УДК 621.394.147
DOI: 10.55648/1998-6920-2022-16-1-46-55

Конструкция равновесных корректирующих кодов с синхронизацией

Е. П. Мачикина, В. Г. Насенник

В статье предлагается конструкция корректирующего блочного кода с возможностью синхронизации при пакетной последовательной передаче данных в зашумленном канале, который имеет невысокую сложность реализации кодера и декодера. Проведено исследование корректирующей способности построенных кодов в сравнении с некоторыми каскадными кодами.

Ключевые слова: равновесные коды, коды с синхронизацией, исправление ошибок, DC-free, DC-balanced, RLL.

1. Введение

При разработке электронной аппаратуры для промышленного оборудования часто возникает задача передачи по последовательному каналу связи высокоскоростных команд и данных между блоками аппаратуры в условиях сильных электромагнитных помех от расположенных рядом узлов и кабелей промышленного оборудования – сервоприводов, коллекторных двигателей, искрящих контактов реле. Для защиты от импульсных помех применяется помехоустойчивое кодирование. Для защиты от кондуктивных помех по общему проводу небольшой амплитуды применяются дифференциальные линии связи, но при кондуктивных помехах большой амплитуды необходимо применение трансформаторной или конденсаторной гальваноразвязки или использование оптических линий связи. По сравнению с медными линиями оптоволоконные линии связи требуют более дорогих трансиверов, коннекторов и оборудования для монтажа коннекторов, а также имеют ограниченное применение в буксируемых кабельных цепях, где кабели подвергаются регулярным деформациям. Для совместимости с гальваноразвязкой применяемый код не должен иметь постоянной составляющей, быть равновесным – при биполярном кодировании количество нулей и единиц в потоке должно быть равным. Кроме того, предъявляется ограничение на величину динамического разбаланса (running disparity), чтобы не происходило насыщения трансформаторов или конденсаторов, обеспечивающих гальваническую развязку – эта величина должна быть как можно меньше и как можно быстрее возвращаться к нулю. Для высокоскоростных оптических трансиверов также часто требуется, чтобы применяемый код не имел постоянной составляющей.

На выбор способа кодирования информации для таких каналов обмена данными влияет необходимость решения следующих задач:

– обеспечение синхронизации данных (битовой, байтовой, блочной и т.д.);

– исправление и обнаружение ошибок;

– обеспечение совместимости с каким-либо видом гальванической развязки.

Кроме того, для методов кодирования могут выдвигаться и дополнительные требования, связанные со сложностью реализации кодера и декодера данных при передаче информации. Известны блочные и свёрточные коды, которые одновременно обеспечивают исправление ошибок и равновесность [1, 2].

Известные в литературе методы блочного кодирования данных не обеспечивают одновременной реализации всех упомянутых требований, поэтому часто на практике применяются комбинированные (каскадные) коды. Внутренний код обычно обеспечивает синхронизацию и совместимость с гальванической развязкой, а внешний код – обнаружение и исправление ошибок, возникающих в канале.

В качестве внутреннего кода обычно используют равновесные коды, которые обеспечивают совместимость с гальванической развязкой за счёт равенства частоты появления 0 и 1 в потоке и ограничения на динамический разбаланс.

Битовая синхронизация передачи данных обычно достигается за счёт частого переключения между передаваемыми символами, т.е. ограничением максимальной длины последовательности одинаковых символов в потоке (run length limit, RLL) [3, 4]. Это свойство похоже на ограничение динамического разбаланса, но не тождественно ему. Примерами кодов, которые обеспечивают оба этих свойства, могут служить манчестерское кодирование, 4B6B, 8B10B, 64B/66B и т.д. [3, 5] Похожие требования на ограничение длины последовательности одинаковых бит возникают в магнитной или оптической записи информации.

В качестве внешнего кода можно использовать свёрточные или блочные коды с исправлением ошибок – код Хэмминга, код Голея, коды Боуза–Чоудхури–Хоквингема (БЧХ), коды Рида–Маллера, коды Рида–Соломона и др. Однако решение каждой из упомянутых проблем по отдельности увеличивает избыточность кода и заметно усложняет реализацию кодера/декодера данных.

Получающиеся кодовые конструкции можно сравнивать между собою по трём основным характеристикам:

1. относительная скорость кода (code rate), т.е. отношение количества «полезных» бит к количеству передаваемых бит. Обратная к ней величина называется избыточностью кода.

2. корректирующая способность, т.е. зависимость вероятности ошибочного приёма блока от вероятности ошибки бита в канале.

3. сложность реализации или объем аппаратных ресурсов, требуемых для реализации кода. Например, сложность можно оценивать как количество регистров (FF) и комбинаторных логических элементов (LUT) в программируемых логических матрицах (FPGA).

