УДК 519.725

В.П. Семеренко

Винницкий национальный технический университет, Винница

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ АЛГОРИТМЫ ДЛЯ ИСПРАВЛЕНИЯ НЕЗАВИСИМЫХ ОШИБОК В ЦИКЛИЧЕСКИХ КОДАХ

Предложена многоуровневая графовая модель циклического кода на основе теории линейных последовательностных схем. С помощью графовой модели кода разработаны алгоритмы исправления независимых ошибок, проведена оценка корректирующей способности кода, введен подкласс легкодекодированных циклических кодов, предложен критерий для распознавания ошибок за пределами минимального кодового расстояния, которые обнаруживаются и исправляются. Проведен асимптотический анализ сложности рассмотренных алгоритмов при их последовательной и параллельной реализации.

Ключевые слова: *циклические коды, исправление ошибок, линейные последовательностные схемы, граф, параллельная обработка.*

Введение

Различные классы циклических кодов (коды СРС, БЧХ, Рида-Соломона) имеют оптимальное соотношение между корректирующей способностью и простотой программно-аппаратной реализации, что и обусловило их широкое использование во многих практических сферах [1, 2]. К сожалению, в последние годы мало внимания уделяется дальнейшей разработке теории этого класса помехоустойчивых кодов. В результате до сих пор в циклических кодах используют алгоритмы (например, алгоритм Берлекэмпа-Месси), предложенные более 40 лет назад. Однако стремительное развитие науки и техники выдвигает новые критерии и новые требования к вычислительным алгоритмам, в частности, на первый план выдвигается задача эффективного отображения вычислительных алгоритмов на архитектуру современных высокопроизводительных вычислительных систем [3].

Цель работы. Работа посвящена разработке методов и алгоритмов декодирования двоичных циклических кодов минимальной сложности и пригодных как для последовательной, так и для параллельной обработки.

1. Постановка задачи

Современная технология изготовления вычислительных систем на основе сверхбольших интегральных схем выдвигает новые требования к архитектуре этих систем. Наиболее удобными для микро- и наноэлектронной реализации являются однородные вычислительные структуры, состоящие их малого числа базовых типов элементарных модулей, соединенных между собой короткими и регулярными связями. Преимуществом такой архитектуры является возможность организации конвейерной и систолической обработки данных. А для максимального использования этого преимущества необходима разработка соответствующих параллельных алгоритмов для решения различных задач, в частности, для задач декодирования в системах передачи данных. Очень перспективным для решения задач кодопреобразования является специальный вид фильтров – линейные последовательностные схемы (ЛПС) [4]. ЛПС можно также рассматривать и как особую разновидность конечных автоматов, обладающих свойствами линейности. На основе теории ЛПС в работе предлагаются новые графовые модели и разработанные на их основе алгоритмы жесткого декодирования циклических кодов. Новые алгоритмы эффективны не только для традиционной реализации, но и матрично-конвейерной обработки.

Будем рассматривать циклический (n,k)-код Ω над полем Галуа GF(2) с минимальным кодовым расстоянием d_{min}. Кодовый вектор $Z(x)=(z_1,z_2,\ldots,z_n)$ кода Ω имеет длину n, размерность k и позволяет исправлять все независимые ошибки кратности 1, 2, … , τ_{min} $\left(\tau_{min}=\dfrac{d_{min}-1}{2}\right)$.

Для представления циклического кода Ω будем использовать математический аппарат ЛПС. Согласно [4], ЛПС Λ с l входами, m выходами и r элементами памяти в дискретные моменты времени t задается функцией состояний (переходов)

$$S(t+1)=A\times S(t)+B\times U(t),\quad GF(2), \qquad (1)$$

и функцией выходов

$$Y(t)=C\times S(t)+D\times U(t),\quad GF(2), \qquad (2)$$

где $A=\left\|a_{ij}\right\|_{r\times r}$, $B=\left\|b_{ij}\right\|_{r\times l}$, $C=\left\|c_{ij}\right\|_{m\times r}$, $D=\left\|d_{ij}\right\|_{m\times l}$ – характеристические матрицы ЛПС,

© В.П. Семеренко

$S = \|s_i\|_r$, $U = \|u_i\|_l$, $Y = \|y_i\|_m$ – векторы состояний, входной и выходной.

