

Возможности FPGA фирмы Xilinx для цифровой обработки сигналов

Илья ТАРАСОВ,
к. т. н.
tile@kc.ru

Задачи цифровой обработки сигналов — самые разнообразные и требуют определенного уровня производительности микроэлектронной элементной базы. Преимущества параллельных вычислений оказываются довольно велики, поэтому для их реализации в последнее время часто применяются ПЛИС с архитектурой FPGA. Ее развитие и появление большого числа дополнительных возможностей — предмет данной статьи. Непосредственным поводом к публикации стало недавнее появление еще одного семейства, Spartan 3A DSP. Кроме того, недавно в дополнение к уже появившимся платформам высокопроизводительного семейства Virtex-5 добавилась еще одна, Virtex-5SXT. Ввиду этого стало возможным провести подробное рассмотрение семейств FPGA Xilinx, ориентированных на цифровую обработку, изучить их особенности и отличия, а также рассмотреть ряд вопросов, касающихся применения ПЛИС для задач повышенной производительности.

Задачи цифровой обработки сигналов (DSP)

Термин «цифровая обработка сигналов» (DSP — Digital Signal Processing) охватывает довольно широкую область К классу DSP (сигнальных процессоров) можно с уверенностью отнести ряд процессоров. Часто отличительной чертой таких устройств считают наличие в них аппаратной поддержки операции «умножение с накоплением» (MAC,

Multiply and Accumulate). Чем же важна эта операция, и почему она ассоциируется с цифровой обработкой сигналов? Само по себе умножение с накоплением сводится к вычислению суммы произведений вида:

$$y = \sum_{i=0}^N k_i \times x_i.$$

Эта сумма произведений приближенно может считаться равной интегралу:

$$\int k(t) \times x(t) dt,$$

к которому, в свою очередь, сводится большой набор математических методов анализа сигналов. Наиболее известным из них является преобразование Фурье. Оно состоит в вычислении интеграла от произведения исследуемого сигнала $x(t)$ и гармонической функции:

$$\dot{X}(\omega) = \int x(t)(\cos(\omega t) + j \times \sin(\omega t)) dt.$$

В данном случае используется комплексное представление, которое раскладывается на вещественную и мнимую части. Заменяя интегрирование суммированием, мы получаем дискретное преобразование Фурье (ДПФ), которое широко используется для анализа и преобразования переменных сигналов в самых разных областях. Пример результатов преобразования Фурье показан на рис. 1. В качестве исследуемого сигнала был взят синусоидальный сигнал, частота которого линейно изменялась от 50 до 70 Гц. Нетрудно убедиться, что на графике именно в этой области находятся наибольшие значения результатов преобразования. Это позволяет, например, точно определять частоту входного сигнала в условиях помех, характерные частоты самих помех, регулировать тембр звуковых сигналов и решать многие другие задачи.

Получение каждой точки представленного графика сопряжено с необходимостью вычислить сумму произведений значений сигнала

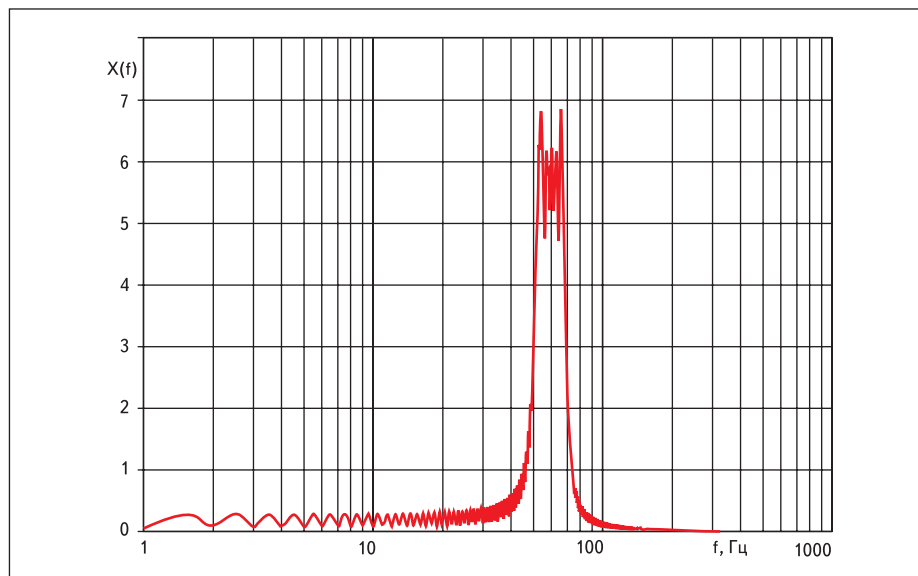


Рис. 1. Результаты Фурье-анализа некоторого сигнала, частота которого линейно изменяется от 50 до 70 Гц

и гармонической функции. Чем большим количеством отсчетов представлены сигнал и функция, тем больше точность. Поэтому, при «методе грубой силы» получения графика, подобного представленному на рис. 1, может потребоваться несколько миллионов операций умножения и столько же — сложения.

Для преобразования Фурье существует также эффективная методика, позволяющая кардинально уменьшить объем вычислений. Она называется быстрым преобразованием Фурье (БПФ, или Fast Fourier Transform, FFT). Эта методика подробно описана в специальной литературе, а главным ее достоинством является то, что вместо объема вычислений, пропорционального N^2 , он становится пропорционален $N \times \log_2 N$, что особенно эффективно для подробного анализа с использованием большого числа частотных каналов. Некоторым недостатком БПФ является фиксированная сетка частот, привязанная к частоте, с которой берутся входные отсчеты. Однако популярность БПФ все же весьма велика, и часто производительность сигнальных процессоров при его выполнении приводят как одну из важных технических характеристик. Очевидно, что возрастающие тактовые частоты процессоров позволяют добиваться все большей и большей точности.