В этой статье предлагается конструкция корректирующего блочного кода с возможностью синхронизации, т.е. решающего одновременно все упомянутые задачи и при этом имеющего невысокую сложность реализации кодера и декодера, а также вполне конкурентоспособные характеристики относительной скорости и корректирующей способности в сравнении с каскадными кодами.

2. Конструкция кода и его свойства

Обозначим триплет (тройку битов) с одной единицей на i-том месте как $\alpha^i, i = 0, 1, 2$, т.е. $\alpha^0 = (001)$, $\alpha^1 = (001)$, $\alpha^0 = (001)$. Триплеты с двумя единицами, которые являются инверсиями триплетов α^i, $i = 0, 1, 2$, будут обозначаться как $\overline{\alpha}^i$, $i = 0, 1, 2$. Например, $\overline{\alpha}^2 = (011)$.

В качестве кодовых слов используются комбинации триплетов в определенном порядке, т.е. кодовое слово строится как

$$\begin{gathered} c(i,j) = \alpha^i \overline{\alpha}^j \alpha^{f(i,j)} \overline{\alpha}^{g(i,j)}, \overline{c}(i,j) = \overline{\alpha}^i \alpha^j \overline{\alpha}^{f(i,j)} \alpha^{g(i,j)} \\ f(i,j) = (a_0 + a_1 \cdot i + a_2 \cdot j) \bmod 3, \\ g(i,j) = (b_0 + b_1 \cdot i + b_2 \cdot j) \bmod 3, \end{gathered} \tag{1}$$

где $a_0, b_0 \in \{0, 1, 2\}$, $a_1, a_2, b_1, b_2 \in \{1, 2\}$, $i, j = 0, 1, 2$. Параметры $a = (a_0, a_1, a_2)$ и $b = (b_0, b_1, b_2)$ являются одинаковыми для всех кодовых слов одного кода. Код, построенный по

правилу (1), обозначим как $\mathrm{Code}(a, b)$. Нетрудно видеть, что $\mid \mathrm{Code}(a, b) \mid = 18$, поскольку $i, j = 0, 1, 2$ и код $\mathrm{Code}(a, b)$ содержит инверсию каждого кодового слова. Такая мощность кода дает возможность однозначно кодировать четырёхбитовые последовательности.

Покажем, что различным параметрам $a = (a_0, a_1, a_2)$ и $b = (b_0, b_1, b_2)$ соответствуют различные коды, построенные согласно (1). Сравним кодовые слова кодов $\mathrm{Code}(a, b)$ и $\mathrm{Code}(a', b')$ для $a \neq a'$ и $b \neq b'$. Предположим, что для всех значений i, j кодовые слова из $\mathrm{Code}(a, b)$ и $\mathrm{Code}(a', b')$ совпадают. Поскольку все кодовые слова имеют одинаковое строение как комбинации триплетов, сравнивать будем «показатели» для третьего и четвёртого триплетов для всех $i, j = 0, 1, 2$

$$\begin{gathered}(a_0 + a_1 \cdot i + a_2 \cdot j) \bmod 3 = (a'_0 + a'_1 \cdot i + a'_2 \cdot j) \bmod 3, \\ (b_0 + b_1 \cdot i + b_2 \cdot j) \bmod 3 = (b'_0 + b'_1 \cdot i + b'_2 \cdot j) \bmod 3.\end{gathered} \tag{2}$$

Очевидно, что (2) возможно, только если $a_k - a'_k \bmod 3 = 0$ и $b_k - b'_k \bmod 3 = 0$ для всех $k = 0, 1, 2$. Поскольку все параметры могут принимать только значения от 0 до 2, то $a_k = a'_k$ и $b_k = b'_k$, $k = 0, 1, 2$. Такое однозначное соответствие между набором параметров (a, b) и множеством кодов $\mathrm{Code}(a, b)$ доказывает следующее утверждение

Утверждение 1. *Количество различных кодов* $\mathrm{Code}(a, b)$*, построенных согласно (1), равно 144.*

Различные свойства построенных кодов сформулированы в утверждениях 2–5.

