

АППАРАТНАЯ РЕАЛИЗАЦИЯ ДЕКОДЕРА ВИТЕРБИ

Стешенко В.Б., Петров А.В.

МГТУ им. Н.Э.Баумана

105005, Москва, 2-я Бауманская ул., д.5, тел. (+7-095) 263 67 36, E-mail: steshenk@sm.bmstu.ru

Реферат: В работе рассматриваются особенности аппаратной реализации декодеров Витерби с учетом современной технологии и методов проектирования.

Введение.

Кодирование с исправлением ошибок применяется во многих цифровых системах связи. При этом используются многие методы кодирования и декодирования, в том числе сверточное кодирование и декодирование Витерби. Сверточное кодирование, применяемое совместно с декодированием Витерби, стало в настоящее время одним из наиболее известных методов исправления ошибок. Причина этого состоит как в простоте реализации, так и в относительно большом достигаемом при этом выигрыше от кодирования. Этот сравнительно большой выигрыш объясняется тем, что алгоритм можно строить в расчете на использование демодулятора с мягким решением, увеличивающим выигрыш примерно на 2 дБ по сравнению с демодулятором с жестким решением [1].

Несмотря на обилие публикаций, в них обычно не освещаются практические аспекты аппаратной реализации рассматриваемых алгоритмов декодирования.

Несмотря на кажущуюся простоту алгоритма декодирования, существуют ряд проблем при реализации алгоритма для задач цифрового телевидения. Так как необходимо обеспечить высокую помехоустойчивость, высокую скорость передачи данных на различных скоростях, необходимо обеспечить высокий параллелизм вычислений при достаточно большой глубине декодирования. В этом состоит сложность реализации. Более того, был проведен анализ использования программной реализации алгоритма. Анализ показал, что быстродействие в таком случае не будет требуемым. Поэтому реализацию данного алгоритма для задач цифрового телевидения необходимо производить аппаратно.

Цель работы – показать основные особенности аппаратной реализации декодера Витерби для задач цифрового телевидения, определяемого европейским стандартом EN 300 744.

1. Внутренний кодер.

Внутренний кодер, который помещается в передатчике состоит из сверточного кодера (Convolutional Encoder) и блока “Puncturing”[2]. Сверточный кодер состоит из сдвиговых регистров и сумматоров по модулю два. Таким образом из одного входного потока данных получается два – X и Y. Блок “Puncturing” преобразует эти два потока в один, “выкалывая” при этом 1 бит (X или Y). Таким образом закодированная последовательность может передаваться на разных скоростях. Таблица 1 показывает связь скорости передачи с передаваемым потоком данных [2].

Обычно работу кодера описывают с помощью графа. Состояния графа соответствуют выходам регистров, значение бита над стрелкой показывает входной бит, при котором граф переходит в следующее состояние, а значения двух бит внутри стрелки показывают X и Y которые образуются при переходе.

Таблица 1.

Скорость	Передаваемая последовательность данных
1/2	X1 Y1
2/3	X1 Y1 Y2
3/4	X1 Y1 Y2 X3
5/6	X1 Y1 Y2 X3 Y4 Y5
7/8	X1 Y1 Y2 Y3 Y4 X5 Y6 X7

2. Аппаратная реализация алгоритма декодирования Витерби.

Другой способ представления связи между входной и выходными последовательности (аналогичный графу) называется решетчатой структурой. Входной символ 0 соответствует выбору верхнего ребра, а 1 – нижнего, каждому состоянию соответствует горизонтальный ряд вершин, выходная пара X и Y показана около ребер. Всякая входная последовательность соответствует определенному пути на решетке.

Задачу декодирования сверточного кода можно рассматривать как задачу нахождения пути по решетчатой диаграмме с помощью некоторых правил декодирования. Необходимо выбрать путь, который лучше всего согласуется с принятой последовательностью. Для этого на каждом шаге поступления входных данных необходимо вычислять метрики для каждого пути на решетке [1].

Для вычисления метрик, накопления путей служит базовый решетчатый элемент – трелис (trellis), показанный на рис. 1. Он состоит из двух частей: вычислителя метрик (рис. 5а и 5б) и накопителя пути. Декодер содержит всего 32 трелиса. Каждый трелис переходит из состояния Cond0 в состояние Cond0' или в Cond1', и из состояния Cond1 в состояние Cond0' или в Cond1', при этом вычисляются метрики. Количество состояний всех трелисов соответствует количеству состояний декодера и кодера и равно 64.

