УДК 681.3.06.

И. И. Билинский к. т. н., доц.; В. И. Билинський;

А. О. Мельничук

РЕАЛИЗАЦИЯ ДЕТЕКТОРА ВЫДЕЛЕНИЯ КОНТУРА НА ПЛИС ФИРМЫ XILINX

В статье приводится аппаратная реализация детектора определения контура. Предложен параллельноконвеерный алгоритм предварительной обработки изображения, при котором на первом этапе выполняется параллельно-рекурсивная обработка данных с использованием гауссовых фильтров с разными степенями размытости. На втором – конвеерная обработка данных, которые входят в "скользящее окно". Для реализации такого прибора избрана ПЛИС XC3S1000 семейства Spartan-3 фирмы Xilinx. Синтез схемы выполнялся в САПР Xilinx Project Navigator 7.1, моделирование выполнялось при помощи программы моделирования схем ModelSim SE PLUS 5.1. Такая реализация прибора позволила повысить быстродействие работы детектора в 20 – 25 раз.

Ключевые слова: предварительная обработка изображения, свертка, пиксель, низкочастотный фильтр, програмированая логическая интегральная схема.

Введение. В основе многих процедур обработки и анализа изображения лежит операция линейной обработки "скользящим окном", суть которой состоит в том, что некоторая ограниченная двухмерная область – "окно обработки" – последовательно занимает все возможные положения в плоскости изображения. Для каждого положения окна по значению отсчетов, которые лежат в нем исчисляют значения одного отсчета выходного изображения [1].

Просторанственно-инвариантная оброботка такого вида описывается общим соотношением [2]

$$g(n,m) = G[\{f(n-k,m-l)\},(n,m) \in D],$$
(1)

где f(n,m), g(n,m) – двухмерная последовательность отсчетов входного и выходного изображения соответственно; G – оператор превращения; D – конечное множество отсчетов, которое определяет размерность окна обработки.

Это значит, что значения дискретного сигнала изображения на выходе системы есть результатом цифровой свертки входного дискретного сигнала с конечно-импульсной характеристикой – КИХ-фильтром. Однако такой расчет свертки имеет практическое значение только для малых размерностей окна и небольших изображений, поскольку объем исчислений пропорционален размерности окна и размерности изображения.

Анализ предварительных источников. Известно много алгоритмов быстрой свертки для КИХ-фильтров, но из-за сложности обработки данных не всегда решается задача высокого быстродействия. Другим направлением повышения быстродействия есть аппаратная реализация цифровых фильтров, которая требует использования сложных приборов и больших аппаратных ресурсов. В этом случае наиболее эффективным вариантом аппаратной реализации является использование программированных логических интегральных схем (ПЛИС).

ПЛИС, в последнее время, становятся все более распространенной элементной базой для применения в приборах цифровой обработки сигналов (ЦОС). Благодаря развитой архитектуре, высокой тактовой частоте и невысокой цене ПЛИС незаменимы при макетировании и мелкосерийном производстве.

Существует несколько архитектур цифровых КИХ-фильтров на ПЛИС: параллельная, последовательная и последовательно-паралельная. Цифровые фильтры с параллельной архитектурой имеют максимальную продуктивность и минимальную задержку, но занимают много логических ресурсов микросхемы. Последовательные фильтры имеют наименьшую продуктивность и максимальную задержку, но несмотря на это, они более компактные в сравнении с приборами с параллельной архитектурой [3].

В работах [4, 5] предложено метод выделения утонченного контура на основе нахождения общих точек пересечения фильтрованных изображений вследствие использования низкочастотных гауссовых КИХ-фільтров с разным степенем размытости, который может быть описанный выражением:

$$J(x,y) = p(H^{*}(n,m)|_{\sigma^{1}} - H^{**}(n,m)|_{\sigma^{2}}),$$
(2)

где $H^*(n,m)$, $H^{**}(n,m)$ – изображения, полученные вследствие низкочастотной фильтрации при соответствующих параметрах размытости $\sigma 1$ и $\sigma 2$;

р – коэффициент масштабирования.

