

Проектирование КИХ-фильтра на умножителе

методом правого сдвига и сложения в базисе ПЛИС

Андрей СТРОГОНОВ,
д. т. н.
andreistrogonov@mail.ru
Алексей БЫСТРИЦКИЙ,
к. т. н.
bystritsky@bk.ru
Алан ДЖИОТИ

Самые быстрые умножители состоят из двумерной матрицы одноразрядных сумматоров и называются матричными умножителями. Наиболее распространены матричные умножители по схемам Бо-Вулли и Пезариса, а также древовидного формата. В отличие от умножителей, использующие метод сдвига и сложения, матричный умножитель представляет собой законченную логическую схему без элементов памяти. Реализовывать такие умножители для операндов большой разрядности в базисе ПЛИС в проектах пользователя возможно, но неэффективно по двум причинам. Во-первых, прямая реализация потребует чрезвычайно много логических ресурсов и времени, а во-вторых, структура матричных умножителей уже используется в каком-либо виде в качестве аппаратных умножителей ПЛИС, обеспечивая наивысшее быстродействие.

Общие сведения по программным умножителям

В ПЛИС для повышения их функциональных возможностей встраивают (например, для серии Cyclone III фирмы Altera) аппаратные умножители, которые могут быть сконфигурированы в виде одного умножителя 18×18 либо в виде двух умножителей 9×9 . Так, ПЛИС EP3CL5200 содержит 396 аппаратных умножителей 18×18 , а на оставшихся ресурсах может быть реализован 891 программный умножитель 16×16 . В итоге суммарное число умножителей составляет 1287 без какого-либо значительного использования логических ресурсов.

Для устройств цифровой обработки сигналов хорошо себя зарекомендовали софт-умножители (программные умножители), которые не требуют ресурсов аппаратных умножителей, встроенных в базис ПЛИС. Повысить производительность устройств цифровой обработки сигналов позволяет также использование параллельного векторного умножителя и безумножительные схемы умножения на основе распределенной арифметики [1].

Рассмотрим параллельные программные умножители, способные вычислять произведение за один такт синхроимпульса, что обеспечивает наивысшую производительность устройств цифровой обработки сигналов. Программные умножители БИС программируемой логики (БИС ПЛ) фирмы Actel серий Fusion, IGL00 и ProASIC3 ре-

ализуются на блочной памяти меньшей размерности, чем у ПЛИС фирмы Altera, и их можно рассматривать как 8-входовые LUT или таблицы произведений [2, 3]. Таблица произведений множимого, записанная во фрагмент блочной памяти, называется LUT.

В таблице 1 приведен умножитель размерностью 3×3 , реализованный с помощью 6-входовой LUT.

Например, в ПЛИС фирмы Actel используется ОЗУ емкостью $256 \text{ слов} \times 8 \text{ бит}$ (256 8-разрядных слов), а в ПЛИС фирмы Altera — память М4К, которая может быть сконфигурирована как $128 \text{ слов} \times 36 \text{ бит}$ или $256 \text{ слов} \times 18 \text{ бит}$ для серии Cyclone II. Такие умножители получили название RAM-LUT умножители, или LUT-based умножители.

На рис. 1 показан умножитель 4×4 на базе синхронного ОЗУ емкостью 256 8-разрядных слов. Множимое (младшие четыре разряда адресной шины) и множитель (старшие четыре разряда адресной шины), представленные 4-разрядным двоичным кодом, объединяются в 8-разрядную адресную шину, адресуя своим уникальным кодом содержимое конкретной строки ОЗУ (операнды), являющееся 8-разрядным произведением.

Недостатком такого умножителя является резкое возрастание требуемого объема блочной памяти в случае увеличения его разрядно-

Таблица 1. Умножитель размерностью 3×3 , реализованный с помощью 6-входовой LUT

		Множимое							
		000	001	010	011	100	101	110	111
Множитель	000	000000	000000	000000	000000	000000	000000	000000	000000
	001	000000	000001	000010	000011	000100	000101	000110	000111
	010	000000	000010	000100	000110	001000	001010	001100	001110
	011	000000	000011	000110	001001	001100	001111	010010	010101
	100	000000	000100	001000	001100	010000	010100	011000	011100
	101	000000	000101	001010	001111	010100	011001	011110	100011
	110	000000	000110	001100	010010	011000	011110	100100	101010
	111	000000	000111	001110	010101	011100	100011	101010	110001

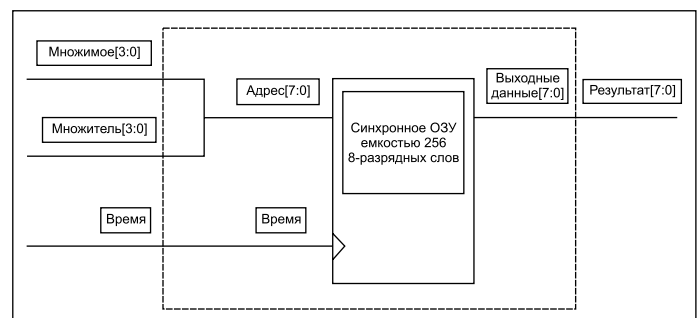


Рис. 1. Программный умножитель размерностью 4×4 на базе ОЗУ емкостью 256 8-разрядных слов (256×8) фирмы Actel