Однако с увеличением производительности процессоров стали проявляться и фундаментальные недостатки преобразования Фурье. Еще самому Фурье в начале XIX века указывали на тот факт, что он пытается анализировать конечные по времени сигналы путем представления их в виде суммы бесконечных по времени функций. Необходимость «обрезания» сигнала за пределами интервала интегрирования приводит к тому, что при подробном анализе результирующая функция становится зашумленной, на основные гармоники, соответствующие объективным процессам в исследуемом сигнале, накладываются артефакты. Кроме того, поскольку исследуемый сигнал также представлен в цифровом виде, то есть дискретен и по времени, и по амплитуде, он представляет собой не плавно меняющуюся функцию, а серию микроскопических ступенек. Чем сильнее эффекты дискретизации входного сигнала, тем сильнее эффект Гиббса, заключающийся в возникновении локальных экстремумов на основном графике. Все эти эффекты наглядно видны на рис. 1, график на котором теоретически должен представлять собой прямоугольник, ограниченный частотами 50 и 70 Гц. Наличие артефактов затрудняет получение точного значения преобладающей частоты в исследуемом сигнале.

Другой недостаток преобразования Фурье становится понятен, если задать вопрос: а каким образом менялась входная частота? Дело в том, что и при возрастании, и при убывании, и даже при одновременном действии всех частот в диапазоне от 50 до 70 Гц, график на рис. 1 будет одним и тем же. Ины-

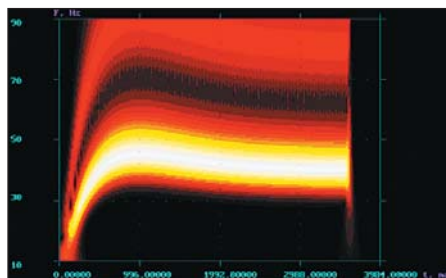


Рис. 2. Результаты вейвлет-анализа сигнала доплеровской частоты

ми словами, преобразование Фурье не позволяет определить закон изменения параметров сигнала во времени. Обе проблемы могут быть достаточно успешно решены путем перехода к частотно-временному анализу, ярким примером которого является модуляция гармонической функции каким-либо окном.

Одним из новейших направлений в теории обработки сигналов является вейвлет-анализ (wavelet — дословно «маленькая волна»), который предусматривает представление исследуемого сигнала в виде суммы конечных по времени функций («маленьких волн», или, в русскоязычной литературе, всплесков). Поскольку вейвлет-функции ограничены по времени, для каждого из отрезков, на котором действует входной сигнал, необходимо вычислить свой набор преобразований. Таким образом, получаемый график становится трехмерным, то есть вейвлет-анализ заключается в вычислении функции двух аргументов: времени и частоты. Пример анализа переменного сигнала с помощью вейвлет-функции представлен на рис. 2, где виден характер изменения частоты и амплитуды основной гармоники. Кроме того, подбор вейвлет-функции позволил существенно уменьшить амплитуду локальных экстремумов, в результате чего стало возможным определение основной гармоники с высокой точностью.

Однако далеко не все вейвлет-функции допускают использование приема, аналогичного быстрому преобразованию Фурье. Говоря точнее, самые интересные в практическом смысле функции такого упрощения как раз не допускают. Поэтому для получения высокой точности вейвлет-анализа требуется выполнение огромного количества операций умножения и сложения. Для рис. 2 число выполненных операций составляет приблизительно 30 млрд.

Гораздо более распространенным и наглядным примером применения вейвлетов являются форматы JPEG2000 и MPEG. Именно за счет применения тщательно подобранных вейвлет-функций стала возможной высокая степень сжатия изображения и звука. Эффект заключается в том, что при передаче результатов выполнения вейвлет-анализа можно быстро восстановить исходный сигнал, пользуясь только небольшим количеством переданных коэффициентов. Ближайшей анало-

гией является MIDI-файл, который кодирует только ноту, ее длительность и используемый инструмент. Передача этих параметров позволяет «проиграть» указанную ноту со всеми особенностями конкретного инструмента. Естественно, для этого требуется знать форму сигнала для каждого из них. Соответственно, если программа-декодер имеет информацию о том, какие вейвлет-функции использовались для сжатия сигнала, получение ею набора коэффициентов приведет к «проигрыванию» запомнившихся вейвлетов. Поскольку они подобраны так, чтобы формировать характерные особенности звука и видео, уже после небольшого числа шагов восстановления звук и изображение будут мало отличаться от исходного.

Широко распространена цифровая фильтрация, которая также сводится к вычислению суммы произведений входных отсчетов, помноженных на набор постоянных коэффициентов. Существует огромное множество фильтров, которые могут подавлять все частоты, большие или меньшие некоторого значения (фильтры низких или высоких частот), пропускающие или подавляющие сигнал в определенной полосе, производящие коррекцию амплитуды по сложному закону и т. п. Такие фильтры широко используются в аудиоаппаратуре, системах связи и коммуникации и т. д. Не заостряя внимания на теоретических вопросах, можно заметить, что и здесь вычислительной основой являются две операции: вычисление произведений пар значений и суммирование этих произведений.

Таким образом, нетрудно убедиться, что в основе целой группы методов обработки сигналов лежит повторение пары операций: умножения и суммирования (в действительности — накопления, поскольку требуется не просто сложение пар чисел, а вычисление суммы всех произведений). Очевидно, что наибольшую производительность при использовании алгоритмов такого типа будут иметь те процессоры, которые способны выполнить умножение и накопление за возможно меньшее число тактов. Идеальным вариантом является одноктактное исполнение. Далее мы рассмотрим, каким образом эти операции реализуются в ПЛИС.

