

Реализация симметричного разделенного КИХ-фильтра на ПЛИС

В.В. Горочный

НИУ МИЭТ, Зеленоград

Аннотация: Требовалось разработать вариант построения фильтра для реализации его на программируемой логической интегральной схеме (ПЛИС) в условиях ограничений ресурсов ПЛИС и времени на обработку. Были рассмотрены варианты построения фильтров и последовательная оптимизация структуры для реализации на ПЛИС. Были построены две модели в среде Matlab, где проводилось сравнение результатов обработки сигнала фильтрами при одинаковых параметрах. Моделирование показало, что фильтры идентичные.

Ключевые слова: цифровая обработка сигналов (ЦОС), фильтр с конечной импульсной характеристикой (КИХ-фильтр), конвейер, ПЛИС.

Введение

Использование цифровых вычислительных машин в системах, где требуется приём и обработка данных, позволяет проводить сложные операции преобразования входящих данных, включающие в себя вычленение полезной составляющей принимаемого сигнала за счет использования цифровых фильтров реализованных или программным обеспечением, или аппаратными средствами.

В условиях ограничений по времени и ресурсам программируемой логической интегральной схемы (далее – ПЛИС) на обработку требовалось проработать варианты фильтров, которые бы удовлетворяли требованиям по занимаемым ресурсам и времени на осуществления фильтрации.

Формулы

Предпочтительным выбором был фильтр с конечной импульсной характеристикой (далее – КИХ-фильтр) из-за отсутствия обратной связи в его структуре и линейной реализации [1], что изначально исключало проблемы округления при использовании математики с фиксированной точкой и как следствие этого, вероятность выхода фильтра из строя.

Формат данных для записи чисел с фиксированной запятой записывается в регистры как целочисленное значение, где для хранения

целой и дробной части знака выделяются строго фиксированные поля. Использование такого формата данных позволяет выполнять математические операции с дробными числами без возможности аппаратной поддержки с достаточной скоростью и точностью, заложенной в количестве разрядов для дробной части [2].

Применение математики с плавающей запятой сопряжено с большой сложностью обработки математических операций без специализированных блоков аппаратной поддержки [3], хоть это и дает больший диапазон возможных чисел по сравнению с числами с фиксированной точкой, но в данном контексте это не требуется.

Для расчётов и внутренней математики был выбран формат данных $fix27s14$ (Q13.14), он является расширенным до 27 бит форматом Q-чисел [4] с фиксированной точкой от AMD.

Основанием выбора формата данных с целочисленным полем, равным 12 бит служит разрядность входящих данных с аналого-цифрового преобразователя (далее – АЦП), что позволяет записывать целую часть без потерь. Дробная часть выбрана 14 бит для хранения значений в разрядной сетке с весом бита 0,00006104, что позволяет записывать числа с четвертым значащим знаком после запятой, и это обусловлено требованием для коэффициентов фильтров.

В качестве основы был взят классический линейный КИХ-фильтр, представленный на рис. 1.

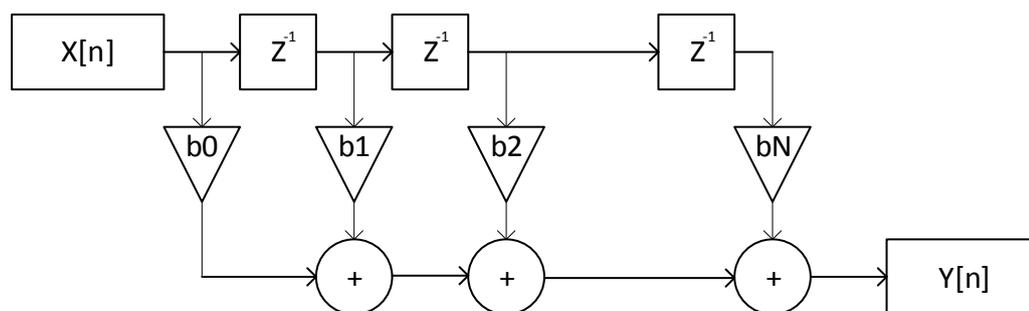


Рис. 1 – Линейный КИХ-фильтр

Фильтр, представленный на рис. 1, описывается формулой:

$$Y[n] = \sum_{i=0}^N b_i \cdot x(n - i)$$

КИХ-фильтр в линейном исполнении можно реализовать в ПЛИС и применять в устройствах, но данная структура нерационально использует ресурсы, поскольку для каждой линии задержки требуется хранить собственный коэффициент для умножителя.

