

БЫСТРОЕ ОБНОВЛЕНИЕ КОЭФФИЦИЕНТОВ В ПАМЯТИ АДАПТИВНОГО КИХ-ФИЛЬТРА НА РАСПРЕДЕЛЕННОЙ АРИФМЕТИКЕ

А.Е. Новиков

Белорусский государственный университет информатики и радиоэлектроники,
кафедра электронных вычислительных средств
220013, Республика Беларусь, г. Минск, ул. П.Бровки, 6
телефон: +(37517)2938631; e-mail: novikov@bsuir.by

Приводится один из способов быстрого обновления коэффициентов в памяти адаптивного фильтра с конечной импульсной характеристикой (КИХ) на распределенной арифметике с подстройкой коэффициентов фильтра по методу наименьших квадратов (МНК). Устройство выполнено на микросхеме SPARTAN 3 и может быть использовано для подавления электрического эхо-сигнала в системах телекоммуникаций. Проведен анализ результатов.

Ключевые слова – адаптивный КИХ-фильтр, метод нормированных наименьших квадратов, распределенная арифметика, электрическое эхо.

1 ВВЕДЕНИЕ

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

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

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. Альтернативой может стать реализация КИХ-фильтра на распределенной арифметике [1,2,3]. В данной работе описан способ организации коэффициентов в памяти такого фильтра, при котором возможно их быстрое обновление.

2 АДАПТИВНЫЙ КИХ-ФИЛЬТР НА РАСПРЕДЕЛЕННОЙ АРИФМЕТИКЕ

Имеются два абонента – дальний (А) и ближний (Б). В случае, если абонент Б говорит одновременно с абонентом А (рис. 1), сигнал от абонента Б равен

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

где $x_B(m)$ – сигнал от абонента Б;
 $x_A^{echo}(m)$ – эхо-сигнал абонента А.

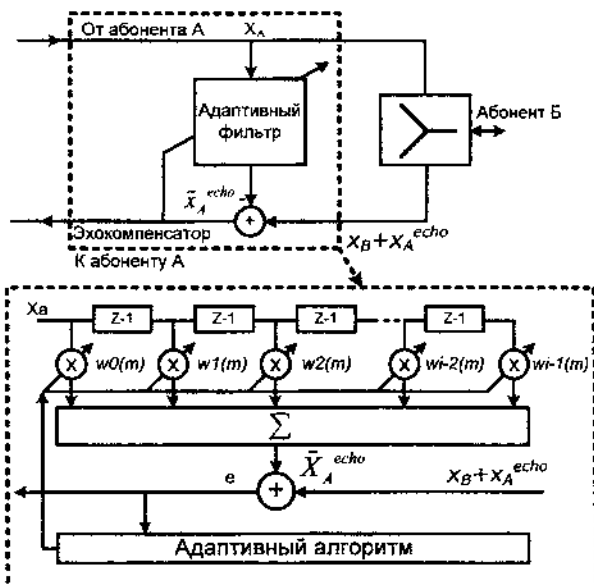


Рис. 1. Адаптивный КИХ-фильтр в составе эхокомпенсатора

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

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

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, который практически не влияет на процесс адаптации, поэтому при моделировании его учитывать не будем.

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.

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

где $w_i(m)$ – коэффициенты адаптивного фильтра;
 $\bar{x}_A^{echo}(m)$ – 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.

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

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

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

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

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

Отсчеты сигнала, которые должны быть отфильтрованы, представим как B -разрядные двоичные числа в дополнительном коде:

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

где b_{il} – это l бит двоичного дополнительного кода $x(m-i)$. 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 ($i=0, \dots, K-1$) члены в квадратных скобках могут 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:

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

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

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

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 $B + \lceil \log_2(n) \rceil$.

Второе слагаемое в этой сумме обусловлено необходимостью установки дерева сумматоров для получения суммарного сигнала с выходов меньших фильтров.

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.

