

Рис. 2

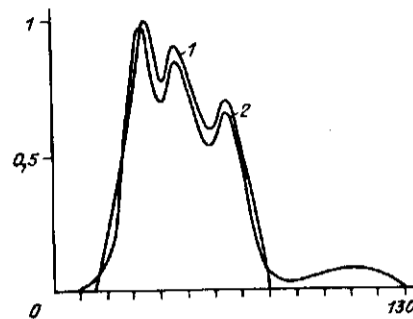


Рис. 3

венное восстановление профиля исходной функции (в смысле достижения сверхразрешения, сохранения амплитудных соотношений и формы максимумов), нежели линейный метод Фурье-фильтрации.

СПИСОК ЛИТЕРАТУРЫ

1. Уидроу Б., Стирнз С. Адаптивная обработка сигналов.—М.: Радио и связь, 1989.
2. Даджион Д., Мерсеро Р. Цифровая обработка многомерных сигналов.—М.: Мир, 1988.
3. Хемминг Р. В. Цифровые фильтры.—М.: Недра, 1987.
4. Джейнс Э. Т. О логическом обосновании методов максимальной энтропии // ТИИЭР.—1982.—70, № 9.
5. Кей С. М., Марпл С. Л. Современные методы спектрального анализа: Обзор // ТИИЭР.—1981.—69, № 11.
6. Mead R. L. Approximate solution of Fredholm linear integral equation by the maximum entropy method // J. Math. Phys.—1986.—70, N 9.—P. 2403.
7. Кэдзоу Дж. А. Спектральное оценивание: Метод переопределенной системы уравнений рациональной модели // ТИИЭР.—1982.—70, № 9.
8. Аратский Д. Б., Солдатов Е. А., Фидельман В. Р. О вычислительных аспектах спектрального оценивания методом максимума энтропии.—Деп. в ВИНТИ 10.03.87, № 1751-В87.

Поступило в редакцию 10 декабря 1990 г.

УДК 681.325.3

В. А. Козлачков, И. И. Коршевер, П. А. Полозков, И. Г. Ремель,
К. В. Тесленко

(Новосибирск)

ОБНАРУЖЕНИЕ И ИСПРАВЛЕНИЕ ОШИБОК В НАКОПИТЕЛЯХ НА МАГНИТООПТИЧЕСКИХ ДИСКАХ

Кратко рассмотрены алгоритмы кодирования и декодирования в системе Рида — Соломона для оценки их сложности относительно требуемого быстродействия кодера — декодера (КОДЕКа), «на лету» исправляющего испорченные данные считываемой информации с магнитооптического диска с реверсивной записью. Описана реализация КОДЕКа, удовлетворяющего требованиям стандарта ISO, оценены параметры устройства. Намечены пути микронной реализации КОДЕКа.

Введение. Осваиваемые в настоящее время устройства оперативного хранения на магнитооптических дисках (МОД) чрезвычайно перспективны с точки зрения достигаемой плотности упаковки информации и удельной

стоимости хранения [1]. Однако надежность такого способа хранения низка из-за значительного уровня статических помех, вызываемых локальными неоднородностями носителя, и случайных помех, возникающих в процессе взаимодействия оптической головки с магнитооптическим носителем. Для повышения достоверности хранимых данных применяют усложненные методы кодирования и декодирования, позволяющие обнаруживать и исправлять более одной ошибки в ситуациях сбоя. Эти методы, развитые ранее в теории и практике систем телекоммуникаций [2], впервые появились в архитектуре компьютера, и их освоение породило специфические проблемы. Необходимо было решить ряд методических (выбор типа кода, обладающего достаточной корректирующей способностью и вместе с тем минимальной избыточностью) и практических (обеспечение высокого быстродействия при минимальной задержке корректирующего воздействия и минимальной структурной сложности) вопросов.

