

*ФЕДОРЕЧКО О.И.,
ВИНОГРАДОВ Ю.Н.,
ИВАНОВ А.Н.*

МЕТОД ИСПРАВЛЕНИЯ “ПАЧКИ” ОШИБОК В КАНАЛАХ С КОДОВО-ИМПУЛЬСНОЙ МОДУЛЯЦИЕЙ НА ОСНОВЕ УМНОЖЕНИЯ БЕЗ МЕЖРАЗРЯДНЫХ ПЕРЕНОСОВ

В статье предложен метод исправления “пачки” ошибок в каналах с импульсно-кодовой модуляцией. Метод основан на математической операции двоичного умножения без переносов. Предложенный метод позволяет ускорить и уменьшить вычислительную сложность процедуры коррекции как по сравнению с кодами Рида-Соломона, так и по сравнению с известными кодами на основе взвешенных контрольных сумм. Вначале приведено математическое обоснование предложенного метода. Подробно изложена процедура коррекции пачки ошибок. Приведен численный пример. Выполнен аналитическое сравнение сложности предложенного и известного методов, которое показывает достигаемые преимущества. Предлагаемый способ может найти широкое применение в современных системах передачи данных реального времени.

In article the method for burst errors correction in channels with pulse-code modulation is proposed. The method is based on the mathematical operation of binary multiplication without carry. The offered method permits accelerated performance with reduced computational complexity as compared to Reed – Solomon codes, as compared to known weighed check sum codes while using the same number of bits for error control. The mathematical background of the proposed method is first presented. The procedure for burst errors is then explained. A numerical example for procedure is given. An analytical comparison of the complexity of both the proposed and the known methods is presented, that demonstrates the improvements attained. The proposed method can find wide application in modern real time data transmission.

Введение

В современных условиях динамичного роста скорости передачи данных, важное значение приобретает время, затрачиваемое на контроль возникающих ошибок. В большинстве систем контроль и исправление ошибок должно производиться без задержек, в темпе передачи данных. Особенно остро проблема быстрой реализации операций контроля стоит для каналов передачи данных с кодово-импульсной модуляцией. Такие каналы широко используются для обмена данными в компьютерных системах. Операции, связанные с контролем и коррекцией ошибок передачи данных выполняются в таких каналах встроенными аппаратными средствами. Для быстрого и эффективного контроля ошибок, схема таких средств должна быть возможно проще. Сложность схемы зависит от метода обнаружения и исправления ошибок передачи данных, а также от типа ошибок и их кратности.

Быстрое развитие технологий обмена данными меняет характер возникающих в каналах ошибок. Если, несколько десятилетий назад основной причиной ошибок были явления гауссова шума [77], то в настоящее время основными негативными факторами в плане возникновения

ошибок передачи данных становятся внешние помехи [55]. Это обусловлено расширением использования беспроводных технологий передачи данных и мобильной связи. Меняется и характер внешних помех: возрастает их длительность. Это приводит к тому, что они воздействуют на несколько информационных сигналов. Вследствие этого, доминирующим типом ошибок становятся “пачки” битовых искажений [99].

Таким образом, в современных условиях, задача создания метода коррекции “пачки” ошибок, обеспечивающего эффективную аппаратную реализацию встроенных средств контроля передачи цифровых данных в каналах с кодово-импульсной модуляцией является важной и актуальной.

Анализ известных методов коррекции “пачки” ошибок

Значительная практическая значимость задачи коррекции “пачки” ошибок в каналах с кодово-импульсной модуляцией стимулировала создание ряд методов ее решения.

Наиболее известный подход к коррекции “пачек” ошибок связан с использованием корректирующих кодов на основе циклических ко-

дов [1]. Корректирующие коды других классов (коды Хемминга, коды Голлея и др.) ориентированы на исправление произвольно локализованного множества искаженных битов, количество их разрядов с увеличением кратности ошибок растет экспоненциально и поэтому они изначально проигрывают циклическим кодам при решении задачи коррекции “пачки” ошибок.

Специально для коррекции “пачки” ошибок на основе циклических кодов созданы коды Файра, коды Миласа, коды Абрамсона и коды Рида-Соломона [1]. Во всех этих кодах контрольный код образуется как результат деления полинома информационного сообщения на образующий код полином. Фактически приведенные коды отличаются способом формирования образующего полинома.

На практике наибольшее распространение получили коды Рида-Соломона.