Размерности матриц ЛПС Λ и параметры циклического кода Ω связаны через коэффициент r, который для кода равен числу контрольных разрядов кодового вектора $C(x)$ при систематическом кодировании ($r = n - k$). При аппаратной реализации ЛПС с одним входом и одним выходом удобно использовать ЛПС с такими характеристическими матрицами:

$$A = \begin{Vmatrix} 0 & 0 & 0 & \cdots & 0 & g_0 \\ 1 & 0 & 0 & \cdots & 0 & g_1 \\ 0 & 1 & 0 & \cdots & 0 & g_2 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 0 & \cdots & 0 & g_{r-2} \\ 0 & 0 & 0 & \cdots & 1 & g_{r-1} \end{Vmatrix}; \quad B = \begin{Vmatrix} 1 \\ 0 \\ 0 \\ \cdots \\ 0 \\ 0 \end{Vmatrix}; \quad (3)$$

$$C = \| 0 \quad 0 \quad \cdots \quad 0 \quad 1 \|;$$

$$D = \| 0 \quad 0 \quad \cdots \quad 0 \quad 0 \|.$$

Элементы последнего столбца матрицы A из (3) представляют собой коэффициенты порождающего многочлена кода Ω:

$$g(x) = g_0 + g_1 x + g_2 x^2 + \ldots + g_{r-1} x^{x-1} + g_r x^r \quad . \quad (4)$$

2. Многоуровневая графовая модель циклических кодов

Поскольку циклические коды могут быть описаны с помощью ЛПС, являющейся конечным автоматом, поэтому в качестве графовой модели таких кодов естественно выбрать диаграмму переходов (ДП) автомата. Для r-мерной ЛПС Λ над полем $GF(2)$ ДП представляет собой ориентированный граф $G_{FA}(V_{FA}, E_{FA})$, в котором 2^r вершин из множества вершин V_{FA} соответствуют 2^r внутренним состояниям автомата, а дуги из множества дуг E_{FA} показывают направления переходов между внутренними состояниями. В общем случае из вершины v_j ($v_j \in V_{FA}$) может выходить нулевая дуга и единичная дуга, а также могут входить нулевая дуга и единичная дуга.

Порождающий многочлен $g(x)$ из (4) определяет не только корректирующие свойства циклического кода, но и структуру графа G_{FA}. Если порождающий многочлен $g(x)$ является непримитивным или равен произведению нескольких многочленов, тогда граф G_{FA} содержит некоторое количество нулевых циклов (НЦ) длины не более n, образованных нулевыми дугами. Эти НЦ можно упорядочить по следующим уровням.

На нулевом уровне будет располагаться тривиальный НЦ (ТНЦ), состоящий из одной вершины v_0, для которой входящая и выходящая нулевые дуги объединяются и образуют петлю. Далее, на первом уровне находится основной НЦ (ОНЦ) длины n, который связан с ТНЦ парой противоположно направленных единичных дуг. Все остальные НЦ, которые согласно [5],[6], также будем именовать периферийными НЦ (ПНЦ), распределяются по следующим уровням таким образом. На втором уровне располагаются те ПНЦ, каждый из которых связан с ОНЦ единичными дугами (обычно две пары противоположно направленных единичных дуг). На $(\tau+1)$-ом уровне каждый ПНЦ имеет единичные дуги с НЦ τ-го уровня и отсутствуют единичные дуги с НЦ уровней $(\tau-1)$ и менее (τ = 1,2,3, …). Такие единичные дуги будем также именовать “вертикальными”, причем дугу от НЦ τ-го уровня к НЦ $(\tau+1)$-го уровня будем именовать прямой “вертикальной”, а дугу от НЦ $(\tau+1)$-го уровня к НЦ τ-го уровня будем именовать обратной “вертикальной”. Пары противоположно направленных единичных дуг могут быть также и между НЦ одного уровня, будем именовать их “горизонтальными”. Если из вершины $v_{beg}^{\alpha\to\beta}$, принадлежащей α-му НЦ i-го уровня, выходит единичная дуга к вершине $v_{end}^{\alpha\to\beta}$, принадлежащей β-му НЦ j-го уровня, тогда вершину $v_{beg}^{\alpha\to\beta}$ будем именовать начальной “вертикальной” связывающей вершиной (ВСВ) α-го НЦ относительно β-го НЦ, вершину $v_{end}^{\alpha\to\beta}$ – конечной ВСВ β-го НЦ относительно α-го НЦ ($j = i-1$ или $j = i+1$). Одна и та же вершина в некоторых случаях может одновременно быть начальной ВСВ и конечной ВСВ относительно НЦ различных уровней.

