ВОЛОГДИН Э.И.

СТАТИСТИЧЕСКОЕ КОМПРЕССИРОВАНИЕ АУДИО СИГНАЛОВ

Конспект лекций

Санкт Петербург
2012 г

Содержание

1. Введение 3
2. Кодирование с адаптацией шага квантования 4
2.1. ИКМ с неравномерным шагом квантования 4
2.2. Транскодер ИКМ с переменным шагом квантования 7
2.3. ИКМ с плавающей запятой 8
2.3. Транскодер ИКМ с плавающей запятой 11
3. Кодирование с линейным предсказанием 12
3.1. Основы теории 12
3.2. Дифференциальная ИКМ и дельта модуляция 15
3.3. Адаптивная дифференциальная ИКМ (ADPCM) 18
4. Энтропийное кодирование 19
4.1. Информационная энтропия 19
4.2. Коды переменной длины 22
4.3. Код Хаффмана 23
4.4. Арифметическое кодирование 27
5. Компрессирование без потерь по стандарту SACD 30
5.1. Структура кодирования 30
5.2. Кодирование с предсказанием 31
5.3. Энтропийное кодирование 33
5.4. Мультиплексирование и демультиплексирование звуковых каналов 34
6. Компрессирование без потерь многоканального цифрового потока 35
6.1.Система компрессирования MLP 35
6.2. Кодирование в системе MLP 36
6.3. Матрицирование без потерь 37
6.4. Кодирование с предсказанием 38
6.5. Применение буферной памяти 39
6.6. Кодирование/декодирование субпотоков 40
Список литературы 42

1. Введение

В настоящее время в цифровой аудиотехнике для записи и передачи звуковых сигналов по каналам связи используются два вида аудио систем: без компрессирования и с компрессированием звуковых сигналов. Студийная запись производится всегда без компрессирования с использованием линейных методов аналого-цифрового преобразования на основе импульсно-кодовой или сигма-дельта модуляции.

Необходимость компрессирования возникает прежде всего при передачи звуковых сигналов по каналам связи и при необходимости увеличения длительности звукозаписи на дисковых оптических носителях и флеш-картах. В цифровых системах компрессирования основной задачей является сокращение скорости цифрового потока при записи или передачи данных. Системы с компрессированием по принципу работы и назначению разделяются на две группы

- психоакустическое (перцептуальное) кодирование,
- статистическое кодирование.

Психоакустическое кодирование всегда осуществляется с потерями, так как при кодировании сокращение скорости цифрового потока достигается благодаря устранению информационной избыточности звуковых сигналов. При этом убирается информация, которую слух человека не воспринимает. Избыточность эта огромна, поэтому возможно сокращение скорости цифрового потока до 20…40 раз без заметных на слух искажений.

Статистическое кодирование в принципе осуществляется без потерь за счет сокращения статистической избыточности звуковых сигналов и применения оптимальной с точки зрения теории информации упаковки цифровых данных. Избыточность эта не так велика, поэтому сокращение скорости цифрового потока, возможно не более чем 2…4 раза. Применительно к вычислительной технике разработано множество алгоритмов компрессирования без потерь, такие как PkZip, WinZip, WinRar . Они хорошо компрессируют текст, рисунки, но не могут компрессировать звуковые файлы. Так, например, текстовый файл объемом 4,56 Мбайт компрессируются до 696 кбайт, тогда как звуковой файл объемом 33,72 Мбайта этим же алгоритмом компрессируется только 31,57 Мбайта, всего на 6,5%. Применительно к звуку нужны другие идеи компрессирования

Основой статистического кодирования звуковых сигналов являются следующие методы сокращения скорости цифрового потока:

- квантование звуковых сигналов с неравномерным шагом,
- кодирование с предсказанием,
- энтропийное кодирование,
- адаптация шага квантования,
- адаптация предсказания.

Они могут быть использованы по отдельности и все сразу. Использование этих методов не сокращает информационной избыточности звуковых сигналов.

При цифровой звукозаписи скорость цифрового потока определяется емкостью носителя записи C_d и требующимся временем записи T_{rec}

$$V_d = \frac{C_d}{T_{rec}}.$$

За последние 20…30 лет в результате совершенствования технологии и принципов записи плотность ее на оптических и магнитных дисках повысилась во много раз и достигла 60 Мбит/мм^2. В результате емкость оптических дисков формата Blue-Ray составляет 50 Гбайт, близки к практической реализации голографическая и фотонная запись, которые обеспечивают емкость дисков прежних размеров до 600…800 Мбайт.

В результате при записи звуковых сигналов существенно упростились проблемы с возможностью повышения частоты дискретизации и числа разрядов ИКМ преобразователей, не стало жестких ограничений в отношении времени записи и числа записываемых звуковых каналов. Однако проблемы многоканальной записи пока остались. При емкости диска формата DVD-A 4,7 Гбайт аудиоданных максимальная скорость цифрового потока, при которой длительность записи один час, составляет около 10 Мбит/с . При этом без компрессирования 6-канальная запись возможна только в формате 24/48 кГц. С повышением частоты дискретизации до 192 кГц скорость цифрового потока увеличивается в 4 раза до 27,5 Мбит/с. Приходится либо уменьшать время записи в 4 раза или использовать сжатие цифрового потока без потерь с коэффициентом компрессии около 2,7.

В каналах передачи цифровой звуковой информации проблем значительно больше. Частотный диапазон кабельных и эфирных линий передачи разделяется на полосы, выделяемые пользователям, чем уже полоса, тем больше может одновременно передаваться сообщений по одному каналу. Число полос ограничено, поэтому каждый лишний бит, расширяющий полосу передачи, стоит очень дорого. Именно поэтому в этой области и ведутся интенсивные разработки по сокращению скорости цифровых потоков, часто в ущерб качеству. Скорости цифровых потоков здесь не сопоставимо меньшие, чем в аппаратуре звукозаписи.

Компрессирование в цифровых каналах используется в телефонных линиях связи, при передачи речевых и музыкальных сигналов в сети Интернет, в сотовых системах связи, при передачи вещательных сигналов по кабельным линиям и при многоканальной звукозаписи на оптических дисках . Речевой сигнал передается с использованием 8-разрядного кодирования с частотой дискретизации 8 кГц при скорости цифрового потока 64 или 32 Кбит/с. При передаче вещательных сигналов число разрядов увеличивается до 10…12, и и частота дискретизации повышается до 22,05 или 44,1 кГц. При этом скорость цифрового потока составляет 128 и 256 Кбит/с.

2. Кодирование с адаптацией шага квантования

2.1. ИКМ с неравномерным шагом квантования

Обычная линейная ИКМ не имеет механизма использования статистической избыточности ЗС. Однако, квантователь с постоянным шагом является оптимальным в смысле минимизации среднеквадратической ошибки квантования для сигналов с равномерной плотностью вероятностей распределения (PDF-Pulse Density Function)) мгновенных значений ЗС. У реальных музыкальных сигналов PDF чаще подчиняется закону Гаусса и

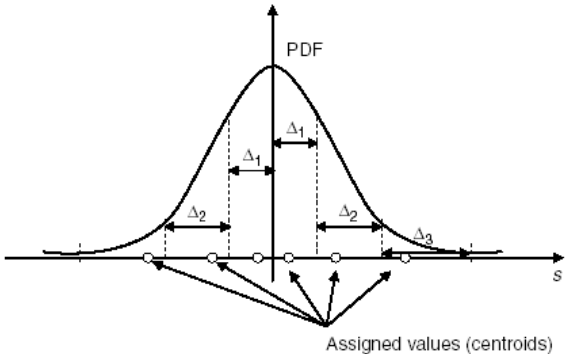

Рис.1. Статистически оптимизированное квантование

применение линейной ИКМ не эффективно.

В статистически оптимизированной ИКМ шаг квантования определяется PDF квантуемого сигнала. Часто встречающиеся амплитуды ЗС квантуются с меньшим шагом, а редко встречающие с –большим (рис.1). Например, при квантовании сигнала с Гауссо-

вой PDF среднеквадратическая ошибка квантования минимизируется, если рассчитываемый шаг квантования соответствует центроидам статистического распределения (рис.1.). Такое неравномерное квантование может использоваться в двух целях: уменьшение шума квантования при малых уровнях ЗС или только уменьшение скорости цифрового потока.

Другим путем *сокращения скорости цифрового потока* при аналого-цифровом преобразовании является использование *неравномерного квантования*, при котором шаг квантования является функцией входного сигнала конвертора. Для малых сигналов этот шаг делается малым, а для больших – более грубым (рис.2.). Компрессирование осуществляется по *логарифмическому закону*, поэтому шаг квантования изменяется также по этому за-

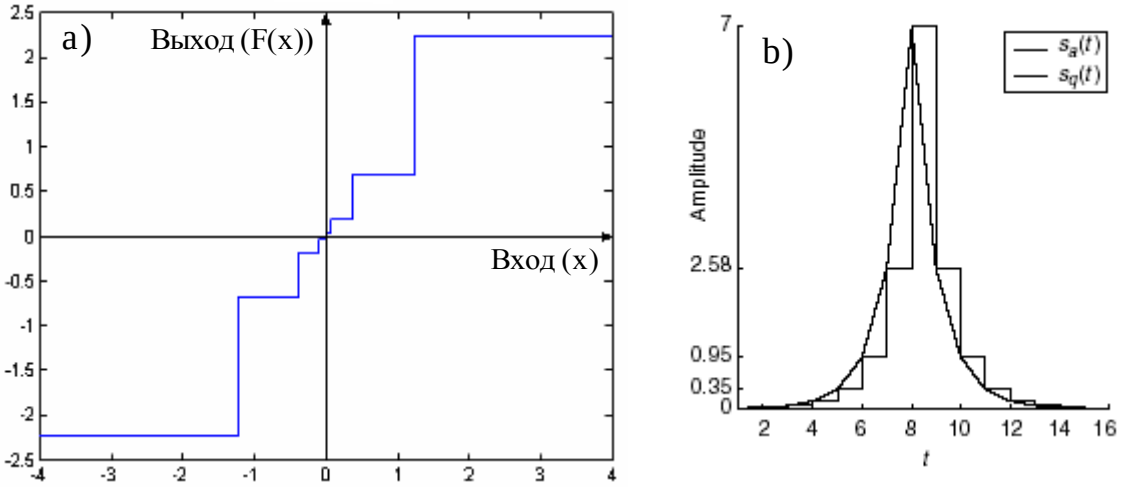

Рис.2. ИКМ с неравномерным шагом квантования. a) - передаточная функция, b) -квантованный сигнал

кону. Такое квантование часто называют *мгновенным компандированием*, так как конечный результат здесь такой же, как в аналоговых компандерных системах.

Упрощенной моделью тракта с АЦП и ЦАП с неравномерным шагом квантования может быть схема, приведенная на рис.3., в которой на входе тракта включен «аналоговый компрессор», на выходе – «аналоговый экспандер», тогда как аналого-цифровое и цифро-

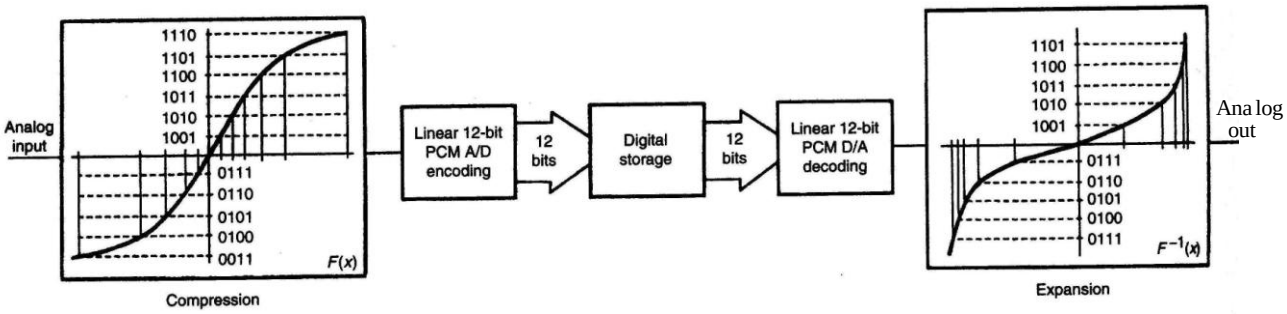

Рис.3. Компандерная цифровая линия передачи (мгновенное компандирование)

аналоговое преобразования осуществляются с помощью линейной ИКМ.

В линейных системах при больших уровнях ЗС значение SNR часто превышает необходимое значение, а при малых уровнях – SNR слишком мало. Применение неравномерного квантования уменьшает ошибки преобразования при малых уровнях ЗС, и увеличивает при больших уровнях. Поэтому при выборе степени компрессирования необходимо принимать компромиссные решения, при которых уменьшение числа разрядов квантователя не уменьшает динамический диапазон и среднее значение SNR.

Нелинейные законы изменения шага квантования могут быть представлены в виде графиков (рис.4.). По “μ – закону” производится компрессирование шага квантования в цифровых телефонных линиях Северной Америки и Японии. По “ A – закону” производится компрессирование шага квантования в цифровых телефонных линиях Европы. В

соответствии с рекомендациями CCITT используется 8-разрядное квантование с частотой дискретизации 8 кГц, при которых скорость цифрового потока равна 64 Кбит/с. Аналитически эти законы могут быть представлены в виде

μ – закон

$$F(x)=\frac{\operatorname{sgn}(x)}{\ln(1+\mu)}\ln\left[1+\mu|x|\right],\quad |x|\le 1 \quad . \qquad (1.)$$

Обратное преобразование

$$F^{-1}(y)=\operatorname{sgn}(y)\cdot\ 1/\mu\ \left[(1+\mu)^{|y|}-1\right],\quad -1\le y\le 1 \quad . (2)$$

A – закон

$$F(x)=\left\{\begin{array}{ll}\dfrac{A}{1+\ln A}\cdot|x|, & |x|\le 1/A\\ \dfrac{\operatorname{sgn}(x)}{1+\ln A}\cdot\left[1+\ln A\cdot|x|\right], & \dfrac{1}{A}\le|x|\le 1\end{array}\right\},\quad (3)$$

где $\operatorname{sgn}(x)=\begin{array}{ll}1, & if\ x>0\\ -1 & if\ x<0\end{array}$ - знаковая функция, x – нормализованный по отношению к $x_{\max}$ входной сигнал, $F(x)$ – выходной сигнал компрессора, $F^{-1}(y)$ – выходной сигнал экспандера, μ, A – степень сжатия. В приведенных формулах с равным успехом вместо натурального логарифма могут использоваться двоичные и десятичные логарифмы. Принципиального значения это не имеет.

Улучшение значения SNR при малых уровнях ЗС тем больше, чем круче подъем передаточной функции, но при этом ухудшается SNR на более высоких уровнях. В случае использовании рекомендованных значений коэффициентов сжатия $\mu=255$ и $A=87,56$

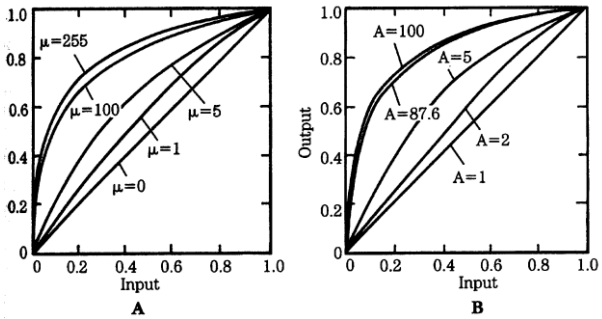

Рис.4. Передаточные характеристики компрессора A – "μ – закон", B – "A – закон"

при 8-разрядном квантовании достигаются такие же значения динамического диапазона и SNR, как при линейном 12-разрядном квантовании. Именно поэтому в модели на рис.4. предполагается, что аналого-цифровое и цифро-аналоговые преобразования осуществляются линейной 12-разрядной ИКМ.

В формулах (1-3) при нормализации уровня входного сигнала x уровень сигнала в квантах относится к максимально возможному уровню ЗС. Диапазон изменения уровней сигнала с динамическим диапазоном 96 дБ такой же, как у 16-разрядного ЗС от минус 32768 до 32767. Диапазон уровней 8 разрядного сигнала от минус 128 до 127. Если требует передать сигнал с динамическим диапазоном 96 дБ с применением нелинейного квантования с 8-разрядным компрессированием, тогда передаточная функция компрессора должна определяться равенством

$$F(x)=\operatorname{sgn}(x)\cdot 128\cdot\frac{\log_2(1+\mu\cdot\left|\dfrac{x}{32768}\right|)}{\log_2(1+\mu)}$$

На приемном конце после декомпрессирования осуществляется восстановление исходного ЗС с использованием инверсной функции (2).

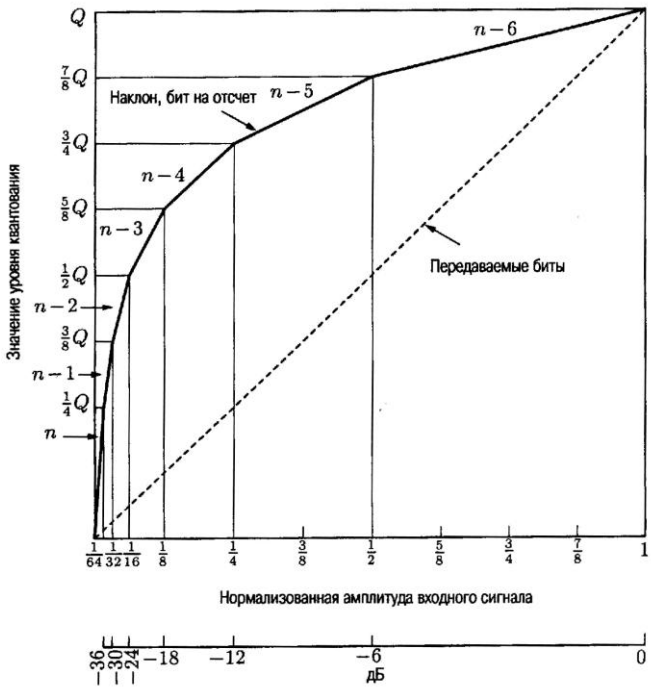

Рис.5. Линейно-сегментная аппроксимация

В практической реализации цифровых систем компрессирования часто передаточные функции задаются с использованием линейно-сегментной аппроксимации. В зависимости от числа используемых сегментов линейно-ломанную зависимость обозначают буквой и двумя цифрами. Например, запись А 87,6/13 означает, что используется аппроксимация по $A-$закону при $A = 87,6$ с 13 –ю аппроксимирующими отрезками (рис.5.). В пределах каждого сегмента шаг квантования постоянен, но при переходе в другой сегмент он изменяется в 2 раза. Число уровней квантования в пределах каждого сегмента постоянно.

