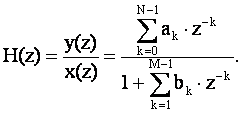
1.Программирование цифровых рекурсивных фильтров (ЦРФ). Распределение памяти данных. Циклическое программирование ЦРФ.

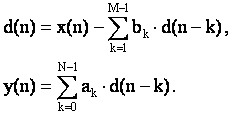
Прямая форма записи уравнения цифрового рекурсивного фильтра во временной области выглядит следующим образом:

 (2.2.5)

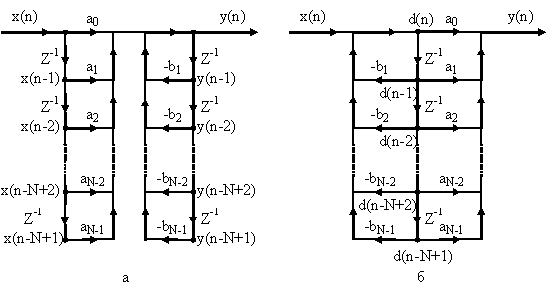
Передаточная функция цифрового рекурсивного фильтра (ЦРФ) или фильтра с бесконечной импульсной характеристикой (БИХ или IIR – Infinite Impulse Response) в Z-области записывается уравнением:

 (2.2.6)

Кроме прямой формы (2.2.5) имеется и другая форма (называемая канонической) уравнения ЦРФ с использованием промежуточных переменных d:

 (2.2.7)

Обе эти формы (2.2.5) и (2.2.7) для случая N = M могут быть представлены следующими схемами (рис. 2.2.5):

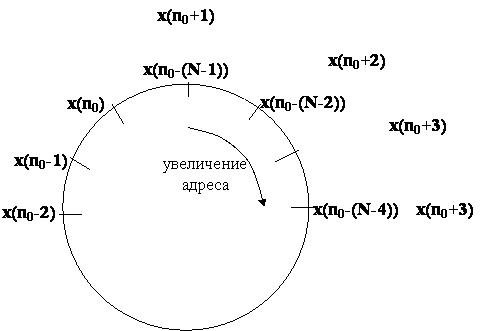


### Рисунок 2.2.5 – Граф-схемы цифровых рекурсивных фильтров

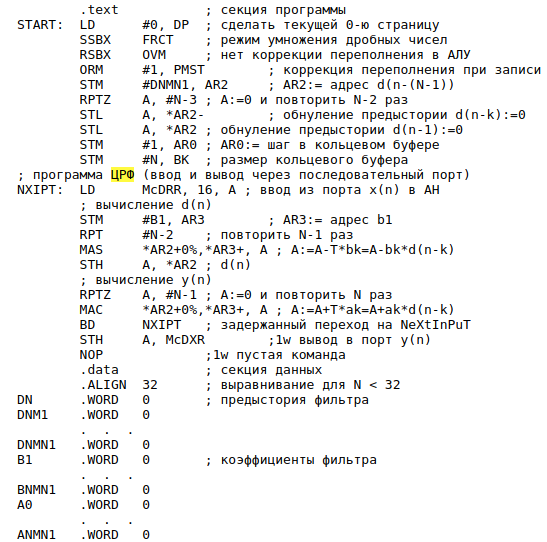
Прямая форма (рис. 2.2.5а) требует 2N+1 коэффициентов и 2N+1 рабочих ячеек для переменных X, Y (переменные x(n), y(n) могут храниться в одной ячейке памяти). При канонической реализации (рис. 2.2.5б) число переменных в два раза меньше (N+2).

Обычно при программировании ЦФ используется разбиение фильтра высокого порядка на ряд фильтров более низкого порядка (например 2-го порядка), поскольку при программировании фильтра высокого порядка трудно избежать переполнения для промежуточных результатов вычисления и большой разницы в величине коэффициентов.  
  
Расположение отсчетов предыстории в памяти данных аналогично их расположению у нерекурсивного фильтра: DN (мл.) -> DNM1-> DNM2 (ст.).

Наличие в процессоре циклической адресации позволяет написать программу без использования команды задержки DELAY за счет организации записи нового значения x(n) на место самого старого значения x(n-(N-1)). На рисунке 2.2.4 показано расположение данных в циклическом буфере. Внутреннее кольцо показывает расположение данных предыстории на момент времени n0, а внешнее, замещающие их данные на моменты времени n0+1, n0+2, n0+3 и т.д.



#### *Программа ЦРФ с использованием циклической адресации:*



2.Компрессия данных. Кодеки. Кодек G711.

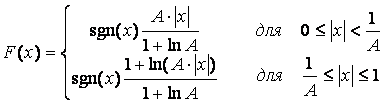
Под **кодеками** понимаются различные математические модели используемые для цифрового кодирования и компрессирования (сжатия) аудио информации. Многие из современных кодеков используют особенности восприятия человеческим мозгом неполной информации: алгоритмы голосового сжатия пользуются этими особенностями, вследствие чего не полностью услышанная информация полностью интерпретируется головным мозгом. Основным смыслом таких кодеков является сохранение баланса между эффективностью передачи данных и их качеством.