Многоуровневый граф G_{FA} циклического (n,k)-кода с минимальным кодовым расстоянием d_{min} имеет следующие параметры:

1) на τ-ом уровне количество НЦ равно:

$$N_\tau = C_n^\tau \Big/ n \; ; \; \tau = 2 \div \tau_{min},$$

где C_n^τ – число сочетаний из τ по n;

2) не содержится “горизонтальных” единичных дуг между НЦ τ-го уровня

$$(\tau = 2 \div \tau_{min} - 1);$$

3) на τ-ом уровне в каждом НЦ содержится не менее τ конечных ВСВ относительно всех НЦ $(\tau-1)$-го уровня $(\tau = 1 \div \tau_{min})$.

В графе G_{FA} вершины соответствуют внутренним состояниям ЛПС. Последовательность векторов внутренних состояний ЛПС, которые соответствуют вершинам одного цикла в графе G_{FA}, также образуют цикл. Поскольку совокупность циклов из

векторов состояний имеет такую же структуру, что и совокупность циклов из вершин, поэтому для характеристики циклов из векторов состояний будем использовать те же термины: ТНЦ, ОНЦ и ПНЦ. В векторном представлении цикл ТНЦ будет представлен в виде r-разрядного вектора, состоящего из всех нулей, а остальные НЦ – множеством из n r-разрядных векторов.

В автоматной модели для связи между собой НЦ, образованных векторами состояний ЛПС, введем следующие обозначения: начальное "вертикальное" связывающее состояние (ВСС) $S_{beg}^{\alpha\to\beta}(t)$ α-го НЦ i-го уровня относительно β-го НЦ j-го уровня (соответствует вершине $v_{beg}^{\alpha\to\beta}$), и конечное ВСС $S_{end}^{\alpha\to\beta}(t)$ β-го НЦ i-го уровня относительно α-го НЦ j-го уровня (соответствует вершине $v_{end}^{\alpha\to\beta}$). В дальнейшем различие между НЦ, образованных вершинами графа, и НЦ, образованных векторами состояний ЛПС, определяется контекстом: в графовых моделях подразумеваются графовые НЦ, а при математических преобразованиях – автоматные НЦ.

3. Интерпретация ошибок на основе графовых моделей циклических кодов

Если на входы ЛПС, находящейся в нулевом начальном состоянии $S(0)$, подать двоичную последовательность $L(x)$, тогда ЛПС через n временных тактов снова возвратится в состояние $S(0)$. Множество всех двоичных последовательностей $L(x)$ длины n, переводящих ЛПС из нулевого начального состояния $S(0)$ снова в состояние $S(0)$, образует циклический (n,k)-код Ω над полем Галуа $GF(2)$. Каждая такая последовательность $L(x)$ есть кодовый вектор $Z(x)$ циклического (n,k)-кода.

Такое определение циклического кода с позиций теории ЛПС эквивалентно определению этого кода в алгебре многочленов (полиномов), в которой кодовый вектор должен делиться без остатка на порождающий многочлен кода.

При передаче данных по каналу связи вследствие различных помех некоторые разряды кодового слова случайным образом могут быть искажены, т.е. будет получен кодовый вектор $Z_{err}^{(\tau)}(x)$ с независимыми ошибками кратности τ. Взаимосвязь между кодовыми векторам $Z(x)$ и $Z_{err}^{(\tau)}(x)$ выражается через вектор ошибок $R_{err}^{(\tau)}(x)$:

$$R_{err}^{(\tau)}(x) = Z(x) + Z_{err}^{(\tau)}(x), \quad GF(2).$$

В алгебре многочленов результатом деления кодового вектора $Z_{err}^{(\tau)}(x)$ на порождающий многочлен $g(x)$ кода будет ненулевой остаток. Аналогично, при подаче на входы кодового вектора $Z_{err}^{(\tau)}(x)$ ЛПС через n временных тактов из нулевого начального состояния $S(0)$ перейдет в некоторое ненулевое состояние, которое будем именовать синдромом ошибки $S_{err}^{(\tau)}(n)$ кратности τ.

Путь длины n по дугам графа G_{FA}, которые соответствуют значениям разрядов кодового вектора $Z(x)$, начинается и оканчивается в вершине v_0. Для кодового вектора $Z_{err}^{(\tau)}(x)$ конечная вершина пути длины n будет иной.

ТЕОРЕМА 1. Путь длины n по дугам графа G_{FA}, которые соответствуют значениям разрядов кодового вектора $Z_{err}^{(\tau)}(x)$, начинается в вершине v_0 и оканчивается в вершине v_{err}, принадлежащей НЦ уровня τ графа G_{FA}, $\tau = 1 \div \tau_{min}$.

