

**РЕШЕНИЕ ЗАДАЧ ЛИНЕЙНОЙ АЛГЕБРЫ  
НА АССОЦИАТИВНОМ ПРОЦЕССОРЕ**

**В.Н.Алеева**

**Введение**

В данной работе приведены параллельные алгоритмы решения задач линейной алгебры на ассоциативном процессоре для крупноблочной обработки информации, описанном в [1]. Подробно рассматривается решение системы линейных уравнений.

Показано, что длительность решения пропорциональна размерности задачи  $\ell$ , т.е. порядку системы. Аналогичные результаты получены и для задач обращения матрицы и вычисления определителя. Раньше для ассоциативных процессоров была получена квадратичная зависимость от  $\ell$  [2]. Линейная оценка времени в нашем случае достигается благодаря наличию микрокоманды выделения максимума, а также за счет использования специальных коммутрующих матриц.

Автор не задавался целью описать самые быстрые алгоритмы, тем не менее оценка длительности выполнения приводимых алгоритмов доказывает целесообразность использования ассоциативных процессоров для решения задач линейной алгебры.

**Решение системы линейных уравнений**

В этом разделе будет рассмотрена некоторая реализация ал-

горитма Гаусса-Жордана решения системы линейных уравнений (СЛУ) с выбором ведущего элемента.

Пусть требуется решить СЛУ  $A_0 x = b_0$  ( $A_0$  имеет размеры  $l \times l$ ). СЛУ, полученную на  $i$ -м шаге, обозначим через  $A_i x = b_i$ . В качестве ведущего элемента  $i$ -го шага  $a_{i_2, j_2}$  берется элемент, удовлетворяющий условию максимума

$$|a_{i_2, j_2}^{(i-1)}| = \max |a_{ij}^{(i-1)}|,$$

$1 \leq i \leq l, i \neq i_1, \dots, i_{(i-1)}; 1 \leq j \leq l, j \neq j_1, \dots, j_{(i-1)}$   
и условию однозначности  $j_2 = \min j;$

$j: a_{ij}^{(i-1)}$  удовлетворяет условию максимума  
 $i_2 = \min i;$

$i: a_{ij_2}^{(i-1)}$  удовлетворяет условию максимума.

Преобразующая матрица  $P_i$  отличается от единичной только  $i_2$ -м столбцом, который равен

$$P_i^{i_2} = (c_1^i, \dots, c_l^i), \text{ где } c_i^i = -\frac{a_{ij_2}^{(i-1)}}{a_{i_2, j_2}^{(i-1)}}, (i \neq i_2);$$

$$c_{i_2}^i = \frac{1}{a_{i_2, j_2}^{(i-1)}};$$

$a_{ij}^{(i-1)}$  - элементы матрицы  $A_{(i-1)}$ .

Тогда все строки и столбцы матрицы  $A_i$  есть  $l$ -мерные векторы, у которых одна компонента равна единице, а остальные - нулю. Таким образом, либо  $x = b_i$ , либо вектор  $x$  может быть получен из  $b_i$  путем перестановки компонент.

Для описания реализации алгоритма на ассоциативном крупноблочном процессоре вводятся следующие обозначения:

$B_{i-1}$  - расширенная матрица системы  $A_{i-1} x = b_{i-1}$ ;

$B_i$  - матрица, полученная из  $B_{i-1}$  заменой элементов  $i$ -й строки нулями;

$C^i$  - матрица, состоящая из  $(l+1)$  одинаковых столбцов

$P_i^{i_2} B_i^{i_2}$  - матрица, состоящая из  $l$   $i_2$ -х строк матрицы  $B_{(i-1)}$ .

Согласно введенным обозначениям, имеет место равенство

$B_i = P_i B_{(i-1)} = B_{(i-1)}^{i_2} + C^i \times B^i$  ( $C^i \times B^i$  - покомпонентное произведение матриц  $C^i$  и  $B^i$ ).

Таким образом, на  $z$ -м шаге необходимо вычислить матрицу  $P^z = C^z \times B^z$ , а затем сумму  $B_{(z-1)}^{-b_z} + P^z$ .

