

# Реализация обратного преобразователя модулярной арифметики, совмещенного с операцией округления для задач ЦОС

В.М. Амербаев, Д.В. Тельпухов, Е.С. Балака, А.В. Константинов

Федеральное государственное бюджетное учреждение науки Институт проблем проектирования в микроэлектронике Российской академии наук, Nofrost@inbox.ru

**Аннотация** — Описан подход в организации вычислений в целых числах на базе аппарата модулярной арифметики. Получена формула для вычисления наименьшего динамического диапазона для реализации БПФ в режиме точных вычислений на основе кодов вычетов по попарно взаимно-простым модулям. Предложен метод совмещения операции обратного преобразования и операции финального округления, что вкупе с режимом вычислений в целых числах позволяет строить устройства ЦОС с улучшенными характеристиками точности и быстродействия.

**Ключевые слова** — модулярная арифметика, полиадический код, быстрое преобразование Фурье, обратный преобразователь, цифровая обработка сигналов.

## I. ВВЕДЕНИЕ

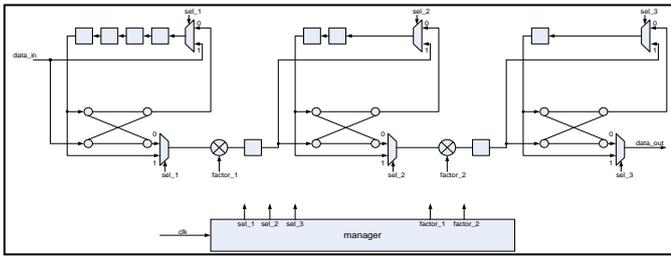
Отличительной чертой задач цифровой обработки сигналов является поточный характер обработки больших объемов данных в реальном масштабе времени, требующий от технических средств высокой производительности и возможности интенсивного обмена с внешними устройствами. Это достигается в настоящее время благодаря специфической архитектуре процессоров цифровой обработки сигналов (ЦОС), называемой базовой архитектурой. Базовая архитектура – это совокупность характерных особенностей процессора, направленная на повышение его производительности и отличающая процессоры ЦОС от микросхем других типов. В значительной мере базовая архитектура обусловлена как широким использованием конвейерного режима работы [1], так и параллельными структурами обработки информации [2]. В то время как конвейеризация алгоритмов ЦОС достигается разработкой различных конвейерных архитектур, нужный уровень параллелизма может быть достигнут благодаря использованию аппарата модулярной арифметики. Модулярная арифметика позволяет производить распараллеливание трактов обработки данных без какого-либо изменения существующих технологий. Однако, такие важные операции машинной арифметики, как деление, формирование признака реполнения, округление, перевод из одной

системы счисления в другую, а также алгоритмы декодирования системы самокоррекции являются существенно последовательно - параллельными. Перечисленные операции принято называть немодульными.

Острая необходимость в повышении точности и быстродействия устройств ЦОС приводит к идее исследования устройств в режиме точных вычислений на базе модулярной арифметики. Суть метода заключается в выборе модулей, произведение которых способно покрыть динамический диапазон рассматриваемой арифметической функции. Ускорение за счет исключения затратных операций округления, а также увеличение точности за счет исключения ошибок округления являются преимуществами данного подхода. К недостаткам можно отнести увеличение аппаратных затрат, а также существенное увеличение вычислительной нагрузки на обратный преобразователь. И если увеличение аппаратуры является прямым следствием расширения динамического диапазона, то нагрузку на обратный преобразователь можно уменьшить путем совмещения его с операцией округления

## II. БПФ В ЦЕЛЫХ ЧИСЛАХ

Актуальность данной работы тесно связана с разработкой ускоренного преобразователя быстрого преобразования Фурье (БПФ) конвейерной структуры с использованием модулярных систем кодирования. Конвейерные БПФ процессоры – это специфический класс процессоров для вычисления дискретного преобразования Фурье (ДПФ), который эксплуатирует так называемые быстрые алгоритмы, одним из которых является алгоритм Кули-Тьюки. Этот класс характеризуется непрерывной обработкой информации в реальном времени, по мере того как информация проходит через процессор. Существует множество различных архитектур построения конвейерных процессоров БПФ, отличающихся структурой так называемых «бабочек» [1], а также наличием или отсутствием обратных связей. Для построения БПФ в системе остаточных классов была выбрана архитектура Radix-2 Single-path Delay Feedback [3], т.к. она имеет наименьший расход памяти и наиболее простой механизм управления [1] (рис. 1).



