Литература

1. Шахнович, И. В. Персональные беспроводные сети стандартов IEEE 802.15.3 и 802.15.4 [Текст] / И. В. Шахнович // Электроника: НТБ. – 2004. – № 6. – С. 32–36.
2. Шахнович, И. В. Современные технологии беспроводной связи [Текст] / И. В. Шахнович. – М.: Техносфера, 2004. – 288 с.
3. Вишневский, В. М. Широкополосные беспроводные сети передачи информации [Текст] / В. М. Вишневский, А. И. Ляхов, С. Л. Портной, И. В. Шахнович. – М.: Техносфера, 2005. – 592 с.
4. Yushchenko, A. G. High Unloaded-Qs WDR Filters Designing [Text] / A. G. Yushchenko // International Journal of Infrared and Millimeters Waves. – 2001. – Vol. 22, Issue 12. – P. 1831–1836. doi: 10.1023/a:1015031802727
5. Стешенко, С. А. Метод частичных областей с учетом особенностей во внутренних задачах с произвольными кусочно-координатными границами. Часть 2. Плоско-поперечные соединения и "in-line" объекты [Текст] / С. А. Стешенко, С. А. Приколотин, А. А. Кириленко, Д. Ю. Кулик, Л. А. Рудь, С. Л. Сенкевич // Радиофизика и электроника. – 2013. – Т. 4 (18), № 3. – С. 13–21.
6. Миттра, Р. Аналитические методы теории волноводов [Текст] / Р. Миттра, С. Ли. – М., 1974. – С. 181–242.
7. Никольский, В. В. Декомпозиционный подход к задачам электродинамики [Текст] / В. В. Никольский, Т. И. Никольская. – М.: Наука, 1983. – 304 с.
8. Ющенко, А. Г. Intellectual CAD for Three-Tier Wide Band WDR Filters [Text] / А. Г. Ющенко, Д. Б. Мамедов, Д. М. Зайцев // WET. – 2012. – № 1. – С. 30–35.
9. Ющенко, А. Г. Evolutionary design of seven-tier LM-mode filters optimized with original knowledge-based CAD system [Text] / Д. Б. Мамедов, А. Г. Ющенко // Вест. НТУ «ХПИ» Техніка та електрофізика високих напруг. – 2014. – № 21. – С. 159–171.
10. Приколотин, С. А. Метод частичных областей с учетом особенностей во внутренних задачах с произвольными кусочно-координатными границами. Часть 1 [Текст] / С. А. Приколотин, А. А. Кириленко // Радиофизика и электрон. – 2010. – Т. 15, № 1. – С. 17–29.
11. Xiao, S.-Q. Millimeter wave technology in wireless PAN, LAN, and MAN [Text] / S.-Q. Xiao. – CRC Press, 2008.
12. IEEE Std 802.15.3c-2009. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs) [Text]. – Amendment 2: Millimeter-wave-based Alternative Physical Layer Extension. – IEEE, 2009.
13. IEEE Std 802.15.3 Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for High Rate Wireless Personal Area Networks (WPANs) [Text]. – IEEE, 2003.

Розглянуті теоретичні основи кодів CRC за допомогою математичного апарату лінійних послідовнісних схем (ЛПС). Проаналізована інтерпретація CRC як контрольної суми (Cyclic Redundancy Check) та як вкорочених циклічних кодів (Cyclic Redundancy Code). Дані рекомендації з вибору породжувальних поліномів для CRC. Запропоновано метод паралельного обчислення CRC зі скороченням числа ітерацій в $\rho(\rho \le r)$ разів для довільного поліному степені r

Ключові слова: CRC коди, вкорочені циклічні коди, контрольна сума, породжувальний поліном, лінійна послідовнісна схема

Рассмотрены теоретические основы кодов CRC с помощью математического аппарата линейных последовательностных схем (ЛПС). Проанализирована интерпретация CRC как контрольной суммы (Cyclic Redundancy Check) и как укороченных циклических кодов (Cyclic Redundancy Code). Даны рекомендации по выбору порождающих полиномов для CRC. Предложен метод параллельного вычисления CRC с сокращением количества итераций в $\rho(\rho \le r)$ раз для произвольного полинома степени r

Ключевые слова: CRC коды, укороченные циклические коды, контрольная сумма, порождающий полином, линейная последовательностная схема

УДК 681.32
DOI: 10.15587/1729-4061.2015.47860

ТЕОРИЯ И ПРАКТИКА CRC КОДОВ: НОВЫЕ РЕЗУЛЬТАТЫ НА ОСНОВЕ АВТОМАТНЫХ МОДЕЛЕЙ

В. П. Семеренко
Кандидат технических наук, доцент
Кафедра вычислительной техники
Винницкий национальный
технический университет
Хмельницкое шоссе, 95,
г. Винница, Украина, 21021
E-mail: VPSemerenko@ukr.net

1. Введение

Важной научно-технической задачей является обеспечение в различных системах передачи данных высокой надежности и достоверности с помощью помехоустойчивого кодирования.

Среди помехоустойчивых кодов наиболее часто применяются циклические коды, а среди них – мно-

жество кодов CRC. Их сфера применения: передача данных в компьютерных сетях [1], проверка записанной информации на оптических дисках [2], контроль отказов аппаратуры на базе ПЛИС [3], и многое другое.

Главное преимущество кодов CRC – быстрые вычисления и простота программно-аппаратной реализации. Однако эти неоспоримые преимущества сыграли и совсем неожиданную роль.

Специалисты помехоустойчивого кодирования уже давно решили, что все основные теоретические вопросы относительно двоичных циклических кодов, к которым принадлежат коды CRC, решены и свое внимание перенесли на совершенствование и разработку других кодовых конструкций. Ведущий в мире научный журнал по теории информации и кодированию "IEEE Transactions on Information Theory" за последние десятилетия лишь несколько раз обращался к тематике CRC [4].

Поэтому инженерам-практикам пришлось самим предлагать решения для каждой конкретной технической задачи вне связи с общей стратегией развития циклических кодов. В итоге сложилась ситуация, когда непонятно даже, что следует понимать под кодами CRC, настолько широка их нынешняя трактовка.

Поэтому давно назрела задача критического осмысления полученных результатов, дальнейшей разработки теории кодов CRC и на ее основе решения актуальных практических вопросов.

2. Анализ литературных источников и постановка проблемы

Известны две расшифровки аббревиатуры CRC: *Cyclic Redundancy Check* – циклический избыточный контроль; *Cyclic Redundancy Code* – циклический избыточный код.

Далее будет показано их отличие, сейчас отметим лишь суть CRC – установление факта отсутствия или наличия ошибок в контролируемой последовательности данных путем вычисления CRC-суммы (по первой интерпретации) или синдрома ошибки (по второй интерпретации).

Традиционный способ вычисления CRC-суммы (синдрома ошибки) путем деления полиномов в столбик известен еще с 60-х годов прошлого столетия. Без сомнения, он слишком устарел и не может быть эффективным во многих высокоскоростных приложениях. Поэтому большая группа публикаций связана с разработкой способов быстрого вычисления CRC.

Вычисление CRC быстро и эффективно реализуется при аппаратной реализации с помощью регистров сдвига с линейной обратной связью (РСЛОС). Однако в большинстве компьютеров общего назначения нет аппаратных средств для вычисления CRC, поэтому используются программные алгоритмы с большими затратами времени.