Коды Рида-Соломона представляют собой недвоичные циклические коды, предусматривающие разделение контролируемого информационного блока на символы. Эти коды позволяют исправлять любые искажение в h символах с использованием $2 \cdot h$ контрольных символов. Поскольку положение символов жестко фиксировано, а m -битовая “пачка ошибок” может возникнуть, начиная с любого из битов передаваемого блока, то для гарантированного ее исправления, код Рида-Соломона должен обеспечивать возможность коррекции 2-х символов ($h=2$) длиной m бит, то есть использовать 4 контрольных символа или $4 \cdot m$ контрольных разряда.

Кодирование, то есть вычисление контрольных разрядов на передатчике, состоит в нахождении остатка от деления n -символьного информационного блока на 4-символьный полином. Операция деления состоит из n циклов, на каждом из которых выполняется 4 операции умножения символов и одна операция сложения. Операция умножения на полях Галуа m -разрядных символов включает m циклов, в рамках каждого из которых, в среднем, выполняется две операции сдвига и одна операция суммирования над k -битовыми числами. Исходя из этого, время T_{C1} кодирования определяется в виде:

$$T_{C1} = 4 \cdot n \cdot (3 \cdot k \cdot t_{\bar{e}} + t_{\bar{e}}), \quad (1)$$

где $t_{\bar{e}}$ – время выполнения логической операции над k -разрядным символом.

Декодирование кодов Рида-Соломона включает два этапа: обнаружение ошибок и их исправление. Для обнаружения ошибок вычисляется 4 k -разрядных синдромов. Если значения всех синдромов равно нулю, то полагается, что сообщение передано без ошибок. При вычислении каждого из синдромов производится n операций умножения и такое же количество операций сложения. Поэтому, время T_{O1} обнаружения ошибок практически совпадает со временем кодирования.

Исправление ошибок передачи производится следующим порядком: вначале решается система из 4-х символьных уравнений для определения компонент локатора ошибок; методом перебора определяются позиции искаженных символов; решается система из 4-х символьных уравнений для определения символов искажений. Общее время T_{S1} коррекции определяется формулой:

$$T_{S1} \approx 4 \cdot n \cdot (3 \cdot m \cdot t_{\bar{e}} + t_{\bar{e}}). \quad (2)$$

Из изложенного следует, что коррекция “пачки” из 2-х искаженных символов в коде Рида-Соломона требует значительного времени и достаточно сложных аппаратных решений, поскольку последние должны решать несколько разнородных математических задач.

Указанные недостатки обусловлены универсальным характером кодов Рида-Соломона, который не учитывает особенностей возникновения одиночной “пачки” ошибок в каналах с кодово-импульсной модуляцией.

Для повышения эффективности коррекции “пачки” ошибок было предложено использовать контрольный код на основе арифметической взвешенной контрольной суммы (АВКС) [2]. Этот контрольный код использует для кодирования операции арифметического умножения и сложения, а для коррекции ошибок – операцию целочисленного арифметического деления. Это позволяет существенно уменьшить время коррекции, сделав его независимым от n -числа символов в блоке. Однако при аппаратной реализации вычисление арифметических операций требует много времени, поскольку необходимо учесть последовательный характер формирования переносов. Кроме того, контрольные коды на основе АВКС требуют большего, по сравнению с кодами Рида-Соломона числа контрольных разрядов.

Целью исследований является разработка метода коррекции одиночной “пачки” ошибок в каналах с кодово-импульсной модуляцией,

обеспечивающего упрощение аппаратной реализации и ускорение выполнения процедур контроля ошибок и их исправления.

Метод исправления «пачки» ошибок на основе умножения без переносов

Предлагаемый метод использует логические операции и ориентирован на эффективную аппаратную реализацию. В основе метода лежит использование взвешенной контрольной суммы, вычисление которой выполняется с применением операции умножения без межразрядных переносов. Эта операция ниже обозначается знаком \otimes .

Произведение без межразрядных переносов $D=U \otimes V = \{d_{2^r}, \dots, d_3, d_2, d_1\} = d_1 + 2 \cdot d_2 + 4 \cdot d_3 + \dots + 2^{2^r} \cdot d_{2^r}$, $\forall l \in \{1, \dots, 2^r\}$: $d_l \in \{0, 1\}$ двух r -разрядных чисел $U = \{u_r, \dots, u_2, u_1\} = u_1 + 2 \cdot u_2 + \dots + 2^r \cdot u_r$ и $V = \{v_r, \dots, v_2, v_1\} = v_1 + 2 \cdot v_2 + \dots + 2^r \cdot v_r$, $\forall i \in \{1, \dots, r\}$: $v_i, u_i \in \{0, 1\}$ вычисляется следующим образом:

$$D = U \cdot v_1 \oplus (U \cdot v_2) \lll 1 \oplus \dots \oplus (U \cdot v_r) \lll (r-1), \quad (3)$$

где \cdot – операция логического умножения, $p \lll q$ – операция логического сдвига числа p влево на q разрядов.