На рис. 1 показано распределение памяти накопителя при решении СЛУ,  $k$ -я строка рабочего поля имеет следующую структуру:  $F_{k1}^{(1)} F_{k2}^{(2)} F_{k3}^{(3)} F_{k0}^{(4)} (F_{k1}^{(4)} \dots F_{k1}^{(6)}) \dots (F_{km}^{(6)} \dots F_{km}^{(6)}) (F_{k,m+1}^{(2)} \dots F_{k,m+1}^{(6)}) \dots (F_{k,2m}^{(2)} \dots F_{k,2m}^{(6)}) F_{k1}^{(7)} F_{k1}^{(8)} F_{k1}^{(9)}$ . Все матрицы в накопителе размещаются вертикально последовательно по столбцам. Числа представляются в системе с фиксированной запятой и имеют разрядность  $m$ .

Исходная информация размещается так: в ячейки

$$F_{k1}^{(1)} F_{k1}^{(1)} \dots F_{km}^{(1)} \quad (k = 1, \dots, l(l+1))$$

записывается матрица  $B_0$ . Против элемента  $b_{ij}^{(1)}$  ( $i=1, \dots, l, j=1, \dots, l+1$ ) матрицы  $B_0$  во вспомогательном поле № I (ВП I) записан индекс  $i$ , а в ВП 2 - индекс  $j$ . Кроме того, для решения задачи необходимо специальным образом настроить коммутирующие матрицы  $K_1, K_2, M_1, M_2$ . Эта настройка будет описана ниже.

Перейдём теперь к описанию реализации  $z$ -го шага алгоритма.

### 1. Поиск ведущего элемента

Здесь необходимо заметить, что в столбце меток  $b_1$  после выполнения  $(z-1)$ -го шага помечены те элементы  $a_{ij}^{(z-1)}$  матрицы  $A_{(z-1)}$ , для которых  $i = i_1, \dots, i_{(z-1)}$ , либо  $j = j_1, \dots, j_{(z-1)}$ . Таким образом, для выполнения условия максимума необходимо найти элемент  $a_{ij}^{(z-1)}$  такой, что он является максимальным по абсолютной величине среди элементов, не помеченных в столбце меток  $b_1$ , а для выполнения условия однозначности искомый элемент должен иметь, в отличие от других элементов, удовлетворяющих условию максимума, наименьший номер строки рабочего поля.

Максимальный элемент среди не помеченных в  $b_1$  выделяется с помощью микрокоманды ПМ (см. [1]). При этом может быть выделено несколько равных максимальных элементов. На рис. 2 изображена схема ячейки столбца индикаторов. После выделения максимальных элементов на выходах  $z d_k$  столбца индикаторов вырабатывается вектор-столбец  $z d$ , в котором первому выделенному элементу соответствует единица, а остальным - нули. Вектор  $z d$  записывается в столбец меток  $b_3$ . Таким образом, в  $b_3$  помечен ведущий элемент.

