

IP-ядра ПЛИС/СБИС для адаптивного формирования диаграммы направленности антенной решетки и пространственно-временной фильтрации

О. Ю. Зиновчик*, А. А. Фортунатов*, П. Б. Родионов*,

Московский физико-технический институт (национальный исследовательский университет),
Москва, Россия

Поступила в редколлегию 03.07.2026 г. Принята 01.08.2026 г.

Аннотация—В работе представлены два IP-ядра ПЛИС для адаптивной пространственно-временной фильтрации сигналов антенной решётки: ядро вычисления аппроксимации корреляционной матрицы и ядро адаптивного фильтра. Предложена гетерогенная архитектура, в которой ПЛИС выполняет потоковую обработку сигналов, а расчёт весовых коэффициентов вынесен на внешний вычислитель. Для сокращения потребляемых ресурсов применены оптимизации на трёх уровнях: учёт эрмитовости и стационарности корреляционной матрицы (снижение размерности с N^2M^2 до NM^2), алгоритм комплексного умножения Карацубы (экономия 25% умножений) и временное мультиплексирование каналов. Дополнительно использованы каскадные соединения DSP48 для уменьшения маршрутизации и повышения компактности размещения в ПЛИС Xilinx. Ядра синтезированы для ПЛИС Xilinx Zynq UltraScale+ MPSoC (ZU7EV) с параметрами $M = 16$ каналов, $N = 12$ временных отсчётов, $F_s = 30$ МГц. Результаты показывают линейный рост ресурсов DSP48 от числа отсчетов M и временных тапов N для корреляции и только от временных тапов N для фильтрации. Достижимость рабочей частоты 480 МГц и ограничение по времени накопления корреляций (35 мс из-за разрядности аккумуляторов). Архитектура масштабируема под различные значения параметров (M , N , F_s) и ограничения ПЛИС.

КЛЮЧЕВЫЕ СЛОВА: адаптивная пространственно-временная фильтрация, антенные решетки, корреляционная матрица, цифровая обработка сигналов, ПЛИС, DSP48.

DOI: 10.53921/18195822_2026_26_2.1_583

1. ВВЕДЕНИЕ

Современные радиолокационные и связные системы часто работают в условиях интенсивного воздействия активных источников помех, для подавления которых в антенных решетках применяют адаптивную пространственную фильтрацию. Диаграмма направленности, используемая для непрерывной подстройки коэффициентов адаптивного фильтра, в реальном времени формируется на основе оценки корреляционной матрицы сигналов, принятых антеннами. Однако классической пространственной фильтрации недостаточно при работе с широкополосными сигналами: переотражения помех, многолучевое распространение и различия в групповых задержках трактов приема сигналов требуют совместной пространственно-временной обработки [1–3]. Такой подход учитывает не только угловое положение источников, но и их спектрально-временные характеристики.

Реализация полноценной адаптивной пространственно-временной обработки в реальном времени на ПЛИС предъявляет жесткие требования к вычислительным ресурсам: размер корреляционной матрицы растет квадратично при увеличении числа рассматриваемых сиг-

налов, а вычисление коэффициентов фильтра требует вычисления матрицы, обратной матрице корреляции, или решения системы линейных уравнений. Несмотря на существование многочисленных аппаратных реализаций адаптивных фильтров, непосредственная реализация адаптивной пространственно-временной фильтрации осложняется квадратичным ростом корреляционной матрицы и высокой стоимостью вычисления весовых коэффициентов. Реализация подобной адаптивной системы на ПЛИС требует большого объема вычислительных ресурсов, имеющихся только у наиболее мощных ПЛИС [4, 5]. По этой причине требуется гетерогенная архитектура, позволяющая оставить высокоскоростную потоковую часть на FPGA, а наиболее ресурсоемкие вычисления, связанные с формированием диаграммы направленности и расчетом коэффициентов фильтрации, вынести во внешний вычислитель.

В настоящей работе предложены и реализованы два вычислительных ядра ПЛИС, требующиеся для адаптивной пространственно-временной фильтрации. В отличие от существующих реализаций, в работе одновременно объединены оптимизации, учитывающие свойства корреляционной матрицы и физическое расположение вычислительных компонентов DSP48 в ПЛИС серии AMD (Xilinx). Таким образом в работе рассматриваются оптимизации трех уровней: алгоритмические, архитектурные, физического размещения. Данная работа фиксирует объем и типы данных, передаваемых между ПЛИС и внешним вычислителем, но при этом конкретный тип вычислителя и алгоритм, вычисляющий коэффициенты фильтра по матрице корреляции, остаются за рамками данной работы. Оба ядра были успешно синтезированы в ПЛИС Xilinx Zynq Ultrascale+ MPSoC, в работе приведена статистика использования ресурсов при различных параметрах системы, описаны методы адаптации ядер под конкретные пользовательские задачи и ограничения.

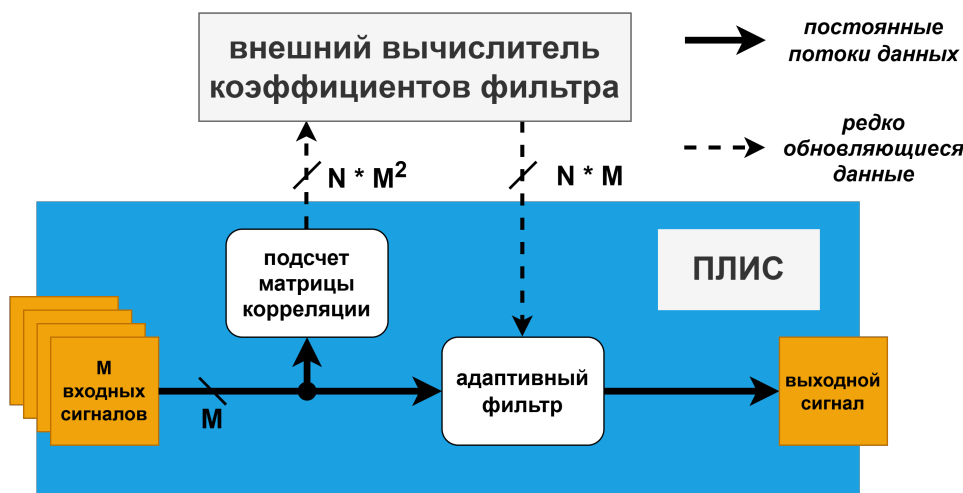


Рис. 1. Схема предлагаемой гетерогенной архитектуры для адаптивной пространственно-временной фильтрации.