Вначале было предложено использовать специальные таблицы поиска с заранее вычисленными промежуточными результатами, что позволяет уменьшить число итераций вычисления CRC [5]. Однако для хранения указанных таблиц требуются громоздкие блоки памяти, а медленное считывание информации из них сокращает выигрыш во времени. Позже были разработаны другие способы вычисления CRC, не требующие таблиц поиска. В этом случае уменьшение числа итераций осуществляется алгоритмически, что влечет за собой использование более сложных методов вычисления CRC. В [6] приведены точные расчеты сложности вычислений при программной реализации CRC.

Главной проблемой, волнующей всех пользователей кодов CRC – оптимальный выбор порождающих полиномов. В 1993–2004 годах этим вопросом занималась группа специалистов и в итоге были опубликованы таблицы "хороших" полиномов [7, 8]. Но, эти полиномы были найдены, в основном, экспериментально, без серьезного математического обоснования. Инженеры были вынуждены на веру принять эти результаты, не понимая "таинства" выбора полиномов CRC. Поэтому авторы в [9] указывают на необходимость более тщательного исследования примитивных полиномов как основной разновидности полиномов CRC.

Следует также отметить работы, посвященные возможности исправления ошибок с помощью кодов CRC. В [10] был предложен метод исправления одиночных ошибок, а в [11] он был обобщен для исправления двойных ошибок. Исследователи также начали проверять возможность исправления пакетов ошибок [12], однако, приведенные авторами результаты экспериментов не могут заменить собой строгого математического доказательства.

3. Цель и задачи исследований

Целью данной работы является разработка теоретической основы кодов CRC (контрольной суммы CRC) на основе математического аппарата линейных последовательностных схем (ЛПС) и решение актуальных вопросов практического использования CRC.

Для достижения поставленной цели необходимо решить следующие задачи:

– показать суть кодов CRC (контрольной суммы CRC) с позиций теории ЛПС;

– показать отличия различных порождающих полиномов CRC и дать рекомендации по их оптимальному выбору;

– предложить методы параллельного вычисления CRC с помощью многоканальной ЛПС.

4. Что такое CRC?

Краткий ответ на этот вопрос дает уже расшифровка аббревиатуры "CRC". Случайно, или нет, но двойной вариант расшифровки абсолютно правильно указывает на два направления исследований CRC. Как правило, рассматривается только один вариант, либо между ними не делают различий. Однако эти различия существенны.

Начнем с интерпретации CRC как Cyclic Redundancy Check – циклического избыточного контроля. Близ-

ким синонимом, который не меняет сути, является термин "контрольная сумма".

Этот термин имеет широкую сферу использования. Если имеется некоторая входная последовательность I произвольной длины, тогда по определенному правилу вычисляется контрольная сумма Σ, представляющая собой как бы сжатое представление I. Главный критерий оптимальности правила вычислений Σ – возможные искажения последовательности I должны изменять первоначальную контрольную сумму Σ.

Если рассматривать k-разрядную последовательность I как некоторый полином $\iota(x)$ степени k–1, тогда остаток $\phi(x)$ от его деления на выбранный порождающий полином $g(x)$ степени r по правилам двоичного поля Галуа и даст нам искомую r-разрядную контрольную сумму Σ:

$$\phi(x)=\frac{\iota(x)}{g(x)},\quad GF(2). \qquad (1)$$

Но как оценить свойства такой контрольной суммы и можно ли ее использовать в серьезных практических приложениях? В следующих разделах будет дан ответ на этот вопрос сравнением $\phi(x)$ с контрольной суммой, которая используется в криптографии.

Теперь рассмотрим интерпретацию CRC как *Cyclic Redundancy Code* – циклического избыточного кода. С этой целью рассмотрим процедуру кодирования систематического циклического кода. Для кодирования полинома $\iota(x)$ степени $k-1$ его вначале следует умножить на x^r, а затем разделить на порождающий полином $g(x)$ степени r:

$$\psi(x)=\frac{x^r\iota(x)}{g(x)},\quad GF(2) \qquad (2)$$

В итоге получаем кодовый полином степени $n-1$

$$z(x)=\psi(x)\iota(x),$$

которому соответствует n-разрядное кодовое слово

$$Z=\psi_0\dots\ \psi_{r-2}\ \psi_{r-1}\ \iota_0\dots\iota_{k-2}\ \iota_{k-1}. \qquad (3)$$

В системах передачи данных в канал связи кодовое слово Z поступает, начиная со старшего разряда ι_{k-1}.

Младшие r разрядов кодового слова Z можно также рассматривать как контрольную сумму Σ. А в чем ее отличие от контрольной суммы, вычисленной по способу (1)?

В первом случае придется сравнивать вычисленную сумму Σ_r с переданной суммой Σ_s. Во втором случае свидетельством безошибочной передачи является равенство нулю вычисленной суммы Σ_r (интерпретируемой как синдром ошибки кода), и можно будет не только обнаруживать ошибки, но и их исправлять.

Для теоретического обоснования свойств контрольных сумм (1) и (2) рассмотрим методы представления циклических кодов.

5. Методы представления циклических кодов

Важнейшей задачей циклических кодов есть разработка эффективных методов их декодирования. Эта задача может иметь различные решения в зависимости от того, какой математический аппарат выбирается для представления кодов.

Еще со времени создания циклических кодов основным способом их описания остается полиномиальный. В этом случае n-разрядному кодовому слову (3) циклического (n, k)-кода ставится в соответствие кодовый полином степени (n–1) с коэффициентами из поля Галуа GF(q):

$$z(x)=z_0+z_1x+z_2x^2+\dots+z_{n-1}x^{n-1},\quad GF(q).$$

Тогда циклическому коду будет принадлежать множество полиномов $z(x)$, которые делятся без остатка на заданный порождающий полином

$$g(x)=g_0+g_1x+\dots+g_{r-1}x^{r-1}+g_rx^r,\quad GF(q) \qquad (4)$$

степени r $(r=n-k)$.

Со времени своего появления и до сих пор коды CRC ассоциируются исключительно с полиномиальной арифметикой. Все руководства рекомендуют именно такой способ вычислений, несмотря на его очевидные недостатки. Помимо трудоемкой операции деления полиномов, трудно увидеть альтернативные стратегии процедур кодирования и декодирования.

Известны и другие способы описания циклических кодов, например, матричное или через корни порождающего полинома $g(x)$ [13]. Увы, они также неудобны для практической реализации.

Перспективным способом представления циклических кодов являются автоматные модели, которые основаны на специальном классе конечных автоматов в полях Галуа – линейных последовательностных схемах (ЛПС). Будем использовать ЛПС в двоичном поле Галуа GF(2), хотя все дальнейшие рассуждения легко обобщаются и на недвоичные поля Галуа GF(q) $(q>2)$.

Согласно [14] ЛПС с l входами, m выходами и r элементами памяти в дискретные моменты времени t над полем Галуа GF(2) описывается функцией переходов

$$S(t+1)=A\times S(t)+B\times U(t),\qquad GF(2) \qquad (5)$$

и функцией выходов

$$Y(t)=C\times S(t)+D\times U(t),\qquad GF(2), \qquad (6)$$

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; $S(t)=\left|s_i\right|_r$ – слово состояния; $U(t)=\left|u_i\right|_l$ – входное слово; $Y(t)=\left|y_i\right|_m$ – выходное слово.