В графе G_{FA} вершина v_0 соответствует состоянию $S(0)$, а вершина v_{err} – состоянию $S_{err}^{(\tau)}(n)$ ЛПС. Кодовый путь, который начинается в вершине v_0 ошибки, и заканчивается в вершине v_{err}, будем называть прямым кодовым путем ξ_{rv}, а тот кодовый путь, который начинается в вершине v_{err} ошибки, и заканчивается в вершине v_0, будем называть обратным кодовым путем ξ_{rv}. Тот НЦ, который содержит вершину v_{err}, будем в дальнейшем именовать НЦ ошибки.

Между структурой вектора ошибок $R_{err}^{(\tau)}(x)$ и конфигурацией путей ξ_{st} и ξ_{rv} существует взаимно однозначное соответствие. Конфигурация нулевых и единичных дуг в кодовых путях между вершинами v_0 и v_{err} точно соответствует расположению единиц в векторе ошибок $R_{err}^{(\tau)}(x)$. Следовательно, определение структуры вектора ошибок $R_{err}^{(\tau)}(x)$ сводится к определению прямого ξ_{st} или обратного ξ_{rv} кодовых путей по графовой модели. Для циклического кода, способного исправлять τ независимых ошибок, существует два прямых кодовых путей ξ_{st} (соответствуют векторам $Z_{err}^{(\tau)}(x)$ и $R_{err}^{(\tau)}(x)$) и τ обратных кодовых путей ξ_{rv} между указанными вершинами v_0 и v_{err}. Поиск одного из этих путей с помощью различных графовых моделей ЛПС и составляет суть новых методов исправления ошибок в циклических кодах.

На рис. 1 показан один вариант прямого и обратного путей по дугам графа G_{FA} между вершиной v_0 и вершиной v_{err} в НЦ ошибки на уровне τ.

