы, что позволяет выбирать наиболее приемлемый по технико-экономическим показателям вариант аппаратной реализации алгоритмов.
3. Возможности современных микросхем и приведенные выше характеристики аппаратной реализации рассматриваемых алгоритмов фильтрации изображений позволяют производить одновременно обработку изображения двумя алгоритмами фильтрации в реальном масштабе времени.
4. В случае реализации алгоритма с помощью комбинационной схемы появляется возможность применения более дешевых низкоскоростных микросхем FPGA.

Список литературы

1. Программно-аппаратный комплекс слежения в реальном масштабе времени за движущимися объектами / Б.А. Залесский [и др.] // Информационные технологии программы Союзного государства «Триада». Основные результаты и перспективы. – Минск : ОИПИ НАН Беларуси, 2010. – С. 262–269.
2. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном 3x3 на основе неполных сортирующих сетей / А.И. Кравчонок // Информатика. – 2009. – № 1(21). – С. 91–102.
3. Кравчонок, А.И. Алгоритмы медианной фильтрации с окном 3x3 при помощи MMX- и SSE2-инструкций процессоров семейства x86 / А.И. Кравчонок // Информатика. – 2010. – № 1(25). – С. 41–57.
4. Залесский, Б.А. Отслеживание динамических объектов и их распознавание с помощью графовых алгоритмов / Б.А. Залесский, А.И. Кравчонок // Информатика. – 2006. – № 2 (10). – С. 17–26.
5. Кузелин, М.О. Современные семейства ПЛИС фирмы Xilinx : справочное пособие / М.О. Кузелин, Д.А. Кнышев, В.Ю. Зотов. – М. : Горячая линия – Телеком, 2004. – 440 с.
6. Spartan-3. FPGA Family Data Sheet: DC and Switching Characteristics [Electronic resource]. – Mode of access : <http://www.xilinx.com/support/index.htm>. – Date of access : 23.10.2013.
7. Virtex-4. FPGA Family Data Sheet: DC and Switching Characteristics [Electronic resource]. – Mode of access : www.xilinx.com/support/documentation/data_sheets/ds302.pdf. – Date of access : 23.10.2013.

Поступила 23.10.2013

*Объединенный институт проблем
информатики НАН Беларуси,
Минск, Сурганова, 6
e-mail: alexpolja@tut.by*

A.S. Poljakov, V.E. Samsonov

EVALUATION OF CHARACTERISTICS OF HARDWARE IMPLEMENTATION OF SOME IMAGE FILTERING ALGORITHMS

Processing speed and hardware requirements of two popular image filtering algorithms in hardware implementation based on microchips of FPGA type are provided. A possibility of using algorithms for processing of standard frames of video sequences is considered.