Изучение методических проблем привело к появлению стандарта для накопителей информации на магнитооптическом диске с реверсивной записью [3], в котором для увеличения достоверности считывания информации с 10^{-5} до 10^{-12} ошибок на байт рекомендуется применять код Рида — Соломона. Этот тип кода относится к классу линейных циклических блочных кодов, обнаруживающих и исправляющих ошибки [2]. Определенный стандартом помехозащищенный код есть укороченный циклический (1200, 1040) код Рида — Соломона с расслоением на 10 кодовых слов по 120 байт каждое. По сравнению с форматом кодирования для МОД с однократной записью (CD — ROM, [1]) такой код значительно более избыточен. Подобные коды называются кодами с большим межкодовым расстоянием (в данном случае 17) и обладают повышенной корректирующей способностью — до восьми испорченных байтов в одном кодовом слове или до 80 байтов в одном блоке данных при 160 избыточных байтах на 1024 байта полезной информации.

Однако реализация кодера—декодера по указанному стандарту (далее КОДЕК), работающего «на лету» и сопоставимого по сложности с остальным оборудованием дискового контроллера, до сих пор остается проблемой.

Именно такая задача и была поставлена перед авторами.

В работе кратко рассмотрены алгоритмы кодирования и декодирования представленных в системе Рида — Соломона байтовых последовательностей для оценки их сложности относительно требуемого быстродействия КОДЕКа с учетом установленной стандартом скорости записи 375 Кбайт/с. Далее описано устройство КОДЕКа, удовлетворяющее сформулированным к нему требованиям и предназначенное для работы в составе макетно-моделирующего комплекса МОД; оценены параметры устройства.

В заключении работы намечены пути микроэлектронной реализации КОДЕКа.

Обоснование выбора реализации КОДЕКа. Для емкости сектора, равной 1024 байта, информационные байты, дополненные байтами-пропусками и байтами проверки на четность, составляют информационную часть сектора данных, образующую матрицу 104×10 . Индексация элементов матрицы на рис. 1, a (N — номер блока), приведенная к системе счисления по основанию $(104, 10)$, соответствует физической последовательности размещения байтов $K = 0 \dots 1039$ на дорожке:

$$K = 10k_1 + k_0, \quad \text{где } k_0 = 0, 1, \dots, 9; \quad k_1 = 0, 1, \dots, 103.$$

Для расслоения блока данных на 10 информационных слов образуем транспонированную систему счисления $K = 104k_0 + k_1$, где перечисление элементов ведется сначала по строкам ($k_1 = 0 \dots 103$), а затем по столбцам ($k_0 = 0 \dots 9$).

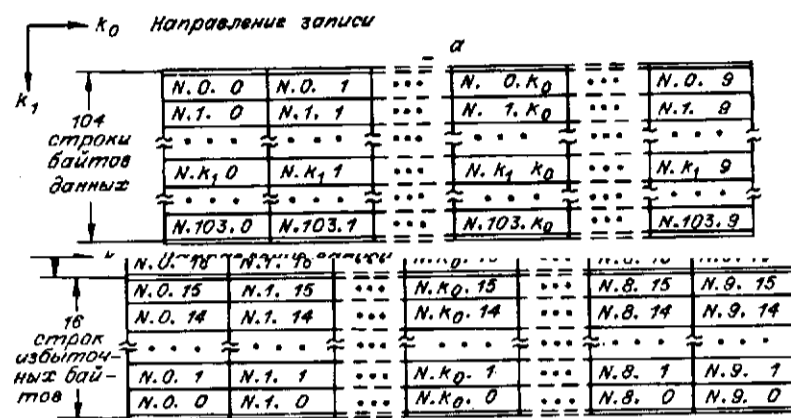


Рис. 1

В соответствии с теорией линейных блочных кодов [2] любой столбец такой системы может быть рассмотрен как одно из 10 информационных слов, выраженное полиномом

$$I(x) = \sum_{j=16}^{119} i_j x^j,$$

где i_j — информационный байт со значением (i) и с индексами (N, k_0, j) , N и k_0 любые, $j = 119 - k_1$.

Сто четыре строки информационных байтов с новыми индексами составляют на рис. 1, b информационную часть полного сектора данных размером 1200 байтов.

Чтобы получить систематический код Рида — Соломона, необходимо поделить исходный полином $I(x)$ на полином, генерирующий код, который задан стандартом [3]

$$g(x) = \prod_{j=120}^{135} (x - \alpha^j)$$

и имеет степень 16, где коэффициенты $\{\alpha^j\}$ принадлежат полю Галуа (GF), построенному на основе неприводимого полинома:

$$P(x) = x^8 + x^5 + x^3 + x^2 + 1.$$

Полиномиальное деление (кодирование), как и все вычисления при декодировании, выполняется в этом поле.

