

Проектирование параллельных КИХ-фильтров в базе ПЛИС

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

Перемножители сигналов играют ключевую роль в проектировании высокопроизводительных цифровых фильтров. В статье показаны различные варианты реализации КИХ-фильтров с использованием перемножителей на мегафункциях `ALTMULT_ACCUM`, `ALTMULT_ADD` и `ALTMEMMULT` САПР Quartus II компании Altera [1] в базе ПЛИС. Авторы также сосредоточили свое внимание на реализации умножения методом правого сдвига с накоплением, применяемого для разработки масштабирующего аккумулятора.

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 -я входная переменная; C_k — весовой коэффициент k -й входной переменной, который является постоянным для всех n ; K — число отводов фильтра.

В качестве простейшего примера рассмотрим три варианта проектирования параллельного КИХ-фильтра на четыре отвода:

$$y = C_0 x_0 + C_1 x_1 + C_2 x_2 + C_3 x_3 \quad (2)$$

с использованием мегафункций САПР ПЛИС Quartus II компании Altera [1], объединенных общей идеей использования перемножителей цифровых сигналов и «дерева сумматоров». Коэффициенты фильтра и входные отсчеты возьмем из работы [2]. В [2] рассматривалось проектирование КИХ-фильтра на четыре отвода

с использованием последовательной распределенной арифметики.

Предположим, что коэффициенты фильтра целочисленные, со знаком, известны и равны: $C_0 = -2$, $C_1 = -1$, $C_2 = 7$ и $C_3 = 6$. На вход КИХ-фильтра поступают входные отсчеты -5 , 3 , 1 и 0 . Правильные значения на выходе фильтра: 10 , -1 , -40 , -10 , 26 , 6 и т. д., то есть согласно формуле (2).

Первый вариант

Это параллельная реализация КИХ-фильтра на четыре отвода с помощью четырех блоков умножения с накоплением. В проекте использу-

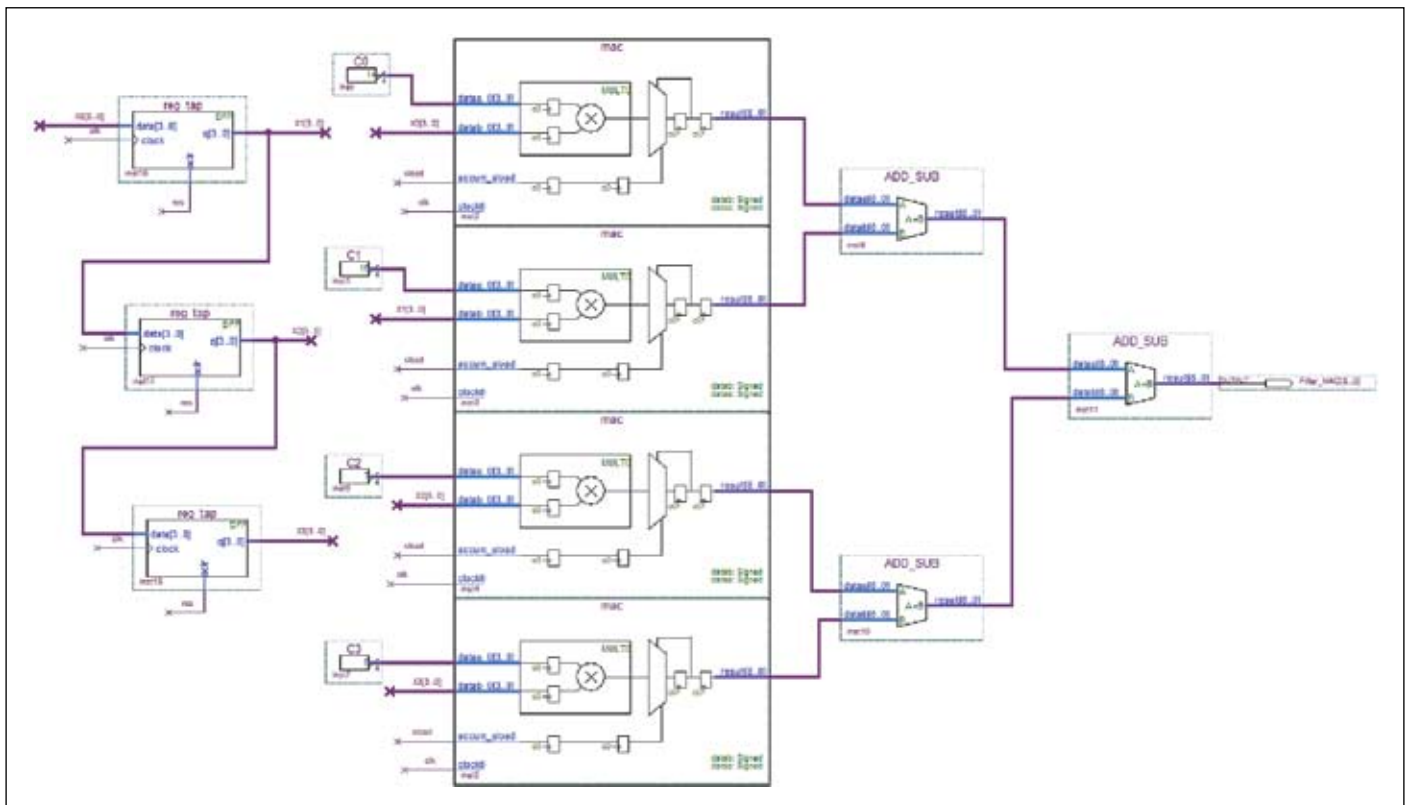


Рис. 1. Параллельная реализация КИХ-фильтра на четыре отвода с использованием четырех блоков в САПР ПЛИС Quartus II (мегафункция `ALTMULT_ACCUM`)