Если использовать симметричные КИХ-фильтры, то для хранения коэффициентов понадобится в 2 раза меньше памяти от изначального количества [5].

Также, если учитывать архитектурные особенности ПЛИС, а именно - то, что память адресуется числами в двоичной системе счисления, то наилучшим вариантом станет использование фильтров с нечетным значением глубины фильтра, что исключит наличие операции сложения с одним операндом и одной операции умножения [6].

Для реализации подобного фильтра потребуется использовать структуру, которая представлена на рис. 2.

Фильтр, представленный на рис. 2, описывается формулой:

$$Y[n] = \sum_{i=0}^N b_i \cdot (x(2n - i) + x(i))$$

Данные для обработки поступают постоянно, но в определенный промежуток времени, который зависит от технических возможностей АЦП и частоты опроса ПЛИС. Диаграмма работы представлена на рис. 3.

Если частота осциллятора превосходит частоту приёма данных с АЦП в K раз, при этом $K \geq N_{\text{КИХ}}$, то выполнение обработки данных произойдет за 1 цикл (далее по тексту под циклом будет приниматься отрезок времени между обновлениями данных с АЦП).

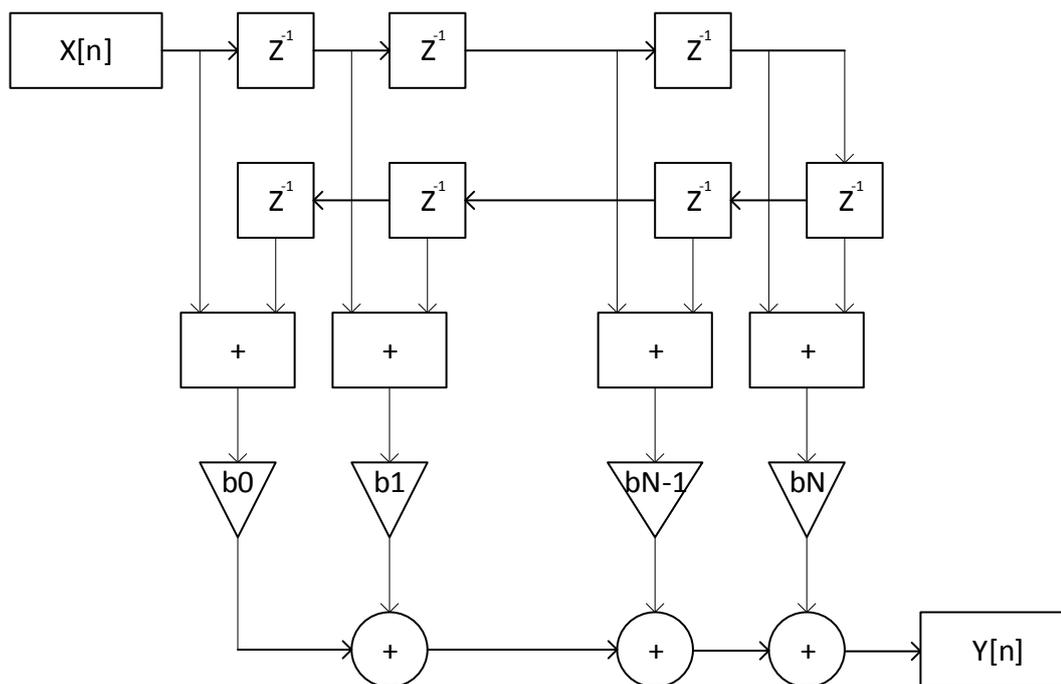


Рис. 2. – Симметричный КИХ-фильтр.

