

УДК 004.42 + 004.942

**А.Я. Анопrienко** (канд. техн. наук, проф.),  
**С.В. Иваница** (аспирант), **Е.В. Бурлака** (магистрант)

Донецкий национальный технический университет  
[anoprien@cs.dgtu.donetsk.ua](mailto:anoprien@cs.dgtu.donetsk.ua), [isv@cs.dgtu.donetsk.ua](mailto:isv@cs.dgtu.donetsk.ua)

## **ПРОГРАММНАЯ РЕАЛИЗАЦИЯ ПОСТБИНАРНОГО КОДИРОВАНИЯ ИНТЕРВАЛОВ**

Представлена программная реализация извлечения из тетракода как полного набора двоичных и десятичных значений (преобразование тетракод → двоичное/десятичное число), так и значений границ интервала с оценкой их предельных колебаний (преобразование тетракод → интервал).

**Ключевые слова:** интервал, программа, тетралогика, тетракод.

### ***Введение***

В настоящее время наблюдается стремительное увеличение объемов и сложности компьютерных вычислений, приводящее в свою очередь к возрастанию потребности контроля достоверности и точности вычислительных процессов. Возникающие при этом различного рода ошибки (например, связанные с неконтролируемым округлением чисел при использовании форматов с плавающей запятой) часто незаметны, однако они существенно искажают результат и, в конечном счете, оказываются причинами ряда вычислительных катастроф [1]. Анализ подобных ситуаций позволяет предположить, что возможности классической двоичной логики на сегодняшний день исчерпаны, и при дальнейшем возрастании сложности вычислений использование двоичных компьютерных систем не гарантирует получения достоверных результатов. Этот факт может послужить началом пересмотра логики современных ЭВМ в рамках перехода к расширенному кодо-логическому базису [2,3].

Рассмотрение расширенного двумерного логического пространства в [4] привело к появлению новых логических состояний, что позволило значительно расширить возможности формализованной логической оценки различных нюансов реальных процессов и ситуаций. На основе такого набора логических значений были введены в рассмотрение логические системы  $L_N$  различных  $N$  порядков (гиперлогики), среди которых особый интерес вызывает логика четвертого порядка — **тетралогика**  $L_4 = \{0, 1, A, M\}$  — представленная одним из многих вариантов набора четырех (из 17 возможных) логических значений. В работе [5] введено

понятие **тетракода**  $S_4 = \{0, 1, A, M\}$ , как системы кодирования количественной информации, построенной на основе тетралогии  $L_4$ . Каждое значение тетракода (тетрит [6]) кодирует соответствующее состояние тетралогии: 0 — ложь (логический 0); 1 — истина (логическая 1); A — неопределенность (ненаступление событий «истина» и «ложь»); M — множественность (обоюдное наступление событий «истина» и «ложь»). Рассматривая тетракод в качестве носителя количественной информации, можно высказать следующие предположения:

- тетриты 0 и 1 сводятся к битам 0 и 1 соответственно, так как они кодируют аналогичные логические состояния;
- тетрит M сводится к битам 0 и 1 одновременно, поэтому представим двумя точками на числовой оси;
- тетрит A сводится к битам 0 или 1 (в момент сведения его значение неизвестно), поэтому представим одной из двух возможных точек на числовой оси.

На основании данных предположений в [7] рассмотрены особенности представления (позиционирования на числовой оси) границ вещественных интервалов [8] одним значением тетракода. Приведены особенности представления диапазонов «плавающих» границ (т. е. ширины их предельных колебаний), предполагающих наличие совокупностей тетритов M и A в младших разрядах тетракода.

В данной работе рассматривается вариант применения тетракодов для получения множества значений на вещественной числовой оси, и, как следствие, сведение тетракода к крайним значениям этого множества для представления интервальных чисел.

**Целью данной работы** является программная реализация процесса сведения тетракода к набору числовых значений, выраженных как множеством всех представимых данным тетракодом чисел, так и парой чисел, являющихся «скользящими» границами интервала. Для представления результатов такого преобразования были разработаны две аналогичные программные реализации «Перевод тетракода T [0:31] {0,1,M,A} в 2/10 форматы чисел» в средах Adobe Flash CS4 (язык ActionScript, версия 3.0) и MS Visual Studio 2010 (язык C#, версия 4.0).