Если задана максимальная длина m «пачки» битовых искажений, то информационный N -битовый блок $B = \{b_1, b_2, \dots, b_N\}$, $\forall l \in \{1, \dots, N\}$: $b_l \in \{0, 1\}$ предлагается рассматривать состоящим из $n = N/m$ символов $B = \{X_1, X_2, \dots, X_n\}$.

Поскольку длина пачки ограничена m битами, то она может исказить непересекающиеся подмножества битов только двух смежных символов X_i и X_{i+1} , $i \in \{1, \dots, n-1\}$.

Для исправления такого вида ошибок предлагается использовать контрольную сумму, которая состоит из трех компонент:

1) Первая C_1 – сумма по модулю два символов, с нечетными номерами в блоке:

$$C_1 = X_1 \oplus X_3 \oplus \dots \oplus X_{n-1}. \quad (4)$$

2) Компонента C_2 – сумма по модулю два символов, которые находятся на четных позициях:

$$C_2 = X_2 \oplus X_4 \oplus \dots \oplus X_n. \quad (5)$$

3) Третья компонента C_3 представляет собой сумму по модулю два произведений без межразрядных переносов символов блока на коды их порядковых номеров:

$$C_3 = (1 \otimes X_1) \oplus (2 \otimes X_2) \oplus \dots \oplus (n \otimes X_n). \quad (6)$$

Компоненты контрольной суммы C_{1s} , C_{2s} та C_{3s} для информационного блока B вычисляются на передатчике. Порядок передачи пакета данных приемнику следующий: третья, первая и вторая компоненты контрольного кода, а затем информационный блок B : C_{3s} , C_{1s} , C_{2s} , B .

На приемнике по полученному блоку данных B вычисляются компоненты контрольного кода C_{1R} , C_{2R} , C_{3R} и разности Δ_1 , Δ_2 , Δ_3 компонент контрольного кода приемника и передатчика:

$$\begin{aligned} \Delta_1 &= C_{1S} \oplus C_{2R}, \\ \Delta_2 &= C_{2S} \oplus C_{2R}, \\ \Delta_3 &= C_{3S} \oplus C_{3R}. \end{aligned} \quad (7)$$

По значениям Δ_1 , Δ_2 и Δ_3 определяется наличие ошибки в принятом информационном блоке: если $\Delta_1=0$, $\Delta_2=0$ и $\Delta_3=0$, то принятый информационный блок B не содержит ошибок, в противном случае выполняется классификация типа возникших при передаче ошибок.

Перечень возможных вариантов искажений, вызванных «пачкой» ошибок приведен ниже:

1) $\Delta_1 \neq 0$, $\Delta_2 = 0$ и $\Delta_3 \neq 0$ и при этом $\Delta_3 > \Delta_1$ – «пачка» ошибок исказила один информационный символ, который находится на нечетной позиции.

2) $\Delta_1 = 0$, $\Delta_2 \neq 0$ и $\Delta_3 \neq 0$ – «пачка» ошибок исказила один информационный символ, который находится на четной позиции.

3) $\Delta_1 \neq 0$, $\Delta_2 \neq 0$ и $\Delta_3 \neq 0$, $\Delta_1 < \Delta_2$ – «пачка» ошибок исказила два смежных символа, причем информационный символ, который находится на нечетной позиции был искажен первым.

4) $\Delta_1 \neq 0$, $\Delta_2 \neq 0$ и $\Delta_3 \neq 0$, $\Delta_1 > \Delta_2$ и при этом $\Delta_3 > \Delta_1$ и $\Delta_3 \neq \Delta_1$ – «пачка» ошибок исказила два смежных символа, причем информационный символ, который находится на четной позиции был искажен первым.

5) $\Delta_1 \neq 0$, $\Delta_2 \neq 0$ и $\Delta_3 \neq 0$, при чем $\Delta_3 = \Delta_1$, то «пачка» ошибок исказила первый символ информационного блока и вторую компоненту контрольного кода.

6) $\Delta_1 \neq 0$, $\Delta_2 = 0$ и $\Delta_3 = 0$ – «пачка» исказила первую компоненту контрольного кода.

7) $\Delta_1 = 0$, $\Delta_2 \neq 0$ и $\Delta_3 = 0$ – «пачка» исказила вторую компоненту контрольного кода.

8) $\Delta_1 = 0$, $\Delta_2 = 0$ и $\Delta_3 \neq 0$ – «пачка» исказила третью компоненту контрольного кода.

