Модификация аппаратного генератора поворотных коэффициентов блока дискретного преобразования Фурье стандарта LTE.

А.А. Волохатый, магистр кафедры инфокоммуникационных систем ГУАП; avolohatiy@vu.spb.ru

Ф.К. Борисовский, ассистент кафедры безопасности информационных систем ГУАП; fb@vu.spb.ru.

Аннотация

Данная работа посвящена обзору современных подходов в аппаратном построении генератора поворотных коэффициентов блока ДПФ. Предлагается способ модификации генераторов с точки зрения затрачиваемых ресурсов: объемов памяти, логических блоков.

Введение

Технологии OFDM и SC-FDMA широко применяются в современной цифровой связи. Большинство стандартов связи, такие как LTE и 5G, основаны на использовании этих методов. Одним из ключевых элементов данных технологий является алгоритм дискретного преобразования Фурье (ДПФ).

Требования, выдвигаемые современными стандартами связи, не позволяют осуществить программную реализацию преобразования, что ставит необходимость аппаратной разработки блока ДПФ. Существуют различные способы построения подобных блоков, описанные в различных работах [1, 2]. Основные исследования направлены на улучшение организации общей структуры блока ДПФ: блоков «бабочек», методов хранения данных. Меньшее внимание уделяется созданию и хранению поворотных коэффициентов (ПК).

В данной работе приводится сравнение основных подходов к формированию ПК и предлагается модификация, которая позволяет сократить аппаратные затраты при построении схемы.

Основные подходы к формированию поворотных коэффициентов

Поворотный коэффициент [3] — один из основных элементов преобразования Фурье, который служит для объединения ДПФ меньшего размера.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}, \tag{1}$$

где W_N^{nk} — поворотный коэффициент, равный $e^{\frac{2\pi}{N} \cdot nk}$.

При аппаратной реализации блока преобразования Фурье ПК либо хранят в памяти, либо формируют при помощи ряда алгоритмов. Можно выделить три основных алгоритма:

1. coordinate rotation digital computer (CORDIC) [4],
2. рекурсивный косинусный генератор [5],
3. рекурсивный комплексный генератор [6].

Хранение в памяти

Хранение в памяти является наиболее простым способом получения ПК. Для использования данного подхода необходимо сформировать все требуемые коэффициенты и загрузить их память. Затем производить выгрузку из памяти очередного ПК при прохождении очередной стадии преобразования.

Стандарт LTE требует выполнять ДПФ размером от 12 до 2048. В таком случае аппаратный блок преобразования Фурье должен хранить 22 000 ПК. Данный набор содержит множество повторяющихся значений. Выделив среди них уникальные, можно сократить общий объем до 7 000. Каждый ПК занимает в памяти 2 ячейки (по одному на действительную и мнимую части), что увеличивает объем памяти вдвое. Например, для хранения всех ПК размером 18 бит потребуется $7000 \cdot 2 \cdot 18 = 252000$ бит или 30 кбайт.

CORDIC

CORDIC — алгоритм вычисления тригонометрических функций, описанный Д. Волдером в 1956 и 1959 годах. Данный алгоритм явля-

ется итеративным и позволяет вычислить поворот вектора с использованием только операций сложения и сдвига [4]. Вычисление ПК в точности является результатом операции поворота вектора на угол $\frac{2\pi}{N} \cdot nk$. Вычисление ПК с помощью алгоритма CORDIC осуществляется по формуле (2):

$$\begin{aligned} x_{i+1} &= (x_i - y_i \cdot 2^i \cdot d_i), \\ y_{i+1} &= (y_i + x_i \cdot 2^i \cdot d_i). \end{aligned} \tag{2}$$

Направление поворота задается параметром d_i, который рассчитывается по формуле (3):

$$d_i = \begin{cases} -1, \text{если } z_i < 0, \\ 1, \text{если } z_i \geq 0. \end{cases} \tag{3}$$

Параметр z_i является накопителем поворота и рассчитывается по формуле (4):

$$z_{i+1} = (z_i - d_i \cdot \arctan(2^{-i})). \tag{4}$$

Используя формулы (2) — (4), бит за битом рассчитывается результат поворота вектора ПК на заданный угол. Таким образом, число циклов, которое понадобится для расчета полного ПК, равно его битовому размеру.

