данным. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 1997. (Препринт № 1.)

3. Качанова Т. Л., Фомин Б. Ф. Симметрии, взаимодействия в локальностях, компоненты поведения сложных систем. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 1998. (Препринт № 2.)

4. Качанова Т. Л., Фомин Б. Ф. Основания системологии феноменального. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 1999.

5. Качанова Т. Л., Фомин Б. Ф. Введение в язык систем. СПб.: Наука, 2009.

6. Fomin B. F., Kachanova T. L. Physics of Systems is a postcybernetic paradigm of systemology // Intern. Symp. Science 2.0 and Expansion of Science: «S2ES» in the context of The 14th World-Multi-Conference «WMSCI 2010», Orlando, Florida, USA, June 29th–July 2nd, 2010. P. 244–249.

7. Качанова Т. Л., Фомин Б. Ф. Квалитология системного знания. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2014.

8. Ивин А. А. Аксиология. М.: Высш. шк., 2007.

9. Микешина Л. А. Ценностные предпосылки в структуре научного познания. М.: Прометей, 1990.

10. Качанова Т. Л., Фомин Б. Ф. Метатехнология системных реконструкций. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2002.

11. Fomin B. F., Kachanova T. L. Physics of Open Systems: Generation of System Knowledge // J. of Systemics, Cybernetics and Informatics. 2013. Vol. 11, № 2. P. 73–82.

12. Fomin B. F., Kachanova T. L. Cognition of ontology of Open Systems // Procedia Computer Science J. Elsevier B. V. 2017. Vol. 103. P. 339–346.

13. Язык открытых систем и экспертиза системного знания / В. О. Агеев, Т. Л. Качанова, Б. Ф. Фомин, О. Б. Фомин // Тр. VIII Междунар. конф. «Идентификация систем и задачи управления» (SICPRO'09), М., 2009. М.: ИПУ РАН, 2009. С. 144–190.

14. Качанова Т. Л., Фомин Б. Ф. Методы и технологии генерации системного знания. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2012.

T. L. Kachanova, B. F. Fomin, O. B. Fomin
Saint Petersburg Electrotechnical University «LETI»

GENERATING SCIENTIFICALLY PROVEN KNOWLEDGE ABOUT ONTOLOGY OF OPEN SYSTEMS

Contains a brief review of conception, scientific approach, methodological foundations, principles, axioms and main scientific results, which, in turn, have laid the foundations of a new cyber-physical paradigm for system research of open, natural, social, anthropogenic, and complex technical systems. Main purpose of this paradigm is: to automatically produce scientifically proven knowledge (gathered from a huge amount of semi-structured, multimodal, and heterogeneous empirical data) about general ontology of open systems given in hundreds and thousands indicators; to carry out a deep automatic research of obtained ontological knowledge with the aim to obtain full scientific understanding and rational explanation of such knowledge; to perform automatic analysis of the value (correctness, fullness, and completeness) of this knowledge; and to automatically shape resources of knowledge about general ontology of open systems and system problems.

Open systems, Big Data, physics of open systems, ontological knowledge, knowledge mining from empirical data, value of ontological knowledge, multidimensional knowledge-centric system analytics

УДК 621.3.037,372.9:004.3

А. Х. Мурсаев, Е. О. Стеблина
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Высокопроизводительный, экономичный кодек Хемминга

Передача информации по открытым каналам немыслима без применения средств защиты от помех. Одним из эффективных способов такой защиты является помехозащищенное кодирование. Примером достаточно простого и стойкого к помехам кодирования является кодирование Хемминга. Для обеспечения высоких скоростей и надежности передачи используются аппаратно-реализованные модули шифрации и дешифрации передаваемых данных – кодеки Хемминга. Предлагается развитие архитектуры хеммингoвских кодеков на базе многопотоковой битовой организации, чем обеспечивается уменьшение затрат оборудования, повышение производительности а также легкая перестройка на различные версии кода.

Помехозащищенное кодирование, аппаратурная реализация

В современных условиях передача информации по открытым каналам немыслима без применения средств защиты от помех. Одним из эффективных способов такой защиты является помехозащищенное кодирование. Преобразование передаваемой информации в помехозащищенную

форму в стандартном процессорном ядре часто не обеспечивает достаточной производительности. Для повышения пропускной способности канала все чаще используется аппаратная реализация устройств помехозащищенного кодирования и декодирования. Дополнительные возможности, в том числе перестройку на различные версии кодирования, дает применение микросхем программируемой логики, в том числе репрограммируемых систем на кристалле (PSoC).

