

ЛАБОРАТОРНАЯ РАБОТА №3

Выполнение быстрого преобразования Фурье дискретного сигнала на микроконтроллерах с ядром ARM с использованием библиотек CMSIS

Цель работы: Научиться применять элементы библиотеки CMSIS при разработке программ для микроконтроллеров STM32F4; выполнять быстрое преобразование Фурье с использованием библиотеки CMSIS.

Расчетная трудоемкость работы: 12 часов.

Предварительное задание: освежить знания о принципе преобразования Фурье, изучить сущность алгоритма БПФ, ознакомиться с библиотечными функциями CMSIS, предназначенными для выполнения БПФ (https://arm-software.github.io/CMSIS_5/DSP/html/group_ComplexFFT.html).

1. КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Алгоритм быстрого преобразования Фурье (БПФ)

Физический смысл преобразования Фурье можно описать следующим образом – функция, полученная в результате преобразования Фурье, показывает частотный спектр исходного сигнала. В технике знание спектра сигнала дает широкие возможности обработки и применения сигналов. Элементы преобразования Фурье используются также в алгоритмах сжатия аудио-, фото- и видеофайлов.

Так как цифровая техника оперирует дискретными сигналами, то на практике применяется дискретное преобразование Фурье (ДПФ), то есть преобразование сигнала, заданного в цифровом виде. Функция дискретного преобразования Фурье для некоторой функции x_n , заданной дискретно и имеющей k элементов, вычисляется по формуле:

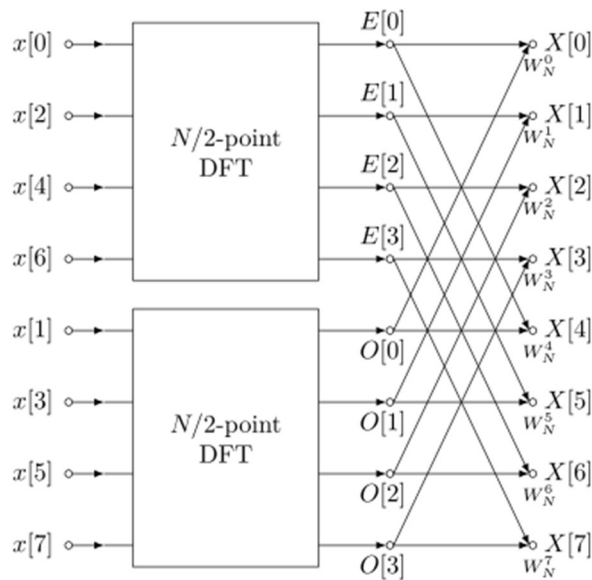
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} kn} = \sum_{n=0}^{N-1} x_n \left(\cos(2\pi kn/N) - i \cdot \sin(2\pi kn/N) \right),$$

где N – количество элементов в исходной последовательности, i – мнимая единица. Таким образом, чтобы получить один единственный элемент из выходной последовательности X_k , нужно вычислить N элементов и произвести их сложение. Аналогичные операции необходимо произвести для каждого элемента выходной последовательности.

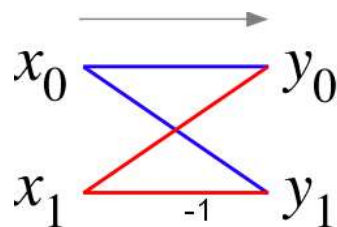
Видно, что алгоритм непосредственного дискретного преобразования Фурье обладает высокой вычислительной сложностью, то есть потребует много машинных ресурсов для выполнения преобразования. Чтобы упростить эту процедуру был изобретен ряд алгоритмов так называемого быстрого

преобразования Фурье (БПФ), позволяющих получить тот же результат более простым способом. Наибольшее распространение получил алгоритм Кули-Тьюки.

Разберем вкратце сущность алгоритма Кули-Тьюки. Пусть есть исходная функция x длиной N элементов. Допустим, мы хотим получить на выходе преобразование Фурье X длиной также N элементов. В прямом алгоритме это потребовало бы вычислить N^2 элементов и произвести их сложение. Однако алгоритм Кули-Тьюки предлагает сначала разбить исходную последовательность на две части длиной $N/2$, и для каждой части отдельно вычислить преобразование Фурье длиной $N/2$. В результате мы получаем две промежуточные последовательности E и O длиной $N/2$ каждая. Суммарно это потребует гораздо меньше операций.



Далее из этих двух промежуточных последовательностей по специальному алгоритму вычисляется результирующая последовательность X . Этот алгоритм использует свойство симметрии в уравнениях преобразования, и поэтому вычисляется по элементам последовательности перекрестно. Такое перекрестное вычисление называют «бабочкой» (radix):



Операция «бабочка» является элементарным шагом алгоритма БПФ. Алгоритм разбивает исходную последовательность на составляющие таким образом, чтобы получить минимальное количество вычислительных операций.

1.2 Быстрое преобразование Фурье с помощью библиотеки DSP CMSIS

В библиотеке CMSIS функции, предназначенные для выполнения преобразования Фурье, находятся в каталоге *CMSIS\DSP_Lib\Source\TransformFunctions*. Наиболее популярной является функция, расположенная в файле *arm_cfft_radix4_f32.c*. Функция выполняющая БПФ, называется *arm_cfft_radix4_f32*. Она реализует алгоритм преобразования путем разбивки входной функции на четыре части. Функция поддерживает обработку входных массивов, длина которых равна 2^n , т.е. 16, 64, 128 и т.д.

1.3 Пример БПФ в библиотеке CMSIS

Пример применения функции *arm_cfft_radix4_f32* приведен в файле, расположенном в директории *CMSIS\DSP_Lib\Examples\arm_fft_bin_example*. Файл *arm_fft_bin_example_f32.c* содержит код, в котором реализована обработка входного массива с помощью функции *arm_cfft_radix4_f32*. Файл *arm_fft_bin_data.c*, расположенный в этом же каталоге, содержит массив данных, который можно использовать в качестве тестовой последовательности. Массив представляет собой цифровой сигнал, в котором на синусоиду частотой 10 кГц наложен белый шум. Длина массива составляет 2048 элементов. Преобразование Фурье от такого сигнала покажет, что в спектре имеется один пик, соответствующий основной частоте, и некоторое спектральное распределение шума.

2. ПРОГРАММА РАБОТЫ

2.1 Изучите ассортимент функций цифрового преобразования сигналов, содержащиеся в каталоге *DSP_Lib\Source\TransformFunctions*.

2.2 Изучите встроенный пример применения БПФ, расположенный в каталоге *CMSIS\DSP_Lib\Examples\arm_fft_bin_example*.

2.3 Напишите программный код, в который будет интегрирована функция БПФ из примера. Продемонстрируйте работу примера, выведя на ЦАП сначала исходный сигнал, а затем его преобразованную функцию.

3. СОДЕРЖАНИЕ ОТЧЕТА

1. Название и программа работы.
2. Задание
3. Код программы с подробными комментариями

4. Выводы по работе

4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что представляет собой функция, являющаяся результатом преобразования Фурье от некоторой исходной функции?
2. Как работает алгоритм быстрого преобразования Фурье? Почему такое преобразование называется быстрым?
3. Как применять функции CMSIS для преобразования Фурье? Какие возможности это дает разработчику?