Процедура кодирования каждого отсчета состоит в следующем. Сначала определяется полярность сигнала и в зависимости от нее формируется символ первого разряда (0 или 1). Затем производится двоичное кодирование номера сегмента, в пределах которого находится уровень входного сигнала. Для кодирования номера сегмента нужны трехразрядные кодовые комбинации. Далее кодируется уровень сигнала в пределах сегмента. Если число

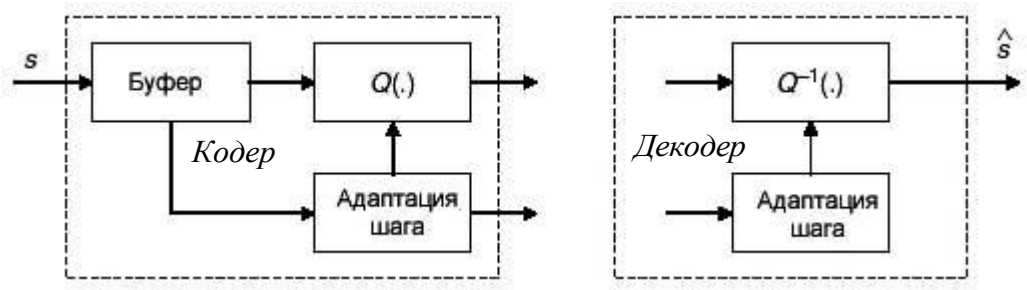

Рис.6. Схема адаптации шага квантования с предсказанием «вперед»

таких уровней в пределах сегмента равно 64, то для кодирования требуется 6-значная кодовая комбинация. Общее число разрядов в кодовом слове будет 10.

Статистические характеристики ЗС зависят от динамического диапазона ЗС, жанра музыкального произведения и многих других факторов, поэтому для повышения эффективности компрессирования применяют адаптацию шага квантования. На рис.6. приведен пример схемы адаптации по предыдущим выборкам (адаптация «вперед»). Информация о шаге квантования передается из кодера в декодер. Возможна также адаптация по последующим выборкам (адаптация «назад»). Такого рода ИКМ называют адаптивной и в международных документах обозначают как АРСМ.

2.2. Транскодер ИКМ с переменным шагом квантования

Сокращение скорости цифрового потока может производиться без использования неравномерного квантования с помощью транскодера, осуществляющего преобразование выборок линейной ИКМ. В технической литературе такую операцию часто называют *почти мгновенным компандированием*. В таком транскодере осуществляется линейная 14-разрядная ИКМ с частотой дискретизации 32 кГц. Цифровой поток данных разбивается на группы длительностью 1 мс, включающие в себя 32 выборки по 14 бит. При таком ма-

лом времени ЗС не может сильно измениться, поэтому все выборки группы могут иметь одинаковый шаг квантования. Почти мгновенное компандирование заключается в том, что 14-разрядные кодовые слова преобразуются транскодером в 10-разрядные по определенному алгоритму, при котором меняется шаг квантования.

Обычно при таком компрессировании используются 5 различных шкал с равномерным шагом квантования внутри каждой шкалы и разными шагами квантования в каждой из шкал (рис.7). Выбор шкалы квантования определяется уровнем входного сигнала за время около 1 мс. Минимальный шаг квантования имеет шкала 5 (диапазон 0), а максимальный – шкала 1 (диапазон 4). Число шагов квантования в каждой шкале равно 512, поэтому кодовые слова содержат по 10 бит.

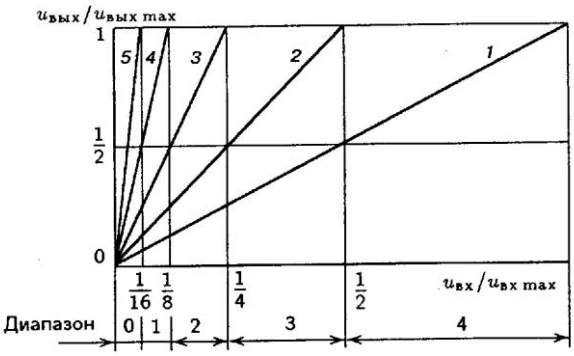

Рис.7. Шкалы квантования

Алгоритм преобразования 14⇒10 бит состоит в следующем. В каждой группе выборок выделяется выборка с максимальным кодовым значением, и во всех выборках группы отбрасываются 4 разряда. Какие именно отбрасываются разряды, зависит от кодовой комбинации в 4 старших разрядах выделенного кодового слова: 1111 - отбрасываются 4 старших разряда; 1110 - отбрасываются 3 старших разряда и 1 младший; 1100 - отбрасываются два старших разряда и 2 младших; 1000 - отбрасываются один старший разряд и 3 младших; 0000 - отбрасываются 4 младших разряда. Отбрасывание разрядов по такому алгоритму приводит к изменению шага квантования. При отбрасывании 1 младшего разряда шаг увеличивается в 2 раза, а при отбрасывании 4 младших разрядов - шаг увеличивается в 16 раз.

Таким образом, при рассматриваемом методе компрессирования шаг квантования зависит не от мгновенного значения выборки, а от ее максимального значения на интервале времени 1 мс. Для восстановления на приемной стороне абсолютного значения каждой выборки блоки по 32 выборки сопровождаются служебной информацией с кодовой комбинацией из 3 бит, определяющей по какой шкале производилось квантование.

2.3. ИКМ с плавающей запятой

ИКМ с плавающей запятой является нелинейной системой компрессирования, в которой осуществляется сокращение скорости цифрового потока путем использования неравномерного шага квантования. В этой системе весь диапазон изменения ЗС разбивается на поддиапазоны (шкалы) (рис.8). В каждом их этих поддиапазонов осуществляется аналого-цифровое преобразование с помощью линейной ИКМ со своим шагом квантования. Чем больше уровень ЗС, тем больше шаг квантования шкалы.

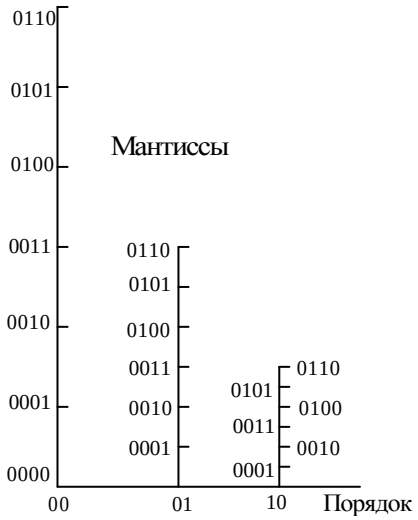

Рис.8. Шкалы квантования

Результирующее кодовое слово состоит из двух частей: мантисса (кодовое слово ИКМ) и порядок, кодовая комбинация которого определяет шаг квантования и шкалу. Может использоваться и знак порядка, указывающий направление сдвига запятой.

Осуществление импульсно-кодовой модуляции с плавающей запятой похоже на измерение напряжения с помощью многодиапазонного вольтметра с одной шкалой со 100 делениями. 50 делений этой шкалы могут соответствовать: 0.5; 5 и 50 В; 50; 5 и 0.5 мВ. Результат такого измерения складывается из отсчета шкалы и ее масштабного множителя. Очевидно, что

погрешность переключения шкал не может превышать половины деления шкалы.

В цифровой аудиотехнике ИКМ с плавающей запятой используется в двух случаях.

- Для расширения динамического диапазона ЗС при аналого-цифровом преобразовании без увеличения числа разрядов линейного квантователя.
- Для сокращения скорости цифрового потока без уменьшения динамического диапазона.

Первое применяется в студийной звукозаписи, второе – в технике связи при передаче данных.

В ИКМ с плавающей запятой (рис.9) входной сигнал как в обычной ИКМ проходит через антиэлайзинговый ФНЧ . Затем в блоке УВХ он дискретизируется, и преобразуется в непрерывную последовательность прямоугольных импульсов положительной и отрицательной полярности. После УВХ сигнал одновременно подается на множество усилите-

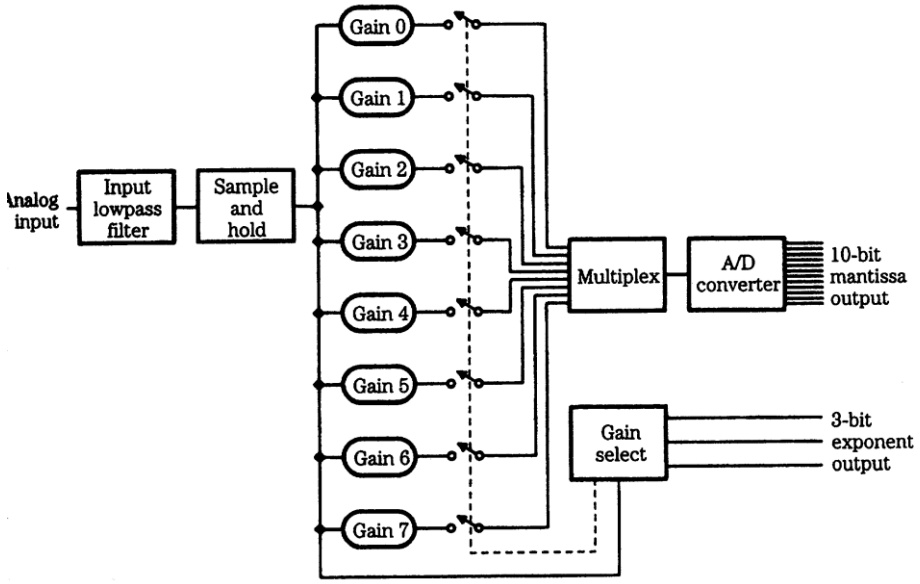

Рис.9. ИКМ с плавающей запятой

лей, число которых определяется динамическим диапазоном входного сигнала. Коэффициент усиления устанавливается с определенным шагом, который может быть 1, 3, 6, 9 или 12 дБ, возможна и любая другая дискретность изменения коэффициента усиления. Коммутатор Gain select управляется входным сигналом, и он переключает выходы усилителей в зависимости от уровня входного сигнала таким образом, чтобы выходной сигнал был максимален, но не было перегрузки. Через мультиплексор сигналы с усилителей подаются линейный квантователь, осуществляющий аналого-цифровое преобразование. Двоичное кодовое слово на выходе квантования (на рисунке оно 10-разрядное) называется мантиссой. Кодовая комбинация из 3 бит определяет положение переключателя (коэффициент усиления) и называется экспонентной.

В ИКМ с плавающей запятой общее число разрядов кодового слова q состоит из знакового разряда, m разрядов мантиссы и p разрядов порядка. В пределах каждой шкалы общее число уровней квантования постоянно и определяется только мантиссой

$$N_m = 2^m - 1. \qquad (4.)$$

Шаг квантования Q_j для каждой шкалы разный и определяется дискретностью d и порядком p

$$Q_j = Q_{\min} \cdot 2^{d \cdot j}, \qquad (5)$$

где $j = 0,1...(2^p - 1)$, $Q_{\min}$ – минимальный шаг квантования в децибелах.

В общем случае параметр дискретности шкалы d выбирается из ряда таких чисел: 1/4,

1/3, 1/2, 1, 2,… так, чтобы изменение шкалы происходило кратно 6 дБ (1,5; 2; 3; 6 или 12 дБ).

Динамический диапазон рассматриваемой системы определяется как отношение максимально допустимой амплитуды ЗС на входе $A_{\max}$ к минимальному шагу квантования $Q_{\min}$

$$A_{\max} = Q_{\min} \cdot 2^m \cdot 2^{d(2^p-1)},$$

поэтому он определяется равенством

$$DR = 6 \cdot m + 6d(2^p - 1). \qquad (6)$$

Если, например, в кодере используется 16-разрядная линейная ИКМ ($m = 16$), порядок компандрования задается значением $p = 2$, а дискретность изменения коэффициента усиления $d = 1$. Тогда коэффициент усиления может принимать только 4 значения : 0, 6, 12 и 18 дБ и, следовательно диапазон входных сигналов расширяется на $6 \cdot (2^p - 1)$ дБ с 96 для линейной ИКМ до 114 дБ для ИКМ с плавающей запятой. Если необходимо сократить скорость цифрового потока при передаче ЗС с динамическим диапазоном 96 дБ, то можно уменьшить разрядность ИКМ до 14 бит, тогда динамический диапазон выходного сигнала будет равен 102 дБ.

Известно множество вариантов реализации ИКМ с плавающей запятой. Они отличаются прежде всего способом регулирования коэффициента усиления (мгновенное, почти мгновенное, силлабическое и др.). В мгновенной системе коэффициент усиления выбирается для каждой выборки, поэтому он многократно скачкообразно за время периода кодируемого сигнала. Без заметных искажений это очень трудно реализовать, особенно на высоких частотах.

В силлабическом варианте реализации необходимое усиление устанавливается также за время выборки, но затем оно сохраняется неизменным в течение нескольких выборок, период удержания составляет 100…300 мс. Поскольку уровень усиления эффективно следует за огибающей входного сигнала, переключения усилителей происходят лишь несколько раз в секунду. Для сигналов очень низкой частоты силлабический алгоритм регулирования лучше мгновенного. Напротив, при мгновенном алгоритме регулирования шум квантования сопровождает почти не слышимый сигнал.

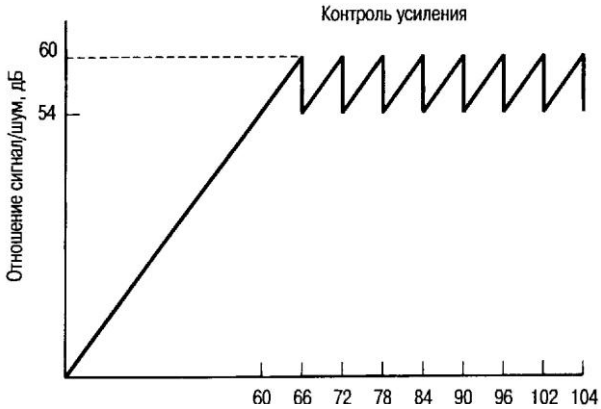

Рис.10. Зависимость SNR от уровня ЗС

Поскольку уровень сигнала определяет коэффициент усиления, неизбежно появление шумовой модуляции. Она может стать слышимой для низкочастотного сигнала высокого уровня, так как в этом случае она не маскируется ЗС. Из-за эффекта шумовой модуляции следует различать понятия динамический диапазон и отношение сигнал/шум. Динамический диапазон можно определить как отношение максимального среднеквадратического значения уровня сигнала к RMS – уровню шума квантования при отсутствии сигнала, тогда как SNR измеряется, когда сигнал присутствует.

График зависимости SNR от уровня сигнала для типичного преобразователя с плавающей запятой с 10-разрядной мантиссой, 3-разрядным порядком и шагом изменения коэффициента усиления равным 6 дБ, приведен на рис.10. Хотя теоретически эта система обеспечивает такой же динамический диапазон, как 17-разрядная линейная ИКМ (свыше 100 дБ) отношение сигнал/шум неприемлемо для высококачественных цифровых трактов. Как видно максимальное значение SNR достигает только 60 дБ, а при переключении шкал SNR уменьшается до 54 дБ.

2.3. Транскодер ИКМ с плавающей запятой

Транскодер предназначен для компрессирования цифрового потока путем преобразования кодовых слов линейной ИКМ с фиксированный запятой в кодовые комбинации с плавающей запятой . Структурная схема транскодера, осуществляющего преобразование 16-разрядной ИКМ в 10-разрядную ИКМ с плавающей запятой, приведена на рис.11. В этой схеме кодовые комбинации с выхода кодера ИКМ разделяются на группы по несколько слов, которые подаются на сдвиговый регистр памяти . В каждой группе выделяется слово с максимальным кодовым значением. Для него рассчитывается порядок (масштабный коэффициент шкалы) с тем, чтобы это слово оказалось в пределах шкалы.

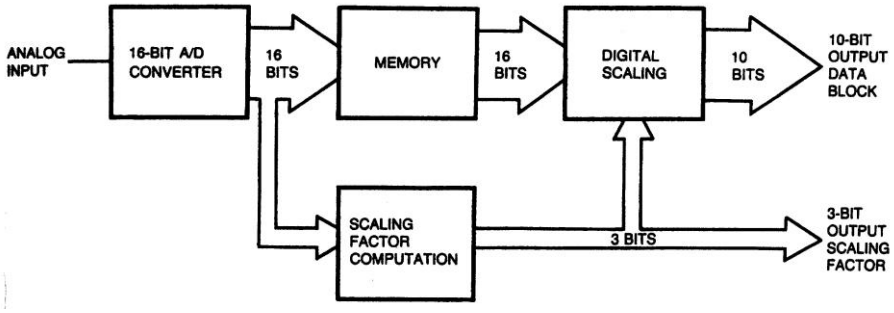

Рис.11.Схема преобразования 16-разрядной линейной ИКМ в 10-разрядную ИКМ с плавающей запятой

Это значение порядка является общим для всех слов группы. Для определения мантисс в осуществляется цифровое усиление, при котором все слова регистра делятся (умножаются) на выбранный порядок.

Например, ИКМ выборки с 16-разрядными разрядами, обеспечивающие динамический диапазон 96 дБ, могут быть заменены 10 разрядными кодовыми словами мантисс и 3-разрядным порядком с дискретностью изменения шкалы 6 дБ. При этом динамический диапазон по формуле (6) составляет 102 дБ, 60 дБ приходиться на мантиссы и 42 дБ – на порядок. Таким образом, при переходе к ИКМ с плавающей запятой динамический диапазон остался примерно тем же, но на каждой выборке экономится по 3 бита. Значение порядка относится к группе выборок, поэтому скорость кодовых слов масштабных коэффициентов также понижается. Этим достигается сокращение скорости цифрового потока примерно в 1,5 раза.