Утверждение 2. *Код* $\mathrm{Code}(a, b)$ *является равновесным.*

Доказательство. Действительно, поскольку кодовые слова, построенные согласно (1), состоят из двух триплетов с одной единицей и из двух триплетов с двумя единицами, то каждое кодовое слово содержит ровно 6 нулей и 6 единиц. □

Утверждение 3. *В последовательности кодовых слов кода* $\mathrm{Code}(a, b)$ *количество последовательных 1 или 0 не больше 4.*

Доказательство. Поскольку кодовые слова, построенные согласно (1), состоят из чередующихся триплетов с одной единицей и с двумя единицами, то внутри каждого кодового слова содержится не более трех последовательных одинаковых символов. Тройки нулей или единиц получаются на стыках триплетов. Последовательности из четырех одинаковых символов могут возникать только на стыках кодовых слов. Например, когда кодовое слово с четвертым триплетом (011) стыкуется с кодовым словом, которое начинается на (110). Последовательности из пяти и более одинаковых символов в кодовой последовательности невозможны. □

Утверждение 4. *Динамический разбаланс кода* $\mathrm{Code}(a, b)$ *не больше 2.*

Доказательство. Рассмотрим динамический разбаланс в потоке битов. Предполагаем, что используется биполярное представление – бит 1 передаётся импульсом напряжения положительной полярности единичной амплитуды, а бит 0 передаётся импульсом напряжения отрицательной полярности такой же амплитуды. Перед началом передачи динамический разбаланс полагается равным нулю.

После передачи первого бита динамический разбаланс по своей абсолютной величине становится равным 1.

Если второй бит отличается от первого бита, то в этом случае динамический разбаланс обнуляется. Если же второй бит такой же, как и первый, то в этот момент динамический разбаланс достигает абсолютной величины 2.

Поскольку в триплете всегда два нуля и одна единица или две единицы и один нуль, то после передачи третьего бита триплета динамический разбаланс становится равен по абсолютной величине 1.

В зависимости от четвёртого бита динамический разбаланс либо обнуляется, либо достигает абсолютной величины 2. После пятого бита динамический разбаланс становится равным по абсолютной величине 1.

Поскольку кодовые слова, построенные согласно (1), состоят из пар триплетов с разными инверсиями, то каждый второй триплет пары восстанавливает баланс, нарушаемый первым триплетом, т.е. после шестого бита динамический разбаланс обнуляется.

Таким образом, наибольший динамический разбаланс достигается только после второго или после четвёртого бита и достигает абсолютной величины 2. Во всех остальных случаях он по абсолютной величине меньше 2. □

Утверждение 5. *Для кодового расстояния кода* $\mathrm{Code}(a,b)$ *верно, что*

$$d(\mathrm{Code}(a,b)) \;=\; \begin{cases} 6, & (a_1+a_2+b_1+b_2) \bmod 2 = 1, \\ 4, & (a_1+a_2+b_1+b_2) \bmod 2 = 0. \end{cases} \tag{3}$$

Доказательство. Поскольку кодовые слова состоят из триплетов, то расстояние Хэмминга [6] между двумя кодовыми словами кода будет складываться из расстояний между соответствующими триплетами. Рассмотрим несколько возможных случаев.

Случай 1. Для двух кодовых слов $c(i,j) = \alpha^i\overline{\alpha}^j\alpha^{f(i,j)}\overline{\alpha}^{g(i,j)}$ и $c(k,l) = \alpha^k\overline{\alpha}^l\alpha^{f(k,l)}\overline{\alpha}^{g(k,l)}$ определим минимально возможное расстояние Хэмминга между ними. Обозначим $\triangle i = (i-k) \bmod 3$, $\triangle j = (j-l) \bmod 3$. Если $\triangle i \neq 0$ и $\triangle j \neq 0$, то кодовые слова $c(i,j)$ и $c(k,l)$ точно отличаются в первом и втором триплетах, а в третьем и четвертом триплетах могут совпадать, т.е. $d(c(i,j),c(k,l)) \geq 4$. Равенство будет достигаться, только если

$$\begin{aligned} f(i,j) - f(k,l) &= (a_1 \cdot \triangle i + a_2 \cdot \triangle j) \bmod 3 = 0, \\ g(i,j) - g(k,l) &= (b_1 \cdot \triangle i + b_2 \cdot \triangle j) \bmod 3 = 0. \end{aligned} \tag{4}$$

Тогда если $\triangle i = \triangle j \neq 0$, то соотношения (4) выполняются при $(a_1+a_2) \bmod 3 = 0$, $(b_1+b_2) \bmod 3 = 0$, т.е. a_1 и a_2 имеют разную четность, b_1 и b_2 имеют разную четность. Значит, если $(a_1+a_2+b_1+b_2)$ – четное, то $d(c(i,j),c(k,l)) = 4$. Аналогичное рассуждение, если $\triangle i \neq \triangle j$. Если $\triangle i = 0$ или $\triangle j = 0$, то $f(i,j) \neq f(k,l)$, $g(i,j) \neq g(k,l)$ и $d(c(i,j),c(k,l)) \geq 6$.