Размерности матриц ЛПС и параметры циклического (n, k)-кода Ω связаны через коэффициент r, который для кода равен числу контрольных разрядов кодового слова Z при систематическом кодировании $(r=n-k)$. Над полем Галуа GF(2) в ЛПС с одним входом и одним выходом могут быть использованы такие матрицы:

$$A=\begin{vmatrix}0 & 0 & 0 & \dots & g_0\\ 1 & 0 & 0 & \dots & g_1\\ 0 & 1 & 0 & \dots & g_2\\ \dots & \dots & \dots & \dots & \dots\\ 0 & 0 & 0 & 1 & g_{r-1}\end{vmatrix},\ B=\begin{vmatrix}1\\0\\0\\\dots\\0\end{vmatrix},\ C=\begin{vmatrix}0 & \dots & 0 & 1\end{vmatrix},\ D=\begin{vmatrix}0\end{vmatrix}. \quad (7)$$

Элементы последнего столбца матрицы A из (7) представляют собой коэффициенты порождающего полинома (4) кода Ω.

Простейшей аппаратной реализацией ЛПС является РСЛОС. Далее будут рассмотрены и более сложные реализации ЛПС.

На основе ЛПС можно построить автоматно-графовую и автоматно-аналитическую модели циклического (n, k)-кода [15].

Поскольку ЛПС является конечным автоматом, поэтому в качестве автоматно-графовой модели можно выбрать граф переходов-выходов этого автомата. Для r-мерной ЛПС над полем GF(2) такой граф переходов-выходов 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 нулевые и единичные дуги из множества дуг E_{FA} показывают направления переходов между внутренними состояниями. Отдельные вершины графа G_{FA} с помощью нулевых дуг объединяются в нулевые циклы (НЦ).

Вершинам графа G_{FA} соответствуют внутренние состояния ЛПС. Последовательность слов внутренних состояний ЛПС также образуют НЦ. Совокупность НЦ из слов состояний имеет такую же структуру, что и совокупность НЦ из вершин графа G_{FA}.

Если автоматно-графовая модель удобна для наглядного представления сути процедуры декодирования, то автоматно-аналитическая модель предназначена для аппаратной и программной реализации кодирования и декодирования циклических кодов.

ПРИМЕР 1. Для циклического (7,4)-кода с порождающим полиномом $g(x)=1+x+x^3$ с помощью автоматно-аналитического метода выполнить декодирование кодового слова

$$Z=\begin{bmatrix}0 & 1 & 0 & 0 & 0 & 1 & 1\end{bmatrix}.$$

Поставленная задача эквивалентна вычислению CRC.

Заданному полиному $g(x)$ соответствует такие матрицы ЛПС:

$$A=\begin{vmatrix}0 & 0 & 1\\ 1 & 0 & 1\\ 0 & 1 & 0\end{vmatrix},\quad B=\begin{vmatrix}1\\0\\0\end{vmatrix},\quad C=\begin{vmatrix}0 & 0 & 1\end{vmatrix},\quad D=\begin{vmatrix}0\end{vmatrix}.$$

Операциям деления полиномов в полиномиальной арифметике соответствуют операции рекурсивного вычисления состояний ЛПС по формуле (5). Покажем подробно первые действия.

Для вычислений по формуле (5) сформируем компоненты входного слова ЛПС с учетом того, что в канал связи кодовое слово Z поступает начиная со старших разрядов:

$$S(1)=A\times S(0)+B\times U(0)=$$
$$=\begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}=\begin{bmatrix}1\\0\\0\end{bmatrix},$$

. . . . . .

$$S(6)=A\times S(5)+B\times U(5)=$$
$$=\begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\0\\1\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}1\end{bmatrix}=\begin{bmatrix}0\\1\\0\end{bmatrix}+\begin{bmatrix}1\\1\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix},$$

$$S(7)=A\times S(6)+B\times U(6)=$$
$$=\begin{bmatrix}0 & 0 & 1\\ 1 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}1\\0\\0\end{bmatrix}\begin{bmatrix}0\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}+\begin{bmatrix}0\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\0\end{bmatrix}.$$

$U(0)=z_6=1;\ U(1)=z_5=1;\ \dots;$

$U(5)=z_1=1;\ U(6)=z_0=0.$

Начальное состояние $S(0)$ ЛПС выбираем нулевым. Дальнейшие состояния ЛПС вычисляются согласно (5) следующим образом.

Последнее состояние $S(7)$ ЛПС и представляет собой синдром ошибки, т. е. искомое CRC.

$\perp$

Приведенный пример продемонстрировал лишь основную концепцию вычисления CRC на основе теории ЛПС. Программно-аппаратная реализация осуществляется проще и быстрее. Во-первых, не нужно формировать и сохранять характеристические матрицы ЛПС, достаточно иметь лишь полученное из полинома $g(x)$ порождающее слово. Для данного примера оно равно последнему столбцу матрицы A:

$$\begin{bmatrix}1\\1\\0\end{bmatrix}.$$

Во-вторых, операции умножения матриц в поле Галуа заменяется операциями сдвига и сложения.

Покажем реализацию предыдущего примера на основе операций сдвига и сложения (рис. 1). Слова состояний ЛПС записываются в виде r-разрядных строк. В (i+1)-ю строку переписываются сдвинутые на одну позицию вправо разряды i-й строки, а на место самого левого разряда записывается (n–i–1)-й разряд кодового слова Z. Если последний разряд i-й строки был нулевым, тогда при сдвиге вправо он теряется, а если он был единичным, тогда выполняется операция коррекции: (i+1)-я строка суммируется по модулю 2 с транспонированным порождающим словом.

Приведенный способ вычисления CRC применяется при последовательном поступлении кодового слова Z. Как и при использовании полиномиальной арифметики, количество итераций вычисления CRC равно разрядности Z. В среднем, половина итераций очень короткие: только поразрядный сдвиг слова состояния ЛПС.

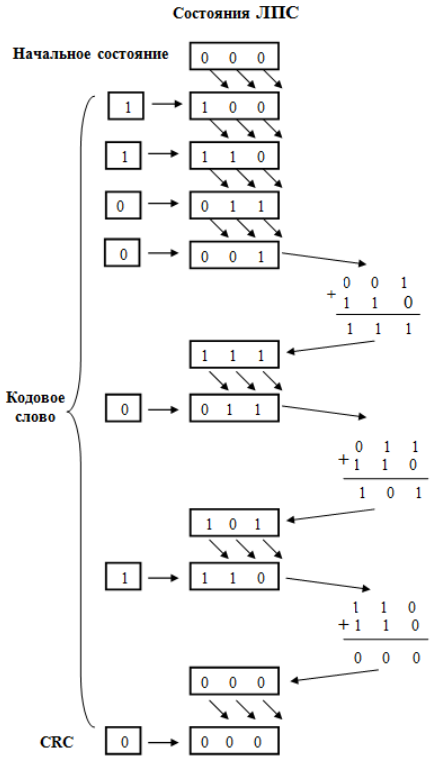

Рис. 1. Пример последовательного вычисления CRC на основе операций сдвига и сложения для порождающего полинома $g(x)=1+x+x^3$

6. CRC как контрольная сумма

Самые строгие требования к контрольным суммам предъявляются в криптографии, где они именуются хеш-функциями. Различают блоковые и потоковые хеш-функции. Рассмотрим CRC как потоковую хеш-функцию с позиций теории ЛПС.

