

## Раздел VI. Вычислительные комплексы нового поколения и нейрокомпьютеры

УДК 681.3:007

Я.Е. Ромм, Д.А. Чабанюк

### СРАВНЕНИЕ СЛОВ С ЕДИНИЧНОЙ ВРЕМЕННОЙ СЛОЖНОСТЬЮ

*Излагается способ сравнения слов числового и строкового типа с применением вертикальной обработки. Способ конструируется на основе разрядного распараллеливания алгебраического сложения двоичных чисел и применяется к алгоритмам сортировки, а также слияния строковых элементов в качестве этапа информационного поиска. Представлен исходный метод поразрядно-параллельного сложения без вычисления переноса, его знакоразрядная модификация для поразрядно-параллельного сравнения чисел и строк. Для выполнения алгебраического сложения без вычисления переноса используется обратный двоичный код отрицательных чисел и дана модификация дополнительного кода, исключающая вычисление переноса. При сравнении строковых элементов все символы слова кодируются двоичными числами в ASCII-коде, набор двоичных кодов символов слова интерпретируется как одно двоичное число в позиционной системе счисления. При сравнении путем алгебраического сложения таких чисел они выравниваются по старшему разряду. Включения способа сравнения в параллельные алгоритмы сортировки и слияния увеличивает глубину распараллеливания до параллелизма на уровне операций сравнения слов и одновременно на уровне разрядных операций. Предложен способ выделения знака старшего ненулевого разряда слова, полученного в результате сравнения. Способ включает алгоритмические и схемные решения, обеспечивающие максимальное разрядное распараллеливание. Дополнительно рассматривается проблема вхождения подстроки в строку. Синтезирован алгоритм максимально параллельного поиска вхождения подстроки в строку. Даны оценки временной сложности, в частности максимальный параллелизм поиска вхождения подстроки в строку характеризуется оценкой времени  $O(1)$ , независимо от числа символов в строке. Приводятся примеры, алгоритмы и схемы выполнения предложенного метода.*

*Поразрядно-параллельное сложение без вычисления переноса; знакоразрядный код; поразрядно-параллельное сравнение чисел и строк; параллельная сортировка и поиск; временная сложность.*

Ya.E. Romm, D.A. Chabanyuk

### COMPARISON OF WORDS WITH THE COMPLEXITY OF IDENTITY

*Was described method of comparing words numeric and string type using the vertical processing. The method is constructed on the basis of the discharge parallelization of algebraic addition of binary numbers and applied to sorting algorithms, as well as the merger of strings as an information search step. Presented by the original method was bit by bit parallel addition without calculating transfer in bit code of the sign modification for by bit parallel comparison of numbers and strings. To perform algebraic addition without calculating a reverse transfer binary negative numbers and given additional code modification, eliminating the calculation of transfer. When comparing string elements all word characters encoded binary numbers in ASCII-code set of binary characters per word is interpreted as a binary number in a positional number system. When comparing by algebraic addition of the numbers they are aligned to the senior class. Inclusion in the method of comparing parallel algorithms for sorting and merging increases the maximum*

depth parallel to the ideal level parallelism comparisons of words and at the same time based on the level of operations. The method of allocation was the senior non-zero sign bit in the word, the resulting comparison. The method includes algorithmic and schematics, providing maximum bit parallelization. Additionally there was the problem of occurrence of a substring in a string. Was synthesized parallel search algorithm for the maximum occurrence of a substring in a string. Estimates are given for the time complexity, in particular the maximum parallelism occurrence of a substring in search string characterized by the time complexity  $O(1)$ , regardless of the number of characters per string. Examples are algorithms and schemes implementation of the proposed method.

Bit by bit-parallel addition without calculating transfer; bit code of the sign; Bit by bit-parallel comparison of numbers and strings; parallel sorting and searching; time complexity.

