1972 г. ОПТИМИЗАЦИЯ Выпуск 6(23)

УДК 681.32.001/3.06

МИКРОПРОГРАММНЫЙ ОДНОРОДНЫЙ ПРОЦЕССОР, ОРИЕНТИРОВАННЫЙ НА ЗАДАЧИ ЛИНЕЙНОЙ АЛГЕБРЫ

Ю.Н.Корнев, С.В.Пискунов, С.Н.Сергеев

Введение

В настоящей статье описывается логическая схема процессора для решения задач линейной алгебры. Работа процессора иллюстрируется на примере выполнения алгоритма решения задачи линейного программирования. В процессоре реализуются операции над "крупными" единицами информации (матрицами, векторами), причём возможно одновременное выполнение операций над массивами чисел, расположенными в произвольных местах памяти процессора. Процессор может быть эффективным средством реализации автоматической крупноблочной системы программирования, которая обсуждалась в работе Л.В.Канторовича [1].

Разработка логической схемы процессора, проведенная в данной статье, является практическим примером осуществления идеи построения однородных вычислительных устройств, предложенной авторами в работах [2,3,4].

Основу этой идеи составляет построение такой алгоритмической системы (алгоритмов обобщенных подстановок), которая дает с п о с о б п е р е х о д а от записи алгоритма (набора алгоритмов) решения задачи (набора задач) к вычислительному устройству, выполняющему этот алгоритм (набор алгоритмов).

Алгоритмы обобщенных подстановок и способ перехода от записи алгоритма к логической схеме вычислительного устройства описаны в работе [4]. Поэтому в статье, после краткой харак-

теристики представления информации, основное внимание уделено вопросам формирования набора алгоритмов (таких как алгоритм умножения матрицы на строку, сортировка и пересылка массивов и др.), которые можно рассматривать как "кирпичики" для построения более сложных алгоритмов. В качестве сложного алгоритма, который строится из данных простых, был взят алгоритм решения задачи линейного программирования. Этот алгоритм был записан в виде алгоритма обобщенных подстановок, и для него была проведена предварительная оценка времени работы (в количестве тактов).

§ I. Процессор

Процессор представляет собой множество клеток (автоматов), соединенных в трехмерный массив. В дискретные моменты времени по общим шинам из устройства управления на все клетки (или некоторые группы клеток) поступают команды. Каждая клетка изменяет свое состояние и состояния некоторых других клеток, с которыми соединен ее выход записи, причем эти изменения являются функцией состояния данной клетки, поступившей на нее команды и состояний некоторого множества клеток-соседей, от которых она получает входную информацию.

§ 2. Структура памяти процессора.

Пусть c_1, c_2, c_3, n_1, n_2, n_3 -целые числа и $1 \le c_1 \le n_1$, $1 \le c_2 \le n_2$, $1 \le c_3 \le n_3$.

Память процессора есть клеточное множество, состоящее из четырех подмножеств - блоков (рис. I).

Первый блок памяти процессора будем называть рабочим полем (РП). Каждая клетка этого блока имеет имя $(1, c_1, c_2, c_3)$, где I - номер блока, c_1, c_2, c_3 - координаты клетки.

Второй блок памяти процессора назовем полем меток (ПМ). Именем клетки этого блока называется двойка чисел $(2, c_1)$.

Третий и четвертый блоки: блок маски (БМ) и рабочий регистр (РР) имеют одинаковый размер n_2 и имена их клеток соответственно обозначаются двойками чисел $(3, c_2)$, $(4, c_2)$.

Клеточное множество с именами $\{(I, x_1, x_2, c_3) / x_1 = 1,2,\ldots,n_1;\ x_2 = 1,2,\ldots,n_2\}$ назовем слоем c_3.

Клеточное множество с именами $\{(I, c_1, x_2, c_3) / x_2 = 1,2,\ldots,n_2\}$ назовем ячейкой c_1 в слое c_3 или c_1-й ячейкой c_3-го слоя.

Пусть клетка $(2, c_1)$ поля меток имеет состояние α, тогда ячейку c_1 в слое c_3 будем называть ячейкой слоя c_3 с меткой α.

Пусть ячейка c_1 в слое c_3 имеет вид:

$$\{(s_1,(1,c_1,1,c_3)),(s_2,(1,c_1,2,c_3)),\ldots,(s_{n_2},(1,c_1,n_2,c_3))\}.$$

Слово $S = s_1, s_2, \ldots, s_{n_2}$ - будем называть кодом ячейки c_1 в слое c_3. В этом случае будем также говорить, что ячейка c_1 в слое c_3 находится в состоянии S.

Аналогичным образом определяются код блока маски, код рабочего регистра и понятия состояний блока маски и рабочего регистра.

Пусть БМ находится в состоянии $T = t_1, t_2, \ldots, t_{n_2}$ и $t_{i_1} t_{i_2} \ldots t_{i_k} = \underbrace{\mu\mu\ldots\mu}_{k \text{ раз}}$, все другие $t_{i_j} \neq \mu$ и ячейка c_1 в слое c_3 находится в состоянии S. Слово $S_1 = s_{i_1} s_{i_2} \ldots s_{i_k}$ будем называть частью кода μ (или кодом под маской μ) ячейки c_1 в слое c_3.

Аналогично определяется часть кода регистра РР.

§ 3. Представление информации в памяти процессора

Вся информация задачи хранится и перерабатывается в РП процессора.

Каждый элемент квадратного массива A представляется ячейкой с меткой α, которую будем интерпретировать как имя массива A. Пусть блок маски находится в состоянии СС...СТТ. ...ТЧЧ...Ч. Часть кода $Ч$ ячейки с меткой α представляет значение $A[i,j]$, часть кода C ячейки с меткой α представляет значение i, часть кода T - значение j. В некоторых случаях места i и j могут поменяться.

Линейный массив кодируется таким же образом, только часть кода C (или T) ячеек массива или не используется или используется для представления других массивов такой же структуры.

Скалярные величины представляются частью кода U ячейки с меткой α , где метка α является именем этой скалярной величины.

Рассмотрим представление массивов задачи линейного программирования (ЛП) в рабочем поле процессора (рис. 2). Обозначения массивов и индексы (за небольшим исключением), а также схема алгоритма взяты из работы Гасса [5] стр. 119-135 .

Массив A записан в РП по столбцам и отмечен меткой M_1 . Ячейки с меткой M_2 являются разделителями столбцов массива A.

Часть кода C ячейки с меткой M_1 представляет значение индекса строки, а часть кода T этой ячейки - значение индекса столбца. Отметим, что элементы $A[i,j]=0$ можно исключать из представления массива A .