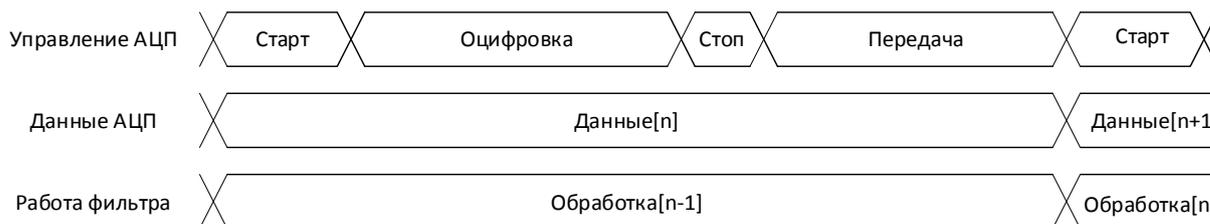


Рис. 3. – Диаграмма работы АЦП и фильтра.

Представленный симметричный КИХ-фильтр на рис. 2 уже более удобен для реализации на ПЛИС, но наличие большого числа умножителей приведет к значительному расходованию DSP элементов, а при их нехватке и LUT элементов [7].

Для более рационального использования ресурсов DSP, следует перейти на конвейерную систему построения КИХ-фильтра [8, 9], однако, для сохранения высокой скорости работы и уменьшения времени критического пути в логике ПЛИС, потребуется использовать промежуточные регистры [10], что накладывает другие требования на соотношения частот осциллятора и АЦП:

$$K + s \geq N_{\text{КИХ}}$$

где s – количество ступеней конвейера.

На рис. 4 представлена структура конвейерной реализации симметричного КИХ-фильтра.

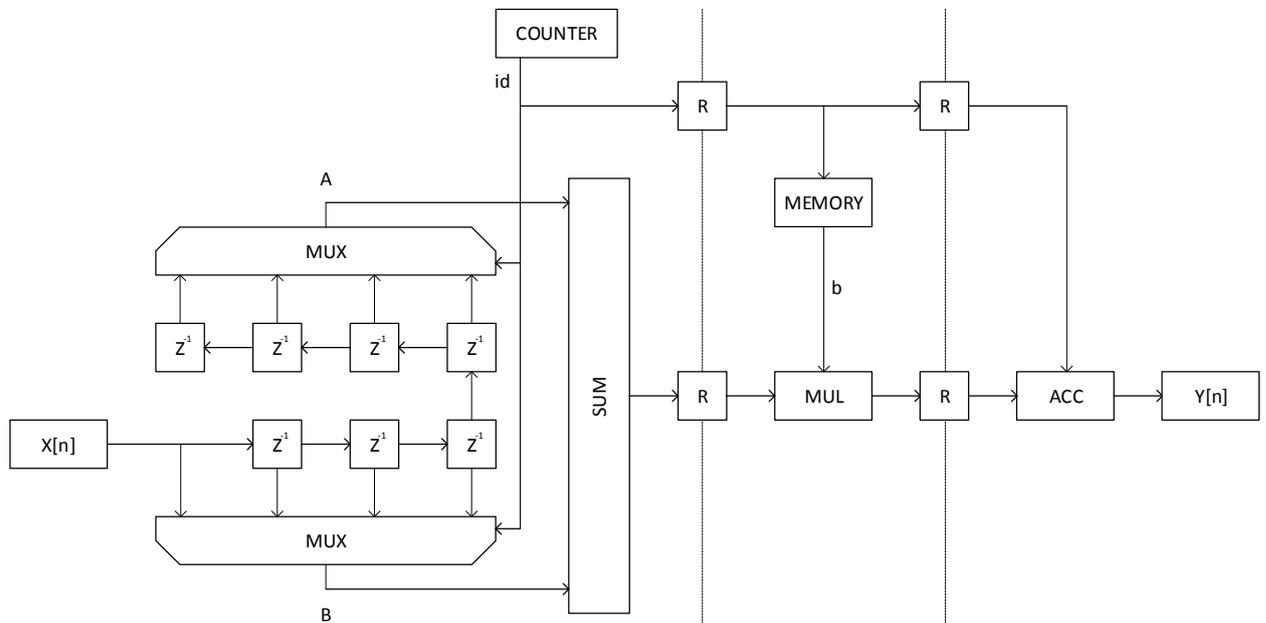


Рис.4. – Конвейерная реализация симметричного КИХ-фильтра.