**Постановка вопроса.** Ускоряющаяся компьютеризация всех областей деятельности и информационных процессов, значительный рост объемов обрабатываемой информации приводит к трудностям при последовательном выполнении информационного поиска на основе классических алгоритмов [1, 2]. Как основу ускорения поиска можно рассматривать разрядное распараллеливание сравнения слов. С этой целью ниже излагается метод вертикального полноразрядного бинарного сложения без вычисления переноса, заимствованный из [3, 4], и его модификация на случай поразрядно-параллельного сравнения чисел и элементов строкового типа. Модифицированный метод применяется к параллельным алгоритмам сортировки в качестве этапа информационного поиска. Целью модификации является максимально параллельное выполнение сравнения, упорядочения и поиска на уровне одновременно алгоритмических и разрядных операций. Новизна предложения сводится к максимально параллельной схеме выделения знака старшего ненулевого разряда в знакоразрядном однорядном коде результата сравнения.

**Поразрядно-параллельное алгебраическое сложение в знакоразрядном коде без вычисления переноса.** Описание метода поразрядно-параллельного алгебраического сложения заимствовано из [3, 4]. Пусть рассматриваются два двоичных полинома, –

$$P_1 = \sum_{i=0}^n \beta_i 2^i, \beta_i = \begin{cases} 0 \\ 1 \end{cases}, P_2 = \sum_{i=0}^n \gamma_i 2^i, \gamma_i = \begin{cases} 0 \\ 1 \end{cases}. \quad (1)$$

Пусть  $P_1$  и  $P_2$  расположены в двух входных разрядных сетках (РС),  $PC_{\text{вх}}^{(0)}$  и  $PC_{\text{вх}}^{(1)}$ , номера клеток которых совпадают с индексами коэффициентов из (1).

$\beta_n$	$\beta_{n-1}$	...	$\beta_1$	$\beta_0$	$PC_{\text{вх}}^{(0)}$
$\gamma_n$	$\gamma_{n-1}$	...	$\gamma_1$	$\gamma_0$	$PC_{\text{вх}}^{(1)}$

(2)

Синхронно и взаимно независимо (СВН-параллельно) по всем номерам разрядов  $j$  из (2) выполняется операция  $\beta_j + \gamma_j$  суммирования по вертикали (СВ<sub>*j*</sub>).

Результат СВ<sub>*j*</sub> представляется в двоичном коде

$$\beta_j + \gamma_j = \sum_{\ell=0}^1 \Delta_{\ell j} 2^\ell, \Delta_{\ell j} = \begin{cases} 0 \\ 1 \end{cases}, j=0,1,\dots,n. \quad (3)$$

При каждом  $j = \text{const}$  коэффициенты из (3) располагаются по диагонали двух выходных РС, –  $PC_{\text{вых}}^{(0)}$  и  $PC_{\text{вых}}^{(1)}$ , образуя диагональную от  $j$ -го разряда запись ( $D_j$ -запись) вида

$$\begin{aligned}
 \text{СВ}_j: & \quad + \begin{array}{c} j+1 \quad j \quad j-1 \\ \left\{ \begin{array}{c|c|c|c} \hline & & \beta_j & \\ \hline & & \gamma_j & \\ \hline & & & \end{array} \right\} \begin{array}{l} \text{PC}_{\text{ВХ}}^{(0)} \\ \text{PC}_{\text{ВХ}}^{(1)} \end{array} \\
 \text{Д}_j \text{ - запись:} & \quad \left\{ \begin{array}{c|c|c|c} \hline & & \Delta_{0j} & \\ \hline & \Delta_{1j} & & \\ \hline & & & \end{array} \right\} \begin{array}{l} \text{PC}_{\text{ВЫХ}}^{(0)} \\ \text{PC}_{\text{ВЫХ}}^{(1)} \end{array} .
 \end{array} \quad (4)$$

За предварительный результат сложения принимается промежуточный набор разрядных значений из  $\text{PC}_{\text{ВЫХ}}^{(0)}, \text{PC}_{\text{ВЫХ}}^{(1)}$ :

$$\begin{array}{c|c|c|c|c|c|c|c|c|c} \Delta_{n+1} & \Delta_n & \Delta_{n-1} & \dots & \Delta_j & \dots & \Delta_1 & \Delta_0 & \text{PC}_{\text{ВЫХ}}^{(0)} & \\ \hline \nabla_{n+1} & \nabla_n & \nabla_{n-1} & \dots & \nabla_j & \dots & \nabla_1 & \nabla_0 & \text{PC}_{\text{ВЫХ}}^{(1)} & \end{array}, \quad (5)$$

где  $\Delta_j = \Delta_{0j}, \nabla_j = \Delta_{1(j-1)}$  из (4),  $j = 0, 1, \dots, n, n+1$ . Ниже пара слагаемых (2) называются входным набором, пара слагаемых (5) – промежуточным набором.