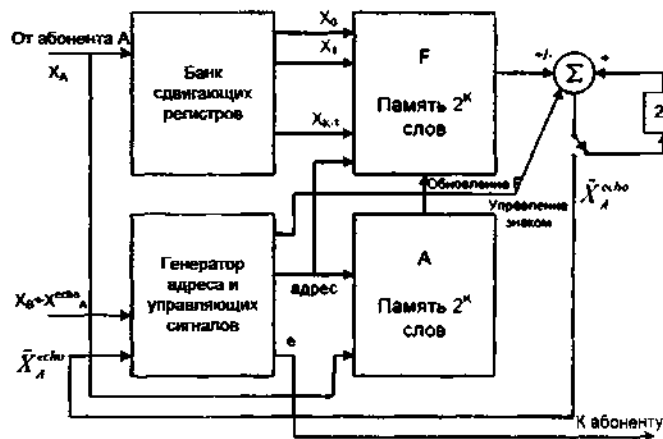


Рис. 4. Общая схема адаптивного КИХ-фильтра на распределенной арифметике

3 БЫСТРОЕ ОБНОВЛЕНИЕ КОЭФФИЦИЕНТОВ В ПАМЯТИ 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, являясь вычислительно сложной задачей и займет много времени, вследствие чего снизится пропускная способность фильтра. В [4] предложена структура, требующая меньшее число циклов, чем применение расчет и запись каждого коэффициента по отдельности.

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

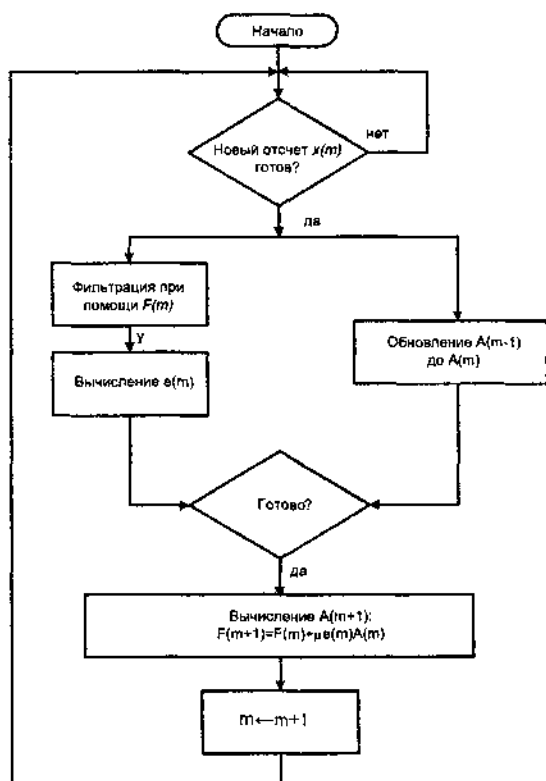


Рис.5. Упрощенный алгоритм работы адаптивного КИХ-фильтра на распределенной арифметике

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 i -той ячейки A это слагаемое

$$\sum_{i=0}^{K-1} c_i^{(r)} x(m-i)$$

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

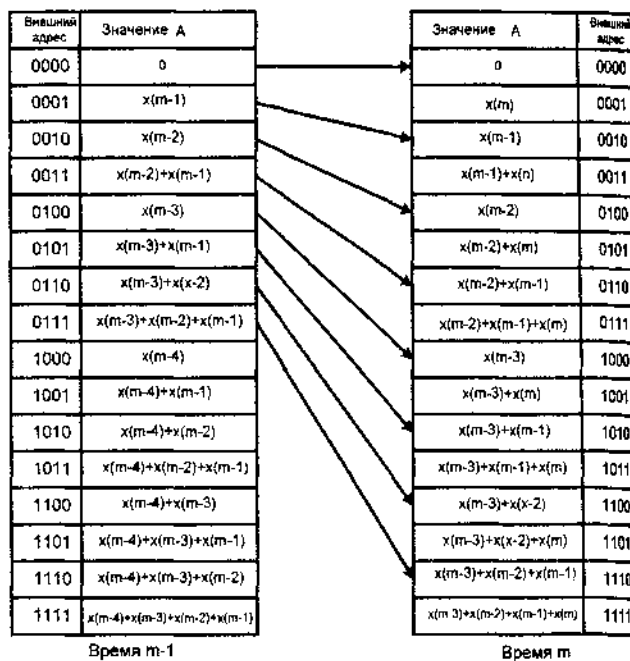


Рис.6. Обновление ячеек памяти $A(m-1)$ до $A(m)$

На рис. 6 показано обновление $A(m-1)$ 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) $A(m-1)$. 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

$$A_{i(2l+1)}(m) = A_{2l}(m) + x(m), \quad l = 0, \dots, 2^{K-1} - 1 \quad (14)$$

Обновление $A(m-1)$ до $A(m)$ происходит в 2 этапа.