Транскодер такого типа используется для компрессирования звуковых сигналов в системе цифрового радиовещания Musicam-DAB. В этом транскодере дискретность шага шкалы $d = 1/3$ (2 дБ) и число шкал равно 64, которые перекрывают динамический диапазон 128 дБ. За нулевой уровень принято звуковое давление минус 96 дБ. При всех расчетах используются нормализованные числа с плавающей запятой. В этом транскодере число разрядов мантиссы может быть от 5 до 10, а порядок равен 6. Масштабный коэффициент рассчитывается один на группу длительностью 8 мс, состоящей из 12 символов. Иногда передается один масштабный коэффициент на 3 группы (36 символов). При этом достигается понижение скорости цифрового потока в 2 раза.

Аналогичный транскодер используется в системе цифрового радиовещания Nicam-3, разработанной корпорацией BBC, позволяющая передавать 6 звуковых каналов по одной стандартной телефонной линии со скоростью 2048 кбит/с . Субъективные тесты прослушивания, показали, что при сжатии 14-разрядного сигнала линейной ИКМ до 10 – разрядной ИКМ с плавающей запятой, звучание примерно такое же, как у 13-разрядной линейной ИКМ .

3. Кодирование с линейным предсказанием

3.1. Основы теории

В настоящее время при аналого-цифровом преобразовании методы двоичного кодирования звуковых сигналов с линейным предсказанием широко используются в сотовых линиях связи, в сети Интернет (IP- телефония), при записи интерактивных оптических дисков по стандарту CDROM, в системе высококачественной звукозаписи по стандартам DVD-Audio и Super Audio CD.

Из классической теории сигналов известно, что значения выборок, взятых через интервал Котельникова, *взаимно некоррелированы*, если спектр сигнала в занимаемой им полосе частот равномерен (белый шум). Однако на практике в основном используются сигналы, спектр которых неравномерен, поэтому корреляция между выборками не равна нулю. При этом *степень корреляции возрастает с увеличением частоты дискретизации*. Типичным примером таких сигналов является речь, где корреляция между соседними выборками достаточно велика при соблюдении требований теоремы Котельникова в процессе дискретизации.

На этом факте основан *принцип сокращения скорости цифрового потока* при записи звуковых сигналов или передачи по линиям связи цифровых данных, названый *«кодирование с предсказанием»*. Последовательность коррелированных выборок исходного сигнала подается на один из входов вычитающего устройства, а на другой его вход, поступает сигнал предсказания, аппроксимированный из предыдущих или последующих выборок. Полученный сигнал ошибки предсказания преобразуется в двоичный код с помощью мало разрядного квантователя, который и передается по линии связи.

На приемном конце линии передачи имеется такой же, как в передатчике предсказатель. Так как он оперирует с теми же значениями предыдущих выборок, предсказанное значение нового отсчета будет таким же, как в передатчике. Добавив к нему принятое значение ошибки, можно восстановить исходное значение выборки. При этом, чем сильнее корреляционные связи между выборками сигнала, тем точнее можно сформировать сигнал предсказания.

Передача разностного сигнала в большинстве случаев требует меньшего числа разрядов. Поскольку информация в канале достаточно предсказуема — кодирование с предсказанием снижает объем передаваемой по каналу информации.

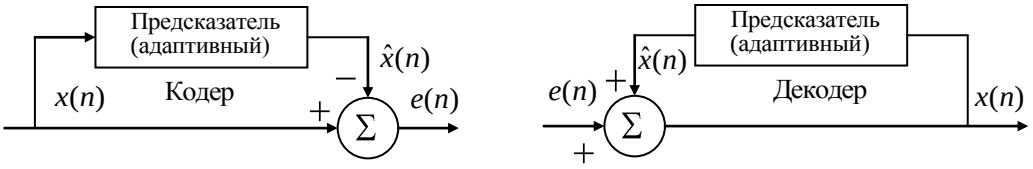

Рис.12. Структура кодека с (адаптивным) предсказанием «вперед»

Известны три структурных схемы построения кодера, в которых осуществляется предсказание «вперед», предсказание «назад» и оба предсказания одновременно. Упрощенная схема кодека с предсказанием «вперед» приведена на рис.12. В этой схеме входной сигнал $x(n)$ может быть в виде выборок (дискретизированные отсчеты), и тогда на выходе кодера включается мало разрядный *квантователь* Q (на схеме квантователи не показаны), преобразующий разностный сигнал в двоичный код. При этом на входе декодера должен быть *деквантователь* (Q^{-1}), осуществляющий двоичное декодирование. Входной сигнал может быть в виде квантованных выборок с большим числом разрядов, тогда на выходе кодера включается *реквантователь*, понижающий число разрядов в выборках.

При включении линейного предсказателя в прямую цепь кодера оцениваемое значение $x(n)$ выборки $-\hat{x}(n)$ рассчитывается как линейная комбинация *предыдущих выборок*

$$\hat{x}(n) = -\sum_{i=1}^{L} a_i x(n-i)\,, \qquad (7)$$

где a_i - это коэффициенты предсказания. Текущая ошибка предсказания определяется равенством

$$e(n) = x(n) + \sum_{i=1}^{L} a_i x(n-i)\,. \qquad (8)$$

Коэффициенты линейного предсказателя a_i определяются из условия минимума общей ошибки предсказания «вперед»

$$\varepsilon_f = \sum_n e^2(n) = \sum_n \left[\left(x(n) + \sum_{i=1}^{L} a_i x(n-i) \right)^2 \right]. \qquad (9)$$

Минимизация ε по отношению к a_i те. $d\varepsilon / da_i$ для $i = 1, 2, \ldots L$ определяется следующим равенством

$$r_{ss}(m) - \sum_{i=1}^{p} a_i r_{ss}(m-1) = 0\,, \qquad (10)$$

для $m = 1, 2, \ldots L$, где $r_{ss}(m)$ автокорреляционная последовательность сигнала $e(n)$. Через автокорреляционную функцию с помощью матричного преобразования могут быть рассчитаны все коэффициенты предсказателя a_i.

Предсказание «вперед» реализуется с помощью рекурсивного FIR-фильтра с бесконечным импульсным откликом (рис.13). Этот фильтр осуществляет *линейную аппроксимацию* огибающей передаваемого сигнала по L выборками, коэффициенты предсказания a_i одновременно являются аппроксимирующими коэффициентами. Сигнал предсказания с фильтра в форме Z- преобразования имеет вид

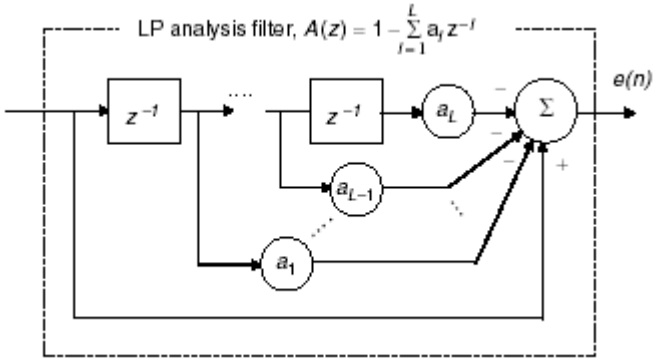

Рис.13. Кодер с предсказанием «вперед» с FIR-фильтром

$$A(z) = 1 - \sum_{i=1}^{L} a_i z^{-1}\,. \qquad (11)$$

Порядок этого фильтра определяет порядок предсказателя и порядок аппроксимации, чем он выше, тем меньше ошибка предсказания. В качестве предсказателей используются интерполяционные FIR и IIR-фильтры. Чаще используются FIR фильтры, которые значительно более быстродействующие, но чем выше их порядок, тем больше искажений он вносит. Обычно порядок таких фильтров не выше четвертого, однако, иногда этот порядок повышается до 128. Порядок IIR-фильтров всегда выше, обычно он выбирается в пределах 8…10.

В обычном предсказателе коэффициенты a_i рассчитываются только при настройке кодера и декодера. *В адаптивном предсказателе* эти коэффициенты являются функцией

входного сигнала, что обеспечивает большую точность предсказания и, следовательно, меньшую скорость цифрового потока в линии. Для их расчета входной цифровой поток разбивается на блоки, в рамках которых и рассчитываются эти коэффициенты. Размер блока выбирается исходя из условия коррелированности последовательности выборок. Обычно их длительность выбирается в пределах 3… 20 мс. В схеме кодера с адаптивным предсказателем «вперед» рассчитываемые коэффициенты предсказателя передаются отдельно, чтобы можно было реконструировать сигнал в приемнике.

Рис.14. Схема кодера с предсказанием «назад»

В схеме кодирования с предсказанием «назад» (рис.14), линейный предсказатель включается в цепь обратной связи. На вход такого кодера обычно подается квантованный сигнал и на его выходе включается реквантователь (на схеме не показан). Кодер имеет примерно те же характеристики, что и и схема с предсказанием «вперед» Однако, оцениваемое значение $x(n-L)$ выборки $-\hat{x}(n-L)$ рассчитывается как линейная комбинация *последующих выборок* L выборок

$$\hat{x}(n-L) = -\sum_{i=1}^{L} a_i x(n-L+i), \qquad (12)$$

где a_i - это коэффициенты предсказания. Текущая ошибка предсказания определяется равенством

$$e(n) = x(n-L) + \sum_{i=1}^{L} a_i x(n-L+i). \qquad (13)$$

Коэффициенты линейного предсказателя a_i определяются из условия минимума общей ошибки предсказания «назад»

$$\varepsilon_b = \sum_n e^2(n) = \sum_n x(n-L) + \sum_n \left[\left(\sum_{i=1}^{L} a_i x(n-i) \right)^2 \right]. \quad (14)$$

Реализуется такой предсказатель на основе нерекурсивного IIR-фильтра с конечным импульсным откликом, обычно адаптивного .

В схеме кодера с адаптивным предсказателем «вперед» и «назад» в прямой цепи используется адаптивный FIR- фильтр, а в цепи обратной связи – IIR- фильтр (рис.15). Такие же фильтры используются и в предсказателе декодера. Ошибка предсказания определяется равенством

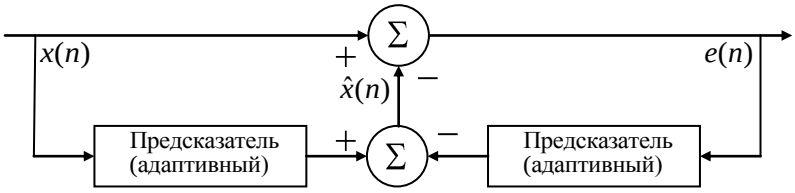

Рис.15. Схема кодера с предсказанием «вперед» и «назад»

$$e(n) = x(n) - \sum_{i=1}^{L} a_i x(n-i) + \sum_{i=1}^{N} b_i x(n-i), \qquad (15)$$

где a_i и b_i – коэффициенты аппроксимации одного и другого фильтра, L и M – число аппроксимируемых выборок (порядок фильтров). Всегда порядок IIR- фильтра выше порядка FIR-фильтра.

Очень важно, что в результате кодирования с предсказанием *происходит декорреляция* выборок кодируемого выходного сигнала. Поэтому *спектр ошибок предсказания*

приближается к равномерному, а ошибка предсказания $e(n)$ становится случайной величиной. Распределение вероятности значений этой ошибки подчиняется закону Гаусса (рис.16)

$$P_G(x) = \frac{1}{\sqrt{2\pi\sigma_x}} \cdot e^{-\frac{x^2}{2\sigma_x^2}} \quad (16)$$

или Лапласа

$$P_L(x) = \frac{1}{\sqrt{2\sigma_x}} \cdot e^{-\frac{|x|\cdot\sqrt{2}}{\sigma_x}}, \quad (17)$$

где σ_x – дисперсия отклонений.

Это позволяет использовать эффективное *энтропийное кодирование*, которое еще больше понижает скорость цифрового потока. При декодировании с предсказанием корреляция выборок восстанавливается.

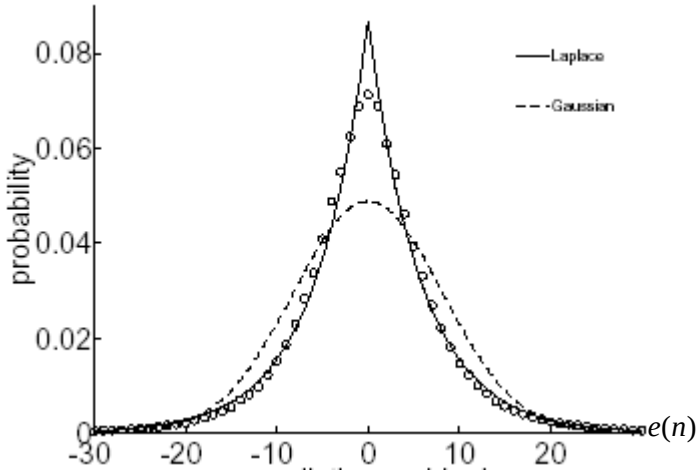

Рис.16. Распределение вероятностей ошибок предсказания

При кодировании с предсказанием для понижения скорости цифрового потока часто используется адаптация шага квантования, при которой шаг квантования ошибки предсказания $e(n)$ является функцией ее величины. Поэтому когда идет речь об адаптивной системе кодирования следует уточнять, что именно адаптируется. В современных системах кодирования часто адаптируются как шаг квантования, так и предсказатель. Современные кодеры с предсказанием реализуется только в программном виде на основе звуковых процессоров, поэтому вопросы адаптации решаются достаточно просто по заданному алгоритму.

3.2. Дифференциальная ИКМ и дельта модуляция

Дальнейшим развитием ИКМ является дифференциальная ИКМ (DPCM). Она более эффективна, так как сокращение скорости цифрового потока в этом случае дополнительно достигается за счет сокращения статистической избыточности, обусловленной корреляцией между соседними выборками. В простейшем случае, вместо того чтобы передавать точные двоичные значения каждой выборки передается только разница между текущим и предыдущим отсчетами. Поскольку разница эта обычно мала, можно использовать меньшее число бит без какого-либо ухудшения характеристик. Операция довольно проста: один отсчет хранится в течение полного периода выборки, и затем добавляется к полученному разностному сигналу для получения следующего отсчета. Этот отсчет затем хранится до тех пор, пока не будет получен разностный сигнал, и т. д.

Дифференциальная ИКМ представляет собой специальный тип кодирования с предсказанием «вперед». В таких схемах кодирования генерируется предсказание для текущего отсчета на основании предыдущих данных. Корректирующий сигнал – это просто разность между этим предсказанием и фактическим сигналом. Самый простой принцип предсказания — это сохранения предыдущего значения.

Действительно, эксперименты показывают, что разница между предыдущим и после-

дующим значением меньше, чем само значение отсчета. Поэтому для большей части огибающей сигнала ошибки может быть не большой. Обычно предсказатель представляет собой простой регистр, который накапливает значение предыдущего отсчета. В линию передается разностный сигнал. Для определения следующего разностного отсчета сигнал восстанавливается в сумматоре сложением этого отсчета с предыдущим значением сигнала .

Возможно, разностный сигнал вычислять с использованием аппроксимации нескольких отсчетов. Такой способ позволяет увеличить точность преобразования, но требует накопления отсчетов. При этом аппроксимированное значение сигнала $\hat{x}(n)$ определяется по приведенной выше формуле (7). При предсказании, основанном на предположении "сохранение предыдущего значения", коэффициенты аппроксимации не меняются, поэтому на приемном конце они те же самые — это позволяет не передавать их по линии, а пе-

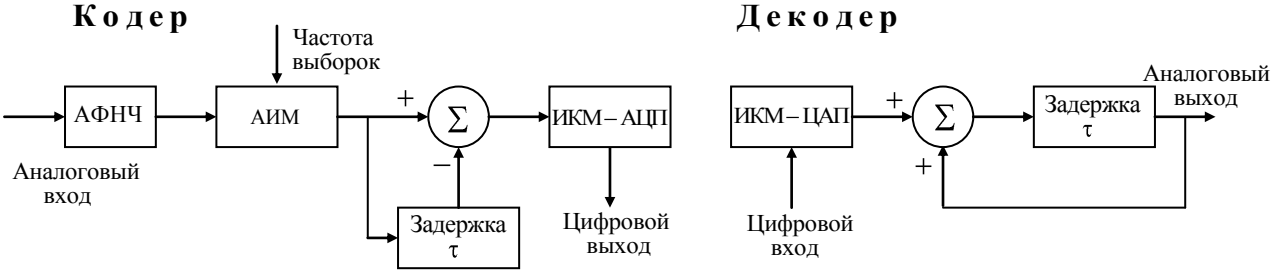

Рис.17. Дифференциальная ИКМ (кодер и декодер)

редать только разность аппроксимированного сигнала.

Приведенная выше формула для DPCM (7) приводится к виду

$$\hat{x}(n) = a \cdot x(n-1),$$

Если $a = 1$, тогда в линию передается разность между текущим и предыдущим значениями:

$$e(n) = x(n) - x(n-1).$$

Восстановление сигнала заключается в том, что к предыдущему значению сигнала прибавляется разность

$$x(n) = x(n-1) + e(n).$$

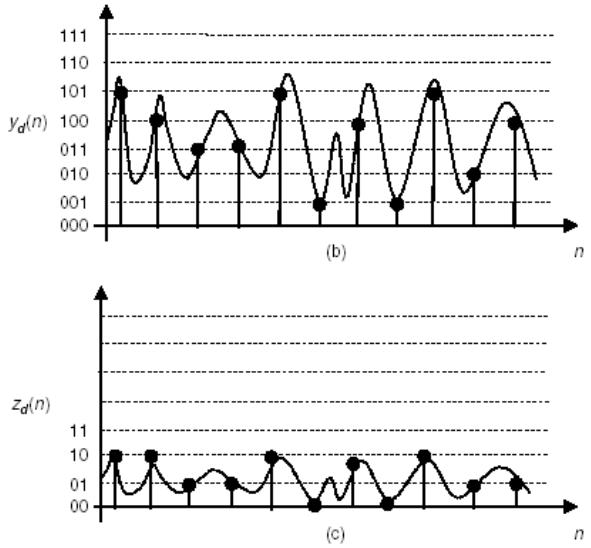

Рис.18. Сигналы по тракту DPCM

На рис.17. приведена типовая схема DPCM (кодер и декодер). В этой схеме как в обычной ИКМ, входной сигнал ограничивается по частоте антиэлайзинговым фильтром АФНЧ и с помощью амплитудно-импульсной модуляции в блоке АИМ производится дискретизация . В сумматоре производится вычисление разности между прямым и задержанным сигналами, величина задержки кратна периоду повторения выборок. В блоке ИКМ-АЦП производится квантование и двоичное кодирование разностного сигнала.