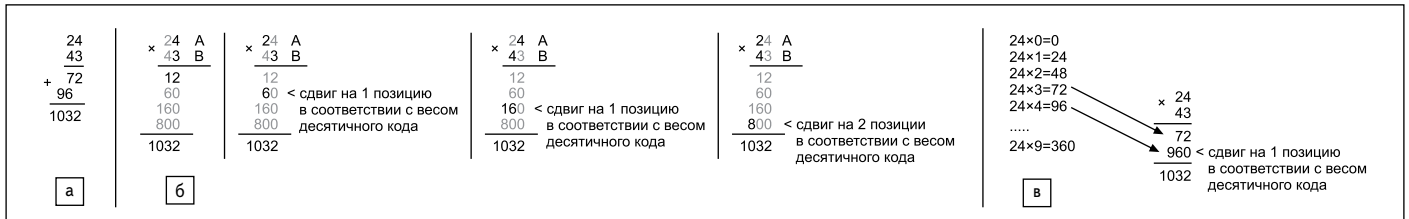


Рис. 2. Принцип умножения: а) «в столбик» по правилу умножения десятичных чисел; б) с использованием частичных произведений; в) на константу

сти. Для множителя размерностью 8×8 требуется 65 536 16-разрядных слов. Поэтому, чтобы предотвратить рост требуемой памяти, на практике используется множитель на суммировании частичных произведений в соответствии со своим весом (partial product multipliers).

На рис. 2 показан пример умножения десятичного числа 24 на 43. Например, произведению 2 на 4 приписывается вес 100, что равносильно сдвигу на две позиции в десятичной системе. В этом случае необходим множитель размерностью 8×8. Однако согласно принципу умножения с использованием частичных произведений требуется четыре множителя размерностью 4×4 для формирования четырех частичных произведений и три сумматора: $(4 \times 3 + ((2 \times 3) \times 10)) + ((4 \times 4) + ((2 \times 4) \times 10) \times 10) = 1032$. На рис. 3 приведена структурная схема такого множителя [2].

Также для сдвига на одну и две десятичные позиции потребуется три сдвиговых регистра на четыре разряда влево и дополнительные блоки, выполняющие операции расширения знака со значением старшего разряда.

Программные множители на константу

Одна из наиболее распространенных операций цифровой обработки сигналов — умножение числа на константу. Для перемножения двух чисел достаточно иметь табли-

цу произведений множимого (константы) на весь ранг возможных цифр множителя (табл. 2) и осуществить корректное суммирование полученных частичных произведений (рис. 2в).

Программные множители фирмы Actel реализуются на 8-входовых LUT. Множимое (константа) предопределено. В этом случае необходимо два блока памяти емкостью 256 8-разрядных слов, позволяющих организовать массив памяти 256 16-разрядных слов из двух блоков емкостью 256 8-разрядных слов, выходная шина которого и есть 16-разрядный результат умножения двух 8-разрядных чисел (рис. 4).

Что же касается программных множителей Altera, то встроенная блочная память TriMatrix в ПЛИС этой фирмы, например типа М9К, используемой в качестве LUT, в которых хранятся частичные произведения, позволяет реализовывать параллельные

умножители, экономя при этом не только аппаратные умножители, но и ресурсы логических блоков. Наличие программных и аппаратных умножителей приводит в целом к увеличению общего числа возможных умножителей.

Использовать программные и аппаратные умножители в проекте можно через мегафункции. Например, мегафункции `lpm_mult`, `altmult_add` и `altmult_accum` позволяют использовать аппаратные умножители. Рассмотрим мегафункцию `ALTMEMMULT` — программный множитель. С помощью этой мегафункции можно осуществлять процесс умножения числа на константу С, при этом константа может храниться в блочной памяти ПЛИС либо загружаться с внешнего порта.

Для ПЛИС серии Cyclone II есть возможность использовать только память М4К (128×36 бит) или режим Avto. Например, ПЛИС EP2С70 содержит 250 блоков М4К. На рис. 5 показан принцип умножения числа на константу. В целом он не отличается от ранее рассмотренного. При этом число и константа могут быть как со знаком, так и без него. А на рис. 6 показан принцип построения программного множителя 16-разрядного числа на 10-разрядную константу (обозначена буквой С) с использованием блоков памяти типа М4К большего размера, чем у БИС ПЛ фирмы Actel, для случая, когда константы хранятся в блочной памяти, то есть отсутствует возможность их загрузки

Таблица 2. Умножение 4-разрядного числа на константу 24 (рис. 2в)

Входы			X[3]	
X[2]	X[1]	X[0]	0	1
000			0	192
001			24	216
010			48	240
011			72	264
100			96	288
101			120	312
110			144	336
111			168	360