Утонченный контур при этом можно получить вследствие поэлементного превращения вида:

$$h(n,m) = \begin{cases} 1, & npu \quad J(n-1), m \end{pmatrix} \cdot J((n),m) < 0 \\ ; \\ 1, & npu \quad J(n-1), m \end{pmatrix} \cdot J((n),m) = 0 \\]_{J((n-1),m) \neq J((nm))}; \\ 0, & npu \quad J(n-1), m \end{pmatrix} \cdot J((n),m) \ge 0. \end{cases}$$
(3)

Поскольку основная трудоемкость такого метода заключается в исчислении арифметической операции свертки, а в данном методе используются две свертки, то время обработки возрастает практически в два раза.

Целью работы есть повышение быстродействия работы детектора определения контура на основании реализации низкочастотного гауссового фильтра на ПЛИС.

Материалы и результаты исследований. В работе предложен паралельно-конвеерный алгоритм предварительной обработки изображения, согласно которому на первом этапе выполняется паралельно-рекурсивная обработка данных при помощи гауссових фильтров с разными степенями размытости. На втором – поэлементные операции вычитания двоих фильтрованных изображений.

Структурная схема такой обработки показана на рис. 1. Такой детектор имеет приборы ввода-вывода, два блока буферной памяти (MS1, MS2), которые обеспечивают бесперебойную запись и считывание данных, два гауссовых фильтра (GF1, GF2), а также прибор поэлементных арифметических операций вычитания (Sub).

На рис. 2 показана структурная схема фильтра, который предполагает выполнение над данными одного пикселя пяти арифметических операций. Такой фильтр состоит из приборов умножения (MD), поэлементного сложения (ADD) и регистров (Rg) для сохранения входных данных пикселя, который обрабатывается, и данных результата обработки.



Рис. 1. Структурная схема детектора выделения контура

Поскольку такая обработка выполняется над данными всех пикселов в соответствии с (1), которые входят в "скользящее окно", то для повышения быстродействия предложено использовать ряд одномерных фильтров, работа которых синхронизирована во времени. При этом гауссовый одномерный фильтр из последовательности отсчетов входного сигнала f(n) размерностью ширины окна в выходную последовательность g(n) и описывается соотношением:

$$g(n) = \sum_{l=-L}^{L^{+}} h(l) f(n-l),$$
(4)

где h(l) – импульсная характеристика фильтра, которая равняется нулю за пределами интервала [$-L^-$, L^+].

Такая схема обработки обеспечивает считывание данных каждого пикселя, который входит в окно обработки, с тактовой частотой и может быть подана в виде суммы значений К пикселов «скользящего окна»

$$h(l) = \sum_{k=0}^{K-1} a_k h_k(l),$$
(5)

$$y(n) = \sum_{k=0}^{K-1} a_k h_k(n),$$
 (6)

$$y_{k}(n) = \sum_{l=-L^{-}}^{L^{+}} h_{k}(l) f(n-l),$$
(7)

где a_k – коэффициенты; $h_k(l)$ – линейно независимые базисные функции, которые разложены в ряд; $y_k(n)$ – сигналы обработки.

Таким образом по каждому сигналу тактовой частоты конвейерно выполняются операции умножения данных пикселов на соответствующие коэффициенты маски, то в каждом цикле умножения происходит сложение результата с предыдущим результатом. Этот процесс продолжается до конца обработки данных всех пикселов, которые входят в "скользящее окно".

Значения интенсивности пикселя как правило есть 24-битным по 8 бит на каждый цвет системы RGB. Это означает, что для фильтрации могут быть использованы три одинаковых одномерных фильтра, работа которых параллельна по времени.

Аппаратная реализация паралельно-конвеерной обработки, которая выполняется детектором выделения контура была реализована на ПЛИС фирмы Xilinx семейства Spartan-3.