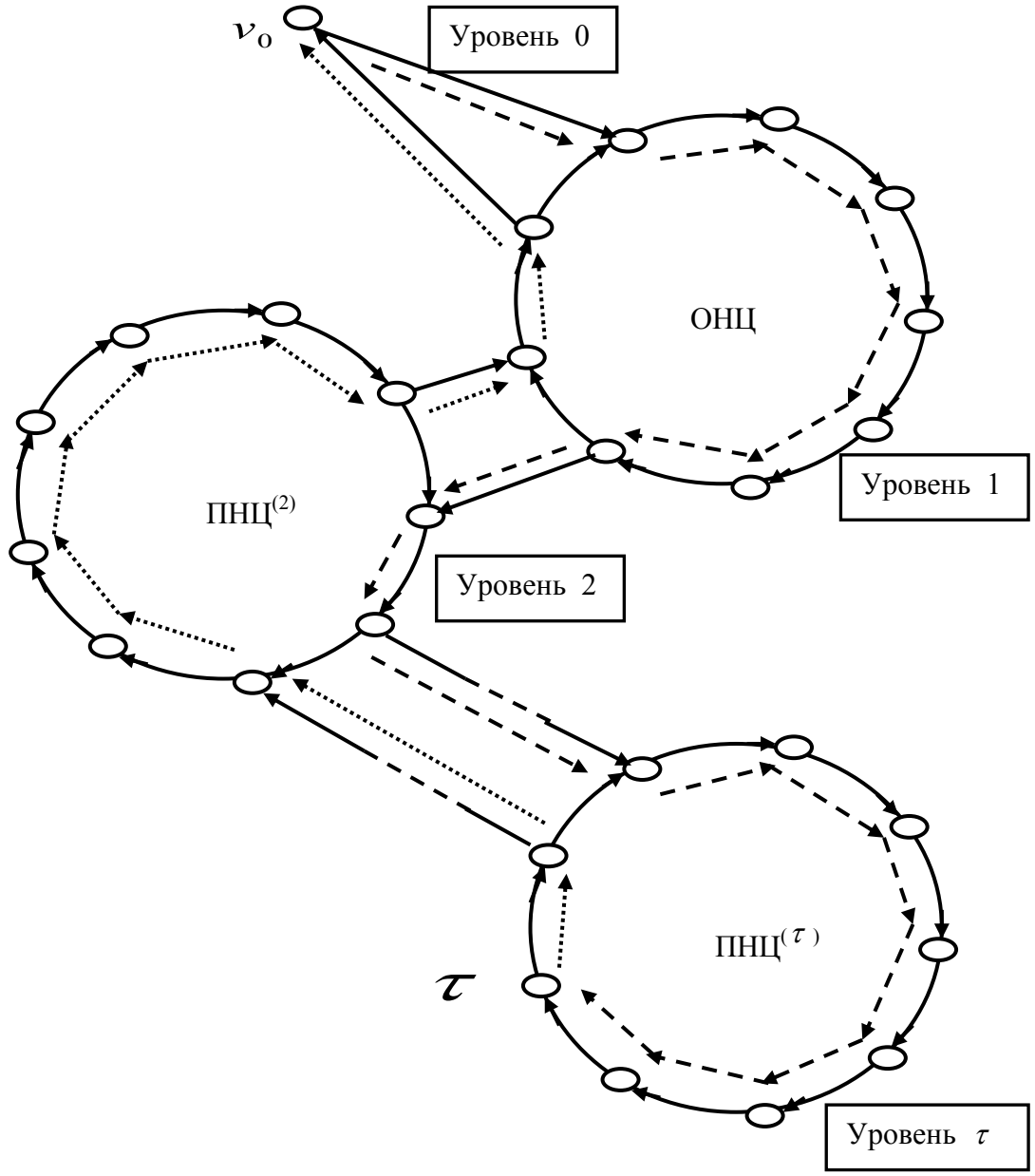

Рис. 1. Кодовые пути по дугам фрагмента графа G_{FA} :
– прямой кодовый путь; – обратный кодовый путь

Поиск прямого ξ_{st} или обратного ξ_{rv} кодовых путей по графу G_{FA} предполагает наличие этого графа. Однако построение графа G_{FA} при каждом исправлении возникшей ошибки займет слишком много времени, особенно для кодов большой размерности. Очевидно также, что не самым лучшим решением этой проблемы было бы его предварительное нахождение и сохранение в полном объеме. Поскольку граф нужен лишь для поиска пути между двумя вершинами, одна из которых (v_0) известна заранее, значит лучше вместо всего графа G_{FA} сохранять лишь возможные пути к вершине v_0. Однако вычисление и сохранение всех возможных путей по дереву равнозначно сохранению всех возможных синдромов исправляемых независимых ошибок, что также должно быть отвергнуто. Рассмотрим вначале решение этих задач в терминах графовой модели.

Внутри каждого НЦ конечная ВСВ связана нулевыми дугами с другими вершинами этого НЦ и максимальная длина пути к другим вершинам не превышает n. Можно сказать, что конечная ВСВ однозначно определяет все остальные вершины этого НЦ. Множество $M_{BCB} = \{v_{end}^{\alpha\to\beta}\}$ конечных ВСВ всех НЦ однозначно определит все вершины графа G_{FA} и при этом мощность множества M_{BCC} будет в n раз меньше мощности множества V_{FA} всех вершин этого графа.

Множества M_{BCC} и M_{path} могут служить базой для построения любого обратного кодового пути

от любой вершины графа G_{FA}. Для машинной реализации этой задачи, сформулированной выше в терминах графовой модели, лучше переформулировать в терминах автоматной модели.

Множество $M_{BCB} = \{v_{end}^{\alpha\to\beta}\}$ конечных ВСВ и множество $M_{path} = \{\xi_{path}^{(\tau)}\}$ базисных путей в графовой модели будут эквивалентны соответственно множеству $M_{BCC} = \{S_{end}^{\alpha\to\beta}(t)\}$ конечных ВСС и множеству $M_{err} = \{R_{base}^{(\tau)}\}$ базисных векторов ошибки в автоматной модели. Таким образом, если сформировать множества M_{BCC} и M_{err}, тогда мы сможем найти требуемый вектор ошибки для каждой конкретной ошибки. Однако, для кодов большой размерности потребуются значительные объемы памяти для хранения этих множеств.

Исследования показали, что множества M_{BCC} и M_{err} могут быть существенно сокращены, а для декодирования некоторых подклассов циклических кодов от них можно вообще отказаться.

ОПРЕДЕЛЕНИЕ 1. Конечное ВСС $S_{end}^{\alpha\to\beta}(t)$ α-го НЦ τ-го уровня относительно β-го НЦ $(\tau-1)$-го уровня называется регулярным, если оно содержит τ единиц, одна из которых расположена в младшем, (первом), разряде ($\tau = 1 \div \tau_{min}$).

В каждом НЦ τ-го уровня среди τ конечных ВСС может содержаться только одно регулярное конечное ВСС. В общем случае множество M_{BCC} состоит из двух подмножеств: подмножества M_{BCC}^{reg}, содержащего только регулярные конечные ВСС, и подмножества N_{BCC}^{nreg}, содержащего все остальные конечные ВСС. Рассмотрим соотношение между нерегулярными и регулярными конечными ВСС для каждого уровня графа G_{FA}.