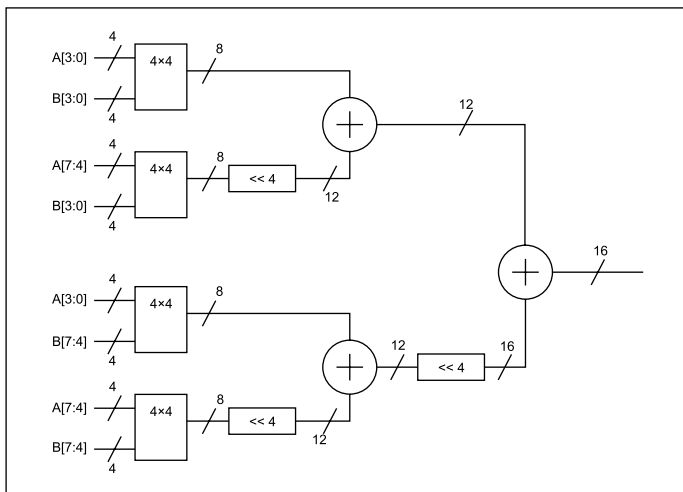


Рис. 3. Структурная схема множителя размерностью 8×8 с использованием четырех умножителей размерностью 4×4

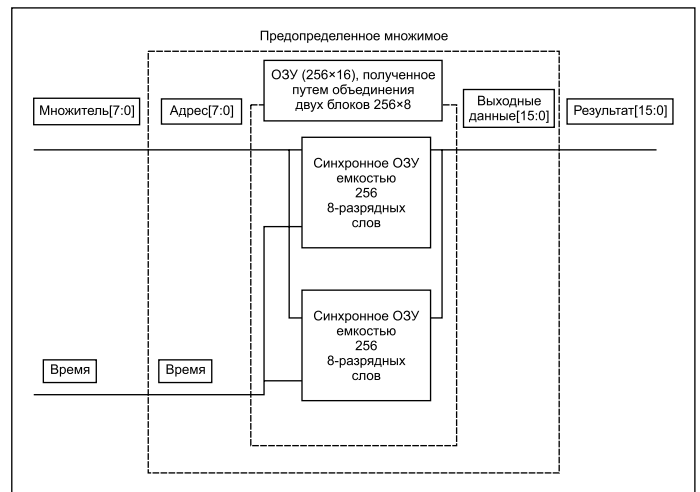


Рис. 4. Программный множитель размерностью 8×8 числа на константу на базе ОЗУ емкостью 256 16-разрядных слов фирмы Actel

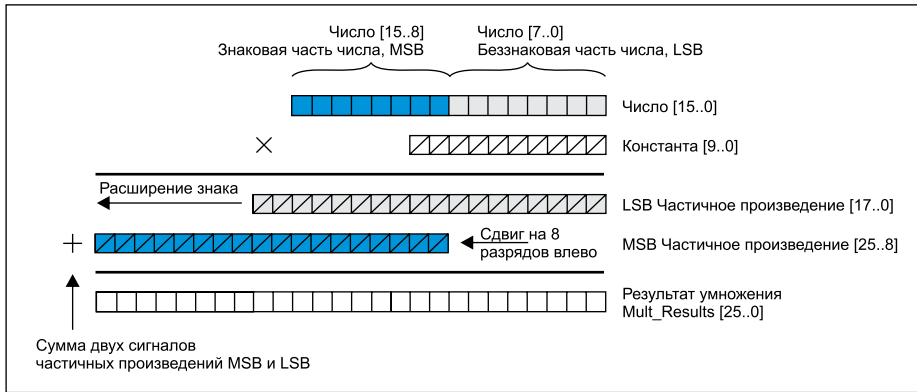


Рис. 5. Принцип параллельного умножения 16-разрядного числа на 10-разрядную константу

При загрузке константы из блочной памяти нужен всего лишь один M4K.

Принцип построения, представленный на рис. 5, не раскрывает все тонкости такого умножителя. В частности, не показана операция расширения знака числа.

На рис. 8 приведен принцип построения программного умножителя на константу размерностью 8×8 с использованием двух 4-входовых LUT ПЛИС серии XC4000 [4]. На рис. 8 обозначено: V — входной 8-разрядный сигнал; P1 и P2 — младшее и старшее частичные произведения; C — константа. В частности, здесь показано, как на практике осуществляется сдвиг на четыре разряда влево. Для умножителя требуется 25 конфигурируемых логических блоков (КЛБ). Объединяя такие умножители в секции (одна секция на отвод фильтра), можно построить высокопроизводительный параллельный КИХ-фильтр, работающий на частоте 50–70 МГц.

Проектирование КИХ-фильтра с использованием умножителя методом правого сдвига и сложения

Рассмотрим уравнение КИХ-фильтра (нерекурсивного цифрового фильтра с конечно-импульсной характеристикой), которое представляется как арифметическая сумма произведений:

$$y = \sum_{k=0}^{K-1} c_k x_k, \quad (1)$$

где y — отклик цепи; x_k — k -ая входная переменная; c_k — весовой коэффициент k -ой вход-

извне (отсутствуют адресные порты для загрузки коэффициентов).

Входной 16-разрядный сигнал разделяется на два 8-разрядных сигнала с именами LSB (младший значащий разряд) и MSB (старший значащий разряд) [3]. Сигнал LSB адресуется к блоку памяти M4K с одноименным названием LSB, а сигнал MSB адресуется к блоку памяти M4K с одноименным названием MSB. В блочной памяти LSB хранятся 256 предварительно вычисленных частичных произведений с именем «LSB Частичное произведение [17..0]» (младшее частичное произведение) разрядностью 18 бит с диапазоном от 0 до 255xС, а в памяти MSB с диапазоном от 0 до (-1)xС хранятся 256 предварительно вычисленных частичных произведений с именем «MSB Частичное произведение [25..8]» (старшее частичное произведение). Далее старшее частичное произведение необходимо сдвинуть

на восемь разрядов влево, а затем осуществить сквозное суммирование.