**Рис. 1. Архитектура БПФ - Radix-2 Single-path Delay Feedback**

Для реализации БПФ в системе остаточных классов (СОК) необходимо создать несколько параллельных каналов данной архитектуры, в которых все арифметические операции реализованы в остатках. Параллельная структура модулярной арифметики позволяет наращивать динамический диапазон вычислений, просто добавляя модульные каналы, что напрямую не ухудшает производительность схемы. Таким образом, мы приходим к идее реализации БПФ в режиме точных вычислений (или что то же самое – вычислений в целых числах). Парадигма такого рода вычислений подразумевает выбор динамического диапазона, который полностью покрывает результаты вычислений БПФ без промежуточных округлений [4].

Наименьший динамический диапазон  $[0, M)$ , необходимый для вычисления БПФ в целых числах, оценивается величиной:

$$M = 2^{k_1 + (\log_2 N - 1)(1 + k_2)} \quad (1)$$

где  $N$  – размер БПФ преобразователя,  $k_1$  – разрядность входных данных,  $k_2$  – разрядность поворачивающих множителей.

Таким образом, для реализации 256 точечного модулярного БПФ с разрядностью входных данных и поворачивающих коэффициентов 16 бит, получаем  $M = 2^{7 \cdot 17 + 16} = 2^{135}$ . Таким образом, наш динамический диапазон должен покрывать 135 битные выходные данные. Покрыть данный диапазон можно 8-битным балансным набором модулей, так как, вообще говоря, 8-битных простых чисел, как оснований модулярной арифметики, хватит, чтобы покрыть динамический диапазон в 172 бита. Таким образом, мы добиваемся вычисления 16 разрядного БПФ по нескольким параллельно работающим 8 битным каналам, обеспечивая тем самым значительное ускорение.

Значительная часть вычислительной нагрузки в данном подходе ложится на интерфейсные устройства – преобразователи из двоичной системы счисления в систему остаточных классов (прямые преобразователи) и обратно (обратные преобразователи). И если вопросы построения прямых преобразователей ориентированных на задачи ЦОС (в т. ч. БПФ) эффективно решаются [5], то вопросы, связанные с обратными преобразователями, требуют большего

внимания. Далее будет рассмотрен эффективный способ построения обратных преобразователей, совмещенных с операцией финального округления результата.

### III. ПОСТРОЕНИЕ ОБРАТНОГО ПРЕОБРАЗОВАТЕЛЯ

Предлагаемый способ реализации операции обратного преобразования базируется на переводе модулярного кода в так называемый полиадический код (также встречается термин «позиционный код со смешанными основаниями»).

Значения разрядов полиадического кода  $(z_1, z_2, \dots, z_n)$  по модулям  $m_1, m_2, \dots, m_n$  могут быть получены из модулярного кода  $(x_1, x_2, \dots, x_n)$ , соответственно по модулям  $m_1, m_2, \dots, m_n$ , решая систему сравнений:

$$z_1 = |x_1|_{m_1}$$

$$z_2 = \left| |m_1^{-1}|_{m_2} (x_2 - z_1) \right|_{m_2}$$

$$z_3 = \left| |m_2^{-1}|_{m_3} (|m_1^{-1}|_{m_3} (x_3 - z_1) - z_2) \right|_{m_3}$$

$$z_4 = \left| |m_3^{-1}|_{m_4} (|m_2^{-1}|_{m_4} (|m_1^{-1}|_{m_4} (x_4 - z_1) - z_2) - z_3) \right|_{m_4}$$