9) $\Delta_1 \neq 0$, $\Delta_2 \neq 0$ и $\Delta_3 = 0$ – «пачка» исказила первую и вторую компоненту контрольного кода.

10) $\Delta_1 \neq 0, \Delta_2 = 0, \Delta_3 \neq 0$ и при этом $\Delta_3 < \Delta_1$ – “пачка” исказила первую и третью компоненту контрольного кода.

В том случае, если после анализа информационного блока на наличие в нем ошибок, была обнаружена ошибка в информационном блоке, то процедура коррекции в этом случае состоит из следующих действий: определения номера первого искаженного символа и выполнения коррекции. В предложенном методе для выполнения эффективной коррекции используется информация, содержащаяся в компоненте Δ_3 : в случае возникновения “пачки” ошибок в ее формировании принимают участие номера позиций искаженных символов в информационном блоке (P_i – номер первого символа искаженного “пачкой” ошибок, а P_{i+1} – номер второго). Если, “пачка” ошибок исказила информационный блок начиная с символа находящегося на нечетной позиции ($\Delta_1 < \Delta_2$), то количество ненулевых разрядов в компоненте Δ_1 указывают сколько раз при формировании компоненты Δ_3 выполнялась сумма по модулю два номера первого искаженного символа, а позиции ненулевых значений в компоненте Δ_1 указывают на сколько разрядов влево были сдвинуты значения номера первого искаженного символа, которые берут участие при формировании компоненты Δ_3 . По количеству единиц в компоненте Δ_2 можно определить количество раз, которое номер второго искаженного символа участвует при формировании компоненты Δ_3 , а номера позиций единиц в компоненте Δ_2 определяют на сколько разрядов участвующие значения номера второго искаженного символа сдвинуты влево от начала. Так, например, при количестве разрядов $k=4$ и значениях компонент $\Delta_1=0101_2$ и $\Delta_2=1000_2$ видно, что при формировании компоненты Δ_3 номера искаженных символов принимали участие в порядке, показанном на рис.1. Так, как в компоненте Δ_1 два ненулевых разряда, это значит, что номер первого искаженного символа принимал участие в формировании разрядов компоненты Δ_3 , два раза, причем первый раз без сдвига (так, как ненулевым является младший разряд компоненты Δ_1), а во второй раз со сдвигом на два разряда влево (так, как ненулевым является третий разряд компоненты Δ_1); из компоненты Δ_2 видно, что номер второго искаженного символа принимал участие в формировании компоненты Δ_3 один раз, причем, он был сдвинут

влево от начала на три разряда (так как ненулевым является четвертый разряд компоненты Δ_2). Если сначала был искажен символ, находящийся на четной позиции, то по компоненте Δ_2 , определяется порядок участия номера первого искаженного символа при формировании компоненты Δ_3 , а по компоненте Δ_1 – второго номера искаженного символа.

Для дальнейшего объяснения алгоритма коррекции вводятся следующие условные обозначения:

- $P_i[j]$ – значение j -го разряда номера первого искаженного символа $j \in \{0.. \log_2 n\}$;
- $P_{i+1}[j]$ – значение j -го разряда номера второго искаженного символа $j \in \{0.. \log_2 n\}$;
- $C[j]$ – значение j -го переменной межразрядных переносов $j \in \{0.. \log_2 n\}$;
- $\Delta_1[j]$ – значение j -го разряда номера первой компоненты контрольного кода $j \in \{0..k\}$;
- $\Delta_2[j]$ – значение j -го разряда номера второй компоненты контрольного кода $j \in \{0..k\}$;
- $\Delta_3[j]$ – значение j -го разряда номера третьей компоненты, $j \in \{0.. \log_2 n + k - 1\}$;
- r – индекс строки матрицы;
- st – индекс столбца матрицы;

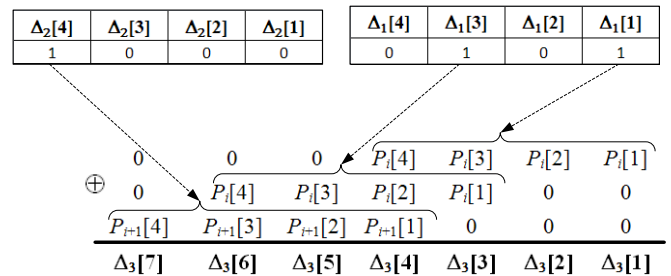


Рис. 1. Пример формирования разрядов Δ_3

Поскольку размер “пачки” ошибок всегда меньше или равен разрядности символа информационного блока, то при искажении двух смежных символов она не может исказить их в одинаковых разрядах. Таким образом, при формировании компоненты Δ_3 номера первого и второго искаженного символа организуют ступенчатую структуру при вычислении разрядов компоненты Δ_3 , с возможностью поразрядного определения значений номеров искаженных символов P_i и P_{i+1} . Символы информационного блока, то следует отметить еще одну особенность, которая вытекает из очевидного факта того, что значение следующего номера искаженного символа P_{i+1} можно получить, если прибавить единицу к номеру символа P_i и при

