

# Программирование цифрового линейно-фазового фильтра в архитектуре ARMv8

*А.М. Водовозов <am.vodovozov@gmail.com>*

*Д.С. Полетаев <dimanzaec@yandex.ru>*

*Вологодский государственный университет,  
160000, Россия, г. Вологда, ул. Ленина, д. 15*

**Аннотация.** Рассматривается задача использования процессоров с архитектурой ARMv8 для ускорения работы алгоритмов мультимедиа и цифровой обработки при решении задач восстановления сигналов в процессе фильтрации. В качестве примера рассмотрена реализация алгоритма работы цифрового КИХ-фильтра с линейной фазо-частотной характеристикой. Предложены формулы расчета фильтра. Алгоритм оптимизирован с использованием векторных SIMD-инструкций архитектуры ARMv8. Представлена реализация алгоритма обработки сигнала на языке Си на чипе BCM2837 с процессором ARM Cortex-A53. Решение обеспечило эффективное восстановление частот, искаженных при передаче сигналов в звуковом диапазоне, и доказывает эффективность использования мобильных многоядерных процессоров ARMv8 для параллельной обработки данных в процессе решения сложных вычислительных задач. Результаты эксперимента показывают, что использование процессоров с архитектурой ARMv8 при решении задач фильтрации сигналов позволяет существенно ускорить работу мультимедиа и алгоритмов обработки сигналов, таких как видеокодер/декодер, 2D/3D графика, игры, обработка звука и речи, обработка изображений, телефония и звук

**Ключевые слова:** цифровая обработка сигналов, линейно-фазовый фильтр, конечная импульсная характеристика, ARMv8, SIMD.

**DOI:** 10.15514/ISPRAS-2018-30(6)-17

**Для цитирования:** Водовозов А.М., Полетаев Д.С. Программирование цифрового линейно-фазового фильтра в архитектуре ARMv8-А. Труды ИСП РАН, том 30, вып. 6, 2018 г., стр. 305-314. DOI: 10.15514/ISPRAS-2018-30(6)-17

## 1. Введение

Появление 64-битного поколения процессоров с архитектурой ARMv8 и соответствующих инструментов компиляции открыло возможность повышения эффективности прикладных программ за счет использования преимуществ этой платформы. Одним из таких преимуществ является новый набор инструкций ARMv8, позволяющий выполнять несколько 128-битных операций векторизации параллельно. Причем для использования этих

операций не требуются каких-либо дополнительные усилия разработчиков, так как за всё отвечают встроенные в ядро аппаратные средства.

Одной из новых областей применения архитектуры ARMv8 можно считать решение задачи восстановления сигналов, искаженных при передаче по линиям связи, сопровождающей построение современных цифровых систем обработки информации и мультимедиа. Для решения такой задачи в структуре создаваемой системы обычно используются цифровые фильтры, которые позволяют восстановить исходный сигнал из принятого на основе данных о частотных характеристиках линии [1,2]. Цифровые фильтры влияют как на амплитуду сигнала, так и на его фазу, внося фазовые искажения.

В задачах управления фазовые искажения, как правило, нежелательны и указанный недостаток может быть исключен за счет использования линейно-фазового цифрового фильтра, который может быть реализован только при условии конечности его импульсной характеристики. В результате, при известных параметрах линий связи, для решения задачи используют фильтры с конечной импульсной характеристикой (КИХ-фильтры). Такие фильтры позволяют получить любую желаемую фазо-частотную характеристику, в том числе и линейную, при которой групповая задержка сигнала является величиной постоянной [3-6].

КИХ-фильтры могут быть реализованы нерекурсивно, т.е. с помощью прямой свертки, и всегда устойчивы. Однако для аппроксимации фильтров, частотные характеристики которых имеют значительный наклон, требуется импульсная характеристика с большим числом отсчетов и, при использовании в процессе реализации фильтра обычной свертки необходимо выполнять большой объем вычислений. Современная микропроцессорная архитектура ARM, активно используемая в задачах управления при создании встроенных приложений, открывает новые возможности при организации вычислений. Так, процессоры с архитектурой ARMv8 имеют мощные блоки SIMD, позволяющие эффективно выполнять параллельные вычисления в режиме реального времени, необходимые для реализации фильтров [7-9].