Массив X записан в РП полностью, с нулевыми элементами, и отмечен меткой M_3 . Ячейка с меткой M_4 отмечает конец массива X . Часть кода T ячейки с меткой M_3 представляет значение индекса i , а часть кода C ячейки с меткой M_3 служит для представления соответствующего номера переменной $X[i]$ в базисе (массив NX).

Массив Y отмечен меткой M_5 (каждое $Y[i]$ соответствует x_{ik}). Ячейка с меткой M_6 отмечает конец массива Y Порядковый индекс i помещен в частях C и T ячейки с меткой M_5 .

Массив U записан в РП по строкам и отмечен меткой M_7 . Часть кода C ячейки с этой меткой представляет значение индекса столбца, а часть кода T представляет значение индекса строки. Разделителями строк служат ячейки с меткой M_8 , ячейки с такими же метками являются резервом памяти при преобразовании обратной матрицы. Конец массива отмечается ячейкой с меткой M_9 .

§ 4. Описание программы реализации алгоритма решения задачи линейного программирования (ЛП)

Элементарной операцией над памятью процессора является обобщенная подстановка, применяемая по правилам, которые описаны в работах авторов [2,4]. Обобщенная подстановка за один рабочий такт преобразует содержимое памяти процессора во всех

местах там, где эта подстановка применима.

Описание программ решения задач в процессоре имеет несколько уровней.

Назовем программой нулевого уровня одноподстановочный алгоритм обобщенных подстановок, программой первого уровня - композицию программ нулевого уровня, программой второго уровня - композицию программ нулевого и первого уровней.

Программа алгоритма решения задачи ЛП использует композицию программ второго уровня и обобщенных подстановок, определенных на поле меток.

Описания программ первого и второго уровней, используемых в программе ЛП, сведены в таблицу I. Эти программы являются крупноблочными операторами алгоритма решения задачи. Содержательный смысл программ описан в графе 7 таблицы 1. В этой же графе описывается смысл формальных параметров этих подпрограмм.

В таблице I отсутствуют описания программы сложения группы чисел $МСл$ (α, ℓ, β), а также программ покомпонентных арифметических операций над двумя массивами, таких как сложение $Сл$ $(\alpha, \ell_1, \ell_2, \ell_3, \beta)$, вычитание $Вч$ $(\alpha, \ell_1, \ell_2, \ell_3, \beta)$, умножение $Ум$ $(\alpha, \ell_1, \ell_2, \ell_3, \beta)$, деление $Д$ $(\alpha, \ell_1, \ell_2, \ell_3, \beta)$, поскольку их запись не встречает принципиальных трудностей, но достаточно громоздка (в простейшем случае для чисел с фиксированной запятой операции сложения группы чисел и умножения двух чисел приведены в [6]). Здесь мы ограничимся только указанием содержательного смысла формальных параметров, используемых в этих подпрограммах. Подпрограмма $МСл$ вычисляет сумму частей кода β ячеек с меткой α, расположенных в слое ℓ. Все ячейки с меткой α расположены "плотно", сумма получается в самой правой ячейке с меткой α. Все остальные подпрограммы для ячеек с меткой α выполняют соответствующие операции над частями кода β ячеек с меткой α, расположенных в слоях ℓ_1, ℓ_2, а результат получается в ℓ_3.

Программа решения задачи ЛП описана в таблице 2.

Правила композиции подстановок и алгоритмов описаны в [4]. В данном случае смысл композиции заключается в следующем: подстановки и программы выполняются в том порядке, как они записаны в графе 3 таблиц I и 2, если в графах 4, 5 этих таблиц не задан (именами меток) условный переход. Иначе порядок выполнения зависит от применимости очередной обобщенной подстановки и определяется указанными метками.

§ 5. Краткое описание структуры процессора

Как отмечалось во введении, существует способ перехода от записи алгоритмов стационарных обобщенных подстановок к вычислительному устройству, выполняющему эти алгоритмы.

Ниже приводится краткое описание логической схемы процессора.

1^0. Элементарная клетка памяти процессора имеет такую же блок-схему, как ячейка памяти данных однородной машины из статьи [4], и несколько большую разрядность входных шин правой и левой части команды.

2^0. Среди обобщенных подстановок программ таблицы 1 можно выделить два класса A и B "нелокальных" обобщенных подстановок. У каждой обобщенной подстановки из A именующая функция левой части имеет вид $(2, x_1)$. Каждой обобщенной подстановке из A при фиксированном имени, например $(2, c_1)$, соответствует класс подстановок, множество имен в контексте которых таково: $\{(1, c_1, 1, 5), (1, c_1, 2, 5), \ldots (1, c_1, n_2, 5)\}$. При построении памяти процессора соответствующие входы клетки с именем $(2, c_1)$ поля меток соединяются с выходами всех клеток с именами $(1, c_1, 1, 5), (1, c_1, 2, 5), \ldots (1, c_1, n_2, 5)$ рабочего поля процессора. Один из возможных способов реализации таких соединений показан на рис. 1: выходы клеток $(1, c_1, 1, 5), \ldots (1, c_1, n_2, 5)$ собираются, а выход сборки соединяется с одним из входов клетки $(2, c_1)$. (Сборка вынесена из клетки $(2, c_1)$.) Кроме того, на этом рисунке показаны входы от соответствующих клеток блока маски и рабочего регистра. Если считать, что клетки $\{(2, c_1), (1, c_1, 1, 1), \ldots, (1, c_1, n_2, 5)\}$ образуют модуль c_1, то тогда память процессора практически является однородным устройством, построенным из таких модулей.

У каждой обобщенной подстановки из B именующая функция левой части имеет вид $(4, x_2)$. Каждой обобщенной подстановке из B при фиксированном имени, например $(4, c_2)$, соответствует класс подстановок, множество имен в контексте которых таково: $\{(1, 1, c_2, 5), (1, 2, c_2, 5), \ldots, (1, n_1, c_2, 5)\}$. При построении памяти процессора соответствующие входы клетки с именем $(4, c_2)$ рабочего регистра соединяются с выходами всех клеток с именами $(1, 1, c_2, 5), \ldots, (1, n_1, c_2, 5)$. Эти соединения также могут быть осуществлены при помощи сборки. К этому же классу относят-

ся обобщенные подстановки, определенные на поле меток и использующие условный переход.

При практической реализации процессора следует ожидать, что команды, соответствующие обобщенным подстановкам из В, потребуют на свое выполнение больше времени, чем команды, соответствующие обобщенным подстановкам из А, а те, в свою очередь, потребуют больше времени на выполнение, чем оставшиеся команды. В соответствии со сказанным работой процессора управляют три тактирующих последовательности T_3, T_2, T_1 с длительностями тактов, соответственно, τ_3, τ_2, τ_1; $\tau_3 > \tau_2 > \tau_1$.