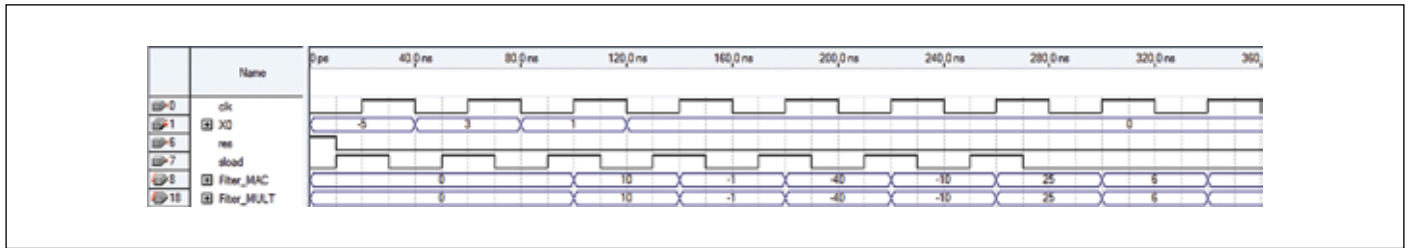


Рис. 3. Временные диаграммы работы параллельных фильтров на четыре отвода с использованием мегафункций ALTMULT_ACCUM и ALTMULT_ADD

ется четыре мегафункции ALTMULT_ACCUM (рис. 1). Каждый блок использует один перемножитель и один сумматор-аккумулятор. Для параллельной реализации фильтра на четыре отвода требуется четыре блока и три дополнительных однотипных многоразрядных сумматора, связанных по принципу «дерево сумматоров». Для того чтобы фильтр работал корректно, необходимо осуществлять синхронную загрузку каждого произведения в каждый сумматор-аккумулятор каждого блока. Для этого служит дополнительный вход мегафункции accum_load. На рис. 1 также показана внешняя линия задержки на четыре отвода из трех 4-разрядных регистров, тактируемых фронтом синхросигнала. Коэффициенты фильтра представляются в двоичном виде с учетом знака числа и загружаются с помощью мегафункции LPM_CONSTANT.

Второй вариант

Параллельная реализация КИХ-фильтра на четыре отвода с помощью четырех перемножителей в блоке на мегафункции ALTMULT_ADD (функция умножения и сложения) в САПР ПЛИС Quartus II показана на рис. 2. В мегафункции ALTMULT_ADD используется три встроенных сумматора. Профильтрованные значения приведены на рис. 3. Сравнивая временные диаграммы, видим, что профильтрованные значения на выходе у двух фильтров, построенных на разных мегафункциях, совпадают.

Значительно упростить разработку КИХ-фильтра позволяет иное использование мегафункции ALTMULT_ACCUM (модификация первого варианта). Фактически это одна мегафункция (один блок с четырьмя перемножителями), в которой линия задержки организована на внутренних регистрах четырех перемножителей. В мегафункции используются встроенные два сумматора и один сумматор-аккумулятор (рис. 4). Временные

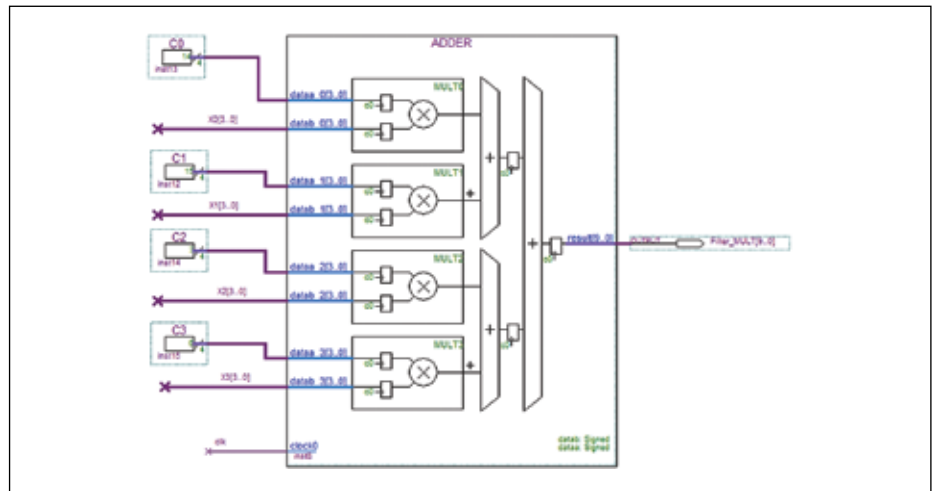


Рис. 2. Параллельная реализация КИХ-фильтра на четыре отвода с использованием четырех перемножителей в блоке (мегафункция ALTMULT_ADD). Линия задержки такая же, как и на рис. 1