2. СОСТАВ СИСТЕМЫ

На рис. 1 показана предлагаемая в работе гетерогенная архитектура для пространственно-временной адаптивной фильтрации. Предлагаемая схема реализована из двух вычислителей: ПЛИС, обрабатывающей непрерывные входные сигналы от АЦП, и внешний по отношению к этой потоковой обработке вычислитель, использующийся в первую очередь для расчета коэффициентов весов адаптивного фильтра. В данной работе рассматриваются только два вычис-

лительных блока, работающих в ПЛИС: блок аппроксимации элементов матрицы корреляции и адаптивный фильтр. Внешним вычислителем может быть еще одна внешняя ПЛИС, ЦПУ, ГПУ, вычислительный блок или даже процессорное ядро на этой же ПЛИС (например, ЦПУ на ресурсах ПЛИС [6] или ARM-ядро в ПЛИС серии Zynq от Xilinx [7]) - конкретная реализация и алгоритм, который работает на этом вычислителе, выходит за рамки данной работы.

Число антенн в данной работе обозначается параметром M , каждый из этих каналов представляет собой непрерывный поток знаковых целочисленных I/Q отсчетов с общей частотой дискретизации F_s . Временная составляющая «пространственно-временной» обработки достигается добавлением этих же сигналов, задержанных на $1, 2, \dots, N - 1$ тактов, где N — это второй основной параметр системы. В математической постановке задачи все NM сигналов рассматриваются как независимые (формула 3.4 в [8]), что приводит к необходимости подсчета $N^2 M^2$ элементов корреляционной матрицы. Заметим, что смещенные таким образом по фазе сигналы не создаются глобально (и потому не показаны на рис. 1), а реализуются в линиях задержек непосредственно в двух рассматриваемых вычислительных ядрах.

В данной работе не ставится задача расчета весов в реальном времени. Внешний вычислитель получает приближение матрицы корреляции сигналов, посчитанное на ограниченном временном отрезке, и на основе этого приближения формирует диаграмму направленности и записывает соответствующие значения весов в фильтр на ПЛИС. При такой постановке задачи не требуется реализовывать скоростную передачу данных между ПЛИС и внешним вычислителем, что позволяет реализовать интерфейс между ними с помощью пары двухпортовых BRAM, имеющихся в ПЛИС.

Обозначим явно интерфейсы каждого вычислительного блока. Ядро вычисления аппроксимации корреляционной матрицы принимает на вход M параллельных сигналов целочисленных знаковых I/Q отсчетов с частотой дискретизации F_s , а результат записывается в виде массива комплексных чисел в BRAM. Ядро адаптивной фильтрации принимает на вход те же потоки, что и первое ядро, дополнительно подгружает весовые коэффициенты из второй BRAM, а результат применения фильтра ко входным потокам (с последующим усреднением всех M каналов) — это сигнал целочисленных знаковых I/Q отсчетов с той же частотой дискретизации F_s . Этот сигнал и есть результат пространственно-временной фильтрации, адаптивность которой заключается в периодическом обновлении весовых коэффициентов фильтра.

3. ОПТИМИЗАЦИЯ АЛГОРИТМА

Математическая корреляционная матрица NM -компонентного вектора сигналов $x(t)$ в соответствии формулой 3.5 в [8] определяется как

$$R = E\{x(t) \cdot x^H(t)\}.$$

Пусть m — это номер пространственного канала (номер антенны), а n — это сдвиг по фазе, выраженный в количестве отсчетов:

$$R_{m,m'}(n, n') = E\{x_m(t - n\Delta t) \cdot x_{m'}^*(t - n'\Delta t)\}.$$

Построенная таким образом матрица является эрмитовой (формула 3.7 в [8]), то есть:

$$R_{m,m'}(n, n') = R_{m',m}^*(n', n). \quad (1)$$

В реальной системе мы вынуждены приближать математическое ожидание усреднением по отрезку времени, то есть по некоторому числу K последовательных отсчетов:

$$\tilde{R}_{m,m'}(n, n') = \frac{1}{K} \sum_{k=0}^{K-1} x_m[k-n] \cdot x_{m'}^*[k-n']. \quad (2)$$

