

Проектирование цифровых фильтров в системе MATLAB/Simulink и САПР ПЛИС Quartus

Андрей СТРОГОНОВ,
д. т. н.
andreis@hotmail.ru

Одна из распространенных операций цифровой обработки сигналов - фильтрация. Для ее реализации могут быть использованы программно-аппаратные ресурсы цифровых сигнальных процессоров; ПЛИС по архитектуре FPGA; различные операционные устройства: регистры, умножители, сумматоры и соответствующее управляющее устройство. Может быть также проведена программная реализация с помощью высокоуровневых языков.

Вид импульсной характеристики цифрового фильтра определяет их деление на фильтры с конечной импульсной характеристикой (КИХ-фильтры) и фильтры с бесконечной импульсной характеристикой (БИХ-фильтры). Широкое применение цифровых КИХ-фильтров вызвано тем, что свойства их хорошо исследованы. Особенности архитектуры ПЛИС FPGA позволяют проектировать компактные и быстрые КИХ-фильтры на основе распределенной арифметики. Использование распределенной арифметики для реализации арифметических функций обеспечивает выигрыш в объеме задействованных ресурсов ПЛИС [1, 2].

Рассмотрим особенности расчета КИХ-фильтра в системе MATLAB/Simulink (пакет Signal Processing, среда FDATool) и с помощью мегафункции Mega Core FIR Compiler САПР ПЛИС Quartus. Для расчета КИХ-фильтра применяются следующие данные: частота взятия отсчетов F_s ; порядок фильтра n ; граница полосы пропускания f_p ; граница полосы задерживания (подавления) f_s ; неравномерность АЧХ в полосе пропускания δ_1 ; минимальное затухание в полосе задерживания δ_2 . На практике, как правило, вместо δ_1, δ_2 задают логарифмические величины R_p, R_s , измеряемые в децибелах:

$$R_p = 20 \lg(1 + \delta_1) / (1 - \delta_1), \quad R_s = 20 \lg \delta_2.$$

Цель примера — спроектировать два одинаковых по характеристикам КИХ-фильтра в среде FDATool и на основе мегафункции Mega Core FIR Compiler. Главным достоинством среды FDATool по сравнению с другими программами расчета КИХ-фильтров является возможность генерации кода языка VHDL с помощью Simulink HDL Coder. Однако код

языка VHDL в САПР ПЛИС Quartus непосредственно использовать нельзя. Сгенерированный код языка VHDL можно применить в ModelSim (Mentor Graphics HDL simulator).

Исходные данные для расчета КИХ-фильтра нижних частот показаны в таблице.

Таблица. Исходные данные для расчета КИХ-фильтра нижних частот

Параметры фильтра	Значение
Фильтр нижних частот	Low Pass
Частота взятия отсчетов F_s , Гц	48 000
Неравномерность АЧХ в полосе пропускания R_p , дБ	1
Минимальное затухание в полосе задерживания R_s , дБ	80
Переходная полоса, Гц	2400
Частота среза, F_c , Гц	9600
Тип окна	Blackman

Пример расчета КИХ-фильтра в среде FDATool показан на рис. 1. Среда FDATool представляет графический интерфейс для расчета фильтров и просмотра их характеристик. На вкладке Design Filter зададим тип синтезируемой АЧХ — фильтр нижних частот, тип фильтра — рекурсивный (FIR), метод синтеза — метод окон (синтез с использованием весовых функций). Более подробную информацию по работе со средой FDATool можно найти в работе [3]. Среда FDATool поддерживает больше методов синтеза, чем мегафункция Mega Core FIR. Преимущество мегафункции в том, что порядок проектируемого КИХ-фильтра (число отводов) оценивается автоматически. Но синтез АЧХ осуществляется методом окон. Этот недостаток компенсируется возможностью загрузки коэффициентов проектируемого фильтра, полученных, например, с использованием среды FDATool. При проектировании КИХ-фильтра в среде FDATool применяются

следующие методы: Equiripple — синтез фильтров с равномерными пульсациями АЧХ методом Ремеза; Least-Squares — минимизация среднеквадратичного отклонения АЧХ от заданной и метод окон (Window). В разделе Filter Order зададим порядок КИХ-фильтра, и именно тот, который рекомендует выбрать мегафункция Mega Core FIR (следует обратить внимание на достаточно высокий порядок КИХ-фильтра, выявляемый с ростом отношения F_s/F_c , что считается одним из недостатков его использования). Мегафункция также предлагает и метод синтеза — окно Блекмена (Blackman). Расчет фильтра осуществляется после нажатия кнопки Design Filter. На рис. 1 показана АЧХ, вычисленная с помощью формата с плавающей запятой (пунктир) и формата с фиксированной запятой (непрерывная линия).