Новые семейства ПЛИС Xilinx

В начале 2007 года были анонсированы две новые разновидности уже имеющихся на рынке семейств Virtex-5 и Spartan-3. Оба семейства ориентированы на интенсивную цифровую обработку сигналов, поэтому есть возможность рассмотреть их в сравнении с семейством Virtex-4SX.

Технические характеристики устройств FPGA Xilinx, ориентированных на цифровую обработку сигналов, приведены в таблице 1.

Чем интересна информация, приведенная в таблице? Прежде всего, в семействе Spartan-3A DSP появились секции DSP (а не просто

Таблица 1. Технические характеристики устройств FPGA Xilinx, ориентированных на цифровую обработку сигналов

Показатели	Типы устройств							
	Spartan-DSP		Virtex-DSP					
	Spartan-3A DSP		Virtex-5SX			Virtex-4SX		
	3SD1800A	3SD3400A	5VSX35T	5VSX50T	5VSX95T	4VSX25	4VSX35	4VSX55
Производительность, ГМАС/с	21	32	106	158	352	64	96	256
Максимальная полоса пропускания блочной памяти, Гбит/с	1512	2268	6653	10454	19325	4608	6912	11520
Максимальная частота блоков DSP	250	250	550	550	550	500	500	500
Секции DSP48	84	126	192	288	640	128	192	512
Блочная память, кбит	1512	2268	3024	4752	8784	2304	3456	5760
Логические ячейки	37 440	53 712	34 816	52 224	94 208	23 040	34 560	55 296
Интерфейсы	622 Мбит/с (LVDS)		1,25 Гбит/с (LVDS), 3,2 Гбит/с (приемопередатчики)			1 Гбит/с (LVDS)		

умножители, как в прочих low-cost FPGA). Польза данного нововведения, может быть, не очевидна, однако эффект от него весьма и весьма интересен. Для пояснения следует сделать некоторое отступление.

Известно, что основные арифметические операции (четыре действия, которые в начальной школе достаточно равномерно используются в математических примерах) совершенно неравнозначны при их аппаратном исполнении. Если сумматор и вычитатель реализуются приемлемым объемом цифровых устройств, то операция умножения занимает существенно большую площадь кристалла (или приходится выполнять умножение за несколько тактов, по алгоритму «сдвиг с накоплением»). Еще сложнее реализуется операция деления. Таким образом, если для реализации обработки сигналов требуется частое выполнение операции умножения, программируемые ресурсы ПЛИС используются не вполне рационально (поскольку далеко не все оказываются задействованы). Поэтому эффективнее оказалось разместить на кристалле ПЛИС аппаратные блоки умножителей. Поскольку эти блоки выполнены обычными металлическими соединениями и не содержат лишних компонентов, их тактовая частота оказывается весьма высокой, а площадь, занимаемая на кристалле, — гораздо меньше той, которую заняли бы логические ячейки, выполняющие такое умножение. Для ПЛИС фирмы Xilinx семейством, где впервые появились выделенные аппаратные умножители 18-разрядных операндов, стало Virtex-II. Такое решение резко увеличило привлекательность ПЛИС в цифровой обработке сигналов (речь об этом пойдет ниже).

Однако возникла следующая, весьма оригинальная проблема. Если ранее считалось, что сложение выполняется гораздо быстрее и проще, чем умножение (и ряд алгоритмов цифровой обработки специально смещают баланс этих операций в сторону увеличения сложности при уменьшении умножений), то аппаратная реализация умножения буквально перевернула эту ситуацию. Сумматор, выполняемый на базе более медленных логических ячеек, внезапно оказался «слабым звеном» в цепи обработки сигналов! Реше-

ние оказалось очевидным — в следующем поколении FPGA блок умножения был заменен специальной «секцией DSP», которая представляла собой умножитель 18-разрядных операндов, совмещенный с 48-разрядным аккумулятором. Результат умножения двух 18-разрядных чисел способен дать 36-битное произведение, а дополнительные 12 разрядов оставлены как резерв на последовательное накопление суммы (поскольку при суммировании разрядность также может возрасти). Частота работы такого блока достаточно высока, а главное, что она оказывается той же при использовании стандартных технических решений, характерных для DSP. Например, блочная память, подающая коэффициенты последовательно на вход секции DSP, также может работать с максимальной тактовой частотой (табл. 1). При этом содержимым памяти могут быть как коэффициенты цифрового фильтра, так и записанные в FIFO входные отсчеты (что обеспечивает максимально быструю их обработку).

Итак, несмотря на то, что семейство Spartan-3 было выполнено с теми же технологическими нормами, что и Virtex-4, оно имело архитектуру ресурсов DSP, подобную семейству Virtex-II, то есть обычные умножители без аккумуляторов. Это обеспечивало приемлемую, однако не максимально возможную производительность. Такое решение в целом понятно, если учесть, что для достижения предельных характеристик больше подходит high-end Virtex-5, а дешевое семейство Spartan-3 имеет умножители скорее как результат тенденции к экономии ресурсов. Если сравнить такие решения с подобной low-cost продукцией других производителей, они окажутся аналогичными.

Тем не менее, для того чтобы привести элементы скоростной потоковой цифровой обработки сигналов в устройства массового сегмента рынка, было выпущено семейство Spartan-3A DSP. Оно в настоящий момент включает в себя всего две микросхемы (достаточно большого логического объема), которые по остальным характеристикам соответствуют семейству Spartan-3A в целом.