## **2. Алгоритм работы КИХ-фильтра**

Реализация КИХ-фильтра предусматривает формирование сигнала на выходе путем свертки дискретного входного сигнала с дискретной импульсной характеристикой желаемой системы [10]. Операцию можно описать выражением:

$$y[n] = \sum_{i=0}^N h[i] \cdot x[n-i], \quad n = 0, 1, 2, \dots, \quad (1)$$

где:

- $x$  – последовательность входных отсчетов (сигнал на входе);
- $h$  – импульсная характеристика длины  $N$  желаемой системы;
- $y$  – последовательность выходных отсчетов (сигнал на выходе системы).

Таким образом, проектирование КИХ-фильтра сводится к поиску нужной импульсной характеристики. Если известна дискретизированная по частоте комплексная частотная характеристика фильтра, то импульсную характеристику будем искать, применив к заданной частотной характеристике фильтра обратное дискретное преобразование Фурье (ОДПФ) [11]:

$$h[n] = \frac{1}{N} \sum_{k=0}^N X[k] \cdot e^{\frac{2\pi \cdot j}{N} \cdot k \cdot n}, \quad n = 0, 1, 2, \dots, N - 1, \quad (2)$$

где:

- $X$  – комплексная частотная характеристика системы;
- $h$  – импульсная характеристика системы;
- $N$  – количество отсчетов импульсной характеристики, а также количество точек, взятых на частотной характеристике.

Расчет фильтра начинается с определения желаемой частотной характеристики, которую можно разбить на две составляющие – амплитудную и фазовую. Фазо-частотная характеристика (ФЧХ) задается линейной, а амплитудно-частотная (АЧХ) может быть задана произвольной. На рис. 1 изображен пример фильтра Баттерворта с АЧХ пятого порядка и линейной ФЧХ. Для удобства расчетов частота дискретизации принимается за 1, поэтому частотные характеристики задаются для частот от 0 до 0.5.

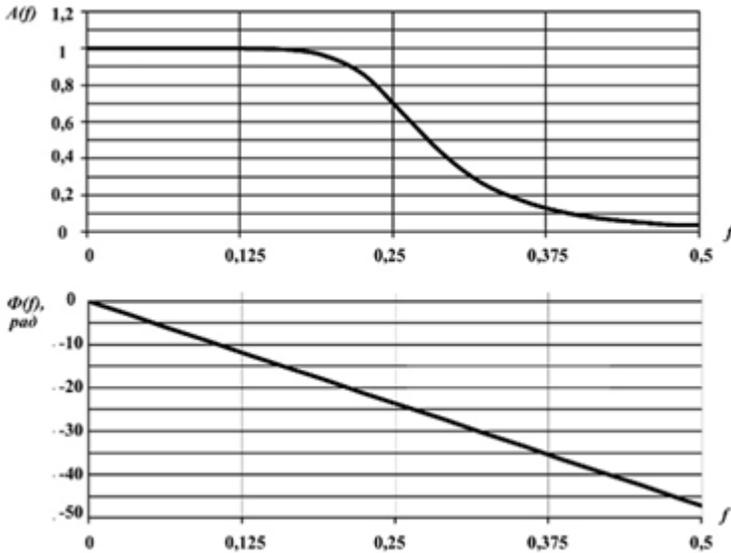


Рис. 1. Пример задания АЧХ и ФЧХ системы.

Fig. 1. Example of setting the frequency response and phase response of the system

Представленные частотные характеристики также можно описать комплексной функцией:

$$X(f) = A(f) \cdot e^{j\Phi(f)} = A(f) \cdot e^{-j2\pi f \frac{N-1}{2}}, \quad (3)$$

где:

- $f$  – относительная частота;
- $A(f)$  – амплитудно-частотная характеристика системы;
- $\Phi(f)$  – фазо-частотная характеристика системы;
- $N$  – количество коэффициентов КИХ-фильтра (на рисунке  $N=31$ );
- $(N-1)/2$  – задержка (число отсчетов), вносимая фильтром длины  $N$ .