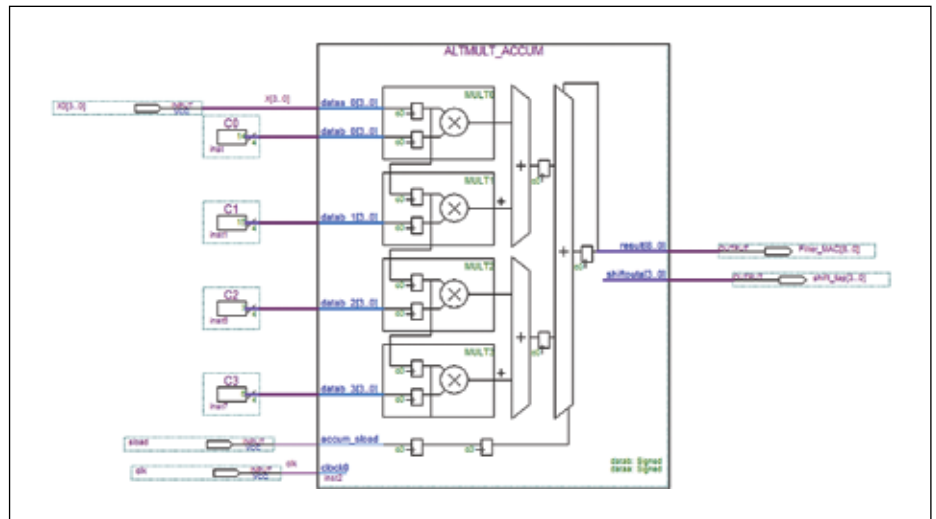


Рис. 4. Параллельная реализация КИХ-фильтра на четыре отвода с использованием одного блока с четырьмя перемножителями в САПР ПЛИС Quartus II (мегафункция ALTMULT_ACCUM). Линия задержки построена на внутренних регистрах перемножителей MULT0-MULT3

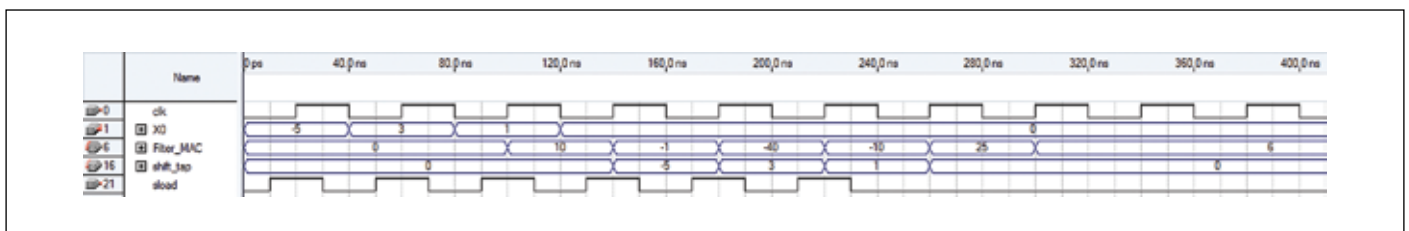


Рис. 5. Временные диаграммы работы фильтра на четыре отвода с использованием мегафункции ALTMULT_ACCUM

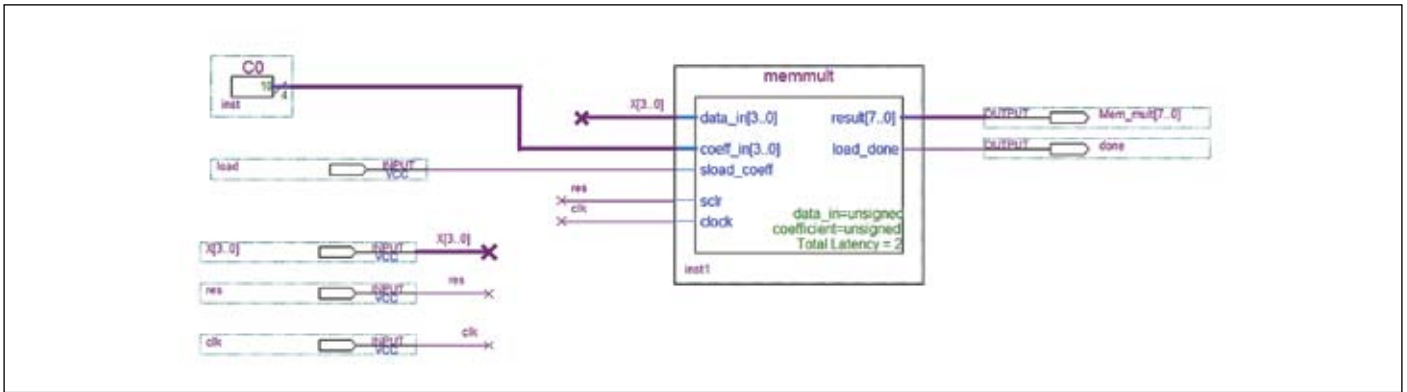


Рис. 6. Умножение 11 на 10 с помощью мегафункции ALTMEMMULT

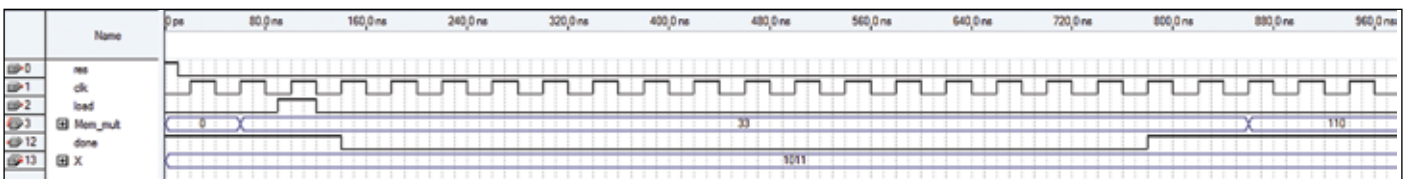


Рис. 7. Временные диаграммы умножения 11 на 10. По умолчанию в мегафункцию загружена константа 3. Результат — 110

диаграммы, показанные на рис. 5, не отличаются от диаграмм на рис. 3.