Если сравнивать реализации симметричного КИХ-фильтра с конвейером, то можно увидеть, что коэффициенты b теперь располагаются в RAM-памяти ПЛИС и не занимают LUT-ресурсы, также требуется только несколько DSP-блоков для последовательного умножения всех пар чисел.

Параметр id , передающийся всем блокам, является управляющим и приводит конвейер в движение после того, как будет установлен равным 0, в начале каждого цикла, до момента, пока id не достигнет значения $N-1$. Расчёт значений с накоплением в конвейере на каждой стадии продолжается, пока значение управляющего не равно $N-1$, полученные значения аккумулируются, и в итоге, к концу работы конвейера, в аккумуляторе получается значение, равное фильтрованному значению для данного цикла работы фильтра.

При отсутствии возможностей повышать частоту осциллятора или уменьшать глубину КИХ-фильтра, чтобы уложиться в количество тактов в

одном цикле, требуется проектировать фильтры, расчеты в которых будут распределены на несколько параллельных частей.

Для наиболее оптимальной структуры таких фильтров следует придерживаться нескольких правил:

- 1) Каждый сегмент фильтра должен быть однотипным.
- 2) Каждый сегмент обрабатывает количество коэффициентов, равное степени 2.

При соблюдении данных условий можно наращивать глубину фильтра без изменения сегмента. При записи коэффициентов фильтра в память сегмента, можно будет делать запись параллельно, а требуемый участок памяти выбирать константой смещения.

На рис. 5 представлена структурная схема сегмента для реализации разделенного симметричного КИХ-фильтра:

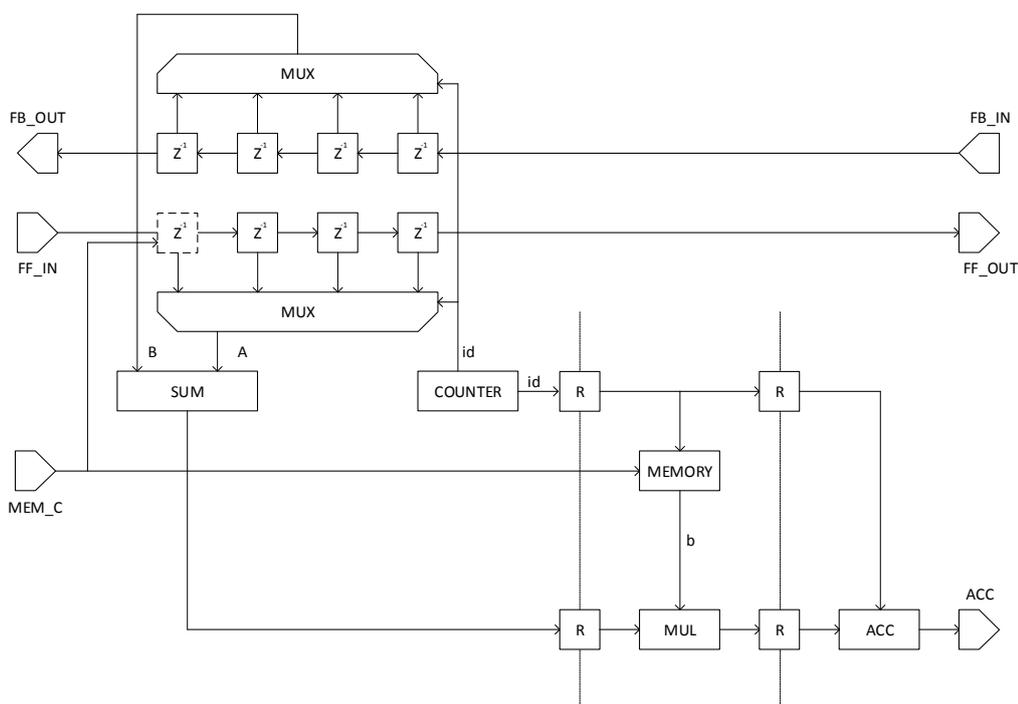


Рис. 5. – Структура сегмента симметричного разделенного КИХ-фильтра.

Последовательное соединение сегментов представлено на рис. 6.