3°. Сопоставление подстановкам команд описано в [4]. В нашем случае команда имеет следующий формат:

Л	П	M_1	M_2	T	N	W

где Л - левая часть команды,
П - правая часть команды,
M_1 - первый адрес,
M_2 - второй адрес,
T - номер тактирующей последовательности,
N - имена блоков памяти с точностью до указания номера слоя,
W - символ записи.

В памяти устройства управления хранятся набор подпрограмм (таблица I) и рабочая программа (таблица 2).

Устройство управления сначала переводит подпрограмму, предназначенную к выполнению, в программу нулевого уровня, а затем начинает выполнять команды следующим образом.

Анализируется очередная команда. При этом, исходя из номера T тактирующей последовательности, назначается время выполнения команды. Символ W пересылается в управляющий разряд регистра R_5 - регистра выдачи правой и левой части команды в пятый слой памяти процессора (см. рис. 3). Левая и правая части команды пересылаются в выходные регистры устройства управления, соответствующие N. Затем эти регистры выдают команду на все клетки соответствующих блоков памяти процессора. Если подстановка, записанная данной командой, применима к информации, перерабатываемой в памяти процессора, и M_1, M_2 не

есть прочерки, то адресом следующей команды будет M_1 , если подстановка не применима, то – M_2 . Если M_1 и M_2 – прочерки, то выполняется команда, следующая в записи программы за выполненной.

§ 6. Заключение

1^0. В работе рассмотрен алгоритм решения практической задачи. Это рассмотрение показало эффективность алгоритмов обобщенных подстановок как алгоритмического языка: программы получаются короткими и достаточно эффективными (по количеству тактов), в программах почти отсутствуют операции перекомпоновки массивов. Это объясняется тем, что алгоритмы обобщенных подстановок осуществляют одновременную и массовую переработку информации с произвольной структурой.

С другой стороны, отметим простоту перехода от записи алгоритмов обобщенных подстановок к логическим схемам устройств, выполняющих эти алгоритмы.

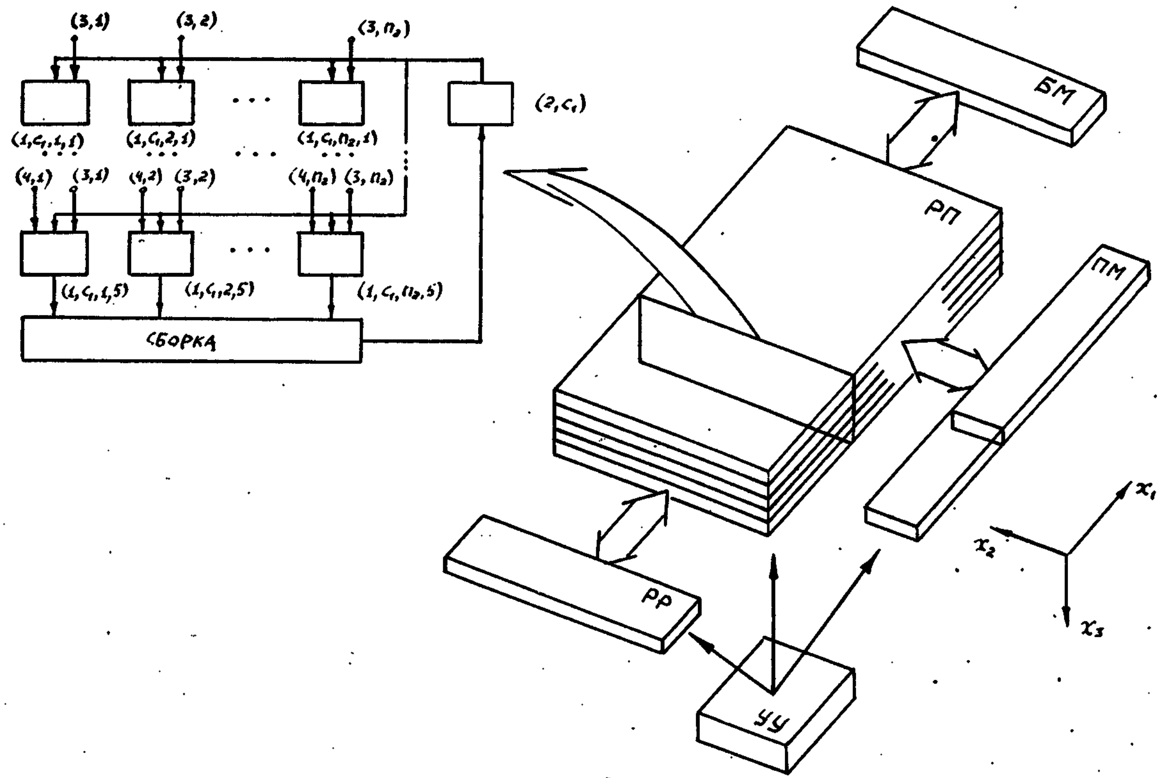
2^0. В предложенных программах, чтобы облегчить первую проработку алгоритма, были выбраны простейшие способы организации хранения массивов в рабочей памяти процессора и пересылки массивов из одной части памяти процессора в другую. Поэтому формула оценки времени работы алгоритма на одной итерации (см. примечание 2 к таблице 2) содержит член, зависящий от m . Можно показать, что существуют другие способы размещения и пересылки массивов информации, уменьшающие эту зависимость. В этом направлении предстоит дополнительная работа.

Литература

1. Канторович Л.В. Перспективы работы в области автоматизации программирования на базе крупноблочной системы.-"Труды Матем. ин-та АН СССР им. Стеклова", 1968, 96, с.5 - 15.
2. Корнев Ю.Н., Пискунов С.В., Сергеев С.Н. Алгоритмы обобщенных подстановок и вопросы их интерпретации.-"Теоретическая кибернетика," труды семинара, 4, Киев, 1970.
3. Корнев Ю.Н., Пискунов С.В., Сергеев С.Н. Об одном подходе к построению и использованию однородных вычислительных устройств.-"Вычислительные системы", 1971, вып.46,с. 130-133.

4. Корнев Ю.Н., Пискунов С.В., Сергеев С.Н. Алгоритмы обобщенных подстановок и вопросы их интерпретации сетями автоматов и однородными машинами. - "Известия АН СССР",1971,6, сер.технич.киберн.,с. 131-142.
5. Гасс С. Линейное программирование. М.,"Физматгиз", 1961.
6. Корнев Ю.Н., Пискунов С.В., Сергеев С.Н. Вопросы построения алгоритмов обобщенных подстановок с выделенным контекстом.-"Вычислительные системы", 1972, вып.47,с. 117-130.

