РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я
№ 6 1995

УДК 519.725

О. В. Мазуров
(Новосибирск)

ДЕКОДИРОВАНИЕ КОДА РИДА — СОЛОМОНА ДЛЯ ВЕКТОРОВ В СИСТЕМАХ ХРАНЕНИЯ ИНФОРМАЦИИ

Рассматривается применение в системах хранения информации дополнительно к аппаратным средствам защиты от потерь информации более мощных методов помехоустойчивого кодирования, реализуемых на программном уровне. Блоки информации представляются в виде векторов элементов поля $GF(2^q)$, для которых осуществляется покомпонентное кодирование кодом Рида — Соломона. Дается алгоритм декодирования, использующий возможное совпадение позиций ошибок в соседних компонентах. При этом оказывается возможным восстановление ошибок, число которых превышает половину кодового расстояния по каждой компоненте.

Применение помехоустойчивых кодов в технических системах хранения информации имеет давнюю и плодотворную практику. Устройства, аппаратно реализующие алгоритмы декодирования таких кодов, позволяют с высокой скоростью обнаруживать и исправлять ошибки, возникающие в носителях, делая поток считываемой информации практически непрерывным. Ставшая массовой технология записи неизменяемой информации на CD-ROM своим успехом, в частности, обязана применению при записи современных методов помехоустойчивого кодирования. Это позволило при высокой плотности записи информации иметь приемлемую надежность ее сохранности (по прогнозам — десятки или даже сотни лет). Для информации, тиражируемой в сотнях тысяч экземпляров, такие прогнозы более чем удовлетворительны. Однако стремительное удешевление устройств записи на CD-ROM индивидуального пользования, так называемых CD-R, приводит к тому, что на компакт-диски записывается все больше и больше уникальной информации. Ценность же такой информации зачастую оказывается столь высокой, что требуются дополнительные усилия по обеспечению сохранности записанной информации.

Применение более мощных методов помехоустойчивого кодирования приходит в противоречие со сложностью декодирующего устройства и скоростью, с которой такое декодирование должно производиться. Однако именно для архивной информации вполне приемлемой может оказаться ситуация, когда кодирование, которое осуществляется один раз, может продолжаться минуты и даже часы, а мощное декодирование, которое включается только тогда, когда аппаратное восстановление не работает, — часы и даже сутки.

Таким образом, речь идет о применении вместе с существующими аппаратными схемами защиты новых методов, реализуемых уже на программном уровне, с целью значительного снижения вероятности потери записанной информации. В данной статье рассматривается простой метод дополнительного кодирования и предлагается алгоритм декодирования, который при определенном характере ошибок имеет большую восстановительную способность, чем традиционные алгоритмы. Метод имеет общий характер и поэтому применим не только, а подчас и не столько к CD-ROM, но и к обычным магнитным носителям.

Трудность применения в устройствах чтения информации более мощных кодов на программном уровне часто заключается в том, что аппаратный контроль такого устройства при определенном количестве ошибок, превышающем его восстановительные способности, считает всю информацию разрушенной, и при этом нет возможности считывать эту информацию такой, какая она есть.

Наиболее простой способ обойти эту трудность представляет собой работа не с битами или байтами, а сразу с целыми блоками, на уровне которых устройство обеспечивает контроль целостности, в качестве кодируемых символов. Вместе с системой контроля нижнего уровня такую схему можно рассматривать как каскадное кодирование. При этом система контроля, реализованная в устройстве, играет роль внутреннего кода, а в качестве внешнего кода естественно выбрать код с максимально достижимым расстоянием, например код Рида — Соломона.

Простейшая схема декодирования такого кода заключается в том, что все блоки, которые не смогли быть прочитаны устройством, рассматриваются как стертые символы во внешнем коде, и если их число не превышает числа избыточных блоков, то они могут быть восстановлены с помощью только алгоритма исправления стираний.

Представление блоков информации в качестве кодируемых символов может осуществляться несколькими способами, а именно, в виде векторов элементов конечного поля $GF(2^q)$. При этом q может быть выбрано из соображений простоты реализации арифметики в данном поле и мощности поля, поскольку количество кодируемых символов (блоков) вместе с избыточными не должно превышать 2^q. Как правило, длина блока является степенью числа 2, поэтому в качестве q можно выбирать значения 8, 16, 32, имея запас длины кодового слова соответственно 2^8, 2^{16}, 2^{32}. Если длина блока не делится на q, то такой блок можно дополнить фиктивными нулевыми битами до нужной длины. В случае блоков переменной длины также можно все блоки привести к максимальной длине, дополняя их нужным количеством нулевых битов (байтов).