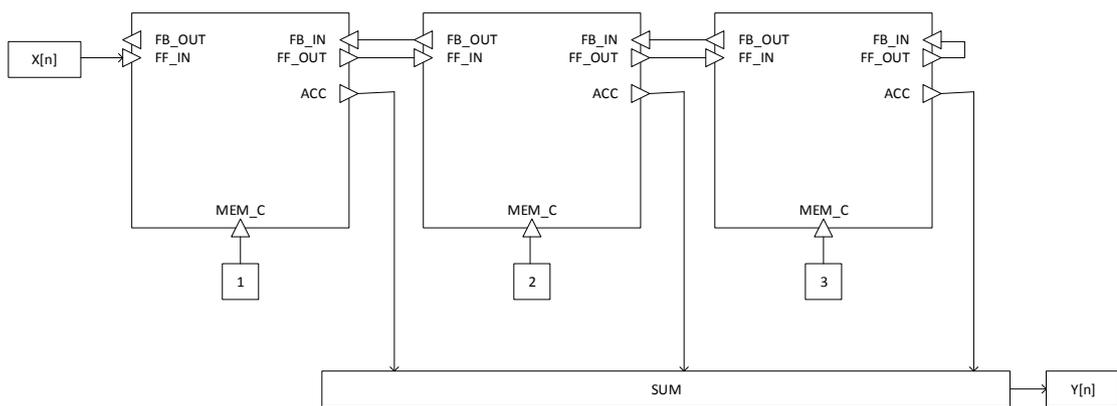


Рис. 6. – Последовательное соединение сегментов симметричного разделенного КИХ-фильтра.

Проверка фильтра проводилась в программной среде MatlabR2021a. Был создан М-файл с описанием структуры разделенного КИХ-фильтра.

Глубина исследуемого фильтра равна 17, используемые коэффициенты фильтра:

$$b = [1 \ 0.08 \ 0.01 \ 0.001 \ 0.001 \ 0.000 \ 0.00 \ 0.00 \ -1];$$

Для сравнения был так же описан симметричный фильтр с линейной структурой (рис. 2).

Тестовый сигнал сгенерирован функцией Matlab:

$$PULSE = wgn(1, 1000000, -1);$$

Сигнал ошибки находится, как:

$$\varepsilon = |Y_{КИХ}(n) - Y_{РКИХ}(n)|^2$$

Графики проведения эксперимента представлены на рис. 7.

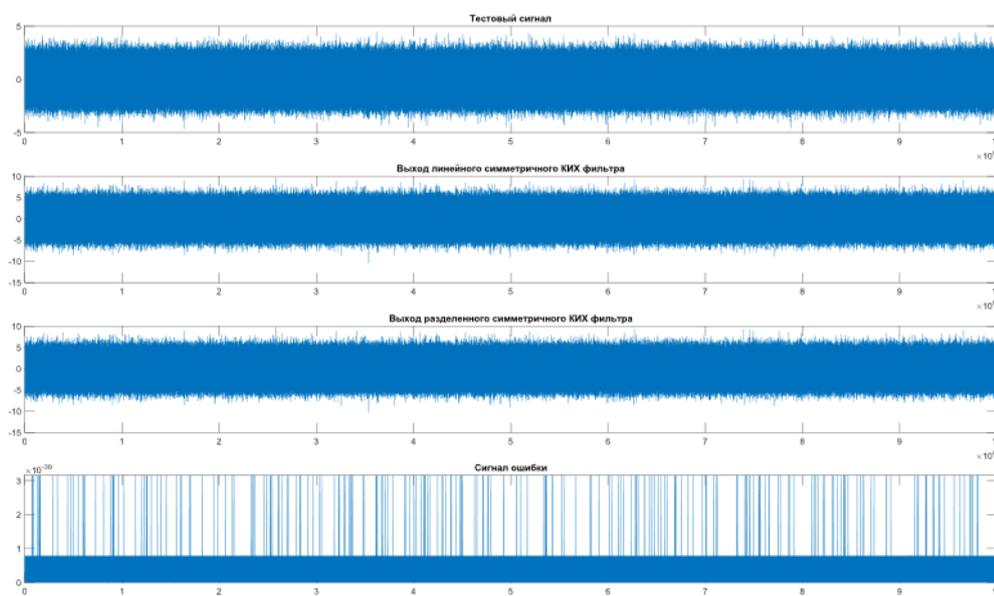


Рис. 7. – Результат проведения эксперимента на случайном сигнале с большим количеством точек

Выводы

Представленный вариант построения цифрового разделенного КИХ-фильтра является аналогичным линейному фильтру, но имеет более оптимизированную структуру для построения её на базе ПЛИС.