УТВЕРЖДЕНИЕ 2. В пределах корректирующей способности циклического (n,k)-кода характеристическое множество M_{BCC} содержит только регулярные конечные ВСС для τ-го уровня графа G_{FA}, если выполняется условие:

$$C_{n-k-1}^{\tau-1} \ge \left\lfloor C_n^{\tau} \Big/ n \right\rfloor \quad \text{для} \quad \tau = 1...\tau_{min}, \qquad (5)$$

где $\lfloor\ \rfloor$ – округление до целого в меньшую сторону.

Обычно регулярные конечные ВСС связаны между собой по соседним уровням простыми математическими соотношениями и от регулярного конечного ВСС из НЦ τ-го уровня легко получить все остальные регулярные конечные ВСС до самого ТНЦ. В таком случае отпадает необходимость в предварительном нахождении и хранении базисных векторов ошибки кратности τ и менее, поскольку такие векторы ошибок можно будет генерировать в процессе исправления ошибок.

ОПРЕДЕЛЕНИЕ 2. Циклические (n,k)-коды, для которых множество M_{BCC} для НЦ с первого по τ_{min}-й уровни содержит регулярные конечные ВСС, называются легкодекодируемыми в пределах минимального кодового расстояния.

Из (5) можно точно определить количество N_{BCC}^{nreg} нерегулярных конечных ВСС, которые необходимо предварительно определять и сохранять для последующего исправления ошибок:

$$N_{BCC}^{nreg} = \left\lfloor C_n^{\tau} \Big/ n \right\rfloor - C_{n-k-1}^{\tau-1} \quad \text{для} \quad \tau = 1...\tau_{min}.$$

ПРИМЕРЫ. Граф G_{FA} для $(23,12)$-кода Голея, который задается порождающим многочленом $g(x) = 1 + x^2 + x^4 + x^5 + x^6 + x^{10} + x^{11}$, содержит ТНЦ, ОНЦ длины 23, 11 ПНЦ второго уровня длины 23 и 77 ПНЦ третьего уровня длины 23. В этом коде десяти двойным ошибкам соответствуют регулярные ВСС и только одной двойной ошибке – нерегулярное конечное ВСС, а из 77 тройных ошибок – 32 нерегулярных конечных ВСС. Следовательно, нужно заранее определить 33 нерегулярных конечных ВСС, что составляет около 1,5 % от всех возможных синдромов независимых ошибок кода Голея. Можно сохранять все 89 регулярных и нерегулярных конечных ВСС, что составит менее 5% всех возможных синдромов. Можно сравнить полученные результаты с известным декодером кода Голея [7], который использует память емкостью 12К для хранения всех возможных синдромов.

Граф G_{FA} для (15,7)-кода БЧХ, который задается порождающим многочленом $g(x) = 1 + x^4 + x^6 + x^7 + x^8$ содержит ТНЦ, ОНЦ длины 15, 7 ПНЦ второго уровня длины 15, 8 ПНЦ длины 15 и 3 ПНЦ длины 5 третьего уровня. Этот код является легкодекодируемым для одиночных и двойных ошибок.

4. Декодирование циклических кодов за границей минимального кодового расстояния

Рассмотрим взаимосвязь корректирующей способности некоторых основных классов циклических кодов с их графовой моделью.

Циклические коды с примитивным порождающим многочленом имеют только ТНЦ и ОНЦ и поэтому могут исправлять только одиночные независимые ошибки и обнаруживать все ошибки четной кратности.

Существуют циклические коды с непримитивным порождающим многочленом и кодовым рас-

стоянием d_{min}, которые имеют в графе G_{FA} только τ_{min} уровней и общее количество N_τ вершин:

$$N_\tau = (C_n^1 + C_n^2 + \ldots + C_n^{\tau_{min}}).$$

Такие циклические коды называются совершенными.

Большинство циклических кодов с непримитивным порождающим многочленом и кодовым расстоянием d_{min} имеют в графе G_{FA} τ_{max}-й уровень и количество N_τ вершин ($\tau_{max} = \tau_{min} + 1$):

$$N_\tau > (C_n^1 + C_n^2 + \ldots + C_n^{\tau_{min}}).$$

Именно наличие ($\tau_{min} + 1$)-го уровня и позволяет соответствующему циклическому коду обнаруживать и исправлять некоторое количество независимых ошибок за границей минимального кодового расстояния d_{min}. Очень важным является различение между обнаруживаемыми и исправляемым ошибками кратности τ_{max}. Именно многоуровневая структура графа G_{FA} позволяет получить четкий и простой ответ на этот вопрос.