Привлекательной чертой ПЛИС Xilinx для реализации алгоритмов ЦОС по сравнению с ПЛИС других производителей есть наличие внутреннего быстродействующего распределенного ОЗУ, который скомпонован в блоки необходимого размера. Использование такого ОЗУ является очень эффективным для реализации алгоритмов ЦОС методом распределенной арифметики, а также для сохранения коэффициентов, результатов промежуточных исчислений и т.д. При этом за счет возможности аппаратного обеспечения параллельности процесса обработки, гибкой адаптации структуры прибора под нужный алгоритм, высокой эффективности интегрирования средств разработки становиться возможным простое построение высокопродуктивной системы ЦОС на одном кристалле в кратчайший термин [6].

Для реализации приведенного выше прибора выбрана ПЛИС XC3S1000 семейства Spartan-3 фирмы Xilinx, основные характеристики которого приведены в таблице.

Таблица

Системные вентили	Логические ячейки	Распределенная память, Кбит	Доступные контакты ввода/вывода	Макс. системная частота, МГц
10^{6}	17 280	$102 \ 10^3$	391	326

Характеристики ПЛИС XC3S1000 семейства Spartan-3

На основании анализа таблицы 1 можно сделать вывод: ПЛИС XC3S1000 семейства Spartan-3 удовлетворяет все параметры для построения детектора.

Описание прибора на уровне регистровых передач выполняется на языке VHDL, для получения файла конфигурации ПЛИС использовался САПР Xilinx Project Navigator 7.1, моделирование работы даного фильтра выполнялось при помощи программы моделирования схем ModelSim SE PLUS 5.1.

На рис. 3 показана принципиальная схема прибора с одним одномерным фильтром, который обеспечивает обработку 8-битовых данных "скользящим окном" размерностью 5×5.

Для выполнения бесперебойной обработки данных используется принцип двухпортовой памяти. После записи данных размерностью 2 Кбайт ОЗУ **mem0** следующие данные записываются в аналогичную память **mem1**. То есть, пока первая порция данных считывается с **mem0** и обрабатывается, другая записывается с порта в **mem1**.

Входные данные записываются по адресам, которые задаться при помощи 12-ти разрядного счетчика count12. Шина адреса записи данных мультиплексируется при помощи мультиплексора mux22X2 с шиной адреса считывания данных. После достижения счетчиком значения 2047 формируется сигнал count(11), который переключает мультиплексор mux22X2 на режим адреса считывания данных, мультиплексор mux_8 на режим считывания данных с памяти mem0 и mem1 и запускает счетчик по модулю 25 count_mod25. Значения count_mod25 поступают на вход постоянно запоминающего прибора (ПЗП) addr_conv, на выходе которого формируется адрес считывания данных по определенному принципу "скользящего окна". После переполнения счетчика count_mod25 формируется сигнал OVF, который инициализирует счетчик по модулю 2022 count2022_mod. Результат счета складывается с результатом на выходе (ПЗП) addr_conv, что позволяет выполнить считывания данных по адресам нового "скользящего окна". Таким образом, адрес считывания данных через мультиплексор попадает на шину адреса O3У. После чего выполняется операция умножения данных в mult на cooтветствующий коэффициент маски, которые поступают с ПЗП coef_conv по результатам адресации счетчика count_mod25.

Каждые 25 тактов результаты умножения складываются между собой, поскольку в этом случае используется маска 5×5. После сложения в **sum_16** выполняется операция деления в **div256** на суммарный коэффициент веса "скользящего окна", результатом которого есть обработанные данные, которые видаются в порт ПК.

По окончанию 25 такта выполняется сброс схемы, сдвиг адресов считывания на один розряд, то есть здвиг «скользящего окна» для обработки следующего пикселя и работа схемы повторяеться.



Рис. 3. Принципиальная схема детектора выделения контура с одним одномерным фильтром с использованием САПР Xilinx Project Navigator 7.1

На рис. 4 показаны часовые диаграммы моделирования работы одномерного фильтра при помощи программы моделирования цифровых схем ModelSim SE PLUS 5.1.

Результаты моделирования засвидетельствовали, что максимальная частота работы троих одномерных фильтров детектора выделения контура, реализованных на одном кристалле ПЛИС XC3S1000, составляет 41 МГц. При этом схема использует весь ресурс кристалла, а для обработки массива данных "скользящим окном" размерностью 5×5 необходимо не больше 30 тактов.