К настоящему времени предложено достаточно много вариантов построения помехозащищенных кодов [1]–[3], но по-прежнему использование простых, но достаточно стойких к помехам кодов остается актуальным в силу их малой избыточности, простоты технической реализации и, в конечном счете, обеспечения высокой пропускной способности канала.

Важным семейством кодов являются линейные двоичные блоковые коды. Информационные и кодовые слова представлены в этих кодах в форме двоичных векторов, а процессы кодирования и декодирования могут быть описаны с помощью аппарата линейной алгебры, при этом компонентами используемых матриц и векторов являются символы 0 и 1. Операции над двоичными компонентами производятся по правилам арифметики по модулю 2. Блочный код заменяет каждый блок из k символов более длинным блоком из n символов, которые после передачи подлежат декодированию. Существуют также древовидные или последовательные коды, в которых значение очередного контрольного символа зависит от всего предшествующего фрагмента сообщения.

Кодер двоичного блокового (n, k)-кода отображает множество 2^k возможных двоичных информационных слов в множество 2^k n-мерных кодовых слов. Вместо k бит информационного вектора в канал передается n бит кодового вектора. Чем больше избыточность кода, тем большей возможностью для защиты от ошибок он обладает, однако с увеличением избыточности затраты на передачу информации также возрастают.

Важнейшими характеристками помехозащищенных кодов являются:

1. Число разрешенных и запрещенных комбинаций. Если n – число символов в блоке, $r = n - k$ – число проверочных символов в блоке, k – число информационных символов, то 2^n – число возможных кодовых комбинаций, 2^k – число разрешенных кодовых комбинаций, а $2^n - 2^k$ – число запрещенных комбинаций.

2. Избыточность кода r/n.

3. Минимальное кодовое расстояние. Минимальным кодовым расстоянием d называется минимальное число искаженных символов, необходимое для перехода одной разрешенной комбинации в другую.

Количество разрядов самокорректирующего кода, необходимых для коррекции, первоначально было определено Хеммингом. Более точную границу получил Плоткин и определил, что существует нижняя граница Плоткина для минимального количества контрольных разрядов r: если требуется минимальное кодовое расстояние d и $n > 2d - 1$, то $r \geq 2d - 2 - \log_2 d$.

Самоконтролирующиеся коды позволяют при передаче данных автоматически обнаруживать ошибки. Использование таких кодов предполагает наличие канала обратной связи. Самокорректирующиеся коды позволяют автоматически исправлять возникшие ошибки.

Примером достаточно простого и стойкого помехозащищенного кодирования является кодирование Хемминга. Известно достаточно много вариантов построения кодов Хемминга, отличающихся числом информационных и контрольных бит в передаваемом сообщении, а также порядком следования информационных и контрольных бит в передаваемом сообщении.

При кодировании Хемминга исходный текст разбивается на блоки равной длины. Биты исходного сообщения (называемые информационными) передаются без изменений, при этом к ним добавляются контрольные биты. В зависимости от конкретного способа кодирования контрольные биты либо распределяются между информационными, либо передаются плотной группой в конце каждого информационного блока. Данная статья предлагает развитие архитектуры хеммингoвских кодеков с целью уменьшения затрат оборудования, повышения производительности и обеспечения легкой перестройки на различные версии кода.

Общий принцип кодирования по Хеммингу – определение признаков четности избранных групп разрядов информационного слова (такие признаки называют контрольными битами). Совокупность контрольных бит и составляет контрольный код, передаваемый совместно с информационными битами. На приемном конце канала связи также вычисляются контрольные биты принятого информационного кода. Если все они совпадают с битами принятого контрольного кода – сообщение принято правильно, а если имеет-

ся отличие хотя бы в одной паре одноименных бит – сообщение было передано с ошибкой. Правильный выбор совокупности контрольных бит позволяет в результате логического анализа их совокупности корректировать априорно выбранное число ошибок.

Код Хемминга относится к типу линейных блочных кодов и является самоконтролирующимся и самокорректирующимся. Как правило, он позволяет исправить одиночную и обнаружить двойную ошибку. Существуют разновидности кодов Хемминга, позволяющие исправить и обнаружить большее количество ошибок. Для каждого числа проверочных разрядов $r = 3, 4, 5, \ldots$ существует классический совершенный код Хемминга с маркировкой $(n, k) = (2^r - 1, 2^r - 1 - r)$, корректирующая способность $t = 1$, $d_{\min} = 3$.

