

УДК 621.396; ГРНТИ 47.49

РЕАЛИЗАЦИЯ БЫСТРОГО ПРЕОБРАЗОВАНИЯ ФУРЬЕ НЕЭКВИДИСТАНТНОЙ ПОСЛЕДОВАТЕЛЬНОСТИ ИМПУЛЬСОВ НА ПЛИС ТИПА FPGA

Кошелев В.И., д.т.н., профессор, зав. кафедрой РТС РГРТУ, e-mail: koshelev.v.i@rsreu.ru

Чинь Н.Х., аспирант кафедры РТС РГРТУ, e-mail: ngochieu.radioscientist@mail.ru

IMPLEMENTATION OF FAST FOURIER TRANSFORM OF NON-EQUIDISTANT PULSE SEQUENCES ON FPGA

Koshelev V.I., Trinh N.H.

The Fast Fourier Transform (FFT) processor is widely used to perform a variety of radio engineering tasks, in particular multi-channel frequency filtering (MFF) of signals with an unknown frequency. However, the use of classical FFT for processing non-equidistant pulse sequences does not ensure optimal processing according to the energy criterion of the signal-to-noise ratio. This study is devoted to the implementation of a modified fast Fourier transform (MFFT) algorithm on programmable logic integrated circuits of the FPGA (Field Programmable Gate Arrays) type, which provides a maximum signal-to-noise ratio when processing non-equidistant pulse sequences. This algorithm is based on changing the FFT structure by introducing additional phase rotation blocks that take into account the wobble of the pulse repetition period of non-equidistant pulse sequences. The modified fast Fourier transform algorithm is described in Verilog HDL and tested on an FPGA board – Altera Cyclone IV.

The aim of this work is to implement FFT and MFFT algorithms on FPGA for processing non-equidistant pulse sequences. A comparative analysis of the obtained experimental results of calculating the FFT and MFFT with the results of computer calculations was carried out.

Key words: multi-channel frequency filter, non-equidistant pulse sequence, fast Fourier transform, FPGA, Altera Cyclone IV.

Ключевые слова: многоканальный частотный фильтр, неэквидистантная последовательность импульсов, быстрое преобразование Фурье, ПЛИС, FPGA, Altera Cyclone IV.

Введение

При обработке радиолокационных сигналов возникают проблемы, связанные с эффектами слепой скорости и слепой дальности. Для их устранения используют сигналы с переменном периодом повторения импульсов или неэквидистантные последовательности импульсов (НПИ) [1-4]. Кроме того, НПИ позволяют расширять диапазон однозначного измерения частоты Доплера [5-7]. Применение БПФ для многоканальной частотной фильтрации (МЧФ) позволяет существенно сократить вычислительные затраты по сравнению с дискретным преобразованием Фурье (ДПФ). Параметры многоканальной фильтрации обнаружителя рассмотрены в ряде работ [8-11]. Однако применение классического БПФ для обработки НПИ приводит к «перетеканию» части энергии сигнала в побочные частотные каналы МЧФ [12-14]. Возникающие при этом энергетические потери может компенсировать модифицированный или комбинированный (совместно классический и модифицированный) алгоритмы БПФ, которые описаны в [15-16].

Реализация алгоритмов БПФ многократно выполнялась с использованием процессоров общего назначения, цифровых сигнальных процессоров (ЦСП) и специализированных микросхем процессоров БПФ. Однако по мере увеличения емкости, повышения производитель-

Процессор быстрого преобразования Фурье (БПФ) широко используется для выполнения разнообразных радиотехнических задач, в частности, для многоканальной частотной фильтрации (МЧФ) сигналов с неизвестной частотой. Однако применение классического БПФ для обработки неэквидистантных последовательностей импульсов (НПИ) не обеспечивает оптимальности обработки по энергетическому критерию отношения сигнал-шум. Данное исследование посвящено реализации на программируемых логических интегральных схемах (ПЛИС) типа FPGA (Field Programmable Gate Arrays) алгоритма модифицированного быстрого преобразования Фурье (МБПФ), обеспечивающего максимум отношения сигнал-шум при обработке НПИ. Этот алгоритм основан на изменении структуры БПФ путем введения дополнительных блоков фазового поворота, учитывающих вобуляцию периода повторения импульсов НПИ. Алгоритм МБПФ описан на языке Verilog HDL и протестирован на плате ПЛИС FPGA - Altera Cyclone IV.

Целью работы является реализация алгоритмов БПФ и МБПФ в ПЛИС FPGA для обработки неэквидистантных последовательностей импульсов. Проведен сравнительный анализ полученных экспериментальных результатов вычисления БПФ и МБПФ с результатами компьютерного вычисления.

ности и снижения стоимости полевых программируемых вентильных ПЛИС (FPGA) они стали перспективным решением для выполнения задач, требующих значительных вычислительных затрат, в том числе таких, как вычисление БПФ [17]. Применение ПЛИС особенно целесообразно в системах, требующих многоканальной обработки данных или многоступенчатой фильтрации, а также для реализации сложных параллельных алгоритмов [18]. Параллельные алгоритмы цифровой обработки сигналов (ЦОС) обеспечивают более высокое быстродействие обработки в сложных устройствах ЦОС по

сравнению с последовательными алгоритмами [19]. Поэтому программируемые пользователем вентильные матрицы FPGA являются привлекательными платформами для высокоскоростной обработки сигналов.

