критерии. На их основе для каждого радиуса фильтра было сформировано множество вычислительных конфигураций. Экспериментально было продемонстрировано, что для фильтров с радиусом 1–4 существует две оптимальные конфигурации с размерами блоков данных 512x14 и 512x30 байт соответственно. В качестве обобщения можно добавить, что найденные вычислительные конфигурации будут оптимальны для алгоритмов обработки изображений, которые используют тот же размер окрестности пикселя, что и рассмотренные алгоритмы фильтрации.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. General-Purpose Computation Using Graphics Hardware [Electronic resource] / 2009. – Mode of access: http://www.gpgpu.org. – Date of access: 10.09.2009.
2. Гонсалес, Р. Цифровая Обработка изображений. / Р. Гонсалес, Р. Вудс – Москва: Техносфера, 2005 – 1072 с.
3. NVIDIA CUDA Programming Guide [Electronic resource] / NVIDIA. – 2009. – Mode of access: http://developer.download.nvidia.com/compute/cuda/2_3/docs/NVIDIA_CUDA_Programming_Guide_2.3.pdf . – Date of access: 10.09.2009.
4. Image Convolution with CUDA [Electronic resource] / NVIDIA. – 2009. – Mode of access: http://developer.download.nvidia.com/compute/cuda/1_1/Website/projects/convolutionSeparable/doc/convolutionSeparable.pdf. – Date of access: 10.09.2009.
5. NVIDIA CUDA Best Practices Guide. [Electronic resource] / NVIDIA. – 2009. – Mode of access: http://developer.download.nvidia.com/compute/cuda/2_3/toolkit/docs/NVIDIA_CUDA_BestPracticesGuide_2.3.pdf

Материал поступил в редакцию 11.11.09

UVAROV A.A., SADYKHOV R.Kh. Execution configuration for cuda implementation of image convolution filter

The paper demonstrates approach for constructing optimal execution configuration for CUDA implementation of image convolution filter with different radius. The main goals of execution configuration are to eliminate uncoalesced memory access to improve memory bandwidth and maximize multiprocessor threads utilization. Several mathematical criteria were proposed to choose optimal data block size for execution configuration. Multiple execution configurations based on selected criteria for every filter radius have been formed. It is experimentally find out that two execution configurations with data block size 512x14 and 512x30 are optimal for image convolution filter with radius from 1 to 4. Also described execution configurations are optimal for image processing algorithms which use the same pixel area like considered image convolution filter.

УДК 621.391.826.4

Новиков А.Е., Петровский А.А.

VHDL-РЕАЛИЗАЦИЯ АДАПТИВНОГО КИХ-ФИЛЬТРА НА РАСПРЕДЕЛЕННОЙ АРИФМЕТИКЕ

Введение. КИХ-фильтр генерирует выходной сигнал $y(n)$, равный сумме задержанных входных отсчетов $x(m)$, умноженных на соответствующие коэффициенты:

$$y(m) = \sum_{i=0}^{K-1} w_i x(m-i), \qquad (1)$$

где K – порядок фильтра;

w_i – i-тый коэффициент фильтра.

В обычной реализации фильтр потребует K операций умножения с накоплением (multiply and accumulate - MAC). Для систем, где максимальная тактовая частота ограничена по соображениям энергосбережения, пропускная способность КИХ-фильтра, определяемая как число отфильтрованных отсчетов в секунду, будет сильно ограничена. Это ограничение может стать серьезным при больших порядках фильтров (большое K). Несмотря на то, что применение нескольких процессоров цифровой обработки сигналов (ЦОС процессоров) может повысить пропускную способность, сопутствующее усложнение устройства, увеличение занимаемой площади и потребляемой энергии может сделать такую реализацию устройства малопривлекательной.

Альтернативой может стать реализация MAC операций как серии таблиц с заранее рассчитанными данными, операций доступа к данным и суммирования. Для этого операцию фильтрации (1) нужно представить в последовательном виде. Такая реализация фильтра, известная как распределенная арифметика (РА), позволит поднять пропускную способность за счет более быстрых вычислений, уменьшить сложность устройства ценой повышения требований к объему памяти. Последующие усовершенствования РА позволяют уменьшить объем необходимой памяти, что делает реализацию цифровых КИХ-фильтров на распределенной арифметике более привлекательной.

Одни из первых работ применения распределенной арифметики для задач цифровой обработки сигналов это [1, 2]. Всесторонний обзор цифровых фильтров на РА дан в [3]. В данной работе описана VHDL-реализация адаптивного КИХ-фильтра на распределенной арифметике, который может применяться в системах компенсации электрического эха. Электрическое эхо возникает в гибридной системе при стыковке двухпроводной и четырехпроводной линий связи из-за невозможности согласования импедансов. Результатом является ухудшение качества связи. Наблюдается явление, когда абонент вместе с голосом собеседника слышит собственный задержанный голос. Как следует из стандарта G.168 [4], "существенная часть" импульсного отклика гибридной схемы, которая образует эхо-сигнал, может достигать 0,016 с. При частоте дискретизации 8 кГц, которая используется при обработке сигналов в каналах связи тоновой частоты, длительность такого импульсного отклика будет равна 128 выборкам. Это означает, что адаптивный фильтр должен иметь приблизительно такую же "длину", чтобы компенсировать эхо-сигналы. Согласно [5], при длительности эхо в 0,016 с для обеспечения комфортного разговора уровень эха должен быть ниже уровня голоса минимум на 10 дБ, а при длительности эха в 0,1 с – уже 31 дБ.

Адаптивный КИХ-фильтр. Имеются два абонента – дальний (A) и ближний ($Б$). В случае, если абонент $Б$ говорит одновременно с абонентом A (рис. 1), сигнал от абонента $Б$ равен

$$y_B(m) = x_B(m) + x_A^{echo}(m) \qquad (2)$$

где $x_B(m)$ – сигнал от абонента $Б$;

$x_A^{echo}(m)$ – эхо-сигнал абонента A.