Практические методы построения совершенного кода Хемминга описаны в различных источниках, например [2], [3].

К (n, k)-коду Хемминга можно добавить проверку четности. Для этого к кодовому слову добавляется дополнительный бит, значение которого равно сумме по модулю 2 всех остальных бит кодового слова. Получится $(n + 1, k)$-код с наименьшим весом ненулевого кодового слова 4, способный исправлять одну и обнаруживать две ошибки [2].

Коды Хемминга накладывают ограничения на длину слов сообщения: длина может быть равна только числам вида $2^r - r - 1$: 1, 4, 11, 26, 57, ... Но на практике информация передается байтами или машинными словами, т. е. порциями по 8, 16, 32 или 64 бит, поэтому использование совершенных кодов не всегда удобно. В этих случаях обычно применяют квазисовершенные коды, исправляющие одну ошибку [2].

Реализация в традиционной архитектуре. Реализация алгоритма Хемминга процессором стандартной архитектуры (программная реализация) часто не обеспечивает необходимой скорости передачи сообщения в связи с наличием «неудобных» для таких процессоров операций последовательной побитовой обработки. Совершенный и квазисовершенный коды Хемминга в общем случае могут быть сформированы и декодированы прямым чтением контрольных кодов их памяти по адресу, задаваемому информационным кодом. По-видимому, такое решение будет оптимальным при небольшой разрядности информационных слов (не более 10–12). Увеличение разрядности информационного слова приводит к значительному росту объема памяти, и другие решения, основанные на формировании контрольных бит в режиме online, становятся предпочтительными.

Известны специализированные аппаратно-реализованные устройства хемминговского кодирования/декодирования, причем уже отмечалась простота и высокая производительность известных схем. Однако большинство известных реализаций ориентировано на параллельную обработку бит информационных слов. Рассмотрим это на примере дешифратора кода Хемминга (7, 4), способного корректировать одиночную ошибку передачи. Здесь и далее первая цифра обозначает суммарную длину защищенного блока, т. е. сумму информационных и контрольных бит, а вторая означает число информационных бит. Для вычисления каждого контрольного бита используется устройство свертки выбранных разрядов информационного кода по четности. Говорят, что выход устройства свертки является контрольным битом соответствующей группы информационных разрядов. Для кода (7, 4) логические схемы, формирующие контрольные биты r_1–r_3, определены как

$$r_1 = \left(\left(i_i \wedge i_2\right) \wedge i_3\right);$$
$$r_2 = \left(\left(i_2 \wedge i_3\right) \wedge i_4\right);$$
$$r_3 = \left(\left(i_1 \wedge i_2\right) \wedge i_4\right),$$

где i_k ($k = 1, 4$) – биты информационного кода, а $\wedge$ – операция исключающего ИЛИ.

Сформированные контрольные биты передаются в приемник совместно с информационными.

На рис. 1, отображающем базовую структуру декодера кода Хемминга, 3 устройства свертки выполняют оценку четности принятых информационных и контрольных бит, сравнивают их с соответствующими контрольными битами принятого кода.

На основании результатов сравнения логическая схема (цифровой корректор ошибок) определяет наличие и местоположение ошибки в переданном кодовом слове. Например, если выявлено несовпадение в первом и втором контрольных битах принятого информационного кода с соответствующими битам принятого контрольного кода, налицо ошибка в передаче бита i_3, а если несовпадение во всех трех схемах сравнения –