УТВЕРЖДЕНИЕ 3. Независимая ошибка максимальной кратности τ_{max} будет исправляемой, если соответствующий ей НЦ ошибки, имеющий длину m ($m \le n$), будет находиться на τ_{max}-м уровне графа G_{FA} и общее количество N_{edg} пар соседних "вертикальных" единичных дуг между этим НЦ и всеми НЦ ($\tau_{max} - 1$)-го уровня будет равно $N_{edg} = (m/n)\tau_{max}$.

Для определения всех независимых ошибок максимальной кратности τ_{max} необходимо построить весь граф G_{FA} и провести попарное сопоставление всех НЦ уровня τ_{max} со всеми НЦ уровня ($\tau_{max} - 1$). Эта задача требует значительных вычислительных ресурсов, однако для каждого кода она выполняется только один раз. Существующие справочные таблицы циклических кодов можно дополнить информацией о том, какие из кодов позволяют исправлять (а не только обнаруживать!) независимые ошибки за границей минимального кодового расстояния и указать точное количество этих ошибок.

5. Сложность последовательной и параллельной реализаций алгоритмов поиска ошибок

Для реализации рассмотренного метода поиска ошибок на основе графовой и автоматной моделей ЛПС разработаны три алгоритма.

Алгоритм 1 предназначен для формирования множества M_{BCC} конечных ВСС и множества M_{err} базисных векторов ошибки в терминах автоматной модели (множества M_{BCB} конечных ВСС и множества M_{path} базисных путей в терминах графовой модели). Суть Алгоритма 2 состоит в построении обратного кодового пути по автоматной модели ЛПС на основе нерегулярных конечных ВСС, выбираемых из множества M_{BCC}. Алгоритм 3 также строит обратный кодовый путь по автоматной модели ЛПС, но с помощью регулярных конечных ВСС, которые вычисляются в процессе декодирования.

В настоящей работе предлагается процедуру поиска независимых ошибок разделить на два этапа. На первом этапе (до начала декодирования) выполняется Алгоритм 1 для нахождения вспомогательных множеств M_{BCC} и M_{err}, а на втором этапе (этапе декодирования и исправления ошибок) выполняется Алгоритм 2 или Алгоритм 3 нахождения позиций ошибок в кодовом векторе. Алгоритм 2 более медленный, однако он всегда определит позиции независимой ошибки в пределах корректирующей способности кода. Алгоритм 3 работает очень быстро, но только для отдельных участков кодового пути. Наилучшей стратегией является их поочередное или параллельное выполнение.

Проведем асимптотический анализ сложности указаных алгоритмов при их последовательной и параллельной реализации. В основе всех алгоритмов лежит операция рекурсивного вычисления очередного состояния ЛПС по формуле (1), которая может быть заменена на определенное количество элементарных операций сравнения или сложения по модулю 2 между $(n-k)$-разрядными булевыми векторами. При оценке верхней границы сложности алгоритма будем учитывать количество таких элементарных операций.

Алгоритм 1 является самым трудоемким, его сложность при последовательной реализации равна $O(n^2 \times C_n^\tau)$, однако он выполняется только один раз для выбранного циклического кода. Вычисления в Алгоритме 2 сводятся к задачам поиска в m-элементных множествах M_{BCC} и M_{err}, где $m = N_{BCC}^{nreg}$ следовательно, сложность этого алгоритма будет полиномиальной при последовательном поиске, и линейной (т.е. $O(\log_2(m))$) при бинарном поиске в заранее отсортированном массиве M_{BCC}. Вычисления в Алгоритме 3 сводятся только к вычислению очередного состояния ЛПС по формуле (1), поэтому сложность этого алгоритма будет линейной, т.е. $O(n)$.

При параллельной реализации указанных алгоритмов с помощью n процессорных элементов, выполняющих операции сравнения или сложения по модулю 2 между $(n-k)$-разрядными булевыми век-

торами, сложность реализации каждым алгоритмом уменьшится соответственно в n раз.

Важно отметить взаимосвязь между скоростью кода $R = k/n$ и сложностью его декодирования. Низко- и среднескоростные циклические коды ($R \le 1/2$) являются в основном легкодекодируемыми, но с ростом скорости кода увеличивается доля нерегулярных конечных ВСС, которые необходимо заранее вычислить и сохранить.

Заключение

Главным критерием пригодности вычислительного алгоритма к наиболее производительному виду параллелизма – матрично-конвейерному – является возможность декомпозиции всего процесса вычислений на пошаговое выполнение однородных действий одинаковой длительности [8].