Если говорит только абонент A, то сигнал, приходящий со стороны абонента $Б$, будет равен:

$$u_B(m) = x_A^{echo}(m). \qquad (3)$$

***Новиков Алексей Евгеньевич**, ассистент Белорусского государственного университета информатики и радиоэлектроники.*

***Петровский Александр Александрович,** д.т.н, профессор, зав. кафедрой ЭВС Белорусского государственного университета информатики и радиоэлектроники.*

Беларусь, БГУиР, 220013, г. Минск, ул. П. Бровки, 6.

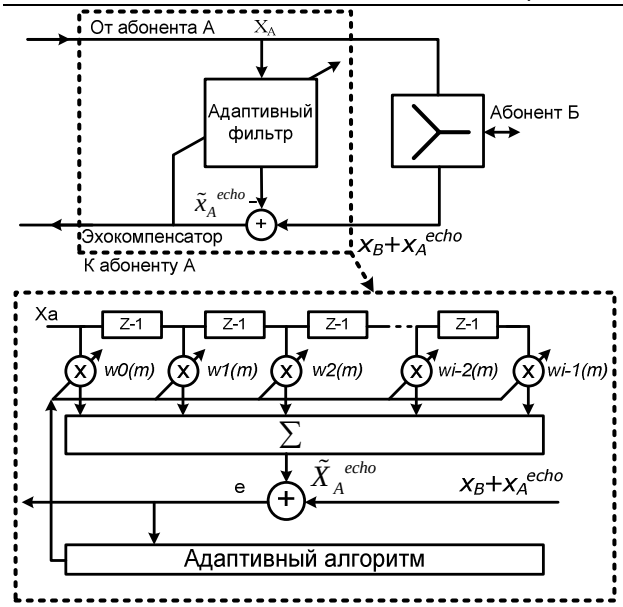

Рис. 1. Адаптивный КИХ-фильтр в составе эхокомпенсатора

На качество работы адаптивных фильтров обычно влияют несколько факторов. Одним из таких факторов является шум канала $n(k)$. В таком случае формула (2) примет вид:

$$u_B(m) = x_B(m) + x_A^{echo}(m) + n(k), \quad (4)$$

где $n(k)$ – шум канала.

Этот шум состоит из теплового шума и шума, обусловленного передачей сигналов по проводам витых пар, расположенных в одном кабеле с витой парой, по которой осуществляется связь с рассматриваемым ближним абонентом. Обычно этот шум ограничен по уровню и имеет известное распределение плотности в полосе используемых частот. Типичное значение уровня шума равно – 30 дБ относительно среднего уровня эхосигналов. Такой шум практически не влияет на процесс адаптации, поэтому при моделировании его учитывать не будем.

Задача адаптивного фильтра – подстроиться под импульсную характеристику гибрида и, пропустив через себя сигнал от дальнего абонента, опорный сигнал, смоделировать эхо, возникающее из-за несогласованности линий.

Адаптивный фильтр имеет 2 входа и 1 выход. На один из входов подается сигнал, называемый опорным (сигнал x_A). Результат фильтрации – смоделированное эхо – поступает на выход фильтра.

$$\tilde{x}_A^{echo}(m) = \sum_{i=0}^{K-1} w_i(m) x_A(m-i), \quad (5)$$

где $w_i(m)$ – коэффициенты адаптивного фильтра;

$\tilde{x}_A^{echo}(m)$ – смоделированное эхо.

Далее происходит вычитание смоделированного эха из сигнала, пришедшего от ближнего абонента (в данном случае это абонент ***Б***). Разница (чем она меньше, тем лучше при молчащем ближнем абоненте) поступает на второй вход адаптивного фильтра.

$$e(m) = x_A^{echo}(m) - \tilde{x}_A^{echo}(m) = \\ = x_A^{echo}(m) - \sum_{i=0}^{K-1} w_k(m) x_A(m-i), \quad (6)$$

где $e(m)$ – разница между «эхо абонента ***А***» и смоделированным эхо (абонент ***Б*** молчит);

K – порядок КИХ-фильтра.

Коэффициенты адаптивного фильтра $w_i(m)$ вычисляются таким образом, чтобы энергия разностного сигнала была минимальной. Это достигается в том случае, когда сигнал $\tilde{x}_A^{echo}(m)$ эквивалентен сигналу $x_A^{echo}(m)$.

Для алгоритма МНК обновление коэффициентов производится по формуле:

$$w(m) = w(m-1) + \mu e(m) x(m), \quad (7)$$

где $x_A(m) = [x_A(m),...,x_A(m-K)]$ – вектор отсчетов входного сигнала;

$w(m) = [w_0(m),...,w_{K-1}(m)]$ – вектор коэффициентов фильтра;

$e(m)$ - разница между настоящим и смоделированным эхо;

μ - шаг адаптации ($0<\mu<1$), определяет скорость сходимости и устойчивость алгоритма.

Вычисления при помощи распределенной арифметики. В РА применяются побитовые последовательные операции. Представление MAC-операций посредством побитовых последовательных операций достигается следующим способом [3]. Отсчеты сигнала, которые должны быть отфильтрованы, представим как B-разрядные двоичные числа в дополнительном коде:

$$x(m-i) = -b_{i0} + \sum_{i=1}^{B-1} b_{il} 2^{-l}, \quad (8)$$

где b_{il} – это l бит двоичного дополнительного кода $x(m\text{-}i)$. Подставляя (8) в (1) и меняя порядок суммирования, получим:

$$y(m) = -\left[\sum_{i=0}^{K-1} b_{i0} w_i\right] + \sum_{l=1}^{B-1}\left[\sum_{i=0}^{K-1} b_{il} w_i\right] 2^{-l}. \quad (9)$$

Для заданного набора w_i (i=0,…,K-1) члены в квадратных скобках могу принять только одно из 2^K возможных значений, которые могут быть сохранены в памяти, которую назовем фильтрующей таблицей и обозначим буквой F. Нужный адрес задается следующим образом:

$$r = \sum_{i=0}^{K-1} c_i^{(r)} 2^i, \quad (10)$$

где $c_i^{(r)}$ – это i-й бит в K-битном представлении адреса r. Содержимое фильтрующей таблицы определяется так:

$$F_{(r)} = \sum_{i=0}^{K-1} c_i^{(r)} w_i, \; r = 0,...,2^{K-1}. \quad (11)$$

КИХ-фильтр на РА представлен на рис. 2. Фильтрующая таблица содержит 2^K возможных комбинаций сумм коэффициентов фильтра. Банк сдвигающих регистров хранит K последовательных входных отсчетов ($x(m\text{-}i)$, i=0,…,K-1). Последовательность крайних правых битов является адресом для фильтрующей таблицы F. Сдвигающие регистры производят сдвиг данных вправо каждый такт. Соответствующие значения F накапливаются B раз, в результате получается выходной отсчет. Бит управления знаком используется для замены сложения вычитанием для содержимого F, адрес которого сформирован знаковыми разрядам, заключенными в первые квадратные скобки выражения (9).