Рассмотрим умножение десятичного числа 11 на 10 на примере мегафункции ALTMEMMULT (рис. 6). Мегафункция ALTMEMMULT предназначена для умножения числа на константу, которая хранится в блочной памяти ПЛИС (M512, M4K, M9K и MLAB-блоки), обеспечивая наивысшее

быстродействие, лимитируемое латентностью. Однако константу можно загрузить и из внешнего порта. Считаем, что 10 — это константа, которая загружается из внешнего порта. По умолчанию в мегафункцию загружена, например, константа 3. Латентность мегафункции — 2, то есть доступность результата умножения числа на константу, если константа хранится в памяти мегафункции,

возможна уже после двух синхроимпульсов (высокий уровень сигнала done, соответствующий порту load_done). Число «3», загруженное в мегафункцию по умолчанию, умноженное на число 11 с входного порта data_in [3..0], дает результат — 33.

Далее синхронный сигнал загрузки load (порт load_coeff) разрешает загрузку числа 10 в перемножитель. Низкий уровень сиг-

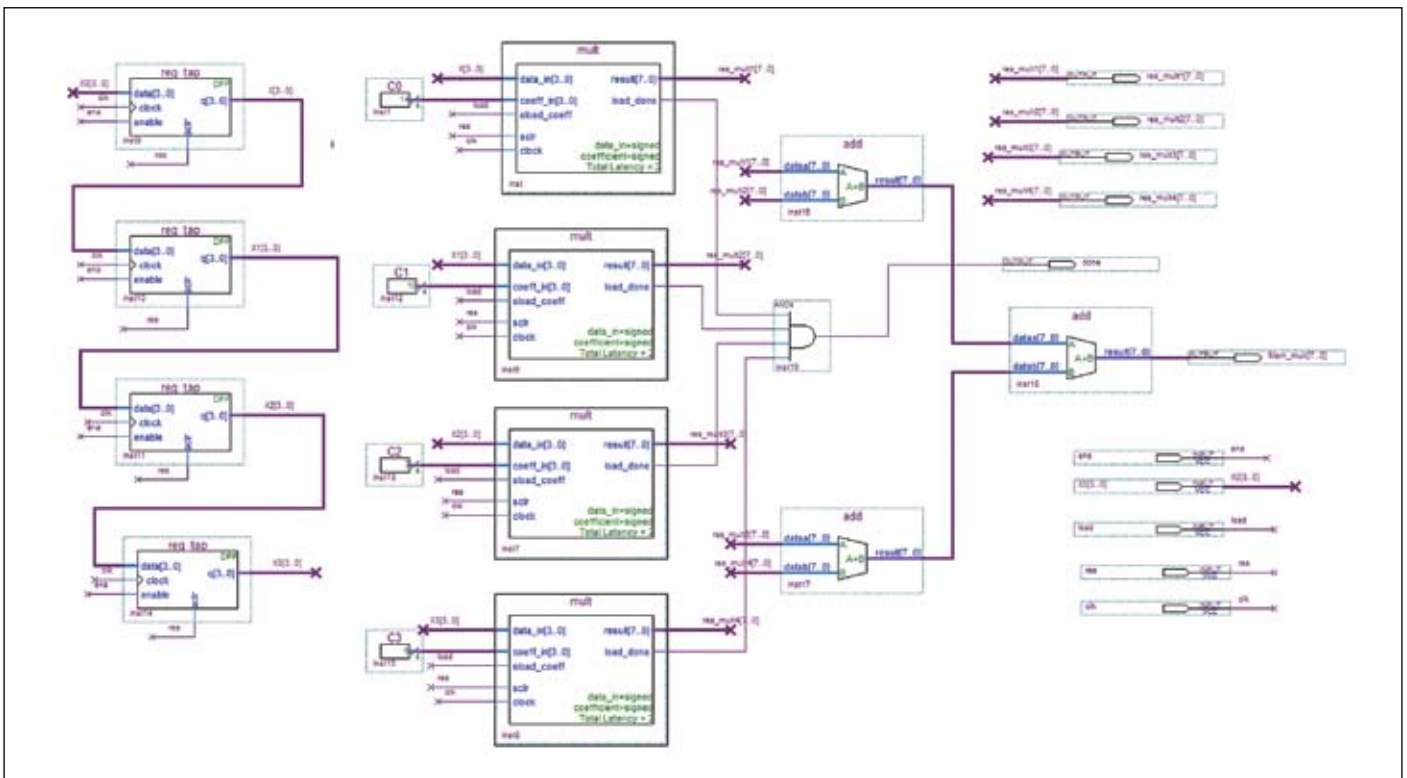


Рис. 8. Параллельная реализация КИХ-фильтра на четыре отвода с использованием четырех умножителей в САПР ПЛИС Quartus II (мегафункция ALTMEMMULT). Линия задержки построена на четырех регистрах, коэффициенты фильтра загружаются из внешнего порта