### **Особенности представления интервальных чисел с помощью тетракода**

Для получения численных значений при декодировании тетракода необходимо придерживаться следующих положений:

- поскольку один тетрит сводится к одному биту, то  $n$ -разрядный тетракод декодируется в набор  $n$ -разрядных двоичных чисел;

- количество двоичных чисел в сведенном из тетракода наборе равно  $2^m$ , где  $m$  — количество тетритов  $M$  в тетракоде;
- наличие тетритов  $M$  в тетракоде позволяет рассматривать его как тетракод, сводящийся к значениям границ интервального числа.

Последнее положение при декодировании тетракода требует более детального рассмотрения. При сведении значений тетритов  $t_i$  ( $i$  — позиция тетрита в  $n$ -разрядном тетракоде,  $n-1 \leq i \leq 0$ ) тетракода к интервалу можно прибегнуть к предложенным в [6] **унарным функциям тетралогии** (табл. 1):  $\text{MIN\_A}$  ( $\text{MAX\_A}$ ) — минимизация (максимизация) неопределенности;  $\text{MIN\_M}$  ( $\text{MAX\_M}$ ) — минимизация (максимизация) множественности.

Таблица 1. Функции минимизации и максимизации для тетритов  $A$  и  $M$

Унарная функция	Значение тетрита $t$			
	0	1	A	M
$\text{MIN\_M}(t)$	0	1	A	0
$\text{MAX\_M}(t)$	0	1	A	1
$\text{MIN\_A}(t)$	0	1	0	M
$\text{MAX\_A}(t)$	0	1	1	M

Используя сочетания приведенных в таблице функций, можно все значения тетритов  $A$  и  $M$  привести к значениям 0 и 1. Тетракод, состоящий из тетритов 0 и 1, фактически является двоичным числом — значением интервальной границы. При сведении  $n$ -разрядного тетракода к значениям интервальных границ предлагается использовать следующие четыре сочетания:

$$\text{MIN\_M}(\text{MIN\_A}(t_i)) = \begin{cases} 0, & \text{if } t_i = 0 \vee t_i = A \vee t_i = M, \\ 1, & \text{if } t_i = 1; \end{cases} \quad (1)$$

$$\text{MIN\_M}(\text{MAX\_A}(t_i)) = \begin{cases} 0, & \text{if } t_i = 0 \vee t_i = M, \\ 1, & \text{if } t_i = 1 \vee t_i = A; \end{cases} \quad (2)$$

$$\text{MAX\_M}(\text{MIN\_A}(t_i)) = \begin{cases} 0, & \text{if } t_i = 0 \vee t_i = A, \\ 1, & \text{if } t_i = 1 \vee t_i = M; \end{cases} \quad (3)$$

$$\text{MAX\_M}(\text{MAX\_A}(t_i)) = \begin{cases} 0, & \text{if } t_i = 0, \\ 1, & \text{if } t_i = 1 \vee t_i = A \vee t_i = M. \end{cases} \quad (4)$$

Таким образом, в качестве оценки предельных колебаний интервальных границ рассматриваются случаи получения их значений для интервалов, имеющих максимальную и минимальную ширину (под шириной интервала понимается диапазон принадлежащих данному интервалу значений, т. е. расстояние между его границами). Такая оценка предельных колебаний позволяет зафиксировать области, в которые гарантированно попадут границы извлеченного из тетракода интервала,

поскольку при декодировании тетрита в интервал поле значений А сводится к случайному набору двоичных чисел. При этом корректное сведение тетракода к интервалу возможно лишь в том случае, если в исходном тетракоде тетриты А и М представлены группами, причем группа тетритов А должна занимать младшие разряды тетракода.

### **Структура программы декодирования тетракода**

На рис. 1, 2 представлены два варианта программы, которая позволяет декодировать 32-разрядный тетракод с учетом описанных выше преобразований.