Операция фильтрации при помощи РА выполняется за B шагов – независимо от K. Для малых порядков фильтров прирост в пропускной способности не является значительным, тогда как использование РА при больших порядках фильтров (K>>B) дает существенный прирост пропускной способности. Еще одним преимуществом фильтров на РА является отсутствие необходимости в использовании аппаратных умножителей.

С увеличением порядка фильтра экспоненциально возрастает и объем требуемой памяти. Например, для КИХ-фильтра 128 порядка потребуется память на 2^{128} ячеек. Эта проблема может быть решена путем разбиения фильтра на несколько меньших фильтров и суммирования их выходов.

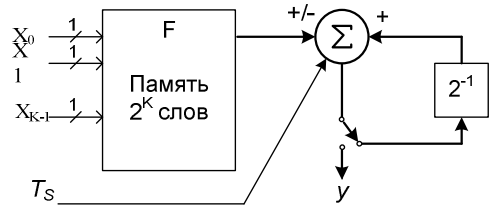

Рис. 2. Реализация вычисления на распределенной арифметике

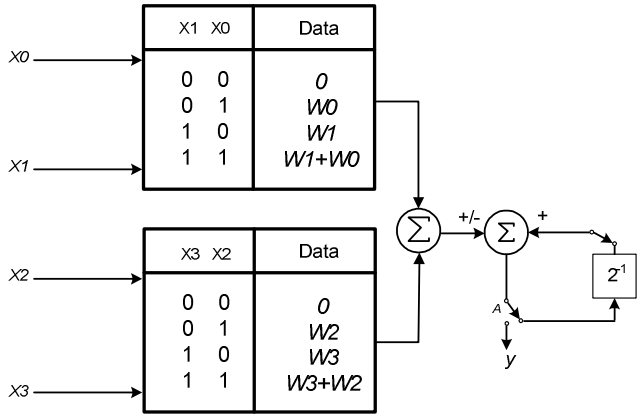

Рис. 3. КИХ-фильтр 4 порядка с 2 таблицами

Суммирование в квадратных скобках в выражении (9) может быть разбито так, что фильтр порядка K может быть разделен на n меньших фильтров, порядок каждого равен k ($K = n \times k$). Теперь общий порядок фильтра K не имеет решающего значения для латентности фильтра. Выражение (9) может быть записано как

$$y(m) = -\left(\sum_{j=0}^{n-1}\left[\sum_{i=jk}^{(j+1)k-1} b_{i0}w_i\right]\right) + 6\sum_{l=1}^{B-1}\left(\sum_{j=0}^{n-1}\left[\sum_{i=0}^{(j+1)k-1} b_{il}w_i\right]\right)2^{-l}. \quad (12)$$

Пример фильтра 4 порядка с 2 таблицами показан на рисунке 3.

Для размещения всех элементов в памяти фильтра, построенного в соответствии с (12), понадобится $n \times 2^k$ ячеек. Общее количество тактов между загрузкой новых отсчетов в стек в такой реализации составит $B + [\log_2(n)]$. Второе слагаемое в этой сумме обусловлено необходимостью установки дерева сумматоров для получения суммарного сигнала с выходов меньших фильтров. Например, если K = 128, то в обычной реализации понадобится 2^{128} ячеек, а, выбрав n = 32 и k = 4, снизим требования к памяти до 512 элементов. Количество тактов при использовании 16 разрядных данных в таком случае составит 21 такт, а при прямой реализации с одной фильтрующей таблицей – 16 тактов.

Поскольку в схемах, приведенных на рисунках 2 и 3, поступление данных происходит по 1 биту за 1 такт, то такие схемы называются 1-BAAT (one-bit-at-a-time). Аналогично при одновременном поступлении подаче L битов схему можно обозначить как L-BAAT. Однако это увеличение быстродействия повлечет за собой увеличение требуемого объема памяти.

Адаптивный КИХ-фильтр на распределенной арифметике. В фильтре применяется адаптация коэффициентов по алгоритму МНК, однако обновляется не каждый коэффициент в отдельности, а целиком содержимое ячеек фильтрующей таблицы согласно выражению (13), которое получено на основании (7), (9) и (13).

$$\sum_{i=0}^{K-1} c_i^{(r)} w_i(m+1) = \sum_{i=0}^{K-1} c_i^{(r)} w_i(m) + \mu e(m) \sum_{i=0}^{K-1} c_i^{(r)} x(m-i) \quad (13)$$

На рис. 4 приведена общая схема адаптивного КИХ-фильтра на распределенной арифметике.

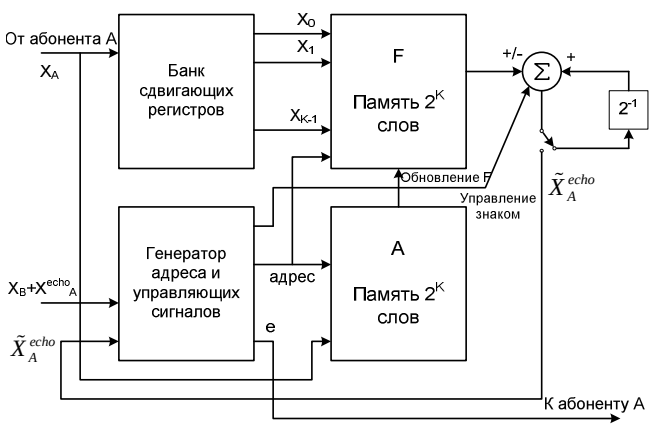

Рис. 4. Общая схема адаптивного КИХ-фильтра на распределенной арифметике

Обновление ячеек фильтрующей таблицы целиком, а не отдельных коэффициентов, требует наличия заранее подготовленных входных данных, которые хранятся во вспомогательной таблице A, организованной таким же образом, как и таблица F. Таким образом, адаптивный КИХ-фильтр на основе РА отличается от обычного КИХ-фильтра на РА удвоенными требованиями к объему необходимой памяти.