Случай 2. Для двух кодовых слов $\overline{c}(i,j) = \overline{\alpha}^i\alpha^j\overline{\alpha}^{f(i,j)}\alpha^{g(i,j)}$ и $\overline{c}(k,l) = \overline{\alpha}^k\alpha^l\overline{\alpha}^{f(k,l)}\alpha^{g(k,l)}$ рассуждения такие же, как и в случае 1.

Случай 3. Найдем расстояние между кодовыми словами $c(i,j) = \alpha^i\overline{\alpha}^j\alpha^{f(i,j)}\overline{\alpha}^{g(i,j)}$ и $\overline{c}(k,l) = \overline{\alpha}^k\alpha^l\overline{\alpha}^{f(k,l)}\alpha^{g(k,l)}$. Обозначим $\triangle i = (i-k) \bmod 3$, $\triangle j = (j-l) \bmod 3$. Если $\triangle i = 0$, то $f(i,j) \neq f(k,l)$, $g(i,j) \neq g(k,l)$ и кодовые слова отличаются во всех триплетах, причем $d(c(i,j),\overline{c}(i,l)) = d(\alpha^i,\overline{\alpha}^i) + d(\alpha^l,\overline{\alpha}^j) + d(\alpha^{f(i,j)},\overline{\alpha}^{f(i,l)}) + d(\alpha^{g(i,j)},\overline{\alpha}^{g(i,l)}) = 3+1+1+1 = 6$. Если $\triangle j = 0$, то так же $d(c(i,j),\overline{c}(k,j)) = 6$. Если $\triangle i \neq 0$ и $\triangle j \neq 0$, то кодовые слова будут отличаться во всех четырех триплетах как минимум в одной позиции, т.е. $d(c(i,j),\overline{c}(k,j)) \geq 4$. Для того, чтобы $d(c(i,j),\overline{c}(k,j)) \geq 6$, нужно, чтобы в кодовых словах третьи или четвертые триплеты были различны, т.е. $(a_1 \cdot \triangle i + a_2 \cdot \triangle j) \bmod 3 = 0$ или $(b_1 \cdot \triangle i + b_2 \cdot \triangle j) \bmod 3 = 0$. Рассуждения, как в случае 1, приводят к выводу, что при нечетном $(a_1+a_2+b_1+b_2)$ расстояние $d(c(i,j),\overline{c}(k,l)) \geq 6$. □

Таким образом, среди построенных кодов имеются 72 кода с кодовым расстоянием 6 и 72 кода с кодовым расстоянием 4. Как известно из литературы [3, 6, 7, 8], для того, чтобы исправить t ошибок в кодовом слове, код должен иметь кодовое расстояние $d \geq 2t+1$. Кодовое расстояние 6 позволяет гарантированно исправить не более 2 ошибок замещения в кодовом слове, а кодовое расстояние 4 – не более одной ошибки. Для кода с кодовым расстоянием 6 кодовые слова, имеющие три ошибки, часто имеют расстояние Хэмминга 3 до нескольких кодовых слов, так что они не могут быть однозначно декодированы, поэтому такие ошибки могут быть обнаружены, но не исправлены. Кодовое расстояние 4 позволяет обнаруживать 2 ошибки в слове.

Полностью аналогичные рассуждения можно провести и для кодов с симметричной инверсией триплетов – $c(i,j) = \alpha^i\overline{\alpha}^j\overline{\alpha}^{f(i,j)}\alpha^{g(i,j)}$ и $\overline{c}(i,j) = \overline{\alpha}^i\alpha^j\alpha^{f(i,j)}\overline{\alpha}^{g(i,j)}$. Отличие заключается только в том, что для этих кодов иначе выглядит утверждение 3, поскольку последовательность из 4 одинаковых бит может встречаться не только на стыках кодовых слов, но и в середине кодового слова. Такая схема позволяет построить ещё 72 кода с кодовым расстоянием 6.

3. Блочная синхронизация

Для достижения блочной синхронизации в литературе предлагаются различные методы [3, 8]. Например, в коде 8B10B используются комма-коды, которыми являются 10 кодовых слов, содержащих 5 одинаковых бит подряд, при этом 5 одинаковых бит подряд не встречаются ни в каких других кодовых словах или стыках кодовых слов [5].

Математический анализ стыков кодовых слов представляется достаточно сложной задачей [2], поэтому для исследования кодов $\mathrm{Code}(a,b)$ была написана программа на языке C, с помощью которой были проанализированы все возможные последовательности кодовых слов для каждого из построенных кодов с расстоянием 6. Анализ получившихся кодов $\mathrm{Code}(a,b)$ с кодовым расстоянием 6 показал, что ни в одном из кодов уникальных комма-кодов нет.