Все переносы в (5) оказываются взаимно отделенными вертикальными парами нулей:

$$\begin{pmatrix} \Delta_k \\ \nabla_k \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad k = \text{const},$$

где  $\Delta_k, \nabla_k$  из (5),  $0 \leq k \leq n+1$ . Это так потому [4], что при любом  $j$  из (4) складываются не более двух единиц одинакового веса. По той же причине в (4) по диагонали  $\text{Д}_j$ -записи от 1 в  $\text{PC}_{\text{ВЫХ}}^{(0)}$  не может располагаться  $\underline{1}$  в  $\text{PC}_{\text{ВЫХ}}^{(1)}$ . На этой основе корректным является следующее преобразование промежуточного набора (5). Все разряды, имеющие значение  $\Delta_j \neq 0$ , из  $\text{PC}_{\text{ВЫХ}}^{(0)}$  (и только из  $\text{PC}_{\text{ВЫХ}}^{(0)}$ ) в (5) подвергаются тождественному преобразованию:

$$\Delta_j \equiv 2\Delta_j - \Delta_j, \quad j = 0, 1, \dots, n. \quad (6)$$

При этом правая часть (6) размещается в  $j$ -й клетке  $\text{PC}_{\text{ВЫХ}}^{(0)}$  и в  $(j+1)$ -й клетке  $\text{PC}_{\text{ВЫХ}}^{(1)}$  в порядке  $\text{Д}_j$ -записи для всех номеров  $j$  преобразуемых разрядов:

$$\Delta_j \rightarrow \begin{array}{c|c|c|c} & & j & \\ & 0 & \Delta_j & \\ \hline & j+1 & & \end{array} \rightarrow 2\Delta_j - \Delta_j \rightarrow \begin{array}{c|c|c|c} & & j & \\ & \Delta_j & -\Delta_j & \\ \hline & j+1 & & \end{array} \begin{array}{l} \text{PC}_{\text{ВЫХ}}^{(0)} \\ \text{PC}_{\text{ВЫХ}}^{(1)} \end{array} \quad (7)$$

Как только что отмечено, в  $(j+1)$ -й клетке  $\text{PC}_{\text{ВЫХ}}^{(1)}$  до применения (6), при условии, что в  $\text{PC}_{\text{ВЫХ}}^{(0)}$   $\Delta_j \neq 0$ , всегда находится ноль, иначе  $\text{Д}_j$ -запись означала бы результат сложения по вертикали трех единиц веса  $j$ -го разряда, тогда как их число не больше двух [3, 4].

Вслед за тем СВН-параллельно по всем  $j = 0, 1, \dots, n, n+1$  выполняется шаг вертикального сложения, результатом которого в каждом разряде всегда будет либо 0, либо  $+1$ , либо  $-1$  [3, 4]. При этом значение знакоразрядного кода располагается в  $\text{PC}_{\text{ВЫХ}}^{(0)}$ , где, таким образом, сформируется окончательное значение вычисляемой суммы  $P_1$  и  $P_2$  из (1).

Временная сложность  $T$  СВН-параллельного сложения в знакоразрядном коде составит  $\approx 2\tau$  ( $\tau$  – время переключения логического элемента) при количестве элементов  $S^\ell$  сумматора пропорциональном числу разрядов слагаемых:

$$T \approx 2\tau = O(1), \quad (8)$$

$$S\ell \approx 4(n+1) = O(n). \quad (9)$$

Оценки (8), (9) инвариантны относительно числа разрядов  $n + 1$ .