Для работы адаптивного фильтра МНК с архитектурой на основе РА, значения фильтрующей таблицы, содержащие все возможные комбинации сумм коэффициентов фильтра, должны быть пересчитаны и обновлены за время обработки одного отсчета. Обновление каждого коэффициента индивидуально в соответствии с формулой (7), а потом запись в F новых коэффициентов, является вычислительно сложной задачей и займет много времени, вследствие чего снизится пропускная способность фильтра. В [6] предложена структура, требующая меньшее число циклов, чем применение расчет и запись каждого коэффициента по отдельности.

На рисунке 5 показан упрощенный алгоритм работы адаптивного КИХ-фильтра. Из рисунка видно, что операция фильтрации происходит параллельно с обновлением вспомогательной таблицы.

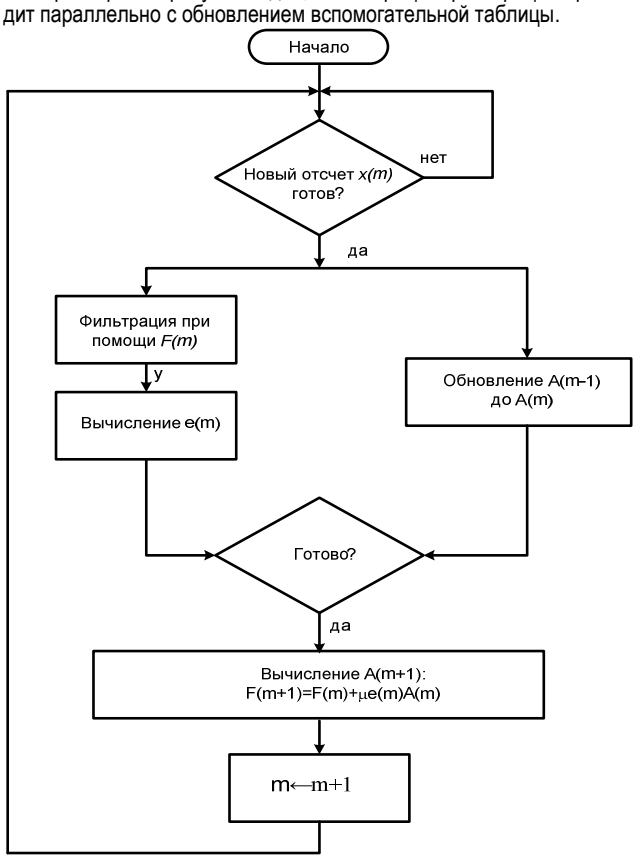

Рис. 5. Упрощенный алгоритм работы адаптивного КИХ-фильтра на распределенной арифметике

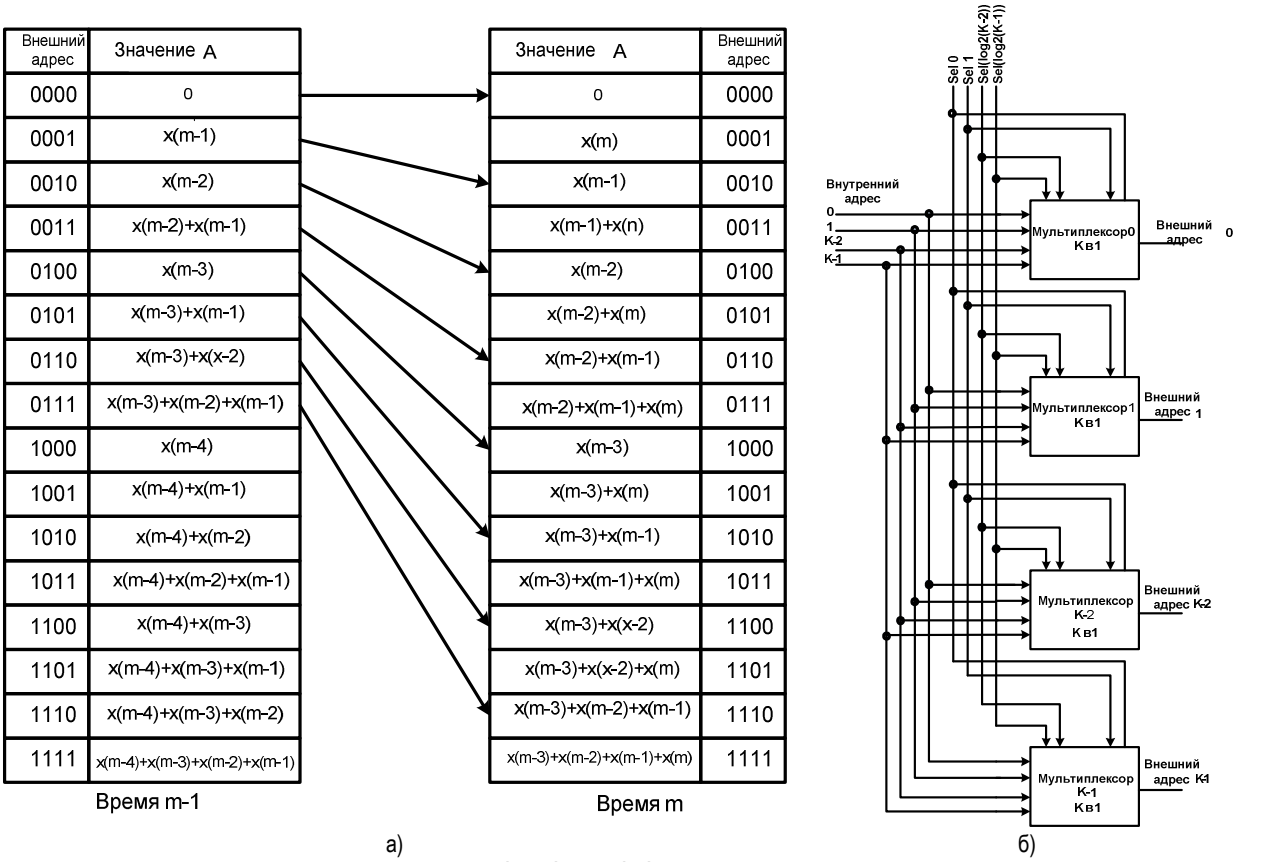

Рис. 6. Обновление ячеек памяти $A(m\text{-}1)$ до $A(m)$ (а) и схема циклического сдвига адресов (б)

Ниже приведено описание и назначение составных модулей фильтра.