Литература

1. Камбалов Е.Ю. Проектирование КИХ-фильтров. Актуальные проблемы энергетики. СНТК-74. с. 795-796.
2. Вычужанин В.В. Реализация цифровой обработки сигналов переменной точности на ПЛИС. Цифровая обработка сигналов, 2011.№3. с. 62-66.
3. Опанасенко В.Н., Сахарин В.Г., Лисовый А.Н., Проектирование модулей с плавающей точкой на ПЛИС с использованием языка VHDL. Математичні машини і системи, 2005, № 3. с. 17-23.
4. ARM Developer Suite AXD and armsd Debuggers Guide. 1.2. ARM Limited. 2001. с. 398.

5. Каплун Д., Меркучева Т., Исследование реализации цифровых фильтров с симметричными амплитудными функциями. Компоненты и технологии. 2009. № 10. с. 108-112.

6. Строгонов А., Быстрицкий А. Проектирование КИХ- фильтров с учётом архитектурных особенностей ПЛИС. Компоненты и технологии, 2014. №8. с.122-127.

7. Cyclone IV Device Handbook, Volume 1. 101. Innovation Drive San Jose, CA 95134. URL:altera.com.

8. Erdogan, A.T.; Hasan, M.; Arslan, T. (2001). IEEE 14th Annual IEEE International ASIC/SOC Conference - Arlington, VA, USA (12-15 Sept. 2001). Proceedings 14th Annual IEEE International ASIC/SOC Conference (IEEE Cat. No.01TH8558). - A low power FIR filtering core, 271–275. doi:10.1109/asic.2001.954710

9. Строгонов А.В., Борисов Д.А., Проектирование быстродействующих перемножителей в базисе ПЛИС. Вестник Воронежского государственного технического университета. 2014. №2. с. 53-55.

10. Oswaldo Cadenas; Graham Megson (2004). A clocking technique for FPGA pipelined designs, 50(11), 687–696. doi:10.1016/j.sysarc.2004.04.001

References

1. Kambalov E.Yu., Proektirovanie KIX-fil'trov. Aktual'ny'e problemy e`nergetiki. [FIR filter design. Actual problems of energy]. SNTK-74. pp. 795-796.

2. Vy`chuzhanin V.V., Cifrovaya obrabotka signalov, 2011. №3. pp. 62-66.

3. Opanasenko V.N., Saxarin V.G., Lisovy`j A.N. Matematichni mashini i sistemi, 2005, № 3. pp. 17-23.

4. ARM Developer Suite AXD and armsd Debuggers Guide. 1.2. ARM Limited. 2001. p. 398.



5. Kaplun D., Merkucheva T., Komponenty` i texnologii. 2009. № 10 (99). pp. 108-112.
6. Strogonov A, By`striczkiy A, komponenty` i texnologii, 2014. №8. pp.122-127.
7. Cyclone IV Device Handbook, Volume 1. 101. Cyclone IV Device Handbook, Volume 1. 101. Innovation Drive San Jose, CA 95134. URL: altera.com.
8. Erdogan, A.T.; Hasan, M.; Arslan, T. (2001). IEEE 14th Annual IEEE International ASIC/SOC Conference - Arlington, VA, USA (12-15 Sept. 2001). Proceedings 14th Annual IEEE International ASIC/SOC Conference (IEEE Cat. No.01TH8558). A low power FIR filtering core, 271–275. doi:10.1109/asic.2001.954710
9. Strogonov A.V., Borisov D.A., Vestnik Voronezhskogo gosudarstvennogo texnicheskogo universiteta. 2014. №2. pp. 53-55.
10. Oswaldo Cadenas; Graham Megson (2004). A clocking technique for FPGA pipelined designs. 50(11), 687–696. doi:10.1016/j.sysarc.2004.04.001