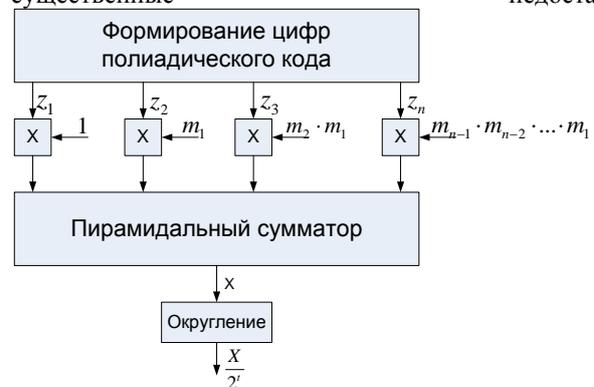
⋮

$$z_N = \left| |m_{N-1}^{-1}|_{m_N} (|m_{N-2}^{-1}|_{m_N} (\dots |m_2^{-1}|_{m_N} (|m_1^{-1}|_{m_N} (x_N - z_1) - z_2) \dots) - z_{N-1}) \right|_{m_N}$$

Исходное число в двоичной форме формируем из полученных на первой стадии цифр полиадической системы:

$$X = z_N m_{N-1} m_{N-2} \dots m_1 + \dots + z_3 m_2 m_1 + z_2 m_1 + z_1$$

Используя вышеописанный подход в организации вычислений в целых числах, на выходе устройства мы получаем число очень большой разрядности. При традиционном подходе округление конечного результата можно провести после операции обратного преобразования (рис 2.), однако этот метод имеет существенные недостатки.



**Рис. 2. Обратный преобразователь модулярного кода на базе полиадической системы счисления**

Эти недостатки выражаются в сложности реализации сумматора потенциально больших чисел, что естественным образом приводит к большим аппаратным и временным затратам.

Существенное сокращение затрат в реализации финальной стадии перевода числа может быть достигнуто при использовании метода совмещения обратного преобразователя с операцией округления.

Рассмотрим математическое обоснование корректности предлагаемого подхода. Пусть нам необходимо вычислить некоторую сумму чисел  $\sum_{i=1}^n a_i$  с последующим округлением. Округление в двоичном коде достигается отбрасыванием двоичных разрядов, что, в свою очередь, эквивалентно делению на  $2^t$ , где  $t$  - число отбрасываемых разрядов. Таким образом, искомый результат выражается формулой:

$$A = \frac{a_1 + a_2 + a_3 + \dots + a_n}{2^t}$$

Представив каждое слагаемое в виде суммы целого числа переполнений за диапазон  $b$  и остатка от деления, получим:

$$A = \frac{(q_1 \cdot 2^t + |a_1|_{2^t}) + \dots + (q_n \cdot 2^t + |a_n|_{2^t})}{2^t},$$

или

$$A = (q_1 + q_2 + q_3 + \dots + q_n) + \frac{|a_1|_{2^t} + |a_2|_{2^t} + |a_3|_{2^t} + \dots + |a_n|_{2^t}}{2^t} \quad (2)$$

Необходимо учесть, что формирование слагаемых вида  $z_i m_{i-1} m_{i-2} \dots m_1$  (при переходе от полиадической к двоичной системе счисления) происходит методом обращения в таблицу. Соответственно, вместо запоминающих устройств, хранящих полноразрядные слагаемые, можно использовать таблицы со значениями  $q_i$  и  $|a_i|_{2^t}$ . Это позволяет разбить используемый пирамидальный сумматор на два параллельных сумматора существенно меньшей разрядности (рис. 3). Таким образом, выход устройства будет равен:

$$\sum_{i=1}^n q_i + \left\lfloor \frac{\sum_{i=1}^n |a_i|_{2^t}}{2^t} \right\rfloor,$$

а ошибка округления будет равняться:

$$\partial \left( \frac{\sum_{i=1}^n |a_i|_{2^t}}{2^t} \right),$$

где  $[x]$  – целая часть  $x$ ,  $\partial(x)$  – дробная доля  $x$ .