**Сравнение двоичных чисел на основе поразрядно-параллельного алгебраического сложения.** Предполагается, что сравниваемые числа имеют формат (1) с нумерацией разрядов справа налево. Пусть число  $A$  принимается за уменьшаемое,  $-B$  за вычитаемое,  $B$  записывается в обратном коде (единицы заменяются нулями, а нули – единицами). Это допустимо [5–7] рассматривать как тождественное преобразование:

$$A - B = A + \left( (2^{n+1} - 1) - B \right) - (2^{n+1} - 1), \quad (10)$$

с учетом

$$2^{n+1} - 1 = 2^n + 2^{n-1} + 2^{n-2} + \dots + 2^1 + 2^0, \quad (11)$$

или, в позиционной системе,

$$2^{n+1} - 1 = \underbrace{1\ 1\ 1\ \dots\ 1\ 1}_{n+1}. \quad (12)$$

Над числом  $A$  и числом  $B$ , взятым в обратном коде, СВН-параллельно по всем номерам разрядов выполняется операция  $СВ_j$ , затем  $A$  и  $B$  в обратном коде суммируются по схеме (2)–(7). Для восстановления правильного результата вычитания учитывается (10)–(12): от однорядного знакоразрядного двоичного кода полученного результата следует вычесть  $2^{n+1} - 1$ . Это равносильно тому, что с полученным результатом складывается  $-2^{n+1} + 1$ , – к младшему разряду добавляется  $+1$ , а к разряду веса  $2^{n+1}$  (к  $(n+2)$ -му справа налево) добавляется  $-1$ .

Знак сравнения  $A$  и  $B$  определяется знаком старшего ненулевого разряда однорядного знакоразрядного двоичного числа, полученного в скорректированном окончательном результате. Если знак этого разряда отрицательный, то  $A < B$ , если положительный, то  $A > B$ , если все разряды нулевые, то  $A = B$ .

**Поразрядно-параллельный способ сравнения строк.** Метод переносится [5, 6] на сравнение элементов строкового типа, для краткости, – строк. Используется представление всех символов в ASCII-коде и его перевод в двоичный код. Набор двоичных кодов всех символов в порядке расположения интерпретируется как единое числовое значение. Сравнение полученных двоичных кодов выполняется путем алгебраического сложения изложенным поразрядно-параллельным способом, при этом в соответствии с лексикографическим упорядочением слов выравнивание весов разрядов выполняется по старшим разрядам. Например, [5], сравниваются два слова "ПОИСК" и "ПОЯС": каждый символ слова представляется в ASCII-коде в двоичной форме. Набор двоичных кодов символов в последовательном расположении интерпретируется как единое число:

"ПОИСК": 11001111 11001110 11001000 11001010 11010001  
 "ПОЯС": 11001111 11001110 11011111 11001010 .

Согласно лексикографическому порядку сравнение выполняется с выравниванием старших разрядов, и код слова "ПОЯС" дополняется недостающими в младших разрядах нулями:

"ПОИСК": 11001111 11001110 11001000 11001010 11010001  
 "ПОЯС": 11001111 11001110 11011111 11001010 00000000

Первое слово – уменьшаемое, второе – вычитаемое, двоичный код второго представляется в обратном коде, после чего СВН-параллельно по всем номерам разрядов выполняется операция  $СВ_j$ :

$$\begin{array}{r}
 11001111 \ 11001110 \ 11001000 \ 11010001 \ 11001010 \\
 + \ 00110000 \ 00110001 \ 00100000 \ 00101110 \ 11111111 \\
 \hline
 11111111 \ 11111111 \ 11101000 \ 11111111 \ 00110101 \\
 0 \ 00000000 \ 00000000 \ 00000000 \ 00000001 \ 1001010
 \end{array}$$

Промежуточная сумма преобразуется согласно (6), (7) с учетом (10), (12), в окончательном результате: 000000000000000000–11–1010000000011001010. Старший ненулевой разряд равен –1, следовательно, "ПОИСК" < "ПОЯС". Сравнение выполняется поразрядно-параллельно с оценками (8), (9) без учета времени выделения знака старшего ненулевого разряда.

**Проблема выделения знака старшего ненулевого разряда.** Возникает проблема максимально параллельного выделения знака старшего ненулевого разряда. Следующий схемный способ (рис. 1) реализуется за время  $O(1)$  путем синхронного распространения сигнала запрета из каждого ненулевого разряда во все разряды младшего веса [8]. Во избежание детализации абстрактно предполагается, что каждый разряд обрабатывается взаимно однозначно сопоставленным этому разряду процессорным элементом (ПЭ), его структура инвариантна относительно номера разряда.