Поступила в ред.-изд.отдел
29 февраля 1972 г.



(3,1)
(3,2)
(3,n2)
(2,c1)
(1,c1,1,1)
(1,c1,2,1)
(1,c1,n2,1)
(4,1)
(3,1)
(4,2)
(3,2)
(4,n2)
(3,n2)
(1,c1,1,5)
(1,c1,2,5)
(1,c1,n2,5)
СБОРКА
БМ
РП
ПМ
РР
УУ
x1
x2
x3

Рис.1

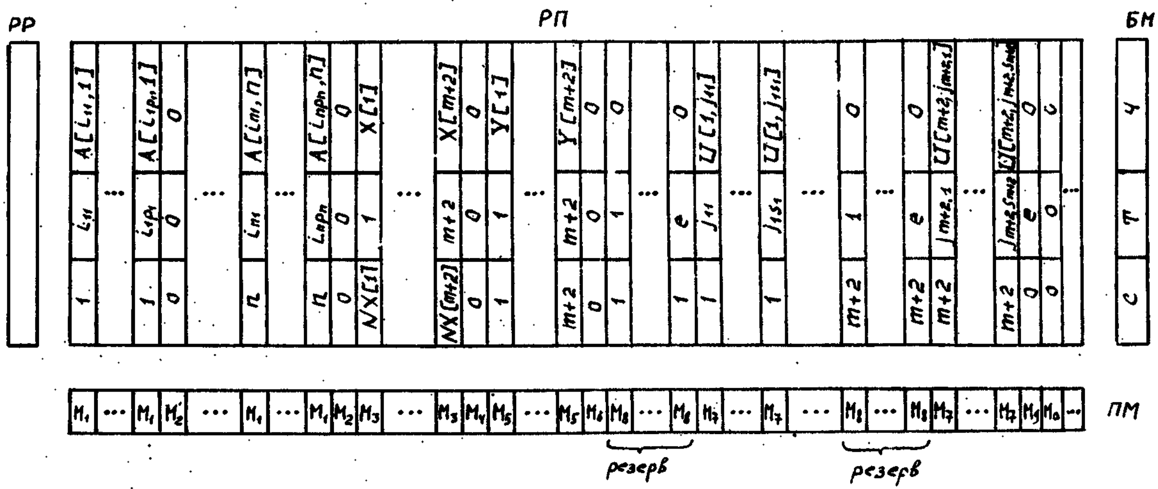


Рис.2

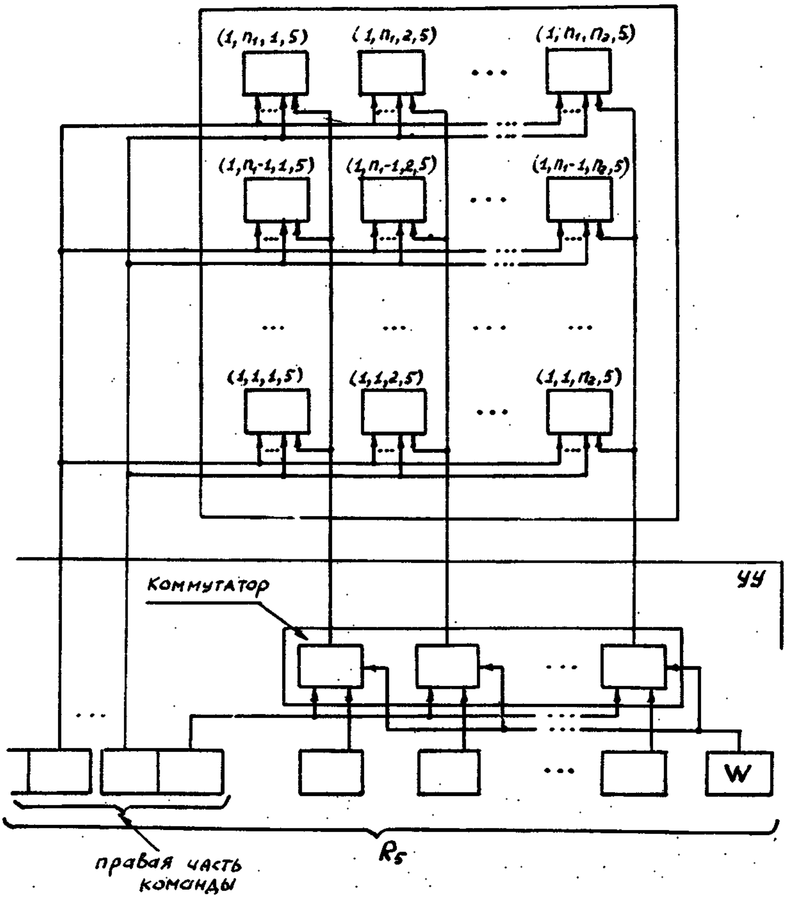

(1, n₁, 1, 5)
(1, n₁, 2, 5)
(1, n₁, n₂, 5)
(1, n₁-1, 1, 5)
(1, n₁-1, 2, 5)
(1, n₁-1, n₂, 5)
(1, 1, 1, 5)
(1, 1, 2, 5)
(1, 1, n₂, 5)
Коммутатор
УУ
W
правая часть
команды
R₅

Рис.3

Приложение

Т а б л и ц а I

Имена программ	Метки	Программы и подстановки	Переход если: Применима	Переход если: Не применима	Время	Содержательный смысл программ и формальных параметров.
I	2	3	4	5	6	7
$\Pi_1(\alpha, \ell)$	1	$(1,(1,x_1,x_2,5))*(\alpha,(2,x_1))\to(0)$	—	—	τ_1	В ячейку ℓ -го слоя с меткой α засылается код с РР.
	2	$(1,(1,x_1,x_2,\ell))*(\alpha,(2,x_1))\to(0)$	—	—	τ_1	
	3	$(0,(1,x_1,x_2,5))*\{(\alpha,(2,x_1)),(1,(4,x_2))\}\to(1)$	—	—	τ_1	
	4	$(0,(1,x_1,x_2,\ell))*\{(\alpha,(2,x_1)),(1,(1,x_1,x_2,5))\}\to(1)$	—	—	τ_1	