На рис. 2 показана синтезируемая АЧХ (задается комплексный коэффициент передачи $|H(f)|$, определенный в диапазоне частот от нуля до $F_s/2$). Частота среза F_c равна 9600 Гц. В мегафункции Mega Core FIR Compiler задается переходная полоса (Transition Bandwidth), равная 2400 Гц, и частота среза, равная 9600 Гц (обозначается как cutoff freq (1)).

В методе окон $|H(f)|$ обратное преобразование Фурье этой характеристики дает бесконечную в обе стороны последовательность отсчетов импульсной характеристики. Для получения КИХ-фильтра заданного порядка эта последовательность усечается путем выбора центрального фрагмента нужной длины. Для ослабления паразитных эффектов в этом методе синтеза усеченная импульсная характеристика умножается на весовую функцию (окно), плавно спадающую к краям.

Вкладка Realize Model позволяет импортировать спроектированный КИХ-фильтр

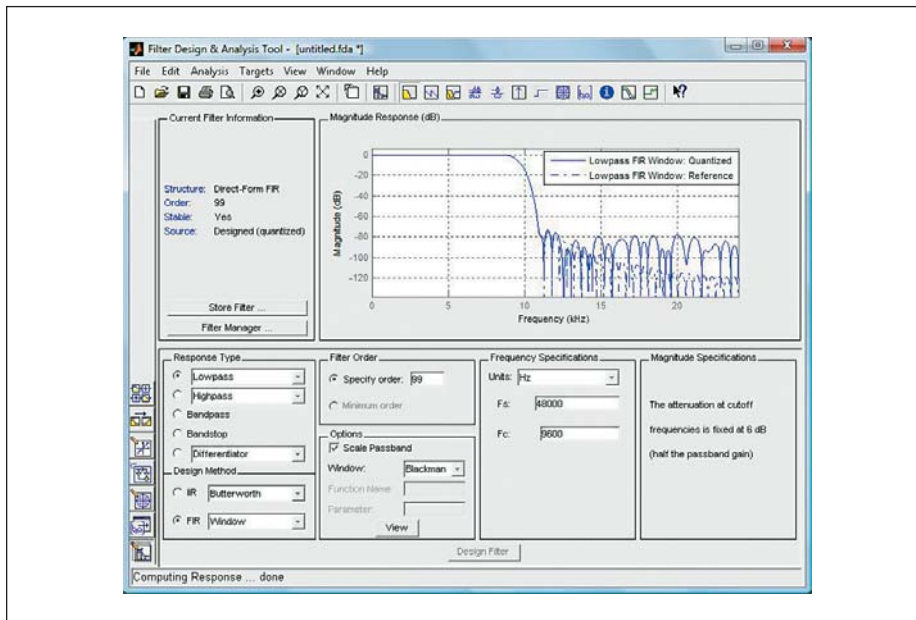


Рис. 1. Интерфейс среды FDATool. Пример расчета АЧХ КИХ-фильтра

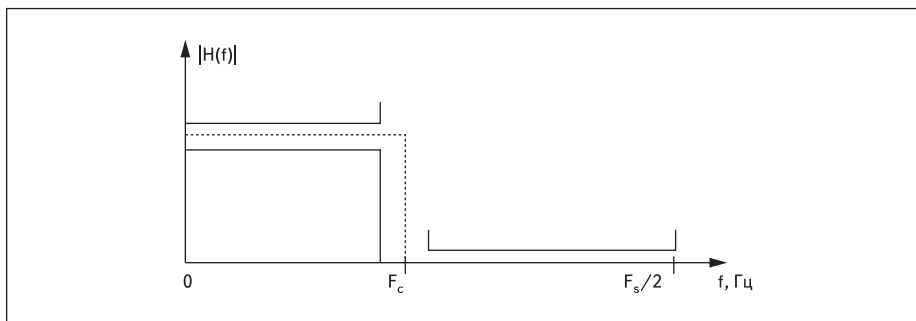


Рис. 2. Характеристики синтезируемой АЧХ (окно Blackman) КИХ-фильтра в среде FDATool