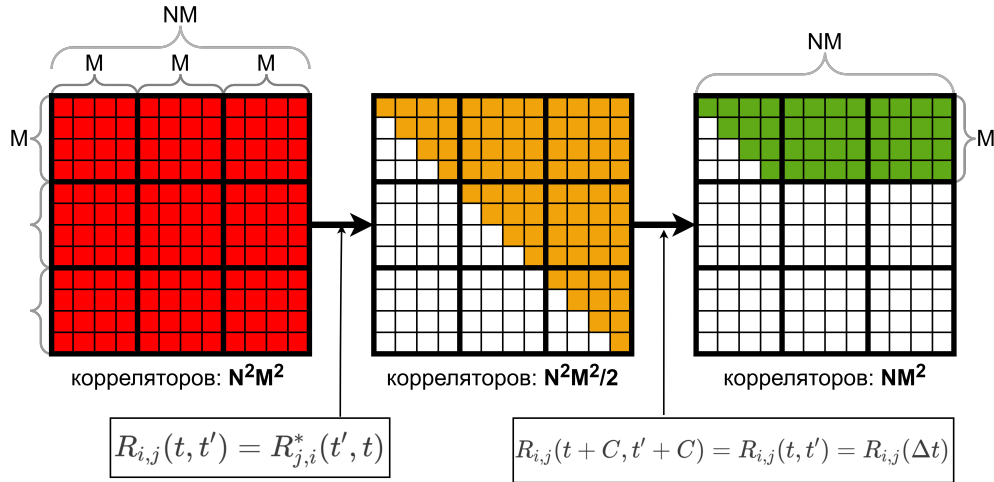


Рис. 2. Число вычисляемых комплексных корреляций: (слева) исходно, (центр) после учета эрмитовости, (справа) после учета зависимости корреляции только от относительной фазы между сигналами.

Воспользуясь свойством эрмитовости (1) матрицы корреляции, получаем, что достаточно вычислить только половину всех корреляций сигналов ($N^2 M^2 \rightarrow N^2 M^2 / 2$), а именно элементы верхней треугольной подматрицы (рис. 2, слева). Дополнительно учитываем, что сигнал обладает одинаковыми статистическими свойствами вне зависимости от его смещения по фазе, а значит корреляция двух сигналов зависит от их относительного фазового сдвига, то есть:

$$\tilde{R}_{m,m'}(n, n') = \tilde{R}_{m,m'}(0, n' - n) = \frac{1}{K} \sum_{k=0}^{K-1} x_m[k-n] \cdot x_{m'}^*[k - (n' - n)],$$

$$\tilde{R}_{m,m'}(\Delta n) = \tilde{R}_{m,m'}(0, n' - n) = \frac{1}{K} \sum_{k=0}^{K-1} x_m[k] \cdot x_{m'}^*[k - \Delta n].$$

Свойство эрмитовости (1) для такой матрицы запишется в следующем виде:

$$\tilde{R}_{m,m'}(\Delta n) = \tilde{R}_{m',m}^*(-\Delta n).$$

Таким образом, применение указанных двух свойств в совокупности означает, что вычислять требуется только $M \cdot M \cdot N = NM^2$ приближений элементов матрицы корреляции (рис. 2, справа).

Отдельно уточним, что, аналогично книге [8], в данной работе под «матрицей корреляции» подразумевается математическое ожидание (и усреднение по ограниченному числу отсчетов) эрмитова произведения исходных сигналов — без последующего поэлементного деления на дисперсии сигналов, без предварительного вычитания среднего значения сигналов (что, однако, может быть реализовано с помощью дополнительных вычислительных блоков). Также для

упрощения реализации в вычислительных ядрах отсутствует операция усреднения результата по объему выборки — ни скользящем средним, ни целочисленным делением. Это означает, что внешний вычислитель должен самостоятельно нормализовать значения приближенной корреляционной матрицы, если этого требует алгоритм.

Итак, коррелятор двух сигналов устроен тривиально: комплексное умножение с аккумулярованием. Наивное вычисление комплексного произведения потребует четырех целочисленных произведений и двух сложений/вычитаний. Для реализации в ПЛИС в первую очередь необходимо сокращать именно число целочисленных умножений, так как в ПЛИС количество DSP-блоков с реализованными в нем эффективными аппаратными умножителями ограничено.

Для сокращения числа вещественных умножений в операции комплексного умножения традиционно используют алгоритм Карацубы (или, по другим источникам, «трюк Гаусса») [9–11], позволяющий вычислить комплексное произведение за 3 целочисленных умножения и 5 сложений/вычитаний ¹:

$$\begin{cases} c_1 = a_{re} \cdot (b_{re} + b_{im}) \\ c_2 = (a_{re} + a_{im}) \cdot b_{im} \\ c_3 = (a_{re} - a_{im}) \cdot b_{re} \end{cases} \Rightarrow \begin{cases} c_{re} = c_1 - c_2 \\ c_{im} = c_1 - c_3 \end{cases} \quad (3)$$

Использование каскадов DSP, описанное далее по тексту, позволяет вынести сложения/вычитания за каскад, что также уменьшает объем производимых вычислений, так что по описываемые далее вычислительные блоки описывают нахождение слагаемых Карацубы (c_1 , c_2 и c_3), вычисления которых полностью параллельны. Суммарно применение алгоритма Карацубы позволяет сэкономить 25% операций целочисленного умножения (для корреляции $4NM^2 \rightarrow 3NM^2$, для фильтра $4NM \rightarrow 3NM$).

4. РЕАЛИЗАЦИЯ БЛОКОВ ОБРАБОТКИ

Одной из техник компактной реализации систолических массивов является временное мультиплексирование, он же мультипампинг вычислительных блоков [13–15]. Техника заключается в формировании одного скоростного потока данных из нескольких M медленных параллельных потоков посредством чередования отсчетов от каждого канала. При этом частота результирующего потока должны быть не менее чем в M раз выше, чем у исходных, чтобы исключить потерю отсчетов. Техника временного мультиплексирования эффективно применять в тех случаях, когда скорость исходных потоков низкая, вычисления над каждым из них идентичны, а вычислительный блок способен работать на повышенной частоте. Таким образом, условие применимости временного мультиплексирования M потоков с частотой дискретизации F_s можно формализовать в виде

$$M \cdot F_s \leq F_{MAX}, \quad (4)$$

где F_{MAX} — максимальная частота работы вычислительного блока.

