

УДК 621.391

А. В. ДИКАРЕВ, канд. техн. наук, доцент,  
Государственный университет телекоммуникаций, Киев

## СЖАТИЕ ДВОИЧНЫХ БЛОЧНЫХ КОДОВ

**Предложен алгоритм устранения избыточности закодированной двоичной информации при ее передаче, обеспечивающий в дальнейшем полное ее восстановление благодаря выполнению всех шагов данного алгоритма в обратном порядке.**

**Ключевые слова:** двоичный блочный код; избыточность; натуральный ряд чисел; сжатие; сигнальная последовательность.

### Введение

Каждое кодовое слово двоичного блочного кода можно представить в виде некоторого натурального (целого положительного) числа в десятичной системе счисления. Любое натуральное десятичное число можно, в свою очередь, поставить в соответствие некоторому отрезку (конечной последовательности натуральных чисел) натурального ряда, заканчивающемуся данным числом. Используя свойства делимости натуральных чисел [1; 2] и «прореживая» исходный отрезок натурального ряда «выкалыванием», т. е. удалением из него чисел, кратных, например, числам 2 и 3, осуществляем поэтапное сжатие исходного отрезка, освобождая его от избыточной информации, а затем восстанавливая в первоначальном виде.

### Основная часть

Итак, выкалывая из найденного указанным способом отрезка все нечетные (либо все четные) числа, получаем отрезок, содержащий в два раза меньше чисел, чем первоначальный. Из оставшегося прореженного отрезка снова выкалываются все нечетные числа. Аналогично действуем до тех пор, когда очередной отрезок уже невозможно будет сжать.

Описанный процесс иллюстрируют табл. 1 и 2, построенные по следующему принципу.

В верхней слева ячейке каждой таблицы расположен отрезок натурального ряда, соответствующий исходному числу.

Например, в табл. 1 указанное число равно 59. Оно представлено в виде отрезка натурального ряда в упомянутой ячейке, а в верхней справа ячейке представлен тот же отрезок, но из него выколоты все четные числа. Поскольку исходное число нечетное, оставшийся после этой операции отрезок ряда нечетных чисел будет содержать на один элемент больше, чем их было в отрезке четных чисел, причем общее их количество будет четным, поскольку  $(59 + 1):2 = 30$ . Во второй слева ячейке число 30 представляется отрезком натурального ряда, а во второй справа ячейке из него удалены все четные числа. Теперь после второго этапа сжатия чисел осталось 15. Число это оказалось нечетным, а поэтому, как и в первом случае, после удаления всех четных чисел получим отрезок ряда, содержащий 8 чисел:  $(15 + 1):2 = 8$ . Число 8 четное, и после удаления из соответствующего отрезка всех четных чисел также получим четное число. На последнем этапе количество оставшихся чисел равно 2.

Восстановление исходного числа 59 осуществляется в обратном порядке.

Пример сжатия четного числа 58, приведенный в табл. 2, показывает, что число, получаемое после очередного этапа сжатия, может быть как четным, так и нечетным, и чтобы правильно восстановить исходное число, потребуется некоторая дополнительная информация.

Таблица 1

Этапы сжатия числа 59 — нечетного

Отрезок натурального ряда	Количество оставшихся чисел	Результат сжатия
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59	30	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30	15	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	8	1 3 5 7 9 11 13 15
1 2 3 4 5 6 7 8	4	1 3 5 7
1 2 3 4	2	1 3

Таблиця 2

Етапи сжатия числа 58 — четного

Отрезок натурального ряда	Количество оставшихся чисел	Результат сжатия
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58	29	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29	15	1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	8	1 3 5 7 9 11 13 15
1 2 3 4 5 6 7 8	4	1 3 5 7
1 2 3 4	2	1 3

Для получения такой информации при осуществлении каждого этапа сжатия формируется двоичная сигнальная последовательность, фиксирующая четность (нечетность) найденного на этом этапе результата при помощи нуля (единицы).

Напомним, что блочный код при длине блока, равной  $N$  символов, может иметь  $2^N$  кодовых слов, каждому из которых можно поставить в соответствие натуральное число. Значения этих чисел представляют собой последовательность от единицы до  $2^N$  [3], элементы которой по представленной методике могут быть сжаты до значения 1 или 2, причем для восстановления исходного натурального числа они используются как начальные. Далее восстановление осуществляется при помощи сигнальной последовательности, поскольку каждому этапу сжатия соответствует один сигнальный бит. При большой длине сигнальной последовательности ее можно рассматривать как натуральное число и по приведенному алгоритму подвергнуть сжатию.

**Эффективностью**, или **коэффициентом, сжатия** назовем отношение длины исходного отрезка натурального ряда (количества элементов соответствующей последовательности натуральных чисел) к длине сжатой последовательности.