Состояние Cond0 принимает четные значения от 0 до 62 для трелисов с номерами от 1 до 32 соответственно, Состояния Cond1, Cond0' и Cond1' вычисляются для каждого трелиса по формулам приведенным на рисунке 5а.

X0, Y0 – константы, при которых состояния трелиса переходят из Cond0 в Cond0' и из Cond1 в Cond1'. X1, Y1 – константы, при которых состояния трелиса переходят из Cond0 в Cond1' и из Cond1 в Cond0'. Значения X0, Y0 (X1, Y1) равняются значениям выходов X и Y кодера при переходе его из одного состояния в другое, соответствующем переходу трелиса в случае жесткого декодирования. В случае мягкого декодирования значения констант равное 0 заменяется на 1100 в дополнительном коде (при 3-битном решении) и на 1000 в дополнительном коде (при 4-битном решении), а значения констант равное 1 заменяется на 0011 в прямом коде (при 3-битном решении) и на 0111 в прямом коде (при 4-битном). С целью универсальности применения декодера как в случае жесткого, так и мягкого декодирования целесообразно сделать значения констант 4-х битными.

На рис. 1б показана реализация вычислителя метрик. M_k^0, M_k^1 - регистры входных метрик, M_{k+1}^0, M_{k+1}^1 - регистры выходных метрик. $\Delta M_0, \Delta M_1$ - вычислители расстояния между константами X0, Y0, X1, Y1 и входными данными X и Y. Сумматоры вычисляют сумму входных метрик с расстоянием, причем сумматоры содержат ограничение по величине. Блоки min выбирают значение минимальных метрик, которые затем записываются в выходные регистры. Кроме того, блоки min выдают признаки а и б для блока накопителя пути, значения которых означают, какая из метрик минимальная.

На рис. 5с представлен накопитель пути. W_k^0, W_k^1 - регистры входных путей, W_{k+1}^0, W_{k+1}^1 - регистры выходных путей. Разрядность регистров соответствует необходимой глубине декодирования. На каждом такте поступления данных старшие разряды входных путей поступают на выходы dropped_bit0 и dropped_bit1, мультиплексорами выбираются остальные разряды входных метрик в зависимости от признаков а и б, получившиеся значения путей поступают в выходные регистры W_{k+1}^0, W_{k+1}^1 , в старшие разряды которых записываются 0 и 1 соответственно.

На рис. 2 показана структурная схема реализованного декодера Витерби. Она состоит из депунктурера, блока трелисов, включающего в себя 32 трелиса, блока минимизации.

Депунктурер преобразует последовательный поток данных в параллельный, в зависимости от скорости передачи. Вместо пропущенных бит депунктурер выдает нулевые значения X или Y. Также депунктурер выдает сигналы set_zero_X0, set_zero_Y0, нулевые значения которых показывают какой бит пропущен X или Y, и в соответствии с этим, блок вычисления констант обнуляет константы X0, X1 или Y0, Y1. Входными сигналами его являются последовательный поток данных, скорость (rate). Выходные данные депунктурера поступают в блок трелисов, где они поступают на все трелисы одновременно. Блок констант выдает константы на все трелисы одновременно, в зависимости от режима декодирования (мягкое или жесткое) и разрядности при мягком декодировании.

Сигнал soft, поступающий на блок трелисов, является режимом декодирования, а сигнал decision – определяющим разрядность при мягком декодировании. Сигнал decision игнорируется блоком трелисов при жестком декодировании. В блоке трелисов, трелисы соединяются следующим образом. Соединяются метрики трелисов, соответствующие одинаковым состояниям разных трелисов, также соединяются пути трелисов. Выходными сигналами блока трелисов являются метрики всех трелисов и соответствующие им сигналы dropped_bit0 и dropped_bit1, которые поступают на блок минимизации. Блок минимизации сравнивает все пришедшие метрики и выдает на выход один из всех сигналов dropped_bit0 и dropped_bit1, соответствующий минимальной метрике. Этот сигнал и будет декодированным данным.

3. Алгоритм работы декодера Витерби.