Динамический диапазон приращений значительно меньше динамического диапазона ЗС, поэтому число разрядов квантователя всегда меньше, чем в линейной ИКМ. В блоке ИКМ-ЦАП декодера производится двоичное декодирование с формированием импульсной последовательности выборок, несущих информацию о приращении ЗС. В сумматоре декодера вычисляется сумма прямого разностного сигнала и задержанного. В результате на выходе восстанавливается исходный ЗС. Принцип работы DPCM поясня-

ется рис.18..

При DPCM величина SNR определяется известным равенством

$$SNR = 6,02 \cdot q + 1,76 + 10 \cdot \lg\left(\frac{f_s \cdot K_{os}}{2F_{\max}}\right),$$

из которого следует, что при уменьшении числа разрядов квантователя с каждым разрядом SNR ухудшается на 6 дБ, а для того чтобы восстановить его значение частоту дискретизации надо увеличивать в 4 раза. Это значит, что если при частоте дискретизации 44,1 кГц 16 разрядная ИКМ обеспечивает отношение сигнал/шум 96 дБ в диапазоне до 20 кГц, то при 12-разрядной ИКМ такое же значение SNR будет при частоте дискретизации около 10 МГц.

При увеличении частоты дискретизации разности между предыдущим и текущим значениями становятся все меньше, поэтому в предельном случае очень высоких частот дискретизации для сигнала ошибки нужен лишь один бит, чтобы указать знак ошибки. Это и есть дельта модуляция (DM).

На рис.19 приведена упрощенная схема дельта модуляции (кодер и декодер). В этой схеме производится одноразрядное кодирование, поэтому функцию дискретизатора, квантователя и двоичного кодера выполняет двухуровневый компаратор. В сумматоре формируется разностный аналоговый сигнал. Выходной сигнал кодера представляет собой двухуровневую импульсную последовательность, тактируемую частотой дискретизации.

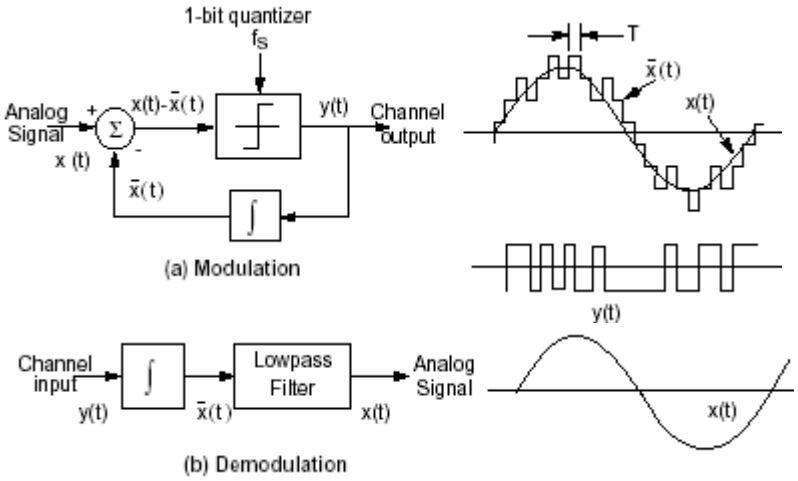

Рис.19. Дельта модуляция

Демодуляция такой кодовой последовательности осуществляется с помощью интегратора и простейшего ФНЧ, подавляющего частоту дискретизации.

Фактически при использовании DPCM и DM кодируется производная ЗС, максимальное значение которой линейно связано с шагом квантования Q и частотой дискретизации f_s

$$\frac{dV}{dt} = Q \cdot f_s .$$

Это значит, что необходимая точность преобразования в равной мере достигается уменьшением шага квантования, или увеличением частоты дискретизации. При использовании DPCM и DM частота дискретизации всегда выбирается значительно выше значения, определяемого теоремой В. П. Котельникова, что существенно упрощает проблемы антиэлайзинговой фильтрации и реконструкции ЗС. При малом числе разрядов с повышением крутизны нарастания ЗС возникает перегрузка, подобно тому, как происходит клипирование в линейной ИКМ, что сопровождается очень сильными искажениями.

При использовании DM динамический диапазон определяется равенством

$$DR = 20\log\left(\frac{0,2\cdot f_s^{1,5}}{F\cdot W^{0,5}}\right), \qquad (18)$$

где W – полоса частот. Для обеспечения SNR = 96 дБ в диапазоне до 20 кГц при использовании DM требуется частота дискретизации 200 МГц. В телефонном канале в диапазоне до 4 кГц при DM обеспечивается SNR = 66 дБ. Совершенствование дельта модуляции привело к созданию очень эффективной одноразрядной сигма-дельта модуляции.

3.3. Адаптивная дифференциальная ИКМ (ADPCM)

Применение DPCM позволяет значительно снизить скорость потока, однако ошибка при восстановлении сигнала зависит от величины шага квантования передаваемого кода. Уменьшение шага квантования (увеличение разрядности кодового слова) уменьшает ошибку и соответственно качество восстановленного сигнала, но увеличивает скорость потока. Увеличение шага квантования уменьшает скорость потока, но увеличивает ошибку восстановления. В рамках классической DPCM с фиксированным шагом квантования это противоречие неразрешимо.

Одно из решений состоит во введении адаптации шага квантования в процесс кодирования-декодирования. Такой алгоритм назван ADPCM (адаптивная дифференциальная импульсно-кодовая модуляция). Стандарт на этот метод кодирования был принят в 1984 году под названием G.726. Данный алгоритм позволяет воспроизводить речь почти с такой же субъективной оценкой качества, как и при использовании ИКМ, при скорости цифрового потока всего только 32 Кбит/с, что вдвое меньше скорости, используемой при ИКМ. Алгоритм ADPCM гарантирует оценки качества звучания по методике MOS на уровне 4.3 , что часто принимается за эталон качества телефонной связи. Этот алгоритм до сих пор используется при передаче речи по спутниковым и другим каналам связи, несмотря на появление кодеков с гораздо более низкой скоростью потока и достаточным качеством восстановления.

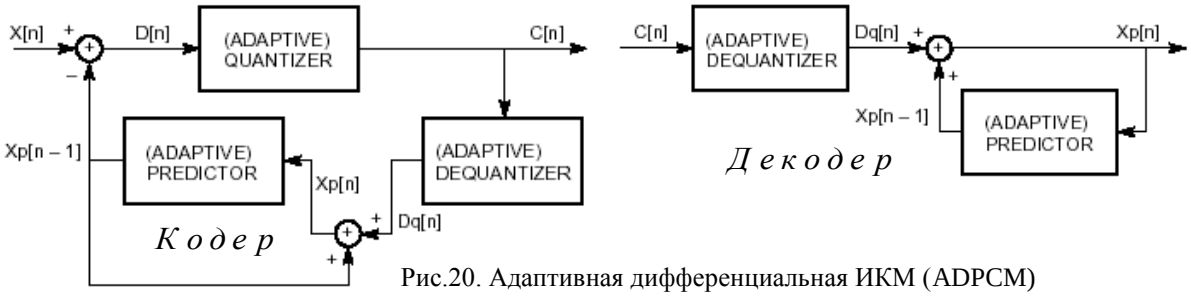

Рис.20. Адаптивная дифференциальная ИКМ (ADPCM)

На рис.20 приведена упрощенная схема ADPCM, на вход которой подается сигнал с кодера линейной ИКМ. Адаптивный квантователь схемы осуществляет изменение шага квантования по заданному алгоритму. Адаптация шага осуществляется по схема, приведенным на рис.4.20. Для формирования разностного сигнала, передаваемого в линию связи, в схеме включен адаптивный реквантователь, который восстанавливает величину шага квантования в цепи обратной связи. Предсказатели в схемах кодера и декодера могут быть как адаптивными, так и не адаптивными. По стандарту ITU-T G.726 сигнал ошибки предсказания передается с использованием только 4 бит при скорости цифрового потока 16, 24 или 40 Кбит/с.

Кодер и декодер реализуются на основе быстродействующего процессора, алгоритм работы у всех фирм-производителей разный. Проблемным вопросом является время адаптации- насколько быстро изменяется шаг квантования. От этого времени существенно зависит качество звучания. Кроме телефонии ADPCM используется при записи интерактивных дисков по стандарту CDROM.

4. Энтропийное кодирование

4.1. Информационная энтропия

В общем случае под информационной энтропией понимается мера хаотичности информации, неопределенность появления какого либо символа первичного алфавита. При отсутствии информационных потерь она численно равна информации на символ передаваемого сообщения.

Понятие энтропии относится только к случайным величинам, имеющим конечное число значений $i = 0,1,2....n)$, распределенных по определенному закону с вероятностью каждого значения равной p_i, причем сумма вероятностей всех исходов равняется единице

$$\sum_{i=0}^{n} p_i = 1$$

Тогда вероятность, что случайная величина x примет значение x_i, определяется равенством

$$P_x(a_i) = p_i .$$

Под энтропией случайного сообщения a_i, имеющего вероятность p_i, понимается количество информации $I(a_i)$, содержащейся в a_i, которое определяется равенством

$$I(a_i) = -p_i \cdot \log_2 p_i . \qquad (19)$$

При использовании логарифма с основанием два, информация измеряется в битах.

Если, например, символ (двоичный) может принимать только две кодовые комбинации с вероятностью 0,5, то энтропия этой кодовой комбинации равна 0,5 бита. Если символ (байт) может принимать n комбинаций, входящих в некий алфавит, то энтропия этого алфавита определяется равенством

$$H(a) = -\sum_{i=1}^{n} p_i \log_2 p_i . \qquad (20)$$

Эта величина называется также *средней энтропией сообщения*, она измеряется числом бит на символ. Величина $\log_2 \frac{1}{p_i} = -\log_2 p_i$ называется частной энтропией, характеризующее только i состояние случайной величины.

Средняя длина передаваемого кода $\bar{L}(a)$ в битах определяется равенством

$$\bar{L}(a) = \sum_{i=1}^{n} P_i(a_i) L(a_i) . \qquad (21)$$

Отношение средней длины передаваемого кода после компрессирования $\bar{L}(a)_{compr}$ к исходному значению $\bar{L}(a)$, называется коэффициентом сжатия.

P_1	P_2	Энтропия
0.99	0.01	0.08
0.90	0.10	0.47
0.80	0.20	0.72
0.70	0.30	0.88
0.60	0.40	0.97
0.50	0.50	1.00

Табл.1. Вероятности и энтропия

В качестве примера рассмотрим энтропию алфавита , состоящего всего из двух символов с вероятностями P_1 и P_2. В этом случае энтропия алфавита выражается числом $H(A) = -P_1 \log_2 P_1 - (1-P_1)\log_2(1-P_1)$.

В табл.1. приведены различные значения вероятностей вместе с соответствующей энтропией. Когда $P_1 = P_2$ необходим, по крайней мере, один бит для кодирования каждого символа. Это означает, что энтропия достигла своего предела и сжатие невозможно. Если вероятности сильно отличаются, то минимальное число требуемых бит на символ снижается. Сжатие возможно, когда энтропия равна 0,08 бит на символ, и совер-

шенно невозможна при энтропии 0,99 бит на символ.

Важно помнить, что энтропия является количеством информации, определенным в контексте вероятностей модели для источника данных. Например, кидание монеты имеет энтропию $-2 \cdot \log_2 0,5 = 1$ бит на одно кидание (при условии его независимости). У источника, который генерирует только одну кодовую комбинацию (вероятность события равна 1), энтропия равна нулю, так как

$$\sum_{i=1}^{\infty} \log_2 1 = 0$$

Энтропия источника данных определяет среднее число битов на элемент данных, требуемых для ее зашифровки без потери информации, при оптимальном кодировании. Заметим, что некоторые биты могут не нести информацию, количество энтропии не всегда выражается целым числом.

Некоторые математические свойства энтропии:

- Неотрицательность: $H(x) \geq 0$.
- Ограниченность: $H(x) < \log_2 x$, равенство, если все кодовые комбинации равновероятны.
- Если $x, y -$ независимы, то $H(x, y) = H(x) + H(y)$.
- Если $x, y -$ имеют одинаковое распределение вероятностей, то $H(x) = H(y)$.

Исходный алфавит кодовых комбинаций, встречающийся на практике, обычно имеет распределение, которое далеко от оптимального, Если исходный алфавит имел n символов, тогда он может быть сравнен с «оптимизированным алфавитом», вероятностное распределение которого однородно. Соотношение энтропии исходного и оптимизированного алфавита – это эффективность исходного алфавита, которая выражается в процентах.

Если кодовые комбинации не независимы, то для учета такого фактора используется условная энтропия. Условной энтропией первого порядка называется энтропия для кодового алфавита, для которого известны вероятности появления одной комбинации после другой. Условная энтропия определяется равенством

$$H(s) = -\sum_i p_i \sum_j p_i(j) \log_2 p_i(j).$$

В этой формуле: $i -$ кодовая комбинация, зависящая от предшествующего символа j, и $p_i(j) -$ вероятность кодовой комбинации i при условии, что предыдущим был символ j. Через частную и общую условную энтропии полностью описываются информационные потери передачи данных в канале с помехами .

Энтропийное кодирование — это кодирование словами (кодами) переменной длины, при котором длина кода символа имеет обратную зависимость от вероятности появления символа в передаваемом сообщении. Обычно при энтропийном кодировании используют для сжатия данных коды, длины которых пропорциональны отрицательному логарифму вероятности символа. Таким образом, наиболее вероятные символы используют наиболее короткие коды.

К энтропийному кодированию относятся три больших класса кодов: префиксные коды , кодирование длин серий и арифметические коды. К префиксным кодам принадлежат код Хаффмана, кодирование Лепеля-Зива, коды Шенона и Шенона-Фано. Среди префиксных кодов наиболее оптимальным является код Хаффмана. Если приблизительные характеристики энтропии потока данных предварительно известны, может быть полезен более простой статистический код, такой как унарное кодирование, гамма-кодирование Элиаса, кодирование Фибоначчи, кодирование Голомба или кодирование Райса.

Энтропийное кодирование эффективно, когда последовательность кодовых символов имеет случайный характер с распределением по закону Лапласа или Гаусса. При исход-

ном линейном квантовании звуковые выборки коррелированы, и тем в большей степени, чем выше частота дискретизации. Поэтому энтропийному кодированию звуковых сигналов всегда предшествует кодирование с предсказанием, при котором происходит декорреляция кодовой последовательности. Реализуется энтропийное кодирование с помощью префиксных кодов, путем кодирования длин серий и использованием арифметических кодов.

Префиксные коды называются так потому, что ни одно кодовое слово не является полным началом (т. е. префиксом) никакого другого слова, что гарантирует однозначность декодирования. Так как длина каждого кодового слова выражается целым числом битов, то префиксные коды неэффективны на алфавитах малой мощности (2-8 символов) или при наличии символов с очень большой (более 30-50%) вероятностью появления и по качеству сжатия могут уступать арифметическим.

Применение блочных кодов, кодирующих не отдельные символы, а блоки из k символов, позволяет построение кодов, сколь угодно близких по качеству кодирования к арифметическим, однако из-за полиномиальной сложности блочного кодирования по размеру блока и ряда других причин блочное кодирование почти не применяется на практике. Как правило, алгоритмы словарного сжатия, и сжатия сортировкой блоков используют для кодирования выхода основного алгоритма сжатия коды Хаффмана.

Кодирование длин серий используется при условии, что в кодовой последовательности может быть подряд много 0 или 1, тогда кодируются только местоположения 1 или 0. Эта идея поясняется рис.21.

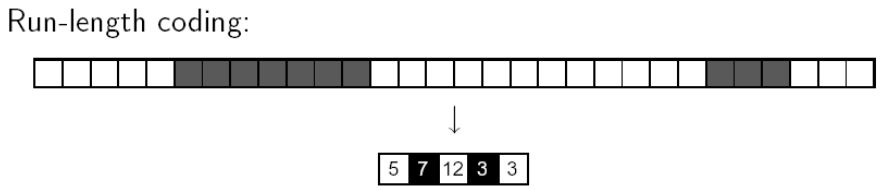

Рис.21. Принцип кодирования длин серий

Арифметические коды не ставят явного соответствия между символами и кодовыми словами, они основаны на других принципах. Качество арифметического кодирования лучше, чем у посимвольного префиксного кодирования, и близко к теоретическому минимуму. Это реализуется также при малой мощности алфавита, и при очень неравномерном распределении вероятностей появления символов.

С другой стороны, кодирование и декодирование арифметических кодов при достаточно большой мощности кодируемого алфавита заметно медленнее кодирования и декодирования префиксных кодов, а разница в качестве сжатия обычно незначительна; по этим и ряду других причин в большинстве случаев префиксное кодирование более предпочтительно для практического использования.

Арифметические коды обычно применяются в сочетании с методами статистического моделирования для кодирования символов в соответствии с предсказанными вероятностями .

Энтропийное кодирование применяется в целом ряде алгоритмов сжатия звука без потерь:

- Apple Lossless — ALAC (Apple Lossless Audio Codec)
- Audio Lossless Coding — также известен как MPEG-4 ALS
- Direct Stream Transfer — DST
- Dolby TrueHD
- DTS-HD Master Audio
- Free Lossless Audio Codec — FLAC
- Meridian Lossless Packing — MLP
- Monkey's Audio — Monkey’s Audio APE

- OptimFROG
- RealPlayer — RealAudio Lossless
- Shorten — SHN
- TAK — (T)om's verlustfreier (A)udio (K)ompressor (нем.)
- TTA — True Audio Lossless
- WavPack — WavPack lossless
- WMA Lossless — Windows Media Lossless

Табл.2. Энтропийное кодирование с предсказанием		
Системы кодирования	Модель предсказания	Энтропийное кодирование
Shorten	FIR-предсказание	Rice
DVD	IIR - предсказание	Haffman
MusiCompress	Адаптивная аппроксимация	Haffman
AudioPack	Полиномиальная аппроксимация	Golombo
LTAC	Ортогонльные преобразования	Rice
IniMDCT	Интегральные преобразования	Haffman
C-LPAC	FIR-предсказание	Моделирование

Краткая характеристика некоторых из этих кодов приведена в табл.2. Из приведенных алгоритмов наибольший практический интерес представляют, те которые стандартизированы и массово применяются в цифровой аудиотехнике при звукозаписи и передачи данных по каналам связи. К ним относятся:

- Audio Lossless Coding — также известен как MPEG-4 ALS, предназначен для широкого применения.
- Direct Stream Transfer — DST, применяется в звукозаписи по стандарту Super Audio CD,
- Meridian Lossless Packing — MLP, применяется в звукозаписи по стандарту DVD-Audio

4.2. Коды переменной длины

Первое правило построения кодов с переменной длиной символов очевидно. Короткие коды следует присваивать часто встречающимся символам, а длинные- редко встречающиеся. Есть и другая проблема. Эти коды надо назначать так, чтобы их можно было декодировать только однозначно, а не двусмысленно.