Кодирование векторов осуществляется покомпонентно, т. е. i-е координаты упорядоченной совокупности всех векторов представляют собой i-е кодовое слово, которое может кодироваться и декодироваться независимо. Проверочные равенства (n, k, d)-кода Рида — Соломона для векторов, следуя [1], запишем в виде

$$\sum_{i=1}^{n-1} \overline{X}_i Z_i^l = 0, \qquad l = 0, \ldots, d-2, \quad d = n-k+1, \quad k < n \le 2^q. \tag{1}$$

Очевидно, что декодирующая способность такого кода больше, чем просто восстановление $d-1$ стертых векторов. Однако для случая, которому соответствует приведенная выше схема декодирования, реализовать эту способность оказывается трудно. Если же мы имеем дело с ошибками, то оказывается, что декодирующая способность этого кода может быть существенно улучшена даже по сравнению с простым покомпонентным декодированием.

Ошибочным следует считать такое состояние блока, при котором внутренний код не диагностирует ошибки, в то время как синдром внешнего кода оказывается отличным от нуля. Такие ошибки могут возникать в результате записи информации в блок без соответствующей модификации избыточных блоков вследствие как сбоев в системах контроля и поддержки целостности закодированной информации, так и несанкционированного доступа.

Другой тип ошибок может рассматриваться в случае, когда нам удается считать информацию из сбойных блоков как есть. При этом в считанной информации могут присутствовать одиночные ошибки, пакеты ошибок и сбои синхронизации (вставки и потери битов). Когда число сбойных блоков оказывается больше числа избыточных, указанный выше алгоритм исправления стираний не работает, однако остается возможность восстановления, поскольку ошибочными могут оказаться не все, а только некоторые компоненты век-

торов. Традиционный алгоритм декодирования ошибок в такой ситуации способен восстановить только до $[(d-1)/2]$ ошибок по каждой компоненте.

Если допустить, что распределение ошибок носит не абсолютно случайный характер, а имеют место серии ошибок, как в случае модификации записанной информации, пакетов ошибок и сбоев синхронизации, то можно предложить следующий алгоритм декодирования.

Обозначим компоненты векторов синдромов для содержащих ошибки векторов $\overline{X}_i'$:

$$D_m(Z) = \sum_{i=0}^{l_m} d_{mi} Z^i = \prod_{i=1}^{l_m} (Z - Z_{\alpha_{mi}}), \qquad l_m < d-1, \tag{3}$$

где l_m — количество ошибок, а α_{mi} — их позиции. Тогда имеют место соотношения

$$\sum_{i=0}^{l_m} d_{mi} S_{m, i+j} = 0, \qquad j = 0, \ldots, d-2-l_m. \tag{4}$$

При $l_m \le [(d-1)/2]$ имеем $d-1-l_m \ge l_m$ линейных уравнений с l_m неизвестными, которые могут быть решены в рамках традиционного алгоритма декодирования кода Рида — Соломона (см., например, [2]). Этот алгоритм обозначим A_0. Если число ошибок оказалось больше $[(d-1)/2]$, то этот алгоритм, как правило, выдает отказ от декодирования, хотя существует вероятность ошибочного декодирования.

В случае отказа рассмотрим две соседние компоненты m и $m+1$. Пусть имеем для этих двух компонент $l_{m, m+1}$ различных позиций ошибок (некоторые позиции могут совпадать, поэтому $l_{m, m+1} \le l_m + l_{m+1}$). Тогда для коэффициентов многочлена позиций ошибок мы сможем составить $d-1-l_{m, m+1}$ уравнений вида (4) относительно синдромов S_{mi} и столько же относительно синдромов $S_{m+1, i}$.

Таким образом, если $2(d-1-l_{m, m+1}) \ge l_{m, m+1}$ или $l_{m, m+1} \le [2/3(d-1)]$, то число уравнений оказывается не меньше числа неизвестных. Если система уравнений, составленная подобным образом, имеет единственное решение и все корни полученного многочлена лежат в допустимом множестве значений $Z_0, \ldots, Z_{n-1}$, то с определенной степенью уверенности можно утверждать, что мы смогли восстановить исходное сообщение.