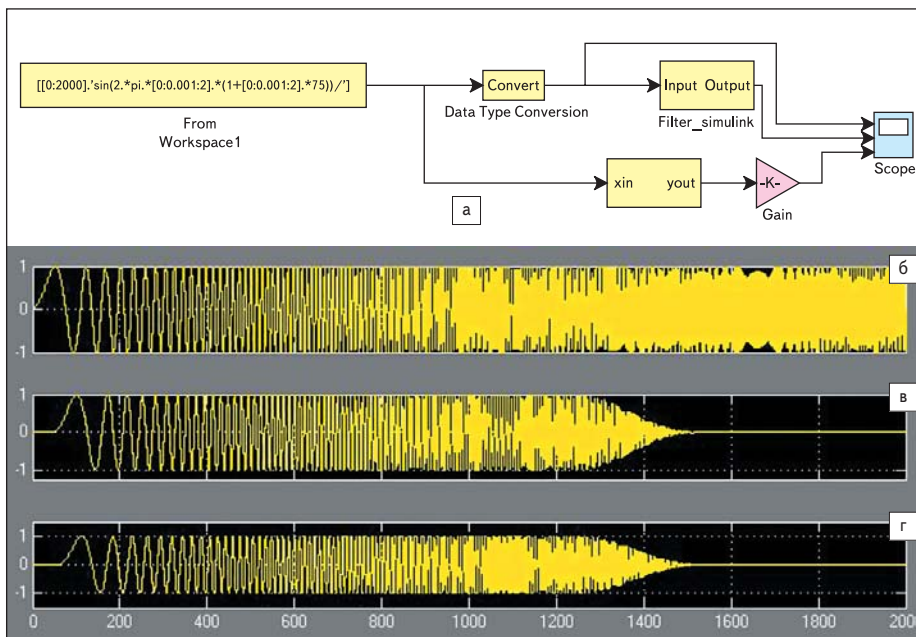


Рис. 3. а) Модель КИХ-фильтра в системе MATLAB/Simulink; б) сигнал до фильтрации КИХ-фильтром нижних частот; в) сигнал после фильтрации КИХ-фильтром нижних частот с использованием среды FDATool; г) с использованием мегафункции Core FIR Compiler САПР ПЛИС Quartus

(модель) в Simulink (рис. 3). На рис. 3а показана модель КИХ-фильтра (имя модели Filter simulink), построенная как с использованием базовых элементов (задержка, сумма, коэффициент усиления) цифровых фильтров, так и с использованием S-функции (модель КИХ-фильтра, построенная с использованием мегафункции Mega Core FIR Compiler). На рис. 3б показан сигнал до фильтрации, а на рис. 3в-г — после нее. Меню Targets и опция Generate HDL позволяют сгенерировать код фильтра на языке VHDL (рис. 4). Выберем параллельную архитектуру КИХ-фильтра, обладающую высокой производительностью.

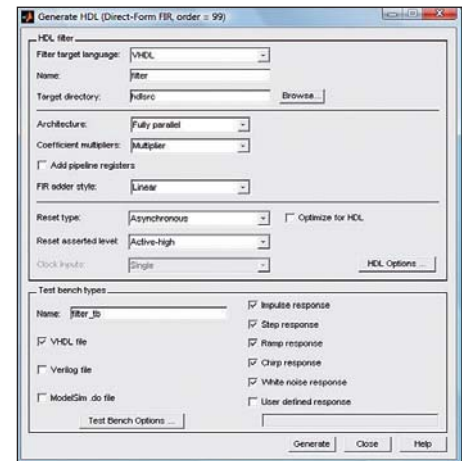


Рис. 4. Окно Simulink HDL Coder

Рассмотрим расчет КИХ-фильтра с помощью мегафункции Mega Core FIR Compiler в САПР ПЛИС Quartus (данная мегафункция может быть также использована и в САПР Max+Plus II). Применение Mega Core FIR Compiler позволяет быстро проектировать цифровой КИХ-фильтр исходя из заданных параметров. Быстрота и малая трудоемкость расчетов делает данное программное обеспечение незаменимым при проектировании КИХ-фильтров в базисе ПЛИС фирмы Altera.