Реализация алгоритмов БПФ на ПЛИС-FPGA известна из ряда публикаций. В [20-21] рассмотрена реализация алгоритма БПФ параллельной и последовательной архитектур. Особенности реализации алгоритма БПФ, а также конвейерная архитектура реализации описаны в [22]. Архитектура БПФ, основанная на использовании распределенной памяти, служит для максимизации пропускной способности и минимизации использования «справочной таблицы»-look up table (LUT) и регистров в [23]. В [24] реализован алгоритм БПФ для обработки радиолокационных сигналов с использованием оператора с плавающей запятой для уменьшения погрешности вычислений, связанных с ограниченным динамическим диапазоном представления чисел в формате с фиксированной запятой.

В предлагаемой работе проведена реализация алгоритмов БПФ в ПЛИС FPGA для обработки неэквидистантных последовательностей импульсов. Представлен проект такого алгоритма на плате FPGA – Altera Cyclone IV и проведен сравнительный анализ полученных FPGA результатов вычисления и результатов вычислений в пакете MATLAB.

Структура алгоритма МБПФ неэквидистантных импульсных сигналов в многоканальном частотном фильтре

Спектральные отсчеты N -точечного сигнала определяются классическим алгоритмом ДПФ с прореживанием по времени как:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{2nk}, \quad (1)$$

$$\text{где } W_N^k = e^{-j2\pi \frac{k}{N}} = \cos\left(2\pi \frac{k}{N}\right) - j \sin\left(2\pi \frac{k}{N}\right).$$

Алгоритм МБПФ полученный в работе [16] применительно к неэквидистантным импульсным последовательностям имеет вид:

$$X(k) = \sum_{n=0}^{\frac{N}{2}-1} x_{2n} W_N^{2nk} + e^{j2\pi v \frac{k}{N}} W_N^k \sum_{n=0}^{\frac{N}{2}-1} x_{2n+1} W_N^{2nk}, \quad (2)$$

где комплексный множитель $e^{j2\pi v \frac{k}{N}}$, определяющий дополнительный фазовый поворот отсчетов, связанный с параметром вобуляции $v = (T_0 + T_1) / T_0$, $T_0 = (T_1 + T_2) / 2$ – средний период повторения импульсов, а T_1, T_2 – чередующиеся периоды повторения импульсов.

Алгоритм Кули-Тьюки эффективно обеспечивает вычисление БПФ для количества отсчетов равном степени 2. Базовая операция Radix-2 «бабочка» является ядром алгоритма БПФ и сводится к вычислению суммы двух комплексных чисел и их разности с последующим умножением на коэффициент вращения (рис. 1). Из (2) видно, что на заключительном этапе алгоритма МБПФ коэффициент вращения модифицируется дополнительным умножением (рис. 2):

$$\begin{cases} W_{v,N}^k = e^{j2\pi v \frac{k}{N}} W_N^k = e^{-j2\pi \frac{k}{N}(1-v)}, \\ W_{v,N}^{k+\frac{N}{2}} = e^{-j2\pi \frac{k+\frac{N}{2}}{N}(1-v)} = -W_{v,N}^k \cdot e^{j\pi v}, \end{cases}$$

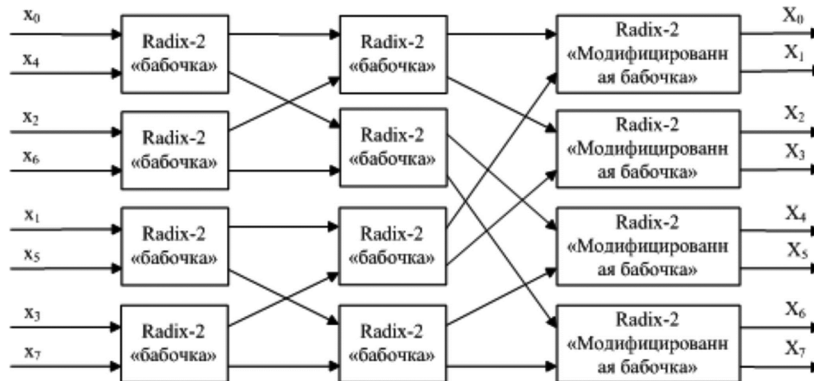


Рис. 1. Структура 8-го точечного БПФ на основании временного прореживания Radix-2 «бабочка»