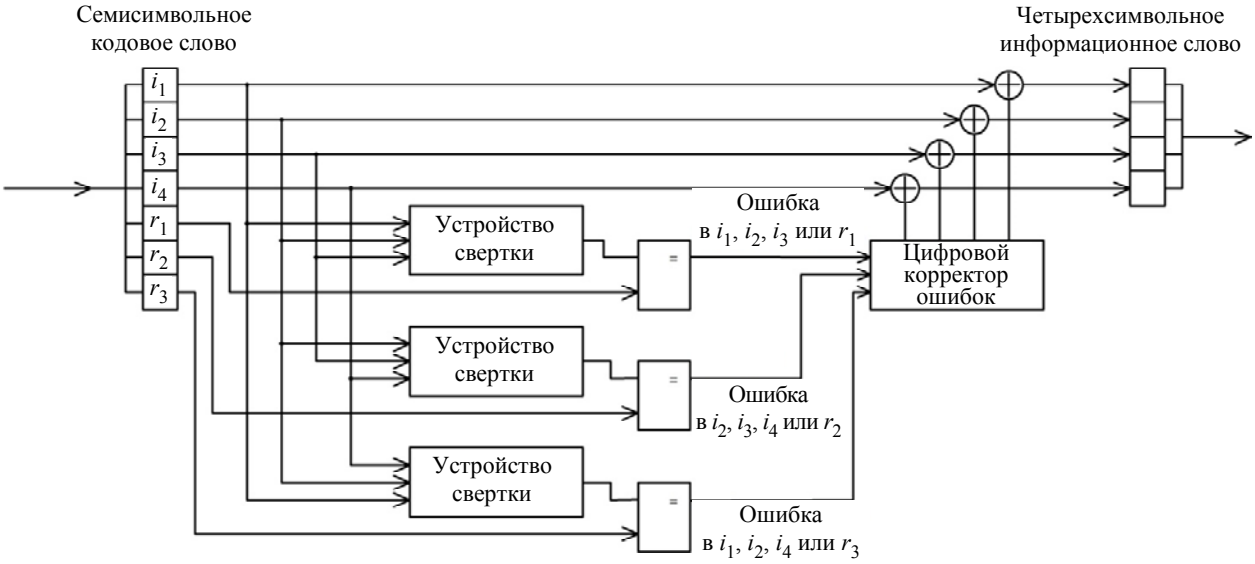

Рис. 1

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	1	0	1	1	0	1	0	1	0	1	1	0	1	0	1	0
1	0	1	1	0	1	1	0	0	1	1	0	1	1	0	0	1
0	1	1	1	0	0	0	1	1	1	1	0	0	0	1	1	2
0	0	0	0	1	1	1	1	1	1	1	0	0	0	1	1	3
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	4

неправильно передан бит i_2. Обнаружение несовпадения лишь в одной схеме сравнения соответствует ошибке передачи контрольных разрядов, а значит, коррекция не требуется. Цифровой корректор ошибки дешифрирует код ошибки, и ошибочный бит инвертируется. Исправленное кодовое слово подается на выход.

Традиционная организация кодека Хемминга неоптимальна, поскольку использует параллельное вычисление контрольных бит в нескольких устройствах свертки. К тому же устройства свертки содержат цепи переноса, вносящие дополнительные задержки, что снижает допустимую скорость передачи. Для увеличения пропускной способности устройства и уменьшения аппаратных затрат предлагается использовать мультипотоковую битовую архитектуру.

Мультипотоковая битовая архитектура. В подавляющем числе коммуникационных систем используется последовательный принцип передачи. Это наталкивает на мысль о целесообразности реализации необходимых преобразований также по принципу bit-by-bit (бит-за-битом) [4].

В системах с последовательной побитовой передачей наиболее удобным представляется передача контрольных бит единым фрагментом, вставляемым после информационного кода (такой код относится к семейству систематических кодов). Это позволяет совместить во времени передачу данных и формирование контрольного кода.

Для определения контрольных бит сформируем двумерный массив масок sindrom_table (ST). Количество строк sindrom_table равно количеству бит в информационном слове IB. Длина каждой строки равна числу контрольных бит CB. Если *p*-й бит информационного кода участвует в формировании *r*-го контрольного бита, то он установлен в единицу, иначе он имеет нулевое значение.

Рассмотрим, например, формирование кода Хемминга $n = 21$, $k = 16$, который обсуждался в работе [5]. Минимальное кодовое расстояние для этого кода $d = 3$, избыточность кода 0.24, код исправляет одиночные ошибки.

Массив масок для формирования этого кода представлен в таблице. В верхней строке таблицы указаны номера информационных бит в кодовом слове. В крайнем правом столбце указаны номера соответствующих контрольных бит. Заметим, что таблица представляет массив масок в повернутом виде – строкам массива соответствуют столбцы таблицы.

Здесь, например, пятый информационный бит участвует в формировании контрольных бит с номерами $r = 0, 3$, а значит, ST (5) = «10010».

Остальные столбцы формируются аналогичным образом. Код каждого столбца называют порождающим кодом.

Предлагаемая структура кодера представлена на рис. 2.