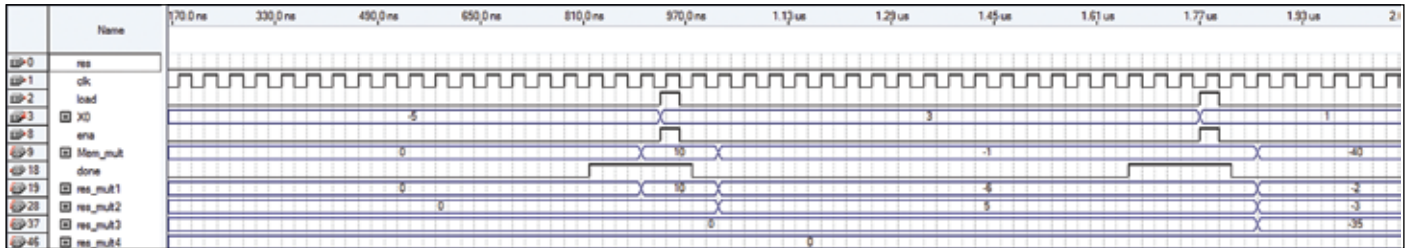


Рис. 9. Временные диаграммы работы фильтра на четыре отвода с использованием мегафункции ALTMEMMULT. Коэффициенты фильтра загружаются из внешнего порта

нала done в течение 16 тактов синхрочастоты говорит о том, что идет процесс умножения. И лишь спустя два синхроимпульса при высоком уровне сигнала done на выходе появляется требуемое число — 110. Таким образом, процесс умножения составляет 20 синхроимпульсов от момента появления сигнала load (рис. 7).

Третий вариант

Применяя мегафункцию ALTMEMMULT, разработаем параллельную реализацию КИХ-фильтра на четыре отвода с исполь-

зованием четырех умножителей (четыре блока по одному умножителю в каждом) в САПР ПЛИС Quartus II и «дерева сумматоров» (рис. 8). Внешняя линия задержки состоит не из трех регистров, как в первых двух вариантах, а из четырех регистров. Дополнительно требуется, как и в первом варианте, три однотипных многоразрядных сумматора. Латентность каждого умножителя — 2. В каждый умножитель по умолчанию загружено число 0. Временные диаграммы работы фильтра на четыре отвода с использованием

мегафункции ALTMEMMULT показаны на рис. 9. Коэффициенты фильтра $C_0 = -2$, $C_1 = -1$, $C_2 = 7$ и $C_3 = 6$ загружаются из внешнего порта. Для этих целей используется мегафункция LPM_CONSTANT.

Рассмотрим вариант, когда коэффициенты фильтра загружаются из блочной памяти в ПЛИС. В мегафункции ALTMEMMULT коэффициенты представляются как целочисленные значения со знаком (рис. 10). Временные диаграммы работы фильтра на четыре отвода с использованием ме-

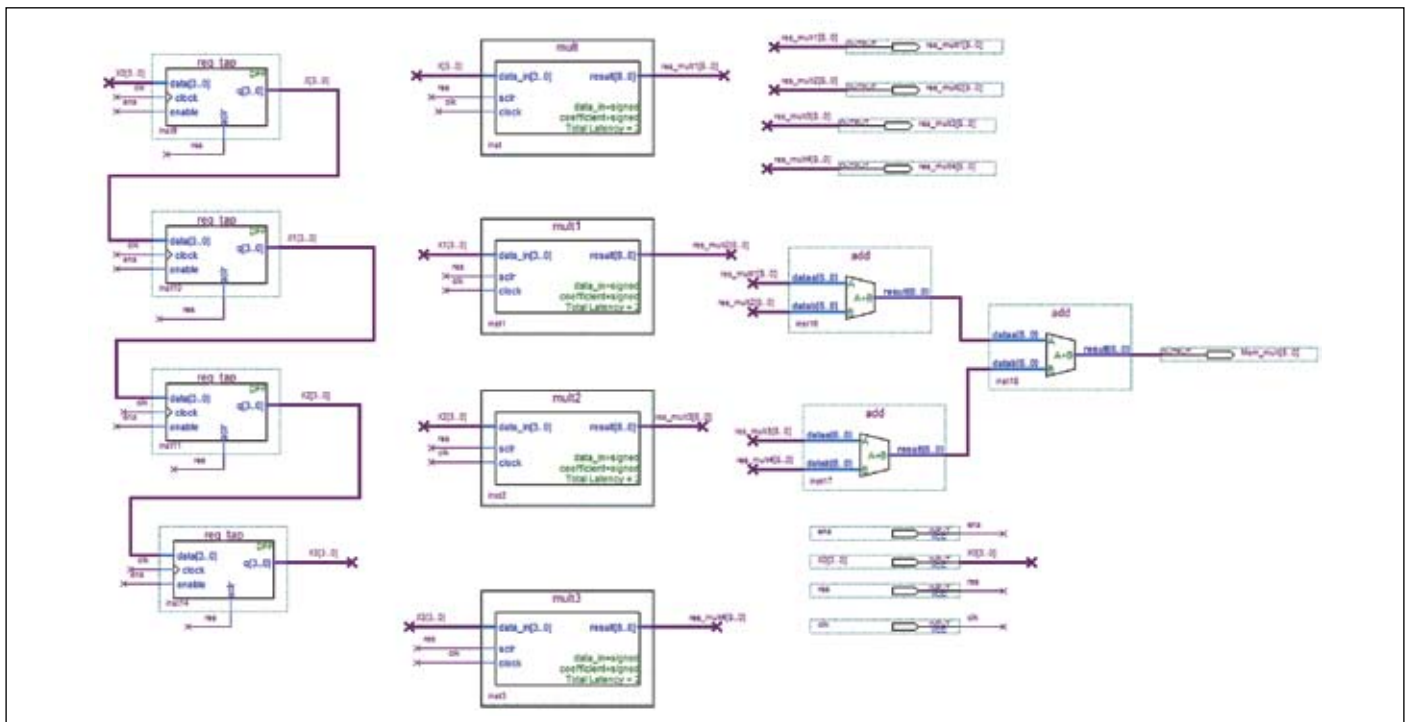


Рис. 10. Параллельная реализация КИХ-фильтра на четыре отвода с использованием четырех умножителей в САПР ПЛИС Quartus II (мегафункция ALTMEMMULT). Линия задержки построена на четырех регистрах, коэффициенты фильтра загружаются из блочной памяти