Любая хэш-функция должна удовлетворять таким основным требованиям:

– быть применимой к блоку данных произвольной длины;

– давать на выходе значение фиксированной длины;

– быть удобной при программной и аппаратной реализации;

– обеспечивать высокую сопротивляемость коллизиям;

– быть односторонней, т. е. обеспечивать практическую невозможность вычисления соответствующей исходной последовательности по известному значению хэш-функции.

Для хэш-функций на основе теории ЛПС первые три требования удовлетворяются из самого определения ЛПС. Проанализируем подробнее остальные требования.

Для 2^w-1 различных ненулевых входных последовательностей длины w r-мерная ЛПС может сформировать 2^r-1 различных хэш-функций длины n. Каждой хэш-функции, таким образом, соответствует $2^{w-r}-1$ одинаковых входных последовательностей. Отсюда, при условии равновероятности входных последовательностей, вероятность возникновения коллизии равна

$$p_0=\frac{2^{w-r}-1}{2^w-1}\approx 2^{-r}. \quad (8)$$

Относительно чувствительности хэш-функции к изменению входного текста известны следующие факты. Если входные последовательности I_i и I_j различаются в одном или двух разрядах, то всегда будут отличны соответствующие им хэш-функции. Если же входные последовательности I_i и I_j различаются более чем двух разрядах, тогда вероятность того, что соответствующие им хэш-функции будут различны, равна

$$p_1=1-p_0=1-2^{-r}. \quad (9)$$

Как видно из формул (8) и (9), при больших значениях r при отсутствии преднамеренного искажения входного сообщения вероятность возникновения коллизий для рассматриваемых хэш-функций на практике пренебрежимо мала.

По определению, односторонняя функция – это эффективно вычислимая функция, для задачи инвертирования которой не существует быстрых алгоритмов. В [16, 17] доказано, что криптостойкие псевдослучайные генераторы существуют тогда и только тогда, когда существуют односторонние функции. Известно также [14], что r-мерная ЛПС, которая реализует формулы (5) и (6) будет являться генератором псевдослучайных чисел, если соответствующий ей порождающий полином (4) является примитивным. Такая ЛПС будет генерировать последовательность символов периода 2^r-1 (М-последовательность), которая при больших r практически неотличима от случайной последовательности. Следовательно, функция, реализующая формулы (5) и (6), для примитивного порождающего полинома (4) является односторонней функцией.

Таким образом, CRC будет достаточно эффективной хеш-функцией, т. е. контрольной суммой, при наличии таких условий: её порождающий полином должен быть примитивным и иметь относительно большую степень ($r\geq 16$), а длина входной последовательности не должна превышать 2^r-1. Таким требованиям в максимальной степени отвечает код Хэмминга (подчеркнем, что в данной статье рассматриваются только циклические, а не обычные коды Хэмминга).

Кстати, в технической диагностике [18] параметры контрольной суммы, называемой сигнатурой, выбраны исходя из указанных требований.

Безусловно, криптографические блоковые хеш-функции (SHA, MDA5 и другие) более надежны, но платой за это служит значительное усложнение их вычислений. Отметим, что при разрядности r=128 и r=256 блоковые хеш-функции уже заменили CRC.

7. CRC как циклический код

Как и любой другой линейный блоковый код, циклический (n, k)-код имеет строго определенную структуру: длина блока (т. е. кода) n, длина информационной части k и количество τ исправляемых ошибок связаны определенными математическими соотношениями [13]. Зачастую параметры известных кодов не подходят для конкретных систем передачи данных. Для кодов CRC при степени порождающего полинома $r\geq 16$ длина w информационного сооб-

щения значительно меньше допустимой размерности кода: $w << k$. В таких случаях CRC код используется как укороченный циклический код.

Существуют различные подклассы циклических кодов: Хэмминга, Файра, БЧХ и другие. Однако, как это не парадоксально на первый взгляд, CRC не является отдельным подклассом в классе циклических кодов.

Объясняется такая ситуация очень просто. Каждый подкласс циклических кодов имеет строгие границы, обусловленные используемыми порождающими полиномами. А для кодов CRC используют различные порождающие полиномы. Во множество кодов CRC включили те коды, которые нашли свое практическое применение для задач обнаружения ошибок вне зависимости их принадлежности к определенному теоретическому подклассу. Именно это обстоятельство и породило многочисленные проблемы и недоразумения, связанные с кодами CRC. Отсюда берет корни задача выбора оптимального полинома или оптимального кода CRC.

Нельзя абстрактно рассуждать о кодах CRC без анализа их происхождения. Определяющей характеристикой любого циклического кода является его порождающий полином, поэтому с него и начнем.

Наиболее часто порождающий полином CRC кода имеет вид:

$$g(x) = (1+x)p(x), \quad GF(2), \qquad (10)$$

где $p(x)$ – примитивный полином степени m.

Первый, кто предложил такие полиномы, был Абрамсон [19], его именем коды с таким полиномом и названы.

Вторым по частоте использования является примитивный порождающий полином. В этом случае мы имеем дело с циклическим кодом Хэмминга. Поскольку полином $p(x)$ в (10) чаще всего также является кодом Хэмминга, поэтому и весь полином (10) иногда называют расширенным кодом Хэмминга.

Встречаются также непримитивные неприводимые порождающие полиномы, которые используются, например в квадратично-вычетных (КВ) кодах.

Очень наглядно свойства порождающего полинома $g(x)$ видны при автоматном представлении циклического кода. В [15] доказано, что возможности циклических кодов по обнаружению и исправлению ошибок однозначно определяются структурой его графа G_{FA} и зависят как от количества НЦ, так и от их взаимного расположения. Основной критерий исправления случайных ошибок кратности τ следующий: число уровней графа G_{FA} должно быть равно не менее τ, причем на τ-ом уровне общее количество вершин должно быть равно числу сочетаний из n по τ.

Самую простую графовую модель имеет циклический код Хэмминга: тривиальный НЦ (ТНЦ), состоящий из одной вершины v_0, и основной НЦ (ОНЦ), содержащий остальные $(2^r - 1)$ вершин (рис. 2). Здесь и на последующих рисунках нулевые дуги показаны сплошными линиями, а единичные – точечными. Единичные дуги внутри НЦ не влияют корректирующую способность кода, поэтому не будем их показывать. Наличие только одного полноценного НЦ ограничивает код Хэмминга исправлением только одиночных ошибок.