Остаток от деления образует избыточные байты c_0, \dots, c_{15} , которые, будучи присоединенными к информационному полиному, составляют вместе с ним кодовое слово:

$$C(x) = I(x) + \sum_{j=0}^{15} c^j x_j.$$

Для кодирования одного столбца исходного информационного блока требуется $16 \times 120 = 1920$ умножений и столько же сложений в поле $GF(256)$.

Полученное таким образом кодовое слово образует полином, корни которого совпадают с корнями порождающего полинома.

На рис. 2 приведена блок-схема процесса кодирования—декодирования. При записи кодового слова в МОД и при его считывании в него вносится последовательность байтовых ошибок $e(0), e(1), \dots, e(119)$, представляемая полиномом

$$E(x) = \sum_{j=0}^{119} e_j x^j.$$

На рисунке этот процесс представлен как внесение ошибок в канал связи. На выходе канала при этом образуется принятый полином $U(x) = C(x) + E(x)$.

Задача декодера — извлечение из принятого полинома (точнее, из его синдромов) полинома ошибок $E(x)$, такого, который, будучи сложением с «испорченным» словом $U(x)$, восстановил бы его первоначальное значение $C(x)$.

Наибольшее распространение при реализации декодеров для кодов Рида — Соломона получил алгоритм быстрого декодирования, описанный в [2], который содержит следующие этапы:

1. Вычисление полинома синдрома ошибок:

$$S(x) = \sum_{k=0}^{15} s(k) x^k.$$

Коэффициенты $s(k)$ могут быть вычислены по схеме Горнера:

$$s(k) = (v_{119} \alpha^k + v_{118})(\alpha^k + \dots + v_1)(\alpha^k + v_0),$$

где $v_{119} \dots v_0$ — принятые байты данных в кодовом слове; $\{\alpha^k\}$ — k -й корень из $g(x)$. Такое прямое вычисление синдрома требует 1904 умножения и 1920 сложений.

2. Вычисление полинома локаторов ошибок $A(x)$ и полинома значений ошибок $W(x)$:

$$W(x) = S(x)A(x) \pmod{x^{16}}.$$

Вычисление этих полиномов производится в соответствии с алгоритмом Берлекэмп — Мессе, который, по оценкам [2], требует $6t^2$ умножений, где t — число ошибок; при $t = 8$ необходимо около 400 умножений.

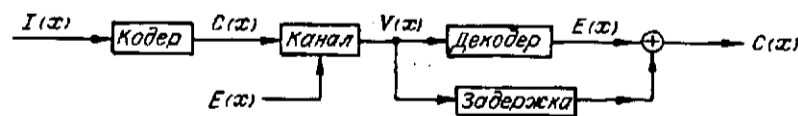


Рис. 2

3. Нахождение корней $A(x)$ для определения позиций ошибок. При этом удобно применить процедуру Ченя, при которой осуществляются перебор и подстановка в $A(x)$ всех возможных значений $\{x\}$ от $\{\alpha^{-119}\}$ до $\{\alpha^0\}$; здесь необходимо выполнить $104(8 - 1) = 728$ умножений и 832 сложения.

При необходимости восстанавливать так называемые «стертые» байты (см. ниже) число операций в п. 2 и 3 удваивается.

4. Если $A(\alpha^{-i}) = 0$, то необходимо определить величину ошибки по алгоритму Форни [2]:

$$e_i = - \frac{W(x)}{A(x)} \Big|_{x=\alpha^{-i}} \quad (i = 0, 1, \dots, 119).$$

Этапы 2 и 4 требуют вычисления обратных величин элементов поля Галуа.

Если число испорченных байтов в кодовом слове не превышает восьми, то после вычитания всех ошибок из соответствующих байтов восстанавливается исходная информация.

Таким образом, для 1-го сектора данных необходимо осуществить $3030 \times 10 - 4700 \times 10$ операций типа «умножение и сложение» ($A \times B + C$) в полях Галуа, или 26—40 операций на один информационный байт. Ориентируясь на оговоренный стандартом поток данных при считывании с диска (поток при записи в 3 раза ниже вследствие добавления циклов стирания и сравнения), равный 375 Кбайт/с (2,66 мкс/байт) [3], получим, что декодер должен осуществлять одну операцию типа $A \times B + C$ за 100—70 нс (производительность декодера 10—15 млн опер./с).