Далее был проведен программный анализа комбинаций из двух кодовых слов. После исследования всех 144 кодов $\mathrm{Code}(a,b)$ в некоторых кодах были найдены пары кодовых слов, которые никогда не могут встретиться в потоке данных при неправильной блочной синхронизации со смещением от 1 до 11 бит. Всего было выявлено 10 таких кодов – 4 кода с симметричной инверсией триплетов и 6 кодов с антисимметричной инверсией. В этих кодах найдено по 2 пары кодовых слов, которые не только не встречаются в потоке произвольных данных, но и, более того, находятся на расстоянии Хэмминга не менее 4 до ближайшей комбинации. Наихудшая ситуация достигается только при смещении на 6 бит. Таким образом, найденные пары кодовых слов пригодны как комма-коды для установления блочной синхронизации, в том числе и при наличии помех — до 1 ошибки замещения в двойном кодовом слове.

Дополнительное сравнение этих найденных 10 кодов между собой позволило заметить следующее.

Во-первых, в кодах с симметричной инверсией триплетов 4 одинаковых бита могут встретиться как на стыках кодовых слов, так и в середине, а в кодах с асимметричной инверсией триплетов 4 одинаковых бита могут встретиться только на стыках кодовых слов. Таким образом, анализ статистики встречаемости последовательностей из 4 одинаковых бит подряд может быть использован для целей синхронизации или обнаружения потери синхронизации, причём коды с асимметричной инверсией имеют некоторое преимущество перед кодами с симметричной инверсией.

Во-вторых, в 4 кодах с антисимметричной инверсией может встречаться регулярная последовательность из чередующихся троек битов 000 и 111, затрудняющая однозначную интерпретацию, что может рассматриваться как слабость.

После исключения таких кодов из рассмотрения остаются два эквивалентных кода без упомянутых слабостей с параметрами $a = (2,2,2)$, $b = (2,1,2)$ и $a = (2,1,1)$, $b = (2,2,1)$.

Обозначим найденные коды как 4B12B_1 и 4B12B_2, соотношения (1) для этих кодов будут иметь вид: $f(i,j) = (2 + 2\cdot i + 2\cdot j) \bmod 3$, $g(i,j) = (2 + i + 2\cdot j) \bmod 3$ для кода 4B12B_1 и $f(i,j) = (2 + i + j) \bmod 3$, $g(i,j) = (2 + 2\cdot i + j) \bmod 3$ для кода 4B12B_2.

Коды 4B12B_1 и 4B12B_2 приведены в табл. 1 и 2. Кодовые слова записаны в предположении, что передача битов в канал связи происходит слева направо, начиная с крайнего левого бита.

Анализ расстояния Хэмминга от всех возможных 12-битных слов до кодовых слов 4B12B

Т а б л и ц а 1. Код 4B12B_1

i	j	$f(i,j)$	$g(i,j)$	$c(i,j)$	$\overline{c}(i,j)$
0	0	2	2	001 110 100 011	110 001 011 100
0	1	1	1	001 101 010 101	110 010 101 010
0	2	0	0	001 011 001 110	110 100 110 001
1	0	1	0	010 100 010 110	101 011 101 001
1	1	0	2	010 101 001 011	101 010 110 100
1	2	2	1	010 011 100 101	101 100 011 010
2	0	0	1	100 110 001 101	011 001 110 010
2	1	2	0	100 101 100 110	011 010 011 001
2	2	1	2	100 011 010 011	011 100 101 100

Т а б л и ц а 2. Код 4B12B_2

i	j	$f(i,j)$	$g(i,j)$	$c(i,j)$	$\overline{c}(i,j)$
0	0	2	2	001 110 100 011	110 001 011 100
0	1	0	0	001 101 001 110	110 010 110 001
0	2	1	1	001 011 010 101	110 100 101 010
1	0	0	1	010 110 001 101	101 001 110 010
1	1	1	2	010 101 010 011	101 010 101 100
1	2	2	0	010 011 100 110	101 100 011 001
2	0	1	0	100 110 010 110	011 001 101 001
2	1	2	1	100 101 100 101	011 010 011 010
2	2	0	2	100 011 001 011	011 100 110 100

показывает, что некоторые слова оказываются на расстоянии Хэмминга 3 только до 1 кодового слова из 4B12B и поэтому могут быть однозначно декодированы. Таким образом, для каждого кодового слова из 4B12B можно однозначно декодировать 1 слово без ошибок, $C_{12}^1 = 12$ – слов с одной ошибкой, $C_{12}^2 = 66$ – слов с двумя ошибками и 12 из $C_{12}^3 = 220$ – слов с тремя ошибками.

Одну из найденных пар кодовых слов можно выделить в качестве комма-кодов и обозначить как K_0 и K_1, остальные кодовые слова – $D_0, \ldots, D_{15}$. Для кода 4B12B_1 в качестве K_0 и K_1 подходят кодовые слова $c(0,0)$ и $\overline{c}(0,2)$, а для кода 4B12B_2 – кодовые слова $c(2,2)$ и $\overline{c}(0,0)$.