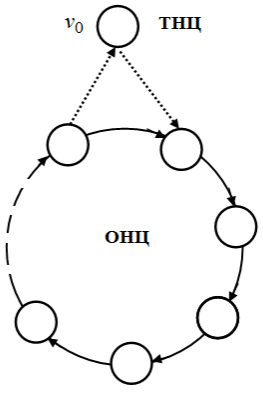

Рис. 2. Упрощенная графовая модель циклического кода Хэмминга

Более сложную графовую модель имеет код Абрамсона на основе порождающего полинома (10): имеется одна вершина v_0 (ТНЦ) на нулевом уровне, ОНЦ длины n на первом уровне, периферийный НЦ (ПНЦ) длины n на втором уровне и одна вершина v_2 на третьем уровне (рис. 3). В итоге код может исправлять не только одиночные ошибки, но и смежные двойные ошибки, т. е. пакеты ошибок длины 2. Но самое полезное свойство такого кода состоит в его способности обнаружения большого количества разнообразных ошибок.

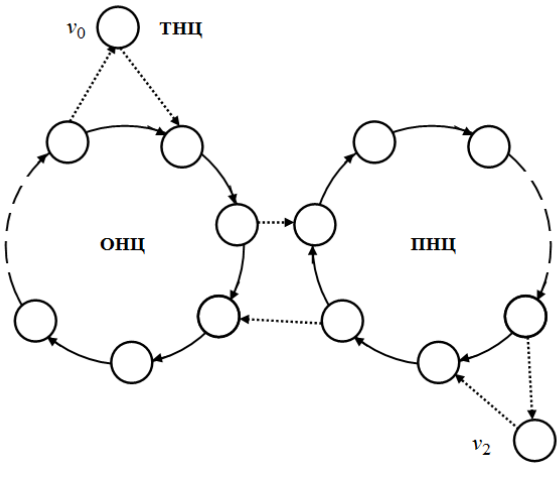

Рис. 3. Упрощенная графовая модель кода Абрамсона

Ошибка будет обнаружена, если ее вычисленный по формуле (5) синдром $S_{err}(t)$ будет ненулевым. В автоматно-графовой модели вершиной ошибки v_{err}, которая соответствует синдрому $S_{err}(t)$, должна быть любая вершина, кроме v_0, иначе ошибка не будет обнаружена. И еще одно важное свойство графовой модели – по ней можно проследить путь ошибки. Все одиночные ошибки попадают по одиночным дугам из ТНЦ в ОНЦ, двойные – в ПНЦ. Далее путь начинает циклически повторяться: тройные ошибки снова попадают в ОНЦ или в вершину v_2, 4-ные – в ТНЦ или в ПНЦ, 5-ные – в ОНЦ и т. д. В итоге четные ошибки могут попадать в ТНЦ и будут неотличимы от исправного состояния. Зато все нечетные ошибки в ТНЦ не попадут и будут обнаружены, хотя они и выходят за пределы минимального кодового расстояния $d_{min} = 4$ для этого кода.

В теории ЛПС [14] доказано, что увеличение числа сомножителей в порождающем полиноме циклического кода всегда увеличивает число НЦ в составе его графовой модели с одновременным уменьшением их длины.

В частности, умножение полинома $g(x)$ на множитель $(1+x)$ увеличивает в исходном графе G_{FA} вдвое число НЦ и уменьшает вдвое их длину. В итоге увеличивается на единицу число контрольных разрядов $(r=r+1)$ и уменьшается на единицу число информационных разрядов $(k=k-1)$, т. е. происходит операция сужения кода.

Если при переходе от кода Хэмминга к коду Абрамсона возрастает обнаруживающая способность кода, то при дальнейшем увеличении числа НЦ будет возрастать корректирующая способность кода.

Именно это свойство и положено в основу построения различных подклассов циклических кодов. Для каждого синдрома исправляемой ошибки должна быть соответствующая вершина в графе G_{FA}, причем для каждой группы ошибок, представляющих класс смежности в таблице декодирования [13], должен быть отдельный НЦ. И чем больше будет НЦ в графе G_{FA}, тем более высокую корректирующую способность будет иметь соответствующий циклический код. А увеличение числа НЦ прямо пропорционально уменьшает их длину.

Повышение корректирующей способности означает увеличение кодового расстояния d_{min}, следовательно, будет расти и кратность обнаруживаемых ошибок в пределах d_{min}. Однако полное обнаружение ошибок определенной кратности за пределами d_{min} теория не позволяет, поэтому превзойти обнаруживающую способность кода Абрамсона не удастся. Кроме этого, уменьшение длины НЦ (т. е. длины кода) требует уменьшения и длины w контролируемого информационного сообщения, что нежелательно.

В табл. 1 показано соответствие порождающего полинома графовой модели кода, а в табл. 2 приведены краткие сведения об обнаруживающих и корректирующих способностях наиболее известных подклассов двоичных циклических кодов.

Таблица 1

Порождающие полиномы и графовые модели кода

Структура НЦ графовой модели	Тип порождающего полинома	Подклассы циклических кодов
Один НЦ длины $n=2^r-1$	примитивный	Код Хэмминга
Два НЦ длины $n=2^{r-1}-1$	$g(x)=(1+x)p(x)$, $p(x)$ – примитивный	Код Абрамсона
Более двух НЦ длины $n<2^{r-1}-1$	непримитивный неприводимый, либо приводимый (разложимый), кроме вида (10)	Коды КВ, БЧХ

Автоматно-графовая модель циклических годов очень наглядно демонстрирует фундаментальное свойство этих кодов: если код прекрасно работает с одним типом ошибок, он будет плохо работать с другим типом ошибок. Поэтому вызывает, по меньшей мере, сомнение в целесообразности выбора такого критерия, когда лучшими кодами CRC считаются коды с наибольшими возможностями по исправлению ошибок (т. е. с большим значением d_{min}). Не существует универсального подкласса циклического кода для всех типов ошибок, и не нужно "заставлять" определенный код выполнять несвойственные ему функции.

Таблица 2

Обнаруживающая и корректирующая способность циклических (n,k)-кодов

Подклассы циклических кодов	Обнаружение ошибок		Исправление ошибок	
	случайные ошибки кратности τ	пакеты ошибок длины b	случайные ошибки кратности τ	пакеты ошибок длины b
Код Хэмминга	$\tau=1, \tau=2$	$b=2$	$\tau=1$	–
Код Абрамсона типа $g(x)=(1+x)p(x)$	$\tau=1, \tau=2, \tau=2i+1, i\geq 1$	$b>2$	$\tau=1$	$b=2$
Коды КВ, БЧХ	$\tau\leq(d_{min}-1)$	$b\leq n-k$	$\tau\leq\frac{d_{min}-1}{2}$	$b\leq\frac{n-k}{2}$

8. Рекомендации по выбору порождающих полиномов для CRC

Практически любая статья [7, 8], посвященная CRC, содержит таблицу порождающих полиномов для CRC и комментарий по их свойствам. Обычно авторы сетуют на отсутствие подробных руководств по выбору "хороших" полиномов среди ранее опубликованных и предлагают новые. При этом они используют довольно субъективные критерии по оценке новых полиномов, так что читателям-инженерам с каждым годом приходится тратить все больше времени на анализ увеличивающегося списка полиномов. И даже довольно авторитетные специалисты отмечают, что "выбор полиномов является очень сложной проблемой" [20].