Основные характеристики семейства Virtex-5 уже рассматривались в предыдущих пуб-

ликациях в «КиТ». Можно отметить, что Virtex-5SXT, так же как и Virtex-4SX, обладают «повышенной концентрацией» ресурсов DSP. Кроме того, важным является то, что в семействе Virtex-5 в DSP-ориентированные ПЛИС добавлены скоростные приемопередатчики (transceivers), о чем говорит буква T в обозначении. Это мелкое, на первый взгляд, дополнение влечет за собой интересные и полезные для пользователя последствия. Анализируя технические характеристики устройств Virtex-4SX, можно было убедиться, что в предельном варианте они способны обеспечить громадный объем операций на одном кристалле (256 млрд умножений с накоплением, каждое из которых дает 36-битный результат, то есть более 1 Тбайт в секунду!). Возникает закономерный вопрос: каким образом можно передать и принять такой объем данных?

Выше было показано, что часть алгоритмов DSP подразумевает многоканальную обработку, при которой входящий поток данных обрабатывается одновременно несколькими исполнительными устройствами. Результатом выполнения множества операций тоже вполне может быть одно-единственное число. Однако это не снижает общей актуальности проблемы, заставляя ограничиваться теми алгоритмами, которые при большом количестве операций внутри кристалла имеют относительно небольшой входной и выходной потоки данных.

Одним из вариантов решения проблемы (не кардинального, но качественно иного) является введение на кристалл аппаратных приемопередатчиков. В семействе Virtex-4 такие приемопередатчики ставились только в «полнофункциональные» (full-featured) устройства FX. Однако возможность обеспечить большой поток передаваемых данных (к тому же через различные варианты интерфейсов, в том числе Ethernet) оказалась достаточно привлекательной, и в семействе Virtex-5 только платформа LX существует в варианте без приемопередатчиков (впрочем, есть и Virtex-5LXT). Платформы же SXT и пока неанонсированная FXT содержат высокоскоростные приемопередатчики, в список поддерживаемых стандартов которых добавили PCI-Express. Таким образом, не только обрабатывать, но и доставлять большие массивы данных в FPGA стало гораздо удобнее и проще.

Преимущества FPGA при реализации алгоритмов DSP

Практически с момента своего появления FPGA позиционировались как устройства, превосходящие сигнальные процессоры по соотношению производительность/цена. В то же время известно, что программируемые микросхемы не могут конкурировать с ASIC по цене и тактовой частоте. В чем же заключается преимущество ПЛИС и какие факторы являются при этом определяющими?

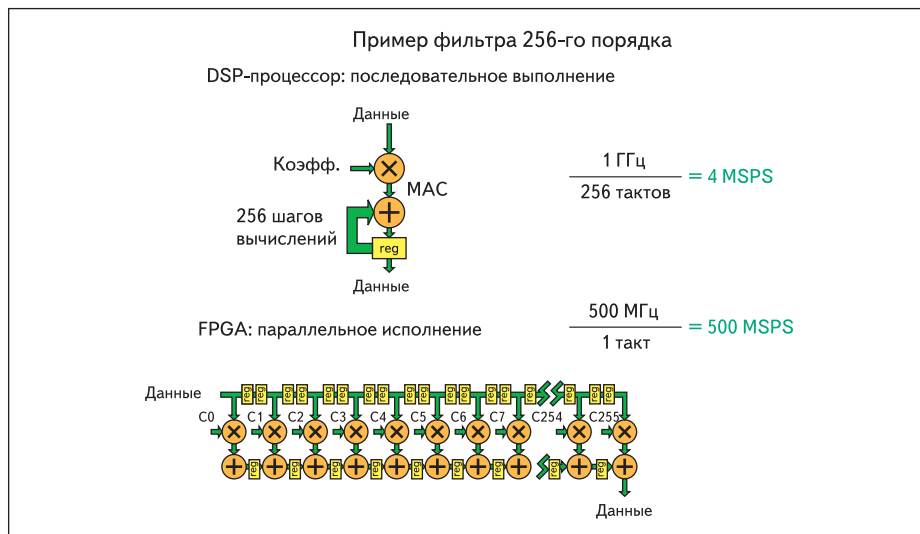


Рис. 3. Выполнение цифровой фильтрации в сигнальном процессоре и FPGA

На рис. 3 проиллюстрированы различия между сигнальным процессором и FPGA при выполнении операции цифровой фильтрации.

На этом рисунке видно, что, хотя сигнальный процессор, созданный по сопоставимой технологии, в среднем обладает более высокой тактовой частотой, единственный поток исполнения команд обуславливает резкое уменьшение общего числа операций (если под «операцией» понимать всю последовательность умножений и сложений, требующуюся для реализации всего фильтра или иного алгоритма). Несмотря на то, что некоторые сигнальные процессоры допускают выполнение 2 или 4 операций «умножения с накоплением» одновременно, для фильтров высокого порядка общее падение частоты оказывается достаточно существенным. В то же время FPGA вполне могут обеспечить однократное исполнение всех операций, используя параллельный расчет. Этому способствует, очевидно, наличие большого числа блоков DSP, выполняющих умножение с накоплением. Нетрудно заметить, что для эффективного использования этого преимущества следует ориентироваться на алгоритмы и методы, подразумевающие распараллеливание операций — фильтры высоких порядков, быстрое преобразование Фурье, вейвлет-анализ и т. п.