Рекурсивный косинусный генератор

Рекурсивный косинусный генератор — метод формирования ПК, основанный на вычислении косинуса двойного угла [5]. Формирование каждого нового ПК осуществляется по формуле(5).

$$\begin{aligned} \sin(n \cdot \Delta) &= 2\cos(\Delta)\sin((n-1) \cdot \Delta) - \sin((n-2) \cdot \Delta) \\ \cos(n \cdot \Delta) &= 2\cos(\Delta)\cos((n-1) \cdot \Delta) - \cos((n-2) \cdot \Delta) \end{aligned} \tag{5}$$

Для аппаратной реализации косинусного генератора необходимо использовать 2 блока умножения и 2 блока суммирования. Так же необходимо хранить составляющие ПК с двух предыдущих итераций. Схема данного генератора приведена на рисунке 1(а).

Рекурсивный комплексный генератор

Рекурсивный комплексный генератор — метод формирования ПК, основанный на использовании комплексного умножителя. Каждый но-

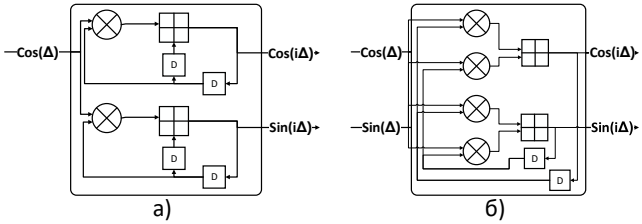

Рис. 1: а) Схема рекурсивного косинусного генератора. б) Схема рекурсивного комплексного генератора.

вый ПК формируется по формуле(6).

$$\begin{aligned} \sin n \cdot \Delta &= \cos \Delta \sin\left((n-1)\cdot\Delta\right) + \sin \Delta \cos\left((n-1)\cdot\Delta\right) \\ \cos n \cdot \Delta &= \cos \Delta \cos\left((n-1)\cdot\Delta\right) - \sin \Delta \sin\left((n-1)\cdot\Delta\right) \end{aligned} \tag{6}$$

Аппаратная реализация данного блока будет включать 4 блока умножения и 2 блока сложения. Схема данного генератора приведена на рисунке 1(б). Воспользовавшись результатами исследования [6], можно уменьшить размер комплексного умножителя до 3-х операций умножения и 5-и операций сложения.

Итоговое сравнение описанных выше подходов по параметрам затрачиваемых ресурсов и производительности при ширине компоненты 18 бит приведены в сводной таблице 1.

Таблица 1: Сравнение подходов к формированию ПК

Параметр	Память	CORDIC	Рекурсивный косинусный генератор	Рекурсивный комплексный генератор
Память	30 кбайт	18 кбайт	1кбайт	1 кбайт
Задержка	1 цикл	18 циклов	2 цикла	2 цикла
Ресурсы	1 сумматор	54 сумматора, 2 умножителя	2 сумматора, 2 умножителя	5 сумматоров, 3 умножителя

По данным сравнения, приведенным в таблице 1, можно сделать вывод, что наиболее эффективными способами получения ПК являются рекурсивные косинусный и комплексный генераторы. Данные генераторы обладают малой задержкой и требуют меньшее число ресурсов.

Корректирующие последовательности

Ошибки при генерации ПК

Как видно из формул(5),(6) и схем 1(а, б), для формирования каждого нового ПК требуется использовать коэффициенты, вычисленные на предыдущих стадиях. Такой способ формирования в условиях аппаратной реализации приведет к накоплению ошибок. На рисунке 2 приведена зависимость ошибки от номера сформированного коэффициента для комплексного генератора ПК шириной 18 бит. Как видно

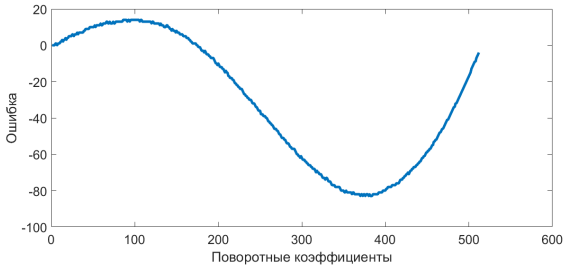

Рис. 2: Ошибка при формировании ПК. Размер преобразования 2048.

из рисунка 2, размер ошибки может достигать 8 бит. В преобразовании будут участвовать ПК размером 18 бит, из которых лишь 10 не будут содержать ошибок. Возникновение ошибочных бит негативно скажется на точности всего преобразования. Для устранения подобных погрешностей требуется использовать корректирующие последовательности (КП), которые, как правило, хранят в памяти.

Размер корректирующей последовательности

Ширина КП зависит от способа формирования поворотных коэффициентов. Для рекурсивного косинусного генератора требуется КП шириной 3 бита [5] на каждую из составляющих коэффициента. На рисунке 3(а) приведена схема рекурсивного косинусного генератора с использованием КП.