При поступлении сигнала о начале передачи блока Start счетчик Ct и триггеры контрольных бит сбрасываются в ноль. Назовем это состояние первой фазой цикла. Далее состояние счетчика инкрементируется синхронно с подачей информационных бит, и пока содержимое счетчика не превысит числа информационных бит IB, эти биты передаются на выход (вторая фаза цикла). При этом из таблицы выбирается строка, содержащая единицы в разрядах, в которых поступающий в текущем такте бит информационного слова участвует в формировании соответствующего контрольного бита. Логические схемы NSS_0–NSS_{CB-1} (NSS – Next State Selector) обеспечивают подачу на информационные входы триггеров сигналов, позволяющих реализовать приведенные далее переключения.

Обозначим st(u, v) – u-й бит v-й строки массива ST.

Во второй фазе NSS переключаются таким образом, что при $x_k = 1$ и st(p, k) = 1, где k – номер бита поступающего информационного слова X (и, соответственно, состояние счетчика)), а p – порядковый номер контрольного бита, состояние p-го триггера инвертируется, а во всех остальных случаях состояние сохраняется.

Преобразования на этом этапе можно представить соотношениями

$$\begin{aligned} R(0) &= 0, \\ R(k) &= R(k-1) \oplus x_k \otimes \mathrm{ST}(k), \end{aligned} \qquad (1)$$

где $R(k) = (r_0, r_2, ..., r_{CB-1})$ – вектор состояния регистра контрольных бит после k-го шага.

Сложение и умножение выполняется по правилам арифметики по модулю 2.

Далее, в третьей фазе цикла, NSS переключаются так, что вход каждого триггера соединяется с выходом предыдущего. Таким образом, реализуется последовательная передача контрольных бит на выход шифратора. При достижении счетчиком состояния IB + SB работа схемы приостанавливается до поступления следующего информационного блока.

Соответствующая принятому представлению структура декодера представлена на рис. 3.

Заметим, что альтернативой предлагаемой структуры может стать структура, в которой для накопления контрольных признаков используется закольцованный регистр сдвига [1].

На рис. 3 показана структура приемника закодированного сигнала, содержащего вычислитель контрольных бит информационного кода, в точности соответствующий такому же блоку передатчика, приведенному на рис. 2.