1) Модуль фильтра РА выполняет операцию фильтрации входных данных с текущими значениями коэффициентов в фильтрующей таблице F.

2) Вспомогательная таблица A содержит все возможные комбинации сумм K последних входных отсчетов. Следовательно, содержимое r-той ячейки A – это слагаемое $\sum_{i=0}^{K-1} c_i^{(r)} x(m-i)$ из выражения (13). Размер таблицы и структура идентичны F, содержащейся в модуле фильтра.

3) Управляющий модуль генерирует адрес и управляющие сигналы для обновления A и F.

Один шаг фильтрации и адаптации предложенного алгоритма в момент m показан на рисунке 5. Обозначения $A(m)$ и $F(m)$ использованы для описания таблиц в момент m. Операция фильтрации (9) над $x(m)$, $x(m\text{-}1)$,…, $x(m\text{-}K\text{+}1)$ производится используя $F(m)$. Пока происходит фильтрация, $A(m\text{-}1)$ обновляется до $A(m)$. Когда фильтрация и обновление A завершены, $F(m)$ обновляется до $F(m\text{+}1)$. Когда обновление до $F(m\text{+}1)$ завершится, шаг адаптации во времени m закончится. Фильтр ожидает прихода нового отсчета $x(m\text{+}1)$, и алгоритм (рис. 4) повторяется. Обновление A и F по прибытии $x(m)$ более детально описывается далее.

На рис. 6 показано обновление $A(m\text{-}1)$ до $A(m)$ для порядка фильтра $K = 4$. Можно заметить, что содержимое по четным адресам (заканчивающимся на 0) $A(m)$ содержит младшую половину (адреса ячеек начинается с 0) $A(m\text{-}1)$. Также можно заметить, что содержимое ячеек по нечетным адресам (адрес заканчивается на 1) $A(m)$ может быть получено из содержимого по четным адресам $A(m)$ в соответствии с уравнением

$$A_{(2l+1)}(m) = A_{(2l)}(m) + x(m), \qquad (14)$$

$$l = 0,\ldots,2^{K-1}-1.$$

Обновление $A(m\text{-}1)$ до $A(m)$ происходит в 2 этапа.

Этап 1. Младшая половина $A(m\text{-}1)$ отображается по четным адресам в $A(m)$, как показано стрелками на рисунке 6а. Вместо физического перемещение данных, это отображение выполняется путем циклического сдвига влево K адресных линий A. Циклический сдвиг адреса позволяет содержимому памяти оставаться таким же, но внешняя логика видит таблицу вновь отображенной. Циклический сдвиг адреса может быть организован в соответствии со схемой на рисунке 6б. Термин «внутренний адрес» обозначает физический адрес, а термин «внешний адрес» обозначает адрес, видимый внешней логикой.

Можно заметить, что внешний адрес, ссылаясь на данный внутренний адрес в момент m, есть внешний адрес, циклически сдвинутый влево, ссылавшийся на тот же внутренний адрес в момент $m\text{-}1$. Таким образом, эффект циклического сдвига адреса может быть достигнут путем включения K мультиплексоров K в 1. $Log2(K)$ адресных линий K мультиплексоров подключены к $Log2(K)$ битам счетчика, который увеличивает свое значение с каждым тактом синхронизации. Так, циклический сдвиг адреса, переотображение A, может быть завершено мгновенно к приходу нового отсчета $x(m)$.

Этап 2. Нужно заметить, что циклический сдвиг адреса отображает верхнюю половину $A(m\text{-}1)$, содержащую суммы, включая старший отсчет $x(m\text{-}4)$, по нечетным адресам в момент m. Содержимое по этим нечетным адресам $A(m)$ перезаписывается значениями в соответствии с уравнением (14). Другими словами, содержимое по нечетным адресам $A(m)$ получается путем чтения соответствующих предшествующих четных позиций и добавления нового отсчета $x(m)$, после чего результат сохраняется обратно по нечетным адресам.

Когда обновление $A(m)$ будет завершено, так же как и фильтрация, производится обновление $F(m\text{+}1)$. Из (13) следует, что обновление r-той ячейки F от m до $m\text{+}1$ выполняется в соответствии с уравнением

$$F_{(r)}(m+1) = F_{(r)}(m) + \mu e(m) A_{(r)}(m). \qquad (15)$$

$F(m+1)$ обновляется путем чтения соответствующих ячеек (таких же) в обоих $F(m)$ и $A(m)$, умножения выхода $A(m)$ на $\mu e(m)$, добавления этого произведения к выходу $F(m)$, и сохранения результата по тем же адресам в $F(m+1)$. Этот процесс повторяется для адресов с 1 по 2^{k-1}. Значение по адресу 0 не обновляется, потому что там всегда находится 0. Адреса и управляющие сигналы для обновления F генерируются управляющим модулем.

Операция умножения содержимого $F(m)$ на $\mu e(m)$ может быть реализована на обычном аппаратном умножителе. Однако в целях экономии места на кристалле умножение может быть заменено сдвигом. При этом $\mu e(m)$ квантуется на N уровней, каждый из которых кратен степени 2, и значение $F(m)$ сдвигается вправо на требуемое число разрядов.

Выше был описан принцип построения адаптивного КИХ-фильтра на распределенной арифметике для одной пары таблиц (фильтрующей и вспомогательной). Для большего количества пар таблиц принцип построения фильтра остается таким же, как и для обычного КИХ-фильтра.