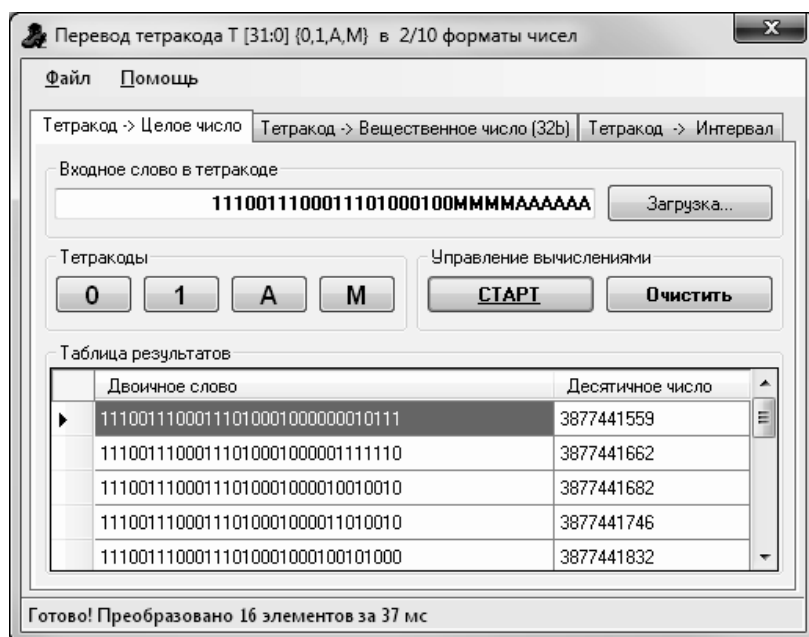


Рисунок 1 - Вариант программы, реализованной на языке С# (результат работы режима «Тетракод → Целое число»)

Программа позволяет выполнить преобразование тетракодов в двоичные и десятичные наборы чисел. Исходный тетракод может быть «целочисленным» (сводиться к двоичным 32-разрядным целым беззнаковым числам) либо «вещественным» (сводиться к двоичным 32-разрядным числам с плавающей запятой).

Программа имеет **4 режима работы**, в которых выполняются следующие преобразования:

- 1) *Тетракод → Целое число*: преобразование «целочисленного» тетракода в набор целых 32-разрядных чисел (рис. 1).
- 2) *Тетракод → Целочисленный интервал*: сведение тетракода к паре 32-разрядных целых чисел, являющихся границами интервала.
- 3) *Тетракод → Вещественное число*: преобразование тетракода в набор вещественных 32-разрядных чисел.

- 4) *Тетракод* → *Вещественный интервал*: сведение тетракода к паре 32-разрядных вещественных чисел, являющихся границами интервала (рис. 2).



Рисунок 2 - Вариант программы, реализованной на языке ActionScript (результат работы режима «Тетракод → Вещественный интервал»)

Программа имеет «интеллектуальный» механизм ввода значений, обеспечивающий «комфортный» ввод данных (ввод данных с клавиатуры и с экрана, возврат на предыдущий шаг ввода, очистка поля, и др.) и не позволяющей работать с некорректно введенной информацией (блокировка работы при событиях «введен не тетракод» для всех режимов и «некорректное размещение тетритов А и М в тетракоде» для режимов 2 и 4). В нижней части окна программы отображается сопроводительная информация, которая включает следующие параметры: время выполнения преобразования тетракода и количество полученных при этом элементов (режимы 1 и 3); середина, ширина и «собранные» границы интервала (режимы 2 и 4). Результирующий набор полученных значений представлен в двоичных и десятичных числах.

В программе предусмотрен ввод предустановленных значений тетракода (рис. 2, поле «Выбор примеров»), а также загрузка из файла заранее подготовленных значений (рис. 1, кнопка «Загрузка...»). Имеется мультиязычная поддержка (русский, украинский и английский языки) интерфейса и справочных материалов.

Программа занимает около 250 КБ (flash-приложение) и 60 КБ (C#-приложение) дискового пространства. Вариант программы на flash

отличается ярким динамическим интерфейсом, и изначально планировался для использования в виде веб-приложения.

### Примеры декодирования тетракода

Рассмотрим декодирование тетракода на примере работы программы в 3-м и 4-м режимах. Значения тетритов исходного тетракода  $T$  представлены на рис. 3.