На рис. 1 изображены основные операции, СВН-параллельно выполняемые ПЭ, и линии связи (ЛС) между ПЭ. Если значение разряда есть "0", от взаимно однозначно сопоставленного этому разряду ПЭ на вход элемента "НЕ" отдельно сопоставленного вертикального набора ПЭ (на вертикали слева, рис. 1) по диагональной ЛС подается сигнал "0", одновременно этот же сигнал подается на все расположенные выше ПЭ вертикального набора путем размножения по вертикали на все выше расположенные ЛС. Выходные ЛС параллельны входным, и по ним в разрядные ПЭ поступает инвертированный сигнал "1". Разрядные ПЭ принимают инвертированные сигналы на вход собственной логической схемы "И", сохраняя свое входное значение и также подавая его на вход своей схемы "И". В результате все нулевые значения разрядов сохраняются.

Если значение разряда есть "±1", от взаимно однозначно сопоставленного ему ПЭ на вход элемента "НЕ" аналогично сопоставленного ПЭ вертикального набора по связующей их диагональной ЛС также подается сигнал "0", но на все расположенные выше ПЭ вертикального набора путем размножения сигнала на все расположенные выше ЛС в этом случае подается сигнал "1". По выходным ЛС обратно в данный разряд поступает инвертированный сигнал "1", а во все разрядные ПЭ справа по расположенным выше ЛС поступает инвертированный сигнал "0". Разрядные ПЭ принимают инвертированные сигналы на вход собственной логической схемы "И", по-прежнему сохраняя свое входное значение и, подавая в этом случае на вход той же схемы "И" значение "1". Таким образом, единичные значения разрядов сохраняются, если впереди стоящими разрядными ПЭ на соответственные ЛС не была подана "1", которая в вертикальном наборе ПЭ инвертируется в "0".

В результате только ПЭ старшего ненулевого разряда примет значение "1", остальные получат значение "0". ПЭ с единичным значением даст на выход схемы сохраненный знак своего разряда [8].