Необходимо отметить, что описанный выше алгоритм декодирования успешно исправляет до восьми ошибок в одном кодовом слове, не анализируя так называемых «стираний», — выявленных демодулятором модуляционного кода RLL2/7 позиций некоторой части ошибок в момент считывания [2]. При анализе «стираний» число испорченных байтов в кодовом слове, которое может быть восстановлено, меняется от 8 до 16. В приведенных выше оценках нижние границы относятся к случаю без анализа «стираний», а верхние — когда «стираний» анализируется.

На рис. 3 представлена функциональная схема декодера (номера ее блоков 1—4 соответствуют пунктам алгоритма декодирования, описанного выше).

Разработки в области КОДЕКов Рида — Соломона проводятся по двум основным направлениям: аппаратно-конвейерному и микропрограммному. В первом направлении, в котором создаются главным образом КОДЕКи для систем телекоммуникаций [4], декодер, функциональная схема которого приведена на рис. 3, реализуется обычно как специализированное конвейерное устройство, в котором отдельные этапы декодирования выполнены как аппаратные субблоки устройства (в некоторых случаях такие устройства выполнены в идеологии систолических матриц). Как правило, эти устройства принципиально последовательные (поразрядные). Производительность таких устройств достигает десятков миллионов бит/с соответственно достигаемой ныне тактовой частоте СБИС. Логично ожидать, что сложность устройства,

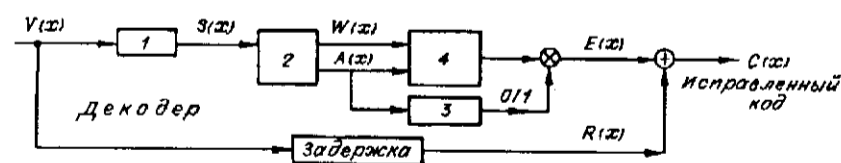


Рис. 3

выполненного при такой идеологии, достигает как минимум 30 последовательно включенных элементарных устройств (в простейшем случае регистров) соответственно числу операций на один символ. Подобные устройства строятся на суперкристаллах, содержащих сотни тысяч транзисторов.

Микропрограммный путь, используемый в менее быстродействующих каналах, реализуется с помощью микропроцессоров, система команд и операционные узлы которых усилены для выполнения вычислений в поле Галуа [5]. Дальнейшее рассмотрение в данной работе сосредоточено именно вокруг такого варианта.

Микропроцессорная реализация КОДЕКа. Микропроцессора, соответствующего сформулированным выше требованиям, в номенклатуре микропроцессорных устройств, освоенных отечественной промышленностью, не существует. Наиболее приемлемым оказался 8-разрядный микроконтроллер 1818 ВМ01 с производительностью 4 млн операций типа «регистр — регистр».

Функциональная схема КОДЕКа представлена на рис. 4: *DR1 ... DR3* — драйверы, *CNT* — счетчик адреса, *MUX* — мультиплексор адреса, *DC* — дешифратор, *PROM* — память микропрограмм; *a0 ... a11* — адрес микрокоманды, *i0 ... i15* — микрокоманда, *RAM* — память данных 1 Кбайт, *ivb* — шина «адреса — данных» 1818 ВМ01, *RG* — регистр адреса. Операция « $A \times B + C$ » в поле Галуа реализуется в среднем за 4—5 тактов, что достигается оптимизированным для выбранного микроконтроллера алгоритмом вычисления синдромов (наибольшей по времени части декодирования). Умножение выполняется с помощью таблицы логарифмирования (потенцирования) в поле Галуа. Таким образом, коэффициент загрузки микроконтроллера достигает шести, что потребовало бы от однопроцессорной реализации производительности порядка 58 млн опер./с. Реально один процессор, обладающий производительностью 4 млн опер./с, в состоянии исправить «на лету» восемь ошибок в потоке данных порядка 25,6 Кбайт/с. В рассмотренном ниже макете системы используется 10 таких микропроцессоров, что обеспечивает декодирование «на лету» потока данных, следующих с частотой 256 Кбайт/с. Так как необходимость в декодировании и исправлении ошибок возникает нерегуляр-