Входное вещественное 32-разрядное число в тетракоде:

Знак	Порядок	Мантисса
0	1 1 1 1 0 0 1 1	0 0 0 0 0 0 0 0 1 1 1 0 0 M M M M M M A A A

Знак    Порядок    Мантисса

0 1 M A ← Backspace

Рисунок 3 - Поле ввода вещественного тетракода flash-приложения (элементы checkbox определяют доступность полей к вводу значений)

Результаты работы программы в режиме 3 (сведение тетракода к набору вещественных чисел) представлены на рис. 4.

Двоичное слово	Вещественное число
0 11110011 000000001110000000000001	8,336071E+34
0 11110011 00000000111000000001001	8,336079E+34
0 11110011 00000000111000000010010	8,336088E+34
0 11110011 00000000111000000011111	8,336101E+34
0 11110011 00000000111000000100101	8,336107E+34

Готово! Преобразовано 128 элементов за 16 мс

Рисунок 4 - Результат сведения тетракода к набору вещественных чисел

За 16 мс программа выполнила сведение тетракода  $T$  к набору из  $2^8 = 128$  вещественных чисел (в тетракоде 8 тетритов  $M$ ). При этом тетриты  $M$  в своей совокупности обеспечили полный перебор двоичных значений, а тетриты  $A$  в каждом числе из набора случайным образом свелись к нулевым или единичным битам. Повторные запуски программы приведут к получению различных по содержанию наборов чисел, поскольку тетриты  $A$  каждый раз будут заполняться случайными двоичными значениями.

В 4-м режиме работы программы (сведение тетракода к вещественному интервалу) возможны модификации получения результата (см. рис. 2): получение диапазонов колебаний границ («минимальная

ширина интервала», «максимальная ширина интервала») и получение интервала с учетом случайного замещения двоичными значениями тетритов А. При отсутствии тетритов А в тетракоде, программа будет возвращать одинаковые интервалы не зависимо от выбранной модификации режима.

На рис. 5 представлены результаты работы программы при сведении тетракода  $T$  к интервальным границам в каждой модификации.