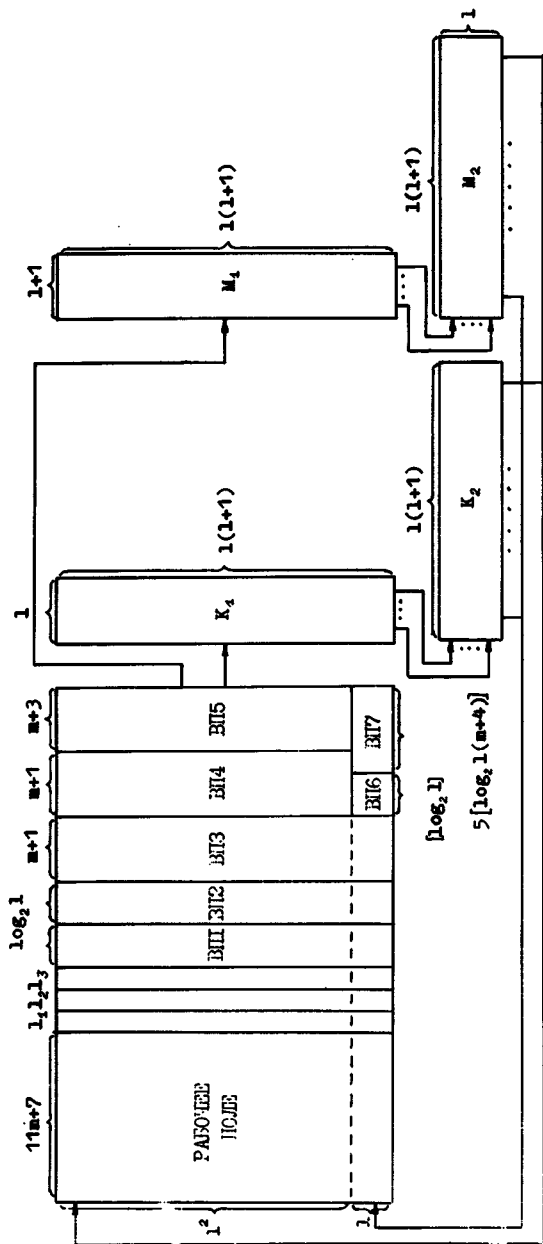


Рис. I

Далее производятся следующие вспомогательные операции:

- а) в столбце  $l_1$  помечаются элементы  $i_2$ -й строки и  $j_2$ -го столбца;
- б) в столбце  $l_2$  помечаются элементы  $i_2$ -й строки;
- в) в  $i_2$ -ю строку поля ВП 6 записывается  $j_2$ .

Для выполнения этих операций используется содержимое полей ВП 1 и ВП 2.

## 2. Вычисление матрицы $C^2$

$j_2$ -й столбец матрицы  $A_{(2-1)}$  переписывается последовательно по разрядам, включая знаковый разряд, в ВП 3. Вместо ведущего элемента в ВП 3 записывается единица. В ВП 4 во все строки записывается ведущий элемент. Затем выполняется деление ВП 3 на ВП 4. Применяемая здесь микропрограмма ДЕЛЕНИЕ отличается от микропрограммы, описанной в [1], так как абсолютные величины частных не превосходят единицы. В данном случае микропрограмма ДЕЛЕНИЕ вычисляет  $(m+1)$  разряд дробной части частного, при этом частные записываются в ВП 5. Затем выполняется округление по младшему разряду. Знаки частных во всех строках, кроме помеченной в  $l_3$ , заменяются на противоположные.

Таким образом, в строках, содержащих  $j_2$  в ВП 2, записан столбец матрицы  $C^2$ . ВП 3 и ВП 4 очищаются.

Формирование матрицы  $G^2$  можно выполнить с помощью коммутирующих матриц  $K_1$  и  $K_2$  (рис. 1). В принципе, матрицы  $K_1$  и  $K_2$  можно построить из  $\theta$ -ячеек (см. [1]), которые обеспечивают режим коммутации. Однако для экономии оборудования можно использовать упрощенные ячейки ( $\lambda$ -ячейки), в которых содержатся только необходимые для коммутации цепи.

На рис. 3 показана функциональная схема  $\lambda$ -ячейки. Входы  $x$  левого крайнего столбца матрицы  $K_1$  должны быть соединены с выходами  $x'$  правого крайнего столбца матрицы-накопителя, входы  $x$  левого крайнего столбца матрицы  $K_2$  соединены с выходами  $x'$  нижней строки матрицы  $K_1$ , а выходы  $x'$  нижней строки матрицы  $K_2$  — с управляющими входами левой границы матрицы-накопителя.

При решении СЛУ настройка матриц  $K_1$  и  $K_2$  выполняется один раз, например перед реализацией алгоритма. Весь процесс настройки требует  $2l$  команд и выполняется следующим образом.

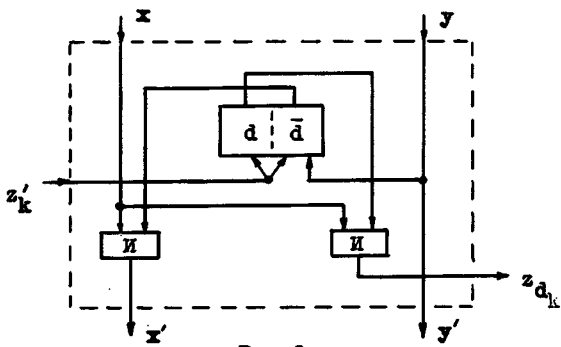


Рис.2

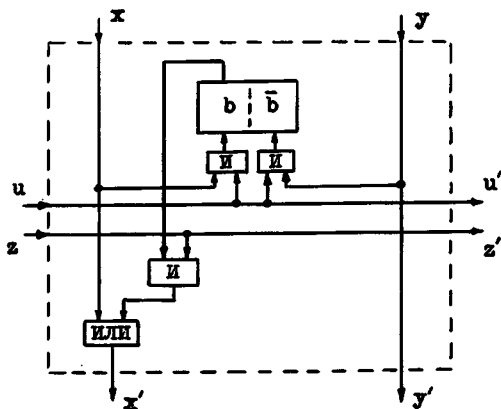


Рис.3

Сначала настраивается матрица  $K_1$ . Для этого в  $S$ -й ( $S = 1, \dots, b$ ) команде производится запись в  $S$ -й столбец вектора