этом значение их разрядов отличаются на значение переменной C которая содержит значения межразрядных переносов при выполнении суммы по модулю два P_i с P_{i+1} : $C = P_i \oplus P_{i+1}$. Следовательно, значение номера второго искаженного символа P_{i+1} может быть выражено через значение номера первого искаженного символа P_i и значения межразрядных переносов C :

$$P_{i+1} = P_i \oplus C. \quad (8)$$

Учитывая описанные выше особенности, и ступенчатый характер формирования компоненты Δ_3 была разработана процедура нахождения номера первого искаженного «пачкой» ошибок символа.

Процедура нахождения номера первого искаженного символа состоит из двух этапов:

1) Определение позиции младшего искаженного «пачкой» ошибок разряда в первом искаженном символе P_i .

2) Восстановление значений разрядов весового коэффициента.

Первый этап алгоритма, состоит в определении позиции самого младшего искаженного «пачкой» ошибок разряда в первом искаженном символе, его необходимо найти для того, чтобы правильно локализовать местонахождение начала номера первого искаженного символа P_i , который сдвинутый на наименьшее количество разрядов влево относительно начала, при вычислении компоненты Δ_3 . Для этого необходимо определить: где находится начало «пачки» ошибок, а именно в символе с четным или с нечетным порядковым номером в информационном блоке. Если значение Δ_1 меньше значения Δ_2 и при этом $\Delta_1 \neq 0$, то начало пачки ошибок находится в символе с нечетным значением номера позиции в информационном блоке, иначе – в символе с четным значением. В первом случае анализируется значение компоненты Δ_1 начиная с младшего разряда, в другом – выполняется аналогичный анализ Δ_2 . Анализ выполняется до тех пор пока не будет найдено первый ненулевой разряд. Номер позиции этого разряда в информационном символе сохраняется в переменных r и st .

Второй этап алгоритма вычисления разрядов номера первого искаженного символа состоит в последовательности действий:

1) Начинается цикл поразрядного нахождения значения номера первого искаженного

символа P_i от 1 до $\log_2(n)$, счетчику присваивается значение ноль $a=0$.

2) Инициализация переменной $counter=r+1$.

3) Начинается цикл суммирования по модулю два разрядов номера первого искаженного символа P_i , которые формируют текущий разряд st компоненты Δ_3 . Цикл от $a-1$ до нуля включительно, значению j присваивается значение $a-1$.

3.1) Если $counter < k$, то переход на пп.3.2 иначе на пп. 4.

3.2) Если $\Delta_1[counter] \neq 0$ или $\Delta_2[counter] \neq 0$, то переход на пп. 3.3 иначе – на пп.3.4.

$$3.3) P_i[a] = P_i[a] \oplus P_i[j];$$

3.4) Если $\Delta_1 > \Delta_2$ и при этом $\Delta_1[counter] \neq 0$ или $\Delta_1 < \Delta_2$ и при этом $\Delta_2[counter] \neq 0$, то переход на пп. 3.5 иначе – на пп.4

$$3.5) P_i[a] = P_i[a] \oplus C[j];$$

4) Инкремент переменной $counter$.

5) Декремент переменной j .

6) Проверка окончания цикла: если $j=0$, то переход на пп 7. Иначе на пп. 3.1.

$$7) P_i[a] = P_i[a] \oplus \Delta_3[st].$$

8) Определение переноса и запись его в переменную C . Если $P_i[a]=1$ и $C[a]=1$, то фиксируется наличие переноса $C[a+1]=1$, иначе $C[a+1]=0$.

9) Инкремент переменной st .

10) Инкремент переменной a .

11) Проверка окончания цикла: если $a < \log_2(n)-1$, то переход на пп 2. Иначе -выход на конец процедуры, поскольку определение разрядов номера первого искаженного символа окончено.

После окончания процедуры определения номера первого искаженного символа производится коррекция искаженных символов информационного блока. Коррекция производится в зависимости от типа искажения, которое возникла в информационном блоке в результате воздействия внешней помехи. Перечень возможных искажений блока и соответствующие операции коррекции представлены ниже:

1) Если «пачка» ошибок исказила один информационный символ, который находится на нечетной позиции ($\Delta_1 \neq 0$, $\Delta_2 = 0$ и $\Delta_3 \neq 0$ и при этом $\Delta_3 > \Delta_1$), то коррекция проводится путем выполнения суммы по модулю два информационного символа находящегося на позиции P_i и компоненты Δ_1 .