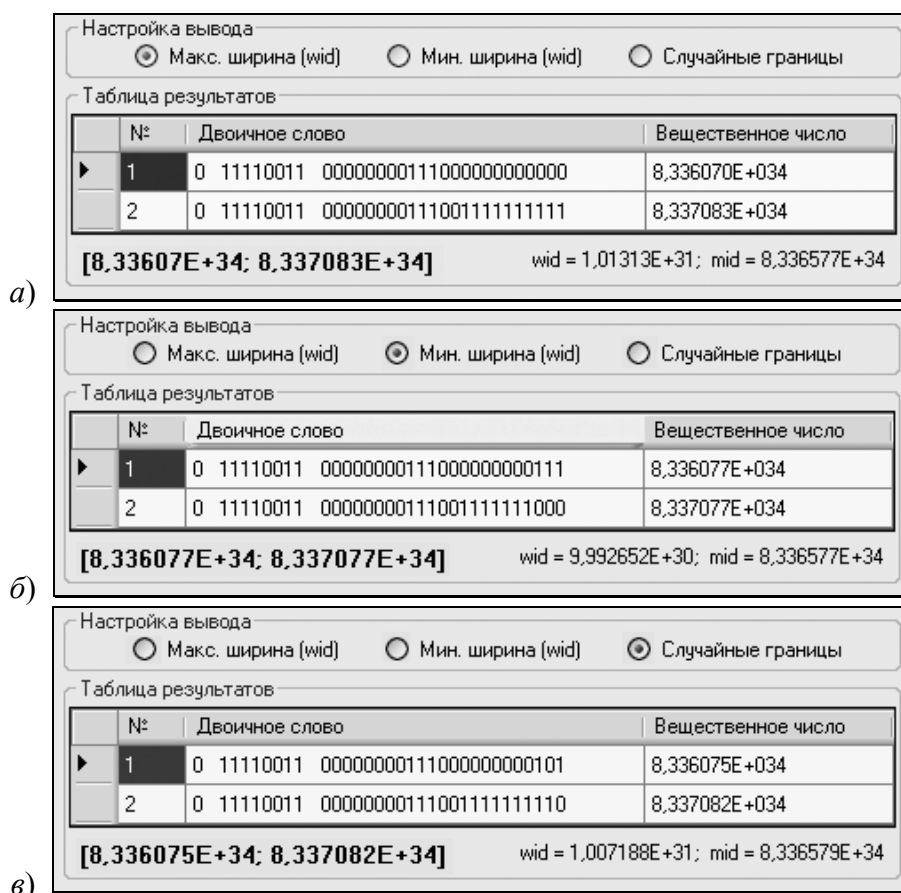


Рисунок 5 - Результат сведения тетракода к вещественному интервалу (*а* — получение «крайне широких» границ; *б* — получение «крайне узких» границ, *в* — получение «случайных» границ)

Модификацию «максимальная ширина» (рис. 5, *а*) можно трактовать следующим образом: определение минимального и максимального значения в декодированном из тетракода наборе чисел. Действительно, каков бы не был набор чисел (с учетом разрядов А), ни одно число не будет меньше левой и больше правой границ интервала «максимальной ширины», численные значения которых можно получить из тетракода  $T$ , используя соотношения (1) и (4) для левой и правой границы соответственно, т. е.

$$[\text{MIN\_M}(\text{MIN\_A}(T)); \text{MAX\_M}(\text{MAX\_A}(T))] \text{ — искомый интервал.}$$

Применение данной модификации представляет собой интервальную оценку «сверху», т. е. позволяет оценить максимальный «захват» участка числовой оси сведенным из тетракода интервалом.

Модификация «минимальная ширина» (рис. 5, б) отображает «худшие» случаи относительно тетритов  $A$  при определении границ интервала. Границы интервала «минимальной ширины» из тетракода  $T$  можно получить, используя соотношения (2) и (3) для левой и правой границы соответственно, т. е.

$[\text{MIN\_M}(\text{MAX\_A}(T)); \text{MAX\_M}(\text{MIN\_A}(T))]$  — искомый интервал.

Применение данной модификации представляет собой интервальную оценку «снизу», т. е. позволяет оценить минимальный «захват» участка числовой оси сведенным из тетракода интервалом.

Действительно, границы интервала (с учетом разрядов  $A$ ), полученные при модификации «случайные границы» (рис. 5, в), никогда не будут меньше по модулю соответствующих границ интервала «минимальной ширины». Таким образом, границы сведенного из тетракода  $T$  интервала при любом случайном наборе бит для разрядов  $A$  будут принадлежать следующим диапазонам:

$[\text{MIN\_M}(\text{MIN\_A}(T)); \text{MIN\_M}(\text{MAX\_A}(T))]$  — для левой границы;

$[\text{MAX\_M}(\text{MIN\_A}(T)); \text{MAX\_M}(\text{MAX\_A}(T))]$  — для правой границы.

Ширина этих диапазонов определяет пределы колебаний границ интервала, полученного при декодировании тетракода  $T$ . Такой интервал может быть записан в виде

$$\left[ \begin{array}{l} [\text{MIN\_M}(\text{MIN\_A}(T)); \text{MIN\_M}(\text{MAX\_A}(T))]; \\ [\text{MAX\_M}(\text{MIN\_A}(T)); \text{MAX\_M}(\text{MAX\_A}(T))] \end{array} \right] \Rightarrow$$

$$\Rightarrow [\text{MIN\_M}(\text{Rnd\_A}(T)); \text{MAX\_M}(\text{Rnd\_A}(T))],$$

где  $\text{Rnd\_A}$  — функция, сводящая тетрит  $A$  к двоичным 0 или 1, которые выбираются случайным образом.

## **Заключение**

В данной работе рассмотрен способ постбинарного кодирования интервалов путем сведения тетракода к значениям интервальных границ. Предложены способы интервального оценивания (оценки предельных колебаний) границ интервала с использованием функций алгебры тетралогии. Представлена программная реализация рассматриваемой задачи — программа декодирования тетракода, которая позволяет выполнять:

- преобразование тетракода в набор целочисленных значений;
- преобразование тетракода в набор вещественных чисел;



– декодирование тетракода в границы интервалов с проведением интервального оценивания «сверху» и «снизу».