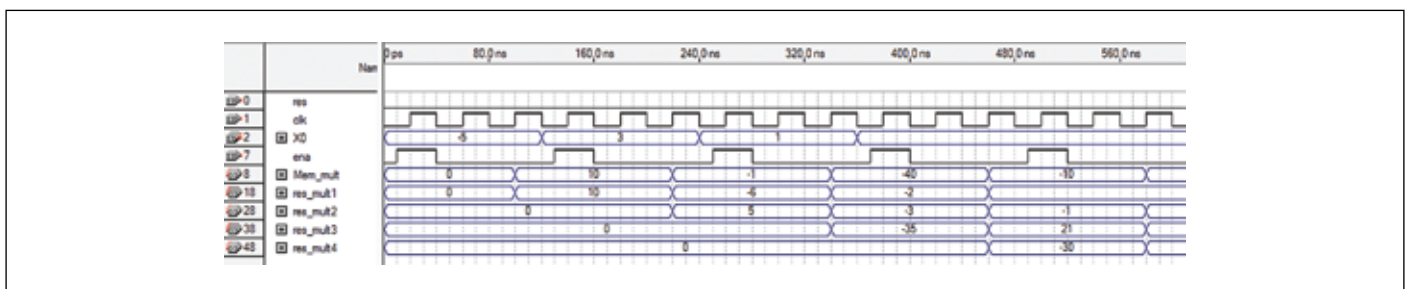


Рис. 11. Временные диаграммы работы фильтра на четыре отвода с использованием мегафункции ALTMEMMULT. Коэффициенты фильтра загружаются из блочной памяти ПЛИС

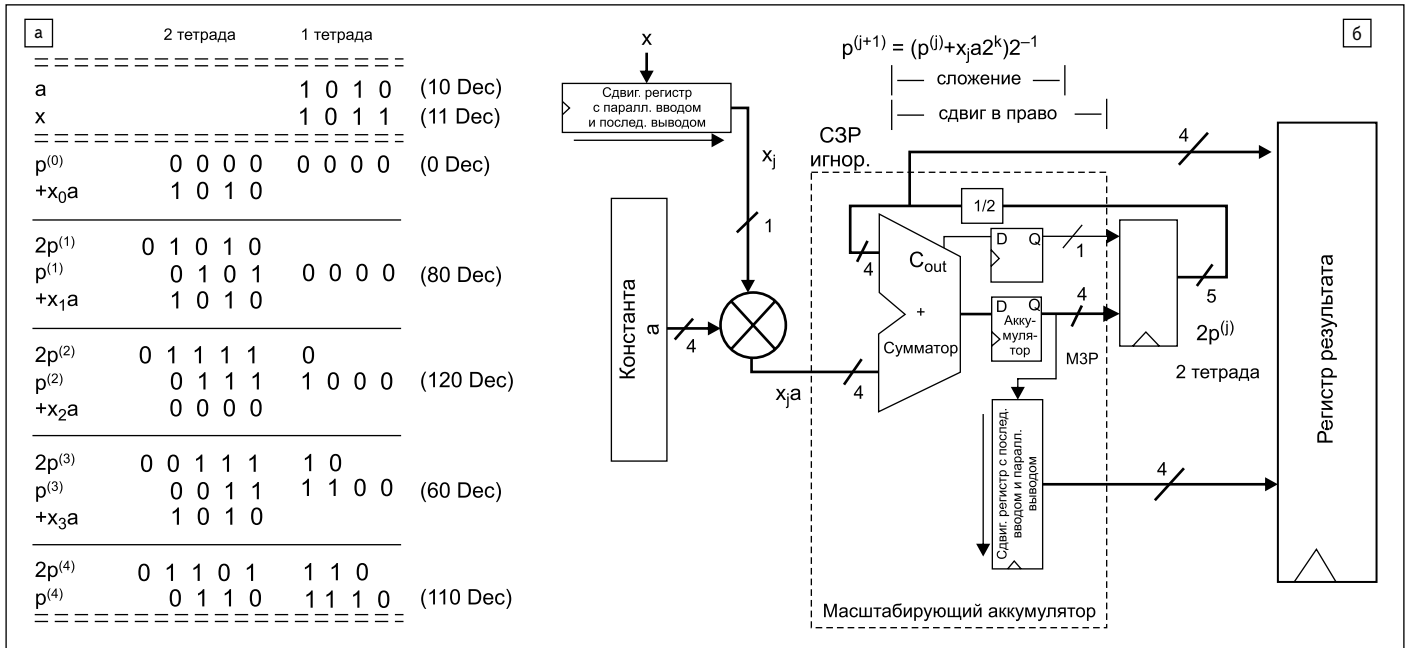


Рис. 12. Идея схемы умножения методом правого сдвига с накоплением: а) алгоритм умножения десятичного числа 11 на 10; б) структурная схема метода