Если на каждом этапе сжатия из отрезка натурального ряда выкалывать лишь четные числа, данный отрезок будет сжиматься ровно в два раза. Длина сигнальной последовательности в этом случае будет слишком большой и равной по длине сжимаемого блока. Поэтому ее сжатие с построением новых сигнальных последовательностей требуемого эффекта не даст.

Ситуация значительно улучшится, если из данного отрезка натурального ряда выкалывать наряду с четными числами и числа, кратные трем, как это сделано в табл. 3. В результате эффективность (коэффициент) сжатия увеличивается до трех. Если, кроме того, из натурального ряда выкалывать числа кратные 5, как это сделано в первой справа ячейке табл. 3, коэффициент сжатия приближается к 4.

Эффективность сжатия будет еще выше, если из натурального ряда кроме чисел, кратных 2, 3 и 5, выкалывать каждый четвертый элемент. Коэффициент сжатия достигнет значения 4,2, но при этом так увеличится неопределенность в отыскании последнего элемента натурального ряда, что сигнальная последовательность теряет всякий смысл. Поэтому целесообразно ограничиться выкалыванием из натурального ряда чисел, кратных двум и трем.

Таблиця 3

Варианты этапов сжатия числа 130

Отрезок натурального ряда	Количество оставшихся чисел	Результат сжатия
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 112 126 127 128 129 130	44	<i>Выкалывание кратности 3</i> 1 5 7 11 13 17 19 23 25 29 31 35 37 41 43 47 49 53 55 59 61 65 67 71 73 77 79 83 85 89 91 95 97 101 103 107 109 113 115 119 121 125 127 128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	34	<i>Выкалывание кратности 5</i> 1 7 11 13 17 19 23 29 31 37 41 43 47 49 53 59 61 67 71 73 77 79 89 91 97 101 103 107 109 113 119 121 127 128
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	28	<i>Выкалывание кратности 4</i> 1 7 11 17 19 23 29 37 41 43 47 53 59 61 67 73 77 79 89 97 101 103 107 113 119 121 127 128
1 2 3 4 7 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28	10	1 7 11 13 17 18 19 21 23 27
1 2 3 4 5 6 7 8 9 10	4	1 3 5 7 9
1 2 3 4	1	1

Как видим, вместо кодового слова, которое заменяется натуральным числом, имеется двоичная сигнальная последовательность, являющаяся, в сущности, его эквивалентом, где каждый бит характеризует один этап сжатия [2; 5].

Получателю информации вместо кодового слова посылается его сжатый эквивалент. Если же натуральный ряд, соответствующий сжато эквиваленту, описывается комбинацией первых двух или трех символов сигнальной последовательности, то получателю передается только она. Поскольку этапов сжатия даже в большом натуральном числе не может быть много, то объем информации в сигнальной последовательности редко превысит 5-6 бит. Например, для сжатия числа 20 000 выкалыванием только четных членов натурального ряда потребуется всего 16 этапов. Если же число этапов слишком велико, сигнальную последовательность можно сжать по описанной методике. В частности, процесс сжатия чисел 53 и 27, которыми могут быть представлены сигнальные последовательности, иллюстрируют табл. 4 и 5. Если три первых бита сигнальной последовательности выделить на представление эквивалентов сжатых целых чисел, равных 7 и 8, то в обоих случаях длина сигнальной последовательности составляет 5 бит.

Этапы сжатия числа 53

Таблица 4

Отрезок натурального ряда	Количество оставшихся чисел	Результат сжатия
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53	19	1 5 7 11 13 17 19 23 25 29 31 35 37 41 43 47 49 51 53
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19	7	1 5 7 11 13 17 19

Этапы сжатия числа 27

Таблица 5

Отрезок натурального ряда	Количество оставшихся чисел	Результат сжатия
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27	10	1 5 7 11 13 17 19 23 25 27
1 2 3 4 5 6 7 8 9 10	5	1 5 7 9 10

**Выводы**

1. При блочном кодировании любое двоичное кодовое слово можно представить натуральным числом в десятичной системе счисления, значение которого не превышает  $2^N$ , где  $N$  — размер блока.
2. Каждое натуральное число, соответствующее кодовому слову, можно сжать до размера нескольких двоичных символов, а затем восстановить в первоначальном виде и за счет их трех-четырёхкратного повторения исправить имеющиеся в кодовом слове ошибки.
3. Восстановление исходного натурального числа осуществляется на основании сигнальной двоичной последовательности, адекватно отражающей этапы его сжатия, причем в порядке, обратном порядку сжатия.
4. Если сигнальная последовательность слишком велика, то ее можно сжать до нескольких символов, воспользовавшись предложенным алгоритмом.
5. Алгоритмы сжатия и восстановления исходного числа являются симметричными и взаимно обратными.