Пусть четыре двоичных символа a_1, a_2 a_3 и a_4 длиной 2 появляются с равной вероятностью 0,25, тогда им присваиваются двух битовые комбинации: 00, 10, 01 и 11. Это значит, что для передачи каждой кодовой комбинации требуются два бита. Так как все кодовые комбинации имеют равные вероятности, то коды переменной длины не в состоянии сжать эти данные.

Пусть теперь эти четыре символа имеют разные вероятности, приведенные в табл.4.3.

Символ	Вероятность	*Code*1	*Code*2
a_1	0.49	1	1
a_2	0.25	01	01
a_3	0.25	010	000
a_4	0.01	001	001

Табл.3. Коды переменной длины

В этом случае имеется избыточность, которую можно устранить с помощью переменных кодов и сжать так, чтобы требовалось меньше двух бит на символ. В соответствии с теорией информации наименьшее число бит на символ в среднем равно 1,57, это есть энтропия множества символов. В табл.3 *code* 1 присваивает самому часто встречающемуся символу самый короткий код. В этом коде среднее число бит на символ (энтропия) равна: $1\times0,49+2\times0,25+3\times0,25+3\times0,01=1,77$. Это число весьма близко к теоретическому минимуму.

Пусть последовательность из 20 символов закодирована 37 битами в *code* 1:

1|010|01|1|010|010|001|01|1|1|01|01|1|1|010|1|1|01|010|1,

Если попробовать декодировать эту последовательность, то окажется, что *code* 1 из табл.

3. не может быть декодирован и использовать его нельзя. Первый бит последовательности равен 1, поэтому первым символом может быть только символ a_1, так как ни какой другой код не начинается с 1. Следующий код последовательности равен 0 , но коды a_2, a_3 и a_4 оба начинаются с 0, поэтому декодер должен читать следующий бит, он равен 1, однако коды для a_2 и a_3 оба имеют в начале 01 – декодер не знает, как ему поступить, декодирование невозможно.

В отличие от $code$ 1 к $code$ 2 имеет важное свойство , которое называется свойством префикса. Это свойство предполагает, что если некоторая последовательность битов выбрана в качестве символа, то ни один код другого символа не может начинаться с этой последовательности, не может быть префиксом или приставкой. Раз строка «1» уже выбрана для символа a_1, то ни одна другая кодовая комбинация не может начинаться с 1. Если строка «01» выбрана в качестве кода для a_2, то никакая другая кодовая комбинация не может начинаться с 01.

Таким образом, при выборе кода переменой длины следует назначать более короткие коды часто встречающимся символам и выбранный код должен удовлетворять свойству префикса. Для декодирования префиксного кода декодер должен знать префиксный код каждого символа.

Решение этой задачи достигается 3 путями. Первое - префиксный код выбирается и используется как кодерам, так и декодером. Во втором варианте кодер выполняет свою работу в два этапа. Сначала он читает сжимаемый файл и собирает необходимые статистические сведения. На втором этапе на основе полученной статистической информации выбирается наилучший префиксный код и происходит сжатие, при этом таблицу кода следует передавать по линии связи декодеру. В третьем варианте кодер сразу начинает работать, не зная статистических свойств сжимаемого файла. По мере сжатия и сбора статистической информации кодер улучшает префиксный код, что приводит к увеличению компрессии. Декодер повторяет все операции кодера. Такое кодирование называется адаптивным.

4.3. Код Хаффмана

Метод сжатия информации на основе двоичных кодирующих деревьев был предложен Д. А. Хаффманом в 1952 году задолго до появления современного цифрового компьютера. Обладая высокой эффективностью, он и его многочисленные адаптивные версии лежат в основе многих методов, используемых в современных алгоритмах кодирования. Код Хаффмана редко используется отдельно, чаще работая в связке с другими алгоритмами кодирования. Метод Хаффмана является примером построения *кодов переменной длины*, имеющих *минимальную среднюю длину*. Этот метод производит *идеальное сжатие*, то есть *сжимает данные до их энтропии*, *если вероятности символов точно равны отрицательным степеням числа 2.*

Стоит отметить, что за 50 лет со дня опубликования, метод Хаффмана ничуть не потерял своей актуальности и значимости. Так с уверенностью можно сказать, что мы сталкиваемся с ним, в той или иной форме практически каждый раз, когда архивируем файлы, смотрим фотографии, фильмы, посылаем факс или слушаем музыку.

Идея, лежащая в основе метода Хаффмана, достаточно проста. Вместо того чтобы кодировать все символы кодовой последовательности одинаковым числом бит, как это делается в ASCII кодировке, и где на каждый символ отводится ровно по 8 бит, в коде Хаффмана кодируются символы, которые встречаются чаще, меньшим числом бит, чем те, которые встречаются реже. Коды Хаффмана имеют префикс, что и позволяет однозначно их декодировать, несмотря на их переменную длину. *Динамический алгоритм Хаффмана* на входе получает *таблицу частот символов* в сообщении. Далее на основании этой таблицы строится *дерево кодирования Хаффмана*. Поэтому первый алгоритм, опубликованный

Хаффманом, был двухпроходный. На первом проходе строится частотный словарь кодовых символов и генерируются коды. На втором проходе происходит непосредственно кодирование.

Код Хаффмана задается алфавитом $A = a_1, a_2, a_3, ...a_n$ - из n различных символов с известной частотой их появления $P = p_1, p_2, p_3, ...p_n$. Под термином «частота» понимается относительная частота событий, их вероятность, специалисты в области теории информации это понятие определяют абстрактным термином «вес». Этому алфавиту соответствует набор двоичных кодовых комбинаций $C = c_1, c_2, c_3, ...c_n$ длиной $L = l_1, l_2, l_3, ...l_n$. В коде Хаффмана:

$$(1)\ c_i \text{ не является префиксом } c_j,$$

$$(2)\ \sum_{i=1}^{n} p_i l_i - \text{ минимальна},$$

поэтому он называется минимально-избыточным префиксным кодом. Длина кода Хаффмана всегда не превосходит $\text{ceil} - \log_2 P_i$, где ceil – означает округление числа в скобках до целого значения в большую сторону.

Свойство (1) называется свойством префиксности. Оно позволяет однозначно декодировать коды переменной длины. Сумму в свойстве (2) можно трактовать как размер закодированных данных в битах. Это позволяет оценить степень сжатия, не прибегая непосредственно к кодированию.

Известно, что любому бинарному префиксному коду соответствует определенное бинарное дерево. Бинарное дерево, соответствующее коду Хаффмана, называется деревом Хаффмана. Строится оно на основе таблицы частот (кодовая таблица) встречаемости символов в сообщении. Задача построения кода Хаффмана равносильна задаче построения соответствующего ему дерева.

Алгоритм построения дерева Хаффмана рассмотрим на конкретном простейшем при-

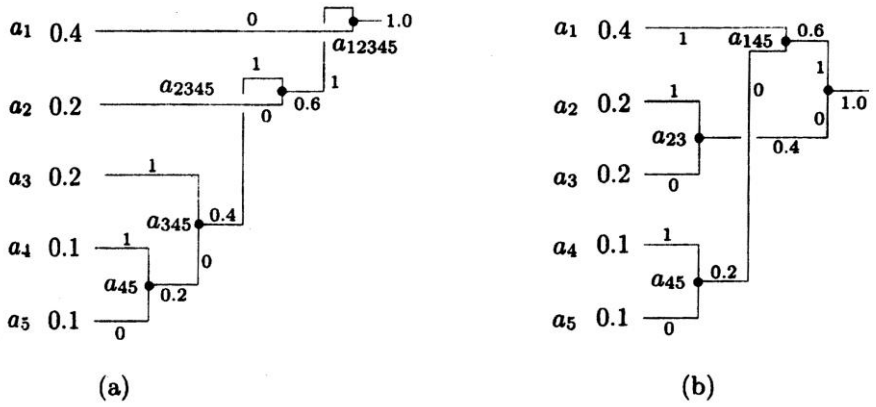

Рис.22. Коды Хафмана

мере. Корень дерева располагается сверху, его листьями служат символы алфавита $A = a_1, a_2, a_3, ...a_n$, а ветви – это линии соединяющие корень дерева с листьями (рис.4.22). Построение дерева начинается снизу вверх до корня дерева, затем начинается спуск вниз по дереву, чтобы построить двоичный код для каждого символа. Он строится справа налево (от младшего разряда к старшему). Алгоритм кодирования включается в себя следующие шаги.

- Составляется список кодируемых символов с указанием их вероятностей, удобно расположить их в таблице в порядке убывания их вероятности (рис.22.)

- Объединяются два символа алфавита с наименьшими вероятностями a_4 и a_5 , и создается вспомогательный символ a_{45}, которому приписывается вероятность 0,2, равная сумме вероятностей выбранных символов. Далее этот новый символ участвует на равных в построении дерева.
- Объединяются следующие два символа алфавита с наименьшими вероятностями a_3 и a_{45}, (из них один новый), и создается вспомогательный символ a_{345} . Ему приписывается вероятность 0,4 , равная сумме вероятностей выбранных символов .
- Объединяются два символа алфавита с наименьшими вероятностями a_2 и a_{345}, , и создается вспомогательный символ a_{2345}. Ему приписывается вероятность 0,6, равная сумме вероятностей выбранных символов .
- Наконец объединяются два оставшихся символа a_1 и a_{2345}, заменяем их

вспомогательным символом a_{12345} с вероятностью 1. Это корень дерева.

Для создания кодовых комбинаций символов произвольно приписываем 1 бит верхней ветке и 0 бит нижней ветке дерева для каждой пары ветвей дерева. В результате получаем следующие коды: 0, 10, 111, 1101 и 1100 . Средняя длина этого кода 2,2 бита на символ, а энтропия равна 2,12 бита на символ.

Некоторый произвол в построении дерева позволяет получить множество кодов Хаффмана с одинаковой средней длиной. На рис.22 показано, как можно объединить символы по другому алгоритму и получить иной код Хаффмана. Средняя длина этого кода также равна 2,2 бит/символ, но лучшим является код с наименьшей дисперсией. Дисперсия показывает, насколько сильно откланяется длина кодовых комбинаций l_i от среднего значения $\bar{L}$ и рассчитывается по формуле

$$D = \sum_{i=1}^{n} p_i (l_i - \bar{L})^2 . \qquad (22)$$

Дисперсия кода на рис.22.а) равна 1,36 , а дисперсия кода на рис.22.в) существенно меньше, и равна 0,16 . Когда на дереве имеются более двух символов с наименьшей вероятностью, следует объединять символы с наибольшей и наименьшей вероятностью, это сокращает общую дисперсию кода .

Интересен результат преобразования обычного двоичного 8-разрядного кода с алфавитом из 256 символов длиной 8 бит (байт) в код Хаффмана. При этом в кодовую таблицу входят:

4 - 2 разрядных кода;
8 - 3 разрядных кодов;
16 - 4 разрядных кодов;
32 - 5 разрядных кода;
64 - 6 разрядных кодов;
128 - 7 разрядных кодов;
2 - 8 разрядных кода.

В итоге мы имеем также 256 различных комбинаций, которыми можно кодировать байт. Из этих комбинаций лишь 2 по длине равны 8 битам. Если мы сложим число битов, которые представляют все эти кодовые комбинации, то в итоге получим 1554 бит или 195 байтов. Так в результате кодирования 256 байт сжаты до 195, и получен выигрыш максимально 33%, это без учета информации, находящейся в таблице кодирования. Некоторые кодовые комбинации , такие как 10000000 и 01111111, сжимаются в 4 раза.

Заметим, что если символы алфавита равновероятны, то применение кодов переменной длины не дает никаких преимуществ. Это объясняется тем, что при размере алфавита, определяемого значением 2^n, получаются коды фиксированной длины. В других случаях коды весьма близки к кодам с фиксированной длиной. В табл.4. в качестве примера приведены средняя длина и дисперсии таких кодов Хаффмана с алфавитом от 5 до 8 сим-

волов. Однако, есть примеры строк, в которых символы равновероятны, но не являются

n	p	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	Ср.длина	Дисперсия
5	0.200	111	110	101	100	0				2.6	0.64
6	0.167	111	110	101	100	01	00			2.672	0.2227
7	0.143	111	110	101	100	011	010	00		2.86	0.1226
8	0.125	111	110	101	100	011	010	001	000	3	0

Табл. 4. Коды Хаффмана для 5…8 символов

случайными, и их можно сжимать. Хорошим примером является последовательность $a_1, a_1 ... a_1, a_2, a_2 ... a_2, a_3, a_3$. Заметим, что код Хаффмана не работает с двух символьным алфавитом. В таком алфавите одному символу придется присвоить код 0, другому - 1. Метод Хаффмана не может присвоить одному символу код короче одного бита.

В общем случае декодер тем или другим способом получает информацию о дереве Хаффмана и алфавите кода, только в этом случае возможно декодирование. Алгоритм декодирования очень прост. Эта операция начинается с корня дерева и читается первый бит сжатого файла. Если это нуль, следует двигаться по нижней ветке дерева, если единица, то двигаться надо по верхней ветке дерева. Далее читается второй бит и происходит движение по следующей ветке по направлению к листьям. Когда декодер достигнет листа дерева, он узнает код первого несжатого символа (обычно это символ ASC11). Процедура повторяется для следующего бита, начиная опять с корня дерева.

В кодере и декодере кода Хаффмана может использоваться одно и то же дерево, усредненное по многочисленным сообщениям. Тогда его не надо строить и передавать вместе с сообщениями, отпадает необходимость первого прохода при кодировании . Иногда такое дерево может оказаться не оптимальным, поэтому удобно иметь несколько деревьев, одинаковых в кодере и декодере, для передачи информации различного характера.

Классический алгоритм Хаффмана имеет один существенный недостаток. Для восстановления содержимого сообщения декодер должен знать таблицу кодирования, которой пользовался кодер. Следовательно, длина сжатого сообщения увеличивается на длину этой таблицы, которая должна посылаться впереди данных, что приводит к увеличению размеров выходного файла, и, следовательно, снижению степени сжатия. Кроме того, необходимость наличия полной статистики по вероятности кодовых комбинаций перед началом собственно кодирования требует двух проходов по сообщению: одного для построения модели сообщения (таблицы кодирования - дерева), другого для собственно кодирования.

Следующим шагом в развитии алгоритма Хаффмана является его адаптивная версия. Адаптивное сжатие Хаффмана позволяет не передавать модель сообщения вместе с ним самим и ограничиться одним проходом по сообщению, как при кодировании, так и при декодировании. Практически любая форма кодирования может быть конвертирована в адаптивную. Схема адаптивного кодирования/декодирования работает благодаря тому, что при кодировании и при декодировании используются одни и те же процедуры . И компрессор, и декомпрессор начинают с "пустой" модели (не содержащей информации о сообщении) и с каждым просмотренным символом обновляют ее одинаковым образом.

Хорошо было бы, чтобы кодер не тратил зря кодовое пространство на символы, которые не встречаются в сообщении. Если речь идет о классическом алгоритме Хаффмана, то те символы, которые не встречаются в сообщении, уже известны до начала кодирования, так как известна таблица частот и символы, у которых частота встречаемости равна 0. В адаптивной версии алгоритма мы не можем знать заранее, какие символы появятся в сообщении. Можно проинициализировать дерево Хаффмана так, чтобы оно имело все 256 символов алфавита (для 8-и битовых кодов) с частотой появления, равной 1. В начале кодирования каждый код будет иметь длину 8 битов. По мере адаптации модели наиболее часто встречающиеся символы будут кодироваться все меньшим и меньшим количеством

битов. Такой подход работоспособен, но он значительно снижает степень сжатия, особенно на коротких сообщениях.

Лучше начинать моделирование с пустого дерева и добавлять в него символы только по мере их появления в сжимаемом сообщении. Но это приводит к очевидному противоречию: когда символ появляется в сообщении первый раз, он не может быть закодирован, так как его еще нет в дереве кодирования. Чтобы разрешить это противоречие вводится специальный код, который означает, что следующий символ закодирован вне контекста модели сообщения.

4.4. Арифметическое кодирование

Метод Хаффмана обладает очень высокой эффективностью только при условии, что у формируемого кода переменной длины средняя длина равна энтропии алфавита. Однако, это возможно только в случаях, когда вероятности символов алфавита являются степенями числа 2, то есть равны 1/2, 1/4, 1/8 и т.п. Тогда по методу Хаффмана каждому символу алфавита присваивается код с целым числом битов. Теория информации предсказывает, что при вероятности символа, скажем, 0.4, ему в идеале следует присвоить код длины 1.32 бита, а метод Хаффмана присвоит этому символу код длины 1 или 2 бита, что снижает эффективность кода. Эта проблема решается при арифметическом кодировании.

Арифметическое кодирование – это блоковое кодирование, при котором *кодируются не символы*, как в коде Хаффмана, а *блоки данных, состоящие из множества символов*. Сжатый блок данных представляется *одним кодовым словом*, часто очень большой длины. При кодировании обрабатывается *символ за символом* входного блока данных и *добавляются биты* к сжатому файлу. Блок аудиоданных, сжатый арифметическим кодером, рассматривается как некоторая *двоичная дробь из интервала* [0, 1). Результат сжатия можно представить как последовательность двоичных цифр из записи этой дроби.

Алгоритм арифметического кодирования наиболее наглядно представляется в графическом виде. В этом алгоритме блок данных, включающий все символы, представляется в виде отрезка линии с границами [0,1). Каждый символ блока представляется отрезком на числовой оси с длиной, равной вероятности его появления и началом, совпадающим с концом отрезка символа, предшествующего ему в алфавите. Сумма всех отрезков, очевидно должна равняться единице (рис.23).

Кодируются символы шагами, последовательно, один за другим. На каждом шаге текущий интервал, соответствующий вероятности кодируемого сигнала, уменьшается пропорционально вероятности следующего символа. Этот символ «вырезает» из текущего интервала подынтервал пропорционально своей вероятности. После каждого шага кодирования, а текущий интервал становится все меньше, поэтому требуется все больше бит для его представления в двоичном коде. Кодом сообщения является интервал, выделенный после кодирования всех его символов, точнее, любое число, входящее в этот интервал. Длина полученного интервала пропорциональна вероятности кодируемого блока данных. Результатом кодирования является единственное двоичное кодовое слово, в котором сжата вся информация входного блока .

