

Преподу: Декодер не описан в должной мере. В конце пара личных ласкутков, которые потом пойдут в нужные файлы и места.

После завершения дизассемблирования, был проведен полный анализ алгоритмов кодирования и декодирования. В результате было выяснено, что кодек состоит из нескольких основных модулей: свёрточный кодер, модуль прямого чередования (interleaving), модуль обратного чередования (deinterleaving), декодер- Витерби.

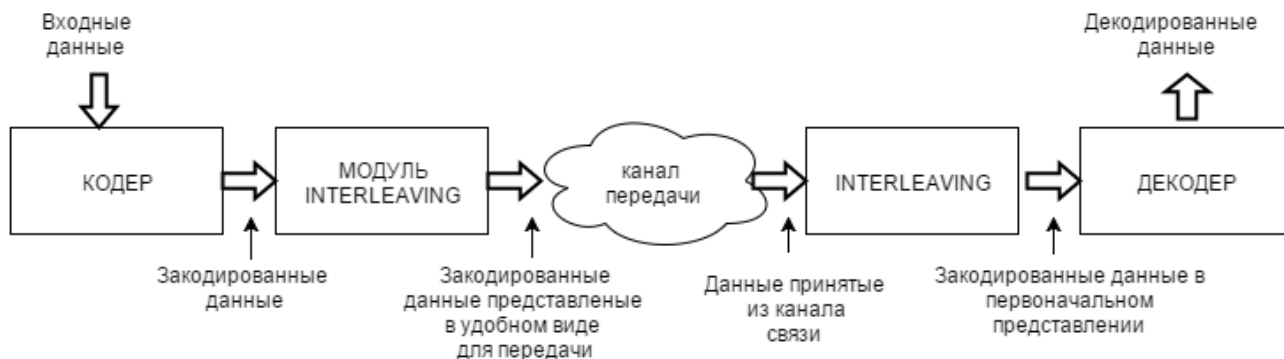


Рис.1. Структурная схема работы кодека.

Структура кодека приведена на рисунке 1. Ниже рассмотрим структура каждого из модулей.

Кодер

Кодер выполняет кодирования указанного участка памяти в одном из двух режимов: кодирование двумя битами или четырьмя. При этом каждый кодируемый бит будет представлен по завершению двумя или четырьмя битами соответственно. В основе данного модуля лежит 4 полинома. Для режима кодирования двумя битами используется только два полина, для кодирования четырьмя битами – все четыре. Полиномы зависят от предыдущих шести бит. Это говорит о том, что кодер с памятью. Для снижения временных затрат при кодировании, значения полиномов для различных состояний просчитаны заранее и хранятся в виде таблицы. Результат кодирования обратно записывается в память.

Модули чередования

После того, как данные успешно закодированы, они подвергаются чередованию. Это необходимо для снижения вероятности неудачного декодирования. Дело в том, что во время передачи данных, вследствие воздействия шумов происходят битовые ошибки, часть данных теряется или принимается неверно. Декодер способен восстановить часть потерянных данных за счет свойств кода, однако качество декодирования и распознавания ошибок, напрямую зависит от количества ошибочных бит, следующих подряд друг за другом (пачки ошибок). В канале же при возникновении зашумления – портится целая последовательность бит, длиной от нескольких десятков до нескольких сотен бит. Для того, чтобы минимизировать длину последовательностей, ошибочных бит, применяются прямая и обратная операции чередования. Модуль interleaving перемешивает биты в соответствии со стандартным алгоритмом. Затем данные поступают в канал передачи, откуда считываются другим устройством. На принятом устройстве закодированные данные поступают в модуль обратного чередования, который восстанавливает исходную последовательность бит, которые в свою очередь уже могут быть декодированы.

Алгоритмы чередования не представляют большого интереса, поэтому подробно не приведены. Можно только уточнить, что основной задачей чередования является максимальный разброс соседних бит по посылке.

Декодер

На последнем этапе закодированные данные поступают в декодер, где по алгоритму Витерби производится декодирование. Стоит отметить, что данная реализация декодера-Витерби является реализацией с мягкими весами, т.е. на вход Витерби поступают биты с весами. В данном случае вес – это вероятностная мера, утверждающая с некоторой точностью, что принятый бит - это ноль или

единица. Тело алгоритма состоит из двух основных частей: прямой и обратной проходки. При прямой записываются метрики возможных путей декодирования (по решетке Витерби). В случае же, если на некотором этапе цикла, истинный путь окажется хуже 64-ех других путей – он будет отброшен, что приведет к ошибке. Конечно, теоретически можно было бы записывать все пути, однако это вызвало бы огромные временные и ресурсные затраты. Отслеживание именно 64-ех путей, во-первых, кратностью степени двойки, а во-вторых отслеживание 64-ех путей дает необходимый уровень надежности, но при этом более экономично нежели 128 путей.

при этом происходит частичная или полная коррекция ошибок. Успешность коррекции определяется следующим: статистическими свойствами канала передачи данных и корректирующей способностью алгоритма Витерби. В данном случае глубина декодирования равна 6, то есть алгоритм просчитывает сразу 64 возможных исхода, из которых выбирает лучший.

Отдельный кусок о чередовании (потом его в нужное место затолкаю, где-нибудь в середину, где о кодере рассказывать буду)

Чередование передаваемых данных – это специальный прием при передаче данных в канале. При перемешивании данных в строго определенном порядке (и последующей ликвидации чередования), добивается эффект дробления ошибок, возникших в канале при передаче. Дело в том, что при возникновении ошибки, портится сразу несколько подряд идущих бит (от ХЗ до бесконечности). Вероятность успешного декодирования последовательности, в которой ошибочные биты идут подряд друг за другом близка к нулю. Поэтому необходимо применять чередование.

Отдельный кусок по Витерби-реализации

Так же стоит отметить, что данная реализация декодера-Витерби является реализацией с мягкими весами, т.е. на вход Витерби должны поступать биты с весами. В данном случае вес – это вероятностная мера, утверждающая с некоторой точностью, что принятый бит - это ноль или единица.

PAUSE