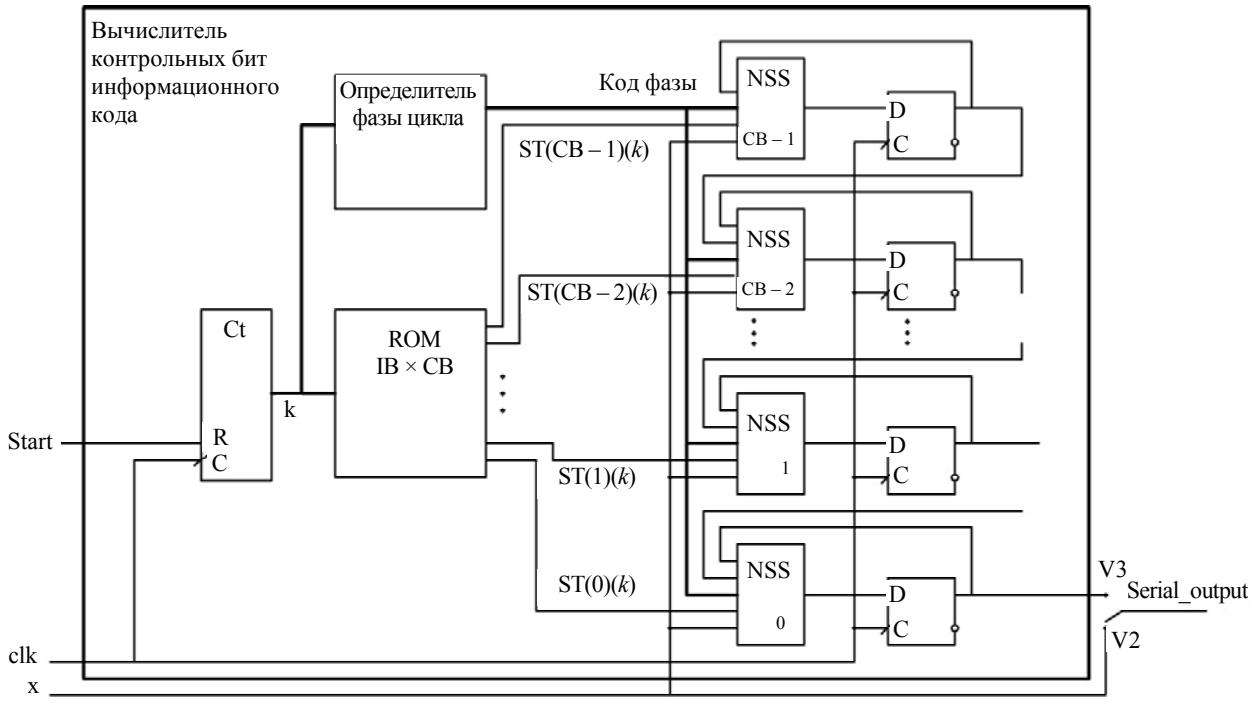

Рис. 2

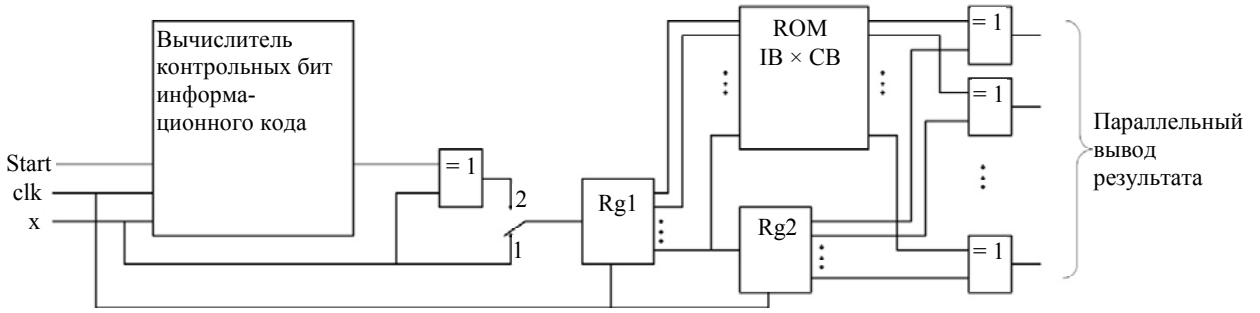

Рис. 3

Работа приемника, как и передатчика, состоит из трех фаз. В первой фазе после появления нового блока (например, синхронизированного с сигналом Start) в вычислителе контрольных бит выполняется сброс счетчика и регистров.

Во второй фазе работы последовательность информационных бит приемника поступает на регистр сдвига Rg1 и далее через него в Rg2. Одновременно так же, как и в передатчике, вычисляются контрольные биты и результат накапливается во внутреннем регистре вычислителя контрольных бит.

В третьей фазе работы приемника, после того как все информационные биты были переданы, заново вычисленные контрольные биты последовательно подаются на выход вычислителя контрольных бит, а оттуда – на один вход устройства сложения по модулю 2. Вход декодера подключается ко второму входу устройства сложения по модулю 2, таким образом происходит побитовое сравнение вычисленных и принятых контрольных бит.

Результат сравнения поступает в регистр сдвига. При этом остающиеся биты информационного кода «проталкиваются» в Rg2. К концу этой фазы в Rg2 зафиксирован принятый и, возможно, ошибочный информационный код, а в Rg1 сформирован синдром ошибки. Можно видеть, что синдром ошибки – это код, в котором установлены в единицу биты, соответствующие позициям, в которых не совпадают биты принятого контрольного кода и соответствующие им признаки четности принятого информационного кода. Синдром ошибки поступает в качестве адреса в запоминающее устройство. На выходе запоминающего устройства присутствует единственная единица, позиция которой соответствует номеру неисправного информационного бита, если таковой есть. После этого биты регистра Rg2 складываются по модулю 2 с этим кодом, фактически инвертируя неисправный бит. Результат подается на параллельный выход приемника.

Заметим, что альтернативой предлагаемой структуре может стать реализация, в которой для накопления контрольных признаков используется циклический регистр сдвига для фиксации промежуточных значений контрольных бит [1]. Соотношения (1) при этом дополняются операцией циклического сдвига. Для такой конфигурации можно построить и использовать общий для всех шагов порождающий код – исключить sindrom_table. Это позволяет уменьшить (хотя и незначительно) объем оборудования. Правда, построение порождающего кода, обнаруживающего позицию ошибки передачи сообщения, в общем случае является трудоемкой задачей, а некоторые версии кода не могут генерироваться в такой структуре. К тому же исправление ошибок сопряжено с дополнительными временными или аппаратными затратами.