Кроме того, важным преимуществом FPGA является их способность обеспечивать не просто высокую скорость обработки, но еще и непрерывную обработку и стабильную скорость. Дело в том, что понятие «пиковая производительность» приобретает несколько разный оттенок при использовании сигнальных процессоров и FPGA. В случае с сигнальным процессором тактовая частота, показанная на рис. 3 как 1 ГГц, достаточно условно соответствует количеству операций непосредственно с фильтром. В программе процессора могут быть предусмотрены и другие действия, например обработка прерываний. Кроме то-

го, при соответствующем построении системы пиковая скорость может упасть из-за промаха кэша и множества других факторов. Иными словами, понятие «пиковая производительность» имеет статистический характер, а реальное значение производительности может меняться не только в зависимости от выбранного алгоритма, но еще и в процессе работы программы при возникновении соответствующих условий.

В то же время для FPGA термин «максимальная тактовая частота» относится к наиболее выгодным условиям трассировки кристалла — все соединения выполнены с использованием коротких цепей, связанные программируемые ячейки расположены рядом, максимальная длина цепей ускоренного переноса ограничена (то есть максимальная разрядность счетчиков невелика — например для Virtex-4 допускается 14-разрядный счетчик). Неудачная трассировка соответственно снижает допустимую тактовую частоту, однако весьма важно то, что после завершения проектирования она остается постоянной! Разумеется, и в проекте на базе FPGA тоже можно использовать внешнюю память, которая внесет проблемы с дополнительными тактами ожидания, как и в сигнальном процессоре, однако наличие скоростных синхронных ресурсов и достаточного количества блочной памяти существенно облегчает построение стандартных узлов цифровой обработки сигналов. Кроме того, что также важно, в FPGA тоже можно реализовать процессоры (например процессор на логических ячейках типа MicroBlaze), однако через этот процессор совершенно необязательно пропускать весь поток обрабатываемых данных. Более того, рекомендуется реализовывать высокопроизводительную цифровую обработку с использованием ресурсов DSP, работающих без непосредственного и постоянного контроля процессора. Процессорное ядро может выполнять организацию интерфейса, на-

стройку, мониторинг, загрузку коэффициентов и прочие операции, реализовывать которые аппаратно чересчур сложно. При этом единственное процессорное ядро может обеспечивать управление несколькими сотнями DSP-блоков ПЛИС, которые постоянно выполняют обработку входящего потока даже без участия процессора.

Ресурсы FPGA для выполнения операций умножения и умножения с накоплением

Развитие архитектуры FPGA связано с постоянным добавлением устройств, облегчающих выполнение операций цифровой обработки сигналов. В первую очередь это касается умножения. На рис. 4 показано одно из технических решений — вентиль MULT_AND. Его назначением является уменьшение объема логических ячеек, требующихся для реализации умножения «в столбик» — путем сдвига и сложения. Вкратце напомним, что умножение двоичных чисел в столбик ничем не отличается от умножения десятичных чисел, однако в силу того, что в двоичном представлении имеются только символы 0 и 1, возникают лишь четыре варианта: $0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$ и $1 \times 1 = 1$. Эти умножения на каждый из разрядов по очереди образуют промежуточные суммы, которые затем подвергаются суммированию (с предварительным сдвигом, что легко проследить, выполнив умножение в столбик на бумаге). Нетрудно видеть, что умножение однобитных двоичных чисел реализуется вентилем 2И. Однако сложность в том, что при реализации умножения на логических ячейках первый их ряд оказывается задействованным именно для умножения (хотя 4-входовая LUT могла бы сделать и больше). Решением этой часто встречающейся задачи и стал специальный вентиль, располагающийся в каждой LUT. Его размеры невелики, а функция встречается часто и существенно повышает производительность ПЛИС, так что такое увеличение сложности логической ячейки в целом вполне оправдано. Наличие вентиля MULT_AND оставляет свободной соответствующую LUT. Важно, что алгоритмы САПР автоматически используют эти ресурсы при реализации блоков умножения. Также интересно, что и в семействах с выделенными ап-

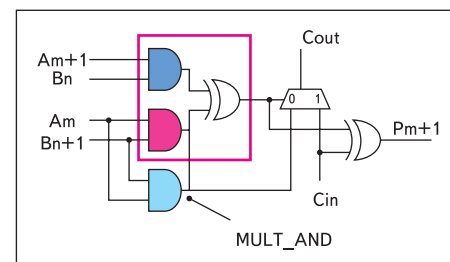


Рис. 4. Вентиль MULT_AND, предназначенный для ускорения операций умножения на логических ячейках