2) Если “пачка” ошибок исказила один информационный символ, который находится на четной позиции ($\Delta_1=0, \Delta_2 \neq 0$ и $\Delta_3 \neq 0$), то коррекция проводится путем выполнения суммы по модулю два информационного символа находящегося на позиции P_i и компоненты Δ_2 .

3) Если “пачка” ошибок исказила два смежных символа, причем информационный символ, который находится на нечетной позиции был искажен первым ($\Delta_1 \neq 0, \Delta_2 \neq 0$ и $\Delta_3 \neq 0, \Delta_1 < \Delta_2$), то коррекция проводится путем выполнения суммы по модулю два информационного символа находящегося на позиции P_i и компоненты Δ_1 и выполнения суммы по модулю два информационного символа находящегося на позиции P_{i+1} и компоненты Δ_2 .

4) Если “пачка” ошибок исказила два смежных символа, причем информационный символ, который находится на четной позиции был искажен первым ($\Delta_1 \neq 0, \Delta_2 \neq 0$ и $\Delta_3 \neq 0, \Delta_1 > \Delta_2$ и при этом $\Delta_3 > \Delta_1$ и $\Delta_3 \neq \Delta_1$), то коррекция проводится путем выполнения суммы по модулю два информационного символа находящегося на позиции P_i и компоненты Δ_2 и выполнения суммы по модулю два информационного символа находящегося на позиции P_{i+1} и компоненты Δ_1 .

5) Если “пачка” ошибок исказила первый символ информационного блока и вторую компоненту контрольного кода ($\Delta_1 \neq 0, \Delta_2 \neq 0$ и $\Delta_3 \neq 0$, при чем $\Delta_3 = \Delta_1$), то коррекция проводится путем выполнения суммы по модулю два первого информационного символа P_1 и компоненты Δ_1 .

Например, $k=4$ при возникновении «пачки» ошибок искажаются 7-й и 8-й символы. Пусть на передатчике $X_{7S}=\{1,1,1,0\}$ и $X_{8S}=\{0,1,1,1\}$, а на приемнике $X_{7R}=\{1,0,0,0\}$ и $X_{8R}=\{1,1,1,1\}$. Соответственно $\Delta_1=\{0,1,1,0\}$ $\Delta_2=\{1,0,0,0\}$. $\Delta_3=(7 \ll 1) \oplus (7 \ll 2) \oplus (8 \ll 3) = 14 \oplus 28 \oplus 64 = 82_{10} = 1010010_2$.

Разряды номеров искаженных символов формирующие компоненту Δ_3 для этого примера представлены на рисунке 2.

	0	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0
\oplus	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0	0
	$P_{i+1}[4]$	$P_{i+1}[3]$	$P_{i+1}[2]$	$P_{i+1}[1]$	0	0	0
	$\Delta_3[7]$	$\Delta_3[6]$	$\Delta_3[5]$	$\Delta_3[4]$	$\Delta_3[3]$	$\Delta_3[2]$	$\Delta_3[1]$

Рис. 2. Пример формирования разрядов компоненты Δ_3

Проанализировав значения компонент Δ_1 и Δ_2 можно сделать вывод, что самый младший искаженный разряд находится на второй позиции

$r=2$ (при нумерации строк начиная с единицы). Следовательно, определение разрядов номера первого искаженного символа необходимо начинать из второй строки и второго столбца. Вычисление третьей компоненты контрольной суммы с учетом формулы 11 изображено на рисунке 3.

	0	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0
\oplus	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0	0
	$P_i[4] \oplus C[3]$	$P_i[3] \oplus C[2]$	$P_i[2] \oplus C[1]$	$P_i[1] \oplus 1$	0	0	0
	$\Delta_3[7]$	$\Delta_3[6]$	$\Delta_3[5]$	$\Delta_3[4]$	$\Delta_3[3]$	$\Delta_3[2]$	$\Delta_3[1]$

Рис. 3. Вычисление третьей компоненты контрольной суммы

Вычисление значений разрядов предлагается проводить начиная с того столбца st , который соответствует номеру первого искаженного разряда r . В приведенном примере необходимо начинать со второго столбца. Воспроизведение значения разрядов номера первого искаженного символа выполняется поразрядно, как это показано на рис.4.