Применение техники временного мультиплексирования всех M потоков для данной задачи позволяет реализовать $3NM^2$ блоков корреляции на $3NM$ блоках DSP, а фильтр, состоящий из $3NM$ целочисленных произведений со сложениями, реализуется на $3N$ блоках DSP.

На рис. 3 представлена архитектура базового блока вычисления корреляции двух сигналов (вход #1 и вход #2). Благодаря применению техники временного мультиплексирования, блоки задержки z^{-1} и z^{-2} не только реализуют задержки в $1, 2, \dots, N - 1$ тактов, но и позволяют вычислять попарные корреляции сигналов на разных DSP приведенного каскада. На рис. 3

¹ В ПЛИС семейства Versal от Xilinx операцию комплексного умножения возможно реализовать двумя блоками DSP58 в режиме DSPCPLX [12]

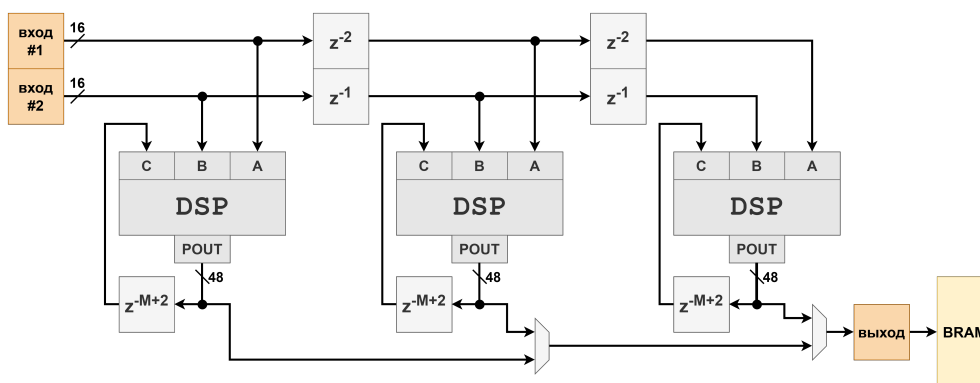


Рис. 3. Архитектура блока вычисления корреляции M пар каналов, мультиплексированных по времени.

приведены только первый, промежуточный и замыкающий сегменты каскада вычислений, полная же длина такого каскада равна NM .

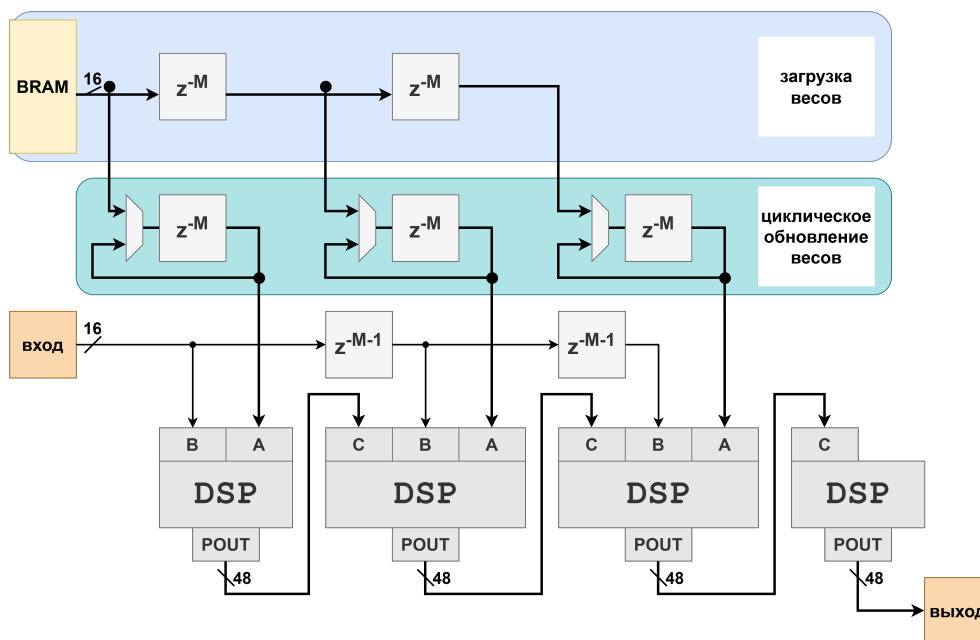


Рис. 4. Архитектура базового блока фильтрации M каналов, мультиплексированных по времени.

На рис. 4 приведена архитектура базового блока фильтра. На рис. 4 приведены только первый, промежуточный, замыкающий и агрегирующий (суммирующий все M отфильтрованных потоков) сегменты каскада вычислений, полная же длина такого каскада равна $N + 1$. Архитектура также включает подсистему циклического обновления весов, каждый такт подающую релевантные веса для каждого из N временных тапов фильтра, и подсистему загрузки весов, записываемые внешним вычислителем в BRAM.

Каждый из каскадов на рис. 3 и рис. 4 считает одно из слагаемых Карацубы (3), и для полноценной обработки комплексных потоков требуется по три параллельно работающих указанных каскада, по одному на каждое слагаемое. Приведенные вычислительные архитектуры могут быть реализованы на любом ПЛИС без привязки к конкретному производителю. В сле-

дующем разделе будет рассмотрена оптимизация размещения обоих вычислительных ядер в ПЛИС для достижения их большей компактности.

5. РАЗМЕЩЕНИЕ КОМПОНЕНТОВ

Следующая область оптимизаций реализуемой архитектуры применима только к ПЛИС от AMD (Xilinx), поскольку в них аппаратные блоки DSP48 организованы в вертикальные колонки по всей длине кристалла, и внутри этих колонок соседние DSP48 соединены особым образом, рассмотренным в этом разделе.