Перед кодированием определяются экспериментально или берутся из таблиц частоты следования каждого символа алфавита. Эта операция обычно осуществляется на первом проходе алгоритма сжатия. Однако, если можно получить хорошие оценки частот символов из другого источника, первый проход можно опустить.

Основой арифметического кодирования является алгоритм определения верхней и нижней границ интервалов кодирования каждого символа.. Они , определяются равенствами:

для первого символа

$$G(low)_1 = L_1, \qquad G(high)_1 = H_1,$$

для последующих символов

$$G(low)_i = L_{i-1} + L_i \cdot P_{i-1}, \quad G(high)_i = L_{i-1} + H_i \cdot P_{i-1},$$

где H_i – верхняя граница i символа на шкале [0, 1), L_i – нижняя граница i символа на шкале [0,1), P_i – вероятность i символа . С каждым новым следующим символом переменные $G(low)$ и $G(high)$, пересчитываются .

В качестве примера рассмотрим кодирование блока данных, в который входят информационные символы w, u и v с вероятностями $P_w = 0,2$, $P_u = 0,35$, $P_v = 0,2$. Эти символы передаются в виде последовательности: $wuvw$. В этот же блок входят служебные символы: x, y, z , которые не кодируются. В соответствии с приведенным формулами рассчитывается верхняя и нижняя границы интервалов кодируемых символов на отрезке [0, 1), которые приведены на рис.23 . Результатом кодирования является интервал от 0,5822 до 0,5850, в котором произвольно можно выбрать любое число. Примем среднее значение интервала 0,5836, ему соответствует двоичное значение 1011011001100. Среднюю длину кода можно найти, разделив размер выхода (в битах) на размер входа (в символах). Отметим, что вероятности, которые использо-

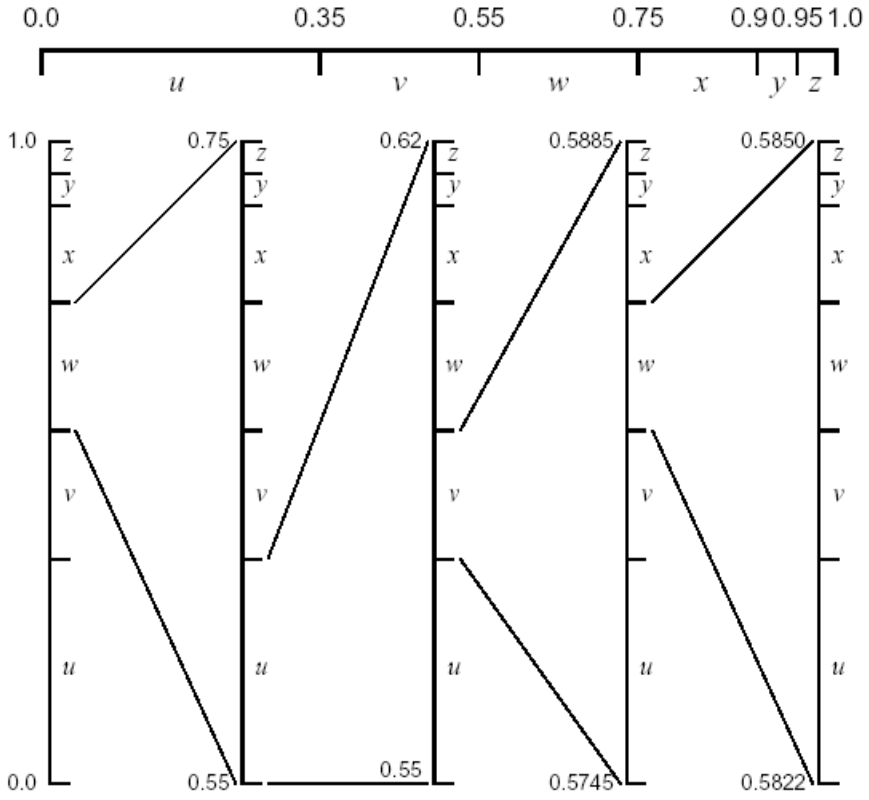

Рис.23. Алгоритм арифметического кодирования

вались в процессе кодирования, могут меняться от блока к блоку, это используется в адаптивной стратегии арифметического кодирования.

Декодирование осуществляется также достаточно просто в соответствии с алгоритмом определяемом равенством

$$N_i = \frac{N_{i-1} - G(low)_{i-1}}{P_{i-1}}$$

Пусть кодовое значение сжатого блока равно 0,58 (в десятичной форме.). На интервале [0, 1) это кодовое значение попадает в подынтервал [0.3, 0.8), который соответствует символ w , значит он является первым в кодовой последовательности. По этой формуле рассчитываем, что следующий символ является числом 0,44, которое также попадает в подынтервал [0.3, 0.8), и, следовательно, вторым символом будет также w . Таким образом декодирование происходит до тех пор, пока не будут определены все символы блока. Декодирование кончает-

ся, когда декодер обнаруживает кодовую комбинацию, означающую конец файла.

Следующий пример иллюстрирует громоздкость вычислений при арифметическом кодировании. В этом примере в блок данных входят три разных символа, вероятности которых приведены в табл. 4.5а. Заметим, что эти вероятности сильно отличаются друг от друга. Одна - большая, 0.975, а другие -существенно меньше. Это случай *асимметричных вероятностей.* Кодирование строки $a_2\ a_2\ a_1\ a_3\ a_3$», выдаёт числа с точностью в 16 знаков, приведенные в табл. 4.6, в которой для каждого символа в двух строках записаны последовательные значения *Low* и *High* .

На первый взгляд кажется, что полученный код длиннее исходной строки. Однако следует иметь в виду, что каждый символ состоит из 8 или более бит и сжатие определяется отношением общего числа бит входного блока (строки) к числу бит выходного кодового слова кодера.

Симв.	Вероятн.	Область	Симв.	Вероятн.	Область
a_1	0.001838	[0.998162,1.0)	eof	0.000001	[0.999999,1.0)
a_2	0.975	[0.023162,0.998162)	a_1	0.001837	[0.998162,0.999999)
a_3	0.023162	[0.0,0.023162)	a_2	0.975	[0.023162,0.998162)
			a_3	0.023162	[0.0,0.023162)
	(a)			(b)	

Табл. 4.5. **(Асимметрические) вероятности трех символов.**

a_2	$0.0 + (1.0 - 0.0) \times 0.023162$	=	0.0231 62
	$0.0 + (1.0 - 0.0) \times 0.998162$	=	0.998 162
a_2	$0.0231\,62 + 0.975 \times 0.023162$	=	0.04574495
	$0.0231\,62 + 0.975 \times 0.998162$	=	0.996369 95
a_1	$0.04574495 + 0.950625 \times 0.998162$	=	0.99462270125
	$0.04574495 + 0.950625 \times 1.0$	=	0.99636995
a_3	$0.99462270125 + 0.00174724875 \times 0.0$	=	0.99462270125
	$0.99462270125 + 0.00174724875 \times 0.023162$	=	0.994663171025547
a_3	$0.99462270125 + 0.00004046977554749998 \times 0.0$	=	0.994 62270125
	$0.99462270125 + 0.00004046977554749998 \times 0.023162$	=	0.994623638 610941

Табл. 4.6. **Кодирование строки $a_2a_2a_1a_3a_3$.**

Символ	Code-Low		Область
a_2	0.994 62270125 – 0.023162	= 0.97146170125	/ 0.975 = 0.99636995
a_2	0.99636995 – 0.023162	= 0.97320795	/ 0.975 = 0.998162
a_1	0.998162 – 0.998162	= 0.0	/ 0.00138 = 0.0
a_3	0.0 – 0.0	= 0.0	/ 0.023162 = 0.0
a_3	0.0 – 0.0	= 0.0	/ 0.023162 = 0.0

Табл. 4.7. **Декодирование строки $a_2a_2a_1a_3a_3$.**

Вероятности символов из табл.4.5 а равны 0.975, 0.001838 и 0.023162. Эти величины требуют довольно много десятичных цифр для записи, а конечные значения *Low* и *High* в табл.4.6 равны 0.99462270125 и 0.994623638610941. Опять кажется, что тут нет никакого сжатия, однако анализ энтропии показывает отличное сжатие и в этом случае.

Вычисляем вероятность строки «$a_2\ a_2\ a_1\ a_3\ a_3$» и получаем число

$$0.975^2 \cdot 0.001838 \cdot 0.023162^2 \approx 9.37361 \cdot 10^{-7}$$

а ее энтропия будет равна

$$-\log(9.37361 \cdot 10^{-7}) \approx 20.0249$$

В двоичном представлении значения переменных *Low* и *High* соответственно равны 0.111111101001111110010111111001 и 0.111111101001111110011110 1. Можно выбрать любое число из этого промежутка, и мы выбираем «1111111010011111100». Этот код имеет длину 19 (он и теоретически должен быть 21-битным, но числа в табл. 4.6 имеют ограниченную точность).

Если размер входного файла известен, тогда можно его кодировать без символа “eof” . Кодер может начать с записи его размера (в несжатом виде) в заголовок выходного файла. Декодер прочитает этот размер, начнет процесс декомпрессии и остановится после извлечения всех символов. Если декодер читает сжатый файл байт за байтом, то кодер может добавить в конце несколько нулевых битов, чтобы сжатый файл мог быть прочитан блоками по 8 бит. Файл объема 1 МВ будет сжиматься, скажем, до 500 КВ, в котором будет записано всего одно двоичное число.

Любое практическое применение арифметического кодирования должно основываться на оперировании с целыми числами, так как арифметика чисел с плавающей запятой работает медленно и при этом происходит потеря точности.

5. Компрессирование без потерь по стандарту SACD

5.1. Структура кодирования

В системе *SACD* в 1-битовом формате производится одновременно стерео и 6 канальная запись. Скорость цифрового потока информационных данных в одном канале равна 2.82 Мб/с, а емкость пользователя носителя составляет 4.7 Гбайт. При этом общая скорость цифрового потока достигает 22.6 Мб/с и длительность такой записи всего 27.8 мин. Для обеспечения длительности записи 74 мин необходимо использовать компрессию скорости цифрового потока со средним коэффициентом сжатия 2.66. Чтобы при компрессировании качество передачи звуковой информации не снижалось, используется алгоритм компрессии без потерь, подобный хорошо известному алгоритму “pkzip”, который разработан для персональных компьютеров.

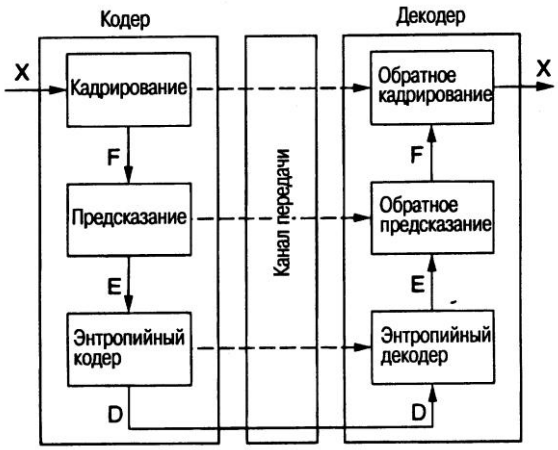

Рис.24. Кодек DST

В общем виде алгоритм компрессии однобитового цифрового потока, под фирменным названием DSD (Direct Sream Digtal), представлен на рис.24. Первым этапом является преобразование непрерывного потока двоичных данных в формате DSD в последовательность блоков данных достаточно большой емкости, называемых кадрами. На следующем этапе производится кодирование с предсказанием, при котором производится компрессирование скорости формируемого цифрового потока. Дальнейшее компрессирование осуществляется с использованием энтропийного кодирования сигнала ошибки предсказания. В результате на выходе кодера формируется непрерывный однобитовый сжатый цифровой поток под названием DST(Direct Stream Transfer). , который записывается на оптический диск в формате SACD.

Коэффициент компрессии определяется равенством

$$K_{comp} = \frac{N_{DSD}}{N_{DST}},$$

где N_{DSD} и N_{DST} – число бит в блоке до и после компрессирования. Этот коэффициент может меняться от блока к блоку. Основной интерес представляет его среднее значение за

все время музыкальной программы.

При декодировании в процессе воспроизведения записи осуществляется энтропийное декодирование, декодирование с предсказанием и преобразование восстановленных блоков данных в непрерывный двоичный поток в формате DSD.

Организация цифрового потока. Перед кодированием исходный 1 битовый звуковой поток, состоящий из символов $b \in -1, +1$, разделяется на блоки по 37632 бита. При частоте дискретизации 2.82 МГц частота повторения блоков равна 1/75 с. Такая организация потока данных преследует несколько целей.

Прежде всего, блоковая организация цифрового потока необходима для обеспечения возможности использования кодирования с адаптивным предсказанием. Такое кодирование эффективно только при высокой степени корреляции между звуковыми выборками, по которым осуществляется предсказание. Это значит, что звук в пределах длительности блока должен быть стационарным или квази- стационарным процессом. Частота повторения блоков выбирается исходя из необходимости обеспечения этого требования.

В качестве энтропийного кодирования в процессе компрессирования используется арифметический код, который принципиально может работать только при блоковой организации данных. В этом коде сжимаются данные блоков, и они преобразуются в последовательность двоичных слов.

Последнее замечание, появляется возможность легкого доступа к аудиоданным в проигрывателе. По этой же причине, блоки должны кодироваться независимо, что позволяет их декодировать по отдельности , без всякой связи с другими блоками.

5.2. Кодирование с предсказанием

Такое кодирование предназначено для компрессирования цифрового потока за счет сокращения статистической избыточности звукового сигнала. Это достигается тем, что кодируется не сам звуковой сигнал, а ошибка предсказания. Поэтому главным в таком кодировании является то, чтобы эта ошибка была минимальна.

В схеме кодера DST на рис.25 предсказатель включает в себя фильтр предсказателя , цифровой вычитатель (XOR) и реквантователь Q(z). В такой схеме осуществляется предсказание «вперед» по более ранним выборкам ЗС. Передаточная функция фильтра предсказателя $H(z) \cdot z^{-1}$ создает временную задержку выходного сигнала, на один такт, что необходимо для декодирования. Коэффициенты цифрового фильтра рассчитываются для каждого блока данных, хранятся в таблице и передаются через тракт записи в закодированном виде для обеспечения работы декодера. Они меняются в зависимости от характера звукового сигнала, поэтому процесс предсказания в этой схеме является адаптивным.

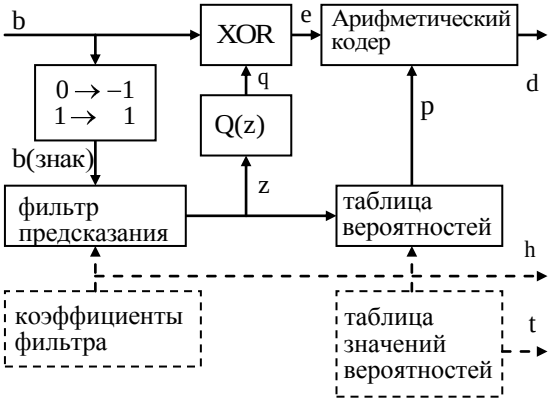

Рис.25. Схема кодера DST

В блоке XOR осуществляется формирование сигнала ошибки предсказания e, путем определения разности между сигналом исходного цифрового потока b и предсказанным сигналом с выхода фильтра q. Если в цифровых потоках b и q передаваемые данные сильно коррелированы, то сигнал ошибки e не имеет избыточности, является случайной величиной с распределением близким к закону Гаусса или Лапласа. Однако, сигнал e и коэффициента фильтра предсказателя вместе несут такую же информацию как сигнал b, что позволяет реконструировать сигнал в декодере без искажений.

Фильтр предсказания разрабатывается по стандартным методам на основе минимиза-

ции среднеквадратической ошибки интерполяции (MMSE-minimum mean squared error). В приведенной схеме для одно-битового цифрового потока используется FIR фильтр. Сигнал z с выхода этого фильтра является многобитовым словом, поэтому на вычитатель он подается через реквантователь Q(z), формирующий однобитовый поток данных.

Выходной сигнал фильтра является функцией предшествующих бит

$$z(n) = \sum_{i=1}^{L} h_i \cdot b(n-i), \quad (23)$$

где h_i – коэффициенты FIR фильтра, n – номер выборки, i – задержка, L – длина фильтра. Назначение фильтра – предсказать следующий бит b. Ошибка предсказания в момент n

$$\varepsilon(n) = z(n) - b(n). \qquad (24)$$

Квадратичное значение этой ошибки

$$\varepsilon^2(n) = (z(n) - b(n))^2.$$

Условие минимизации ошибки записывается в виде равенства

$$\sum_{n=1}^{M} \varepsilon^2(n) = \sum_{n=1}^{M} \left(\sum_{i=1}^{L} h_i \cdot b(n-i) - b(n) \right)^2,$$

где M – число бит в блоке. Это уравнение в процессе кодирования решается относительно всех коэффициентов h_i для каждого блока данных.

В общем случае FIR фильтр должен быть минимально фазовым (все его нули находятся внутри единичной окружности в Z- плоскости). Для достижения оптимального баланса между точностью предсказания и числом бит, необходимым для описания фильтра, коэффициенты фильтра предсказания квантуются 9-разрядными двоичным кодом с плавающей запятой. В кодере и декодере используется идентичные фильтры, поэтому фильтр плеера является наиболее дорогим элементом, и поэтому важно, чтобы этот фильтр стоил как можно дешевле.