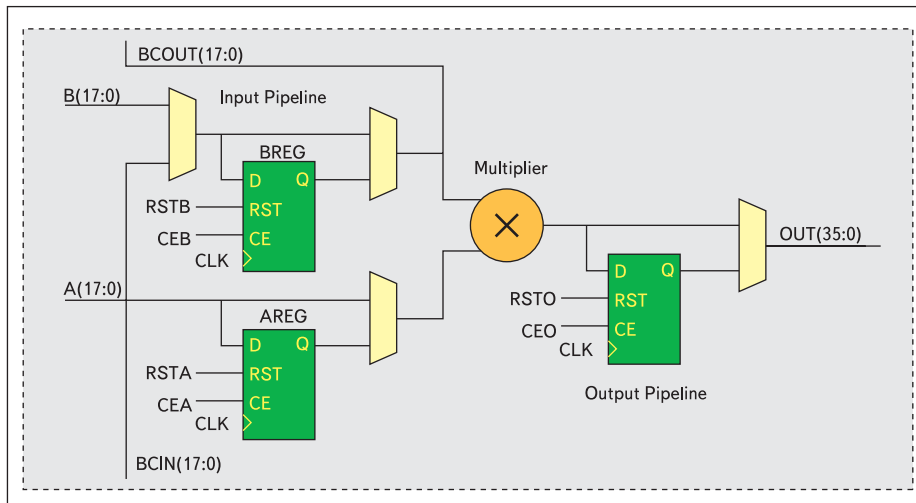


Рис. 5. Блок умножения в FPGA Spartan-3E

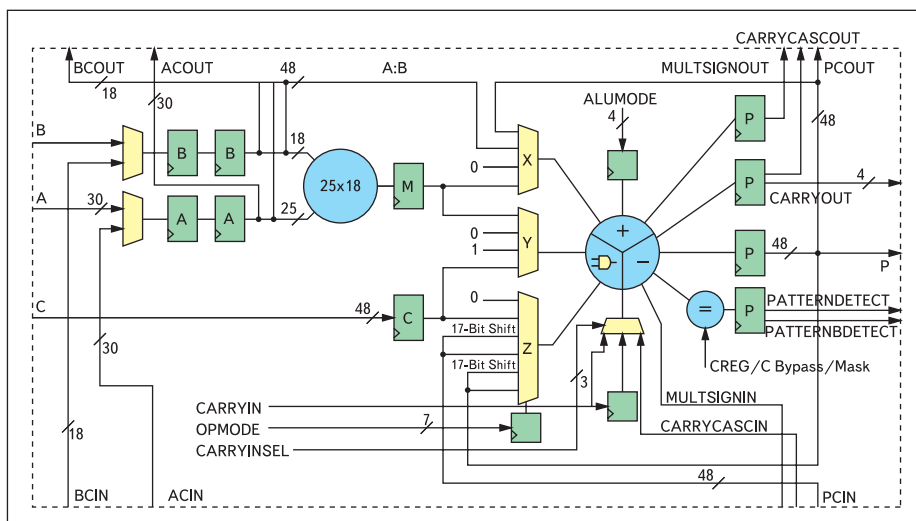


Рис. 6. Секция DSP48E, входящая в состав ПЛИС Virtex-5

паратными умножителями можно добавлять в проект и умножители на ячейках, которые будут иметь меньшую тактовую частоту, но дополнительно увеличивают максималь-

ную производительность кристалла. При этом характеристики ПЛИС в задачах DSP обычно приводятся без учета этого достаточно мощного резерва.

Таблица 2. Основные особенности разновидностей блоков DSP48

Показатели	Типы блоков			Описание
	DSP48 (Virtex-4)	DSP48E (Virtex-5)	DSP48A (Spartan-3A DSP)	
Умножитель	18×18	25×18	18×18	Уменьшает требования к ресурсам ПЛИС при реализации алгоритмов DSP
Предварительный сумматор	—	—	Да	Уменьшает время распространения сигнала во входных цепях при реализации КИХ-фильтров
Каскадированные входы	1	2	1	Организация аккумуляторов разрядностью больше 48
Каскадированные выходы	Да	Да	Да	Организация аккумуляторов разрядностью больше 48
Выделенный вход C	—	Да	Да	Дополнительный, 3-й вход сумматора для реализации более сложных выражений
Сумматор	2-входовый 48 бит	3-входовый 48 бит	2-входовый 48 бит	Аппаратная поддержка операции «умножение с накоплением»
Динамическая установка режимов	Да	Да	Да	Изменение функциональности блока в процессе работы (умножение, умножение с накоплением, суммирование и т. д.)
Логические функции в АЛУ	—	Да	—	Реализация арифметико-логических операций (аналогично процессорному АЛУ)
Детектирование шаблонов	—	Да	—	Аппаратное распознавание переполнения, антипереполнения, окончания счета
Поддержка SIMD в АЛУ	—	Да	—	Организация параллельной обработки нескольких потоков данных
Цепи ускоренного переноса	Вход	Вход и выход	Вход и выход	Передача бита переноса между блоками DSP

Дальнейшим развитием ресурсов цифровой обработки стал блок умножения. На рис. 5 показан его вариант, размещенный в ПЛИС Spartan-3E. По сравнению с семейством Spartan-3 в блок умножения добавлены регистры, показанные на рис. 5 зеленым цветом. Они служат для организации конвейера, так что сигналы поступают на вход умножителя по более коротким проводникам, что позволяет увеличить тактовую частоту. Однако от использования этих регистров можно и отказаться.

На рис. 6 представлен вариант более мощного блока цифровой обработки, DSP48E, который размещен в ПЛИС семейства Virtex-5. Как мы видим, этот блок имеет большое количество дополнительных ресурсов, кроме собственно умножителя и аккумулятора. Имеются как регистры, обеспечивающие опциональную конвейеризацию, так и многочисленные ресурсы, облегчающие каскадирование, то есть наращивание разрядности секций DSP. Режим работы блока может динамически изменяться в процессе работы проекта (всего существует около 40 режимов), так что после суммирования можно легко переключить какие-либо блоки на выполнение вычитания новых произведений из накопленной суммы. Все эти возможности также могут быть автоматически использованы алгоритмами синтеза САПР ПЛИС.

Основные отличия между разновидностями блоков DSP48

Для удобства последующего анализа основные особенности блоков DSP48 сведены в таблице 2.

В таблице 2 в целом перечислены результаты добавления соответствующих ресурсов в секции DSP. В качестве общей тенденции можно отметить конвейеризацию и каскадирование, которые облегчают построение высокочастотных блоков большой разрядности. Для семейства Virtex-5 не очевидным, но важным следствием перехода к формату 18×25 стала возможность производить умножение мантисс чисел с плавающей точкой одинарной точности при использовании только 2 секций (вместо 4 в Virtex-4). Формат с плавающей точкой постепенно увеличивает свою популярность ввиду появления микросхем соответствующей производительности, и тот факт, что новое семейство существенно лучше подходит для операций с ним, дополнительно повышает привлекательность Virtex-5. В то же время даже в Spartan-3 можно параллельно выполнять десятки потоков вычислений с плавающей точкой.