Применение техники временного мультиплексирования кратно уменьшает объем используемых вычислительных ресурсов, однако оно же приводит к необходимости работать с сигналами повышенной частоты. Зачастую рекомендованная частота работы блоков DSP48 превосходит частоту работы основной части дизайна [16], так что подведение входящих и исходящих линий данных к блокам DSP48 средствами общей маршрутизации может приводить к долгой автоматизированной имплементации, «несошедшимся таймингам», артефактам при работе дизайна в ПЛИС.

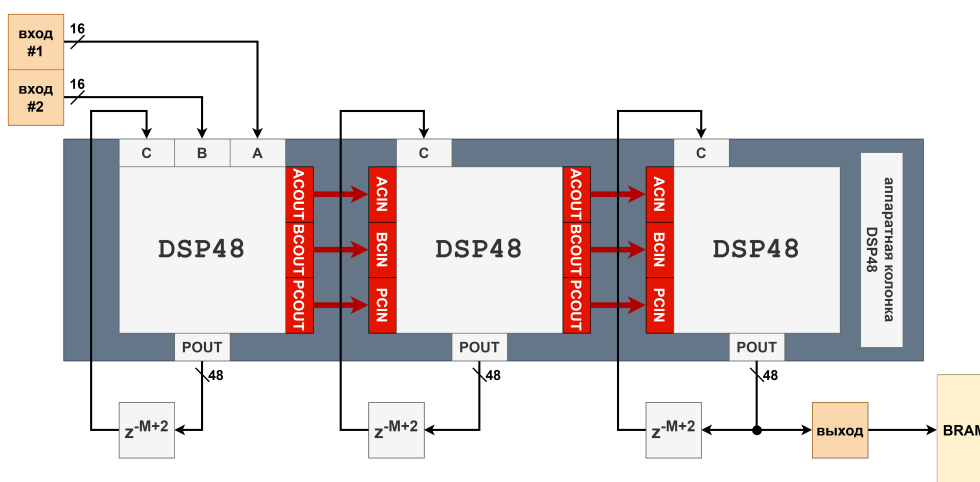


Рис. 5. Архитектура блока вычисления корреляции M пар каналов, мультиплексированных по времени, с применением каскадных соединений.

Аппаратные колонки DSP48 в ПЛИС фирмы AMD (Xilinx) позволяют реализовывать обмен данными между непосредственными соседними ячейками одного систолического массива без привлечения общей маршрутизации. Каждый блок DSP48 может предоставить своему «соседу» в каскаде: 48-битный результат вычислений (PCOUT $\rightarrow$ PCIN), задержанный на 1-2 такта 30-битный аргумент A (ACOUT $\rightarrow$ ACIN), задержанный на 1-2 такта 18-битный аргумент B (BCOUT $\rightarrow$ BCIN) [17]. На рис. 5 и рис. 6 представлены архитектуры коррелятора и фильтра, оптимизированные под физическое расположение блоков DSP48, красным цветом подсвечены задействованные каскадные соединения между элементами каскада. В блоке фильтра (рис. 6) была использована передача результата вычислений (P), в блоке корреляции (рис. 5) были использованы все основные каскадные соединения (A, B, P). Последнее особенно важно, поскольку в блоке аппроксимации корреляционной матрицы число соединенных подряд блоков DSP48 равно NM , что при больших значениях N и M может означать использование всей аппаратной колонки DSP48 во всю длину ПЛИС.

Также играет важную роль объем дополнительных логических ресурсов, требующихся для работы каждого элемента вычислительного каскада. Без учета управляющих сигналов, опре-

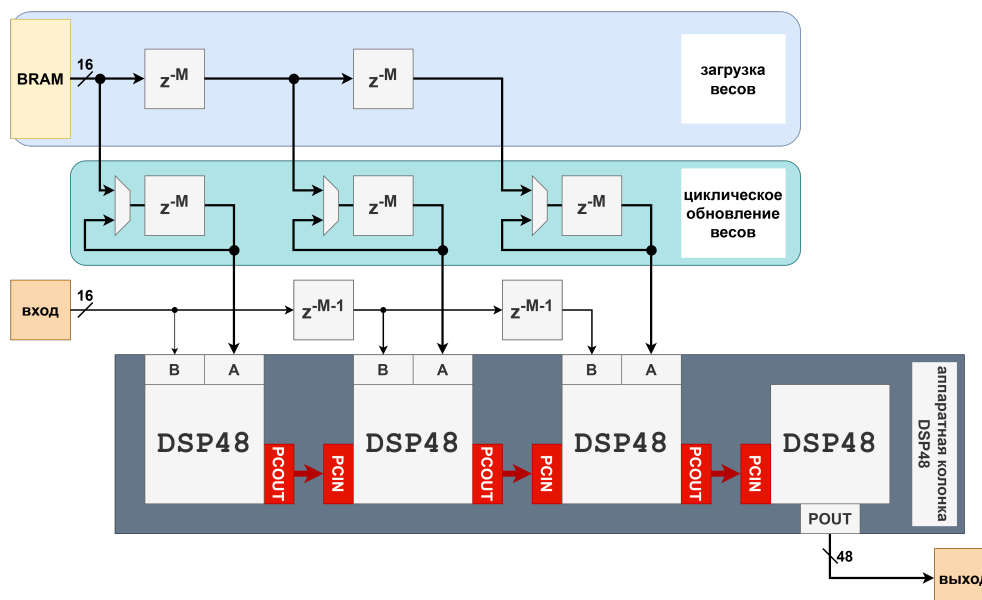


Рис. 6. Архитектура блока фильтрации M каналов, мультиплексированных по времени, с применением каскадных соединений.