Изначально, термин кодек происходил от сочетания слов **КОД**ирование/**ДЕК**одирование, то есть устройств, которые преобразовывали аналог в цифровую форму. В современном мире телекоммуникаций, слово кодек скорее берет начало от сочетания **КО**мпрессия/**ДЕК**омпрессия.

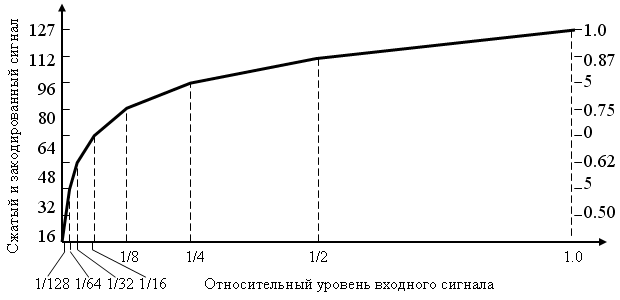
При компрессии (сжатии) данных по амплитуде (английский термин COMPANDING = COMpressing and exPANDING (сжатие и расширение)) используются два основных стандарта:  
m**-Law** – используется в США, Японии;  
A-Law – принят в Европе.   
Первый из них описывается выражением:



где F(x) – сжатое выходное значение, x – нормализованный выходной сигнал (диапазон от –1 до +1), μ=255 – параметр сжатия, sgn(x) – знак (±) x, а второй выражением:



где F(x) – сжатое выходное значение, x – нормализованный выходной сигнал (диапазон от –1 до +1), А=87,6 – параметр сжатия, sgn(x) – знак (±) x.  
В графическом виде процесс сжатия можно проиллюстрировать следующим образом:



Оба метода дают примерно одинаковую картинку и заметные отличия начинаются с уровня 1/32. При этом первый метод дает несколько завышенные значения относительно логарифмической шкалы что позволяет перекрыть несколько больший диапазон входного сигнала (до 13 бит после запятой), тогда как предел для второго метода – 12 бит.  
При кодировании формируется восьмибитное значение следующего формата Y = PSSSQQQQ, где P – бит знака (0 – плюс, 1 – минус), SSS – трехбитный номер сегмента, QQQQ – четырехбитный уровень квантования.  
В процессе передачи полученные значения кода могут инвертироваться целиком или четные/нечетные биты.

G.711

Кодек G.711 это самый базовый кодек ТфОП (PSTN). В рамках данного кодека используется импульсно-кодовая модуляция PCM. Всего в мире используется 2 метода компандирования (усиления сигнала) G.711: **µ – закон** в Северной Америке и **A – закон** в остальной части мира. Данный кодек передает 8 – битное слово 8 000 раз в секунду. Если умножить 8 на 8 000, то получим 64 000 бит – то есть 64 Кб/с, скорость потока, создаваемого G.711.

Многие люди скажут, что G.711 это кодек, в котором отсутствует компрессирование (сжатие), но это не совсем так: сам по себе процесс компандирования является одной из форм компрессирования. Все мировые кодеки «выросли» на базе G.711.

Важная особенность G.711 в том, что он минимально загружает процессор машины, на которой он запущен.

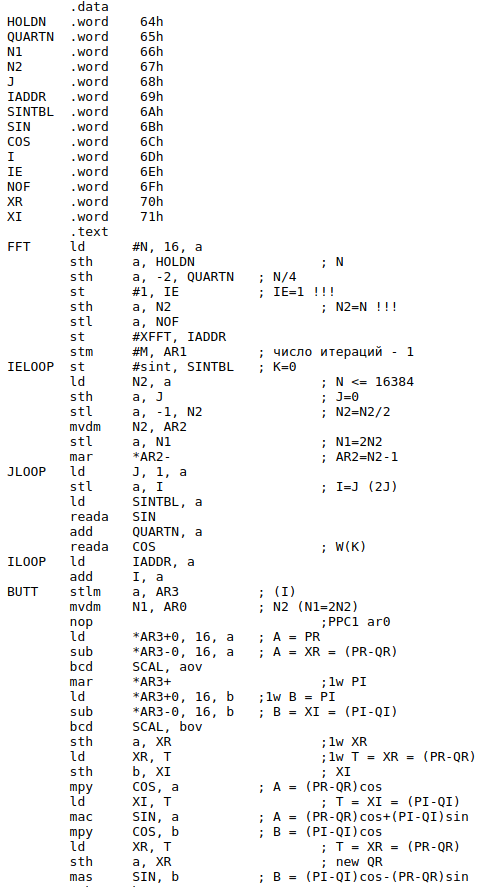
24. Программная реализация алгоритма БПФ с поблочно-плавающей запятой. Основные преимущества данного метода.