$$\underbrace{\frac{0 \dots 0}{s-1} \ 1 \ \frac{0 \dots 0}{b-s} \ \dots \ \frac{0 \dots 0}{s-1} \ 1 \ \frac{0 \dots 0}{b-s}}_{b+1}.$$

При настройке матрицы  $K_2$  в  $S$ -й команде производится запись в  $S$ -ю строку вектора

$$\underbrace{\frac{0 \dots 0}{s-1} \ 1 \ \frac{0 \dots 0}{b-s} \ \dots \ \frac{0 \dots 0}{s-1} \ 1 \ \frac{0 \dots 0}{b-s}}_{b+1}.$$

Формирование матрицы  $C^2$  выполняется последовательно по разрядам за  $(m+1)$  шаг.

$p$ -й ( $p = 1, \dots, m$ ) шаг заключается в том, что на  $p$ -й разряд столбца  $c_1^2, \dots, c_b^2$ , записанного в ВП 5, подается признак "1", а слово, полученное на выходах  $x'$  нижней строки матрицы  $K_2$ , поступает на управляющие входы левой границы матрицы-накопителя и записывается в столбец ВП 3, предназначенный для  $p$ -го разряда.

$(m+1)$ -й шаг выполняется так же, как предыдущие, но производится формирование не столбца одноименных разрядов, а столбца знаков. После этого ВП 5 очищается.

### 3. Ф о р м и р о в а н и е м а т р и ц ы $B^2$

Формирование матрицы  $B^2$  можно выполнить с помощью коммутирующих матриц  $M_1$  и  $M_2$ , которые, как  $K_1$  и  $K_2$ , можно построить из  $\lambda$ -ячеек. Связи матриц  $M_1$  и  $M_2$  с матрицей-накопителем такие же, как у  $K_1$  и  $K_2$ . Настройка матрицы  $M_1$  ( $M_2$ ) выполняется за  $(b+1)$  команду следующим образом: в  $S$ -й команде ( $S = 0, \dots, b$ ) в строки (столбцы) с номера  $(bs+1)$ -го по  $b(s+1)$ -й записывается вектор

$$\frac{0 \dots 0}{s-1} \ 1 \ \frac{0 \dots 0}{b-s}.$$

Формирование матрицы  $B^2$  выполняется последовательно по разрядам за  $(m+1)$  шаг. Шаг  $p$ -й ( $p = 1, \dots, m$ ) заключается в том, что на  $p$ -й разряд элементов строки с номером  $i_2$  подается признак "1", а слово, полученное на выходах  $x'$

нижней строки матрицы  $M_2$ , поступает на управляющие входы левой границы матрицы-накопителя и записывается в ячейки  $F_{k,m+p}^{(6)}$  тех же строк, что и матрицы  $A_{(z-1)}$  и  $C^z$ .

$(m+1)$ -й шаг аналогичен предыдущим, но поиск ведется по столбцу знаков, а запись - в ячейки  $F_{kz}^{(0)}$ .