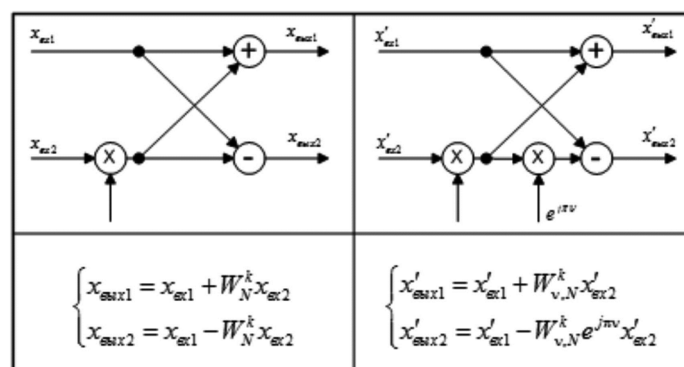


Рис 2. Граф вычисления операции Radix-2 «бабочка» (левый) и модифицированной операции «бабочка» (правый)

Таблица 1. Ресурсы семейства устройств Cyclone IV E

Устройство	Логический элемент (ЛЭ)	Встроенная память (кбит)	Встроенные умножители (18*18)
EP4CE6	6272	270	15
EP4CE10	10320	414	23
EP4CE15	15408	504	56
EP4CE22	22320	594	66
EP4CE30	28848	594	66
EP4CE40	39600	1134	116
EP4CE55	55856	2340	154
EP4CE75	75408	2745	200
EP4CE115	114480	3888	266

Таблица 2. Ресурсы семейства устройств Cyclone IV GX

Устройство	Логический элемент (ЛЭ)	Встроенная память (кбит)	Встроенные умножители (18*18)
EP4CGX15	14400	540	0
EP4CGX22	21280	756	40
EP4CGX30	29440	1080	80
EP4CGX50	49888	2502	140
EP4CGX75	73920	4158	198
EP4CGX110	109424	5490	280
EP4CGX150	149760	6480	360

Архитектура алгоритма БПФ на ПЛИС типа FPGA

Реализация БПФ на ПЛИС типа FPGA может выполняться на базе параллельной либо последовательной архитектуры, выбор которой зависит от конкретного приложения и ресурсов используемого чипа. Параллельная архитектура обеспечивает более высокое быстродействие, но при этом требуется большая логическая область. С другой стороны, последовательная архитектура часто используется в общих приложениях. При параллельной реализации на каждом этапе алгоритма МБПФ параллельно вычисляют операцию «бабочка». Этапы при этом обрабатываются последовательно и промежуточные значения хранятся в интегрированном оперативном запоминающем устройстве (ОЗУ) [20]. Однако такая реализация параллельной архитектуры требует большого количества встроенных умножителей и объема ОЗУ. Ресурсы семейства устройств Altera Cyclone IV [25] представлены в табл. 1 и табл. 2.

Каждое комплексное умножение эквивалентно 4 действительных умножений, поэтому на каждом этапе вычисления МБПФ кроме заключительного этапа содержится $2N$ операций действительного умножения и $4N$ операций действительного умножения на заключительном этапе. Подробнее вычислительные затраты МБПФ НПИ проанализированы в [26]. В FPGA умножение обычно реализуется с помощью встроенных умножителей в блоке ЦОС (DSP). Количество встроенных умножителей ограничено поэтому необходимо эффективно использовать их. Из табл. 1 и табл. 2 видно, что при $N > 256$ полная параллельная реализация данного модифицированного алгоритма БПФ не может быть выполнена на семействе Altera Cyclone IV.

С целью проверки возможности реализации МБПФ НПИ на ПЛИС FPGA использована последовательная архитектура алгоритма БПФ. Последовательная реали-

зация алгоритма БПФ требует только один блок radix-2 «бабочка». Такой блок используется для каждого коэффициента вращения, что позволяет преодолеть недостаточное количество логических вентилях и встроенных умножителей. Результаты промежуточных вычислений блока radix-2 сохраняются в ОЗУ, количество циклов повторения вычислений для N отсчетов равно $N \log_2(N) / 2$. Каждый цикл включает 2 такта для загрузки данных в блок «бабочка», поэтому результат будет получен за $N \log_2(N)$ тактов. Хотя при последовательной обработке требуется меньшая логическая область, скорость обработки данных оказывается значительно ниже, чем параллельной реализации. На рис. 3 представлена последовательная архитектура БПФ, которая может использоваться для его реализации на плате Cyclone IV EP4CE6.

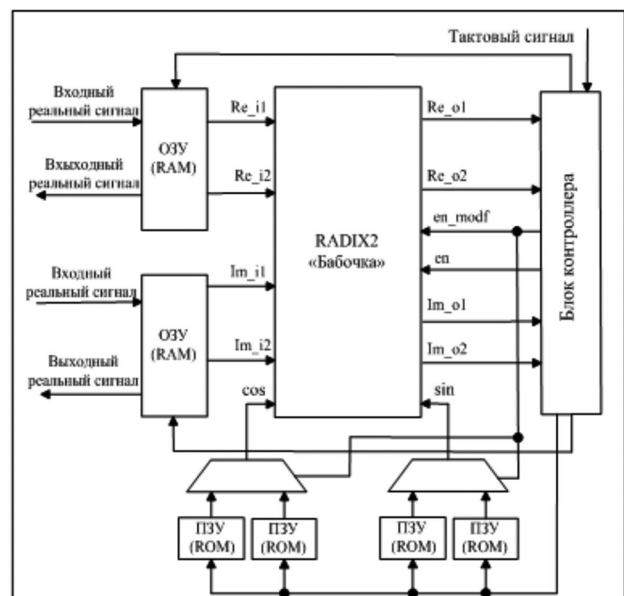


Рис. 3. Параллельная архитектура алгоритма БПФ

Таблица 3. Нормированные модифицированные коэффициенты вращения для разных коэффициентов