Кратко рассмотрим самую цитируемую работу по выбору порождающих полиномов для кодов CRC [7]. Эта статья ориентирована на инженеров встроенных сетей, и, наверное, по этой причине написана без единой формулы. Однако только математика может дать четкие и ясные ответы на все вопросы в этой области техники. Иначе все словесные доводы в конечном итоге сводятся к очередной попытке предложить "хорошие" полиномы вместо "плохих". Но не бывает плохих полиномов, плохим и неоптимальным может быть только их выбор.

В качестве примера рассмотрим несколько из числа "самых лучших", по мнению авторов [7], CRC полиномов степени 8 (табл. 3). В этой таблице информация в первых трех столбцах взята из [7], т. е. для каждого полинома дан только один существенный параметр – величина минимального кодового расстояния d_{min}. Однако для правильного выбора полинома определяющую роль играют длина кода n и тип порождающего полинома (приводим в последних двух столбцах), и именно такой информации недостает инженерам.

Полином 0xA6 определяет код Хэмминга, который имеет минимальные возможности по обнаружению и исправлению ошибок, зато позволяет работать с самыми длинными информационными сообщениями. Полином 0x97 определяет код Абрамсона, который обладает наилучшими обнаруживающими способностями, но длина информационного сообщения уменьшается вдвое. Наконец, полином 0x9C принадлежит КВ коду,

наименьшая длина которого компенсируется возможностью исправления одиночных и двойных ошибок.

Таким образом, вывод о том, какой из рассмотренных полиномов является лучшим, можно сделать только из анализа поставленной задачи поиска ошибок в конкретных технических условиях.

Таблица 3

Характеристики некоторых порождающих полиномов

16-ричное обозначение полинома	Порождающий полином CRC-кода	Минимальное кодовое расстояние d_{min}	Длина кода n	Тип порождающего полинома
0xA6	$1+x^2+x^5+x^6+x^8$	3	255	неприводимый, примитивный
0x97	$1+x^3+x^5+x^6+x^7+x^8=$ $=(1+x)(1+x^5+x^7)$	4	127	приводимый, типа (10)
0x9C	$1+x^3+x^4+x^5+x^8$	5	17	неприводимый, непримитивный

Поэтому рекомендуется следующая процедура выбора порождающего полинома кода CRC.

1. С учетом характеристик канала связи выбирается наиболее характерная для него модель ошибок и требуемые действия (только обнаружение или также исправление ошибок).

2. Выбор подкласса циклического кода. Для обнаружения максимального количества ошибок различных типов наилучшим выбором будет код Абрамсона с полиномом типа (10). Если достаточным будет нахождение только одиночных или двойных ошибок, тогда можно выбрать код Хэмминга. Для исправления ошибок необходимо выбирать более мощный код из семейства кодов БЧХ.

3. Для заданной длины w информационного сообщения вычислить необходимую степень r порождающего полинома кода. При этом должно выполняться неравенства:

$2^r - r \geq w+1$ для кода Хэмминга;

$2^{r-1} - r \geq w+1$ для кода Абрамсона типа (10).

О важности этого шага часто забывают на практике. Если теоретическая длина n $(n=k+r)$ кода будет меньше фактической длины n_w $(n_w = w+r)$, тогда код не будет корректно выполнять свою роль по поиску ошибок.

Единственная рекомендация – выбирать значение r достаточно большой величины. При отсутствии каких-либо других ограничений (например, при аппаратной реализации) оптимальным значением будет $r \geq 16$. Эта рекомендация следует также из формул (8) и (9).

4. Выбрать порождающий полином степени r выбранного в п. 2 кода. Для этого можно воспользоваться различными источниками, содержащими таблицы примитивных полиномов. Можно также самостоятельно вычислить полином с требуемыми свойствами. Например, для получения полинома степени r классического кода CRC достаточно умножить примитивный полином степени $r-1$ на $(1+x)$.

Часто авторы статей приводят найденные экспериментально какие-то дополнительные аргументы об особенностях того или иного полинома. На самом деле правильный математический выбор полинома гарантирует его нормальную обнаруживающую способность. Имеется только одно существенное исключение: в циклическом (n, k)-коде с порождающим полиномом $g(x)$ не обнаруживаются случайные ошибки, расположенные в циклическом интервале длины $(n-k+1)$ в тех позициях, которые совпадают с ненулевыми разрядами полинома $g(x)$. В итоге не будет обнаружено n конфигураций ошибок кратности $\tau = n_g$ (n_g – число ненулевых разрядов полинома $g(x)$).

Не будем приводить перечень порождающих полиномов для кодов CRC, их опубликовано достаточное количество [7]. Необходимо лишь дополнить имеющиеся таблицы сведениями о типе полинома и длине кода. Тогда оптимальный выбор порождающего полинома для конкретного приложения станет тривиальной задачей.

В литературе редко акцентируется внимание на свойстве примитивности полинома, разве только для небольших значений степеней [14]. И решить такую задачу в рамках традиционной полиномиальной модели циклических кодов довольно затруднительно. В этом случае очень поможет автоматное представление циклических кодов. Несложная программа по приведенному в [15] алгоритму позволит одновременно вычислить длину кода и построить его нулевые циклы. Если для полинома $g(x)$ степени r будет получен единственный НЦ максимальной длины $n = 2^r - 1$, следовательно, этот полином будет примитивным.

9. Исправление ошибок в CRC кодах

Как уже отмечалось, каждый подкласс циклических кодов ориентирован на выполнение какой-то одной основной функции. Но имеется функция, с которой успешно справляются все коды – исправление одиночных ошибок.

На практике для кодов CRC это свойство не используется: при обнаружении ошибки в пакете данных осуществляется его повторная передача. Такая стратегия эффективна только в каналах с большим уровнем шумов.

В [21] приведена статистика ошибок в сетях ATM в условиях эксплуатации с использованием волоконно-оптических линий связи: 65 % составляют одиночные ошибки, 23 % составляет двух- и трехкратные ошибки, 12 % – пакетные ошибки кратности от 5 и более. Для таких каналов исправление одиночных ошибок может быть оправданным.

Следует отметить еще одну причину, по которой исправление одиночной ошибки в коде CRC (как для кода Хэмминга, так и для кода Абрамсона) может быть неэффективным. Дело в том, что коды CRC с порождающим полиномом большой степени являются сильно укороченными циклическими кодами. Для задачи обнаружения ошибок это обстоятельство не играет никакой роли: процедура вычисления синдрома ошибки (контрольной суммы) займет n_w тактов времени

(т. е. закончится к окончанию периода времени передачи). Однако длительность процедуры исправления одиночных ошибок известными методами (Меггитта, Касами [13]) пропорциональна теоретической длине n кода. Для сетей Ethernet с форматом кадра IEEE 802.3 указанные величины составляют: n_w не более 1200 бит, а $n = 2^{32} - 1$.

С помощью автоматной модели циклических кодов эта проблема успешно решается. Для этого достаточно перейти к использованию специальной ЛПС, которая функционирует по обратной шкале времени (от "настоящего" к "прошлому") [22]. Более подробно этот вопрос изложен в [23].

10. Методы ускорения вычисления CRC

Одной из актуальных инженерных задач, связанных с CRC кодами, является ускорение вычисления контрольных сумм. Эта задача имеет свои особенности при аппаратной и программной реализации, а также зависит от способа передачи данных.