Умножение фактически осуществляется за один такт синхроимпульса, который необходим для загрузки входных значений сигналов LSB и MSB в адресные порты блоков памяти. Еще два такта требуются для конвейеризации задержки вычислений, так как выходные значения блоков памяти, представляющие собой частичные произведения, должны быть еще просуммированы с соответствующими весами для получения 26-разрядного результата умножения.

В качестве примера на рис. 7 показаны настройки мегафункции ALTMEMMULT для умножения 4-разрядного числа, представленного дополнительным кодом, и 4-разрядной константы, загружаемой из внешнего порта. В этом случае требуется 20LUT логических блоков плюс один блок памяти типа M4K и 20 триггеров (20LUT+1M4K+20reg).

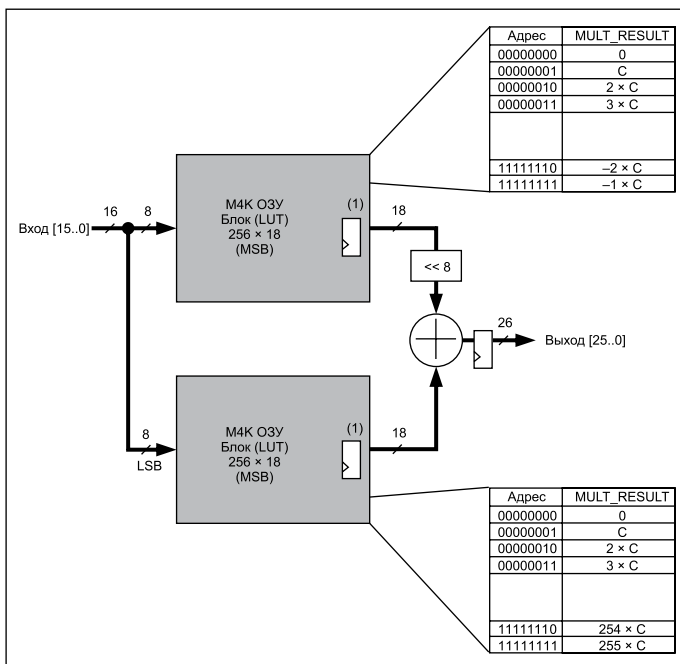


Рис. 6. Программный параллельный умножитель 16-разрядного числа на 10-разрядную константу размерностью 16×10 с использованием двух блоков памяти типа M4K в качестве LUT (Altera)

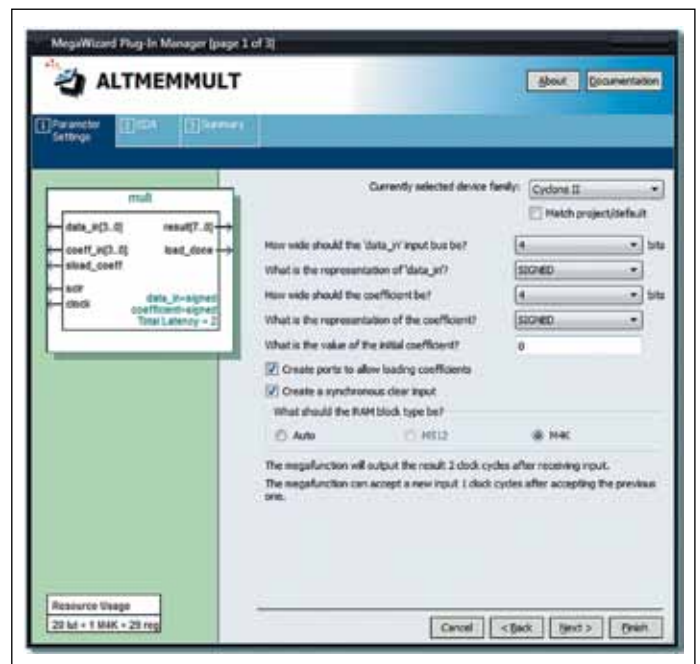


Рис. 7. Мегафункция ALTMEMMULT, настроенная для реализации программного умножителя 4×4