деляющих режим работы DSP48 в каждый момент времени, основным дополнительным ресурсом являются линии задержки. В ПЛИС Xilinx они эффективно реализуются на блоках SLICEM, включающих в себя LUT, работающую в режиме 16-ти однобитных сдвиговых регистров SRL16E, и дополнительных 16 однобитных регистров [18]. Таким образом, один SLICEM позволяет реализовать линию задержки до 17 тактов для 16-битных потоков данных.

По рис. 5 и рис. 6 можем оценить, сколько SLICEM приходится на один DSP48 каскада. Для блока фильтра — три SLICEM (один для линии загрузки весов, один для циклического обновления весов и один для линии задержки аргумента), для блока корреляции — тоже три SLICEM (задержка 48-битных аккумуляторов реализуется на $48/16=3$ блоках SLICEM).

6. РЕЗУЛЬТАТЫ

Для реализации вычислительной системы была выбрана ПЛИС AMD (Xilinx) Zynq UltraScale+ MPSoC (ZU7EV), с 1728 блоками DSP48, организованных в аппаратные колонки до 144 блоков DSP48 подряд [19]. При максимальной конвейеризации блока DSP48 (что было учтено при реализации блоков) максимальная частота его работы равна не менее 600 МГц (точное значение определяется Speed Grade ПЛИС и выбранным напряжением VCCINT) (табл. 83 в [16]). Параметры реальной решаемой задачи, требующейся к реализации на плате, следующие: $M = 16$ каналов, $N = 12$ временных отсчетов, 14-битные потоки данных с частотой дискретизации $F_s = 30$ МГц.

Таблица 1. Сравнение оценки количества целочисленных умножений и вычислителей после каждой стадии оптимизации для блоков приближения матрицы корреляции и фильтра.

стадия оптимизации	кол-во DSP (и при $M = 16, N = 12$)	
	корреляция	фильтрация
до оптимизаций	$4N^2M^2$ (147456)	$4NM$ (768)
свойства корр. матрицы	$4NM^2$ (12288)	$4NM$ (768)
алгоритм Карацубы	$3NM^2$ (9216)	$3NM$ (576)
временное мультиплексирование	$3NM$ (576)	$3N$ (36)

В табл. 1 приведено сравнение объема вычислений и вычислителей после каждой стадии оптимизации. Без применения приведенных в данной работе оптимизаций реализовать блок приближения матрицы корреляции, требующий 147456 DSP48, на выбранном ПЛИС с 1728 блоками DSP48 было бы невозможно. Также, как было замечено ранее, независимость от числа каналов M количества DSP48, используемых фильтром, достигается за счет временного мультиплексирования всех M каналов.

Формулы, приведенные в табл. 1 являются оценками использования DSP, не учитывающими, например, сложения в формуле (3), которые в реализации были вынесены за приведенные вычислительные каскады. С учетом этих поправок более точные оценки финального использования ресурсов DSP будут следующими:

$$N_{DSP}^{\text{корреляция}} = 3NM + 2, \quad (5)$$

$$N_{DSP}^{\text{фильтрация}} = 3N + 6. \quad (6)$$

Таблица 2. Сравнение ресурсов, занимаемых ядром корреляции при разных значениях параметров N и M .

число отсчетов N	число антенн $M = 4/8/16$			
	DSP	CLB LUT	CLB Register	Block RAM Tile
1	14 / 26 / 50	491 / 847 / 1334	922 / 1516 / 2510	0 / 0 / 1.5
6	74 / 146 / 290	2206 / 3732 / 7316	3885 / 7247 / 14353	0 / 1.5 / 3
12	146 / 290 / 590	3735 / 7323 / 14874	7246 / 14352 / 28502	1.5 / 3 / 11

Таблица 3. Сравнение ресурсов, занимаемых ядром фильтрации при разных значениях параметров N и M .

число отсчетов N	число антенн $M = 4/8/16$			
	DSP	CLB LUT	CLB Register	Block RAM Tile
1	9	1131	1431	1
6	24	1737	2236	1
12	42	2468	3202	1

В табл. 2 и 3 соответственно приведены результаты использования ресурсов ПЛИС после имплементации в Vivado 2024.1 на ПЛИС XCZU7EV-FFC1156-2-i. Оценки использования DSP удовлетворяют формулам (5) и (6), за исключением случая ($M = 16$, $N = 12$). Именно в этом случае длина одного каскада DSP48 оказалась равной $16 \cdot 12 = 192$, что превосходит высоту аппаратных колонок DSP48 на этом кристалле, равную 144. Из-за этого пришлось разделить каждый из трех каскадов на два, что привело к дополнительным расходам DSP48 и управляющую логику.

На рис. 7 показан результат размещения вычислительных блоков фильтра и приближения корреляционной матрицы в САПР Vivado 2024.1. Желтым цветом на рисунке подсвечены использованные DSP48, красным — сдвиговые регистры SLICEM, реализующие задержки потоков с помощью SRL16E. На снимке видно характерное вертикальное расположение DSP48 по аппаратным колонкам и SRL16E вдоль них. Отметим, что корреляционных каскадов шесть, а не три, по причине, упомянутой ранее в работе.

Также заметим, что количество блоков SRL16E, требующихся на один блок DSP48 в каскаде блока корреляции, совпадает с оценкой — три SRL16E на один DSP48. Как видно на рис. 7 справа, высота двух блоков DSP48E2 в выбранном ПЛИС равняется высоте пяти SLICEM, что объясняет задействование Vivado одного SLICEM с противоположной стороны аппаратной колонки DSP48 с периодичностью приблизительно пять блоков SLICEM (рис. 7, центр). Достаточность использования каскадами DSP48 только близлежащих ресурсов SLICEM также определяет компактность реализованных ядер фильтра и корреляции.