Полученные коды 4B12B_1 и 4B12B_2 пригодны для использования при пакетной передаче сообщений. В канале связи постоянно передаются кодовые слова, при этом пакеты данных кодируются с использованием кодовых слов $D_0, \ldots, D_{15}$, а межпакетные интервалы заполняются чередующимися кодовыми словами K_0 и K_1. Кодовое слово D, встречающееся после слова K, интерпретируется как начало пакета данных.

Коды 4B12B_1 и 4B12B_2 не исправляют ошибок синхронизации (удаление и вставка битов), но для этих кодов возможно восстановление синхронизации по паре комма-кодов и обнаружение потери блочной синхронизации. Приёмник устанавливает или восстанавливает блочную синхронизацию при обнаружении в потоке пары K_0K_1.

Потеря синхронизации может быть обнаружена при помощи подсчёта статистики встречаемости троек битов 000 и 111 в последовательности передаваемых бит в канале. При правильной синхронизации такие триплеты не встречаются в потоке, и приёмник может их встретить только в результате ошибок замещения, а при смещении на 1 или 2 бита такие тройки встречаются часто. Соответственно, для детектирования потери блочной синхронизации достаточно иметь всего три счётчика, в которых подсчитывать каждый случай возникновения таких триплетов – со сдвигом на 0, 1 и 2 бита. Если синхронизация правильная, то счётчик со сдвигом 0 регистрирует триплеты 000 и 111 наиболее редко, только когда их возникновение обусловлено ошибками. Если же счётчик со сдвигом 0 опережает два других счётчика на некую достаточно большую величину, это можно интерпретировать как признак потери блочной синхронизации. Авторы экспериментально подобрали величину 30. Уменьшение этой величины приводит к увеличению риска ложного обнаружения потери блочной синхронизации, увеличение же приводит к неоправданно затянутому принятию решения о потере блочной синхронизации.

4. Сравнение с каскадными кодами

Как было указано во введении, кодовые конструкции можно сравнивать между собой по относительной скорости, по корректирующей способности и по сложности реализации. Для сравнительного анализа построенного кода 4B12B были выбраны каскадные коды БЧХ(15, 7) + манчестер и RS(15, 11) + манчестер. В этих кодах в качестве внутреннего кода для обеспечения равновесности и RLL используется манчестерское кодирование, а в качестве внешнего кода для коррекции ошибок — широко известные и подробно исследованные коды БЧХ(15, 7) и RS(15, 11). Коды БЧХ(15, 7) + манчестер и RS(15, 11) + манчестер имеют близкие значения относительной скорости – 0.367 и 0.233, в то время как код 4B12B – 0.333. Однако хотя коды БЧХ(15, 7) и RS(15, 11) и обладают близкими сопоставимыми характеристиками по корректирующей способности и сложности реализации, они вообще не имеют свойства блочной синхронизации, так что сравнение, очевидно, будет не вполне корректное.

Оценим объем аппаратных ресурсов, необходимых для реализации кодирования и декодирования кодом 4B12B. Благодаря малой мощности кода 4B12B реализация процедур кодирования и декодирования сообщений не представляет затруднений и требует пренебрежимо малых ресурсов FPGA. Для реализации табличного декодирования хорошо подходят блоки BRAM, входящие в состав FPGA. Декодер, реализованный с использованием BRAM, требует всего 2 блока BRAM16, входящих в состав FPGA Spartan 6, или 1 блок BRAM36, входящий в состав более поздних семейств FPGA.

Полностью параллельный декодер кода 4B12B требует 94 LUT FPGA Spartan 6 при декодировании только до 2 ошибок на слово или 183 LUT, если включить декодирование ещё и слов с 3 ошибками, которые можно однозначно декодировать к ближайшему кодовому слову (здесь и далее объем фактически затраченных аппаратных ресурсов приведен по результатам компиляции). При реализации существенным образом использовалось то обстоятельство, что полноценные сумматоры здесь не обязательны, а вычисление расстояния Хэмминга можно осуществлять только в рамках множества {0, 1, 2, много}, при этом значительно сокращается количество требующихся ресурсов FPGA.

Последовательно-параллельный декодер кода 4B12B требует 73 FF и 108 LUT и строится на использовании того факта, что инверсия кодового слова тоже является кодовым словом, поэтому можно хранить только одно слово из пары. Аналогичное свойство проявляется в отношении 12 слов с тремя ошибками, которые допускают однозначное декодирование. Это позволило вдвое сократить количество хранимых данных. Таким образом, если данные поступают последовательно по 1 биту за такт, то за 9 тактов можно успеть сопоставить принятое слово со всеми

кодовыми словами по 2 кодовых слова за такт, и декодирование осуществляется со скоростью поступления данных.