**Список использованной литературы**

1. Давенпорт, Г. Высшая арифметика. Введение в теорию чисел / Г. Давенпорт.— М.: Наука, 1965.— 16 с.
2. Василенко, О. Н. Теоретико-числовые алгоритмы в криптографии / О. Н. Василенко.— М.: МЦНМО, 2003.— 328 с.
3. Берлекэмп, Э. Алгебраическая теория кодирования; пер. с англ. / Э. Берлекэмп.— М.: Мир, 1971.— 477 с.
4. Маховенко, Е. Б. Теоретико-числовые методы в криптографии: учеб. пособие / Е. Б. Маховенко.— М.: Гелиос АРВ, 2006.— 320 с.
5. Коблиц, Н. Курс теории чисел и криптографии / Н. Коблиц.— М.: Гели, 2001.— 254 с.

**Рецензент:** доктор техн. наук, профессор **Б. Ю. Жураковский**, Государственный университет телекоммуникаций, Киев.

*О. В. Дікарев***СТИСНЕННЯ ДВІЙКОВИХ БЛОКОВИХ КОДІВ**

Запропоновано алгоритм усунення надлишковості закодованої двійкової інформації при її передаванні, що забезпечує надалі повне її відновлення завдяки виконанню всіх кроків даного алгоритму в зворотному порядку.

**Ключові слова:** двійковий блоковий код; надлишковість; натуральний ряд чисел; стиснення; сигнальна послідовність.

*A. V. Dikarev***BINARY BLOCK CODES COMPRESSING**

This paper presents the algorithm removing excessiveness of binary block encoding transmission information that guarantees restoring it in accordance with opposite direction.

**Keywords:** binary block code; compressing; natural number series; signal series.

УДК 656.8.001

**Л. О. ЯЩУК**, доктор техн. наук, професор, заслужений діяч науки і техніки України, Одеська національна академія зв'язку ім. О. С. Попова

## **Автоматизовані сортувальні центри: ПОЗИТИВ І НЕГАТИВ**

**Розглянуто проблеми створення автоматизованих сортувальних центрів (АСЦ) в Україні та проаналізовано позитивні і негативні явища, якими супроводжується процес створення таких центрів.**

**Ключові слова:** АСЦ; письмова кореспонденція (ПК); концентрація потоків ПК; нормативні строки (НС) пересилання ПК; часові діаграми пересилання ПК.

### **Вступ**

Комплексною програмою створення єдиної національної системи зв'язку (КПЄНСЗ) України, затвердженою Постановою Кабінету Міністрів України від 23 вересня 1993 р. №790, передбачено будівництво чотирьох АСЦ — Київ, Львів, Харків та Одеса.

Оскільки за понад 23 роки, що минули з часу ухвалення КПЄНСЗ, будь-яких помітних зрушень у створенні АСЦ не відбувалось, численні керівники Українського державного підприємства поштового зв'язку (УДППЗ) «Укрпошта», що змінювали один одного, вважали за необхідне «оптимізувати» кількість та місця розташування АСЦ, унаслідок чого було передбачено, а невдовзі відмінено створення АСЦ у Сімферополі, АСЦ з Харкова перемістився у Дніпро, а з Одеси — у Миколаїв, де з втратою Україною Криму і Севастополя існування такого центру не має жодного сенсу.

Для підтвердження доцільності створення чотирьох АСЦ УДППЗ «Укрпошта» навіть залучила відому німецьку компанію Siemens Dematic, яка спеціалізується на виробництві та продажу автоматичних листосортувальних машин (АЛСМ) і тому, природно, не тільки підтвердила доцільність створення чотирьох АСЦ в Україні, а й, як варіант, обґрунтувала доцільність організації в нашій країні восьми таких центрів.

Слід ураховувати, що в КПЄНСЗ відсутнє техніко-економічне обґрунтування доцільності створення саме чотирьох АСЦ, а підтримка всіма керівниками УДППЗ «Укрпошта» цього положення КПЄНСЗ може мати корупційну складову, у жодний спосіб не пов'язану з реальними потребами підприємства.

### **Основна частина**

Проаналізувавши ситуацію, що склалася, спробуємо розібратися в ній і навести техніко-економічне обґрунтування доцільності створення АСЦ в Україні.

Нагадаємо, що для України, як і для інших республік колишнього СРСР, нинішня хвиля автоматизації сортування ПК — друга. Перша хвиля припала на 1970-80 роки, коли в СРСР було здобуто унікальний досвід автоматизованого сортування ПК у сортувальних об'єктах поштового зв'язку (СОПЗ) республіканських, крайових та обласних центрів.

Причиною практично повсюдної відмови від автоматизованого сортування ПК та повернення до її ручного сортування стало багаторазове падіння потоків ПК, зумовлене розпадом СРСР і бурхливим розвитком електронних засобів телекомунікацій, які повністю або частково витиснули поштовий зв'язок з багатьох раніше опанованих ним ніш.

© Л. О. Ящук, 2017