Импульсная характеристика, получаемая через обратное преобразование Фурье, должна быть чисто вещественной, а это означает, что ее Фурье-образ является симметричным [12]. Поэтому перед дискретизацией и расчетом ОДПФ необходимо обеспечить симметрию АЧХ и ФЧХ (рис. 2).

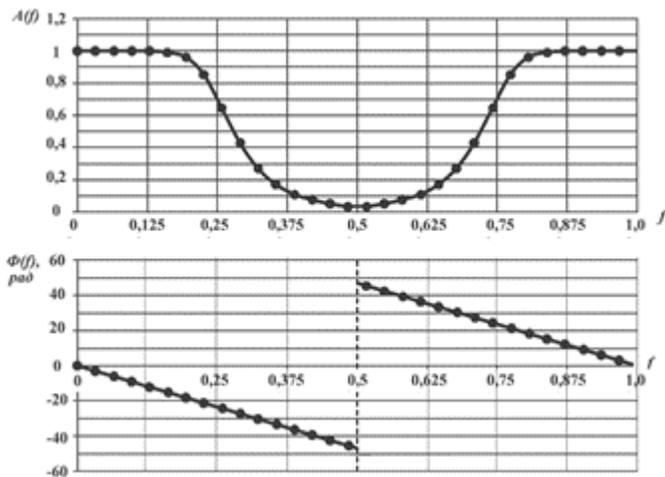


Рис. 2. Дискретизация комплексного коэффициента передачи для ОДПФ  
 Fig. 2. Discretization of complex transfer coefficient for Inverse Discrete Fourier Transform

Тогда значения комплексного коэффициента передачи будут определяться соотношением:

$$X[k] = \begin{cases} A\left(\frac{k}{N}\right) \cdot e^{-j2\pi \frac{N-1}{2} \frac{k}{N}}, & \frac{k}{N} < 0.5 \\ A\left(1 - \frac{k}{N}\right) \cdot e^{-j2\pi \frac{N-1}{2} \frac{N-k}{N}}, & \frac{k}{N} \geq 0.5 \end{cases}, \quad k = 0, 1, \dots, N-1, \quad (4)$$

где:

- $N$  – количество коэффициентов КИХ-фильтра, а также количество точек, взятых на частотной характеристике;
- $k$  – индекс частоты;
- $A(f)$  – желаемая амплитудно-частотная характеристика системы;
- $X$  – дискретизированная комплексная частотная характеристика.

В результате применения к последовательности  $X$  обратного дискретного преобразования Фурье (2) получается импульсная характеристика системы  $h[n]$  (рис. 3).

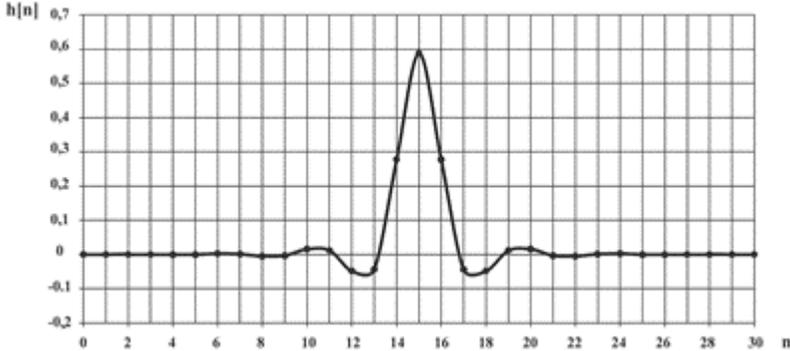


Рис. 3. Коэффициенты КИХ-фильтра, полученные через ОДПФ

Fig. 3. The coefficients of the FIR filter obtained through Inverse Discrete Fourier Transform

Если значения АЧХ системы не превышают 1, то коэффициенты фильтра лежат в диапазоне от -1 до 1. При целочисленной реализации алгоритма коэффициенты приводятся к диапазону целого знакового типа.