Предлагаемые алгоритмы, которые строят кодовые пути в графовой модели, основаны на рекурсивном вычисление очередного состояния ЛПС. Поскольку они отвечают указанному критерию, поэтому могут быть эффективно отображены на архитектуру современных матричных вычислителей в виде сверхбольших интегральных схем.

Предлагаемые алгоритмы также можно использовать и при их последовательной реализации. Сложность реализации Алгоритма 2 не уступает сложности реализации известных методов, и при этом все его действия строго формализованы и понятны.

Для сравнения отметим, что наиболее известный алгоритм (Берлекэмпа-Месси [2]) имеет сложность $O(6\tau^2)$ операций умножения булевых матриц размерности $(n-k)\times(n-k)$, при этом использует несколько разнотипных процедур, одна из которых (процедура Ченя) выполняются методом проб и ошибок.

Предложен метод выделения регулярных ВСС (т.е. синдромов ошибок регулярной структуры), что дало возможность выделить подкласс легкодекодируемых циклических кодов и простой метод исправления ошибок с почти линейной сложностью (Алгоритм 3). Благодаря многоуровневой графовой модели циклического кода на основе теории ЛПС очень легко оценить корректирующие способности кода, различать обнаруживаемые и исправляемые ошибки за пределами минимального кодового расстояния.

Настоящая работа посвящена циклическим кодам над полем Галуа $GF(2)$, дальнейшие исследования направлены на применение предлагаемых подходов и для циклических кодов в недвоичных полях Галуа.

Список литературы

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение: пер. с англ. / Б. Скляр. – 2-е изд., испр. – М.: Издательский дом "Вильямс", 2004. – 1104 с.

2. Блейхут Р. Теория и практика кодов, исправляющих ошибки: пер. с англ. / Р. Блейхут. – М.: Мир, 1986. – 576 с.

3. Миллер Р. Последовательные и параллельные алгоритмы: Общий подход: пер. с англ. / Р. Миллер, Л. Боксер. – М.: БИНОМ. Лаборатория знаний, 2006. – 406 с.

4. Гилл А. Линейные последовательностные машины: пер. с англ. / А. Гилл – М.: Наука, 1974. – 288 с.

5. Семеренко В.П. Параллельное декодирование циклических кодов Боуза-Чоудхури-Хоквингема / В.П. Семеренко // Электронное моделирование. – 1998. – № 1. – С. 82-87.

6. Semerenko V.P. Burst-Error Correction for Cyclic Codes / V.P. Semerenko // Proceeding of International IEEE Conference EUROCON 2009. – S.Petersburg, Russia. – P. 1646-1651.

7. Морелос-Сарагоса Р. Искусство помехоустойчивого кодирования. Методы, алгоритмы, применение: пер. с англ. / Р. Морелос-Сарагоса. – М.: Техносфера, 2006. – 320 с.

8. Кун С. Матричные процессоры на СБИС: пер. с англ. / С. Кун. – М.: Мир, 1991. – 672 с.

Поступила в редколлегию 18.03.2010

Рецензент: д-р техн. наук, проф. В.И. Долгов, Харьковский национальный университет радиоэлектроники, Харьков.

ВИСОКОПРОДУКТИВНІ АЛГОРИТМИ ДЛЯ ВИПРАВЛЕННЯ НЕЗАЛЕЖНИХ ПОМИЛОК В ЦИКЛІЧНИХ КОДАХ

В.П. Семеренко

Запропонована багаторівнева графова модель циклічного коду на основі теорії лінійних послідовнісних схем (ЛПС). За допомогою графової моделі коду розроблені алгоритми виправлення незалежних помилок, проведена оцінка коректуючої здатності коду, введено підклас легкодекодованих циклічних кодів, запропоновано критерій для розпізнавання помилок за межами мінімальної кодової відстані, які виявляються та виправляються. Проведено асимптотичний аналіз складності розглянутих алгоритмів при їх послідовній та паралельній реалізації.

Ключові слова: *циклічні коди, виправлення помилок, лінійні послідовнісні схеми, граф, паралельна обробка.*

HIGH-PERFORMANCE ALGORITHMS FOR CORRECTION OF INDEPENDENT ERRORS IN CYCLIC CODES

V.P. Semerenko

The multilevel count model of cyclic code is offered on the basis of theory of linear sequential charts. By the count model of code the algorithms of correction of independent errors are developed, the estimation of correcting ability of code is conducted, the subclass of lung-coded of cyclic codes is entered, a criterion is offered for recognition of errors outside minimum code distance, which are revealed and corrected. The asymptotic analysis of complication of the considered algorithms is conducted during their successive and parallel realization.

Keywords: *cyclic codes, correction of errors, linear sequential charts, count, simultaneous processing.*