Полученные результаты

Пропускная способность и требования к объему памяти. Пропускная способность определяется как количество отсчетов входного сигнала, обработанных адаптивным фильтром за 1 секунду. Если количество тактов, необходимых для обработки 1 отсчета обозначить через N, то пропускная способность фильтра (Tp) будет определяться как

$$Tp = \frac{f_t}{N}, \qquad (16)$$

где f_t – частота тактовых импульсов.

В фильтре, содержащем n таблиц порядка k, обновление A может быть выполнено за 2^{k-1} тактов. Обновление происходит параллельно с операцией фильтрации, которая будет продолжаться B тактов. Для обновления и фильтрации, таким образом, понадобится $\max(B, 2^{k-1})$ тактов. Обновленная A используется для вычисления значений F за 2^k тактов. И дерево сумматоров потребует $\log_2(n)$ тактов. Отсюда следует, что адаптивному фильтру понадобится $2^k + \max(B, 2^{k-1}) + [\log_2(n)]$ тактов. Другими словами, пропускная способность фильтра определяется как

$$Tp = \frac{f_t}{2^k + \max(B, 2^{k-1}) + [\log_2(n)]}. \qquad (17)$$

Пропускная способность адаптивного КИХ-фильтра на распределенной арифметике для различных порядков фильтра и количества пар таблиц показана на рис. 7. Для оценки пропускной способности была выбрана частота ТИ равная 50 МГц.

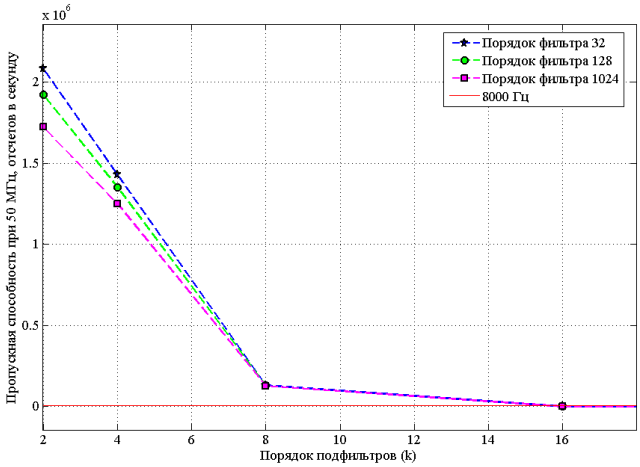

Рис. 7. Пропускная способность фильтра

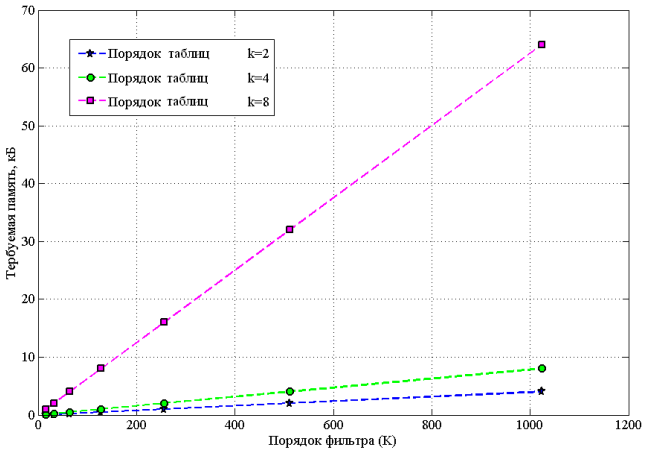

Рис. 8. Требуемая память

Из графика 7 видно, что пропускная способность почти не зависит от порядка фильтра, а зависит от количества и порядка таблиц. Горизонтальная непрерывная линия на рис. 7 показывает тот предел, после пересечения которого сигнал с частотой дискретизации 8 кГц не будет успевать обрабатываться.

Объем требуемой памяти рассчитывается из соотношения

$$M = 2 \times n \times 2^k, \qquad (18)$$

где n – количество пар таблиц;

k – порядок каждой таблицы.

На рис. 8 показана зависимость объема требуемой памяти от порядка фильтра и размера фильтрующих таблиц. Из графика и соотношения (18) видно, что наибольшее влияние на требования к памяти оказывает размер фильтрующих таблиц.

Количество логических элементов и потребляемая мощность. Для цифровых систем, реализуемых на микросхемах FPGA, важным показателем является количество занимаемых логических элементов. В микросхемах Spartan 3, построенных по матричному принципу, под логическим элементом понимают совокупность триггера и 4-входовой LUT, с помощью которых реализуются логические функции.

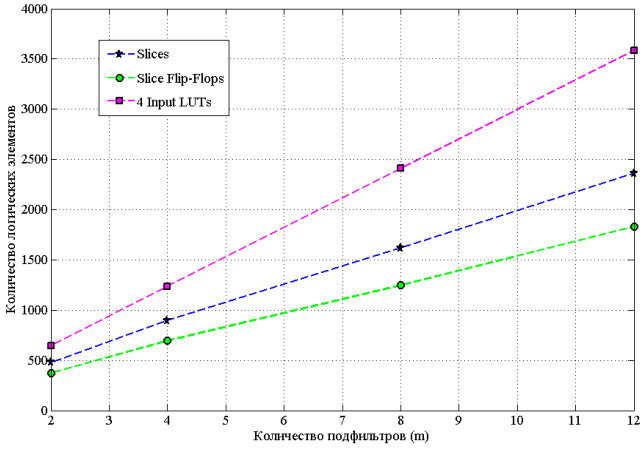

Рис. 9. Зависимость количества логических элементов от порядка фильтра и количества таблиц

В мобильных устройствах важным показателем является потребляемая мощность. Для расчета потребляемой мощности была использована программа XPower из пакета Xilinx ISE.

Для оценки зависимости количества требуемых логических элементов от порядка фильтра и количества таблиц, а также оценки потребляемой мощности, были описаны и синтезированы следующие фильтры:

а) $K = 8$, $n = 2$, $k = 4$, $B = 16$;

б) $K = 16$, $n = 4$, $k = 4$, $B = 16$;

в) $K = 32$, $n = 8$, $k = 4$, $B = 16$;

г) $K = 48$, $n = 12$, $k = 4$, $B = 16$.