При 1-битовых сигналах реализации FIR фильтра является совершенно прямолинейной. В связи с тем, что величина b определяет только знак коэффициента фильтра h выполнение операции по формуле (23) сводится к суммированию. Предсказанное фильтром значение z является многобитовым, оно преобразуется в однобитовый поток в реквантователе $Q(z)$, который выполняет операцию усечения (младшие разряды обрасываются), поэтому

$$q(n) = \begin{matrix} 1, & if\ z(n) \geq 0 \\ 0, & if\ z(n) < 0 \end{matrix},$$

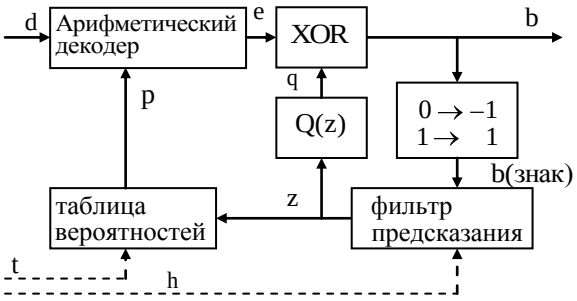

Рис.26. Схема декодера DST

Сигнал ошибки предсказания e формируется при выполнении операции исключительного –ИЛИ с сигналами b и q

$$e(n) = \begin{matrix} 0, & if\ b(n) = q(n) \\ 1, & if\ z(n) \neq q(n) \end{matrix}.$$

Задачей фильтра предсказателя является создание в кодовом потоке сигнала e как можно больше 0, что и это позволяет снизить скорость цифрового потока. Например, поток из 1000 нулей и 1 одной единицы может быть передан указанием только позиции 1. Это позволяет передать 1000 бит 10-разрядным словом (позиция единицы). При этом коэффициент сжатия равен 100.

Для обеспечения возможности реконструкции исходного цифрового потока бит, коэффициенты фильтра предсказания для каждого блока передаются на запись в закодированном виде. В декодере (рис.26) восстанавливается исходный цифровой поток с использованием переданных бит ошибки предсказания e и сигнала с выхода фильтра

предсказания z (все вычисления осуществляются точно также как в кодере)

$$b(n) = \begin{matrix} 0, & if\ e(n) = q(n) \\ 1, & if\ e(n) \neq q(n) \end{matrix} .$$

Важное значение имеет зависимость величины компрессии от характеристик FIR фильтра предсказателя, в частности от порядка фильтра. Из рис.27 видно, что с увеличением порядка фильтра коэффициент компрессии η увеличивается, сначала быстро, потом медленнее. Оптимальное значение $\eta = 2,45$ достигается при порядке фильтра 128, который определяет порядок предсказателя. Однако, на некоторых жанрах музыкальных произведений увеличение порядка фильтра свыше 40 не приводит к увеличению коэффициента компрессии

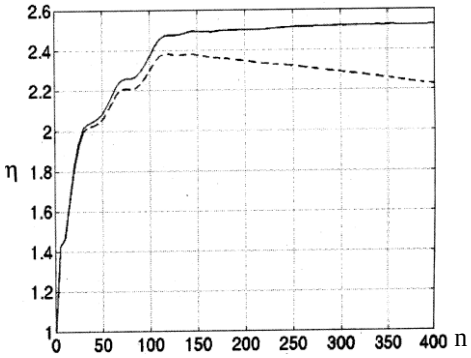

Рис.27. Зависимость коэффициента компрессирования η от порядка предсказателя

5.3. Энтропийное кодирование

При правильном кодировании с предсказанием сигнал e содержит больше 0, чем 1. В соответствии с теорией информации это позволяет осуществить дальнейшую компрессию цифрового потока при условии, что известны вероятности символов 1 и 0 (p, (1-p)) в потоке e, которые приводятся в таблице в функции значения z. Минимальное число бит, с помощью которых, в среднем, единственный бит потока e может быть представлен, определяется равенством

$$N_{бит} = -\ p \cdot \log_2(p) + (1-p) \cdot \log_2(1-p) \ .$$

Если p = 0.5, то фильтр предсказывает 50% вероятность для +1 и –1. При этом требуемое число бит равно 1 и компрессия невозможна. Если p = 0.1, $N_{бит}$ =0.47 и это значит, что возможна компрессия с коэффициентом 2.

Рис.28. Зависимость вероятности правильного и неправильного предсказания от величины ошибки предсказания z

Арифметическое кодирование является энтропийным, и именно оно используется в рассматриваемой системе, так в ней используется блоковая организация потока данных. На вход арифметического кодера подается сигнал ошибки предсказания в виде последовательности1 и 0. Первое означает, что предсказание неправильное ($e = 1$), второе, что предсказание правильное ($e = 0$). Из рис.28 следует, что с увеличением z от 0 вероятность правильного предсказания быстро увеличивается, а вероятность неправильного предсказания также быстро уменьшается, и при $z > 0,6$ она практически равна нулю. Это позволяет осуществлять эффективное сжатие цифрового потока.

5.4. Мультиплексирование и демультиплексирование звуковых каналов

По стандарту SACD кодирование с предсказанием в каждом из 6 звуковых каналов производится отдельно с помощью блоков Source model, а арифметический кодер (деко-

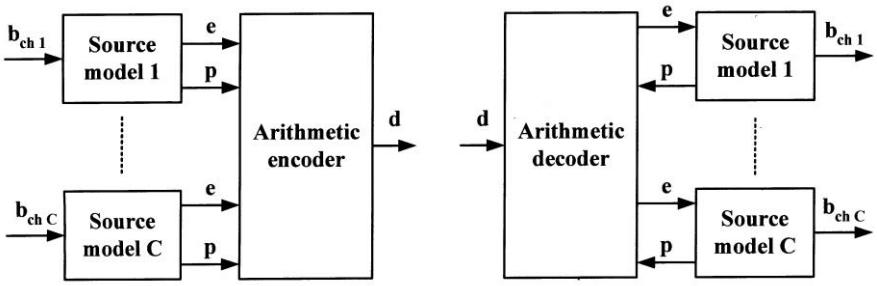

Fig. 7. Channel multiplexing in arithmetic encoder and demultiplexing in decoder.

Рис.29. Кодек арифметического кодирования

дер) один на все каналы (рис.29), и он выполняют функцию мультиплексирования. С каждого кодера с предсказанием на арифметический кодер подается сигнал ошибки предсказания e и вероятности двоичных символов p. На арифметический декодер вероятности символов подаются с декодеров с предсказанием.

Особый случай компрессии – это отсутствие сигнала (тишина). При этом все байты блока имеют одно и тоже значение или 4 бита равны 0 и 4 бита равны 1. С целью упрощения представим тишину в виде повторяющейся последовательности $+1, -1$. Такой сигнал может быть правильно предсказан фильтром первого порядка с коэффициентом $h(0) = -1$. В этом случае z принимает только два значения $+1, -1$. Так как все предсказания правильные цифровой поток e состоит из одних нулей, для арифметического кодера отсутствует информация, которую нужно кодировать, и поэтому коэффициент компрессии в этом случае очень большой

$$\eta = 75632 / 48 = 1568.$$

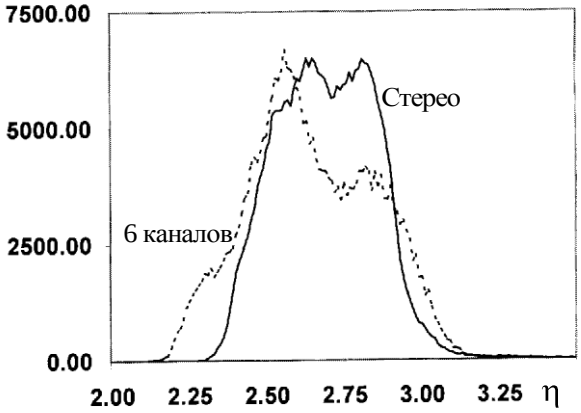

Рис.30. Функция распределения коэффициента компрессирования.

На реальной музыкальной программе коэффициент компрессии меняется от 2,5 до 3,5 (рис.30). При понижении уровня громкости компрессия увеличивается, а при увеличении, особенно на всплесках, резко уменьшается. Из графиков на рис видно, что коэффициент компрессии не бывает меньше 2,2 и больше 3,2. При записи классической музыки и джазовой коэффициент компрессии меняется от 2.636 до 2.826. Как видно из табл.. компрессия позволяет осуществлять 8 канальную запись с временем записи 74…78 мин.

Таблица 8. **Коэффициент компрессии**

	Nch	Kcomp	Тзап,мин
Классическая музыка	6+ 2	2.795 2.777	77 78
Джазовая музыка	6+ 2	2.728 2.626	75 74

6. Компрессирование без потерь многоканального цифрового потока

6.1.Система компрессирования MLP

Фирма Meridian разработала метод компрессирования многоканального цифрового потока без потерь под названием MLP (Meridian Lossless Packing), предназначенный для работы с сигналами в формате линейной ИКМ. Он используется при звукозаписи по стандартам DVD-Audio и CD . В отличие от известных систем перцептуального компрессирования и кодирования с потерями в системе MLP при компрессировании без потерь передаваемый сигнал никоим образом не изменяется, он просто более эффективно упаковывается в поток аудиоданных с меньшей скоростью.

При высококачественной записи звука по стандарту DVD-Audio используется 24-разрядная ИКМ с частотой дискретизации до 96 кГц и числом звуковых каналов до 8 (2 стерео + 6 каналов в формате 5.1). При этом скорость цифрового потока достигает 18,4 Мб/с, что очень сильно ограничивает время записи, которое при емкости диска 4.7 Гбайта не превышает 34 мин. Для увеличения времени записи в процессе кодирования производится компрессирование по системе MLP еще не мультиплексированных звуковых каналов.

Компрессия по этой системе может осуществляться при числе звуковых каналов от 2 до 63, причем, чем больше каналов, тем выше эффективность компрессирования. Частота дискретизации может быть до 192 кГц, а число разрядов - до 24, но при этом это скорость цифрового потока ограничивается значением 9,6 Мбит/с. Компрессирование эквивалентно уменьшению числа разрядов в слове, например, с 24 до 16. В системе MLP компрессированный сигнал может иметь как переменную, так и постоянную скорость цифрового потока.

С целью совместимости многоканальных звуковых систем, выполненных по разным стандартам, при кодировании в системе MLP используется иерархическая структура цифрового потока аудиоданных, содержащая множество субпотоков и иерархическую

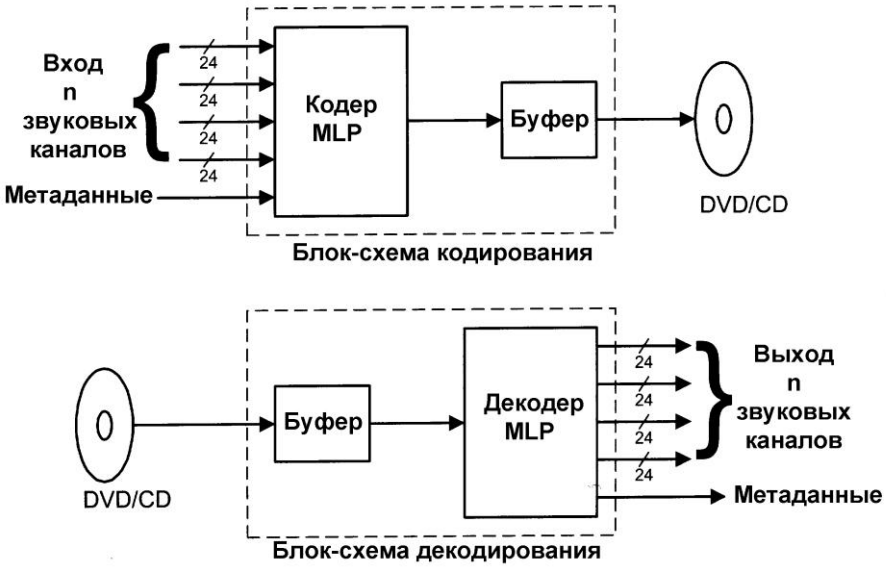

Рис.31. Схема многоканальной системы компрессирования MLP

структуру потоков дополнительных данных. При такой структуре декодер может использовать только часть потока для воспроизведения звука. Поэтому возможно при многоканальной записи воспроизводить ее в стерео варианте. На рис.31 поясняется структура кодека. В схему кодека входят многоканальные кодер, декодер и буферная память, как в кодере, так и в декодере с особой организацией обмена цифровыми данными.

В системе MLP компрессированный цифровой поток имеет переменную скорость при нормальном ЗС. На рис.32. иллюстрируется, что вариации скорости этого потока достигают почти 2 Мбит/с . Приведенный график соответствует следующим условиям испытаний: частота дискретизации 96 кГц, 24 разряда, 6 звуковых каналов с камерной музыкой

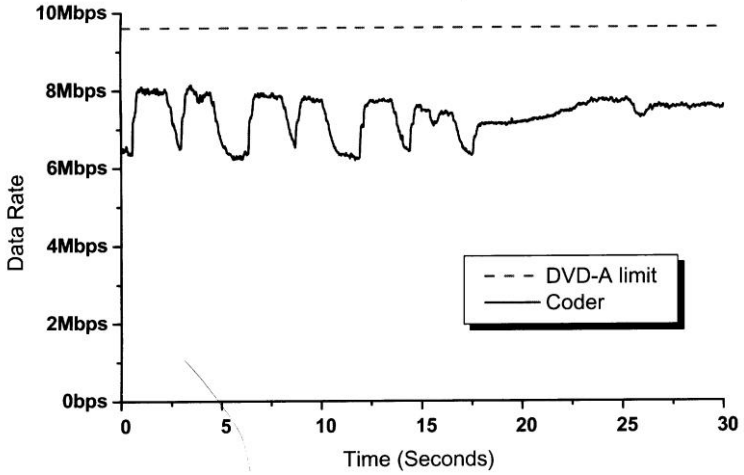

Рис.32. Скорость компрессированного цифрового потока в функции времени

барокко, исходная скорость цифрового потока 13,824 Мбит/с, длительность отрывка 30 с . Этот музыкальный пример показывает широкие вариации скорости компрессированного цифрового потока. Тишина и тихие пассажи компрессируются хорошо, а почти случайные сигналы – плохо. Теоретически случайные звуковые сигналы не компрессируются вообще. К счастью акустические сигналы не могут быть белым шумом во всех каналах одновременно в течение заметного интервала времени.

Известные системы компрессирования без потерь осуществляют оптимизацию по уменьшению средней скорости цифрового потока для уменьшения размеров компрессированного файла. MLP разработала метод компрессии, сложный по кодированию и простой по декодированию, который оптимизирован к высококачественному звуку с высокой частотой дискретизации, и при котором главным является уменьшение пиковой скорости цифрового потока. В основу технологии MLP положены следующие технические идеи:

- поиск каналов, передающих «воздух», в которых не используются все разряды кодовых слов;
- поиск каналов, передающих «воздух», в которых не используется весь возможный диапазон частот;
- удаление внутриканальной корреляции;
- эффективное кодирование ошибок предсказания ,
- сглаживание скорости цифрового потока с помощью буферной памяти.

6.2. Кодирование в системе MLP

Процессы кодирования и декодирования по системе MLP включает в себя следующие операции: компрессирование без потерь, матрицирование звуковых данных без потерь, использование фильтров IIR без потерь, использование буферной памяти, декодирование с самопроверкой и восстановление частот дискретизации.

На рис.33 приведена упрощенная схема многоканального кодера MLP. В этой схеме на вход подаются n 24-разрядных звуковых каналов. Звуковые каналы переупаковываются для оптимизации субпотоков. В каждом канале с помощью регистра осуществляется сдвиг цифровых данных для обнаружения не используемой емкости (такие, где задейст-

вовано меньше 24 разрядов, или не используется вся цифровая шкала). Матрицирование без потерь осуществляет понижение скорости цифрового потока благодаря устранению между канальной корреляции. Для устранения внутриканальной корреляции в каждом канале используется кодирование с предсказанием. При таком кодировании понижается скорость цифрового потока благодаря тому, что далее по каналу передается только де-

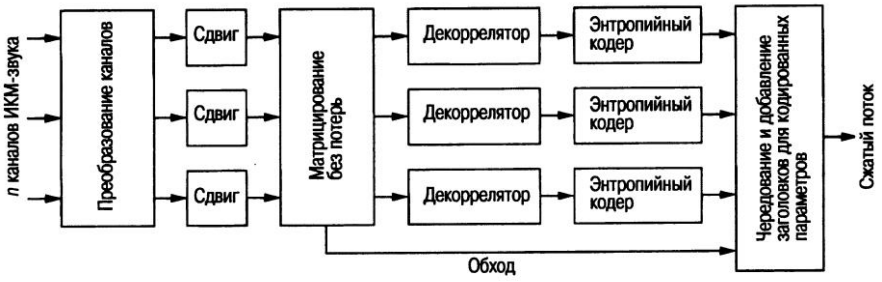

Рис.33. Схема многоканального кодера MLP

коррелированный сигнал ошибки предсказания. Дальнейшее понижение скорости цифрового потока достигается путем применению энтропийного кодирования сигнала ошибки предсказания. Скорость цифровых субпотоков в отдельных каналах сглаживается с помощью буфера FIFO. Затем осуществляется межканальное перемежение, каналы мультиплексируется и формируются блоки данных одного потока, содержащие и служебную информацию. Для каждого цифрового субпотока с помощью буфера емкостью 512 Кбайт формируется постоянная или переменная скорость, с которой осуществляется запись.

6.3. Матрицирование без потерь

В многоканальном аудио системе почти всегда существует информационная связь между каналами. Она слабая только в случае, когда при записи используемые микрофоны

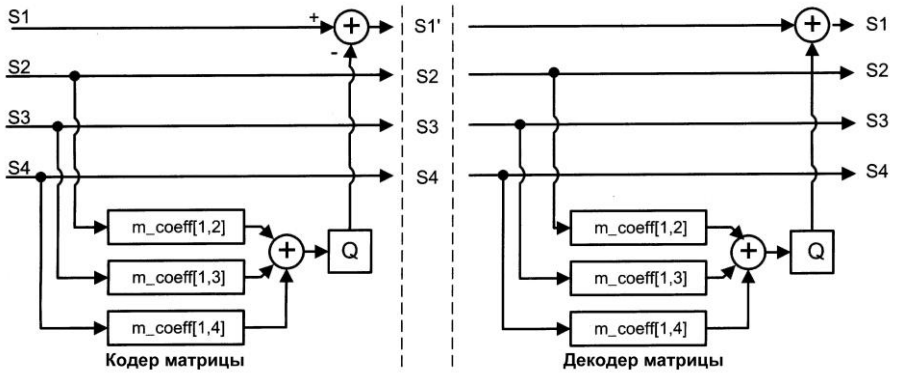

Рис.34. Схема аффиновских преобразований в кодеке MLP

разнесены на значительное расстояние. Сильная корреляционная связь между каналами возникает в следующих случаях:

- Моно представляется как двойное моно, с одинаковыми левым и правым каналами (аудио книга или архивная звукозапись),
- Объемное звучание создается на основе разницы между левым и правым каналами,
- Многоканальный громкоговоритель подключен к многоканальному выходу
- Многоканальный громкоговоритель подключен декодеру амбисоник с выхода в *B*- формате *WXYZ.*

Для устранения корреляционных связей между каналами, которые снижают возможность сокращения скорости цифрового потока, в MLP кодере используется канальная матрица без потерь. Одним из известных способов де-корреляции, является вращение в процессе матрицирования стерео микса из состояния левый/правый к состоянию сум-

ма/разность. Однако, такое матрицирование осуществляется с потерями, так как инверсная матрица реконструирует исходный сигнал, но без округления ошибок.