#### 4. Вычисление матрицы $P^z$

Для вычисления матрицы  $P^z$  воспользуемся микропрограммой **БЫСТРОЕ УМНОЖЕНИЕ**, описанной в [1]. В качестве ячеек  $R_{kp}^z, S_{kp}^z, S_{kp}^1, R_{kp}^1, C_{kp}$  используются соответственно ячейки  $F_{kp}^{(2)}, F_{kp}^{(3)}, F_{kp}^{(4)}, F_{kp}^{(5)}, F_{kp}^{(6)}$ . Выполним (с удвоенной точностью) покомпонентное умножение матриц  $B^z$  и  $C^z$ . В результате разряды элементов матрицы  $P^z$  содержатся в ячейках  $F_{kp}^{(4)}$ , причём произведение  $c_i^z \times a_{i_2,j}^{(z-1)}$  ( $c_i^z \times b_{i_2}^{(z-1)}$ ) находится в одной строке с элементом  $a_{ij}^{(z-1)}$  ( $b_{i_2}^{(z-1)}$ ). Знак произведения  $c_i^z \times a_{i_2,j}^{(z-1)}$  ( $c_i^z \times b_{i_2}^{(z-1)}$ ) формируется в ячейке  $F_{b(ij-1)+i,2}^{(0)}$  ( $F_{b^2+i,2}^{(0)}$ ) с помощью анализа знаков  $C_i^z$  и  $a_{i_2,j}^{(z-1)}$  ( $C_i^z$  и  $b_{i_2}^{(z-1)}$ ). Затем ВП 3 и все ячейки рабочего поля, кроме  $F_{kz}^{(0)}, F_{kz}^{(1)}, F_{k.}, F_{k.}^{(4)}$ , очищаются.

#### 5. Вычисление расширенной матрицы $B_z$

Перед вычислением матрицы  $B_z$  элементы  $i_2$ -й строки матрицы  $B_{(z-1)}$  заменяются нулями. Множество пар

$$M = \{ (a_{ij}^{(z-1)}, c_i^z \times a_{i_2,j}^{(z-1)}), (b_{i_2}^{(z-1)}, c_i^z \times b_{i_2}^{(z-1)}) \} \quad i, j = 1, \dots, b,$$

разбивается на два подмножества: первое подмножество - пары, элементы которых имеют одинаковые знаки, второе подмножество - пары, элементы которых имеют разные знаки. Пары второго подмножества намечаются в ячейках  $F_{k.}^{(2)}$ , и абсолютные величины их элементов сравниваются путём последовательного поразрядного сравнения, начиная со старших разрядов. При этом заполняются ячейки  $F_{k.}^{(8)}$  и  $F_{k.}^{(9)}$  так, что после окончания сравнения против пар  $(a, b)$  в ячейках  $F_{k.}^{(8)}$  и  $F_{k.}^{(9)}$  стоят метки

- 01, если  $|a| < |b|$ ;
- 00, если  $|a| = |b|$ ;
- 10, если  $|a| > |b|$ .



В результате множество  $M$  оказывается разбитым на подмножества  $M^1$ ,  $M^2$  и  $M^3$ . В табл. I приведены метки, соответствующие парам, принадлежащим этим подмножествам, и указаны операции, которые необходимо выполнить над элементами пар.

Т а б л и ц а I.

Подмножества	Метки			Операции при вычислении $B_z$
	$F_{\kappa}^{(1)}$	$F_{\kappa}^{(2)}$	$F_{\kappa}^{(3)}$	
$M^1$	0	X	X	Сложение абсолютных величин элементов пар
$M^2$	1	1	X	Вычисление разности абсолютных величин первого и второго элементов
$M^3$	1	0	X	Вычисление разности абсолютных величин второго и первого элементов

Согласно табл. 2 опрашиваются ячейки  $F_{\kappa_1}^{(1)}$ ,  $F_{\kappa}^{(2)}$ ,  $F_{\kappa}^{(3)}$  и в  $F_{\kappa_3}^{(0)}$  записываются знаки элементов матрицы  $B_z$ .

Т а б л и ц а 2.

$F_{\kappa_1}^{(1)}$	$F_{\kappa}^{(2)}$	$F_{\kappa}^{(3)}$	$F_{\kappa}^{(0)}$
0	0	X	0
0	I	I	0
0	I	0	I
I	0	X	I
I	I	I	I
I	I	0	0

$F_{\kappa_1}^{(1)}$  очищается, а содержимое  $F_{\kappa_3}^{(0)}$  переписывается в  $F_{\kappa_1}^{(0)}$ . Вычисление абсолютных величин элементов матрицы  $B_z$  выполняется отдельно для подмножеств  $M^1$  и  $M^2$ ,  $M^3$ . Сна-

ж) Символом обозначено содержимое ячеек, которое в данном случае безразлично.