1.																									
2.																									
3. 1		-		1/= -														1/=			_				
5	52 15:	3 1,54	155	356	157	158)59	160	<u>)</u> 61	162	163	164)65	166	<u>167</u>	168	169	170	171	172	173	174	175	176	<u>)77 </u>
5. 5	103417	<u>(8</u>	19	<u></u>	<u></u>	263	(264)265	1266	<u>1267</u>	1519	<u>1520</u>	(521	522	<u>)523</u>	775	<u>1776</u>	<u> 1777</u>	<u> 1778</u>	<u>)779</u>	1031	<u>)1032</u>	<u>(1033</u>	<u> 1034</u>	<u>)1035 [</u>
и. Т	215 IO	(85	_xo	<u>(1</u>	<u>1</u> 2	254	(255	<u>xo</u>	(1	<u>1</u> 2	240	<u>(</u> 241	242	(243	<u>)</u> 244	226	<u>1227</u>	<u>)</u> 228	<u>(</u> 229	<u>)</u> 230	212	213	214	<u>)</u> 215	<u>)</u> 216 [
<i>.</i>	24 IO	<u>(</u> 1)2)3	(4	15)6)7)8)9	(10)11	(12	(13)14	(15)16	<u>)(17</u>	<u>)</u> 18	(19	1 20)21	22	23)24 I
8.	1028,0	<u>)</u> 1)2)3	(4	256	257)258	259)260	512	(513)514	(515	(516	768	(769	(770	<u>(771</u>	(772	1024	(1025	(1026	(1027	(1028 🕻
9.	1	(4	7	(4	<u>(</u> 1	14	(15	24	(15	<u>)</u> 4	7	24	(40	24)7	4	(15	24	(15	(4	1)(4)7	<u>)</u> 4	<u>)</u> 1
10.	215 10	(85	10	χ1	12	254	(255	<u>)</u> 0	<u></u> ί1	12	240	241	(242	(243	244	226	1227	1228	(229	(230	212	213	214	(215	1216
11.	1	14	17	14	Ϊ1	14	<u>)</u> 15	124	<u>)</u> 15	14	17	124	X40	124	17	14	Ĭ15	124	<u>115</u>	14	11	14	X7	14	X1
12.	215 10	1340	ΪΩ	Y4	12	1016	<u>)</u> 3825	ΪΩ	¥15	<u>Й8</u>	1680	15784	Yaean	15832	J1708	1904	<u>)</u> 3405	<u>)</u> 15472	<u>)</u> 3435	<u>.</u> 1920	212	J852	Y1498	<u>)</u> 860	1216 J
13.	215 10	1340	'n	Y4	12	1016	¥3825	Ϋ́n) 15	18	11680	15784	Yasan	Ĭ5832	1708	1904	<u>)</u> 3405	<u>15472</u>	13435	<u>1</u> 920	212	1852	Y1498	1860	1216
14.	856 2	15 YO	1340	-Yo	YA	12	Y1016	13825	Yn	115	18	Y1680	Y5784	Y9680	15832	1708	Y904	Y3405	15472	13435	920	1212	1852	Y1498	1860 T
15.	53 2	15	1555		1559	561	11577	15402	_^	15417	5425	17105	11298	9 122569	128401	130109	131013	Y34418	139890	143325	44245	144457	Y45304	146807	147667
16	52 10	1215	,000	YEEE	7999	1659	YEC1	Y1577	YE402	19411	15417	YE425	Y7105	Y12000	122550	20/01	Y20100	Y21012	Y24410	Y20000	42225	144245	YAAAE	7 Y45200	Y4C007
17	52	1015		YEEE		16EQ	YEC1	11577	YEX02		5417	15425	Y7105	Y10000	100500	20401	100100	121012	104410	100000	10020	144245	YAAAST	1/15000	Y40007
18	0EC 10		1240	<u>,555</u> Vo	Y.	1000	101C	Vacat	<u>,0402</u> Vo	15	10417	10420 V1000	VERON	12003 Vocoo	<u>122363</u>	120401	130103 Yoo4	131013 1340E	<u>134410</u> YE470	<u>733630</u> 73405	143323	<u>794240</u>	Voga	<u>14000</u>	<u>Voco</u>
10.	50 12		1340	<u>,u</u>	<u>14</u>	LC I	1010	13020	10	10	10	1000	10704	13000	10032	11700	101010	13400	10472	<u>10400</u> 110005	1320	1212	1002 V 1500/	<u>1430</u>	1000 I
19. 20	53	10	1555		1009	1061	115//	15402		10417	15425	17105	11288	3 ,22569	<u>,28401</u>	130109	131013	134418	139890	143325 V10005	44245	144457	145305	1 146807	<u>,47667 1</u>
20.	5312	15	1555		<u>1559</u>	1561	<u>,1577</u>	1,5402		<u>,15417</u>	15425	<u>]/105</u>	<u>)</u> 1288	<u>9),22569</u>	<u> </u>	130109	<u> </u>	134418	739890	<u>),43325</u>	144245	<u>)</u> 44457),453US	<u>1 146807</u>	<u>)47667 1</u>
21.	57 15:	3818																							
22.	251 7	5																							