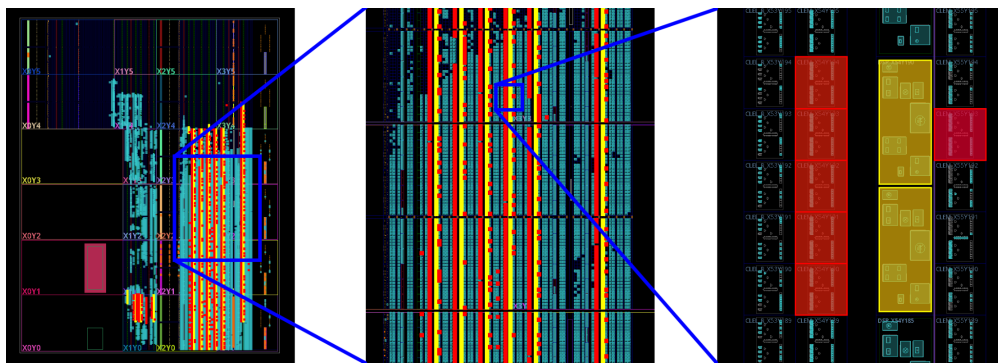


Рис. 7. Снимок экрана Vivado с размещением использованных компонентов: (слева) весь дизайн с подсвеченными блоками фильтра и корреляции, (центр) блок корреляции, (справа) взаимное расположение двух DSP48 и шести SLICEM в каскаде блока корреляции; желтым цветом выделены блоки DSP48E2, красным — SLICEM.

7. ОБСУЖДЕНИЕ

Временное мультиплексирование в данном случае было допустимо применить, поскольку при этом выполняется условие (4): $M = 16$, $F_s = 3$ МГц, $M \cdot F_s = 480$ МГц, что меньше $F_{MAX} = 600$ МГц — максимальной рабочей частоты DSP48 выбранного ПЛИС.

При повышении числа каналов M или частоты дискретизации сигналов F_s частота сигнала, объединяющего таким образом все M каналов, может выйти за рабочий диапазон DSP. Для обработки таких сигналов архитектуры обоих ядер, описанные в данной работе, приемлемы, если мультиплексировать только часть сигналов и при этом использовать несколько параллельных вычислительных блоков. Так, например, если разбить вектор из $2M$ сигналов на два вектора из M сигналов, то вместо одного блока корреляции и одного блока фильтрации с параметрами $(2M, N)$, потребуется два блока фильтрации (с последующим сумматором их результатов) и четыре блока корреляции с параметрами (M, N) . При этом все блоки будут работать на частоте MF_s , а не $2MF_s$ — именно это позволит уложиться в рабочий диапазон частот DSP. Приведенный метод разбиения не значительно увеличивает дополнительные расходы DSP, поскольку с параметрами $(2M, N)$ потребуется $3N \cdot 2M + 3N$ блоков DSP, с разбиением — $4 \cdot 3NM + 2 \cdot 3N$, то есть вдвое больше. Иными словами, понижение частоты работы каждого вычислительного блока в рассматриваемой задаче приводит к пропорциональному росту использования ресурсов DSP при фиксированных числах антенн M и отсчетов N .

Заметим, что увеличение значения параметра N не влияет на частоту сигнала, только увеличивает количество потребляемых ресурсов DSP линейно и для ядра приближения корреляционной матрицы, и для блока фильтрации.

Главное ограничение текущей архитектуры — это временной отрезок, в течение которого считаются корреляции сигналов. Оно определяется условием недопущения переполнения аккумуляторов в блоках DSP. В DSP48E2, примененных в работе, используются 48-битные знаковые сумматоры, а битность знаковых аргументов умножения равна 14 и 15 бит (использовался 14-битный АЦП, а битность второго аргумента на 1 больше, что следует из (3)). Следовательно, число отсчетов K , указанное в (2), приблизительно равно:

$$\frac{2^{47}}{2^{13} \cdot 2^{14}} = 2^{20} = 1048576.$$

При выбранной частоте дискретизации $F_s = 30$ МГц это соответствует 35 миллисекундам — оценки корреляции каждой пары сигналов подсчитывается именно за этот промежуток времени.

8. ЗАКЛЮЧЕНИЕ

В работе были разработаны и оптимизированы вычислительные архитектуры двух критически важных для адаптивной пространственно-временной фильтрации ядер: ядро вычисления корреляции M сигналов и ядро адаптивной фильтрации этих сигналов с весами. Оба блока рассчитаны на работу в описанной гетерогенной системе, в которой основной адаптивный алгоритм вычисления коэффициентов фильтра вынесен на внешний вычислитель.

Для достижения требуемой скорости работы блоков и обработки требующегося объема данных были использованы оптимизации на нескольких уровнях: алгоритмическом (свойства корреляционной матрицы), аппаратном (техника временного мультиплексирования), и на уровне размещения компонентов в ПЛИС (аппаратные каскады DSP). Полученные результаты подтверждают, что именно примененные оптимизации позволили реализовать ядра корреляции и фильтрации в условиях ограниченных вычислительных ресурсов.

Предложенная архитектура масштабируется по числу антенн M и числу временных отсчетов N . Практические пределы масштабирования определяются характеристиками конкретной ПЛИС: доступным числом блоков DSP48, высотой их аппаратных колонок и отношением максимальной рабочей частоты F_{MAX} к частоте дискретизации сигналов F_s . Главным ограничением системы является временной промежуток, по которому считается корреляция сигналов. Его длительность определяется разрядностью сумматоров в DSP48E2 блоках ПЛИС Xilinx, в описываемой в работе задаче (14-битные АЦП, 48-битные сумматоры, $M = 16$, $N = 12$, $F_s = 30$ МГц) длительность этого промежутка составила 35 миллисекунд. Последующее совершенствование предложенной архитектуры скользящего усреднения по времени позволит снять это ограничение.