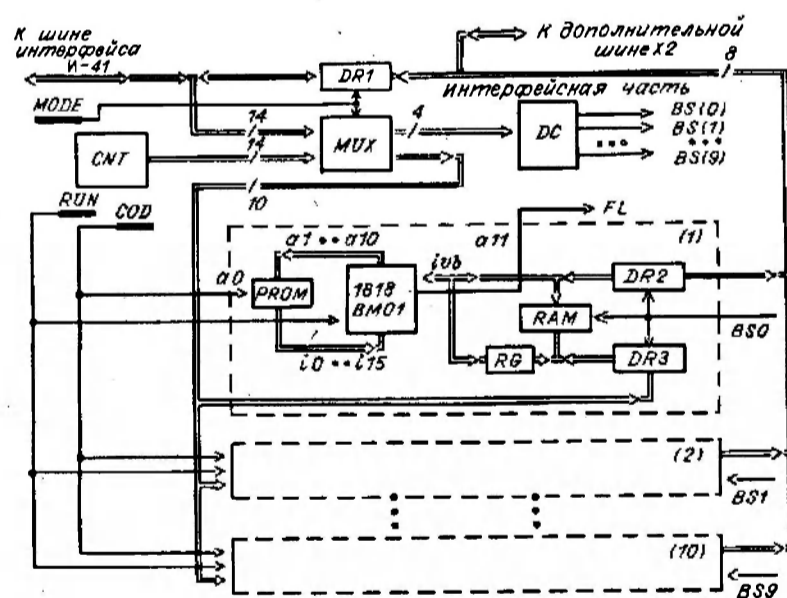


Рис. 4

но (только при обнаружении ошибки в секторе с помощью контрольной суммы CRC [3]), то такая производительность обеспечивает в среднем и заданный стандартом поток 375 Кбайт/с. Кодирование же одного сектора производится в реальном времени при потоке данных до 446 Кбайт/с.

КОДЕК взаимодействует с управляющей ЭВМ (УЭВМ) с помощью стандартного интерфейса И-41 [6]; каждый микроконтроллер снабжен собственной оперативной памятью емкостью 1 Кбайт и микропрограммной памятью $2K \times 16$ бит. Все элементы распределенной таким образом оперативной памяти связаны с помощью драйвера DR1 с шиной данных интерфейса, а обращение к ним в системе И-41 производится с помощью адресов адрА...адрЕ, выбирающих один из 10 банков. Эти адреса последовательно генерируются адресным счетчиком CNT, четыре младших разряда которого декодированы как один десятичный разряд (сигналы BS0, ..., BS9, активизирующие драйверы DR2/DR3 выбранных процессорных элементов).

Предусмотрено три режима работы:

— обмен данными по шине И-41 с УЭВМ для тестирования КОДЕКа и загрузки таблиц логарифмирования (потенцирования);

— блочный обмен данными с накопителем МОД по дополнительной шине; в качестве блоков выступают байты сектора; расслоение 1200 байтов сектора на 10 кодовых слов выполняется автоматически;

— обработка сегмента; КОДЕК при этом производит кодирование или декодирование данных, предварительно размещенных в его памяти.

Задание режима производится комбинацией состояний сигналов COD, RUN и MODE.

Все микроконтроллеры работают с момента их запуска независимо и сигнализируют об окончании процесса кодирования—декодирования с помощью общего флага готовности, объединяющего 10 выходов FL по И.

КОДЕК выполнен на двух идентичных печатных платах размером $233 \times 35 \times 220$ мм, по пять микропроцессорных элементов в каждой. С помощью двух разъемов типа СМП-96 каждая плата соединена с шиной И-41 и с дополнительной шиной X2.