	0	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0
\oplus	0	$P_i[4]$	$P_i[3]$	$P_i[2]$	$P_i[1]$	0	0
	$P_i[4] \oplus C[3]$	$P_i[3] \oplus C[2]$	$P_i[2] \oplus C[1]$	$P_i[1] \oplus 1$	0	0	0
	$\Delta_3[7]$	$\Delta_3[6]$	$\Delta_3[5]$	$\Delta_3[4]$	$\Delta_3[3]$	$\Delta_3[2]$	$\Delta_3[1]$

\downarrow $P_i[1] = \Delta_3[2]$
 \downarrow $P_i[2] = (\Delta_1[2] \cdot P_i[1]) \oplus \Delta_3[3]$
 \downarrow $P_i[3] = (\Delta_1[3] \cdot P_i[2]) \oplus (\Delta_2[4] \cdot (P_i[1] \oplus C[1])) \oplus \Delta_3[4]$
 \downarrow $P_i[4] = (\Delta_1[3] \cdot P_i[3]) \oplus (\Delta_2[4] \cdot (P_i[2] \oplus C[2])) \oplus \Delta_3[5]$

Рис. 4. Вычисление значений разрядов первого искаженного символа

На рисунке 4 видно, что разряды номера первого искаженного символа P_i , можно получить путем выполнения суммы по модулю два значений разрядов, которые находятся ниже него в столбце при формировании соответствующего значения разряда компоненты Δ_3 .

Пошаговое выполнение алгоритма вычисления номера первого искаженного символа представлено ниже.

Действия, которые выполняются на первом шаге:

$$P_i[1] = \Delta_3[2] = 1;$$

Так как соседние разряды точно отличаются на единицу, то есть возможность определить $P_{i+1}[1]$ и также можно определить $C[1]$ первый перенос;

$$P_{i+1}[1] = P_i[1] \oplus 1 = 0;$$

$C[1] = 1$, так как $P_i[1] = 1$, то для получения младшего разряда прибавлялась единица, а это значит, что перенос был.

Действия, которые выполняются на втором шаге:

Переход к следующему, 3-му столбцу.

$$P_i[2] = P_i[1] \oplus 1 \oplus \Delta_3[3] = 1 \oplus 1 \oplus 1 = 1$$

$$P_{i+1}[2] = P_i[2] \oplus C[1] = 1 \oplus 1 = 0;$$

$C[2]=1$, так как $P_i[1]=1$ и $C[1]=1$, то имел место перенос.

Действия, которые выполняются на третьем шаге:

Переход к следующему, 4-му столбцу.

$$P_i[3] = P_i[2] \oplus C[1] \oplus P_i[1] \oplus 1 \oplus \Delta_3[4] = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 1 = 1$$

$$P_{i+1}[3] = P_i[2] \oplus C[2] = 1 \oplus 1 = 0;$$

$C[3]=1$, так как $P_i[2]=1$ и $C[2]=1$, то имел место перенос.

Действия, которые выполняются на четвертом шаге:

Переход к следующему, 5-му столбцу.

$$P_i[4] = P_i[3] \oplus C[2] \oplus P_i[2] \oplus C[1] \oplus \Delta_3[5] = 1 \oplus 1 \oplus 1 \oplus 1 \oplus 0 = 0$$

$$P_{i+1}[4] = P_i[4] \oplus C[3] = 0 \oplus 1 = 1;$$

$C[4]=0$, так как $P_i[4]=0$, а $C[3]=1$, то переноса не было.

В таблице 1 представлены значения переменных после четвертого шага алгоритма.

Табл. 1. Значения переменных P_i, P_{i+1} .

№ разряда \ Переменные	4	3	2	1
P_i	0	1	1	1
P_{i+1}	1	0	0	0
C	0	1	1	1

Определение номеров искаженных символов закончено: $P_i=7$ и $P_{i+1}=8$. коррекция проводится путем выполнения суммирования по модулю символа, находящегося на позиции P_i с значением первой компоненты Δ_1 , и символа, находящегося на позиции P_{i+1} и второй компоненты Δ_2 , $X_{7R} \oplus \Delta_1 = 1000_2 \oplus 1000_2 = 1110_2$ та $X_{8R} \oplus \Delta_2 = 1111_2 \oplus 1000_2 = 0111_2$. Так, как значения символов $X_{7R}=1110_2$ и $X_{8R}=0111_2$ после коррекции совпадают с значениями соответствующих символов на передатчике $X_{7S}=1110_2$ и $X_{8S}=0111_2$, то можно сделать вывод что коррекция была проведена успешно.