Разрядность выходов таблиц зависит от параметра  $t$ , который, в свою очередь, зависит от необходимого количества двоичных разрядов на выходе устройства. Также следует отметить, что поскольку значение второго слагаемого в (2) ограничено сверху количеством модулей  $n$ , то в некоторых конкретных случаях вторым сумматором можно вовсе пренебречь.

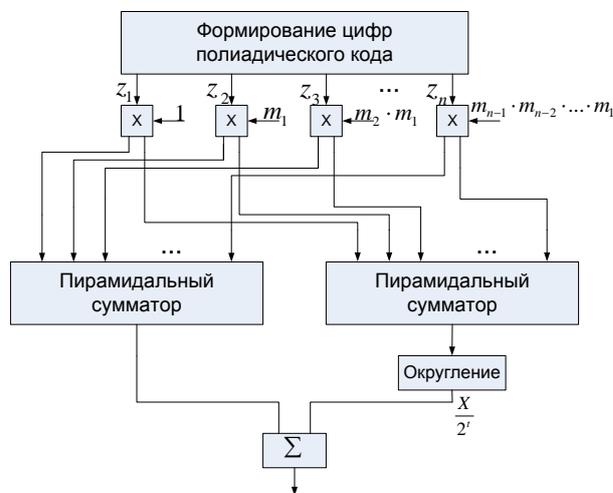


Рис. 3. Обратный преобразователь модулярного кода совмещенный с операцией финального округления

#### IV. Оценки производительности и точности

Рассмотрим оценки быстродействия предложенного алгоритма, и сравним их с оценками быстродействия традиционных схем. Для этого на языке Perl был создан генератор функциональных представлений, реализующий два типа схем по заданным параметрам. На вход генератора подается разрядность входных данных для устройства БПФ, а также длина самого преобразования. По формуле (1) рассчитывается наименьший динамический диапазон  $M$ , необходимый для вычисления БПФ в режиме точных вычислений, и подбираются модули. После этого программа создаёт два RTL - описания обратного преобразователя из модулярного кода в двоичный на языке Verilog. Первый вариант представляет собой обычный преобразователь на базе полиадической системы счисления (рис. 2). Вторая архитектура является предложенным выше обратным преобразователем модулярного кода, совмещенным с операцией финального округления (рис. 3). Синтез проводился средствами САПР Synplify Pro фирмы Synopsys в базе плис Altera STRATIX II. Результаты синтеза сведены в таблицу.

Таблица 1

## Оценки быстродействия разработанных схем обратных преобразователей

Длина БПФ преобразования	8 битные входные данные	
	Архитектура 1 (МГц)	Архитектура 2 (МГц)
3	411,6	441,5
6	340,7	379,9
8	276,4	303,3
10	275,6	321,0
12	273,1	307,0
	16 битные входные данные	
3	385,2	419,4
6	276,4	317,0
8	275,6	322,5
10	273	285,3
12	272,3	284,7

Кроме того, были проведены эксперименты по оценке точности схем БПФ с округлением после каждой стадии алгоритма, а также эксперименты по оценке точности предлагаемой схемы в режиме точных вычислений. Оценка проводилась на БПФ устройствах с разрядностью входных данных от 3 до 16 бит и длиной преобразования от 8 до 256 точек. Оценка производилась средствами Cadence NC-verilog методом эмпирического сравнения результатов преобразования на произвольных выборках с результатом вычисления соответствующего ДПФ с максимальной машинной разрядностью. Следует отметить, что в оценке точности не были рассмотрены ошибки представления входных данных и поворачивающих множителей, а лишь ошибки округления при выполнении арифметических операций.