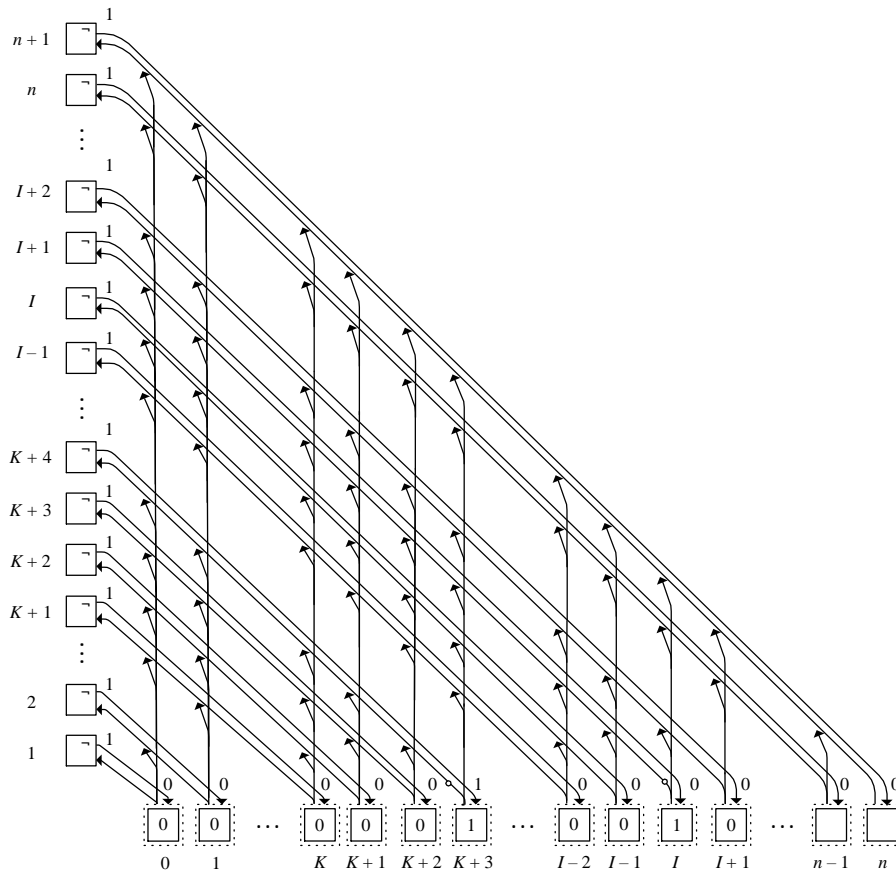


Рис. 1. Схема выделения старшего ненулевого разряда

Очевидно, схема максимально параллельна, выделяет знак старшего ненулевого разряда за время  $T(R)=O(1)$ , число ПЭ пропорционально числу разрядов:  $R=2n$ . Эта оценка инвариантна относительно числа символов слова.

**Разрядное распараллеливание поиска вхождения подстроки в строку.** Даны подстрока  $S$ , состоящая из  $m$  символов, и строка  $T$  из  $n$  символов. Пусть для простоты  $m < n$ . Выполнить поиск вхождения подстроки в строку на основе изложенного метода поразрядно-параллельного сравнения слов можно максимально параллельно за единичное время независимо от числа символов  $T$  и  $S$ . Для этого нужно одновременно выполнить выравнивание начального символа подстроки  $S$  с каждым последовательным символом строки  $T$  от ее начала. Далее, от каждого отдельно фиксированного символа строки  $T$ , по которому выполнено выравнивание с подстрокой  $S$ , без учета предшествующих символов, сравнение с подстрокой  $S$  выполняется так, как, если бы с этого символа начиналась строка. Собственно бинарное сравнение без изменения повторяет способ сравнения элементов строкового типа, изложенный выше, сравнение выполняется параллельно по всем символам и поразрядно-параллельно в их двоичной кодировке. Кроме того сравнение выполняется параллельно по всем сдвинутым символам.

Например, требуется выполнить поиск вхождения в строку "ПОИСК" строки "ИСК". Выравнивание по всем последовательным символам строки  $T$  начального символа подстроки  $S$  влечет соответственные сдвиги слова  $S$ :

$$T: \begin{array}{|c|c|c|c|c|} \hline П & О & И & С & К \\ \hline И & С & К & & \\ \hline & И & С & К & \\ \hline & & И & С & К \\ \hline \end{array}$$

Все сдвинутые слова  $S$  одновременно сравниваются с символами слова  $T$  :

$$T: \begin{array}{|c|c|c|c|c|} \hline П & О & И & С & К \\ \hline И & С & К & & \\ \hline \end{array} \quad S: \begin{array}{|c|c|c|c|c|} \hline О & И & С & К & \\ \hline И & С & К & & \\ \hline \end{array} \quad S: \begin{array}{|c|c|c|c|} \hline И & С & К & \\ \hline И & С & К & \\ \hline \end{array}$$

Сдвиг прекращается, точнее, не выполняется относительно всех тех символов строки, за пределы последнего из которых выходит сдвигаемая подстрока. Такой сдвиг реализуется параллельно и синхронно по всем символам строки.

Далее, выполняются операции предыдущего примера.

Каждый символ строки "ПОИСК" и подстроки "ИСК" представляется в ASCII-коде в двоичной форме. Набор двоичных кодов символов в последовательном расположении интерпретируется как единое число:

"ПОИСК": 11001111 11001110 11001000 11001010 11010001

"ИСК": 11001000 11001010 11010001

"ОИСК": 11001110 11001000 11001010 11010001

"ИСК": 11001000 11001010 11010001

"ИСК": 11001000 11001010 11010001

"ИСК": 11001000 11001010 11010001

На представленные пары чисел синхронно выполняется сравнение путем поразрядно-параллельного алгебраического сложения с кодом подстроки, которому приписывается отрицательный знак. Используется перевод вычитаемого в обратный код и корректировка с учетом аналога дополнительного кода (10)–(12).

В знакоразрядном коде получится –

при сравнении "ПОИСК" и "ИСК": 0000001110000010000–1110–11, старший ненулевой разряд равен 1, "ПОИСК" > "ИСК";

при сравнении "ОИСК" и "ИСК": 00000011000000–11000–111001, старший ненулевой разряд равен 1, "ОИСК" > "ИСК";

при сравнении "ИСК" и "ИСК": 000000000000000000000000, все разряды нулевые, "ИСК" = "ИСК", искомое вхождение идентифицировано.