k	$W_{v,N}^k$	10-бит	12-бит	14-бит	16-бит
0	1+0j	256+0j	1024+0j	4096+0j	16384+0j
1	1-0,0049j	256-1j	1024-5j	4096-20j	16384-80j
2	1-0,0098j	256-3j	1024-10j	4096-40j	16383-161j
3	0,9999-0,0147j	256-4j	1024-15j	4096-60j	16382-241j
4	0,9998-0,0196j	256-5j	1024-20j	4095-80j	16381-322j
5	0,9997-0,0245j	256-6j	1024-25j	4095-101j	16379-402j
507	0,3207-0,9472j	82-22j	328-969j	1313-3879j	5253-15519j
508	0,3183-0,9480j	81-242j	325-970j	1303-3882j	5215-15532j
509	0,3160-0,9488j	80-242j	323-971j	1294-3886j	5177-15544j
510	0,3137-0,9495j	80-243j	321-972j	1284-3889j	5139-15557j
511	0,3114-0,9503j	79-243j	318-973j	1275-3892j	5101-15570j

В зависимости от этапа вычисления БПФ следует применить обычное вычисление или модифицированное вычисление в блоке «бабочка». Также в каждом цикле вычисления необходимо выполнять 2 операции комплексного умножения. Поэтому для обеспечения параллельного вычисления в блоке «бабочка» необходимо использовать 2 подблока комплексного умножения и блок суммирования.

Блок ОЗУ предназначен для загрузки исходных данных, а также сохранения промежуточных значений блока «бабочка» и выгрузки данных. Блоки постоянного запоминающего устройства (ПЗУ) сохраняют коэффициенты вращения W_N^k и модифицированных коэффициентов вращения $W_{v,N}^k$ и передают эти коэффициенты через мультиплексор в блок «бабочка» в соответствии с этапом вычисления.

Блок контроллера генерирует адреса, чтобы передавать данные с блоков ОЗУ и блоков ПЗУ в блок «бабочка» и обратно. Промежуточный результат вычислений с блока «бабочка» передается в блок контроллера и обратно вместе с адресами для синхронизированной записи в блоках ОЗУ. Процессы чтения и записи данных осуществляются одновременно с использованием двухпортового ОЗУ. Блок контроллера управляет чтением данных с блоков ПЗУ и выбором коэффициентов вращения и модифицированного коэффициента вращения в соответствии вычислениями в блоке «бабочка». Также блок контроллера генерирует сигналы для старта и стопа вычислений, необходимые при выгрузке результатов.

Для того чтобы быстро вычислять умножение необходимо заранее рассчитывать константы коэффициентов вращения и хранить их в ПЗУ. При выполнении операций с дробными числами в реальном времени предлагается использование арифметики с фиксированной запятой, что уменьшит требования к оборудованию и времени обработки. При этом коэффициенты вращения необходимо нормализовать, а нормированные значения любых двух соседних точек должны быть различимы, что позволит определить необходимое количество битов для представления чисел с фиксированной запятой [21]. Использование различных способов нормирования коэффициентов для нормализации приведет к различным ошибкам округления. На табл. 3 для примера представлены первые и последние 5 нормированных коэф-

фициентов вращения со степенью вобуляции $v = 0,2$ для 1024-го точечного алгоритма МБПФ. В зависимости от требования точности округления можно выбрать нормированный коэффициент, что рассмотрено ниже при анализе результатов.

Реализация алгоритма БПФ на плате Altera Cyclone IV

Для реализации алгоритма МБПФ на конкретной ПЛИС необходимо определить требуемые ресурсы для использования: встроенных умножителей, сумматоров блока «бабочка», регистров, логических элементов, а также объем памяти для хранения промежуточных значений и коэффициентов вращения.

Реализация 1024-го точечного МБПФ выполняется в 10 этапов. Необходим также дополнительный предварительный этап загрузки комплексных входных данных в блоки ОЗУ в соответствии с инверсным битом адреса и этап выгрузки выходных данных из ОЗУ. Вычисления выполняются в блоке «бабочка» (RADIX2), структура которого представлена на рис. 4. Входные сигналы блока RADIX2 включают: комплексные отсчеты НПИ и коэффициенты вращения, а также управляющий сигнал (en_modf) для выбора выхода блока и управляющий сигнал (en) для синхронизации вычислений.