чала выполняется последовательное по разрядам сложение абсолютных величин элементов пар, принадлежащих  $M^1$ . При этом результат записывается в ячейки  $F_{k0}^{(1)} F_{k1}^{(1)} \dots F_{km}^{(1)} F_{k,m+1}^{(1)} \dots \dots F_{k,2m}^{(1)}$  (для хранения переносов используется ячейка  $F_{k0}^{(1)}$ ). Для совмещения вычислений над элементами подмножеств  $M^2$  и  $M^3$  в строках, содержащих элементы пар подмножества  $M^2$ , необходимо поменять местами содержимое ячеек  $F_{kp}^{(2)}$  и  $F_{kp}^{(1)}$  ( $p=1, \dots, m$ ). Содержимое ячеек  $F_{kp}^{(4)}$  ( $p=1, \dots, m$ ) переписывается в  $F_{kp}^{(3)}$ ,  $F_{kp}^{(4)}$  ( $p=1, \dots, m$ ) очищаются. Затем вычисляется разность абсолютных величин элементов, записанных в ячейках  $F_{kp}^{(4)}$  ( $p=1, \dots, 2m$ ), и элементов, записанных в  $F_{kp}^{(1)}$  и  $F_{k,m+p}^{(1)}$  ( $p=1, \dots, m$ ) для пар из  $M^2$  и  $M^3$ ; при этом результат записывается в ячейки  $F_{k0}^{(1)}$ ,  $F_{kp}^{(1)}$  и  $F_{k,m+p}^{(1)}$  ( $p=1, \dots, m$ ).

Итак, элементы матрицы  $B_2$  вычислены,  $(m+1)$  старших разрядов записаны в ячейках  $F_{k0}^{(1)} \dots F_{km}^{(1)}$ , а  $m$  младших - в  $F_{k,m+p}^{(1)}$  ( $p=1, \dots, m$ ). Столбец  $b_3$ , а также все ячейки рабочего поля, кроме  $F_{k1}^{(1)}$ ,  $F_{k0}^{(1)}$ ,  $F_{kp}^{(1)}$ ,  $F_{k,m+p}^{(1)}$  ( $p=1, \dots, m$ ), очищаются.

## 6. З а м е ч а н и я о б у с т о й ч и в о с т и

Устойчивость процесса обеспечивается выполнением условий

$$|a_{ij}^{(2)}| < 1, \quad |b_i^{(2)}| < 1, \quad (i, j = 1, \dots, l); \quad (1)$$

$$|c_i^{(2)}| \leq 1, \quad (i = 1, \dots, l). \quad (2)$$

Можно считать, что в ячейках  $F_{k1}^{(1)} \dots F_{km}^{(1)} F_{k,m+1}^{(1)} \dots F_{k,2m}^{(1)}$  записаны дробные части элементов. Тогда для выполнения (1) необходимо сдвинуть вправо на один разряд элементы тех строк матрицы  $B_2$ , хотя бы один элемент которых содержит в  $F_{k0}^{(1)}$  единицу. Для этого признак "1" подается на столбец ячеек  $F_{k0}^{(1)}$  и слово, полученное на выходах  $x'$  матрицы  $K_2$  записывается в  $b_3$ . Все элементы, помеченные в  $b_3$ , необходимо сдвинуть вправо на один разряд, при этом  $(2m+1)$ -е разряды сдвигаемых элементов пропадают, а в ячейках  $F_{k1}^{(1)} \dots F_{km}^{(1)} F_{k,m+1}^{(1)} F_{k,2m}^{(1)}$  оказываются остальные  $2m$  разрядов. Элементы ВП 8, помеченные в  $b_2$  и  $b_3$ , необходимо увеличить на единицу. Это выполняется одним шагом "сокращенного сложения" [1],  $b_3$  очищается.

Таким образом, по окончании выполнения всего алгоритма в  $i$ -й ( $i = 1, \dots, \ell$ ) строке поля ВП 7 будет записано число сдвигов  $i$ -й строки расширенной матрицы после того, как в ней был выбран ведущий элемент.

Необходимо заметить, что на  $(\tau+1)$ -м шаге максимальный элемент выбирается не среди всех элементов матрицы  $A_z$ , а среди не помеченных в  $\ell_1$  элементов, поэтому условие (2), вообще говоря, не выполняется. Для выполнения условия (2) производится следующее.