Со времени своего появления CRC коды были ориентированы на последовательную передачу данных. Каждый шаг вычисления контрольной суммы осуществлялся с приходом очередного входного сигнала. Это достаточно наглядно видно при аппаратной реализации вычислителя CRC в виде РСЛОС.

С широким внедрением многоканальной связи (как беспроводной, так и кабельной) данные стали поступать параллельно: байтами и машинными словами по несколько байт. Чтобы не задерживать передачу, традиционный вычислитель CRC должен работать во много раз быстрее.

Однако на практике, особенно при программном вычислении CRC, такой вариант очень сложно реализовать. Поэтому и появились различные методы ускорения вычисления контрольных сумм [5, 6]. Они требуют либо громоздкой памяти для хранения промежуточных данных, либо сложных вычислений.

Наиболее подходящей математической моделью для представления высокоскоростного декодирования циклического кода с порождающим полиномом степени ρ может служить ρ-канальный $(\rho \le r)$ аналог обычной ЛПС [14], имеющей ρ-входов, ρ-выходов и такие характеристические матрицы $A^{(\rho)}$ и $B^{(\rho)}$:

$$A_{(\rho)} = \left[A^{\rho} \right];\ B_{(\rho)} = \left[A^{\rho-1}B \quad \dots \quad AB \quad B \right].$$

Например, если одноканальная r-мерная ЛПС имеет характеристические матрицы A и B вида (7), тогда ее ρ-канальный аналог имеет следующие характеристические матрицы:

$$A^{(\rho)} = \left[A^{\rho} \right];\ B^{(\rho)} = \begin{bmatrix} 0 & \dots & 0 & 1 \\ 0 & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots \\ 1 & \dots & 0 & 0 \end{bmatrix}. \qquad (11)$$

Пусть имеется n_w-разрядное кодовое слово Z, состоящее из ρ-битовых символов. Тогда CRC суммой будет код состояния $S(n_w)$, в которое перейдет ρ-канальная ЛПС после подачи на ее ρ входов слова Z по рекурсивной формуле

$$S(j+1) = A^r \times S(j) + z_j, \quad GF(2), \quad z_j \in Z, \quad j = 1 \div n_w. \qquad (12)$$

ρ-канальную ЛПС можно использовать также и для ускорения вычислений для однобитовых символов кодового слова Z. В этом случае n_w-разрядное слово Z необходимо разбить на υ подслов $Z_1,\dots,Z_\upsilon$ длины ρ $\left(\upsilon = \left\lceil \frac{n_w}{\rho} \right\rceil\right)$, $\lceil \bullet \rceil$ означает округление до целого в большую сторону). Неполное крайнее правое подслово Z_1 следует дополнить справа нулями. Далее каждое подслово Z_i параллельно подается на ρ входов ЛПС.

ПРИМЕР 2. Для циклического (7,4)-кода с порождающим полиномом $g(x) = 1 + x + x^3$ вычислить CRC сумму с использованием трехканальной ЛПС для кодового слова $Z = \left[0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 1 \right]$.

Подготовим вначале матрицу A^3 и матрицу $B_{(3)}$ согласно (15):

$$A^3 = A \times A \times A = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix};$$

$$B_{(3)} = \left[A^2B \quad AB \quad B \right] = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

Далее слово *Z*, дополним справа нулями до достижения длины 9 (3×3) и разделим его на три подслова:

$$Z_3 = \left[0 \quad 1 \quad 0 \right];\ Z_2 = \left[0 \quad 0 \quad 1 \right];\ Z_1 = \left[1 \quad 0 \quad 0 \right].$$

Теперь последовательно определим состояния трехканальной ЛПС:

$$S''(1) = A^{3_R} \times S''(0) + B \times Z_1 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix};$$

$$S''(2) = A^{3_R} \times S''(1) + B \times Z_2 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix};$$

$$S''(3) = A^{3_R} \times S''(2) + B \times Z_3 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

В итоге получим ту же нулевую CRC сумму, что и в примере 1.

Таким образом, можно параллельно вычислять CRC при любом значении степени порождающего полинома кода $(\rho = 16, \dots, 32, \dots 64, \dots)$ при линейном увеличении сложности вычислений. Однако следует пом-

нить, у матрицы A^ρ более сложная структура и ее умножение на слова состояний по формуле (12) нельзя заменить на простые операции сдвига, как это имеет место для матрицы A.

Рассмотренный способ параллельного вычисления синдрома CRC легко реализуется программно, и еще проще аппаратно. В [14] изложен способ синтеза аппаратной схемы ЛПС по ее заданным характеристическим матрицам (7) или (11).

11. Выводы

В работе проанализированы две интерпретации CRC: как контрольной суммы и как циклического кода. На основе теории ЛПС было доказано, что в первом случае CRC находит ошибки в заданном информационном сообщении аналогично криптографической хеш-функции, а во втором случае – по правилам декодирования укороченного циклического кода.

С позиций требований к контрольной сумме лучшие параметры имеет циклический код Хэмминга. Со второй задачей наилучшим образом справляются коды Абрамсона с порождающим полиномом вида (10). Поэтому, только эти типы кодов и можно отнести к кодам CPC.

Математически обоснованное и наглядное представление о корректирующей и обнаруживающей способности любого циклического кода дает его автоматная модель. Достаточно лишь получить структуру нулевых циклов, чтобы иметь достоверную информацию о возможностях этого кода.

Тогда становятся понятными критерии и процедура выбора оптимального кода. Если поставлена задача обнаружения ошибок следует выбирать указанные коды CPC, для задач исправления многократных ошибок имеются свои подклассы кодов. Рекомендуемая последовательность выбора полинома кода CRC разбивает сложившийся миф о сложности выбора таких полиномов.

В работе также решена еще одна важная практическая задача – ускорение вычисления CRC при параллельной передаче данных байтами и словами. Если известный способ [6] позволяет лишь в 2 раза уменьшить число итераций при вычислении CRC, то с помощью ρ-канальной ЛПС степень сокращения итераций составит ρ для произвольного полинома степени r.