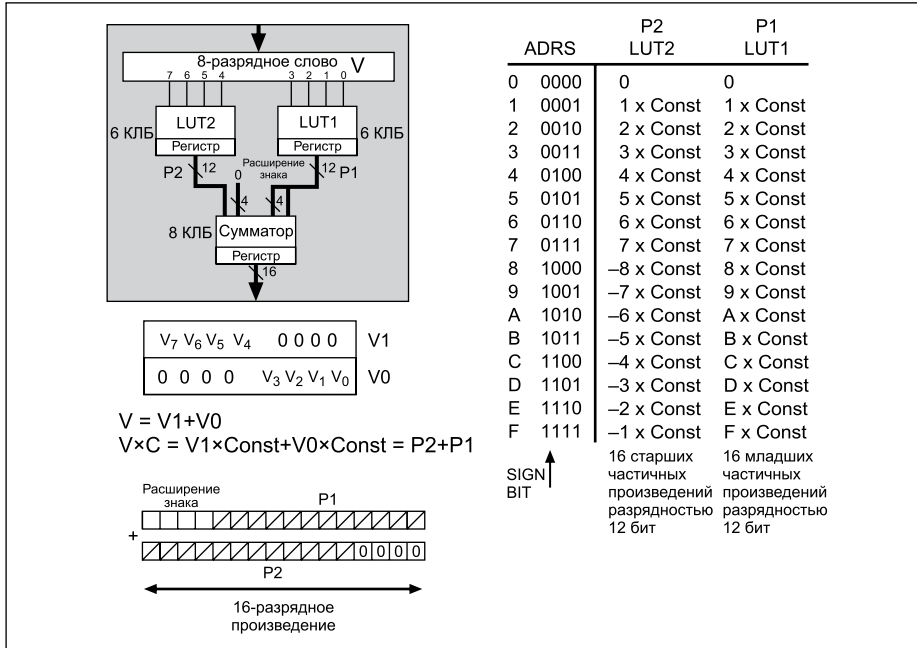


Рис. 8. Программный умножитель на константу размером 8×8 с использованием двух 4-входовых LUT ПЛИС серии XC4000

ной переменной, который является постоянным для всех n ; K — число отводов фильтра.

На рис. 9 показана тестовая схема КИХ-фильтра на четыре отвода $y = C_0x_0 + C_1x_1 + C_2x_2 + C_3x_3$ в САПР ПЛИС Quartus II для реализации в базе ПЛИС серии Cyclone II. Эта схема состоит из линии задержки, четырех умножителей и дерева многоарядных сумматоров. Предположим, что коэффициенты фильтра — целочисленные, со знаком, они известны и равны $C_0 = -2, C_1 = -1, C_2 = 7, C_3 = 6$.

Числа (входные отсчеты) поступают с выходов линии задержки на регистрах `reg_tap` на входы `data_in[3..0]` мегафункции `ALTMEMMULT`. Константы C_0, C_1, C_2, C_3 , в которых хранятся значения коэффициентов в дополнительном коде (14D, 15D, 7D и 6D), подключены к входам `coeff_in[3..0]`. В каждой из четырех мегафункций `ALTMEMMULT` в блочной памяти ПЛИС типа M9K хранятся нулевые коэффициенты (они могут быть и не нулевыми). Режим загрузки с внешнего порта или из блочной памяти определяется опцией `create ports to allow loading coefficients`. Латентность мегафункции — два