Рис. 4. Результаты моделирования одномерного фильтра при помощи программы моделирования схем ModelSim SE PLUS 5.1: 1) тактовая частота CLK; 2) сигнал разрешения EN; 3) сигнал сброса RST; 4) запись данных в память mem1; 5) адрес считывания данных с памяти mem0; 6) считывание данных с памяті mem0; 7) данные счетчика по модулю 25 count_mod25; 8) адрес скользящего окна addr_conv; 9) выдача коэффициентов матрицы coef_conv; 10) данные множителя; 11) коэффициенты умножения; 12) результат умножения mult; 13) запись результата умножения в регистр FD1; 14) считывание данных из регистра FD1; 15) запись данных в регистр FDR; 16) считывание данных из регистра FDR; 17), 18) выходные данные регистров; 19) результат сложения sum_16; 20) запись результата сложения в регистр FD2; 21) считывания данных из регистра FD2; 22) операция деления div256

Это означает, что для обработки одного пикселя тратится 0,73 мкс. Программная реализация такой же процедуры при использовании, например, компьютера с частотой работы 3 ГГц занимает 3,9 мкс. Это дает возможность сделать вывод о более высоком быстродействии работы прибора, реализованного на ПЛИС, которое возросло при этом в 5,2 раза.

Выводы. Аппаратно реализовано низкочастотный гауссовый фильтр детектора выделения контура на основе ПЛИС XC3S1000 семейства Spartan-3. Такая реализация дает возможность существенно повысить быстродействие и может иметь широкое применение в приборах предварительной обработки изображений больших массивов данных.

СПИСОК ЛИТЕРАТУРЫ

1. Прэтт У. Цифровая обработка изображений: Пер. с англ. – М.: Мир, 1982. – 784 с.

2. Сойфер В.А. Методы компьютерной обработки изображений. – М.: ФИЗМАТЛИТ, 2003. – 784 с.

3. Distributed Arithmetic FIR Filter. – Режим доступу: http:// www.xilinx.com. – Заголовок з екрану.

4. Білинський Й.Й. Математична модель локалізації краю зображення об'єкта // Інформаційні технології та комп'ютерна інженерія. – 2007. – № 3 (10). – С. 73 – 79.

5. Білинський Й.Й. Детектор виділення контуру та простих елементів зображення // Відбір і обробка інформації. – 2007. – № 27 (103). – С. 63 – 67.

6. Зотов В. Инструментальный комплекс Spartan-3 Starter Kit // Компоненты и технологии. – 2005. – №7. – С. 96 – 100.

Билинский Иосиф Иосифович – доцент кафедры проектирования компьютерной и телекоммуникационной аппаратуры;

Билинский Владимир Иосифович – аспирант, ассистент кафедры теоретической электротехники и электроизмерения;

Научные труды ВНТУ, 2008, № 2

Мельничук Андрей Александрович — магистрант кафедры проектирования компьютерной и телекоммуникационной аппаратуры.

Винницкий национальный технический университет