### 3. Программирование КИХ-фильтра

КИХ-фильтры требуют большого числа коэффициентов [2], поэтому для их реализации были использованы векторные SIMD-инструкции архитектуры ARMv8 [13]. Архитектура имеет поддержку набора команд SIMD с плавающей запятой, которые могут применять одну и ту же операцию к нескольким упакованным элементам одного типа и размера параллельно, что в несколько раз сокращает время выполнения. Программа реализована на языке Си с использованием ассемблерных команд архитектуры ARM.

В приведенном ниже листинге показан код функции, реализующей вычисление очередного отсчета выходной последовательности в соответствии с формулой (1).

```
//Реализация для ARMv8 с использованием SIMD инструкций
int16_t fir_compute_sample(int16_t* samples_buf_bottom,
                           int16_t* coeffs_arr, unsigned count)
{volatile int16_t result;
```

```
asm volatile(  
"eor V1.16B, V1.16B, V1.16B\n\t" //очищается аккумулятор  
//в основном цикле перемножаются параллельно 8 пар значений  
"FIR_MAIN_CYCLE:\n\t"  
"cmp %[pS]j %[smp_end_mult8]\n\t"  
"bhs FIR_MAIN_CYCLE_END\n\t"  
"ldl { V7.8H }, [%[pC]], #16\n\t"  
//загружается 8 16-битных коэффициентов в V7.8H  
"ldl { V5.8H [*[pS]], #16\n\t"  
//загружается 8 16-битных отсчетов в V5.8H  
"sqrdmulh V3.8H, V5.8H, V7.8H\n\t"  
//перемножается 8 отсчетов на 8 коэффициентов  
"add VI.8H, VI.8H, V3.8H\n\t"  
//результаты накапливаются в 8 аккумуляторах  
"b FIR_MAIN_CYCLE\n\t"  
"FIR MAIN CYCLE END:\n\t"  
""  
"saddlv SI, V1.8H\n\t"  
//складываются все 8 аккумуляторов в V1.S[0]  
"sqxtn VI.4H, V1.4S\n\t"  
//и результат преобразуется в 16 бит с контролем переполнения  
//в доп. цикле перемножается все, что осталось,  
//каждый раз только одна пара чисел  
"FIR_ADD_CYCLE:\n\t"  
"cmp %[pS], %[smp_end]\n\t"  
"beq FIR_END\n\t"  
"ldl { V7.H >[0], [%[pC]], #2\n\t"  
//загружается коэффициент в V7.H[0]  
"ldl { V5.H >[0], [%[pS]], #2\n\t"  
//загружается отсчет в V5.H[0]  
"sqrdmulh V3.4H, V5.4H, V7.4H\n\t"  
//перемножаем отсчет с коэффициентом  
"sqadd VI.4H, VI.4H, V3.4H\n\t"  
//результат добавляется к аккумулятору  
"b FIR_ADD_CYCLE\n\t"  
"FIR_END:\n\t"  
""  
"stl { VI.H }[0], [%[result]]\n\t"  
//аккумулятор выгружается в память  
:  
: [pC] "r" (coeffs_arr),  
: [pS] "r" (samples_buf_bottom),  
: [smp_end] "r" (samples_buf_bottom + count),
```

```
[smp_end_mult8] "r" (samples_buf_bottom + (count - count % 8)),  
[result] "r" (&result)  
);  
return result;  
}
```

Входные отсчеты реализуемой функции должны быть расположены последовательно, начиная с адреса «samples\_buf\_bottom». При обработке сигнала в реальном времени отсчеты могут поступать в циклический буфер. Вызов представленной функции каждый раз при поступлении нового отсчета позволит получить очередной выходной отсчет фильтра.