Нетрудно видеть, что вычислитель одного контрольного бита и в передатчике, и в приемнике сводится к одиночному счетному триггеру. Отсутствие цепей переносов в схемах формирования контрольных бит обеспечивает максимальную в рамках избранной схемотехники пропускную способность. И, наконец, структура универсальна в области хеммингoвского кодирования. Она пригодна для реализации любого варианта кодирования, включая коды Хемминга с исправлением двух ошибок: требуется только изменение числа триггеров-накопителей контрольных разрядов и изменение содержимого памяти массива масок.

Применение данного подхода позволяет создать аналогичные по структуре устройства, реализующие другие алгоритмы кодирования, например кодирование Рида–Соломона и Хсяо.

Шифратор и дешифратор были синтезированы в среде программируемой логики Quartus II 9.1sp2 Web Edition.

Верификация подтвердила их работоспособность. Для формата (21,16) производительность канала передачи составила до 1 Гбит/с, а затраты оборудования приемника составили 100 логических элементов и 26 триггеров, что превосходит другие известные реализации.

СПИСОК ЛИТЕРАТУРЫ

1. Вернер М. Основы кодирования: учеб. для вузов. М.: Техносфера, 2004. 288 с.
2. Золотарев В. В., Овечкин Г. В. Помехоустойчивое кодирование. Методы и алгоритмы: справ. М.: Горячая линия–Телеком, 2004. 126 с.
3. Питерсон Р., Уэлдон Э. Коды, исправляющие ошибки. М.: Мир, 1976. 595 с.
4. Реконфигурируемые мультиконвейерные вычислительные структуры / И. А. Каляев, И. И. Левин, Е. А. Семерников, В. И. Шмойлов; ЮНЦ РАН. Ростов н/Д, 2008. 393 с.
5. Код Хэмминга. Пример работы алгоритма. URL: https://habrahabr.ru/post/140611/.

A. H. Mursaev, E. O. Steblina
Saint Petersburg Electrotechnical University «LETI»

HIGH-PERFORMANCE, COST-EFFECTIVE HAMMING CODEC

The information transmission via open channels is without undoubtedly requires the use of interference protection means. One of the effective means of such protection is error protection encoding. The example of relatively simple and interference-resistant encoding is Hamming encoding. To ensure high speed and reliability of transmission hardware-implemented modules of encryption and decryption of transmitted data are used – Hamming codecs. The article presents development of Hamming codec architecture based on multi-way bit organization. Architecture ensures cost reduction, higher performance and easy reconfiguration for various versions of code.

Error correcting codes, hardware implementation

УДК 004.946

И Вэньлун
Санкт-Петербургский государственный электротехнический университет «ЛЭТИ» им. В. И. Ульянова (Ленина)

Топологический подход к геометрическому моделированию формы листьев растений

Рассматриваются топологические методы и алгебраические средства, применяемые в аспектно-ориентированном подходе к решению задач компьютерного моделирования формы листьев растений с учетом экофизиологических факторов. Поскольку топологическая структура формы листа может рассматриваться как инвариант в процессе его роста, для получения полноценной геометрической модели информация, описывающая форму листа, сначала разделяется на топологическую и геометрическую части; затем применяются метод n-Gmaps и алгебраическая структура «моноид» для построения топологической модели формы листа; наконец, применяются встроенные отображения топологических блоков на соответствующие вершины формы листа. Методы, использованные в данной статье, способствуют повышению абстрагирования модели формы листа растения для возможности ее повторного использования, а комбинаторные топологические операции могут сократить ошибки в процессе округления в ходе численных расчетов геометрического моделирования, тем самым повышая точность моделирования.

Геометрическое моделирование, концептуализация топологии, метод n-Gmaps, моноиды, геометрическая форма листьев растений

В реальном трехмерном пространстве форма роста листа растения в значительной степени влияет на эффективность фотосинтеза, происходящего у него внутри [1]. Для того чтобы количественно регулировать действие такой физиологической функции, исследователи стали активно изучать законы морфогенеза растений [2], [3]. Однако фитометрические системы (ФМС) с традиционным ручным управлением датчиками зависят от погодных условий, так что нелегко до-