Таким образом, подстрока  $S$  в строке  $T$  найдена за один максимально параллельный шаг выполнения сравнения.

Более точно, временная сложность предложенного поиска вхождения подстроки в строку оценивается из соотношения:

$$T(R) \approx 2\tau,$$

отсюда

$$T(R)=O(1),$$

где  $\tau$  – время переключения логического элемента, число ПЭ не больше значения, пропорционального числу разрядов строки, умноженному на число разрядов подстроки, в данных обозначениях –

$$R \leq m \times n.$$

Для сравнения заметим, что типичная оценка времени последовательного алгоритма рассматриваемого поиска составляет [9]  $T = O(mn)$ , наилучшие оценки [1]  $T \leq 2n$  и  $T = n$ , их различие определяется структурами алгоритмов.

Таким образом, предложенный способ соответственно дает ускорение в  $O(mn)$  и  $O(n)$  за счет роста числа разрядных компонентов.

Отметим, что предложенный способ, помимо параллелизма, отличается двумя существенными особенностями. Во-первых, единичная оценка времени сравнения и поиска вхождения формально сохраняется при неограниченном росте числа символов исследуемых слов. Во-вторых, предложенный способ распространяется на сравнение чисел, в том числе с плавающей точкой, без перевода в символьный формат [10].

**Заключение.** Представлен метод поразрядно-параллельного алгебраического сложения без вычисления переноса, который применяется для сравнения чисел и слов по знаку старшего ненулевого разряда знакоразрядной алгебраической суммы (арифметической разности). Предложен максимально параллельный схемный способ выделения этого знака. На этой основе вся бинарная операция полного разрядного сравнения выполняется за время  $T(R)=O(1)$ , которое формально не зависит от числа символов сравниваемых слов. При этом количество процессорных элементов пропорционально наибольшему числу символов пары сравниваемых слов. Метод ориентирован на применение в базовых операциях информационного поиска, в частности для выполнения сравнения в процессах сортировки и слияния. Показано, что применение метода к поиску вхождения подстроки в строку влечет максимальный параллелизм этой операции и ее выполнение с оценкой времени  $O(1)$ , которая формально не зависит от количества символов строки.

#### БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Вирт Н.* Алгоритмы и структуры данных. – М.: ДМК Пресс, 2010. – 272 с.
2. *Кнут Д.* Искусство программирования для ЭВМ. Т. 3. Сортировка и поиск. – М.: Вильямс, 2007. – 832 с.
3. *Ромм Я.Е.* Метод вертикальной обработки потока целочисленных групповых данных. I. Групповые арифметические операции // Кибернетика и системный анализ. – 1998. – № 3. – С. 123-151.
4. *Ромм Я.Е.* Метод вертикальной обработки потока целочисленных групповых данных. II. Приложение к бинарным арифметическим операциям // Кибернетика и системный анализ. – 1998. – № 6. – С. 146-162.
5. *Ромм Я.Е., Чабанюк Д.А.* Применение метода вертикальной обработки информации к операциям сравнения и поиска. – Таганрог: ТГПИ, 2013. – 32 с. Деп. в ВИНТИ 20.05.13, № 141 – В 2013.
6. *Ромм Я.Е., Иванова А.С.* Вертикальное групповое алгебраическое суммирование применительно к сортировке со слиянием и параллельному поиску. – Таганрог: ТГПИ., 2012. – 44 с. Деп. в ВИНТИ 03.09.2012, № 362 – В 2012.
7. *Ромм Я.Е., Иванова А.С.* Вертикальные групповые арифметические операции над целочисленными данными без вычисления переноса // Фундаментальные исследования. – 2012. – № 11 – С. 960-964.
8. *Ромм Я.Е., Чабанюк Д.А.* Сравнение слов с единичной временной сложностью. – Таганрог: ТГПИ, 2013. – 30 с. Деп. в ВИНТИ 30.10.13, № 306 – В 2013.
9. *Charras C., Lecroq T.* Handbook of Exact String Matching Algorithms. – College Publications. – 2004. – 256 p.
10. *Иванова А.С.* Расширение диапазона данных для вертикальной обработки применительно к сортировке со слиянием и поиску: Автореф. дис. ... канд. техн. наук. – Таганрог: ЮФУ, 2013. – 22 с.