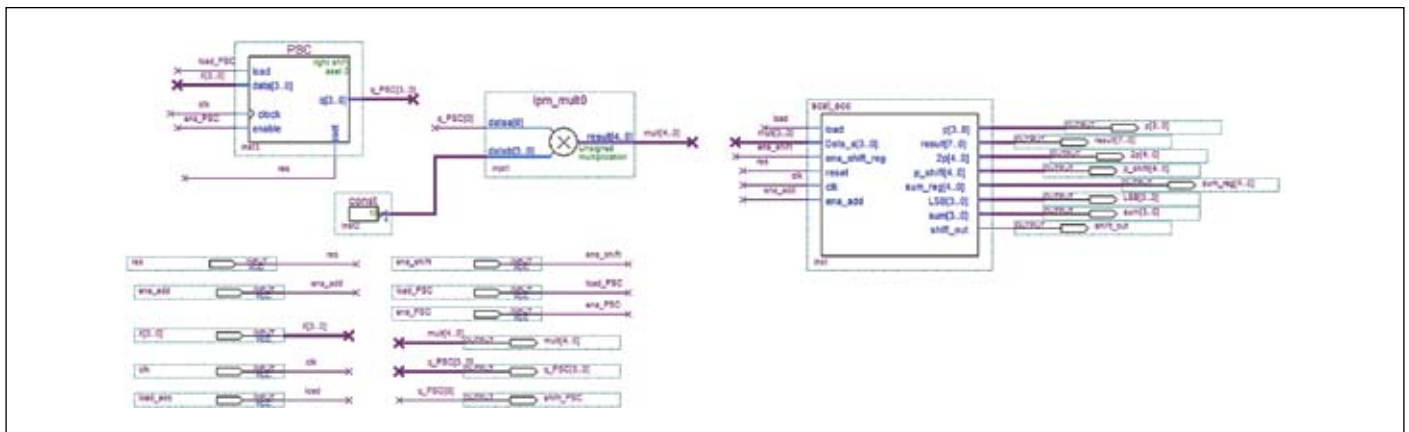


Рис. 13. Схема умножения методом правого сдвига с накоплением

гафункции ALTMEMMULT показаны на рис. 11. Сравнивая рис. 9 и 11, видим, что быстродействие фильтра в этом случае значительно увеличивается за счет хранения коэффициентов в блочной памяти.

Масштабирующий аккумулятор не только входит в состав последовательных КИХ-фильтров, его можно применять и для других целей. Рассмотрим проектирование масштабирующего аккумулятора с помощью метода правого сдвига с накоплением на примере умножения числа 11 на константу 10 (целочисленные беззнаковые значения).

На рис. 12 показана идея схемы умножения методом правого сдвига с накоплением [3]. Для практической реализации в базе ПЛИС потребуется преобразователь параллельного кода в последовательный со сдвигом вправо (мегафункция LPM_SHIFTRREG), умножитель 1-разрядного числа на 4-разрядное (мегафункция LPM_MULT, старший значащий

разряд игнорируется) и масштабирующий аккумулятор (рис. 13).

На рис. 14 показан масштабирующий аккумулятор, который состоит из 4-разрядного

сумматора (мегафункция LPM_ADD_SUB), сдвиговых регистров lpm_shiftreg0 и shift_LSB, вспомогательного регистра ff5 и регистра результата ff8. Регистр ff5 объединяет

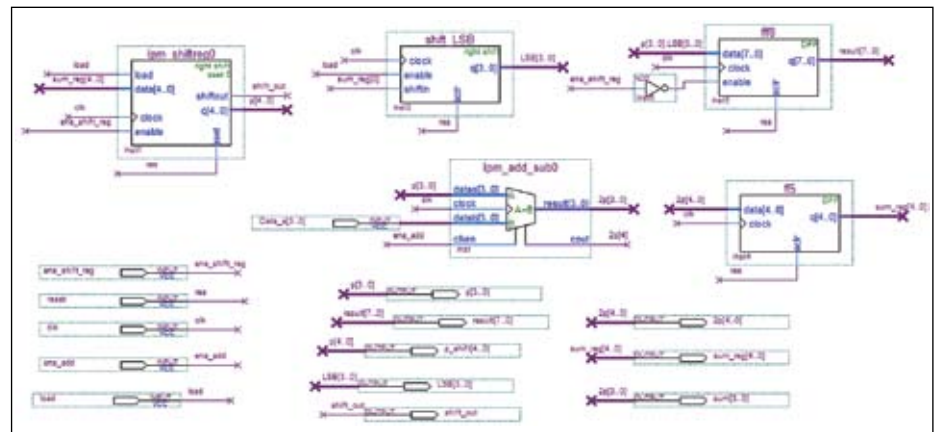


Рис. 14. Масштабирующий аккумулятор

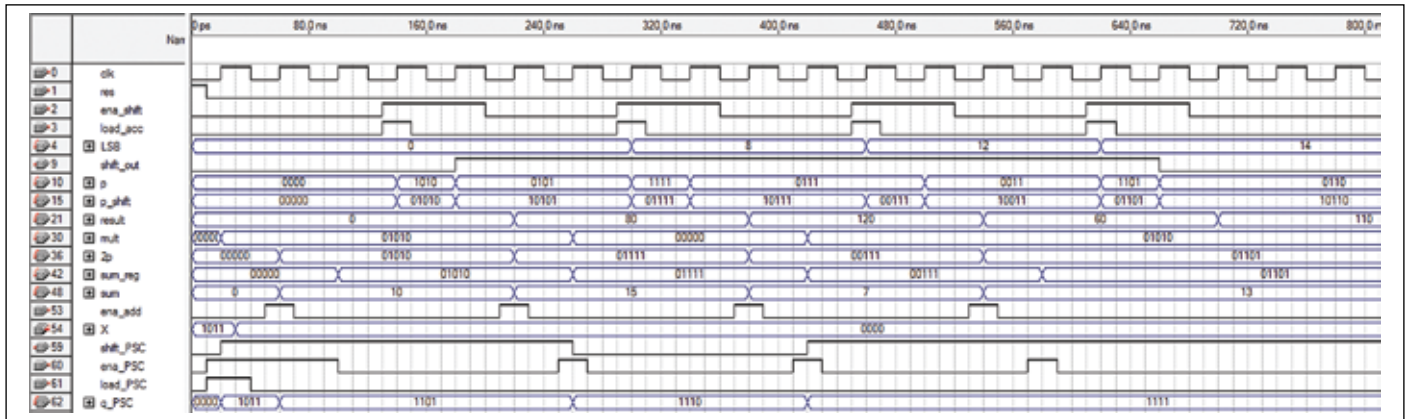


Рис. 15. Временные диаграммы умножения числа 11 на 10 методом правого сдвига с накоплением. Результат — 110