1) Одновременно сдвигаются влево элементы строк, не помеченных в  $\ell_2$  так, чтобы хотя бы в одной ячейке  $F_{\kappa_1}^{(1)}$  была записана единица. Производится это с помощью  $(2m-1)$  циклов. Цикл заключается в том, что дизъюнктивно считывается содержимое ячеек  $F_{\kappa_1}^{(1)}$  строк рабочего поля, не содержащих в  $\ell_2$  единицу, и результат чтения записывается во все ячейки столбца  $\ell_3$ . На  $\ell_2$  и  $\ell_3$  подается признак "00", в выделенных строках рабочего поля производится сдвиг элементов влево на один разряд, а  $\ell_3$  очищается. После выполнения  $(2m-1)$ -го циклов дизъюнктивно считывается содержимое ячеек  $F_{\kappa_1}^{(1)}$  первых  $\ell^2$  строк матрицы-накопителя, не содержащих в  $\ell$  единицу.

а) Если результат считывания есть "0", то дизъюнктивно считывается содержимое ячеек  $F_{\kappa_1}^{(1)}$  строк матрицы-накопителя с  $(\ell^2+1)$ -й по  $\ell(\ell+1)$ -ю, не содержащих в  $\ell_2$  единицу; если результат считывания также "0", то делается вывод о том, что СЛУ имеет бесчисленное множество решений, в противном случае ячейки  $F_{\kappa_1}^{(1)} \dots F_{\kappa_m}^{(1)} F_{\kappa_{m+1}}^{(1)}$  опрашиваются признаком  $\frac{0 \dots 0}{m+1}$  и значение столбца индикаторов записывается в  $\ell_3$ , а затем  $\ell_3$  дизъюнктивно считывается; если результат считывания "0", то СЛУ несовместна.

б) Если результат считывания "1", то алгоритм выполняется дальше.

2) Производится сдвиг элементов строк, помеченных в  $\ell_2$ , так, что первые  $\min(h, m+1)$  разрядов равны нулю,  $h$  - номер первого равного единице разряда  $y$  элементов матрицы  $A_z$ , не помеченных в  $\ell_2$ .

Для выполнения такого сдвига считаются дизъюнктивно не помеченные в  $\ell_2$  элементы матрицы  $A_z$ .

Полученный вектор  $d_z$  (в нем первая единица стоит на  $h$ -м месте) записывается в некоторую  $\kappa_1$ -ю строку рабочего поля в

ячейки  $F_{\kappa_1 1}^{(2)} \dots F_{\kappa_1 m}^{(2)} F_{\kappa_1 m+1}^{(4)} \dots F_{\kappa_1, 2m}^{(4)}$ .

Затем вырабатывается вектор

$$d'_y = \frac{0 \dots 0}{h} \frac{1 \dots 1}{2m-h}.$$

Для этого на шины  $\bar{v}$  ячеек  $F_{11}^{(1)} \dots F_{1m}^{(1)} F_{1, m+1}^{(3)} \dots F_{1, 2m}^{(3)}$  и на шины  $w$  ячеек  $F_{11}^{(2)} \dots F_{1m}^{(2)} F_{1, m+1}^{(4)} \dots F_{1, 2m}^{(4)}$  подается вектор  $d_y$ . Одновременно с этим производится запись в  $F_{\kappa_1 1}^{(2)} \dots F_{\kappa_1 m}^{(2)} F_{\kappa_1, m+1}^{(4)} \dots F_{\kappa_1, 2m}^{(4)}$  единичного вектора. В результате в  $F_{\kappa_1 1}^{(2)} \dots F_{\kappa_1 m}^{(2)} F_{\kappa_1, m+1}^{(4)} \dots F_{\kappa_1, 2m}^{(4)}$  окажется вектор  $d'_z$ . Необходимый сдвиг элементов выполняется за  $S$  ( $S \leq h$ ) циклов.

Цикл заключается в следующем. Вектор  $d'_z$  подается в качестве маски опроса в ячейки  $F_{\kappa_1}^{(1)} \dots F_{\kappa m}^{(1)} F_{\kappa, m+1}^{(3)} \dots F_{\kappa, 2m}^{(3)}$  строк, помеченных в  $b_2$ , а в качестве признака опроса подается нулевой вектор. Результат опроса записывается в  $b_3$ . Столбцы меток  $b_2$  и  $b_3$  опрашиваются кодом "10", и результат опроса записывается в очищенный столбец рабочего поля  $b'_1$ . Если результат дизъюнктивного чтения столбца  $b'_1$  равен "0", то сдвиг заканчивается, в противном случае  $b'_1$  считывается и подается на  $K_1$ , выходной вектор матрицы  $K_2$  записывается в предварительно очищенный столбец меток  $b_3$ .