I	2	3	4	5	6	7
$\Pi_2(\alpha,\mu,\beta)$	1	$\Pi_1(\alpha,2)$	—	—		Перевод метки α, соответствующей ячейкам I-го слоя, у которых часть кода μ равна части кода μ в РР, в метку β. α_1 – рабочая метка
	2	$(1,(1,x_1,x_2,5))*(\alpha,(2,x_1))\rightarrow(0)$	—	—	τ_1	
	3	$(0,(1,x_1,x_2,5))*\{(0,(1,x_1,x_2,1)),(1,(1,x_1,x_2,2)),(\alpha,(2,x_1)),(\mu,(3,x_2))\}\rightarrow(1)$	—	—	τ_1	
	4	$(0,(1,x_1,x_2,5))*\{(1,(1,x_1,x_2,1)),(0,(1,x_1,x_2,2)),(\alpha,(2,x_1)),(\mu,(3,x_2))\}\rightarrow(1)$	—	—	τ_1	
	5	$(\alpha,(2,x_1))*(1,(1,x_1,x_2,5))\rightarrow(\alpha_1)$	—	—	τ_2	
	6	$(\alpha,(2,x_1))\rightarrow(\beta)$	—	—	τ_1	
	7	$(\alpha_1,(2,x_1))\rightarrow(\alpha)$	—	—	τ_1	
$\Pi_3(\alpha,\ell)$	1	$(1,(4,x_2))\rightarrow(0)$	—	—	τ_1	Код из ячейки ℓ-го слоя с меткой α пересылается в РР.
	2	$(1,(1,x_1,x_2,5))*(\alpha,(2,x_1))\rightarrow(0)$	—	—	τ_1	
	3	$(0,(1,x_1,x_2,5))*\{(\alpha,(2,x_1)),(1,(1,x_1,x_2,\ell))\}\rightarrow(1)$	—	—	τ_1	
	4	$(0,(4,x_2))*(1,(1,x_1,x_2,5))\rightarrow(1)$	—	—	τ_3	

I	2	3	4	5	6	7
$\Pi_4(\mu,\alpha,\beta,\gamma,\ell)$	1	$(\alpha,(2,x_1)) * (\beta,(2,x_1+1)) \to (\alpha_1)$	–	–	τ_1	Перепись массива (α,β) в массив γ слоя ℓ в соответствии с порядком, закодированным в части кода μ ячеек с меткой α. α_1, γ_1 - рабочие метки.
	2	$\Pi_3(\alpha_1, 1)$				
	3	$\Pi_2(\gamma,\mu,\gamma_1)$				
	4	$\Pi_1(\gamma_1,\ell)$				
	5	$(\gamma_1,(2,x_1)) \to (\gamma)$	–	–	τ_1	
	6	$\{(\alpha,(2,x_1)),(\alpha_1,(2,x_1+1))\} \to \{\alpha_1,\alpha_2\}$	2	7	τ_3	
	7	$(\alpha_1,(2,x_1)) \to (\alpha)$	–	–	τ_1	
$\Pi_5(\gamma,\alpha,\beta,\mu)$	1	$\Pi_4(\mu,\alpha,\beta,\gamma,2)$				Умножение матрицы γ на вектор (α,β). Значение индексов столбцов помещено в части кода μ.
	2	$УМ(\gamma,1,2,1,4)$			$t_{УМ}$	
	3	$МСл(\gamma,1,4)$			$t_{МСл}$	
$\Pi_6(\alpha,\beta,\ell_1,\gamma,\ell_2,\mu)$	1	$(\alpha,(2,x_1)) * (\beta,(2,x_1+1)) \to (\alpha_1)$	–	–	τ_1	Перепись массива γ слоя ℓ_2 в массив (α,β) слоя ℓ_1. Значение соответствующих индексов кодируется частью
	2	$\Pi_3(\alpha_1,\ell_1)$				
	3	$\Pi_2(\gamma,\mu,\gamma_1)$				
	4	$\Pi_3(\gamma_1,\ell_2)$				

I	2	3	4	5	6	7
	5	$\Pi_1(\alpha_1, \ell_1)$				кода μ ячеек с меткой α и γ. α_1, γ_1 - рабочие метки.
	6	$(\gamma_1,(2,x_1)) \to (\gamma)$	–	–	τ_1	
	7	$\{(\alpha,(2,x_1)),(\alpha_1,(2,x_1+1))\} \to \{\alpha_1,\alpha\}$	2	8	τ_3	
	8	$(\alpha_1,(2,x_1)) \to (\alpha)$	–	–	τ_1	
$\Pi_7(\alpha,\ell,\beta)$	1	$(4,(3,x_2)) * (T,(3,x_2-1)) \to (Z)$	–	–	τ_1	Если число слоя ℓ с меткой α меньше нуля, то α заменяется на β. Z - рабочий символ маски.
	2	$(1,(1,x_1,x_2,5)) * (\alpha,(2,x_1)) \to (0)$	–	–	τ_1	
	3	$(0,(1,x_1,x_2,5)) * \{(\alpha,(2,x_1)),(Z,(3,x_2)),$ $(1,(1,x_1,x_2,\ell))\} \to (1)$	–	–	τ_1	
	4	$(\alpha,(2,x_1)) * (1,(1,x_1,x_2,5)) \to (\beta)$	–	–	τ_2	
	5	$(Z,(3,x_2)) \to (4)$	–	–	τ_1	
$\Pi_8(\alpha,\ell,\beta)$	1	$(1,(1,x_1,x_2,5)) * (\alpha,(2,x_1)) \to (0)$	–	–	τ_1	Если число в слое ℓ с меткой α равно нулю, то метка α меняется на метку β.
	2	$(0,(1,x_1,x_2,5)) * \{(1,(1,x_1,x_2,\ell)),$ $(\alpha,(2,x_1)),(4,(3,x_2))\} \to (1)$	–	–	τ_1	
	3	$(\alpha,(2,x_1)) * (1,(1,x_1,x_2,5)) \to (\alpha_1)$	–	–	τ_2	α_1 - рабочая метка.

I	2	3	4	5	6	7
	4	$(\alpha,(2,x_1)) \to (\beta)$	—	—	τ_1	
	5	$(\alpha_1,(2,x_1)) \to (\alpha)$	—	—	τ_1	
$\Pi_9(\alpha,\mu,\ell_1,\ell_2)$	1	$(1,(1,x_1,x_2,\ell_2)) * \{(4,(3,x_2)),(\mu,(3,x_2)),(\alpha,(2,x_1))\} \to (0)$	—	—	τ_1	Пересылка части кода μ ячейки с меткой α слоя ℓ_1 в слой ℓ_2.
	2	$(0,(1,x_1,x_2,\ell_2)) * \{(1,(1,x_1,x_2,\ell_1)),(\mu,(3,x_2)),(4,(3,x_2)),(\alpha,(2,x_1))\} \to (1)$	—	—	τ_1	
	1	$(1,(1,x_1,x_2,5)) * (\alpha,(2,x_1)) \to (0)$	—	—	τ_1	Среди частей кода μ ячеек с меткой α слоя ℓ_1 отыскивается минимум. Ячейки, в которых этот минимум достигается, отмечаются меткой β. α_1 -рабочая метка.
	2	$(0,(1,x_1,x_2,5)) * \{(\alpha,(2,x_1)),(\mu,(3,x_2))\} \to (1)$	—	—	τ_1	
	3	$(1,(1,x_1,x_2,5)) * (1,(1,x_1,x_2-1,5)) \to (0)$	—	—	τ_1	
	4	$(\alpha,(2,x_1)) \to (\beta)$	—	—	τ_1	
$\Pi_{10}(\alpha,\ell_1,\mu,\beta)$	5	$(1,(1,x_1,x_2,5)) * \{(\beta,(2,x_1)),(0,(1,x_1,x_2,\ell_1))\} \to (1)$	6	8	τ_3	