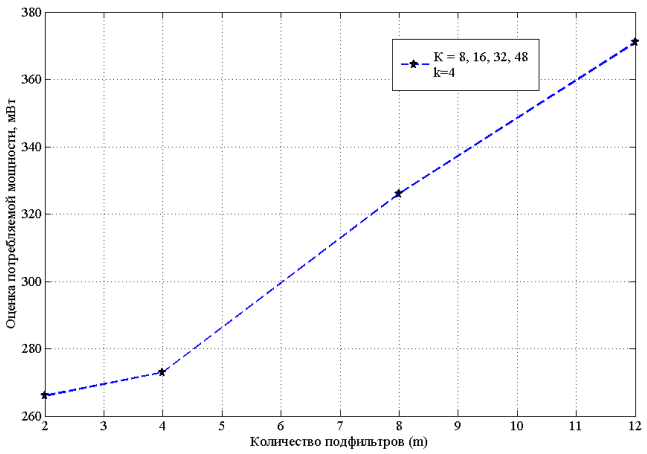

Рис. 10. Зависимость потребляемой мощности от порядка фильтра и количества таблиц

Оценка качества компенсации эха. Как упоминалось выше, для оценки возвращенного эха используется оценка ERLE (Echo Return Loss Enhancement) – данный параметр представляет собой отношение энергий неподавленного и остаточного эхосигналов.

$$ERLE = 10\log_{10}\frac{E\{x_A^2(m)\}}{E\{e^2(m)\}}. \qquad (19)$$

Достижимое значение ERLE может быть оценено путем непосредственного измерения с помощью использования уравнения (19) на скользящем окне длиной Lw отсчетов обрабатываемых сигналов. Эта длина определяется интервалом стационарности сигнала. Например, интервал стационарности сигналов речи примерно равен 30 мс. При частоте дискретизации 8 кГц получим, что Lw = 0.03 × 8000 = 240 отсчетов. В качестве модели был выбран фильтр 32 порядка, предназначенный для компенсации эха с задержкой до 4 мс.

Заключение. На основании метода проектирования адаптивных КИХ-фильтров с применением распределенной арифметики [6] создано VHDL-описание фильтра. Особенностями такой архитектуры устройства фильтра является высокая ее пропускная способность, что позволяет осуществлять фильтрацию в широком частотном диапазоне, невысокие требования к памяти, а также масштабируемость структуры фильтра.

Осуществлена аппаратная реализация фильтра на FPGA. Преимуществом такой аппаратной платформы является возможность динамического выбора и загрузки архитектуры фильтра, гибкость, высокая пропускная способность благодаря параллельным вычислениям в FPGA.

СПИСОК ЦИТИРОВАННЫХ ИСТОЧНИКОВ

1. A. Croisier, D. J. Esteban, M. E. Levilion and V. Rizo, “Digital Filter for PCM Encoded Signals”, U.S. Patent 3 777 130, Apr. 1973
2. A. Peled and B. Lie, IEEE Trans. Acoustics, Sound, Signal Process., vol. ASSP-22, no 4, pp. 456-462, Dec. 1974.
3. S. A. White, , IEEE ASSP Mag., vol. 6, pp. 4-19, Jul.1989.
4. Digital network echo cancellers. ITU-T Recommendation G.168. Series G: Transmission systems and media, digital systems and networks. International telephone connections and circuits - Apparatus associated with long-distance telephone circuits. 04/2000. Geneva. 2001. – 116 p.
5. Huntly H.R. Bell System Technical Journal. Vol. 32. Sept. 1953. – P. 1019-1036.
6. Daniel J. Allred, Heejong Yoo, Venkatesh Kristman, Walter Huang, David V. Anderson. IEEE Transactions on Circuits and Systems. – Vol. 52, NO.