Сравним теперь коды по корректирующей способности. Для двоичного симметричного канала [3, 6, 8] с вероятностью ошибки бита p, $0 \le p \le 1$ вероятность успешного декодирования слова для кода 4B12B составляет:

$$s(p) = C_{12}^0 \cdot (1-p)^{12} + C_{12}^1 \cdot p \cdot (1-p)^{11} + C_{12}^2 \cdot p^2 \cdot (1-p)^{10} + 12 \cdot p^3 \cdot (1-p)^9, \quad (5)$$

а вероятность ошибочного декодирования слова (word error rate):

$$\mathrm{WER}(p) = 1 - s(p). \quad (6)$$

Для проведения сравнения эффективности кода 4B12B с каскадными кодами, построенными из кода БЧХ(15, 7) поверх манчестерского и кода Рида–Соломона (15, 11) поверх манчестерского, была написана программа на языке C, результаты которой представлены на графике (рис. 1). Для каждого кода программно эмулировалась передача 10^8 кодовых слов по каналу связи с заданной вероятностью ошибки бита $0 \le p \le 1$, а значение $\mathrm{WER}(p)$ оценивалось как отношение количества неправильно декодированных слов к общему количеству переданных. Результат вычислительного эксперимента для кода 4B12B показал прекрасное совпадение с теоретически полученным выражением для $\mathrm{WER}(p)$.

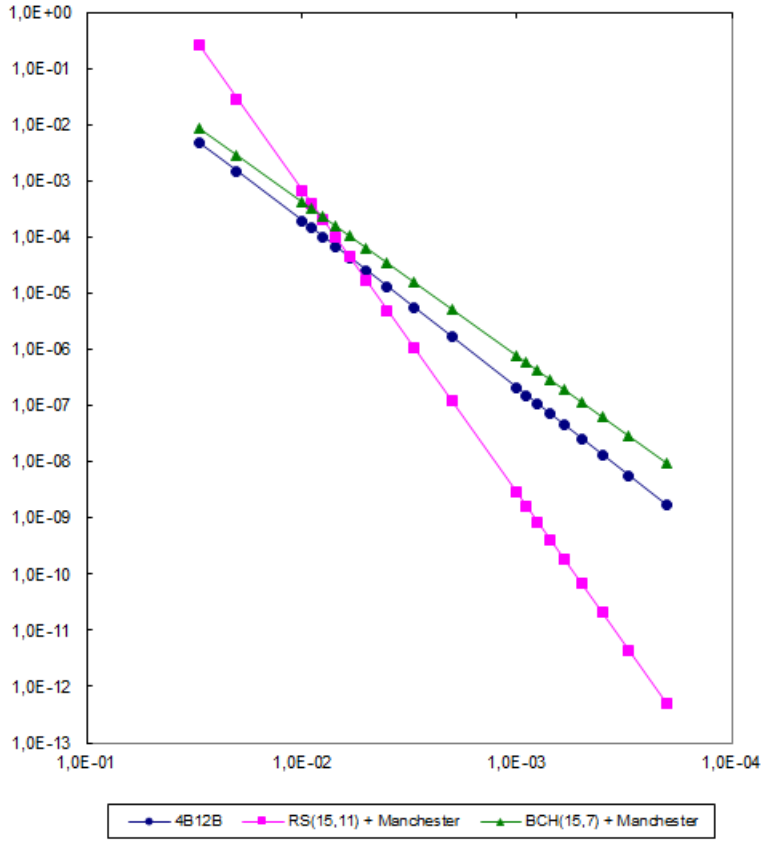

Рис. 1. График вероятности ошибочного приёма слова в зависимости от вероятности ошибки бита в канале.

Видно, что код RS(15,11)+манчестер за счёт значительно большей длины блока (120 бит) имеет преимущество перед кодом 4B12B при вероятностях ошибки менее 0.005 и даже имеет чуть более высокую относительную скорость: 0.367 против 0.333 у 4B12B, но при этом декодер

кода RS(15, 11) требует гораздо больших вычислительных ресурсов, чем декодер 4B12B. Декодер RS(15, 11) со стираниями версии 7.1 из Xilinx ISE 14.7 требует 310 FF и 269 LUT, при этом имеет задержку обработки, равную 45 тактов, так что для обработки непрерывного потока потребуется использовать три таких декодера или втрое ограничивать скорость передачи данных по сравнению с максимально возможной для приёма и обработки в FPGA.