I	2	3	4	5	6	7
	6	$(1,(1,x_1,x_2,5)) * \{(\beta,(2,x_1)),(1,(1,x_1,x_2,\ell_1,))\} \to (0)$	–	–	τ_1	
	7	$(\beta,(2,x_1)) * (0,(1,x_1,x_2,5)) \to (\alpha_1)$	–	–	τ_1	
	8	$(0,(1,x_1,x_2+1,5)) * \{(1,(1,x_1,x_2,5)), (\mu,(3,x_2+1)),(\beta,(2,x_1))\} \to (1)$	–	–	τ_1	
	9	$(1,(1,x_1,x_2-1,5)) * \{(1,(1,x_1,x_2,5)), (\beta,(2,x_1))\} \to (0)$	5	10	τ_3	
	10	$(\alpha_1,(2,x_1)) \to (\alpha)$	–	–	τ_1.	
$\Pi_{11}(\alpha,\ell_1,\ell_2)$	1	$(1,(1,x_1,x_2,\ell_2)) * \{(0,(1,x_1,x_2,\ell_1)), (\alpha,(2,x_1))\} \to (0)$	–	–	τ_1	Пересылка ячейки с меткой α из слоя ℓ_1 в слой ℓ_2.
	2	$(0,(1,x_1,x_2,\ell_2)) * \{(1,(1,x_1,x_2,\ell_1)), (\alpha,(2,x_1))\} \to (1)$	–	–	τ_1	

I	2	3	4	5	6	7
$\Pi_{12}(\alpha, \mu, \ell_1, \ell_2, \beta)$	1	$(1,(1,x_1,x_2,5)) * (\alpha,(2,x_1)) \to (0)$	—	—	τ_1	Если часть кода μ ячейки с меткой α слоя ℓ_1 равна части кода μ ячейки с меткой α слоя ℓ_2, то метка α заменяется на метку β. α_1 – рабочая метка
	2	$(0,(1,x_1,x_2,5)) * \{(0,(1,x_1,x_2,1)), (1,(1,x_1,x_2,2)), (\alpha,(2,x_1)), (\mu,(3,x_2))\} \to (1)$	—	—	τ_1	
	3	$(0,(1,x_1,x_2,5)) * \{(1,(1,x_1,x_2,1)),(0,(1,x_1,x_2,2)), (\alpha,(2,x_1)), (\mu,(3,x_2))\} \to (1)$	—	—	τ_1	
	4	$(\alpha,(2,x_1)) * (1,(1,x_1,x_2,5)) \to (\alpha_1)$	—	—	τ_2	
	5	$(\alpha,(2,x_1)) \to (\beta)$	—	—	τ_1	
	6	$(\alpha_1,(2,x_1)) \to (\alpha)$	—	—	τ_1	
$\Pi_{13}(\alpha, \ell)$	1	$(\otimes,(1,x_1,x_2,\ell)) * \{(\alpha,(2,x_1)), (S,(1,x_1-1, x_2,\ell))\} \to (S)$	—	—	τ_1	Ячейка с меткой α слоя ℓ принимает значение ячейки слоя ℓ слева. Сдвиг меток α вправо.
	2	$(\otimes,(2,x_1)) * (\alpha,(2,x_1-1)) \to (\alpha)$	—	—	τ_1	
	1	$(\otimes,(1,x_1,x_2,2)) * \{(\alpha,(2,x_1)), (S,(1,x_1-1, x_2,2))\} \to (S)$	—	—	τ_1	Ячейки с меткой α слоя 2, 3, 4, принимают значение слева.
	2	$(\otimes,(1,x_1,x_2,3)) * \{(\alpha,(2,x_1)), (S,(1,x_1-1,$				

I	2	3	4	5	6	7
$П_{14}(\alpha)$		$x_2, 3))\} \to (S)$	—	—	τ_1	
	3	$(\otimes, (1, x_1, x_2, 4)) * \{(\alpha, (2, x_1)), (S, (1, x_1-1, x_2, 4))\} \to (S)$	—	—	τ_1	
$П_{15}(\alpha, \ell)$	1	$(\otimes, (1, x_1, x_2, \ell)) * \{(\alpha, (2, x_1)), (S, (1, x_1+1, x_2, \ell))\} \to (S)$	—	—	τ_1	Ячейка с меткой α слоя ℓ принимает значение ячейки слоя ℓ справа. Сдвиг меток α влево.
	2	$(\otimes, (2, x_1)) * (\alpha, (2, x_1+1)) \to (\alpha)$	—	—	τ_1	
	1	$\{(\beta, (2, x_1)), (\alpha, x_1+1))\} \to \{\alpha, \beta_1\}$	2	11	τ_3	Перестановка ячеек с меткой α первого и второго слоя влево до края массива ячеек с меткой β :
	2	$(\otimes, (1, x_1, x_2, 3)) * \{(S, (1, x_1, x_2, 1)), (\beta_1, (2, x_1))\} \to (S)$	—	—	τ_1	
	3	$(\otimes, (1, x_1, x_2, 4)) * \{(S, (1, x_1, x_2, 2)), (\beta_1, (2, x_1))\} \to (S)$	—	—	τ_1	
	4	$(\otimes, (1, x_1, x_2, 1)) * \{(S, (1, x_1-1, x_2, 1)), (\beta_1, (2, x_1))\} \to (S)$	—	—	τ_1	
	5	$(\otimes, (1, x_1, x_2, 2)) * \{(S, (1, x_1-1, x_2, 2)),$				