Программное обеспечение

Все режимы работы блоков DSP, равно как и организация вспомогательных устройств на базе программируемых логических ячеек,

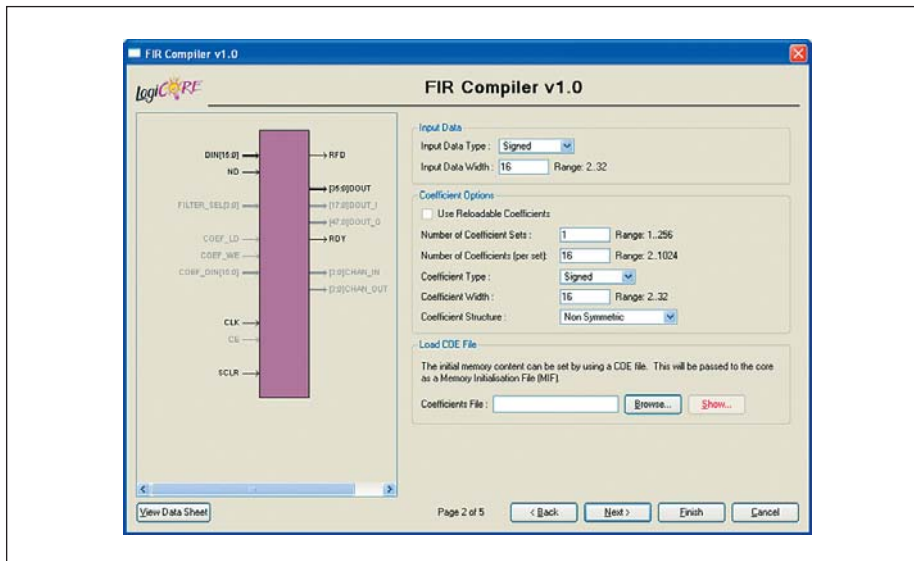


Рис. 7. Окно создания фильтра с конечной импульсной характеристикой в Core Generator

автоматически реализуются с помощью стандартных САПР (для Xilinx это ISE). Тем не менее, несмотря на достаточно мощные выразительные способности языков описания аппаратуры (можно написать, например, $\text{sum} \leq \text{sum} + \text{data} * \text{coeff}$;) широкая распространенность стандартных узлов цифровой обработки сигналов делает актуальной автоматизацию их создания. Поэтому такое массовое устройство, как цифровой фильтр с конечной импульсной характеристикой, реализуется в виде IP-ядра с помощью пакета CORE Generator (рис. 7).

Фильтр, спроектированный с помощью FIR Compiler, имеет важное преимущество: в процессе создания генерируется не только описание его поведения на VHDL/Verilog, но и дополнительные проектные ограничения, фиксирующие относительное расположение компонентов фильтра на кристалле. Известно, что при разработке сложных проектов FPGA большого объема важную роль играет относительное размещение отдельных компонентов. Ввиду взаимозаменяемости логических ячеек ПЛИС один и тот же проект может быть реализован огромным количеством вариантов, но подавляющее их большинство не являются оптимальными. Для того чтобы минимизировать задержки распространения сигнала и добиться наиболее эффективного использования аппаратных ресурсов ПЛИС, часто требуется «оказать помощь» САПР в виде сопроводения исходных текстов файлом проектных ограничений (Implementation constraints file). Такие ограничения, безусловно, можно создавать и самостоятельно, однако при этом требуется достаточно высокая квалификация разработчика и подробное знание особенностей архитектуры конкретного семейства FPGA. В то же время характер и содержание проектных ограничений не так уж сложны и запутанны,

и речь идет, как правило, об указании места относительного расположения отдельных узлов фильтра. Именно это и выполняется приложением CORE Generator для синтезируемых IP-ядер. Таким образом, ориентация на сгенерированные им цифровые узлы позволяет добиться достаточно высокой, а главное, гарантированной производимости.

Процедура проектирования фильтра достаточно проста и заключается в заполнении полей мастера, как показано на рис. 7. Отдельно можно обратить внимание на то, что расчет коэффициентов фильтра в данном случае не производится — разработчик должен предоставить их в виде .coe-файла (файл текстового формата). Задачей FIR Compiler является обеспечение правильного размещения логических ресурсов филь-

тра на кристалле ПЛИС, а не расчет коэффициентов или получение требуемой частотной характеристики. Сами коэффициенты могут быть получены другими способами.

Рассматривая порядок разработки цифровых фильтров, следует заметить, что теоретические положения, методы расчета и имеющиеся программные инструменты весьма и весьма разнообразны. Поэтому со стороны разработчика микросхем было бы неправильно создавать программное средство, ограниченное определенным кругом решаемых задач. Поскольку существующие математические пакеты предоставляют существенно большие возможности для моделирования и анализа систем цифровой обработки сигналов, естественно было бы провести ориентацию на них. Примером интеграции с готовым математическим пакетом является связка System Generator for DSP (продукт Xilinx), MATLAB и Simulink (широко известные продукты для расчета, анализа и моделирования). Установив пакет System Generator, разработчик получает доступ к набору аппаратных компонентов FPGA и вместе с тем может использовать стандартные блоки, такие как фильтры-преобразователи и т. д. На рис. 8 показано окно проектирования цифрового фильтра. В этом окне можно задать желаемые характеристики, выбрать порядок фильтра, автоматически рассчитать его коэффициенты. Впоследствии они могут быть использованы для автоматической настройки IP-ядра.