Для отладки микропрограмм КОДЕКА была создана программно-аппаратная система развития, базирующаяся на ЭВМ «Электроника-79» и присоединенной к ней микроЭВМ «Электроника-60». Команды микропроцессора при этом генерировались на базовой ЭВМ с помощью специально написанного на макроассемблере (МАС) интерпретатора. Собранные с помощью стандартного построителя задач ТКВ (в операционной системе RSX-11M) микропрограммы через линию связи поступали в микроЭВМ, применяемую вместе с подключенным к ней микропроцессором в качестве аппаратного прототипа (память микроЭВМ использовалась при этом как память КОДЕКа). Специально написанный интерактивный отладчик производил загрузку микропрограмм и ввод (вывод) данных по линии связи с базовой ЭВМ. Кроме того, отладчик осуществлял чтение и запись любой зоны памяти микропроцессора и его внутренних регистров, пошаговое и блочное исполнение микропрограмм и исполнение отлаживаемых команд в пакетном режиме, подсчет числа циклов и задание точек останова.

Микропрограммный принцип реализации КОДЕКа позволяет настраивать устройство на произвольный формат кодового слова и выбранную систему кодирования. В частности, для обработки кодов Рида — Соломона, представленных в виде блоков по 512 байт, расслоенных на пять кодовых слов, достаточно одной такой платы.

Проверка описанного модуля в составе стенда для получения характеристик магнитооптических пластин позволила приступить к проектированию специализированной СБИС КОДЕКа. Такая СБИС

представляет собой специализированный 8-разрядный микропроцессор, система команд которого, подобно тому, как это имеет место в архитектуре широко распространенных однокристалльных процессоров для обработки сигналов [7], оптимизирована относительно векторно-конвейерных операций (в частности, относительно операций типа свертки). Увеличение арифметической мощности в таком микропроцессоре специфично и должно обеспечивать операцию $A \times B + C$ в поле Галуа за время порядка 100 нс. Такие арифметические средства реализуются гораздо дешевле, чем обычные АЛУ с ускоренным переносом и матричные умножители, составляющие существенную часть оборудования в современных сигнальных процессорах.

Это позволяет в кристалл объемом 15 тыс. вентилях и на тактовой частоте 10 МГц поместить два таких параллельно и синхронно работающих АЛУ с умножителями в поле Галуа, что обеспечивает заданную стандартом производительность не только в основном режиме, но и при исправлении «стертых» байтов.

Заключение. Устройство КОДЕКА для МОД с многократной записью, обеспечивающее заданные стандартом [3] условия, реализовано на базе многопроцессорной системы, построенной на стандартных 8-разрядных микроконтроллерах. Дальнейшая оптимизация базового процессорного элемента системы приводит к проекту специализированной заказной СБИС доступной сложности.

СПИСОК ЛИТЕРАТУРЫ

1. Типикин А. П., Петров В. В., Бабанин Л. Г. Коррекция ошибок в оптических накопителях информации.—Киев: Наук. думка, 1990.
2. Блейхут Р. И. Теория и практика кодов, исправляющих ошибки.—М.: Мир, 1986.
3. Second Version of ISO/IEC DP10089: 130 mm Rewritable Optical Disk Cartridges: Document ISO/IEC JTC 1/SC 23, 29.09.89, N 283.
4. Shao H. M., Reed I. S. On the VLSI design of a pipeline Reed — Solomon decoder using systolic arrays // IEEE Trans. on Comput.—1988.—37.—P. 1273.
5. А. с. 1529213 СССР. Устройство для арифметической обработки элементов конечного поля // П. Ф. Садыков, М. Ю. Балабаев, А. И. Савченко.—Заявл. 10.11.87; Опубл. 15.12.89, Бюл. № 46.
6. Комплекс базовый СМ 1803: Техническое описание и инструкция по эксплуатации. Ч. 2: Системный интерфейс СМ1800 (И-41) 1.620.002ТО1.
7. Аппаратные и программные средства цифровой обработки сигналов // Тематический выпуск ТИИЭР.—1987.—№ 9.

Поступило в редакцию 20 мая 1991 г.

УДК 621.382.2

А. И. Липтуга, В. К. Малютенко

(Киев)

ПОЛУПРОВОДНИКОВЫЙ ИСТОЧНИК ИК-ИЗЛУЧЕНИЯ

Описаны параметры полупроводникового источника излучения в спектральном диапазоне $5 \div 15$ мкм, принцип действия которого основан на модуляции теплового излучения полупроводникового кристалла за краем фундаментального поглощения. Эффект модуляции и управление параметрами излучения обеспечиваются инжекцией носителей заряда в базу германиевого диода при прямом смещении.