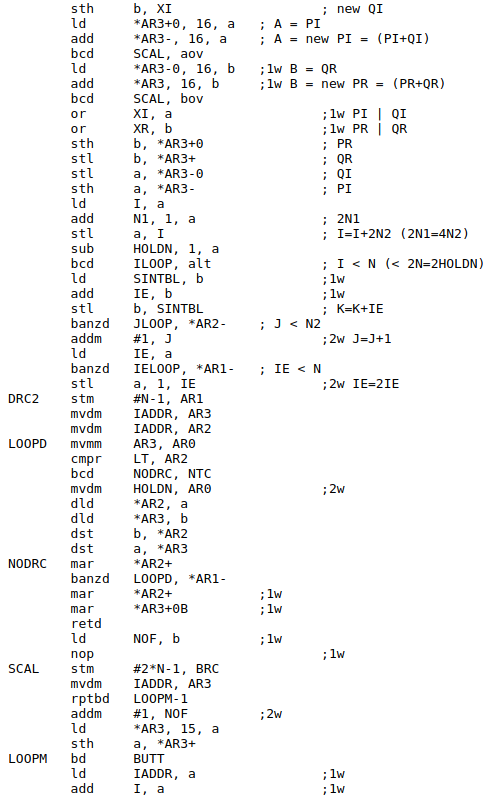
## Принцип поблочно-плавающей запятой

Одним из недостатков рассмотренных выше программ является то, что на каждой итерации выходные значения делятся пополам. В результате для широкополосных сигналов после БПФ получаются значения изображения Фурье с малым динамическим диапазоном. Увеличить динамический диапазон сигнала можно при использовании арифметики с плавающей запятой. Но реализация арифметики с плавающей запятой на процессорах с фиксированной запятой требует неоправданно высоких вычислительных затрат.  
Существует способ совместить достоинства фиксированной и плавающей запятой – поблочно-плавающая запятая. В этом случае значения чисел представляются дробной мантиссой и одним общим порядком. В случае возникновения переполнения все мантиссы делятся пополам, а общий порядок увеличивается на 1. Использование этого метода позволяет получить максимальный диапазон при минимальных затратах. В этом случае максимальная мантисса будет находиться в диапазоне от 0,5 до 1,0.  
Применить арифметику с поблочно-плавающей запятой можно к любой из рассмотренных ранее программ. Пример такой программы БПФ с прореживанием по частоте и прямым расположением входных данных приводится ниже. В этой программе после всех операций типа суммирование/вычитание производится проверка на переполнение и в случае переполнения все данные (уже обработанные и еще не обработанные) делятся пополам, и увеличивается счетчик переполнения, после чего повторяется вычисление «бабочки», на которой произошло переполнение. Для этого в программе выделены две ячейки промежуточных результатов (XR и XI), которые хранят новые значения до завершения вычисления «бабочки».

## Программная реализация алгоритма БПФ

Ниже приведен пример подпрограммы для реализации алгоритма БПФ с поблочно-плавающей запятой, прореживанием по частоте и прямым порядком входных данных.





55. Программирование цифровых нерекурсивных фильтров. Организация циклов.

Программирование цифровых нерекурсивных фильтров

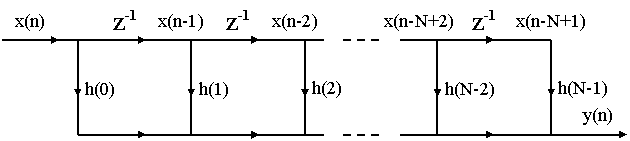
Как известно, уравнение цифрового фильтра (ЦФ) во временной области описывается выражением (прямая форма 1):

 (2.2.1)

Если все коэффициенты bk равны нулю, мы получаем уравнение цифрового нерекурсивного фильтра (ЦНФ), или фильтра с конечной импульсной характеристикой (КИХ или FIR – Finite Impulse Response):

 (2.2.2)

где коэффициенты ak из предыдущей формулы обозначены как h(k) (коэффициенты импульсной характеристики фильтра).  
Такой фильтр N-го порядка может быть представлен следующей схемой (рис. 2.2.1):



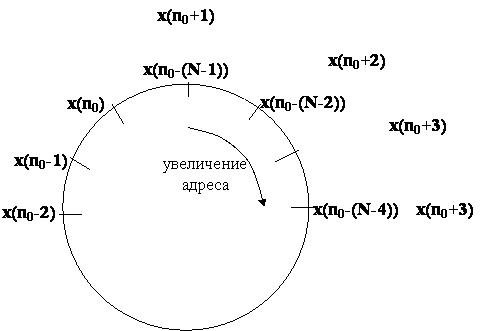
### Рисунок 2.1 – Граф-схема цифрового нерекурсивного фильтра

Передаточная характеристика этого фильтра описывается уравнением

 (2.2.3)

Организация циклов.

Наличие в процессоре циклической адресации позволяет написать программу без использования команды задержки DELAY за счет организации записи нового значения x(n) на место самого старого значения x(n-(N-1)). На рисунке 2.2.4 показано расположение данных в циклическом буфере. Внутреннее кольцо показывает расположение данных предыстории на момент времени n0, а внешнее, замещающие их данные на моменты времени n0+1, n0+2, n0+3 и т.д.



#### *Программа ЦРФ с использованием циклической адресации:*