Настроим мегафункцию так, чтобы АЧХ и импульсные характеристики совпадали с характеристиками фильтра, построенного с использованием FDATool системы MATLAB/Simulink. При запуске мегафункции Mega Core FIR Compiler открываются диалоговые окна мастера, в котором последовательно предлагается задать имя проекта, разрядность входной шины данных КИХ-фильтра, способ представления входных данных: signed — десятичное число со знаком, unsigned — десятичное число без знака.

После нажатия клавиши «Далее» появляется окно выбора способа расчета и формата представления весовых коэффициентов КИХ-фильтра (отводов). Существует возможность чтения коэффициентов как из текстового файла (коэффициенты могут быть получены, например, с использованием системы MATLAB/Simulink), так и генерация с помо-

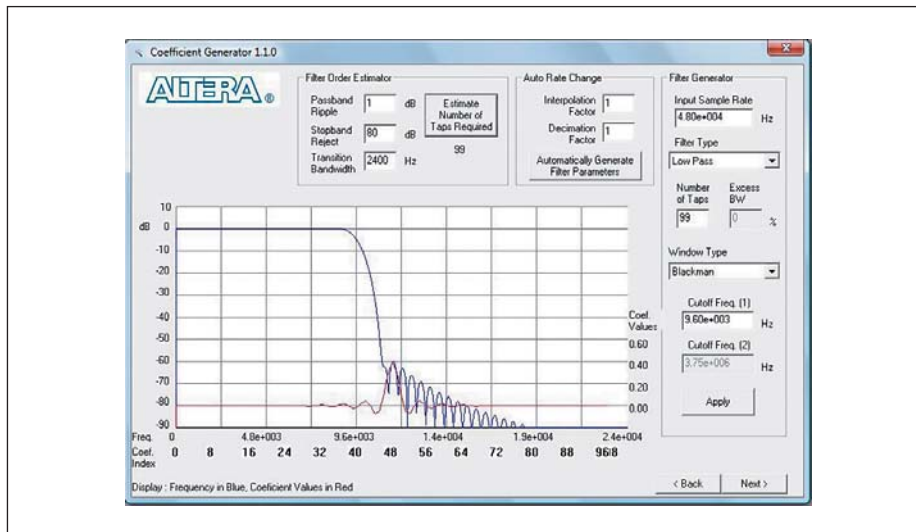


Рис. 5. Характеристики проектируемого КИХ-фильтра (АЧХ и импульсная характеристика)

стью встроенного модуля мегафункции Core FIR Compiler с учетом заданных параметров. Вид окна представлен на рис. 5. Мегафункция автоматически оценивает число отводов КИХ-фильтра при заданном пользователем типе окна.

Мегафункция представляет возможность проанализировать влияние преобразования из формата с плавающей запятой в формат с фиксированной запятой с заданной точностью и влияние величины масштабирования коэффициентов КИХ-фильтра на АЧХ.

Следующий шаг — графический анализ влияния ошибок квантования коэффициентов фильтра на его АЧХ. Ошибки квантования — представление коэффициентов фиксированным набором битов, например 8 бит. Операции сложения и вычитания в формате с фиксированной запятой не приводят к необходимости округления результатов — они могут лишь вызвать переполнение. В отличие от сложения умножение чисел с фиксированной запятой приводит к увеличению числа значащих цифр результата и, следовательно, к необходимости округления. Если результат умножения по модулю не превышает единицы, то применение формата с плавающей запятой даст большую точность [4]. Однако операции сложения в формате с плавающей запятой могут приводить к потере точности. В данном примере выбирается опция преобразования из формата с плавающей запятой в формат с фиксируемой запятой (закладка Floating point to fixed point conversion) с последующим масштабированием коэффициентов с точностью 8 бит. В соответствии с заданной точностью, мегафункция автоматически находит масштабный коэффициент. В нашем случае он составляет 318,8. При масштабировании фильтра все его коэффициенты делятся на 318,8 и на это же значение умножается рассчитанный выходной сигнал. Поэтому на рис. 3а сигнал с выхода КИХ-фильтра (модель под именем Altera