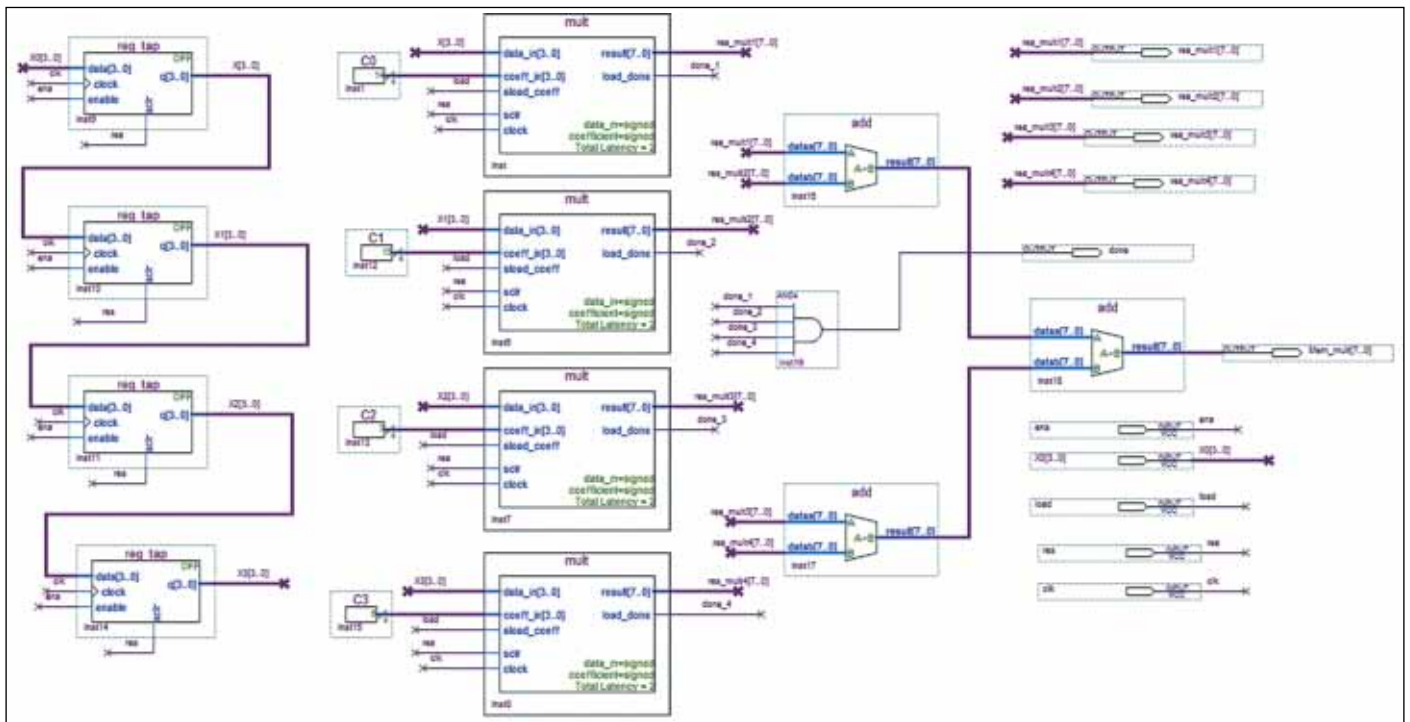


Рис. 9. КИХ-фильтр на четыре отвода с использованием мегафункции ALTMEMMULT

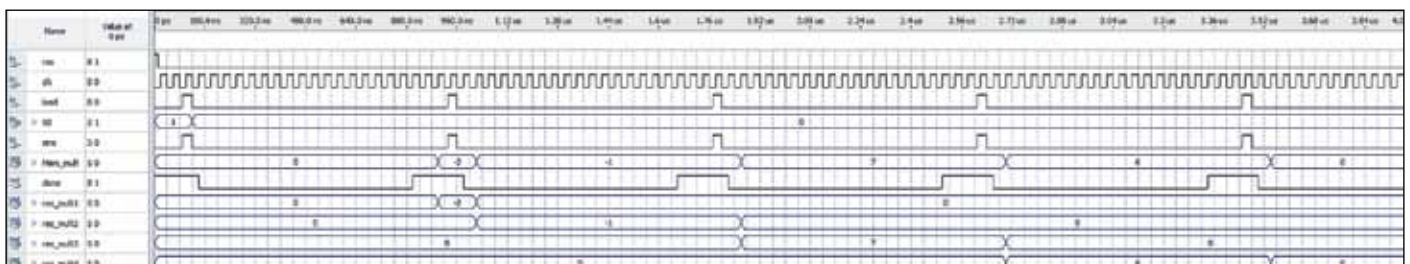


Рис. 10. Результат моделирования прохождения аналога дельта-функции по структуре КИХ-фильтра на четыре отвода

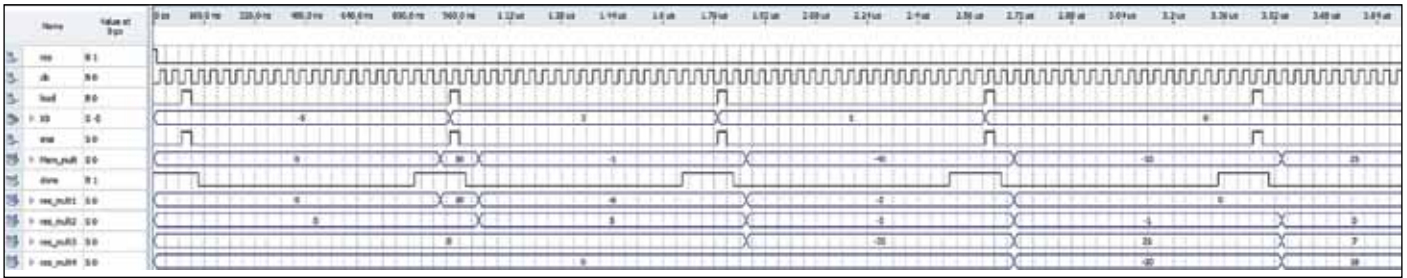


Рис. 11. Временные диаграммы работы КИХ-фильтра на четыре отвода при поступлении входных отчетов -5, 3, 1 и 0. Результат: 10, -1, -40, -10, 25

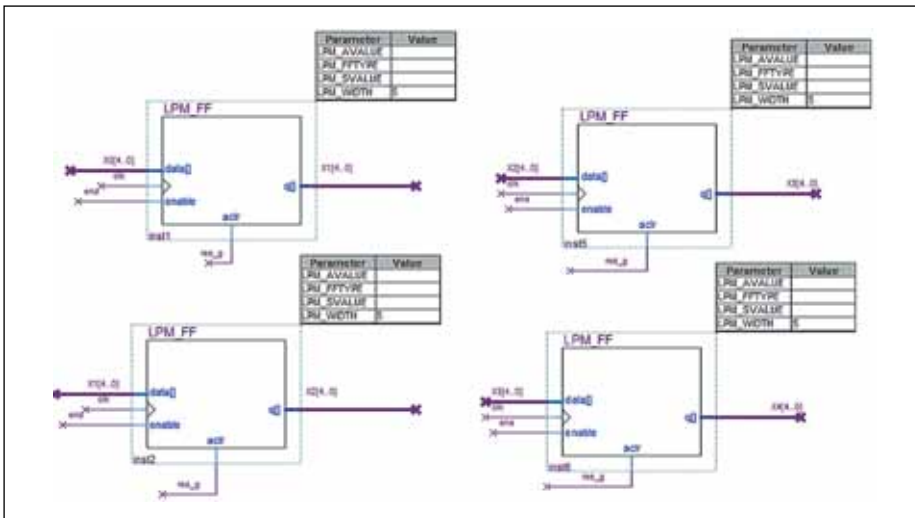


Рис. 12. Линия задержки КИХ-фильтра на четыре отвода на умножителях с использованием метода правого сдвига и сложения