Ширина КП комплексного генератора составляет 2 бита на каждую из составляющих ПК. Сама КП при этом будет состоять из -1, 0, +1. На рисунке 3(б) показана схема рекурсивного комплексного генератора с использованием блока коррекции.

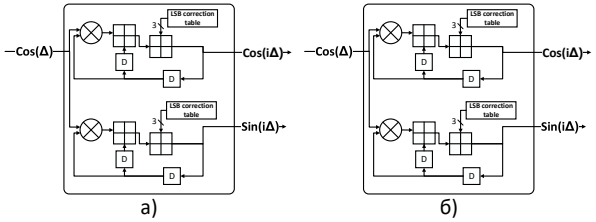

Рис. 3: а) Рекурсивный косинусный генератор с блоком коррекции. б) Рекурсивный комплексный генератор с блоком коррекции.

КП может быть сокращена путем исключения нулевой коррекции. В каждый генерируемый ПК будут вноситься изменения, даже если он вычислен без ошибки. ПК намеренно искажается, но при этом ширина КП сокращается до 1 бита (-1, +1). Размер памяти, хранящей КП, сокращается в 3 раза. Шумы, вносимые намеренными искажениями сигнала, влияют только на последний бит ПК, что снижает точность вычислений, но не оказывает сильного влияния на итоговый результат (разница достигает 6.5 дБ.).

Метод уменьшения ширины КП применим и к рекурсивному косинусному генератору. КП будет состоять из значений (-3, 3). В результате применения такой последовательности произойдет потеря точности на выходе ДПФ. Отношение сигнал/шум (SNR) снизится на 14 дБ.

Возможно улучшение косинусного генератора путем расширения разрядности генератора ПК на 1 бит. Это приведет к усложнению общей схемы преобразования: увеличению объема памяти, размера блоков умножения. В результате значение SNR будет таким же, как при использовании рекурсивного комплексного генератора исходной разрядности.

Результат моделирования различных генераторов с использованием различных модификаций приведены на рисунке 4.

Заключение

В данной работе было проведено сравнение основных подходов к построению генератора поворотных коэффициентов, среди которых были выделены наиболее оптимальные в отношении затрачиваемых ресурсов и времени задержки. Проведено моделирование данных генераторов и оценены размеры КП. Путем намеренного внесения искажений в поворотные коэффициенты удалось сократить КП рекурсивного комплекс-

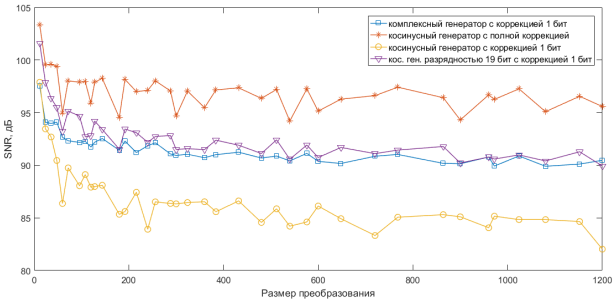

Рис. 4: Результаты моделирования.

ного и косинусного генераторов до 1 бита, сократив размер используемой памяти в 3 раза. Для использования рекурсивного косинусного генератора с шириной КП в 1 бит необходимо расширить разрядность ПК на 1 бит.

Литература

[1] A. Pattan and M. Latha. Fast Fourier Transform Architectures: A Survey and State of the Art // International Journal on Electronics & Communication Technology, vol.5, Oct. 2014, pp. 94-98.

[2] M. Jasmin. A Low Power Pipelined FFT/IFFT Processor for OFDM Applications // International Journal of Advanced Research in Electrical, vol.2, Oct. 2013, pp. 4712-4721.

[3] W. M. Gentleman and G.Sande. Fast Fourier transforms—for fun and profit // AFIPS Proceedings of the Fall Joint Computer Conference, №19, Nov. 1966, pp. 536-578.

[4] A. S. Padekar and S. S. Belsare. Design of a Cordic Based Radix-4 FFT Processor // International Journal of Computer Science and Information Technologies, vol. 5, Apr. 2014, pp. 5125-5128.

[5] J. Park. Design of a radix-8/4/2 FFT processor for OFDM systems // IEEE Trans. Consumer Electron, vol. 39, №1, Feb. 2009, pp. 117-124.

[6] V. Sarada and P. Vigneswaran. Low Power Complex Multiplier based FFT Processor // International Journal of Engineering and Technology(IJET), vol. 7, №4, Sept. 2015, pp. 1330-1335.