1. Установить требуемую глубину декодирования $L \geq 5k$, где k - длина кодового ограничения. Старт осуществляется из состояния 0 по решетке.
2. Перейти на следующий слой решетки, обновив метрики путей. В начальный момент эта операция повторяется для для каждого слоя L раз. Вычисленный код каждого пути заносится в память кодов путей.
3. На L -ом участке производится сравнение метрик путей, выдается код, соответствующий минимальной метрике.
4. Осуществляется прием нового ребра и переход на один слой решетки вперед. Сдвиг памяти кодов происходит на одну ячейку.
5. Вычислить метрики путей.
Выдать код, соответствующий минимальной метрике., перейти к п.4.

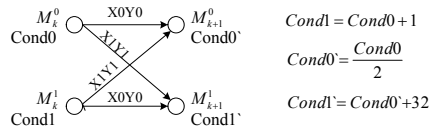


Рис. 1, а

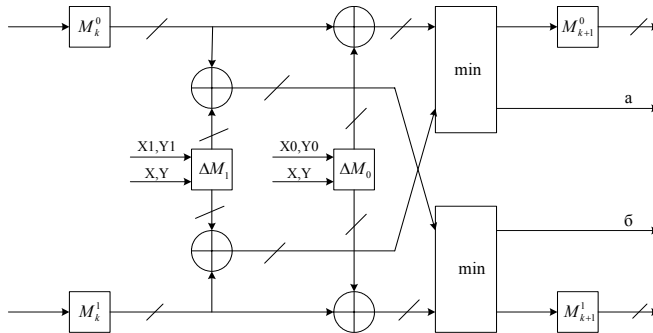


Рис. 1, б

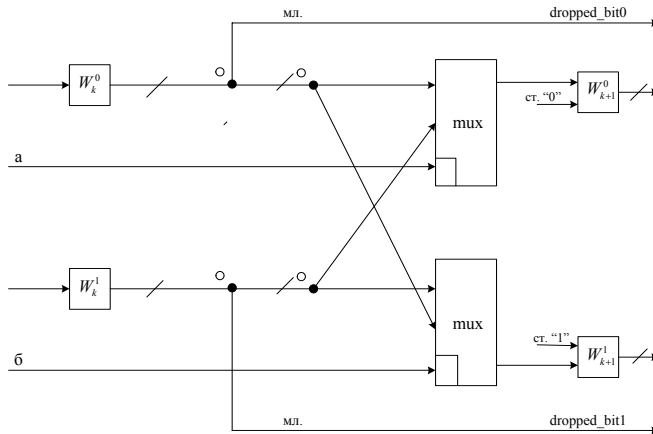


Рис. 1, с

Рис. 1. Трелис.

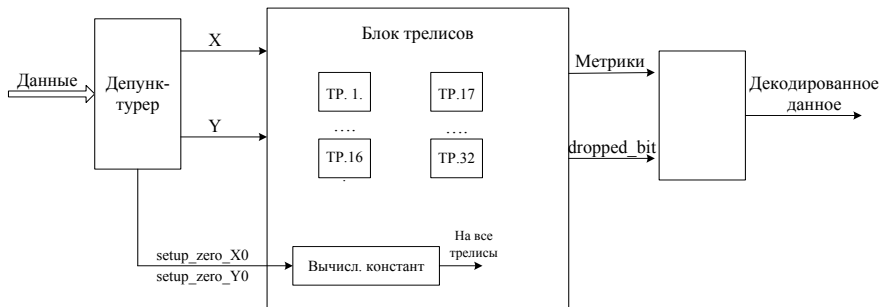


Рис. 2. Структурная схема декодера Витерби.

4. Выводы.

Алгоритм декодирования Витерби реализован на заказной СБИС (ASIC) на технологии 0.25 микрон. По отношению к имеющимся реализациям (по зарубежным статьям и микросхемам-аналогам) данная реализация обладает высоким быстродействием (вычисления производятся за один такт, при тактовой частоте 40 МГц), высокой глубиной декодирования, равной 128 (например, микросхема MC92314 имеет глубину декодирования равную 96), что обеспечивает помехоустойчивость выше, чем у зарубежных микросхем-аналогов при передаче данных на высоких скоростях.

Список литературы

1. Дж. Кларк, мл., Дж. Кейн. Кодирование с исправлением ошибок в системах цифровой связи. М., Радио и связь, 1987.

2. Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television. European Standard (Telecommunications series) EN 300 744 V1.1.2 (1997-08).