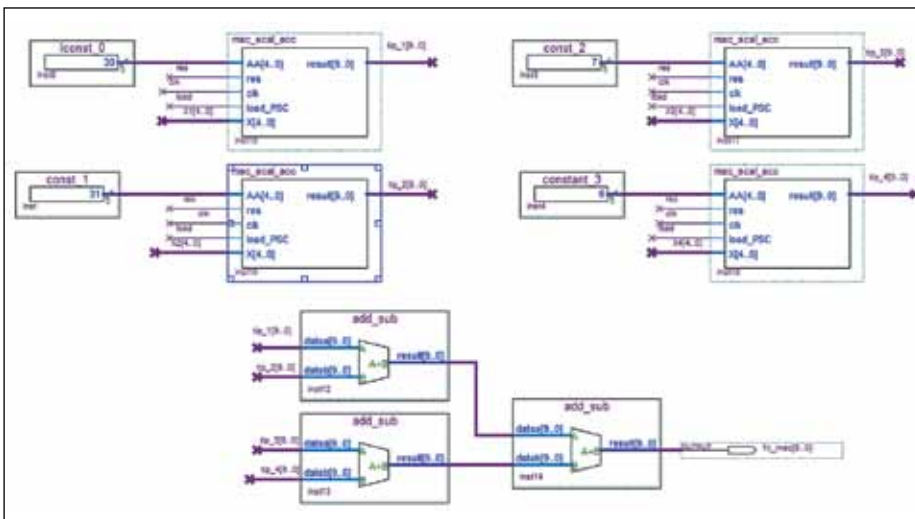


Рис. 13. Умножители и дерево сумматоров КИХ-фильтра на четыре отвода

умножители (mac_scal_acc) с константами и дерево сумматоров, а на рис. 14 — доработанная схема умножителя с учетом работы в составе КИХ-фильтра. Модификация в MAC-блоках выделена красным цветом. На рис. 15 показаны временные диаграммы работы КИХ-фильтра. Требуемые значения на выходе фильтра 10, -1, -40, -10, 25, 6 выделены красным цветом.

В таблице 3 представлены технические характеристики КИХ-фильтра на четыре отвода, реализованного в базе ПЛИС EP2C5F256C6 (4608 логических элементов, 119 808 бит встроенной памяти, 26 аппаратных умножителей 9×9) с использованием разработанного MAC-блока и программных умножителей на мегафункции ALTMEMMULT.

Таблица 3. КИХ-фильтр на четыре отвода, реализованный в базе ПЛИС EP2C5F256C6. Временная модель (Slow-model)

Технические характеристики	Разработанный MAC-блок	Программный умножитель на мегафункции ALTMEMMULT	
		Коэффициенты загружаются из внешнего порта, МГц	Коэффициенты загружаются из памяти (умножитель на константу), МГц
Частота в наилучшем случае, Fmax	240	210	260

Выводы

Смена данных на выходе КИХ-фильтра при использовании MAC-блока в качестве умножителя и умножителя на мегафункции ALTMEMMULT при загрузке коэффициентов из внешнего порта приблизительно одинакова (22 MAC и 21 ALTMEMMULT такта синхроимпульса). Для сравнения в работе [6] показано, что мегафункция ALTMULT_ADD также позволяет построить параллельный КИХ-фильтр на четыре отвода за счет использования встроенных четырех аппаратных перемножителей с размерностью операндов 18×18. Рабочая частота в наилучшем случае для ПЛИС серии Stratix III составила 400 МГц.

Повысить производительность КИХ-фильтра с помощью программного умножителя на мегафункции ALTMEMMULT по-

такта синхросигнала. Смоделируем прохождение дельта-функции по структуре фильтра. Для этого на вход линии задержки фильтра Х0[3..0] в дополнительный код подадим единичный импульс «лог. 1». На выходе фильтра Mem_mult[7..0] видим коэффициенты фильтра (импульсную характеристику) (рис. 10).

Подадим на вход КИХ-фильтра входные отсчеты -5, 3, 1 и 0 (рис. 11). Правильные

значения на выходе фильтра: 10, -1, -40, -10, 25, 6 и т. д., то есть согласно формуле (1).

Рассмотрим КИХ-фильтр на четыре отвода на умножителях (MAC-блоках) с использованием метода правого сдвига и сложения. Дополнительную информацию о разработанном MAC-блоке можно получить в работе [5].