Анализ эффективности предложенного метода

Число k контрольных разрядов, используемых в предложенном методе коррекции одной «пачки» ошибок в каналах с кодово-импульсной модуляцией определяется суммарной разрядностью всех 3-х компонент контрольного

кода. Учитывая, что разрядность двух первых компонент равна k , а разрядность третьей: $k + \lfloor \log_2 n \rfloor$, число контрольных разрядов d определяется в виде:

$$d = 3 \cdot k + \lfloor \log_2 n \rfloor. \tag{9}$$

В отличие от кодов Рида-Соломона, которые накладывают ограничение на число n символов информационного блока: $2^k \leq n$, в предлагаемом методе число символов может быть любым. Это позволяет эффективно использовать метод для коррекции ошибок при передаче блоков большой длины.

При использовании кодов Рида-Соломона, наибольшая эффективность использования контрольных разрядов достигается при $2^k=n$. Если принять это условие для предлагаемого метода, то формула (9) преобразуется к виду: $d=4 \cdot k$. Это означает, что предлагаемый метод и коды Рида-Соломона для решения одинаковой задачи - коррекции одиночной «пачки» ошибок в каналах с кодово-импульсной модуляцией используют одинаковое количество контрольных разрядов.

Основным преимуществом предлагаемого метода коррекции «пачки» ошибок является ускорение выполнения операций кодирования, декодирования и исправления ошибок. Сравнение времен выполнения основных операций, связанных с контролем и исправлением «пачки» ошибки, для предложенного и известных методов приведено в таблице 2.

Табл. 2. Сравнительные характеристики времени выполнения операций

Метод коррекции «пачки» ошибок	Время кодирования	Время декодирования и анализа синдрома	Время коррекции «пачки» ошибок
Предложенный метод	$2 \cdot n \cdot k \cdot t_{\text{л}}$	$2 \cdot n \cdot k \cdot t_{\text{л}}$	$\frac{k^2}{4} \cdot t_{\text{л}}$
Метод на основе АВКС	$4 \cdot n \cdot k^2 \cdot t_{\text{л}}$	$4 \cdot n \cdot k^2 \cdot t_{\text{л}}$	$k^2 \cdot t_{\text{л}}$
Коды Рида-Соломона	$12 \cdot n \cdot k^2 \cdot t_{\text{л}}$	$12 \cdot n \cdot k^2 \cdot t_{\text{л}}$	$12 \cdot n \cdot k \cdot t_{\text{л}}$

Из таблицы 2 следует, что предложенный метод фактически обеспечивает ускорение в k раз критической по времени операции кодирования. Критической, поскольку кодирование должно происходить в темпе передачи данных. Время коррекции «пачки» по сравнению с арифметической взвешенной контрольной суммой уменьшается в 4 раза, а в сравнении с ко-

дами Рида-Соломона на порядки, поскольку количество символов – n на порядки превышает их разрядность – k .

Выводы

Разработан метод обнаружения и коррекции “пачки” ошибок, вызванных внешней помехой в каналах с кодово-импульсной модуляцией. Основным достоинством предложенного метода по сравнению с существующими способами коррекции “пачки” ошибок является малая вычислительная сложность реализации контроля и исправления ошибок. Проведенный теоретический анализ и результаты моделирования показали, что по сравнению с кодами Рида-Соломона, предложенный метод обеспечивает примерно в 8 раз большую скорость кодирова-

ния-декодирования и на 2 порядка меньшее время, требующееся на исправления “пачки” битовых искажений при одинаковом количестве контрольных разрядов.

Эффективность предложенного метода обусловлена тем, что он, в отличие от универсальных методов, учитывает особенности возникновения ошибок в каналах с кодово-импульсной модуляцией и, фактически, является специализированным. Метод ориентирован на аппаратную реализацию контроля ошибок. Метод может быть эффективно использован в скоростных каналах обмена информацией между компонентами компьютерных систем управления, для которых основной причиной ошибок являются внешние помехи.

Список литературы

1. Скляр Б. Цифровая связь. Теоретические основы и практическое применение. М.: Издательский дом “Вильямс”.- 2004.- 1104 с.
2. Турченко Ю. А. Использование арифметических взвешенных контрольных сумм для коррекции ошибок в каналах со спектральной модуляцией // Вісник національного технічного університету України “КПІ”. Інформатика, управління та обчислювальна техніка. – 2009.- № 50.- С. 106-112.
3. Федоречко О.І. Корекція “пачки помилок” в каналах з імпульсно-ковою модуляцією // Матеріали XIII-ї Міжнародної науково-технічної конференції “Системний аналіз та інформаційні технології” – Київ, НТУУ “КПІ”, ІПСА.– 2011. – с. 516.