На рис. 9 показано окно Simulink с введенным проектом. Центральной его частью является цифровой фильтр, окруженный специальными компонентами, обеспечивающими ввод и вывод цифровых данных. Подав на вход сигнал от генератора определенного типа, можно наглядно наблюдать реакцию

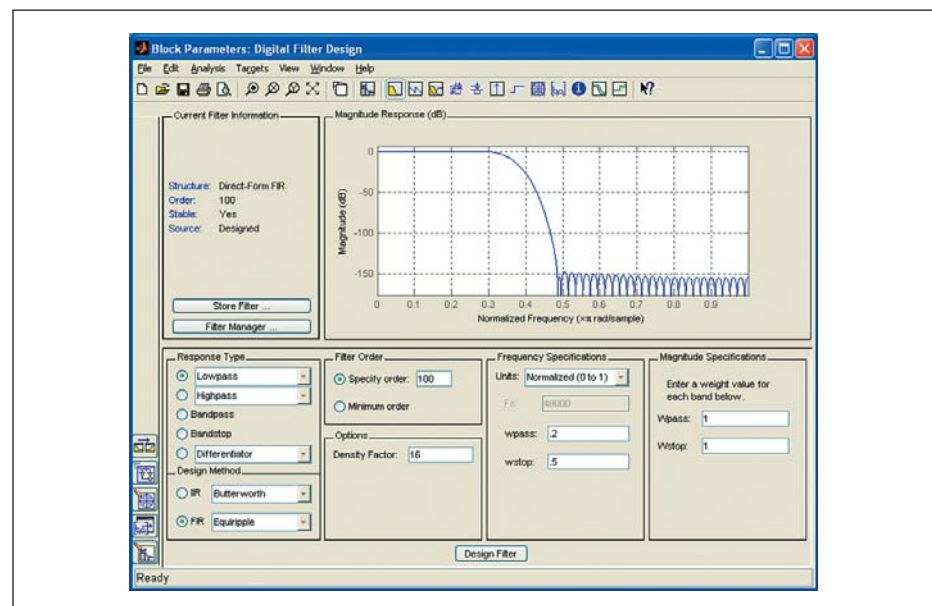


Рис. 8. Окно проектирования фильтра

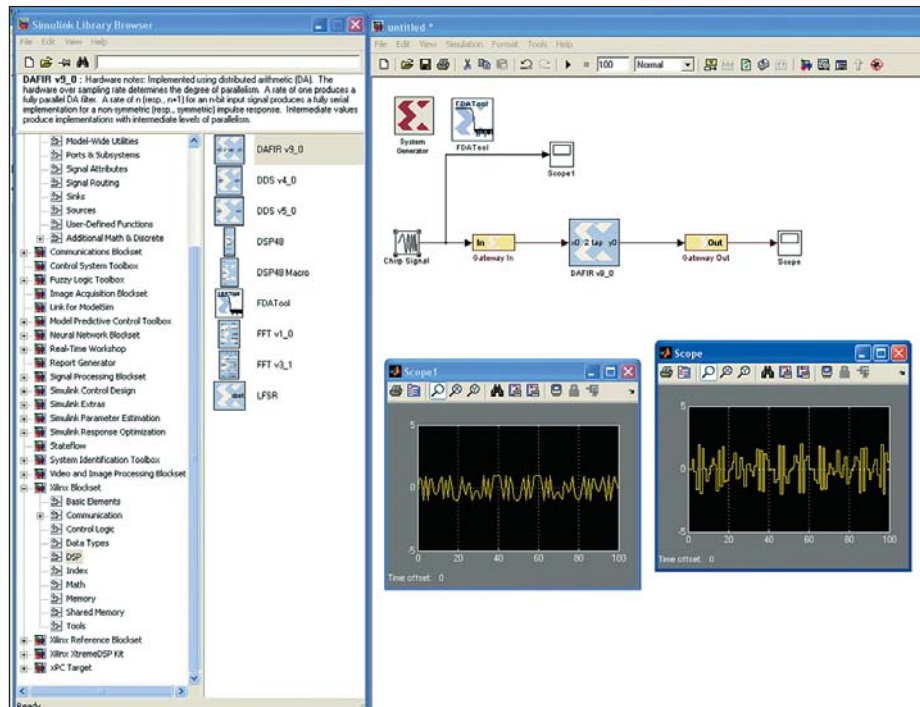


Рис. 9. Внешний вид окна Simulink при работе с дополнением System Generator for DSP

системы. Важно, что все результаты, полученные в Simulink, могут быть легко перенесены в ПЛИС без необходимости описания соот-

ветствующих модулей обработки вручную. Связка ISE + MATLAB + Simulink + System Generator является эффективным решением

для специалистов в области цифровой обработки сигналов, которая позволяет им не вникать в особенности архитектуры ПЛИС, а проводить проектирование в основном в среде MATLAB, получая затем рабочую конфигурацию ПЛИС «одним нажатием кнопки».

Заключение

Можно констатировать, что развитие архитектуры и элементной базы ПЛИС оказало в последнее время существенное влияние не только на практическую сторону проектирования систем цифровой обработки, но и позволило аппаратно реализовать такие алгоритмы, которые ранее не имели практической значимости ввиду крайне низкой производительности использования стандартных вычислительных средств. Вместе с тем появление новых семейств, в том числе привнесение свойств высокопроизводительных ПЛИС в устройства низкой стоимости, позволяет широким кругам разработчиков использовать в своих проектах всю мощь современных методов цифровой обработки сигналов, в том числе алгоритмов, ориентированных на параллельные вычисления. Хорошая поддержка программным обеспечением и высокая востребованность цифровых систем делает изделия на базе FPGA эффективными и конкурентоспособными. ■