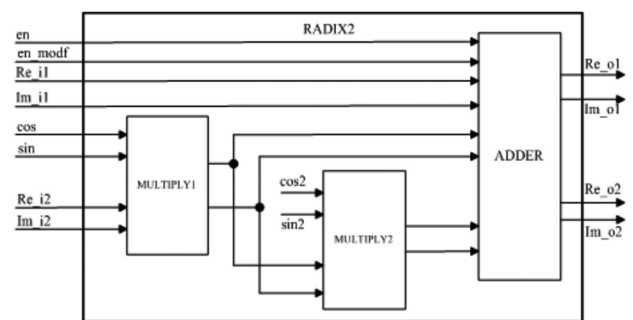


Рис. 4. Структура блока RADIX2 «бабочка»

В отличие от обычного блока «бабочка» используемый в МБПФ блок RADIX2 имеет два блока комплексного умножения и один блок суммирования. Входные сигналы блока RADIX2 с блоков двухпортового ОЗУ поступают в первый блок комплексного умножения (MULTIPLY1), где происходит перемножение входного сигнала с основными коэффициентами вращения. Выходные сигналы блока MULTIPLY1 поступают в блок суммирования (ADDER) и во второй блок комплексного

умножения (MULTIPLY2), в котором вычисляется произведение с дополнительным коэффициентом вращения. Дополнительный множитель $e^{j\pi v}$ поступает во второй блок комплексного умножения, который неизменен для всех циклов и предварительно назначен в блоке RADIX2. Все результаты произведений поступают в блок ADDER для суммирования (и вычитания) и выбора выходного сигнала блока. Структура блоков комплексного умножения одинакова. Она содержит 4 встроенных умножителя, 2 сумматора и 2 делителя, выполняемых оператором сдвига и представлена на рис. 5.

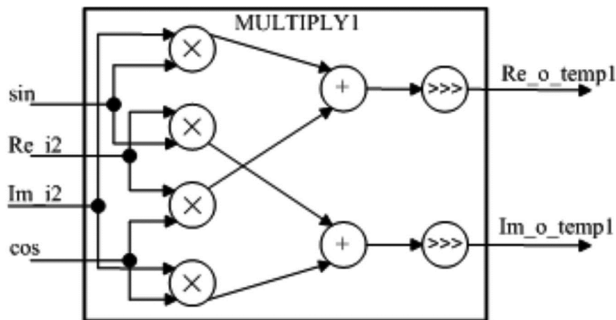


Рис. 5. Структура блока комплексного умножения

Структура блока суммирования (вычитания) изображена на рис. 6, который состоит из 6 сумматоров, 2 мультиплексора и 4 защелки. Выходные сигналы блоков комплексного умножения и выходные сигналы блока ОЗУ поступают в блок суммирования ADDER, в котором выполняются вычисления согласно формулам на рис. 2. Выходные сигналы определяются мультиплексорами в зависимости от этапа вычисления алгоритма БПФ. Также выходные сигналы считываются с учетом синхронизации с помощью защелок.

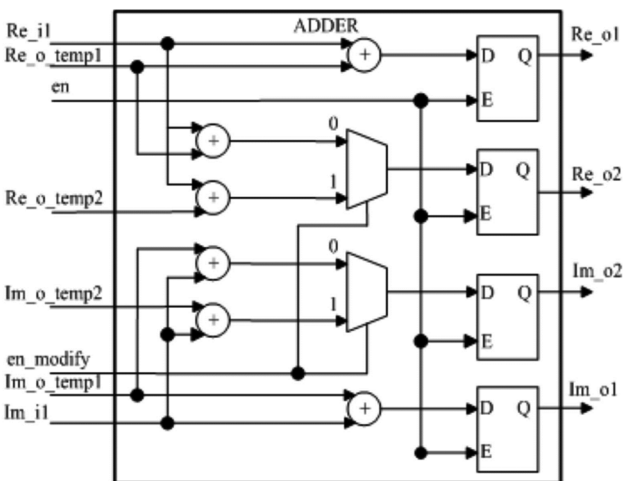


Рис. 6. Структура блока суммирования (вычитания)

Результат синтезирования и проектирования

В этом разделе представлены логические ресурсы, используемые в FPGA для реализации БПФ, а также результаты испытаний предлагаемого модифицированного алгоритма БПФ для обработки НПИ. Язык Verilog HDL использован для описания данного алгоритма, моделирования алгоритмов с помощью программного обеспечения Modelsim и программирования алгоритмов для аппаратно-программной реализации на плате платы с помощью программного обеспечения Quartus Prime

19.1 Little edition. В качестве чипа FPGA использован Cyclone IV EP4CE6 с тактовой частотой 50 МГц (рис. 7). Этот чип в семействе Cyclone IV E не обеспечивает полностью параллельной реализации, но подходит для проверки работоспособности алгоритма МБПФ на плате FPGA. Полученный результат из FPGA сравнивается с результатом вычисления в пакете MATLAB.