Материал поступил в редакцию 20.02.09

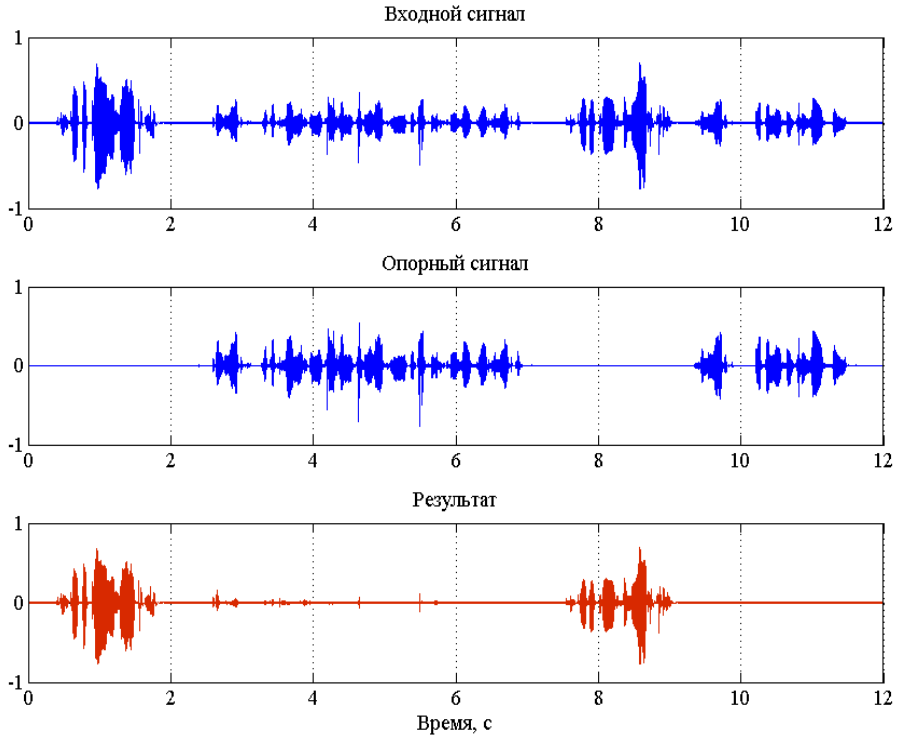

Рис. 11. Входной сигнал, опорный сигнал и результат обработки

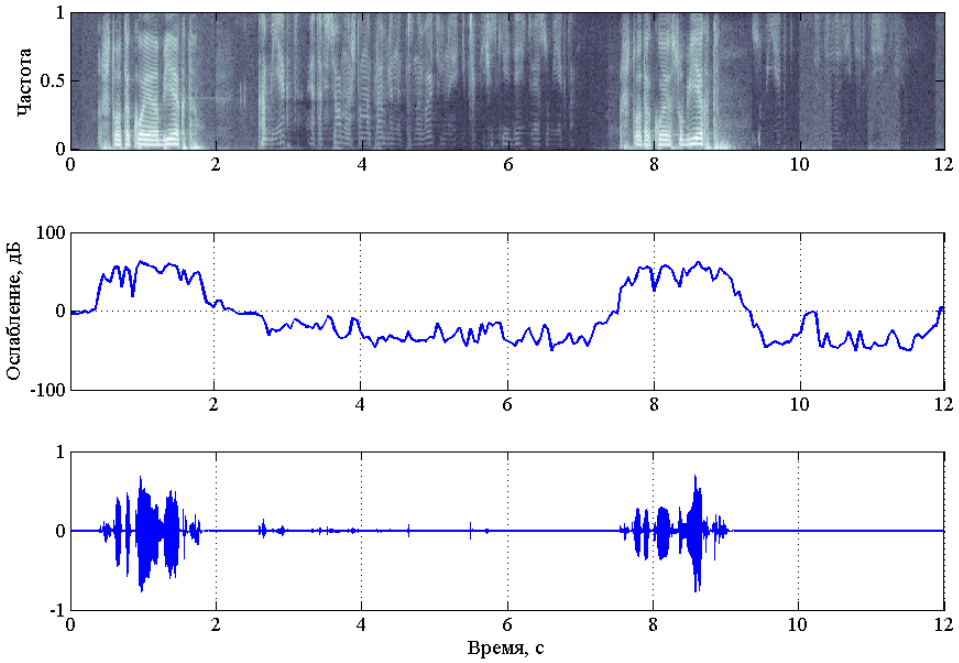

Рис. 12. Спектрограмма обработанного сигнала, график оценки ERLE и результат

NOVIKOV A.E., PETROVSKIY A.A. VHDL Realization of the adaptive FIR-filter on distributed arithmetics

The paper describes a synthesis approach of the FIR-filters based on the distributed arithmetics and the least-squares coefficients adaptation. A device is implemented on the Xilinx Spartan 3 FPGA and can be used in such applications as electrical echo-cancellation in telecommunications, etc. A discussion of the achieved results is present.

УДК 004.822

Ивашенко В.П.

АЛГОРИТМЫ ВЕРИФИКАЦИИ И ИНТЕГРАЦИИ БАЗ ЗНАНИЙ

Введение. Для широкого применения интеллектуальных систем, способных повысить качество решения прикладных задач, необходимо большое количество баз знаний. Быстрой разработке достаточного количества баз знаний могло бы способствовать наличие средств разработки интеллектуальных систем, обеспечивающих разработку и проектирование различных компонентов интеллектуальной системы, включая базу знаний. Среди средств, которые могут рассматриваться в качестве основы для разработки баз знаний, можно выделить: 1) оболочки экспертных систем (CLIPS (FuzzyCLIPS, DYNACLIPS, WxCLIPS), SOAR, OPS83, RT-EXPERT, MIKE, BABYLON, WindExS, ES; ACQUARE, Easy Reasoner, ECLIPSE, EXSYS Professional, SIMER+MIR, АТ ТЕХНОЛОГИЯ, CAKE v2.0) [1]; 2) инструментальные пакеты для разработки экспертных систем (G2, ART, KEE, Knowledge KRAFT); 3) системы, ориентированные на обработку онтологий – Protégé, WebOnto, OntoEdit, WebODE, OilEd, OntoLingua.

Достоинствами приведенных инструментальных средств являются: поддержка представления знаний различного вида различными моделями представления знаний в рамках одной системы; наличие средств визуального проектирования баз знаний; наличие средств верификации базы знаний, включая проверку на непротиворечивость; возможность монотонного расширения базы знаний; наличие средств интеграции баз знаний; наличие средств поддержки обмена данными с внешней средой, включая средства обмена данными в реальном времени. Однако для всех указанных средств характерны следующие недостатки: в силу различных ограничений велики сроки разработки баз знаний (отсутствие развитых технологий разработки); узок круг инженеров баз знаний (из-за высоких стартовых требований к разработчику) – от разработчика требуется владение специальными знаниями по моделям и языкам представления знаний; не полностью решен вопрос интеграции баз знаний.

В работе предлагается подход к созданию технологии проектирования баз знаний, в основе которой лежат следующие положения: модульное проектирование баз знаний, интеллектуализация средств поддержки проектирования баз знаний и семантическое представление знаний. Эта технология включает: унифицированную модель баз знаний; библиотеку ip-компонентов баз знаний и инструментальные средства проектирования баз знаний; методику проектирования и интеграции баз знаний.

Унифицированная модель баз знаний. Одной из основных проблем разработки унифицированной модели базы знаний является проблема разработки единых мер, критериев оценки качества представленных знаний. Проблема заключается в том, что существующие критерии оценки, разработанные для отдельных моделей представления знаний, не определены для других моделей; критерии непротиворечивости, разработанные для теорий в логических моделях не всегда просто применить к моделям, в которых теория в явном виде отсутствует или знания в которых носят императивный характер, а не декларативный; критерии нечеткости представленной информации, разработанные для отдельных нечетких множеств и предикатов, не всегда просто применить в случаях, когда имеются сложноструктурированные знания нечеткого характера; оценки количества информации, разработанные для традиционных способов кодирования, могут иметь различные значения в зависимости от того, как представлять знания такими способами. Поэтому требуется иметь систему мер, которые работают для любых видов знаний в

***Ивашенко В.П.**, аспирант Белорусского государственного университета информатики и радиоэлектроники.*
Беларусь, БГУиР, 220013, г. Минск, ул. П. Бровки, 6.