По сравнению с кодом БЧХ(15, 7) + манчестер код 4B12B имеет некоторое преимущество (примерно в 2 раза) в вероятности ошибочного приёма слова и даже чуть более высокую относительную скорость: 0.333 против 0.233 у БЧХ(15, 7) + манчестер.

Литература

1. *Deng R.H., Herro M.A.* DC-free coset codes // IEEE Transactions on Information theory. 1988. V.34. P. 786–792.
2. *Wadayama T., Han Vinck A. J.* DC-Free Binary Convolutional Coding // IEEE Transactions on Information theory. 2002. V.48, №1, P. 162–172.
3. *Скляр Б.* Цифровая связь. Теоретические основы и практическое применение. Второе издание, исправленное. М.: Вильямс, 2003.
4. *Mercier H., Bhargava V. K., Tarokh V.* A survey of error-correcting codes for channels with symbol synchronization errors // IEEE Communications Surveys & Tutorials. 2010. V.12, №1, P. 87–96.
5. *Widmer A. X., Franaszek P. A.* A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code // IBM Journal of Research and Development. 1983. V.27, №5, P. 440–451.
6. *Хэмминг Р. В.* Теория кодирования и теория информации. Перевод с английского С. И. Гельфанда под редакцией Б. С. Цыбакова. М.: Радио и связь, 1983.
7. *Левенштейн В. И.* Оценки для кодов, обеспечивающих исправление ошибок и синхронизацию // Проблемы передачи информации. 1969. Т. 5, В. 2, с. 3–13.
8. *Питерсон У., Уэлдон Э.* Коды, исправляющие ошибки. Перевод с английского под редакцией Р. Л. Добрушина и С. И. Самойленко. М.: МИР, 1976.

Статья поступила в редакцию 20.12.2021;
переработанный вариант – 28.02.2022.

Мачикина Елена Павловна
к.ф.-м.н., доцент; доцент кафедры прикладной математики и кибернетики СибГУТИ, (630102, Новосибирск, ул. Кирова, 86), тел. (383) 269-82-72, e-mail: machikina@sibguti.ru.

Насенник Виталий Геннадьевич
e-mail: vitaly.nasennik@gmail.com, ORCID: 0000-0002-7654-6953.

References

1. Deng R. H., Herro M. A. DC-free coset codes. *IEEE Transactions on Information Theory*. 1988, vol. 34, pp. 786-792.
2. Wadayama T., Han Vinck A. J. DC-Free Binary Convolutional Coding. *IEEE Transactions on Information Theory*. 2002, vol. 48, No. 1, P. 162-172.
3. Skljar B. *Cifrovaja svjaz'. Teoreticheskie osnovy i prakticheskoe primenenie*. [Digital communication. Theoretical foundations and practical application]. Second edition, revised. Moscow: Williams, 2003.
4. Mercier H., Bhargava V. K., Tarokh V. A survey of error-correcting codes for channels with symbol synchronization errors. *IEEE Communications Surveys & Tutorials*. 2010, vol. 12, no. 1. pp. 87-96.

5. Widmer A. X., Franaszek P. A. A DC-Balanced, Partitioned-Block, 8B/10B Transmission Code. *IBM Journal of Research and Development*. 1983, vol. 27, no. 5, pp. 440–451.
6. Hjemming R. V. *Teorija kodirovanija i teorija informacii. Perevod s anglijskogo S. I. Gel'fanda pod redakciej B. S. Cybakova* [Coding theory and information theory. Translation from English]. Moscow: Radio and communication, 1983.
7. Levenshtejn V. I. *Ocenki dlja kodov, obespechivajushhih ispravlenie oshibok i sinhronizaciju* [Estimates for codes providing error correction and synchronization]. Problemy peredachi informacii [Problems of Information Transmission]. 1969, vol. 5, no. 2. pp. 3-13.
8. Piterson U., Ujeldon Je. *Kody, ispravljajushhie oshibki. Perevod s anglijskogo pod redakciej R. L. Dobrushina i S. I. Samojlenko* [Error-correcting codes. Translation from English]. Moscow: Mir, 1976.

A Construction of Synchronizable DC-balanced Error-Correcting Codes

Elena P. Machikina
Candidate of phisical and mathematical sciences, Siberian State Universiry of Telecommunications and Information Sciences (SibSUTIS, Novosibirsk, Russia), machikina@sibguti.ru.

Vitaly G. Nasennik
ORCID: 0000-0002-7654-6953, vitaly.nasennik@gmail.com.

The paper proposes a construction of error-correcting block codes with synchronization in packet serial data transmission over a noisy channel, which has a low complexity of encoder and decoder implementations. A study of corrective abilities of constructed codes in comparison with some concatenated codes is presented.

Keywords: DC-free codes, DC-balanced codes, synchronizable codes, forward error correction, RLL codes.