1	2	3	4	5	6	7
$\Pi_{16}(\alpha,\beta)$		$(\beta_1,(2,x_1))\} \to (S)$	–	–	τ_1	
	6	$(\otimes,(1,x_1,x_2,x_3)) * \{(S,(1,x_1+1,x_2,3)),$ $(\alpha,(2,x_1))\} \to (S)$	–	–	τ_1	
	7	$(\otimes,(1,x_1,x_2,4)) * \{(S,(1,x_1+1,x_2,4)),$ $(\alpha,(2,x_1))\} \to (S)$	–	–	τ_1	
	8	$(\otimes,(1,x_1,x_2,1) * \{(S,(1,x_1,x_2,3)),$ $(\alpha,(2,x_1))\} \to (S)$	–	–	τ_1	
	9	$(\otimes;(1,x_1,x_2,2)) * \{(S,(1,x_1,x_2,4)),$ $(\alpha,(2,x_1))\} \to (S)$	–	–	τ_1	
	10	$(\beta_1,(2,x_1)) \to (\beta)$	1	1	τ_3	
	11	$(\beta_1,(2,x_1)) \to (\beta)$	–	–	τ_1	
	1	$((\langle\otimes,\otimes\rangle,(2,x_1)) \to (\langle\otimes,0\rangle)$	–	–	τ_1	
	2	$((\langle M_7,0\rangle,(2,x_1)) \to (\langle M_7,1\rangle)$	–	–	τ_1	
	3	$((\langle M_8,0\rangle,(2,x_1)) \to (\langle M_8;1\rangle)$	–	–	τ_1	

I	2	3	4	5	6	7
Π_{17}	4	$(\langle M_9, 0\rangle, (2, x_1)) \to (\langle M_9, 1\rangle)$	—	—	τ_1	Упорядочивание памяти U.
	5	$(\langle M_0, 0\rangle, (2, x_1)) \to (\langle M_0, 1\rangle)$	—	—	τ_1	
	6	$(\langle L_1, 0\rangle, (2, x_1)) \to (\langle L_1, 1\rangle)$	—	—	τ_1	
	7	$(\langle L_3, 0\rangle, (2, x_1)) \to (\langle L_3, 1\rangle)$	—	—	τ_1	
	8	$(\langle M_8, 1\rangle, (2, x_1+1)) * (\langle \otimes, 0\rangle, (2, x_1)) \to$ $\to (\langle M_8, 2\rangle)$	—	—	τ_1	
	9	$\Pi_{14}(\langle M_8, 2\rangle, 1, 2)$				
	10	$\Pi_{11}(\langle M_8, 2\rangle, 2, 3)$				
	11	$\Pi_3(\langle M_9, 1\rangle, 1)$				
	12	$\Pi_1(\langle M_8, 2\rangle, 4)$				
	13	$\Pi_{12}(\langle M_8, 2\rangle, T, 1, 2, \langle M_8, 3\rangle)$				
	14	$(\langle M_8, 3\rangle, (2, x_1)) \to (\langle M_8, 2\rangle)$	19	15	τ_3	
	15	$(\langle \otimes, 2\rangle, (2, x_1)) \to (\langle \otimes, 1\rangle)$	—	—	τ_1	
	16	$\Pi_{15}(\langle \otimes, 1\rangle, 1)$				

I	2	3	4	5	6	7
	17	$(\langle\otimes,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\to$ $\to(\langle M_8,2\rangle)$	–	–	τ_1	
	18	$\Pi_{11}(\langle M_8,2\rangle,2,1)$				
	19	$\Pi_{12}(\langle M_8,2\rangle,T,2,4,\langle M_8,3\rangle)$				
	20	$(\langle M_8,3\rangle,(2,x_1))\to(\langle M_8,0\rangle)$	25	21	τ_3	
	21	$\{(\langle M_8,2\rangle,(2,x_1)),(\langle\otimes,1\rangle,(2,x_1+1))\}\to$ $\to(\langle M_8,0\rangle,\langle\otimes,2\rangle)$	–	–	τ_1	
	22	$\Pi_{14}(\langle\otimes,2\rangle)$				
	23	$Сл(\langle\otimes,2\rangle,2,3,2,T)$			$t_{сл}$	
	24	$(\langle M_8,2\rangle,(2,x_1))\to(\langle M_8,2\rangle)$	13	15	τ_3	
	25	$(\langle M_7,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\to$ $\to(\langle M_7,2\rangle)$	26	33	τ_3	
	26	$\Pi_8(\langle M_7,2\rangle,1,\langle M_7,3\rangle)$				
	27	$(\langle M_7,3\rangle,(2,x_1))\to(\langle M_7,1\rangle)$	28	32	τ_3	

I	2	3	4	5	6	7
	28	$\Pi_{15}(\langle\otimes,1\rangle,1)$				
	29	$(\langle\otimes,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\rightarrow(\langle\otimes,2\rangle)$	—	—	τ_1	
	30	$(\langle M_9,2\rangle,(2,x_1))\rightarrow(\langle M_9,0\rangle)$	36	31	τ_3	
	31	$(\langle M_8,2\rangle,(2,x_1))\rightarrow(\langle M_8,2\rangle)$	9	26	τ_3	
	32	$\{(\langle\otimes,2\rangle,(2,x_1)),(\langle\otimes,1\rangle,(2,x_1+1))\}\rightarrow\{\langle\otimes,0\rangle,\langle\otimes,2\rangle\}$	30	30	τ_3	
	33	$(\langle L_1,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\rightarrow(\langle L_1,0\rangle)$	33	34	τ_3	
	34	$(\langle L_3,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\rightarrow(\langle L_3,0\rangle)$	—	—	τ_1	
	35	$(\langle L_1,1\rangle,(2,x_1+1))*(\langle\otimes,0\rangle,(2,x_1))\rightarrow(\langle L_1,0\rangle)$	35	8	τ_3	
	36	$(\langle M_0,1\rangle,(2,x_1))\rightarrow(\langle M_0,0\rangle)$	—	—	τ_1	

Примечания к таблице I

1. $s \in \{0,1\}$.

2. Символом $\otimes$ обозначается произвольное состояние клетки.

3. Каждая клетка поля меток может принимать состояние из алфавита пар $\langle \alpha, i \rangle$, где α - буквенный символ (возможно, с индексом), а i - целое число. Для краткости в программах, кроме Π_{17}, состояние $\langle \alpha, \otimes \rangle$ везде записывалось как α.

4. Массив ячеек с меткой α, расположенный плотно в РП и ограниченный справа ячейкой с меткой β, в программах называется массивом (α, β).