СПИСОК ЛИТЕРАТУРЫ

1. Карутин С.Н., Харисов В.Н., Павлов В.С. Оптимальные алгоритмы пространственно-временной обработки сигналов для высокоточных приложений. *Радиотехника*, 2018, №9, стр. 131–138.
2. Ефименко В.С., Харисов В.Н., Павлов В.С. Оптимальные алгоритмы пространственно-временной обработки сигналов и их характеристики. *Радиотехника*, 2016, №9, стр. 113–120.
3. Widrow B., Mantey P.E., Griffiths L.J., Goode B.B. Adaptive Antenna Systems. *Proceedings of the IEEE*, 1967, vol. 55, no. 12, pp. 2143–2159.
4. Friberg S., Pålsson P. *Space-Time Adaptive Processing in FPGA* [Электронный ресурс]. Master's thesis, Chalmers Univ. Technol. (Gothenburg), 2012.
5. Mahmood Z.U., Alam M., Jamil K., Al-Hekail Z.O. FPGA Implementation of Space-Time Adaptive Processing (STAP) Algorithm for Target Detection in Passive Radars. *Progress In Electromagnetics Research C*, 2013, vol. 35, pp. 35–48.
6. Advanced Micro Devices, Inc. *MicroBlaze Processor Reference Guide (UG984)* [Электронный ресурс]. 2021. URL: <https://docs.amd.com/r/en-US/ug984-vivado-microblaze-ref>.
7. Advanced Micro Devices, Inc. *Zynq UltraScale+ MPSoC Software Developer Guide (UG1137)* [Электронный ресурс]. 2022. URL: <https://docs.amd.com/r/en-US/ug1137-zynq-ultrascale-mpsoc-swdev>.
8. Джиган В.И. *Адаптивная фильтрация сигналов: теория и алгоритмы*. М.: Техносфера, 2013. 528 с. ISBN 978-5-94836-342-4.
9. Fam A.T. Efficient Complex Matrix Multiplication. *IEEE Trans. Computers*, 1988, vol. C-37, no. 7, pp. 877–879.
10. Блейхут Р. *Быстрые алгоритмы цифровой обработки сигналов*. М.: Мир, 1989. Пер. с англ. И.И. Грушко.
11. Шень А. Gauss multiplication trick? *Математическое просвещение*, сер. 3, 2019, вып. 24, с. 19–33.

12. Advanced Micro Devices, Inc. *Versal ACAP DSP Engine Architecture Manual (AM004)* [Электронный ресурс]. 2025. URL: <https://docs.amd.com/r/en-US/am004-versal-dsp-engine>.
13. Ronak B., Fahmy S.A. Multipumping Flexible DSP Blocks for Resource Reduction on Xilinx FPGAs. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, 2017, vol. 36, no. 9, pp. 1471–1482.
14. Johnsen C.-J., de Matteis T. Temporal Vectorization: A Compiler Approach to Automatic Multi-Pumping. 2022.
15. Canis A., Anderson J.H., Brown S.D. Multi-Pumping for Resource Reduction in FPGA High-Level Synthesis. In: *Design, Automation & Test in Europe Conference (DATE)*, 2013, pp. 194–197.
16. Advanced Micro Devices, Inc. *Zynq UltraScale+ MPSoC Data Sheet: DC and AC Switching Characteristics (DS925)* [Электронный ресурс]. 2025. URL: <https://docs.amd.com/r/en-US/ds925-zynq-ultrascale-plus>.
17. Advanced Micro Devices, Inc. *UltraScale Architecture DSP Slice User Guide (UG579)* [Электронный ресурс]. 2021. URL: <https://docs.amd.com/v/u/en-US/ug579-ultrascale-dsp>.
18. Advanced Micro Devices, Inc. *UltraScale Architecture Libraries Guide (UG974)* [Электронный ресурс]. 2025. URL: <https://docs.amd.com/r/en-US/ug974-vivado-ultrascale-libraries>.
19. Advanced Micro Devices, Inc. *Zynq UltraScale+ MPSoC Data Sheet: Overview (DS891)* [Электронный ресурс]. 2025. URL: <https://docs.amd.com/v/u/en-US/ds891-zynq-ultrascale-plus-overview>.

IP-cores for adaptive beamforming and space-time processing for FPGA/ASIC

O. Yu. Zinovchik, A. A. Fortunatov, P. B. Rodionov

This paper presents two FPGA IP cores for adaptive space-time filtering of antenna array signals: a correlation matrix approximation core and an adaptive filtering core. A heterogeneous architecture is proposed, where the FPGA performs streaming signal processing while weight coefficient calculation is offloaded to an external processor. Resource optimizations are applied at three levels: exploiting Hermitian and stationary properties of the correlation matrix (reducing complexity from N^2M^2 to NM^2 , Karatsuba complex multiplication (saving 25% of multiplications), and time-division multiplexing of channels. Additionally, DSP48 cascading is used to minimize routing and improve layout compactness in Xilinx FPGAs. The cores are synthesized for a Xilinx Zynq UltraScale+ MPSoC (ZU7EV) with $M = 16$ channels, $N = 12$ time taps, and $F_s = 30$ MHz. Results show linear DSP48 resource scaling with M and N for correlation, with N for filtering, an achievable clock rate of 480 MHz, and a correlation accumulation limitation of 35 ms due to accumulator bit width. The architecture is scalable for various values of (M, N, F_s) and FPGA constraints.

KEYWORDS: adaptive beamforming, antenna arrays, space-time adaptive processing, STAP, digital signal processing, correlation matrix, time-division multiplexing, FPGA, DSP48.