Этап 1. Младшая половина $A(m-1)$ отображается по четным адресам в $A(m)$ как показано стрелками на рис. 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 рис. 7. Термин «внутренний адрес» обозначает

физический адрес, а термин «внешний адрес» обозначает адрес, видимый внешней логикой.

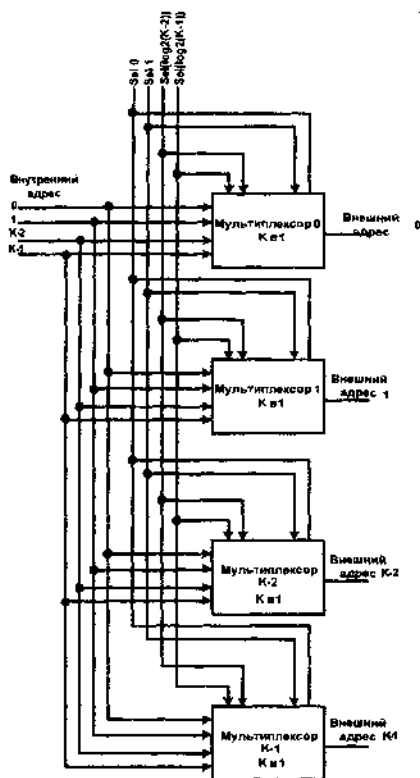


Рис.7. Схема циклического сдвига адресов

Можно заметить, что внешний адрес, ссылаясь на данный внутренний адрес в момент t , есть внешний адрес, циклически сдвинутый влево, ссылавшийся на тот же внутренний адрес в момент $t-1$. Таким образом, эффект циклического сдвига адреса может быть достигнут путем включения K мультиплексоров K в 1. $\log_2(K)$ адресных линий K мультиплексоров подключены к $\log_2(K)$ битам счетчика, который увеличивает свое значение с каждым тактом синхронизации. Так, циклический сдвиг адреса, перетображение A , может быть завершено мгновенно к приходу нового отсчета $x(m)$.

Этап 2. Нужно заметить, что циклический сдвиг адреса отображает верхнюю половину $A(m-1)$, содержащую суммы, включая старший отсчет $x(m-4)$, по нечетным адресам в момент t . Содержимое по этим нечетным адресам $A(m)$ перезаписываются значениями в соответствии с уравнением (14). Другими словами, содержимое по нечетным адресам $A(m)$ получают 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, также как и фильтрация, производится обновление $F(m+1)$. Из (13) следует, что обновление r -той ячейки F от t до $t+1$ выполняется в соответствии с уравнением

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

$F(m+1)$ 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.

4 ЗАКЛЮЧЕНИЕ

На основании метода проектирования адаптивных КИХ-фильтров с применением распределенной арифметики [4] создано VHDL описание фильтра. Аппаратная реализация устройства стала целесообразной благодаря возможности целиком обновлять ячейки памяти, хранящие суммы коэффициентов фильтра, а не коэффициенты каждый по отдельности.

ЛИТЕРАТУРА

- [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, "A new hardware realization of digital filters", *IEEE Trans. Acoustics, Sound, Signal Process.*, vol. ASSP-22, no 4, pp. 456-462, Dec. 1974.
- [3] S. A. White, "Applications of distributed arithmetic to digital signal processing: A tutorial preview", *IEEE ASSP Mag.*, vol. 6, pp. 4-19, Jul. 1989
- [4] Adaptive Filters Using Distributed Arithmetic for High Throughput. Daniel J. Allred, Heejong Yoo, Venkatesh Kristman, Walter Huang, David V. Anderson. *IEEE Transactions on Circuits and Systems*. Vol. 52, NO. 7, July 2005