Т а б л и ц а 2

Соответствие алгоритму по Гассу	Метки	Операторы и подстановки	Переход по применимости		Время
			да	нет	
1	2	3	4	5	6
Проверка	1	$(M_3, (2, x_1)) * (M_4, (2, x_1+1)) \to (L_1)$	—	—	τ_1
$x_{n+m+2} < 0$	2	$\Pi_7 (L_1, 1, M_3)$			
	3	$(L_1, (2, x_1)) \to (M_3)$	∇	4	τ_3
Вычисление δ_j	4	$\Pi_5 (M_1, M_7, M_9, T)$			
Вычисление	5	$(M_1, (2, x_1)) * (M_2, (2, x_1+1)) \to (L_1)$	—	—	τ_1
$\delta_k = \min\limits_j \delta_j$	6	$\Pi_{10} (L_1, 2, 4, L_2)$			
Проверка	7	$(L_1, (2, x_1)) \to (M_1)$			τ_1
$\delta_k < 0$	8	$\Pi_7 (L_2, 2, L_1)$			
	9	$(L_1, (2, x_1)) \to (L_2)$	10	∇	τ_3

I	2	3	4	5	6
	10	$П_{10}(L_2, 2, C, L_1)$			
	11	$(L_2, (2, x_1)) \rightarrow (M_1)$	—	—	τ_1
Вычисление x_{ik}	12	$\{(M_2, (2, x_1+1)), (L_1, (2, x_1))\} \rightarrow \{L_1, M_1\}$	—	—	τ_1
	13	$П_5(M_7, M_1, L_1, T)$			
Пересылка x_{ik} во 2-й слой X и в I-й слой Y	14	$П_6(M_5, M_6, 1, M_7, 2, C)$			
	15	$П_6(M_3, M_4, 2, M_7, 2, C)$			
Вычисление ℓ и θ_0	16	$П_8(M_3, 2, L_3)$			
	17	$П_7(M_3, 2, L_2)$			
Проверка ограниченности линейной формы	18	$(M_3, (2, x_1)) \rightarrow (M_3)$	19	∇	τ_3
	19	$Д(M_3, 1, 2, 3, 4)$			$t_Ф$
	20	$П_{10}(M_3, 3, 4, L_4)$			
	21	$П_{10}(L_4, 1, T, L_5)$			
	22	$(L_4, (2, x_1)) \rightarrow (M_3)$	—	—	τ_1
	23	$(L_2, (2, x_1)) \rightarrow (M_3)$	—	—	τ_1

I	2	3	4	5	6
$X[i] := X[i] -$	24	$П_3(L_5, 3)$			
$-\theta_0 \times Y[i]$	25	$П_1(M_3, 3)$			
	26	$УМ(M_3, 2, 3, 2, 4)$			$t_{УМ}$
	27	$ВЧ(M_3, 1, 2, 1, 4)$			$t_{ВЧ}$
	28	$(L_3, (2, x_1)) \to (M_3)$	—	—	τ_1
	29	$П_3(L_5, 1)$			
	30	$П_2(M_5, T, L_2)$			
	31	$\{(L_1, (2, x_1+1)), (M_1, (2, x_1))\} \to \{M_2, L_1\}$	—	—	τ_1
$NX[l] := k$	32	$П_3(L_1, 1)$			
$X[l] := \theta_0$	33	$П_1(L_5, 2)$			
	34	$П_9(L_5, C, 2, 1)$			
	35	$П_9(L_5, 4, 3, 1)$			
	36	$(L_5, (2, x_1)) \to (M_3)$	—	—	τ_1
	37	$(L_1, (2, x_1)) \to (M_1)$	—	—	τ_1

I	2	3	4	5	6
Для $i \neq \ell$	38	$П_3(L_2, 1)$			
$Y[i] := Y[i]/Y[\ell]$	39	$П_1(M_5, 2)$			
	40	$Д(M_5, 1, 2, 1, 4)$			
	41	$П_3(L_2, 1)$			
Для $i \neq \ell$	42	$П_2(M_7, C, L_1)$			
$U[i,j] := U[i,j] -$	43	$(L_1, (2, x_1)) * (M_8, (2, x_1 - 1)) \rightarrow (L_3)$			
$- U[\ell,j] \times Y[i]$	44	$П_3(L_3, 1)$			
	45	$П_2(M_7, T, L_4)$			
Для $i = \ell$	46	$П_{16}(L_4, M_7)$			
$U[\ell,j] :=$	47	$(M_8, (2, x_1)) * (M_7, (2, x_1 + 1)) \rightarrow (L_4)$	—	—	τ_1
$U[\ell,j] / Y[\ell]$	48	$П_1(L_4, 2)$			
	49	$П_9(L_4, T, 2, 1)$			
	50	$(L_4, (2, x_1)) \rightarrow (M_7)$	—	—	τ_1
	51	$(M_8, (2, x_1)) * \{(M_7, (2, x_1 + 1)), (M_7, (2, x_1 - 1))\} \rightarrow (M_8)$	53	52	τ_3

1	2	3	4	5	6
	52	$(M_8,(2,x_1)) * \{(M_7;(2,x_1+1)),(M_6,(2,x_1-1))\} \to (M_8)$	53	54	τ_3
	53	$П_{17}$			
	54	$\{(L_3,(2,x_1)),(L_1,(2,x_1+1))\} \to (L_1,L_5)$	44	55	τ_3
	55	$(L_3,(2,x_1)) \to (L_1)$	—	—	τ_1
	56	$П_3(L_2,1)$			
	57	$П_1(L_1,2)$			
	58	$Д(L_1,1,2,1,4)$			$t_Д$
	59	$(L_2,(2,x_1)) \to (M_5)$	—	—	τ_1
	60	$П_4(C,M_5,M_6,M_7,3)$			
	61	$УМ(M_7,2,3,2,4)$			$t_{УМ}$
	62	$ВЧ(M_7,1,2,1,4)$			$t_{ВЧ}$
	63	$(L_1,(2,x_1)) \to (M_7)$	—	—	τ_1

Примечания к таблице 2

I. В таблице расписана программа решения задачи ЛП только для одной итерации первого этапа. Переходы с этапа на этап и контроль конца решения задачи дабавляют к программе незначительное число подстановок и не оказывают существенного влияния на время решения задачи, поэтому эти подстановки в таблице 2 не выписаны, а метки, определяющие переход к подстановкам контроля, заменены символом ∇ .

2. Время в тактах работы алгоритма на одной итерации выражается следующей формулой:

$$T \leq 10^3\tau_1 + 10\tau_2 + 2\cdot 10^2\tau_3 + (10^2\tau_1 + 10\tau_2 + 15\tau_3)m + 4t_{ум} + 2t_{мсл} + 2t_{вч} + 2t_{д} ,$$

где $t_{ум}$ - время умножения двух чисел,

$t_{вч}$ - время вычитания двух чисел,

$t_{д}$ - время деления двух чисел,

$t_{мсл}$ - время сложения массива чисел (длина которого не более m).

Известно, что общее время решения задачи пропорционально $2m$. В данном случае в общее время будет включено время работы программы $П_{17}$, частота работы которой зависит от выбора длины буфера e .