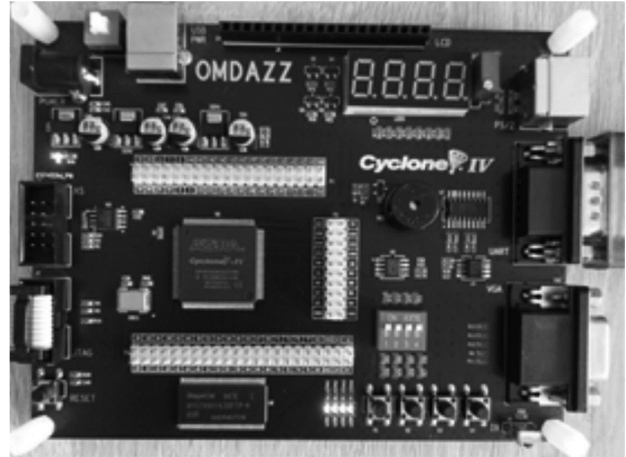


Рис. 7. Плата FPGA на базе микросхемы чипа Altera Cyclone IV EP4CE6

В качестве тестирования использовалась арифметика с фиксированной запятой для представления сигналов и коэффициентов вращения. Сигналы представлены 24 битами со знаком, в том числе 8 битов использованы для дробной части и 15 битов для целой части, 1 бит - для знака. Коэффициенты вращения нормированы на 16 битов (со знаком). В табл.4 представлены ресурсы при реализации алгоритма МБПФ обработки НПИ при разном количестве отсчетов, а в табл. 5 представлено времени обработки.

Таблица 4. Ресурсы для реализации алгоритма МБПФ

N	Логический элемент (ЛЭ)		Встроенная память (кбит)		Встроенные умножители (18*18)	
	Число	Процент	Число	Процент	Число	Процент
128	1440	22,96%	6144	2,22%	15	100%
256	1717	27,38%	12288	4,44%	15	100%
512	2186	34,85%	24576	8,88%	15	100%
1024	3093	49,31%	49152	17,78%	15	100%

Таблица 5. Время обработки алгоритма МБПФ с тактовой частотой 50 МГц

N	128	256	512	1024
t, мкс	17,9	41	92,2	204,8

Из табл. 4 видно, что с увеличением количества отсчетов требуется больший объем памяти для хранения сигналов и большее количество ЛЭ для хранения постоянных коэффициентов вращения. Использован только один блок RADIX2 при обработке, поэтому количество требуемых встроенных умножителей не изменяется. Табл. 5 показывает, что время обработки увеличивается с увеличением количества отсчетов в соответствии с логарифмической зависимостью $t = N \log_2(N) / f$.

На табл. 6 представлены результаты вычисления первых 10 отсчетов из FPGA и результаты вычисления в пакете MATLAB при реализации 1024-го точечного алгоритма МБПФ. Степень вобуляции периода повторения импульсов сигнала $v = 0, 2$. В качестве входного сигнала

использована аддитивная смесь сигнала и белого гауссовского шума с отношением сигнал-шум 15 дБ.

Таблица 6. Результаты вычисления МБПФ из MATLAB и FPGA

n	MATLAB	FPGA
0	1314-j1764	1314-j1764
1	505,46-j849,43	504,69-j849,72
2	348,33-j643,06	348-j643,20
3	233,09-j411,09	232,52-j410,99
4	328,45-j268,33	328,25-j268,43
5	171-j354,30	170,73-j354,53
6	189,47-j133,70	189,13-j133,69
7	197,39-j147	197,05-j147,04
8	111,32-j92,29	111,23-j92,33
9	224,09-j126,32	223,91-j126,49

На рис. 8 показаны ошибки (отклонения) результатов вычисления из FPGA и результатов вычислений из MATLAB, где Δ_{Re} и Δ_{Im} соответствуют отклонениям

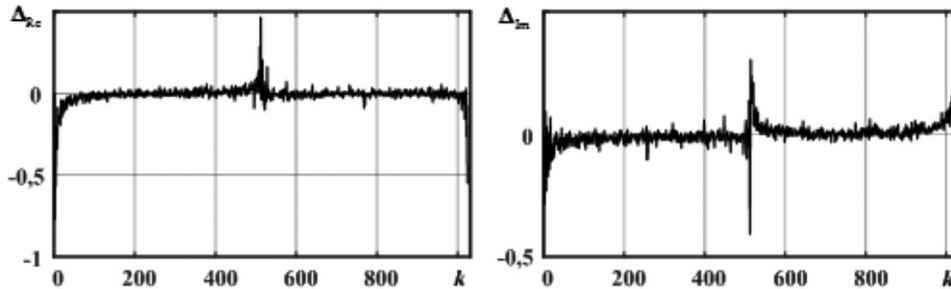


Рис. 8. Ошибки (отклонения) результатов вычисления из FPGA и результатов вычислений из MATLAB

действительных и мнимых результатов вычисления, k -номера отсчетов. Можно видеть, что различия отклонений между результатами FPGA и MATLAB очень малы по сравнению с результатом вычисления. Эти ошибки возникают в процессе округления коэффициентов вращения, а также битовом переполнении дробной части числа с фиксированной запятой.

Для уменьшения этих ошибок предлагается нормирование коэффициентов вращения в сторону повышения двоичного порядка (сдвигом влево), а также расширение диапазона представления чисел с фиксированной запятой. В табл. 7 представлены среднеквадратические ошибки (СКО) результатов вычисления МБПФ из FPGA и отношение сигнал-шум квантования при разных нормированных коэффициентов, где σ_{Re} , σ_{Im} соответствуют СКО по всем отсчетам действительной и мнимой части и $E_c / (\sigma_{Re}^2 + \sigma_{Im}^2)$ – отношение сигнал-шум квантования. Чем больше нормированный коэффициент вращения, тем меньше СКО по всем отсчетам, но при этом необходимо использовать большие ресурсы для обработки.