Элементы, помеченные в  $b_2$  и  $b_3$ , сдвигаются вправо на один разряд, при этом числа поля ВП 7, помеченные в  $b_3$ , необходимо увеличить на единицу. На этом цикл заканчивается. Таким образом, через  $S$  циклов первые  $h$  разрядов элементов, помеченных в  $b_2$ , равны нулю. Тем самым условие (2) полностью удовлетворяется.

## 7. Округление результатов

Так как разрядность элементов на каждом шаге растет, то необходимо выполнять округление, оставляя  $m$  старших разрядов. Для этого в каждой строке рабочего поля суммируется одноразрядное число, записанное в  $F_{\kappa, m+1}^{(3)}$  и  $m$ -разрядное чис-

ло, записанное в  $F_{k1}^{(1)} \dots F_{km}^{(1)}$ . В результате сложения может получиться  $(m+1)$ -разрядное число, т.е. в ячейке  $F_{k0}^{(1)}$  может оказаться единица. Это нарушает условие (1). В данном случае условие (1) обеспечивается аналогично тому, как это делалось выше. После этого может оказаться, что нарушено условие (2). Выполнение условия (2) достигается, как и раньше. Теперь может оказаться, что в ячейках  $F_{k,m+1}^{(3)}$  содержится единица. Поэтому снова суммируются одноразрядное число, записанное в  $F_{k,m+1}^{(3)}$ , и  $m$ -разрядное число, записанное в  $F_{k1}^{(1)} \dots F_{km}^{(1)}$ . В этом случае нарушения условий (1) и (2) не произойдет. Затем ячейки  $F_{k,m+1}^{(3)} \dots F_{k,2m}^{(3)}$  очищаются. На этом округление заканчивается.

### 8. Считывание результатов

Для считывания результата используется  $(b+1)$ -й столбец матрицы  $B_l$  и поля ВП 6 и ВП 7. Пусть вектор решений есть  $x = (x_1, \dots, x_b)$ , а  $B_b = b_{ij}^{(b)}$ .

Тогда, если в  $k$ -й строке поля ВП 6 находится число  $p$ , а ВП 7 - число  $d_k$ , то  $x_p = 2^{d_k} b_{k,b+1}^{(b)}$ . Таким образом, система  $A_0 x = b_0$  решена полностью.

Как видно из описания, длительность выполнения  $z$ -го шага алгоритма Гаусса-Жордана не зависит от размерности задачи, а только от разрядности чисел  $m$ . Таким образом, длительность выполнения всего алгоритма зависит от  $l$  линейно.

**ЗАМЕЧАНИЕ.** Описанная выше методика решения системы линейных уравнений на ассоциативном крупноблочном процессоре может быть распространена на другие задачи линейной алгебры. Так, нами были рассмотрены задачи обращения матрицы и вычисления определителя. Для этих задач была также получена линейная зависимость времени решения от размерности  $l$ . Расход памяти в случае обращения матрицы

$$S_{0M} = 2l^2 (2[\log_2 l] + 14m + 15),$$

$K_1(K_2)$  имеет размеры  $2l^2 \times l$  ( $l \times 2l^2$ ). В случае вычисления определителя

$$S_{B0} = 95 S_{0M} + 5(b+1)[\log_2 b+1] + l[\log_2 l] + 34m + 22,$$

$M_1(M_2)$  имеет размеры  $l^2 \times l$  ( $l \times l^2$ ).

## Л и т е р а т у р а

1. АЛЕЕВА В.Н., ФЕТ Я.И. Ассоциативный процессор для крупно-блочной обработки информации.- в кн.: Оптимизация. вып. 15(32), Новосибирск, 1974, с. 154-177.
2. Crane B.A., Githens J.A. Bulk processing in distributed logic memory.- JEEE. Trans., 1965, EC-14, №2, p.186-196.

Поступила в ред.-изд. отд.

12. II. 1974 г.