На рис. 12 показана линия задержки на двухтактных триггерах с использованием мегафункции LPM_FF. На рис. 13 приведены

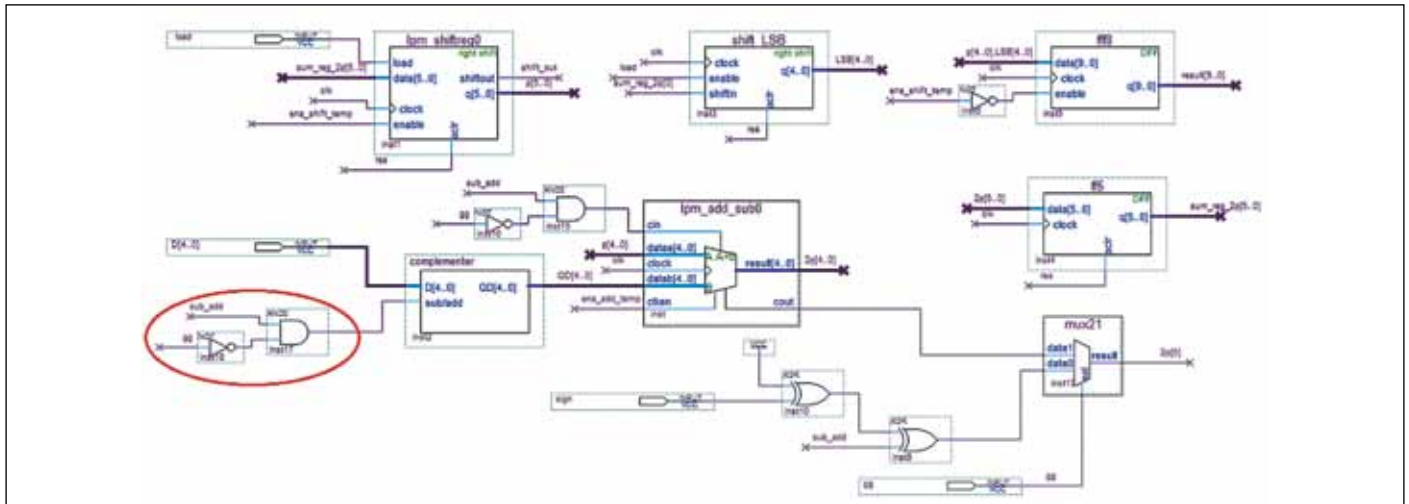


Рис. 14. Доработанная схема умножителя с учетом работы в составе КИХ-фильтра

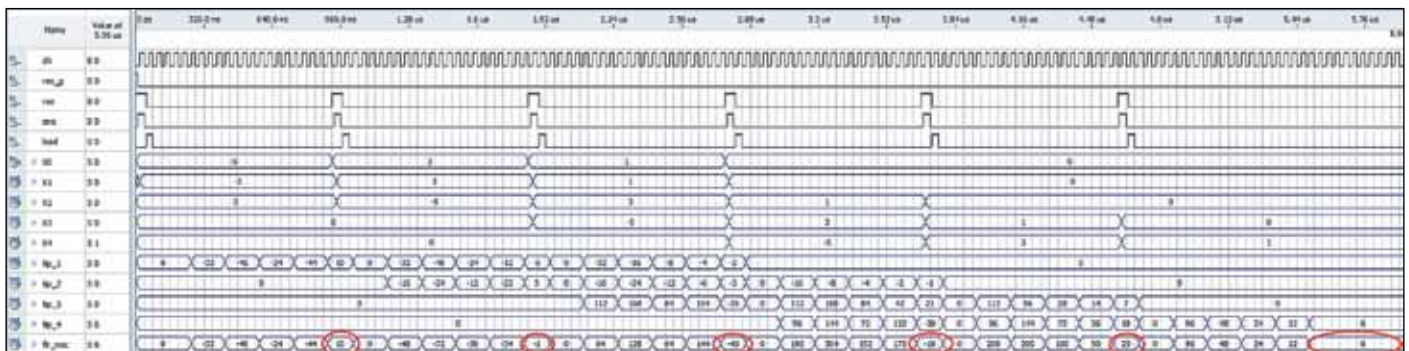


Рис. 15. Временные диаграммы работы КИХ-фильтра на четыре отвода на умножителях с использованием метода правого сдвига и сложения при поступлении входных отчетов $-5, 3, 1$ и 0 . Результат: $10, -1, -40, -10, 25, 6$

зволяет вариант загрузки коэффициентов из блочной памяти ПЛИС (260 МГц).

Разработанный в [4] МАС-блок на основе метода правого сдвига и сложения может быть использован для проектирования высокопроизводительных и компактных КИХ-фильтров небольшой разрядности. При этом на его реализацию в базе ПЛИС потребуются незначительные логические ресурсы (менее 1%), что позволяет сэкономить аппаратные и программные умножители. Недостатком такого МАС-блока является тру-

доемкость его разработки и наличие своего интерфейса в отличие от унифицированных интерфейсов мегафункции ALTMEMMULT, что может затруднить его использование. ■

Литература

1. Строгонов А. КИХ-фильтры на параллельной распределенной арифметике // Компоненты и технологии. 2013. № 5.
2. Application Note AC222. Using Fusion, IGLOO, and ProASIC3 RAM as Multipliers — www.actel.com
3. AN-306-3.0. Application Note 306. July 2004, ver.3.0. Implementing Multipliers in FPGA Devices.
4. Newgard B. Seminar. Signal processing with Xilinx FPGAs. June 9, 1996. xdsp.ppt. Xilinx Hardware Digital Signal Processing.
5. Строгонов А., Винокуров А., Джиоти А. Проектирование умножителя целых чисел со знаком методом правого сдвига и сложения в базе ПЛИС // Компоненты и технологии. 2014. № 1.
6. Строгонов А., Быстрицкий А. Проектирование параллельных КИХ-фильтров в базе ПЛИС // Компоненты и технологии. 2013. № 6.