FIR filter) умножается на 0,0031368. Временные диаграммы на рис. 3в–г показывают, что спроектированные фильтры работают одинаково, что и требовалось в нашем примере. В качестве масштабирующего множителя можно также задать степень двойки (радиокнопка scale by a factor of) — 256. Покажем коэффициенты КИХ-фильтра в формате с плавающей и с фиксированной запятой, подвзвученные масштабированию:

Parallel FIR Filter 99 TAPS			
		4 bits (signed)	
		1 Channel	
		Pipeline delay: 10 clock cycles	
Coefficient 0 :	8.92292e-020	scaled to	0 (0X0)
Coefficient 1 :	-9.33441e-007	scaled to	0 (0X0)
...			
Coefficient 18 :	1.49801e-003	scaled to	0 (0X0)
Coefficient 19 :	-2.89381e-004	scaled to	0 (0X0)
Coefficient 20 :	-2.27251e-003	scaled to	-1 (0Xffffff)
Coefficient 21 :	-1.26514e-003	scaled to	0 (0X0)
Coefficient 22 :	2.14343e-003	scaled to	1 (0X1)
Coefficient 23 :	3.24095e-003	scaled to	1 (0X1)
Coefficient 24 :	-5.04205e-004	scaled to	0 (0X0)
Coefficient 25 :	-4.62497e-003	scaled to	-1 (0Xffffff)
Coefficient 26 :	-2.65247e-003	scaled to	-1 (0Xffffff)
Coefficient 27 :	4.11235e-003	scaled to	1 (0X1)
Coefficient 28 :	6.3542e-003	scaled to	2 (0X2)
Coefficient 29 :	-7.71564e-004	scaled to	0 (0X0)
Coefficient 30 :	-8.68246e-003	scaled to	-3 (0Xffffffd)
Coefficient 31 :	-5.16414e-003	scaled to	-2 (0Xffffffe)
Coefficient 32 :	7.3986e-003	scaled to	2 (0X2)
Coefficient 33 :	1.1794e-002	scaled to	4 (0X4)
Coefficient 34 :	-1.05834e-003	scaled to	0 (0X0)
Coefficient 35 :	-1.57364e-002	scaled to	-5 (0Xffffffb)
Coefficient 36 :	-9.79945e-003	scaled to	-3 (0Xffffffd)
Coefficient 37 :	1.3149e-002	scaled to	4 (0X4)
Coefficient 38 :	2.19545e-002	scaled to	7 (0X7)
Coefficient 39 :	-1.31698e-003	scaled to	0 (0X0)
Coefficient 40 :	-2.96964e-002	scaled to	-9 (0Xffffff7)
Coefficient 41 :	-1.98275e-002	scaled to	-6 (0Xffffffa)
Coefficient 42 :	2.57444e-002	scaled to	8 (0X8)
Coefficient 43 :	4.69944e-002	scaled to	15 (0Xf)
Coefficient 44 :	-1.49761e-003	scaled to	0 (0X0)
Coefficient 45 :	-7.4118e-002	scaled to	-24 (0Xffffffe8)
Coefficient 46 :	-6.01729e-002	scaled to	-19 (0Xffffffed)
Coefficient 47 :	9.41701e-002	scaled to	30 (0X1e)
Coefficient 48 :	0.301734	scaled to	96 (0X60)
Coefficient 49 :	0.398438	scaled to	127 (0X7f)
Coefficient 50 :	0.301734	scaled to	96 (0X60)
Coefficient 51 :	9.41701e-002	scaled to	30 (0X1e)
Coefficient 52 :	-6.01729e-002	scaled to	-19 (0Xffffffed)
Coefficient 53 :	-7.4118e-002	scaled to	-24 (0Xffffffe8)
Coefficient 54 :	-1.49761e-003	scaled to	0 (0X0)
Coefficient 55 :	4.69944e-002	scaled to	15 (0Xf)
Coefficient 56 :	2.57444e-002	scaled to	8 (0X8)
Coefficient 57 :	-1.98275e-002	scaled to	-6 (0Xffffffa)
Coefficient 58 :	-2.96964e-002	scaled to	-9 (0Xffffff7)
Coefficient 59 :	-1.31698e-003	scaled to	0 (0X0)