Литература

1. Столлингс, В. Компьютерные системы передачи данных [Текст] / В. Столлингс; изд. 6-е; пер. с англ. – М., Издательский дом «Вильямс», 2002. – 928 с.
2. Costello, D. J. Applications of Error-Control Coding [Text] / D. J. Costello, Jr., J. Hagenauer, H. Imai, S. B. Wicker // IEEE Transactions on Information Theory. – 1998. – Vol. 44, Issue 6, – P. 2531–2560. doi: 10.1109/18.720548
3. Cyclic Redundancy Check (CRC) in Stratix Series FPGAs [Electronic resource] – Published 1995–2015. – Available at: https://www.altera.com/products/general/devices/stratix-fpgas/about/crc.html
4. Kazakov, P. Fast Calculation on the Number of Minimum Weight Words of CRC Codes [Text] / P. Kazakov // IEEE Transactions on Information Theory. – 2001. –Vol. 49, Issue 1. – P. 1190–1195. doi: 10.1109/18.915680
5. Sarwate, D. V. Computation of Cyclic Redundancy Checks via Table-Lookup [Text] / D. V. Sarwate // Communications of the ACM. – 1988. – Vol. 31, Issue 8. – P. 1008–1013. doi: 10.1145/63030.63037
6. Nguyen, G. D. Fast CRCs [Text] / G. D. Nguyen // IEEE Transactions on Computers. – 2009. –Vol. 58, Issue 10. – P. 1321–1331. doi: 10.1109/tc.2009.83
7. Koopman, P. Cyclic Redundancy Code (CRC) Polynomial Selection For Embedded Networks [Text] / P. Koopman, T. Chakravarty // The International Conference on Dependable Systems and Networks (DSN-2004), 2004. – P. 1–10. doi: 10.1109/dsn.2004.1311885
8. Baicheva, T. S. Determination of the best CRC codes with up to 10-bit redundancy [Text] / T. S. Baicheva // IEEE Transactions on Communications. – 2008. – Vol. 56, Issue 8. – P. 1214–1220. doi: 10.1109/tcomm.2008.070033
9. Ahmad, A. Selection of Polynomials for Cyclic Redundancy Check for the use of High Speed Embedded – An Algorithmic Procedure [Text] / A. Ahmad, L. Hayat // IEEE Trans. on Computers. – 2011. –Vol. 60, Issue 1. – P. 16–20.
10. McDaniel, B. An algorithm for error correcting cyclic redundancy checks [Text] / B. McDaniel. – C/C++ Users Journal, 2003. – P. 6.
11. Babaie S. Double bits error correction using CRC method [Text] / S. Babaie, A. K. Zadeh, S. H. Es-hagi, N. J. Navimipour // 2009 Fifth International Conference on Semantics, Knowledge and Grid. – 2009. – Vol. 5. – P. 254–257. doi: 10.1109/skg.2009.77
12. Mandel, T. Selected CRC Polynomials Can Correct Errors and Thus Reduce Retransmission [Text] / T. Mandel, J. Mache // WITS (DCOSS) 2009.
13. Блейхут, Р. Теория и практика кодов, исправляющих ошибки [Текст] / Р. Блейхут; пер. с англ. – М.: Мир, 1986. – 576 с.
14. Гилл, А. Линейные последовательностные машины [Текст] / А. Гилл; пер. с англ. – М.: Наука, 1974. – 288 с.
15. Семеренко, В. П. Оценка корректирующей способности циклических кодов на основе автоматных моделей [Текст] / В. П. Семеренко // Східно-Європейський журнал передових технологій. – 2015. – Т. 2, № 9 (74). – С. 16–24. doi: 10.15587/1729-4061.2015.39947
16. Impagliazzo, R. Pseudo-random generation from one-way functions [Text] / R. Impagliazzo, L. Levin, M. Luby // Proceedings of the twenty-first annual ACM symposium on Theory of computing - STOC '89, 1989. – P. 12–24. doi: 10.1145/73007.73009
17. Hastad, J. Pseudo-random generators under uniform assumptions [Text] / J. Hastad // Proceedings of the twenty-second annual ACM symposium on Theory of computing - STOC '90, 1990. – P. 395–404. doi: 10.1145/100216.100270
18. Ярмольник, В. Н. Контроль и диагностика цифровых узлов ЭВМ [Текст] / В. Н. Ярмольник. – Минск: Наука и техника, 1988. – 240 с.

19. Abramson, N. M. A class of Systematic Codes for Non-Independent Errors [Text] / N. M. Abramson // IEEE Transactions on Information Theory. – 1959. – Vol. 5, Issue 4. – P. 150–157. doi: 10.1109/tit.1959.1057524
20. Lin, S. Error-Control Coding: Fundamentals and Applications [Text] / S. Lin, D. J. Costello; 2nd. ed. – Upper Saddle River, NJ: Prentice-Hall, 2004.
21. Богданов, В. Н. Защита от ошибок в сетях АТМ [Текст] / В. Н. Богданов, П. С. Вихлянцев, М. В. Симонов // ИНФОРМОСТ. – 2002. – № 3 – С. 20–24.
22. Семеренко, В. П. Темпоральные модели параллельных вычислений [Текст] / В. П. Семеренко // Austrian Journal of Technical and Natural Sciences. – 2014. – Vol. 1. – P. 13–25.
23. Семеренко, В. П. Параллельное декодирование укороченных циклических кодов [Текст] / В. П. Семеренко // Оптико-электронные информационно-энергетические технологии. – 2012. – № 1. – С. 30–41.

Мобільні пристрої, такі як мобільні телефони, стають щороку більш популярними. Використання їх, однак, тягне за собою певний недолік, а саме укладення договору із оператором стільникового зв'язку, де користувач зобов'язується працювати з пристроєм відповідно до заданих правил. Ідеєю є одночасне зниження витрат на використання мобільних телефонів та досягнення повної незалежності від оператора мобільного зв'язку

Ключові слова: мобільні пристрої, мобільні мережі, мережа датчиків, само організаційні протоколи, передача даних

Мобильные устройства, такие как мобильные телефоны, становятся ежегодно более популярными. Использование их, однако, влечет за собой определенный недостаток, а именно заключение договора с оператором сотовой связи, где пользователь обязуется работать с устройством в соответствии с заданными правилами. Идеей является одновременное снижение затрат на использование мобильных телефонов и достижения полной независимости от оператора мобильной связи

Ключевые слова: мобильные устройства, мобильные сети, сеть датчиков, самоорганизационные протоколы, передача данных

UDC 004.72
DOI: 10.15587/1729-4061.2015.47697

THE APPLICATION OF SENSOR NETWORKS WITHIN MOBILE CELL NETWORKS

V. Mosorov
Doctor of Technical Science*
E-mail: volodymyr.mosorov@p.lodz.pl
S. Biedron
Postgraduate student*
E-mail: SBiedron@wpia.uni.lodz.pl
T. Panskyi
Postgraduate student*
E-mail: panskyy@gmail.com
*Institute of Applied Computer Science
Lodz University of Technology
Stefanowskiego str., 18/22, Lodz, Poland, 90-924

1. Introduction

Rapid development of the mobile network infrastructure, easy access to mobile phones, and operators vying for customers by offering them more and more favourable terms and conditions have caused stationary telephones to cease playing any greater role in distance communications. Users have lost interest in the use of devices that restrict their mobility. This process has become so conspicuous that these days many developers no longer install telephone cables in new builds, and many operators abandon stationary telephony services.

Today's society puts great emphasis on mobility. Having a mobile phone is natural and many employers recognise mobile phone as one of the primary equipment when hiring a new employee. Mobile phones not only give us mobility, but also allow us to call any other person on planet Earth. Of course, there still exist dead zones not covered by the network infrastructure, however, their surface is shrinking by the year [1].

The very idea of mobile telephony was postulated in the early nineteen-thirties. However, the first physical network was only established in the late sixties in Stockholm, Sweden. Its range did not exceed 30 km, and the number of users approximated one hundred. The principle of operation of mobile network's basic scheme is very simple. Mobile phone users can make calls provided that they are in the so-called "cell", that is in an area supplied with a signal from one of the base stations located nearby. The strength of the signal and the size of the cell's area relies on the technology applied by the operator, the transmitter's development status, and topology of the area.

Mobile phones at all times inform the base stations whether they are available. The base station receives these signals and transmits them to the next central control network which controls a group of base stations. In turn, control stations are connected to the telephone exchange whose function is to switch calls to the normal telephone network. Mobile network checks the quality of each call and selects the testing station