Аналогичные построения дают нам возможность получить многочлен позиций ошибок по синдромам трех последовательных компонент, если число различных позиций ошибок $l_{m, \ldots, m+2}$ в них не более $[3/4(d-1)]$. Для четырех компонент $l_{m, \ldots, m+3} \le [4/5(d-1)]$ и т. д. При достаточной длине векторов мы можем восстанавливать до $d-2$ ошибок в последовательных компонентах. Описанный алгоритм обозначим A_1.

Оценить вероятности ошибочного декодирования и отказа от декодирования такого алгоритма оказывается очень трудно, поскольку существенным для данного алгоритма является совпадение позиций ошибок в соседних координатах. Тем не менее это не уменьшает его практической значимости, поскольку он дает шанс на восстановление в случаях, когда традиционные методы дают отказ. В частности, мы можем восстановить до $d-2$ полностью измененных векторов, в то время как традиционные методы покомпонентного декодирования ограничивают нас числом $[(d-1)/2]$.

Отметим еще одну возможность увеличения мощности алгоритма декодирования. Она основана на применении традиционного алгоритма восстановления ошибок при наличии стираний. Пусть для m-й компоненты определены $l_m < d - 1$ ошибок. Будем рассматривать найденные позиции как стирания для компоненты $m + 1$. Тогда алгоритм восстановления применим, если $l_m + 2(l_{m,m+1} - l_m) \le d - 1$ или $l_{m,m+1} \le (d - 1)/2 + l_m/2$. В частности, при $l_m > (d - 1)/2$ имеем

$$(d - 1)/2 + l_m/2 > (d - 1)(1/2 + 1/4) > 3/4(d - 1).$$

Рассматривая большее количество смежных компонент и применяя идеи алгоритма A_1, получим следующие необходимые условия восстановления:

$$l_{m,\ldots,m+2} \le 2/3(d - 1) + l_m/3,$$

$$l_{m,\ldots,m+3} \le 3/4(d - 1) + l_m/4 \quad \text{и т. д.}$$

Верхняя граница по-прежнему остается $d - 2$. Этот алгоритм обозначим A_2.

Теперь можно дать неформальное описание алгоритма декодирования, объединяющего в себе все предыдущие.

Шаг 1. Для каждой компоненты независимо применяем алгоритм A_0. Если все компоненты восстановлены, алгоритм заканчивает работу с результатом «успешное декодирование», иначе на шаг 2.

Шаг 2. Применяем алгоритм A_2 ко всем декодированным на предыдущем шаге компонентам как в сторону увеличения, так и в сторону уменьшения индексов. Если все компоненты восстановлены, алгоритм заканчивает работу с результатом «успешное декодирование», иначе на шаг 3.

Шаг 3. Применяем алгоритм A_1 ко всем последовательным недекодированным компонентам. Если ни одна новая компонента не декодирована, то алгоритм заканчивает работу с результатом «отказ от декодирования». Если все компоненты восстановлены, алгоритм заканчивает работу с результатом «успешное декодирование», иначе на шаг 2.

Программная реализация этого алгоритма подтвердила его практическую значимость для систем хранения при описанном выше характере возникновения ошибок.

В заключение отметим применимость этого алгоритма для декодирования векторов при наличии ошибок сбоя синхронизации (одни из самых трудных ошибок для декодирования), когда вставка или потеря нескольких бит в m-компоненте вектора приводит к изменению всех последующих компонент. Поскольку есть тесная зависимость между позициями ошибок в соседних компонентах, приведенный алгоритм может быть успешно применен для исправления ошибок. Если мы работаем в поле $GF(2^{16})$, $GF(2^{32})$ и т. д., то, сравнивая исправленные значения с принятыми, имеем хорошую базу для диагностирования вставок или потерь небольшого количества битов, что позволяет восстанавливать сбои синхронизации и, следовательно, уменьшать количество ошибок в следующих компонентах.

СПИСОК ЛИТЕРАТУРЫ

1. **Форни Д.** Каскадные коды. М.: Мир, 1970.
2. **Муттер В. М.** Основы помехоустойчивой телепередачи информации. Л.: Энергоатомиздат, 1990.

Поступила в редакцию 7 февраля 1995 г.