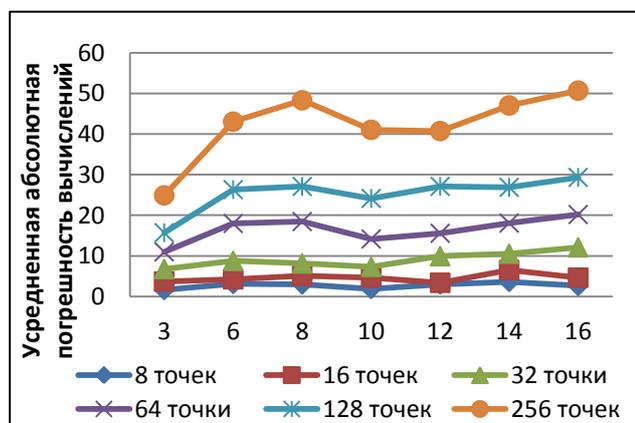


Рис. 4. Оценка точности схем БПФ с промежуточным округлением

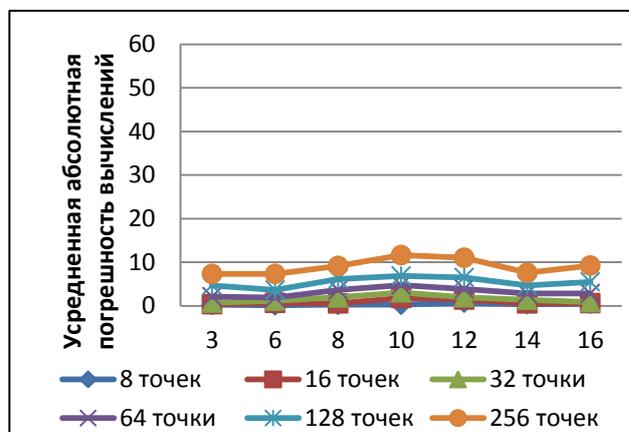


Рис. 5. Оценка точности схем БПФ без промежуточных округлений

## V. ЗАКЛЮЧЕНИЕ

Оценивая полученные экспериментальные оценки можно сделать вывод о том, что предложенный подход вычислений в целых числах на базе модулярной системы вычетов, а также метод совмещения операции обратного преобразования и операции финального округления, позволяет строить устройства ЦОС с улучшенными характеристиками точности и быстродействия. Быстродействие устройств обратных преобразователей, совмещенных с операцией округления, в среднем на 10% выше чем у устройств выполненных традиционным способом, а выигрыш в точности вычислений при использовании режима вычислений в целых числах достигает 4-5 раз.

## ЛИТЕРАТУРА

- [1] He S., Torkelson M. A New Approach to Pipeline FFT Processor // Proceedings 10th International Parallel Processing Symposium (IPPS '96). 1996. P. 766 – 770.
- [2] Шпаковский Г.И. Параллельные микропроцессоры для цифровой обработки сигналов и медиа данных // Минск: БГУ, 2000. 196 с.
- [3] Wold E.H. and Despain A.M. Pipeline and parallel-pipeline FFT processors for VLSI implementation // IEEE Trans. Comput. May 1984. C-33(5). P. 414-426.
- [4] Тельпухов Д.В., Балака Е.С. Быстрое преобразование Фурье в системе остаточных классов // Сборник научных трудов «Параллельная компьютерная алгебра». Ставрополь: СГУ, Всероссийская научная конференция с элементами научной школы для молодежи. 2010. С. 135-139.
- [5] Амербаев В.М., Тельпухов Д.В., Балака Е.С., Константинов А.В. Методы построения прямых преобразователей модулярной логарифметики ориентированных на ЦОС // Проблемы разработки перспективных микро- и наноэлектронных систем – 2010. Сб. научных трудов / под общ. ред. А.Л. Стемпковского. М.: ИППМ РАН. 2010. С. 374-377.
- [6] Amos Omondi, Premkumar Benjamin Residue Number Systems: Theory and Impemetation // Imperial Colledge Press. 2007.