На рис. 9. показаны амплитуды отсчетов A_{mod} , полученные из FPGA с учетом вобуляцией периода повто-

Таблица 7. СКО по всем отсчетам и отношения сигнал-шум квантования

$W_{v,N}^k, W_N^k$	6-бит	10-бит	12-бит	14-бит	16-бит
σ_{Re}	22,48	3,33	0,92	0,22	0,05
σ_{Im}	12,58	6,59	1,68	0,33	0,04
$E_c / (\sigma_{Re}^2 + \sigma_{Im}^2)$, дБ	28,46	39,31	51,02	64,67	79,79

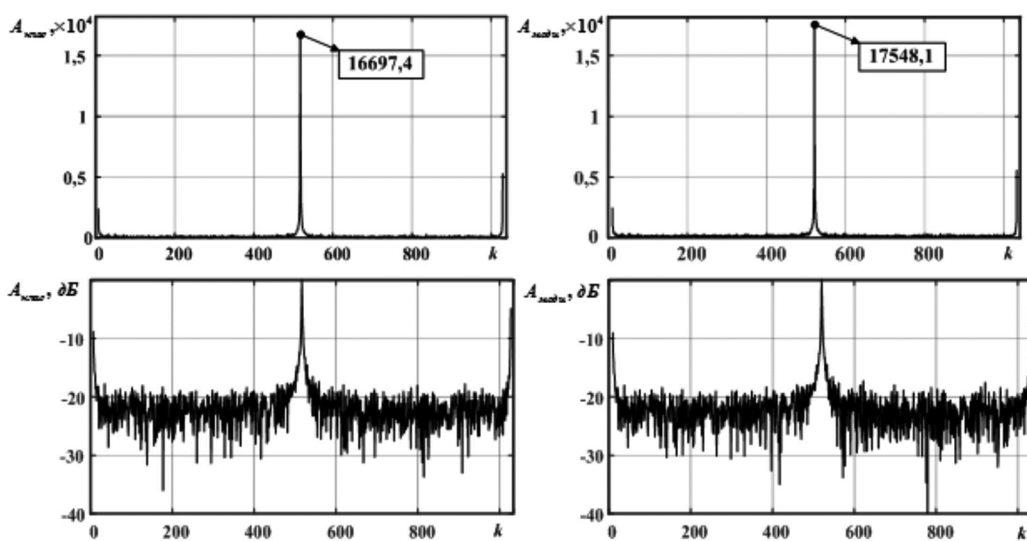


Рис. 9. Амплитуда отсчетов из FPGA по классическому алгоритму (левые) и модифицированному алгоритму БПФ (правые)

рения импульсов и амплитуды отсчетов $A_{\text{клас}}$, вычисленные с использованием классического алгоритма БПФ (1).

Из рис. 9 видно, что применение модифицированного алгоритма повышает амплитуду пика на 5 %, в случае $\nu = 0,2$ для номера частотного отсчета сигнала, равного 511. Это улучшение еще более возрастает при увеличении степени вобуляции периода [13-14], что позволяет увеличить отношение сигнал-шум на выходе МБПФ. Возрастание амплитуды частотного отсчета эквивалентно в радиолокации увеличению дальности обнаружения приблизительно на 2,5 %.

Заключение

В работе рассмотрена архитектура реализации алгоритма МБПФ для обработки НПИ в ПЛИС типа FPGA. Данная архитектура не обеспечивает высокого быстродействия, но требует очень малый ресурс чипа. Результат тестирования показал, что отношение сигнал-шум квантования увеличивается с увеличением нормированного коэффициента вращения, что при нормировании на 16 бит эквивалентно значению ~ 79 дБ. Полученный результат согласуется с теоретическим и подтверждает, что при обработке НПИ на фоне белого гауссовского шума применение алгоритма МБПФ превосходит классический алгоритм БПФ, т.к. амплитуда когерентно накопленного сигнала увеличивается на 5 %. Данный результат имеет большой смысл при проектировании реального процессора в радиотехнических приложениях, использующих НПИ, а также для дальнейшего исследования оптимизации времени обработки НПИ с алгоритмом МБПФ на ПЛИС типа FPGA.

Работа выполнена в рамках государственного задания Министерства науки и высшего образования РФ (FSSN – 2020 – 0003).