#### **4. Заключение**

Результаты экспериментов показывают, что реализация КИХ-фильтра в архитектуре ARMv8 с использованием SIMD-ускорения увеличивает скорость обработки сигнала примерно в 4 раза по сравнению с исходным кодом, позволяя в системах автоматического управления эффективно решать задачу восстановления сигналов датчиков. Например, на вычисление выходного отсчета КИХ-фильтра с длиной  $N = 1000$  на чипе BCM2837 с процессором ARM Cortex-A53 требуется менее 1 мкс. Это означает, что, к примеру, при обработке в реальном времени сигнала звукового диапазона с частотой дискретизации 48 кГц и периодом в 21 мкс фильтр может состоять, приблизительно, из 15 – 20 тысяч коэффициентов, что совершенно достаточно для коррекции АЧХ во всем диапазоне заявленных частот. С ростом числа коэффициентов повышается допустимая крутизна АЧХ фильтра и расширяется его нижняя граница по полосе пропускания.

Эксперимент доказывает перспективность использования мобильных многоядерных процессоров ARMv8 для параллельной обработки данных и решения сложных вычислительных задач мультимедиа и обработки сигналов, таких как видеокодер/декодер, 2D/3D графика, игры, обработка звука и речи, обработка изображений, телефония и звук.

#### **Список литературы**

- [1]. Лайонс Р. Цифровая обработка сигналов. М., Бинум, 2006, 656 стр.
- [2]. Dwivedi A.K., Ghosh S., Londhe N.D. Review and Analysis of Evolutionary Optimization-Based Techniques for FIR Filter Design. Circuits, Systems, and Signal Processing, vol. 37, no. 10, 2018, pp. 4409-4430.
- [3]. Paquelet S., Savaux V. On the symmetry of FIR filter with linear phase. Digital Signal Processing: A Review Journal, vol. 81, 2018, pp. 57-60.
- [4]. Строгонов А. КИХ-фильтры на параллельной распределенной арифметике. Компоненты и технологии, № 5, 2013, стр. 84-88.
- [5]. Бычков Д.Б., Дождев С.Ю. Аспекты оценки эффективности процессорных архитектур. Электронная техника. Серия 3: Микроэлектроника, № 2, 2015, стр. 34-37.

- [6]. Оппенгейм А., Шафер Р. Цифровая обработка сигналов. М., Техносфера, 2006, 856 стр.
- [7]. Ибрагимов Т.Р., Мунерман В.И. Возможность использования процессоров ARMV8 для параллельных вычислений. Системы компьютерной математики и их приложения, № 19, 2018, стр. 152-157.
- [8]. Belloch J.A., Alventosa F.J. Alonso P., Quintana-Ortí E.S., Vidal A.M. Accelerating multi-channel filtering of audio signal on ARM processors. *Journal of Supercomputing*, vol. 73, issue 1, 2017, pp. 203-214.
- [9]. Водовозов А. М. Микроконтроллеры для систем автоматки. М., Инфра-Инженерия, 2016. 164 стр.
- [10]. Winser A. *Digital Signal Processing: Principles, Algorithms and System Design* London, Academic Press, 2017, 617 p.
- [11]. Рабинер, Л., Б. Гоулд Б. Теория и применение цифровой обработки сигналов. М., Мир, 1978, 835 стр.
- [12]. Winser A., Cranos W. *Digital Signal Processing: Principles, Algorithms and System Design*. London, Academic Press, 2017, 617 p.
- [13]. ARMv8 Instruction Set Overview. Доступно по ссылке: [https://class.ee.washington.edu/469/peckol/doc/ARM/ARM\\_v8\\_Instruction\\_Set\\_Architecture\\_\(Overview\).pdf](https://class.ee.washington.edu/469/peckol/doc/ARM/ARM_v8_Instruction_Set_Architecture_(Overview).pdf), (дата обращения: 18.04.18).