результат суммирования $2p[3..0]$ и разряд переноса Cout сумматора. Младший значащий разряд накапливается в регистре shift_LSB. Полученный 5-разрядный сигнал сдвигается вправо на один разряд с помощью сдвигового регистра lpm_shiftreg0, при этом старший значащий разряд отбрасывается, а оставшиеся, более младшие разряды вновь поступают на один из входов сумматора. В регистре ff8 происходит сохранение результата преобразования, который представляет собой 8-разрядный сигнал, полученный объединением сигналов $p[3..0]$ и LSB $[3..0]$.

На рис. 15 показаны временные диаграммы умножения числа 11 на 10. Результат — 110. Сравнивая результат умножения числа 11 на 10 с помощью мегафункции ALTMEMMULT, показанный на рис. 7 и 15, с использованием метода правого сдвига с накоплением, видим, что для умножения во втором случае потребовалось 18 синхроимпульсов. Если в разработанном варианте в качестве перемножителя использовать мультиплексор многоразрядных сигналов, то можно получить бесперебойную схему умножения и построить КИХ-фильтр без операции явного умножения.

Выводы

Рассмотрены различные варианты проектирования параллельных КИХ-фильтров с использованием мегафункций САПР ПЛИС

Таблица. Анализ задействованных ресурсов ПЛИС серии Stratix при реализации параллельных КИХ-фильтров на четыре отвода с использованием мегафункций ALTMULT_ACCUM, ALTMULT_ADD и ALTMEMMULT

Ресурсы ПЛИС серии Stratix	Мегафункция ALTMULT_ACCUM. Четыре блока по одному перемножителю в каждом, внешняя линия задержки из трех регистров / один блок (четыре перемножителя в блоке, встроенная линия задержки)		Мегафункция ALTMULT_ADD. Четыре перемножителя в блоке, внешняя линия задержки из трех регистров	Мегафункция ALTMEMMULT. Четыре блока по одному перемножителю в каждом (коэффициенты фильтра загружаются из внешнего порта)	Мегафункция ALTMEMMULT. Четыре блока по одному перемножителю в каждом (коэффициенты фильтра загружаются из блочной памяти)
	Вариант 1	Модификация варианта 1	Вариант 2	Вариант 3	
Количество АЛМ для выполнения комбинационных функций	18	0	0	181	18
Количество АЛМ с памятью	0	0	0	64	36
АЛМ	18	0	8	152	4
Количество выделенных регистров	12	0	12	248	68
Аппаратные перемножители (ЦСП 18×18)	16	4	4	0	0
Количество АЛМ для выполнения комбинационных функций без использования регистров	18	0	0	49	36
Количество АЛМ под регистрные ресурсы	12	0	12	50	196
Количество АЛМ под комбинационные и регистрные ресурсы	0	0	0	196	18
Рабочая частота в наихудшем случае, МГц	400	400	400	331	400

Примечание. АЛМ — адаптивные логические модули.

Quartus II компании Altera, а также умножение методом правого сдвига с накоплением.

Фильтр на мегафункции ALTMULT_ACCUM (вариант 1) является самым затратным, так как требует 16 аппаратных перемножителей, три дополнительных сумматора и внешнюю линию задержки (таблица).

Наиболее оптимальной по числу ресурсов ПЛИС является модификация первого варианта, которая позволяет построить параллельный КИХ-фильтр на четыре отвода с использованием всего лишь одного блока с четырьмя перемножителями, встроенными двумя сумматорами, сумматором-аккумулятором и линией задержки.

Мегафункции ALTMULT_ADD также позволяет построить параллельный КИХ-фильтр с помощью всего лишь одного блока с четырьмя перемножителями и встроенными тремя сумматорами. Использование внешней линии задержки из трех регистров приводит к незначительному увеличению ресурсов и не сказывается на быстродействии. Экономия ресурсов ПЛИС в первом, модифициро-

ванном и во втором вариантах достигается за счет встроенных четырех аппаратных перемножителей 18×18 .

Фильтр на мегафункции ALTMEMMULT с загрузкой коэффициентов из внешнего порта обладает пониженным быстродействием. Использование же блочной памяти для хранения коэффициентов фильтра внутри ПЛИС значительно упрощает процесс разработки и не приводит к существенному увеличению ресурсов за счет внешней линии задержки на четырех регистрах, дополнительных трех однотипных многоразрядных сумматорах и не снижает быстродействие проекта.

Литература

1. www.altera.com. Integer Arithmetic Megafuncions. UG-01063-3.0. Altera Corporation. Feb. 2012.
2. Строгонов А., Быстрицкий А. КИХ-фильтр на распределенной арифметике: проектируем сами // Компоненты и технологии. 2013. № 3.
3. Computer Arithmetic: Algorithms and Hardware Designs. Oxford U. Press, 2010.