Литература

1. Thomas H.W., Abram T.M. Stagger period selection for moving-target radar. Proc. IEEE. 1976, vol. 123, no. 3, pp. 195-199. DOI: 10.1049/piee.1976.0045.
2. Профатилова Г.А., Соловьев Г.Н., Ефремов В.С., Соловьев А.Г. Повышение эффективности систем селекции движущихся тел в рлс управления воздушным движением. Вестник МГТУ им. Н.Э. Баумана. Сер. «Приборостроение». 2012. № 3. С. 87-92.
3. Справочник по радиолокации / Под ред. М.И. Сколника. Пер. с англ. под общей ред. В.С. Вербы. В 2 книгах. Книга 1. Москва: Техносфера, 2014. 672 с.
4. Бакулев П.А. Радиолокационные системы. М.: Радиотехника, 2015. 437 с.
5. Богатов А.Д., Костров В.В., Терсин В.В. Корреляционно-фильтровое обнаружение и измерение доплеровского смещения частоты неэквидистантной последовательности фазокодированных сигналов. Методы и устройства передачи и обработки информации. 2008. № 10. С. 136-143.
6. Жиганов С.Н. Частотные свойства устройств череспериодной компенсации при обработке неэквидистантных последовательностей импульсов. Радиотехнические и телекоммуникационные системы. 2013. № 10. С. 44-49.
7. Манукьян А.А. Построение двумерных радиоизображений объектов по неэквидистантным частотно-временным по-

следовательностям импульсов. Радиотехника и электроника. 2015. Т. 60. № 3. С. 282-296.

8. Кошелев В.И. Параметры многоканального обнаружителя доплеровских сигналов. Вестник РГРТА. 1997. № 6. С. 32.

9. Кошелев В.И. Параметры многоканального обнаружителя доплеровских сигналов (статья). Вестник РГРТА. Рязань. 2001. № 8. С. 18-20.

10. Кошелев В.И., Белокуров В.А. Выбор числа каналов обнаружителя маневрирующей цели. Вестник РГРТУ 2006. № 18. С. 26-28.

11. Попов Д.И. Обработка неэквидистантных сигналов на фоне пассивных помех. Вестник РГРТУ. 2022. № 80. С. 24-31.

12. Кошелев В.И. Когерентная фильтрация неэквидистантных последовательностей импульсов в системах первичной обработки радиолокационных систем. Успехи современной радиоэлектроники. 2014. № 10. С. 16-22.

13. Кошелев В.И., Чинь Н.Х. Эффективность многоканальной доплеровской фильтрации неэквидистантных последовательностей импульсов. Цифровая обработка сигналов. 2023. № 2. С. 3-8.

14. Кошелев В.И., Чинь Н.Х. Оптимизация алгоритма весовой обработки в многоканальной доплеровской фильтрации. Известия высших учебных заведений России. Радиоэлектроника. 2024. Т. 27. № 2. С. 93-104. <https://doi.org/10.32603/1993-8985-2024-27-2-93-104>.

15. Букварев Е.А., Рябков А.П. Оптимизация многоканального когерентного накопителя пачки импульсов при вобуляции периода зондирования. Труды НГТУ им. Р.Е. Алексеева. 2012. № 2 (95). С. 31-40.

16. Кошелев В.И., Чинь Н.Х. Алгоритм быстрого преобразования Фурье неэквидистантных последовательностей импульсов. Вестник РГРТУ. 2023. № 85. С. 3-13.

17. Uzun I.S., Bouridane A.A.A. FPGA implementations of fast fourier transforms for real-time signal and image processing. Proceedings. 2003 IEEE International Conference on Field-Programmable Technology (FPT) (IEEE Cat. No.03EX798), Tokyo, Japan, 2003, pp. 102-109, DOI: 10.1109/FPT.2003.1275737.

18. Черемисин А.Г. Оценка эффективности применения ПЛИС и процессоров DSP для задач цифровой обработки сигналов. Научно-технический вестник информационных технологий, механики и оптики. 2006. № 32. С. 44-47.

19. Башкиров А.В., Муратов А.В. Преимущество параллельных алгоритмов цифровой обработки сигналов над последовательными алгоритмами при реализации на ПЛИС. Вестник ВГТУ. 2012. Т. 8. № 1. С. 89-92.

20. Карпов С. Математические блоки в радиационностойких ПЛИС Actel. ЭЛЕКТРОНИКА: НТБ. 2009. № 1. С. 36-40.

21. Polat. G.; Ozturk. S.; Yakut. M. Design and Implementation of 256-Point Radix-4 100 Gbit/s FFT Algorithm into FPGA for High-Speed Applications. ETRI J. 2015, no. 37, pp. 667-676.

22. Иванова Н.Н., Галанина Н.А., Моисеев Д.В. Особенности реализации алгоритма БПФ на плис типа FPGA. Вестник Чувашского университета. 2018. № 3. С. 182-191.

23. Nash. J.G. Distributed-Memory-Based FFT Architecture and FPGA Implementations. Electronics. 2018. No. 7, 116. <https://doi.org/10.3390/electronics7070116>.

24. Neo. J., Jung. Y., Lee. S., Jung. Y. FPGA Implementation of an Efficient FFT Processor for FMCW Radar Signal Processing. Sensors 2021. No. 21, 6443. <https://doi.org/10.3390/s21196443>

25. Datasheet, Cyclone IV Device Handbook, vol. 1. Altera. 2013.

26. Кошелев В.И., Чинь Н.Х. Оценка вычислительных затрат алгоритмов БПФ неэквидистантной последовательности отсчетов. VII Международный научно-технический форум СТНО-2024. Сборник трудов. Т. 1. С. 30-35.