## Programming of digital linear phase filter in ARMv8 architecture

*A.M. Vodovozov < am.vodovozov@gmail.com >*

*D.S. Poletaev < dimanzaec@yandex.ru >*

*Vologda State University,*

*15, Lenin st., Vologda, 160000, Russia*

**Abstract.** We consider the problem of using processors with an ARMv8 architecture to speed up the operation of multimedia algorithms and digital processing when solving problems of signal recovery in the filtering process. As an example, the implementation of the algorithm of a digital FIR filter with a linear phase-frequency characteristic is considered. The proposed formula for calculating the filter. The algorithm is optimized using vector SIMD instructions of the ARMv8 architecture. An implementation of the C signal processing algorithm on a BCM2837 chip with an ARM Cortex-A53 processor is presented. The solution provided effective recovery of frequencies distorted when transmitting signals in the audio range and proves the efficiency of using mobile multicore ARMv8 processors for parallel data processing in solving complex computational problems. Experimental results prove that the use of ARMv8 architecture processors when solving signal filtering problems significantly speeds up multimedia and signal processing algorithms such as video encoder / decoder, 2D / 3D graphics, games, sound and speech processing, image processing, telephony and sound.

**Keywords:** digital signal processing; linear phase filter; finite impulse response; FIR; ARMv8; SIMD

**DOI:** 10.15514/ISPRAS-2018-30(6)-17

**For citation:** Vodovozov A.M., Poletaev D.S. Programming of digital linear phase filter in ARMv8 architecture. *Trudy ISP RAN/Proc. ISP RAS*, vol. 30, issue 6, 2018, pp. 305-314 (in Russian). DOI: 10.15514/ISPRAS-2018-30(6)-17

## References

- [1]. Richard G. Lyons. *Understanding Digital Signal Processing* (2nd Edition). Prentice Hall, 2004, 665 p.
- [2]. Dwivedi A.K., Ghosh S., Londhe N.D. Review and Analysis of Evolutionary Optimization-Based Techniques for FIR Filter Design. *Circuits, Systems, and Signal Processing*, vol. 37, no. 10, 2018, pp. 4409-4430.
- [3]. Paquelet S., Savaux V. On the symmetry of FIR filter with linear phase. *Digital Signal Processing: A Review Journal*, vol. 81, 2018, pp. 57-60.
- [4]. Strogonov A. FIR filters on parallel distributed arithmetic. *Komponenty i tekhnologii* [Components and technologies], 2013, no. 5, pp. 84-88 (in Russian).
- [5]. Bychkov D.B., Dozhdev S.Yu. Aspects of evaluating the effectiveness of processor architectures. *Elektronnaya tekhnika. Seriya 3: Mikroelektronika*. [Electronic equipment. Series 3: Microelectronics], 2015, no. 2, pp. 34-37 (in Russian).
- [6]. Oppenheim Alan V., Schafer Roland W., Buck John R. *Discrete-Time Signal Processing*, 2nd ed., Prentice-Hall, 1999, 870 p.
- [7]. Ibragimov T.R., Munerman V.I. Ability to use ARMV8 processors for parallel computing. *Sistemy komp'yuternoy matematiki i ikh prilozheniya* [Computer mathematics systems and their applications], no. 19, 2018, pp. 152-157 (in Russian).
- [8]. Belloch J.A., Alventosa F.J. Alonso P., Quintana-Ortí E.S., Vidal A.M. Accelerating multi-channel filtering of audio signal on ARM processors. *Journal of Supercomputing*, vol. 73, issue 1, 2017, pp. 203-214.
- [9]. Vodovozov A.M. *Microcontrollers for automation systems*. M., Infra-Engineering, 2016, 164 p. (in Russian).
- [10]. Winser A. *Digital Signal Processing: Principles, Algorithms and System Design* London, Academic Press, 2017, 617 p.
- [11]. Рабинер, Л., Б. Гоулд Б. *Теория и применение цифровой обработки сигналов*. М., Мир, 1978, 835 стр.
- [12]. Winser A., Cranos W. *Digital Signal Processing: Principles, Algorithms and System Design*. London, Academic Press, 2017, 617 p.
- [13]. ARMv8 Instruction Set Overview. URL: [https://class.ee.washington.edu/469/peckol/doc/ARM/ARM\\_v8\\_Instruction\\_Set\\_Architecture\\_\(Overview\).pdf](https://class.ee.washington.edu/469/peckol/doc/ARM/ARM_v8_Instruction_Set_Architecture_(Overview).pdf), (accessed: 18.12.18).