Разработанный программный продукт является в своем роде уникальным и может служить в учебных целях для наглядной иллюстрации представления тетракодов в виде числовых (точечных) и интервальных (множественных) значений. В дальнейшем планируется размещение данной программы (варианта, реализованного с помощью flash-технологии) на портале моделирования ДонНТУ [9].

### **Список литературы**

1. Аноприенко А.Я. Интервальные вычисления и перспективы их развития в контексте кодо-логической эволюции / А.Я. Аноприенко, С.В. Иваница // Научные труды Донецкого национального технического университета. Серия «Проблемы моделирования и автоматизации проектирования динамических систем» (МАП-2010). – 2010. – Вып. 8 (168). – С. 150–160.
2. Аноприенко А.Я. Расширенный кодо-логический базис компьютерного моделирования. Кафедра ЭВМ ДонГТУ. [Электронный ресурс] / А.Я. Аноприенко. – Страница доступа: <http://cs.dgtu.donetsk.ua/~anoprien/Publ/1997/base97ex.html>.
3. Аноприенко А.Я. Обобщенный кодо-логический базис в вычислительном моделировании и представлении знаний: эволюция идеи и перспективы развития / А.Я. Аноприенко // Научные труды ДонНТУ. Серия «Информатика, кибернетика и вычислительная техника» (ИКВТ – 2005). – 2005. – Вып. 93. – С. 289-316.
4. Аноприенко А.Я. О некоторых возможностях расширения логического базиса информатики / А.Я. Аноприенко, А.А. Кухтин // В кн. "Тези доповідей міжнародної науково-практичної конференції «Інформатизація в умовах переходу до ринку»", Київ, 5-6 листопада 1992 р. – К., 1992. – С. 30–32.
5. Аноприенко А.Я. Тетралогика и тетракоды / А.Я. Аноприенко // В кн. «Сборник трудов факультета вычислительной техники и информатики». – 1996. – Вып. 1. – С.32–43.
6. Иваница С.В. Особенности реализации операций тетралогика / С.В. Иваница, А.Я. Аноприенко // Научные труды Донецкого национального технического университета. Серия: «Информатика, кибернетика и вычислительная техника» (ИКВТ-2011). — 2011. – Вып. 13 (185). – С. 134–140.
7. Аноприенко А.Я. Особенности постбинарного кодирования на примере интервального представления результатов вычислений по формуле Бэйли-Боруэйна-Плаффа / А.Я. Аноприенко, С.В. Иваница // Научные труды Донецкого национального технического университета. «Информатика, кибернетика и вычислительная техника». — 2010. — Вып.11(164). — С. 19–23.
8. Hansen E. Global optimization using interval analysis / E. Hansen, G.W. Walster. — New York: Marcel Dekker, 2004.
9. Портал моделирования ДонНТУ. Электронный ресурс. Страница доступа: <http://sim.donntu.edu.ua>.

*Надійшла до редакції 23.09.2011.*

*Рецензент: канд. техн. наук, доц. Зеленева И.Я.*

**О.Я. Анопрієнко, С.В. Іваниця, Є.В. Бурлака**  
Донецький національний технічний університет

**Програмна реалізація постбінарного кодування інтервалів.** Представлена програмна реалізація вилучення з тетракоду як повного набору двійкових і десяткових значень (перетворення тетракод  $\rightarrow$  двійкове/десяткове число), так і значень границь інтервалу з оцінкою їх граничних коливань (перетворення тетракод  $\rightarrow$  інтервал).

**Ключові слова:** інтервал, програма, тетралогіка, тетракод.

**A. Anoprienko, S. Ivanitsa, E. Burlaka**  
Donetsk National Technical University

**Software Implementation of Coding Postbinary Intervals.** The paper presents a software implementation, which extracts from tetracode a complete set of binary and decimal values (convert tetracode  $\rightarrow$  binary/decimal) and values of the limits of the interval with an estimate of their marginal fluctuations (convert tetracode  $\rightarrow$  interval).

**Keywords:** interval, program, tetralogic, tetracode.