MLP кодер преобразует обычную матрицу в каскад афиновских преобразований. Каждое афиновское преобразование модифицирует отдельный звуковой канал путем суммирования квантованной линейной комбинации с другими каналами (рис.34). Если кодер выделяет конкретную линейную комбинацию, то декодер должен добавлять ее обратно. Реквантователь Q в этой схеме обеспечивает постоянство числа разрядов входных и выходных кодовых слов, что необходимо при различных вычислительных платформах, поступающей информации.

6.4. Кодирование с предсказанием

При таком кодировании значения будущих выборок предсказываются. Для декодирования необходимо, чтобы правила предсказания передавались в месте с разностным сигналом между фактическим и предсказанными значениями сигналов . Эту функцию выполняет декоррелятор, который так назван потому, что когда он оптимально адаптирован, отсутствует корреляция между передаваемым разностным сигналом и его предыдущем значением.

Полезно рассмотреть, как предсказатель действует (по Шенону) в частотной области. На рис.35. приведен спектр короткого музыкального теста. Если это спектр плоский, это значит, что линейный фильтр предсказателя осуществляет полную декорреляцию цифрового потока. Однако, как видно из рис.35, спектр далеко не плоский. Это значит, что необходимо повысить его вычислительную способность (усиление) декоррелятора, чтобы спектр стал более плоским.

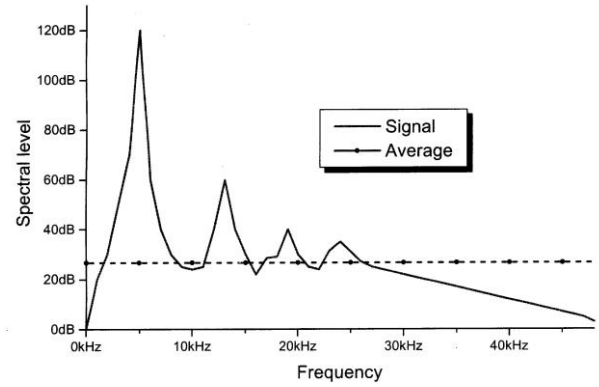

Рис.35. Спектр сигнала ошибки предсказания

В предельном случае полного отсутствия корреляции передаваемый разностный сигнал с плоским спектром является белым шумом. В соответствии с теоремой Gerson-Graven уровень оптимально декоррелированного сигнала определяется средним значением спектра исходного сигнала, представленного в децибелах как линейная функция частоты.. Как видно из рис.35 , это среднее значение в децибелах значительно меньше мощности исходного сигнала, следовательно, скорость цифрового потока уменьшилась.

Как доказывается Шеноном, это уменьшение мощности определяет передаваемую информацию. Степень, до которой музыкальный отрывок может быть декоррелирован («побелеть»), зависит от его содержания и сложности фильтра предсказателя. Теоретически при конечной сложности фильтра эта степень достигается предсказателем на энтропийном уровне. Однако, все коэффициенты, которые определяют этот декоррелятор, должны передаваться к декодеру, также как сигнал ошибки предсказания, для восстановления корреляции и реконструкции сигнала. Поэтому чем точнее работает предсказатель, тем большим числом коэффициентов он определяется и тем меньше возможно снижение скорости цифрового потока. На этом основании приходиться удовлетворяться некоторым балансом между сложностью предсказателя, качеством его работы и достижимой степенью компрессирования.

В большинстве известных схем компрессирования без потерь в качестве предсказателя используются FIR фильтры с конечной импульсной характеристикой, которые позволяют значительно уменьшить скорость цифрового потока на звуковом материале CD дисков. Однако, IIR фильтры, с бесконечной импульсной характеристикой , имеют преимущество, когда важно контролировать высокие скорости цифрового потока и когда у

спектра сигнала большой динамический диапазон.

Длина кодового слова на выходе цифрового фильтра всегда больше, чем на его входе в связи с выполнением операций умножения, что увеличивает скорость цифрового потока. Поэтому в схемах предсказателей на выходах фильтров включаются реквантователи (рис.36), с помощью которых достигается равенство длин слов на входе и выходе фильтра.

Эта операция легко реализуется с FIR фильтрами, в которых вычисления осуществ-

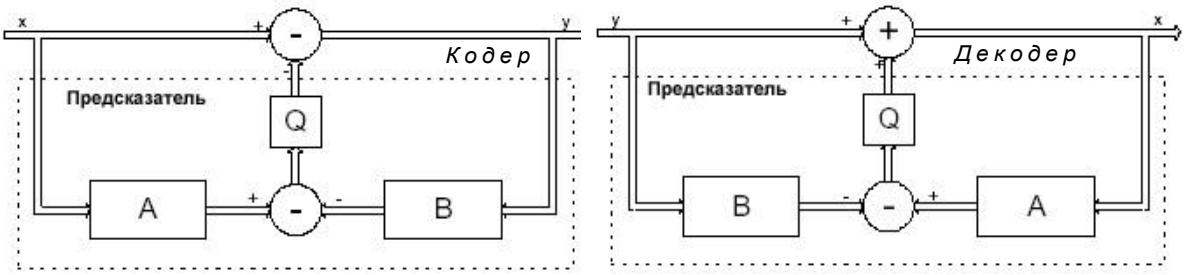

Рис.36. Кодирование/декодирование с предсказанием на основе IIR фильтров

ляются на основе арифметики со словами конечной длины. С IIR-фильтрами сложнее, так как у них другая вычислительная платформа расчета коэффициентов фильтра. Одним словом эти фильтры не стыкуются в одной схеме. Поэтому в MLP используется предсказатель с двумя FIR фильтрами, которые включены в прямую и обратную ветви схемы предсказателя (рис.36). Он осуществляет предсказание как «вперед», так и «назад». Такой комбинированный предсказатель имеет характеристику такую же, как у IIR фильтра. Если фильтр B выключается, то предсказатель имеет характеристику FIR фильтра.

Кодер MLP использует предсказатель в каждом канале, и он может выбирать FIR или IIR фильтры до 8 порядка. В формате DVD-Audio запись производится без компрессирования при скорости цифрового потока до 9,6 Мбит/с. Максимальная скорость потока данных достигает 13,824 Мбит/с. При использовании IIR фильтров коэффициент понижения скорости варьируется в пределах от 1, 64 до 2,08, что вполне остаточно.

6.5. Применение буферной памяти

Пока звуковой сигнал нормальный, кодирование с предсказанием работает хорошо. Однако, на звуковых фрагментах с сибилянтами, синтезированным шумом или перкусси, имеющих большую энтропию, такое кодирование работает плохо. Поэтому в кодере MLP используется специальная форма применения буферной памяти для непрерывного циф-

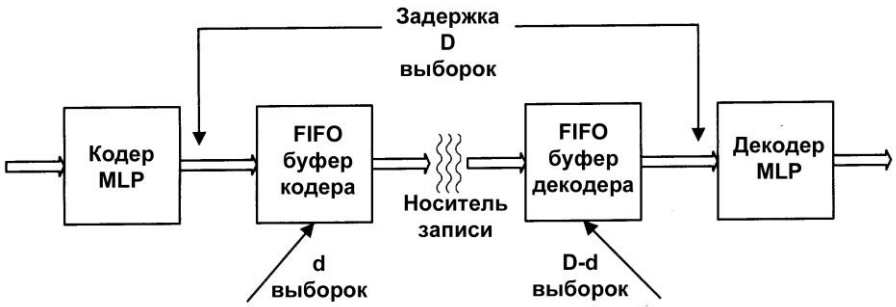

Рис.37. Система буферной памяти в кодеке MLP

рового потока, при которой уменьшаются вариации скорости передаваемого потока данных, и при этом поглощаются переходные процессы, которые трудно компрессировать.

В кодере и декодере используется FIFO буферная память (рис.37). Эти буферы так сконфигурированы, что между выходом кодера и входом декодера существует постоянная задержка 75 мс. Задержка в буфере кодера составляет d выборок (переменная вели-

чина) , а в буфере декодера она равна $D-d$. Для быстрого старта система управления буфером FIFO минимизирует задержку в декодере, поэтому его буфер в нормальных условиях почти пустой и наполняется, когда декодер предвидит высокое кратковремен-

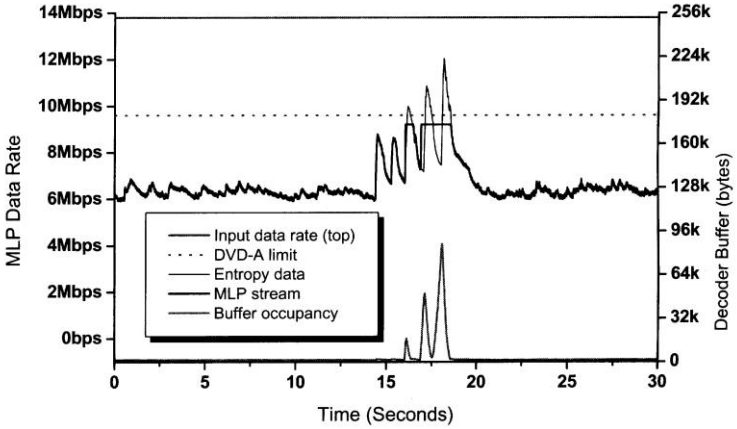

Рис.38. Компрессия скорости цифрового потока с помощью буферной памяти

ное повышение скорости цифрового потока. В этом случае буфер декодера опустошается, и он в состоянии осуществлять передачу данных со скоростью больше, чем может осуществляться воспроизведение записи.

На рис.38 иллюстрируется работа буфера, когда сигнал не поддается компрессии с помощью энтропийного кодирования и сокращение мгновенных значений скорости цифрового потока осуществляется буфером. С его помощью скорость со 12,03 Мбит/с понижается до допустимого значения 9,2 Мбит/с.

6.6. Кодирование/декодирование субпотоков

Входные звуковые каналы при кодировании сначала матрицируются в два и более

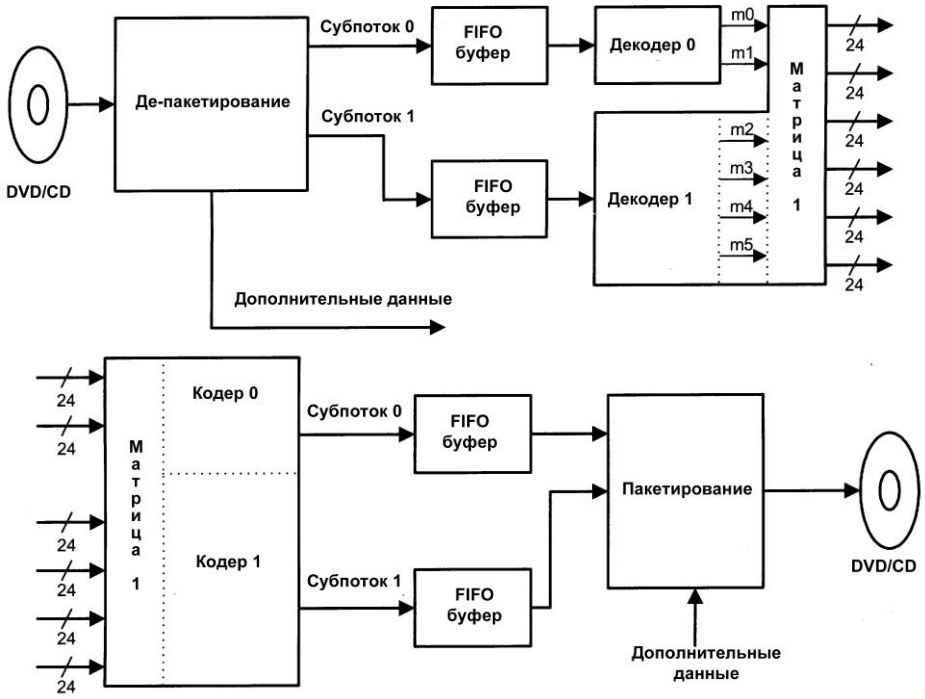

Рис.39. Функциональная схема кодека MLP

субпотоков (рис.39) и подаются на многоканальный кодер MLP, состоящий из нескольких параллельно работающих одноканальных кодеров субпотоков. В каждом субпотоке используется собственный буфер FIFO. Выходные цифровые субпотоки буферов муль-

типлицируются и одновременно организуются в блоки данных, называемые пакетами, которые и поступают на запись.

В декодере все операции осуществляются в инверсном порядке. Сначала производится де-пакетирование и выделяются субпотоки данных. В каждом субпотоке с помощью управляемых буферов производится задержка сигналов и декодирование данных. В последнюю очередь осуществляется дематрицирование, при котором восстанавливаются исходные звуковые каналы и реконструируются звуковые сигналы. Функциональная схема субканального блока декодеров MLP приведена на рис.40. В этой схеме все операции выполняются инверсно по отношению к блоку кодеров MLP на рис.33 .

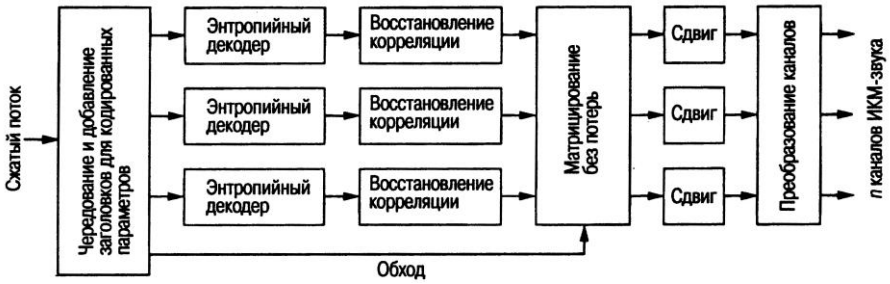

Рис.40. Многоканальный декодер MLP

Кодированный цифровой поток несет всю информацию, необходимую для декодирования: инструкции для декодера, компрессированные данные, проверочные данные кода *CRC*, обнаруживающего ошибки, и тестовую информацию для проверки компрессирования. Выходные данные кодера организуются в блоки длиной по 40…160 слов. Блоки собираются в пакеты длиной по 640…2560 выборок и длительностью 7 мс. При такой организации данных ошибки не распространяются за пределы пакета. Обычно одиночные ошибки возникают не чаще, чем через 1.6 мс.

В табл.9. приведены данные по уменьшению скорости цифрового потока в стереоканале при компрессировании, обусловленной уменьшением числа бит в кодовых словах, для пиковых и средних значений. Обычно компрессия возрастает с увеличением числа каналов, если имеется корреляция между каналами и с уменьшением уровня шума. При частоте дискретизации 44.1 и 48 кГц пиковые значения скорости цифрового потока уменьшаются на 4 бита в выборке, и 16-разрядный поток преобразуется в 12-разрядный. При частоте дискретизации 96 кГц пиковая скорость уменьшается на 8 бит в выборке и 24-разрядный канал компрессируется до 16-разрядного. В 24-разрядном канале уменьшение числа бит в выборке на 11 соответствует коэффициенту компрессии 1.85. Среднее значение коэффициента компрессии по всем музыкальным программ и форматам записи считается равным 2. Фактически он достигает значения до 2.5.

Таблица 9. **Компрессия потока**

$f_д$, кГц	Уменьшение числа бит в слове	
	Пиковое	Среднее
48	4	5…11
96	8	9…13
192	9	9…14

В системе *DVD-A* максимально допустимая скорость цифрового потока при записи равна 9,6 Мб/с. При этой скорости время записи составляет 65 мин. Скорость цифрового потока при 6-канальной записи звука в формате 96/24 составляет 13.824 Мб/с, что значительно выше допустимой. Поэтому запись таких программ возможна только с компрессированием. В табл.10. для системы *DVD-A* приведено время записи звука в различных форматах с использованием компрессирования.

Таблица 10. **Время записи**

Число каналов	fд, кГц	q, бит	Время, час
5.1	96	24	1.7
6	96	24	1.4
2	96	24	4
2	192	24	2
2	44.1	16	12
1	44.1	16	25

Список литературы

1. *Lajos Hanzo, F. Clare Somerville, Dr Jason Woodard*. Voice and Audio Compression for Wireless Communications (Second Edition). John Wiley & Sons, Ltd. 2007,-834.
2. *Day Tracy Yang, Chris Kyriakakis, and C-C. Jay Kuo*. High-Fidelity Multichannel Audio Coding. Hindawi . Publishing Corporation, 2005,-228.
3. *Adistambha Kevin, Burnet Lan*, … A codebook-based Cascade Coder for Embedded Audio Coding. AES E-Library, Paper N 6806. May 2006.-89.
4. *Markus Kuhn*. Information Theory and Coding, Image, video and Audio compresson. Michaelmas 2003
5. *Вологдин Э.И.* Методы и алгоритмы обработки звуковых сигналов. Учебное пособие, часть 2. СПб, ГУТ 2009, -96.
6. *Gerzon M.A., Craven P.G., Stuart J.R., Law M.J., Wilson R.J.* The MLP Lossless Compression System for PCM Audio//Journal of the Audio Engineering Society.-2004, vol.52, Number 3.s, 243-260.
7. *Knaren E., Reefman D., Janssen E., Bruekers F*. Lossless Compression of 1-Bit Audio//Journal of the Audio Engineering Society.-2004, vol.52, Number 3.s. 190-199.
8. J.R Stuart, P.G.Craven,…..MLP Lossless Compresson. AES 9th Regional Convention Tokyo. AES 17th International conference on High Quality Audio Coding 15
9. Craven, P.G. and Gerzon, M.A., ‘Lossless Coding for Audio Discs’, J. Audio Eng. Soc., vol.44 no. 9 pp. 706–720 (September 1996)
10. Craven, P.G., Law, M.J. and Stuart, J.R.,‘Lossless Compression using IIR Prediction Filters’, Preprint #4415, March 1997