#### REFERENCES

1. *Virt N.* Algorithmy i struktury dannykh [Algorithms and data structures]. Moscow: DMK Press, 2010, 272 p.
2. *Knut D.* Iskustvo programmirovaniya dlya EhVM. T. 3. Sortirovka i poisk [The art of computer programming. Vol. 3. Sorting and searching]. Moscow: Vilyams, 2007, 832 p.



3. Romm Ya.E. Metod vertikalnoy obrabotki potoka celochislennykh gruppovykh dannykh. I. Gruppyvye arifmeticheskie operacii [The method of vertical flow of integral group data. I. Group arithmetic operations], *Kibernetika i sistemnyy analiz* [Cybernetics and Systems Analysis], 1998, No. 3, pp. 123-151.
4. Romm Ya.E. Metod vertikalnoy obrabotki potoka tselochislennykh gruppovykh dannykh. II. Prilozhenie k binarnym arifmeticheskim operatsiyam [The method of vertical flow of integral group data. II. The Annex to the binary arithmetic operations], *Kibernetika i sistemnyy analiz* [Cybernetics and Systems Analysis], 1998, No. 6, pp. 146-162.
5. Romm Ya.E., Chabanyuk D.A. Primenenie metoda vertikalnoy obrabotki informatsii k operatsiyam sravneniya i poiska [Application of the method of vertical processing of information for comparisons and searches]. Taganrog: TGPI, 2013, 32 p. Dep. v VINITI 20.05.13, № 141 – V 2013.
6. Romm Ya.E., Ivanova A.S. Vertikalnoe gruppovoe algebraicheskoe summirovanie primenitelno k sortirovke so sliyaniem i parallelnomu poisku [The vertical group of the algebraic summation with respect to merge sort and parallel search]. Taganrog: TGPI., 2012, 44 p. Dep. v VINITI 03.09.2012, № 362 – V 2012.
7. Romm Ya.E., Chabanyuk D.A. Vertikalnye gruppovye arifmeticheskie operatsii nad tselochislennymi dannymi bez vychisleniya perenosa [The vertical group arithmetic integer data without calculation of transfer], *Fundamentalnye issledovaniya* [Fundamental Research], 2012, No. 11, pp. 960-964.
8. Romm Ya.E., Chabanyuk D.A. Sravnenie slov s edinichnoy vremennoy slozhnostyu [Compared with single time complexity]. Taganrog: TGPI, 2013, 30 p. Dep. v VINITI 30.10.13, № 306 – V 2013.
9. Charras C., Lecroq T. Handbook of Exact String Matching Algorithms. College Publications. 2004, 256 p.
10. Ivanova A.S. Rasshirenie diapazona dannykh dlya vertikalnoy obrabotki primenitelno k sortirovke so sliyaniem i poisku: Avtoref. dis. ... kand. tekhn. nauk [The expansion of the range of data for vertical processing applied to the merge sort and search. Cand. of eng. sc. abstract of thesis]. Taganrog: YuFU, 2013, 22 p.

Статью рекомендовал к опубликованию д.т.н., профессор Г.Е. Веселов.

**Ромм Яков Евсеевич** – ФГБОУ ВПО "Таганрогский государственный педагогический институт имени А.П. Чехова"; e-mail: romm@list.ru; 347926, г. Таганрог, ул. Инициативная, 48; тел.: 89094081126; кафедра информатики; зав. кафедрой; д.т.н.; профессор; член Европейской Академии естествознания (EuANH).

**Чабанюк Денис Андреевич** – e-mail: denchabanyuk@gmail.com; тел.: 89508498511; аспирант.

**Romm Yakov Evseevich** – Taganrog State Pedagogical Institute; e-mail: romm@List.ru; 48, Initsiativnaya street, Taganrog, 347926, Russia; phone: +79094081126; the department of information science; head of department; dr. of eng. sc.; professor; the member of European Academy of Natural History (EuANH).

**Chabanyuk Denis Andreevich** – e-mail: denchabanyuk@gmail.com; phone: +79508498511; postgraduate student.