Coefficient 60 :	2.19545e-002	scaled to	7 (0X7)
Coefficient 61 :	1.3149e-002	scaled to	4 (0X4)
Coefficient 62 :	-9.79945e-003	scaled to	-3 (0Xffffffd)
Coefficient 63 :	-1.57364e-002	scaled to	-5 (0Xffffffb)
Coefficient 64 :	-1.05834e-003	scaled to	0 (0X0)
Coefficient 65 :	1.1794e-002	scaled to	4 (0X4)
Coefficient 66 :	7.3986e-003	scaled to	2 (0X2)
Coefficient 67 :	-5.16414e-003	scaled to	-2 (0Xffffffe)
Coefficient 68 :	-8.68246e-003	scaled to	-3 (0Xffffffd)
Coefficient 69 :	-7.71564e-004	scaled to	0 (0X0)
Coefficient 70 :	6.3542e-003	scaled to	2 (0X2)
Coefficient 71 :	4.11235e-003	scaled to	1 (0X1)
Coefficient 72 :	-2.65247e-003	scaled to	-1 (0Xffffff)
Coefficient 73 :	-4.62497e-003	scaled to	-1 (0Xffffff)
Coefficient 74 :	-5.04205e-004	scaled to	0 (0X0)
Coefficient 75 :	3.24095e-003	scaled to	1 (0X1)
Coefficient 76 :	2.14343e-003	scaled to	1 (0X1)
Coefficient 77 :	-1.26514e-003	scaled to	0 (0X0)
Coefficient 78 :	-2.27251e-003	scaled to	-1 (0Xffffff)
Coefficient 79 :	-2.89381e-004	scaled to	0 (0X0)
Coefficient 80 :	1.49801e-003	scaled to	0 (0X0)
...			
Coefficient 97 :	-9.33441e-007	scaled to	0 (0X0)
Coefficient 98 :	8.92292e-020	scaled to	0 (0X0)

Видно, что коэффициенты КИХ-фильтра симметричны относительно центральной величины 127, что обеспечивает линейность фазовой характеристики фильтра.

Далее предлагается выбрать точность представления фильтром промежуточных результатов. Формат хранения этих результатов вынуждает округлять их, что приводит к появлению дополнительных погрешностей (ошибок округления). По умолчанию берется Full Precision (полная точность). На следующем этапе задается фактор интерполяции (или децимации) фильтра, если необходимо спроектировать интерполяционный или децимирующий фильтр (по умолчанию фактор задается равным 1). Следующий шаг — задание архитектуры проектируемого КИХ-фильтра (параллельная или последовательная) и конвейерных свойств фильтра: оптимизация по скорости работы (частоте) или по площади занимаемых ресурсов (число задействованных макроячеек) ПЛИС.

На рис. 6а показана тестовая схема КИХ-фильтра (параллельная архитектура), полученная с использованием мегафункции Core FIR Compiler. На вход clock подается последовательность синхроимпульсов с периодом 40 нс. Вход clken является разрешающим сигналом подачи синхроимпульса clock, поэтому на него подается постоянный сигнал логической единицы. На вход Data[3..0] подается сигнал, подлежащий фильтрации, а с выхода DataOut[3..0] снимаем профильтрованный сигнал. Для того чтобы протестировать спроектированный КИХ-фильтр, подадим на вход Data[3..0] единичный импульс — аналог дельта-функции. В этом случае выходной сигнал Yout[13..0] является рассчитанным коэффициентом КИХ-фильтра, который представлен в формате с фиксированной запятой и масштабирован с точностью 8 бит. Результаты моделирования (импульсная характеристика) демонстрируются на рис. 6б.

Приведем фрагмент кода языка VHDL с использованием Simulink HDL Coder, описывающего КИХ-фильтр с параллельной архитектурой:

