Труды Института математики НАН Беларуси. Минск. 2002. Том 11. С. 140–149

УДК 519.6+681.3.012

# МАТРИЧНОЕ ПРЕДСТАВЛЕНИЕ БЛОКОВ ДЕКОМПОЗИРОВАННЫХ АЛГОРИТМОВ СИСТОЛИЧЕСКОГО ТИПА

А. А. Тиунчик

A matrix model for uniting decomposed dependence graphs is proposed. The model allows one to solve the following problems: to detect similar dependence graphs; to construct unlimited dependence graphs for iterative algorithms on the base of a unique graph model; to construct unlimited dependence graphs for iterative algorithms on the base of a pair of graph models; to detect dependence graphs that can be used for it; to check an existence of a temporal coordination of united graphs.

## Введение

Систолические процессоры – высокоэффективные специализированные многопроцессорные устройства, хорошо приспособленные для СБИС или FPGA реализации и использующие преимущества параллельной и конвейерной обработки информации. В настоящее время в [1–4] разработаны процедуры проектирования систолических процессоров для реализации алгоритмов, представленных как единое целое. К основным этапам проектирования относятся: представление реализуемого алгоритма в специальном виде, построение графовой модели, отображение этой модели на вычислительную структуру. Однако в ряде случаев построение графовой модели затруднительно в силу большой сложности или громоздкости реализуемого алгоритма. В этом случае целесообразно разбить исходный алгоритм на подалгоритмы и построить единую графовую модель из графов, представляющих эти подалгоритмы.

В работе [5] рассмотрено построение единой графовой модели для последовательного перемножения произвольного числа матриц. Единая модель построена на основе формализованной пространственной стыковки графовых моделей для отдельных умножений, однако задача выбора подходящих графовых моделей отдельных подалгритмов решалась на основе перебора различных возможных комбинаций. В работе [6] рассмотрен более общий способ воссоединения графовых моделей в единую структуру, однако формализованная процедура нахождения оператора пространственного совмещения была предложена только для одномерного примера. Целью этой работы является разработка матричного аппарата представления структур обрабатываемых данных, которая позволяет: находить во множестве графовых моделей для реализации одного и того же алгоритма подобные; устанавливать существование аффинных операторов пространственной стыковки и быстро получать их матрицы для построения сложных алгоритмов на основе как единственной графовой модели, так и двух различных графовых моделей; формализованно устанавливать существование единого временного планирования для вычислительного устройства, реализующего весь алгоритм.

Keywords: *parallelization, systolic algorithm, iterative algorithm, space-time coordination*
2000 Mathematics Subject Classification: 65Y05, 68M07

## 1. Основные понятия

Граф зависимостей (ГЗ) является графовой моделью алгоритма. Построение ГЗ основано на представлении исходного алгоритма однородными рекуррентными уравнениями [7]

$$x_i(v) = F_i(x_1(v-\varphi_{x_1}), x_2(v-\varphi_{x_2}), \ldots, x_p(v-\varphi_{x_p})), \qquad 1 \leqslant i \leqslant p, \ p \in \mathbb{N}, \ v \in V \subset \mathbb{Z}^m, \quad (1)$$

где $x_1, x_2, \ldots, x_p$ – переменные алгоритма, $v$ – $m$-мерный вектор (точка $m$-мерной целочисленной решетки $\mathbb{Z}^m$); $F_i$ – функция $p$ переменных $x_1, x_2, \ldots, x_p$, а $\varphi_{x_1}, \varphi_{x_2}, \ldots, \varphi_{x_p}$ – $m$-мерные векторы. Предполагается, что векторы $\varphi_{x_1}, \varphi_{x_2}, \ldots, \varphi_{x_p}$ являются локальными, т.е. компоненты векторов принадлежат множеству $\{0, 1, -1\}$. Область $V$ называется областью вычислений.

Алгоритм (1) может быть представлен ГЗ, вершины которого идентифицируются с точками $v \in V$, а дуги – с векторами $\varphi_{x_1}, \varphi_{x_2}, \ldots, \varphi_{x_p}$. Множество вершин графа совпадает с областью вычислений $V$. Множество дуг $E \subset V \times V$ характеризуется множеством векторов $\{\varphi_{x_1}, \varphi_{x_2}, \ldots, \varphi_{x_p}\}$.

Вектор $\varphi_{x_i}$, соответствующий дуге для пересылки данного $x_i$ из вершины $v_1$ в вершину $v_2$, определяется координатами $\varphi_{x_i} = (v_2 - v_1)$. Если $(v - \varphi_{x_i}) \in V$, то вектор $\varphi_{x_i}$ называется вектором зависимости. Если $(v - \varphi_{x_i}) \notin V$, то точка $v$ называется точкой ввода; если $(v + \varphi_{x_i}) \notin V$, то точка $v$ называется точкой вывода. Предполагается, что длина и направление соответствующего вектора ввода или вывода $\varphi_{x_i}$ не определены.

Предполагается, что ГЗ является строго направленным, т.е. существует такой целочисленный вектор $\tau$, $\tau \in \mathbb{Z}^m$, который образует острые углы со всеми векторами зависимостей $\varphi_{x_i}$:

$$\langle \tau, \varphi_{x_i} \rangle > 0, \quad 1 \leqslant i \leqslant p, \quad (2)$$

где $\langle \cdot, \cdot \rangle$ обозначает скалярное произведение. Вектор $\tau$ называется направляющим вектором ГЗ.

Пространственное отображение ГЗ на вычислительную структуру определяется обычно линейным оператором $\Pi : \mathbb{Z}^m \to \mathbb{Z}^r$ ($r = 0$, 1 или 2) [1–6]. Использование линейных операторов специального вида автоматически сохраняет исходную локальность векторов зависимости, что обеспечивает получение массивов с локальными межсоединениями. Таким образом, проектирование систолических процессоров (не имеющих глобальных межсоединений) предпочтительно осуществлять на основе ГЗ, представляющих собой отдельные подалгоритмы и имеющих только локальные дуги как внутри отдельных блоков, так и между этими блоками.

Необходимо отметить, что если ГЗ не ограничен или его размеры в направлениях, задаваемых некоторыми векторами $\xi_1, \xi_2, \ldots$, неизвестны заранее (и могут даже определяться параметрами, генерируемыми в ходе реализации вычислительного процесса), то необходимо, чтобы подпространство, порождаемое векторами $\xi_1, \xi_2, \ldots$, входило в ядро оператора $\Pi$. Такое отображение позволяет проектировать конечные систолические процессоры для реализации произвольного числа итерационных шагов алгоритма.

## 2. Матричное представление

Пусть заданный алгоритм состоит из отдельных информационно связанных подалгоритмов вида (1). Каждый подалгоритм представлен ГЗ, причем выходные промежуточные результаты одного ГЗ должны использоваться как начальные входные данные для последующих вычислений. Пространственное размещение вершин вывода может существенно отличаться от пространственного размещения вершин ввода. В силу этого возникает задача пространственного соединения и согласования отдельных блоков.

Типичный случай передачи данных – передача векторов, матриц и других упорядоченных структур. Естественно предположить, что и элементы структур, и соответствующие вершины ввода-вывода упорядочены в пространстве и образуют некоторые регулярные структуры.

Пусть области ввода-вывода являются $(m-1)$-мерными прямоугольными параллелепипедами. Пусть входные (выходные) вершины для элементов $s_{i_1,i_2,\ldots,i_{m-1}}$ структуры $S$ размещены в точках $(\nu_{p_0},\nu_{p_1},\nu_{p_2},\ldots,\nu_{p_{m-1}})$, где $\{p_0,p_1,p_2,\ldots,p_{m-1}\}$ – некоторая перестановка чисел $\{0,1,2,\ldots,m-1\}$, $\nu_0=c_0$, $\nu_k=c_k\pm i_k$, $1\leqslant k\leqslant m-1$, а $c_0,c_1,c_2,\ldots,c_{m-1}$ – некоторые константы. Упорядоченные таким образом области ввода и вывода будем называть регулярно структурированными областями ввода ($I$-областями) и регулярно структурированными областями вывода ($O$-областями) соответственно. Предполагается, что эти области используются для передачи одних и тех же наборов данных; следовательно, их размеры должны быть равными.

Описание областей ввода и вывода должно устанавливать соответствие между элементами ввода или вывода и точками пространства, где и происходит ввод или вывод этих элементов. Как правило, описание областей ввода и вывода дается в очень сложной форме. Это затрудняет разработку формализованных средств пространственного преобразования этих областей. С другой стороны, число различных вариантов пространственного размещения $I$-областей и $O$-областей экспоненциально возрастает с ростом размерности пространства $\mathbb{Z}^m$, что существенно затрудняет решение задачи стыковки методом перебора.

Для решения задачи формализованного пространственного соединения ГЗ будем использовать следующее матричное представление областей ввода-вывода. Пусть вектор $d_k\in\mathbb{Z}^m$, $1\leqslant k\leqslant m-1$, имеет компоненты $(0,0,\ldots,0,\pm1,0,\ldots,0)$, где знак ненулевой $\nu_{p_k}$-й компоненты вектора $d_k$ совпадает со знаком компоненты $\nu_k=c_k\pm i_k$, а вектор $d_0\in\mathbb{Z}^m$ имеет компоненты $(0,0,\ldots,0,\pm1,0,\ldots,0)$, где знак ненулевой $\nu_{p_0}$-й компоненты должен быть определен следующим образом: $d_0$ направлен внутрь области вычислений в случае области входа; в противном случае $d_0$ направлен наружу. Чтобы описать $I$- или $O$-область, сформируем матрицу $D=(d_0^{\mathrm{T}},d_1^{\mathrm{T}},\ldots,d_{m-1}^{\mathrm{T}})$, которая сгенерирована вектор-столбцами $d_0^{\mathrm{T}},d_1^{\mathrm{T}},\ldots,d_{m-1}^{\mathrm{T}}$. Таким образом, любая $I$- или $O$-область однозначно определена матрицей $D$. Очевидно, все векторы $d_0^{\mathrm{T}},d_1^{\mathrm{T}},\ldots,d_{m-1}^{\mathrm{T}}$ отличны от нуля и ортогональны. Следовательно, матрица $D$ невырождена, и существует обратная матрица $D^{-1}$ такая, что $DD^{-1}=D^{-1}D=E$. Более того, так как векторы $d_0^{\mathrm{T}},d_1^{\mathrm{T}},\ldots,d_{m-1}^{\mathrm{T}}$ единичны, то столбцы матрицы $D$ образуют ортонормированную систему, а сама матрица $D$ является ортогональной и, следовательно, $D^{-1}=D^{\mathrm{T}}$.

Через $D_I$ и $D_O$ будем обозначать матрицы, определяющие $I$- и $O$-области соответственно. Через $\Phi$ обозначим матрицу, сформированную вектор-столбцами $\varphi_{x_1}^{\mathrm{T}},\varphi_{x_2}^{\mathrm{T}},\ldots,\varphi_{x_p}^{\mathrm{T}}$. Отметим, что матрица $\Phi$ может не быть квадратной.

Определение. Совокупность матриц $D_I,D_O$ и $\Phi$ ГЗ $G$ будем называть матричным представлением ГЗ $G$ и обозначать $M(D_I,D_O,\Phi)$.

## 3. Нахождение подобных графов зависимостей

Определение. ГЗ с матричными представлениями $M(D_{I,1},D_{O,1},\Phi_1)$ и $M(D_{I,2},D_{O,2},\Phi_2)$ будем называть подобными, если существует такой линейный оператор $\mathbf{F}$, что $\mathbf{F}D_{I,1}=D_{I,2}$, $\mathbf{F}D_{O,1}=D_{O,2}$, $\mathbf{F}\Phi_1=\Phi_2$.

Подобные ГЗ появляются в силу многообразия возможных вариантов подачи и пересылки начальных данных по ГЗ. Являясь, по существу (с точностью до линейного преобразования), одним и тем же ГЗ, подобные ГЗ для реализации одного и того же алгоритма усложняют задачу нахождения стыкуемых подграфов, увеличивая число возможных комбинаций. Таким образом, предварительное нахождение и устранение из рассмотрения подобных ГЗ может существенно упростить решение задачи нахождения стыкуемых подграфов.

Отметим, что в силу невырожденности матриц $D_{I,1}$, $D_{O,1}$, $D_{I,2}$ и $D_{O,2}$ матрица оператора $\mathbf{F}$ может быть однозначна найдена по этим матрицам, однако это же нельзя утверждать относительно $\Phi_1$ и $\Phi_2$ : в зависимости от их размерности могут быть как ситуации, когда оператор $\mathbf{F}$ не существует, так и ситуации, когда он определяется не единственным образом. Из сказанного выше вытекает истинность следующей леммы.

Лемма 1. *В случае невырожденности матриц* $\Phi_1$ *и* $\Phi_2$ *для подобия ГЗ* $M(D_{I,1}, D_{O,1}, \Phi_1)$ *и* $M(D_{I,2}, D_{O,2}, \Phi_2)$ *необходимо и достаточно, чтобы* $D_{I,2}D_{I,1}^{-1} = D_{O,2}D_{O,1}^{-1} = \Phi_2\Phi_1^{-1}$.

Отметим, что если получено $n$ исходных ГЗ, то для нахождения всех подобных ГЗ в этом множестве требуется $S_n = \dfrac{n(n-1)}{2}$ сравнений. Однако процедуру перебора можно существенно упростить, если использовать следующие необходимые условия подобия.

Лемма 2. *Для подобия ГЗ* $M(D_{I,1}, D_{O,1}, \Phi_1)$ *и* $M(D_{I,2}, D_{O,2}, \Phi_2)$ *необходимо, чтобы либо* $D_{I,1} = D_{I,2}$, $D_{O,1} = D_{O,2}$, $\Phi_1 = \Phi_2$, *либо* $D_{I,1} \neq D_{I,2}$, $D_{O,1} \neq D_{O,2}$ *и в случае невырожденности матриц* $\Phi_1$ *и* $\Phi_2$ $\Phi_1 \neq \Phi_2$.

Лемма 3. *Для подобия ГЗ* $M(D_{I,1}, D_{O,1}, \Phi_1)$ *и* $M(D_{I,2}, D_{O,2}, \Phi_2)$ *необходимо, чтобы* $\dfrac{\det D_{I,2}}{\det D_{I,1}} = \dfrac{\det D_{O,2}}{\det D_{O,1}}$, *а в случае невырожденности матриц* $\Phi_1$ *и* $\Phi_2$ $\dfrac{\det D_{I,2}}{\det D_{I,1}} = \dfrac{\det D_{O,2}}{\det D_{O,1}} = \dfrac{\det \Phi_2}{\det \Phi_1}$.

Истинность этих лемм следует из единственности оператора $\mathbf{F}$.

Процедура применения леммы 2 является вычислительно легкой, поскольку не требует вычислений, а сводится только к сравнению трех пар матриц. С другой стороны, процедуру применения леммы 2 можно дополнительно ускорить, если предварительно вычислить определители всех матриц (эти определители легко вычислимы в силу специальной структуры матриц, кроме того, они понадобятся также и при реализации процедуры леммы 3), так как из неравенства определителей следует и неравенство самих сравниваемых матриц (поэлементному сравнению подлежат только матрицы с одинаковыми определителями).

## 4. Пространственная стыковка отдельных ГЗ

ГЗ можно рассматривать как геометрический объект, на который можно действовать линейными или аффинными операторами. Два ГЗ будем называть пространственно состыкованными, если соединение любой пары соответствующих друг другу вершин ввода и вывода определяется одним и тем же вектором с координатами из множества $\{0, 1, -1\}$. Стыковка обеспечивает систолический способ воссоздания декомпозированного алгоритма. Нашей целью является разработка средств пространственной стыковки за счет изменения взаимного расположения вершин ввода и вывода.

Пусть $\mathbf{A}$ – аффинный оператор, $\mathbf{A} : \mathbb{Z}^m \to \mathbb{Z}^m$, $\mathbf{A}x = \mathbf{L}x + \mathbf{l}$, где $\mathbf{L}$ – линейный оператор, $\mathbf{l}$ – вектор переноса, $x \in \mathbb{Z}^m$. Линейные операторы и соответствующие им матрицы различать в обозначениях не будем. Граф $G$ может быть преобразован оператором $\mathbf{A}$ в граф $\mathbf{A}G$ : преобразование может включать вращение и отражение $\mathbf{L}$ и перенос $\mathbf{l}$. Знаком "$*$" в матричном представлении графа зависимостей будем обозначать произвольную матрицу ($D_I$, $D_O$ или $\Phi$), вид которой не имеет значения для дальнейшего изложения. Пусть $G_1$ и $G_2$ – ГЗ с $O$- и $I$-областями и матричными представлениями $M(*, D_O, *)$ и $M(D_I, *, *)$ соответственно. Информация из $G_1$ должна перекачиваться к пространственно присоединенному с ним графу $\mathbf{A}G_2$. Очевидно, если существует линейный оператор $\mathbf{L}$ такой, что $O$-область $G_1$ совпадает с $I$-областью $\mathbf{L}G_2$, т.е. $\mathbf{L}D_I = D_O$, то всегда можно найти соответствующий вектор $\mathbf{l}$, чтобы обеспечить пространственное соединение.

Теорема 1. *Любой ГЗ с матричным представлением* $M(*, D_O, *)$ *может быть пространственно соединен с любым ГЗ с матричным представлением* $M(D_I, *, *)$. *Такое соединение может быть выполнено единственным способом.*

Доказательство. Пусть линейное преобразование $\mathbf{L}$ обеспечивает пространственную стыковку $I$-области, описываемой матрицей $D_I$, к $O$-области, описываемой матрицей $D_O$, т.е. $\mathbf{L}D_I = D_O$. Матрица $D_O$ обратима, следовательно, $\mathbf{L}$ определяется единственным образом как

$$\mathbf{L} = D_O D_I^{-1}. \tag{3}$$

Итак, преобразование $\mathbf{A}$ может быть определено для любых $I$- и $O$-области.

Матрицы $D_O$ и $D_I$ порождаются линейно независимыми единичными векторами, следовательно, произведение (3) определяет некоторое вращение гиперкуба на угол $\pi/2$ или $\pi$ либо отражение относительно гиперплоскости, проходящей через одну из его граней. Оператор $\mathbf{L}$ должен сохранять внутреннюю структуру исходного ГЗ, т.е. сохранять локальность векторов зависимости и размещение областей ввода и вывода на границе области вычислений. Выполнение этих требований следует из геометрического смысла вращений и отражений. Теорема доказана.

Отметим, что изменение знака вектора $d_0$ в $D_O$ или $D_I$ на противоположный соответствует получению внутренней стыковки [8]: пересылаемые данные "отражаются" от границы ГЗ и возвращаются в исходную область вычислений.

Рассмотрим случай, когда исходный алгоритм разбит на множество идентичных подалгоритмов и требуется осуществить пространственную стыковку большого количества идентичных ГЗ. Существуют два подхода к решению задачи стыковки множества идентичных ГЗ: 1) построить ряд различных ГЗ для реализации одного и того же подалгоритма и попытаться выбрать из них такие, которые могут быть состыкованы естественным образом; 2) построить один ГЗ и попытаться найти алгебраические средства для таких преобразований этого ГЗ, которые удовлетворяли бы всем требованиям стыковки. Первый подход подробно исследован в [5]. Основным его недостатком в общем случае является высокая трудоемкость процедур построения и исследования различных графовых реализаций подалгоритмов. Рассмотрим возможности второго подхода.

Пусть $G_1, G_2, \ldots, G_Q$ – идентичные ГЗ, которые представляют $Q$ подалгоритмов. Пусть $\mathbf{A}_n$ – аффинный оператор, $\mathbf{A}_n : \mathbb{Z}^m \to \mathbb{Z}^m$, $\mathbf{A}_n x = \mathbf{L}_n x + \mathbf{l}_n$, где $\mathbf{L}_n$ – линейный оператор, $\mathbf{l}_n$ – вектор переноса, $x \in \mathbb{Z}^m$. Тогда граф $G_n$ может быть преобразован операторами $\mathbf{A}_n$ в граф $\mathbf{A}_n G_n$ (различные графы $G_n$ могут быть преобразованы различными операторами $\mathbf{A}_n$).

Линейные операторы $\mathbf{L}_n$ могут изменять пространственное размещение векторов зависимостей, что приводит к появлению большого числа типов вершин. В связи с этим обретает актуальность проектирование регулярной однородной цепи преобразованных ГЗ. Для построения такой цепи нежелательно использование большого числа различных операторов. Рассмотрим случай применения единственного оператора $\mathbf{A}$.

Построение цепи пространственно связанных ГЗ с использованием единственного оператора $\mathbf{A}$ осуществляется следующим образом. Первый ГЗ $G_1$ остается неподвижным, второй ГЗ $G_2$ должен быть присоединен посредством оператора $\mathbf{A}$. Предполагается, что третий ГЗ $G_3$ мог быть заранее связан с $G_2$ оператором $\mathbf{A}$; следовательно, присоединение $G_1$ к $G_2$ означает, что на $G_3$ действует оператор $\mathbf{A}^2$. Таким образом можно продолжать процедуру и получить цепь пространственно связанных ГЗ $G_1$, $\mathbf{A}G_2$, $\mathbf{A}^2G_3$, $\mathbf{A}^3G_4, \ldots, \mathbf{A}^{Q-1}G_Q$, где $n$-я степень оператора $\mathbf{A}$ рассматривается как $n$ последовательных преобразований оператором $\mathbf{A}$:

$$\mathbf{A}^n x = (\mathbf{L}+\mathbf{l})^n x = \underbrace{\mathbf{L}(\mathbf{L}(\mathbf{L}\ldots\mathbf{L}(\mathbf{L}}_{n} x + \underbrace{\mathbf{l}) + \mathbf{l} + \ldots + \mathbf{l}) + \mathbf{l}) + \mathbf{l}}_{n} =$$

$$= \mathbf{L}^n x + \mathbf{L}^{n-1}\mathbf{l} + \mathbf{L}^{n-2}\mathbf{l} + \mathbf{L}^{n-3}\mathbf{l} + \ldots + \mathbf{L}^3\mathbf{l} + \mathbf{L}^2\mathbf{l} + \mathbf{L}\mathbf{l} + \mathbf{l}.$$

Таким образом, аффинное преобразование оператором $\mathbf{A}^n$ сводится к линейному преобразованию $\mathbf{L}^n$ и параллельному переносу, задаваемому вектором $\mathbf{L}^{n-1}\mathbf{l} + \mathbf{L}^{n-2}\mathbf{l} + \ldots + \mathbf{L}\mathbf{l} + \mathbf{l}$. Чтобы ограничить число различных операторов $\mathbf{L}$, $\mathbf{L}^2$, $\mathbf{L}^3, \ldots, \mathbf{L}^{Q-1}$, будем предполагать, что существует число $\lambda$ такое, что

$$\mathbf{L}^\lambda = \mathbf{E}, \quad \lambda \geqslant 1, \tag{4}$$

и $\lambda$ мало (обычно $\lambda = 1$ или $2$). Цепь ГЗ $G_1$, $\mathbf{A}G_2$, $\mathbf{A}^2G_3$, $\mathbf{A}^3G_4, \ldots$ при $\mathbf{L}^\lambda = \mathbf{E}$ будем называть $\mathbf{L}^\lambda$-цепью.

Рассмотрим следующие задачи: 1) возможно ли объединить декомпозированный алгоритм (создать $\mathbf{L}^\lambda$-цепь) на основе единственного ГЗ с матричным представлением $M(D_I, D_O, \Phi)$, если $\lambda \leqslant 2$, и 2) если это так, то какими свойствами должны обладать матрицы, входящие в это матричное представление?

Из теоремы 1 следует, что существует линейный оператор $\mathbf{L}_{OI}$, обеспечивающий пространственную стыковку области ввода данных следующего ГЗ с областью вывода результатов исходного ГЗ, и линейный оператор $\mathbf{L}_{IO}$, обеспечивающий пространственную стыковку области вывода предыдущего ГЗ с областью ввода исходного ГЗ. Так как все ГЗ идентичны, то все возможности пространственной стыковки могут быть установлены на основании исследования $I$- и $O$-областей одного ГЗ.

Теорема 2. *$\mathbf{L}^\lambda$-цепь при $\lambda \leqslant 2$ может быть построена тогда и только тогда, когда $D_O(D_I)^{-1} = D_I(D_O)^{-1}$.*

Доказательство. Пусть $D_I$ и $D_O$ – матрицы описания $I$- и $O$-области соответственно. Тогда $\mathbf{L}_{IO}D_I = D_O$ и $\mathbf{L}_{OI}D_O = D_I$,

$$\mathbf{L}_{OI} = D_I(D_O)^{-1}, \tag{5}$$

$$\mathbf{L}_{IO} = D_O(D_I)^{-1}. \tag{6}$$

Для построения $\mathbf{L}^\lambda$-цепи должен использоваться один оператор. Таким образом, $\mathbf{L}_{IO}$ должен быть равен $\mathbf{L}_{OI}$, а $D_O(D_I)^{-1} = D_I(D_O)^{-1}$. С другой стороны, если $D_O(D_I)^{-1} = D_I(D_O)^{-1}$, то $\mathbf{L}_{IO} = \mathbf{L}_{OI}$. Теорема доказана.

Теорема 3. *Любая составленная из $n$ ГЗ $\mathbf{L}^\lambda$-цепь при $\lambda \leqslant 2$ может быть заключена в цилиндр, радиус которого не зависит от $n$.*

Доказательство. Для доказательства этой теоремы достаточно показать, что любая $\mathbf{L}^\lambda$-цепь при $\lambda \leqslant 2$ распространяется в направлении некоторого фиксированного вектора $\xi$.

Действительно, если $\lambda = 1$, то $\mathbf{L} = \mathbf{E}$, и вся цепь может быть построена путем параллельных переносов на вектор $\mathbf{l}$. Вектор $\xi = \mathbf{l}$ должен входить в ядро оператора отображения всей цепи, при этом будет получен конечный систолический процессор для реализации произвольного числа шагов.

Пусть $\lambda = 2$. Так как $\mathbf{L}^2 = \mathbf{E}$, то для четного $n$

$$\mathbf{A}^n x = \mathbf{E}x + \underbrace{\mathbf{L}\mathbf{l}+\mathbf{l}+\ldots+\mathbf{L}\mathbf{l}+\mathbf{l}+\mathbf{L}\mathbf{l}+\mathbf{l}}_{n} = \mathbf{E}x + \frac{n}{2}(\mathbf{L}\mathbf{l}+\mathbf{l}),$$

а для нечетного $n$

$$\mathbf{A}^n x = \mathbf{L}x + \underbrace{\mathbf{l}+\mathbf{L}\mathbf{l}+\ldots+\mathbf{L}\mathbf{l}+\mathbf{l}}_{n} = \mathbf{L}x + \frac{n-1}{2}(\mathbf{L}\mathbf{l}+\mathbf{l}) + \mathbf{l}.$$

Таким образом, $\mathbf{L}^\lambda$-цепь распространяется в направлении вектора $\mathbf{L}\mathbf{l}+\mathbf{l}$; вектор $\xi = \mathbf{L}\mathbf{l}+\mathbf{l}$ должен входить в ядро оператора отображения. Теорема доказана.

При построении $\mathbf{L}^2$-цепи при $\lambda = 2$ используются не только исходные ГЗ, но и ГЗ, преобразованные линейным оператором $\mathbf{L}$. В преобразованных ГЗ изменяется пространственное размещение векторов зависимости, что приводит к появлению новых типов вершин и, формально, новых ГЗ. Это позволяет рассмотреть еще один способ построения цепи состыкованных ГЗ, основанный на использовании двух различных исходных ГЗ.

Пусть даны ГЗ с матричным представлением $M(D_{I_1}, D_{O_1}, \Phi_1)$ и ГЗ с матричным представлением $M(D_{I_2}, D_{O_2}, \Phi_2)$. Пусть пространственная стыковка $I$-области второго ГЗ к $O$-области первого осуществляется оператором $\mathbf{L}_{O_1I_2}$, а $I$-области первого ГЗ к $O$-области второго – оператором $\mathbf{L}_{O_2I_1}$. Так как последовательное применение этих операторов к последующим ГЗ эквивалентно применению произведения операторов, то во избежание появления большого числа типов новых ГЗ естественно потребовать, чтобы выполнялось равенство $\mathbf{L}_{O_1I_2}\mathbf{L}_{O_2I_1} = \mathbf{E}$.

Через $\mathbf{L}^2_*$-цепь обозначим цепь, построенную из ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$ на основе применения двух аффинных операторов $\mathbf{A}_{O_2I_1} : \mathbb{Z}^m \to \mathbb{Z}^m$, $\mathbf{A}_{O_2I_1}x = \mathbf{L}_{O_2I_1}x + \mathbf{l}_{O_2I_1}$ и $\mathbf{A}_{O_1I_2} : \mathbb{Z}^m \to \mathbb{Z}^m$, $\mathbf{A}_{O_1I_2}x = \mathbf{L}_{O_1I_2}x + \mathbf{l}_{O_1I_2}$, где $\mathbf{L}_{O_2I_1}$ и $\mathbf{L}_{O_1I_2}$ – линейные операторы, $\mathbf{l}_{O_2I_1}$ и $\mathbf{l}_{O_1I_2}$ – векторы переноса, $x \in \mathbb{Z}^m$;

$\mathbf{L}_{O_2I_1}\mathbf{L}_{O_1I_2} = \mathbf{L}_{O_1I_2}\mathbf{L}_{O_2I_1} = \mathbf{E}$. Другими словами, $\mathbf{L}^2_*$-цепь строится параллельным переносом пар построенных состыкованных ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$. Такая процедура построения не противоречит требованию малого числа различных типов ГЗ (это число по-прежнему равно двум).

Теорема 4. *$\mathbf{L}^2_*$-цепь из ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$ может быть построена тогда и только тогда, когда*

$$D_{O_2}(D_{I_1})^{-1}D_{O_1}(D_{I_2})^{-1} = \mathbf{E}. \tag{7}$$

Доказательство. Из теоремы 1 следует, что существуют линейные операторы $\mathbf{L}_{O_2I_1}$ и $\mathbf{L}_{O_1I_2}$, осуществляющие пространственную стыковку ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$:

$$\mathbf{L}_{O_2I_1} = D_{O_2}(D_{I_1})^{-1}, \tag{8}$$

$$\mathbf{L}_{O_1I_2} = D_{O_1}(D_{I_2})^{-1}. \tag{9}$$

Так как для построения $\mathbf{L}^2_*$-цепи требуется, чтобы $\mathbf{L}_{O_2I_1}\mathbf{L}_{O_1I_2} = \mathbf{L}_{O_1I_2}\mathbf{L}_{O_2I_1} = \mathbf{E}$, то с учетом равенств (8) и (9) получаем соотношение (7).

С другой стороны, из соотношения (7) следует существование операторов $\mathbf{L}_{O_2I_1}$ и $\mathbf{L}_{O_1I_2}$, отвечающих требованию $\mathbf{L}_{O_2I_1}\mathbf{L}_{O_1I_2} = \mathbf{L}_{O_1I_2}\mathbf{L}_{O_2I_1} = \mathbf{E}$. Теорема доказана.

Следствие. *Для существования $\mathbf{L}^2_*$-цепи из ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$ необходимо, чтобы* $\dfrac{\det D_{O_2}}{\det D_{I_1}}\dfrac{\det D_{O_1}}{\det D_{I_2}} = 1.$

В заключение отметим, что при исследовании возможностей пространственной стыковки на основе анализа матричных представлений ГЗ матрица $\Phi$ не используется.

## 5. Временно́е согласование объединенного ГЗ

Временно́й режим работы проектируемого систолического процессора определяется функцией $t(v) = \langle \tau, v\rangle - \min\limits_{v\in V}\langle \tau, v\rangle + t_0$, где $\tau$ – направляющий вектор ГЗ, $t_0$ – целочисленная неотрицательная константа. Нахождение непротиворечивого временно́го режима работы систолического устройства сводится к нахождению такого целочисленного вектора $\tau$, для которого

$$\langle \tau, \varphi_{x_i}\rangle > 0, \tag{10}$$

для любого вектора зависимости $\varphi_{x_i}$, $1 \leqslant i \leqslant p$. Очевидно, что существование вектора $\tau$, удовлетворяющего условию (10), равносильно существованию целочисленного решения $\tau$ системы линейных неравенств $\mathbf{\Phi}\tau > \mathbf{0}$.

Теорема 5. *Единая таймирующая функция для $\mathbf{L}^2$-цепи при $\lambda \leqslant 2$ существует тогда и только тогда, когда существует целочисленное решение $\tau$ системы линейных неравенств $\mathbf{B}\tau > \mathbf{0}$, где* $\mathbf{B} = \begin{pmatrix} (\Phi)^{\mathrm{T}} \\ (D_I D_O^{-1}\Phi)^{\mathrm{T}} \end{pmatrix}$.

Истинность теоремы 5 следует из того, что направляющий вектор всего множества пространственно состыкованных ГЗ должен являться целочисленным решением систем неравенств $\mathbf{\Phi}\tau > \mathbf{0}$, $\mathbf{L\Phi}\tau > \mathbf{0}$, $\mathbf{L^2\Phi}\tau > \mathbf{0}$, $\mathbf{L^3\Phi}\tau > \mathbf{0}\ldots$, причем $\mathbf{L}^2 = \mathbf{E}$, $\mathbf{L} = D_I D_O^{-1}$.

Аналогично устанавливается истинность и следующей ниже теоремы о существовании таймирующей функции для $\mathbf{L}^2_*$-цепи.

Теорема 6. *Единая таймирующая функция для $\mathbf{L}^2_*$-цепи из ГЗ с матричными представлениями $M(D_{I_1}, D_{O_1}, \Phi_1)$ и $M(D_{I_2}, D_{O_2}, \Phi_2)$ существует тогда и только тогда, когда существует целочисленное решение $\tau$ системы линейных неравенств $\mathbf{B}\tau > \mathbf{0}$, где* $\mathbf{B} = \begin{pmatrix} (\Phi_1)^{\mathrm{T}} \\ (D_{O_1}D_{I_2}^{-1}\Phi_2)^{\mathrm{T}} \end{pmatrix}$.

## 6. Пример

Рассмотрим построение систолического алгоритма для перемножения последовательности $N \times N$-матриц

$$Y = A_l \cdots A_3 \cdot A_2 \cdot A_1. \tag{11}$$

Эта процедура естественным образом представляется как последовательность умножений матрицы на матрицу: $X_1 = A_2 \cdot A_1$, $X_2 = A_3 \cdot X_1$, $X_3 = A_4 \cdot X_2, \ldots$, $X_{l-1} = A_l \cdot X_{l-2}$. Через

$$Y = A \cdot X \tag{12}$$

обозначим некоторое промежуточное умножение матрицы на матрицу. Воспользуемся одной из систолических реализаций [5] операции (12).

Пусть даны матрицы $A = (a_{ip})$ и $X = (x_{pj})$, $1 \leqslant i, j, p \leqslant N$. Элементы матрицы $Y = (y_{ij})$, $1 \leqslant i, j \leqslant N$, формируются в соответствии с соотношениями $y_{ij} = \sum_{p=1}^{N} a_{ip} x_{pj}$. Реализация такого вычисления может быть представлена рекуррентными уравнениями

$$\begin{gathered} y(i,j,k) = a(i,j-1,k)x(i-1,j,k) + y(i,j,k-1), \\ a(i,j,k) = a(i,j-1,k), \quad x(i,j,k) = x(i-1,j,k), \quad 1 \leqslant i,j,k \leqslant N, \end{gathered} \tag{13}$$

где ввод данных описывается присвоениями $x(0,j,k) = x_{kj}$, $a(i,0,k) = a_{ik}$, $y(i,j,0) = 0$, вывод осуществляется как $y_{ij} = y(i,j,N)$. Рекуррентные уравнения (13) позволяют построить ГЗ реализации (12): вершины ГЗ размещены в кубической области вычислений $\{(i,j,k) : 1 \leqslant i,j,k \leqslant N\}$. Векторами зависимости являются векторы $\varphi_x = (1,\ 0,\ 0)$, $\varphi_a = (0,\ 1,\ 0)$ и $\varphi_y = (0,\ 0,\ 1)$. Таким образом, $\Phi = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$. Направляющим вектором этого ГЗ является вектор $\tau = (\tau_1,\ \tau_2,\ \tau_3)$, где $\tau_1 \geqslant 1$, $\tau_2 \geqslant 1$, $\tau_3 \geqslant 1$.

Дадим матричное описание областей ввода и вывода. Векторами $d_0$, $d_1$ и $d_2$ области ввода являются векторы $d_0^I = (1,\ 0,\ 0)$, $d_1^I = (0,\ 0,\ 1)$ и $d_2^I = (0,\ 1,\ 0)$ соответственно. Аналогично $d_0^O = (0,\ 0,\ 1)$, $d_1^O = (1,\ 0,\ 0)$ и $d_2^O = (0,\ 1,\ 0)$. Таким образом, $D_I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, $D_O = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$.

В работе [5] получено также и множество других рекуррентных уравнений для реализации итерационного шага (12). Приведем матричное описание ГЗ, представляющих эти алгоритмы. ГЗ, представляющий $i$-й алгоритм, будем обозначать ГЗ$_i$, $1 \leqslant i \leqslant 16$. Соответствующие ему матрицы будем обозначать $D_{I,i}$, $D_{O,i}$ и $\Phi_i$. Эти матрицы имеют следующий вид: $D_{I,1} = D_{I,2} = D_{I,9} = D_{I,10} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, $D_{I,3} = D_{I,4} = D_{I,11} = D_{I,12} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$, $D_{I,5} = D_{I,6} = D_{I,13} = D_{I,14} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$, $D_{I,7} = D_{I,8} = D_{I,15} = D_{I,16} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}$, $D_{O,1} = D_{O,2} = D_{O,3} = D_{O,4} = D_{O,5} = D_{O,6} = D_{O,7} = D_{O,8} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$, $D_{O,9} = D_{O,10} = D_{O,11} = D_{O,12} = D_{O,13} = D_{O,14} = D_{O,15} = D_{O,16} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & 0 & 0 \end{pmatrix}$, $\Phi_1 = \Phi_5 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$,

$\Phi_2 = \Phi_6 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\Phi_3 = \Phi_7 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\Phi_4 = \Phi_8 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $\Phi_9 = \Phi_{13} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$, $\Phi_{10} = \Phi_{14} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$, $\Phi_{11} = \Phi_{15} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$, $\Phi_{12} = \Phi_{16} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \end{pmatrix}$. Определители этих матриц легко вычислимы: $\det D_{I,1} = \det D_{I,2} = \det D_{I,7} = \det D_{I,8} = \det D_{I,9} = \det D_{I,10} = \det D_{I,15} = \det D_{I,16} = \det D_{O,9} = \det D_{O,10} = \det D_{O,11} = \det D_{O,12} = \det D_{O,13} = \det D_{O,14} = \det D_{O,15} = \det D_{O,16} = \det \Phi_2 = \det \Phi_3 = \det \Phi_6 = \det \Phi_7 = \det \Phi_9 = \det \Phi_{12} = \det \Phi_{13} = \det \Phi_{16} = -1$, $\det D_{I,3} = \det D_{I,4} = \det D_{I,5} = \det D_{I,6} = \det D_{I,11} = \det D_{I,12} = \det D_{I,13} = \det D_{I,14} = \det D_{O,1} = \det D_{O,2} = \det D_{O,3} = \det D_{O,4} = \det D_{O,5} = \det D_{O,6} = \det D_{O,7} = \det D_{O,8} = \det \Phi_1 = \det \Phi_4 = \det \Phi_5 = \det \Phi_8 = \det \Phi_{10} = \det \Phi_{11} = \det \Phi_{14} = \det \Phi_{15} = 1$.

Нахождение подобных ГЗ в данном случае требует $S_{16} = 120$ операций преобразования и сравнения трех пар матриц. Рассмотрим применение процедуры быстрого нахождения подобных ГЗ на примере нахождения ГЗ, подобных графу зависимостей $\text{ГЗ}_1$. В силу леммы 2 подобными ему могут быть только $\text{ГЗ}_{11}$, $\text{ГЗ}_{12}$, $\text{ГЗ}_{13}$, $\text{ГЗ}_{14}$, $\text{ГЗ}_{15}$, $\text{ГЗ}_{16}$. В силу леммы 3 подобными $\text{ГЗ}_1$ из этого множества могут быть только $\text{ГЗ}_{12}$ и $\text{ГЗ}_{13}$. Непосредственной проверкой выполнения условий леммы 1 убеждаемся в том, что только $\text{ГЗ}_{13}$ подобен $\text{ГЗ}_1$. Аналогичным путем устанавливаем, что парами подобных ГЗ являются $\text{ГЗ}_1$ и $\text{ГЗ}_{13}$, $\text{ГЗ}_2$ и $\text{ГЗ}_{14}$, $\text{ГЗ}_3$ и $\text{ГЗ}_{15}$, $\text{ГЗ}_4$ и $\text{ГЗ}_{16}$, $\text{ГЗ}_5$ и $\text{ГЗ}_9$, $\text{ГЗ}_6$ и $\text{ГЗ}_{10}$, $\text{ГЗ}_7$ и $\text{ГЗ}_{11}$, $\text{ГЗ}_8$ и $\text{ГЗ}_{12}$. Процесс нахождения пар подобных ГЗ отражен в таблице. Так как в правой колонке таблицы встречаются все $\text{ГЗ}_i$, $9 \leqslant i \leqslant 16$, то дальнейшая проверка смысла не имеет.

Нахождение подобных ГЗ на основании лемм 1–3.

| Номера проверяемых ГЗ | Номера ГЗ, удовлетворяющих условию леммы 2 | Номера ГЗ, удовлетворяющих условиям лемм 2 и 3 | Номера подобных ГЗ (ГЗ, удовлетворяющих условию леммы 1) |
|---|---|---|---|
| 1 | 11, 12, 13, 14, 15, 16 | 12, 13 | 13 |
| 2 | 11, 12, 13, 14, 15, 16 | 11, 14 | 14 |
| 3 | 9, 10, 13, 14, 15, 16 | 10, 15 | 15 |
| 4 | 9, 10, 13, 14, 15, 16 | 9, 16 | 16 |
| 5 | 9, 10, 11, 12, 15, 16 | 9, 16 | 9 |
| 6 | 9, 10, 11, 12, 15, 16 | 10, 15 | 10 |
| 7 | 9, 10, 11, 12, 13, 14 | 11, 14 | 11 |
| 8 | 9, 10, 11, 12, 13, 14 | 12, 13 | 12 |

При реализации всего алгоритма (11), определяемой соотношениями (12), результаты вычислений $y_{ij}$ должны использоваться в качестве входных данных $x_{kj}$ для следующего умножения матриц. Вершины для входных значений $x_{ij}$ размещены в точках $(1, j, i)$, вершины вывода расположены в точках $(i, j, N)$. Рассмотрим возможность построения $\mathbf{L}^\lambda$-цепи при $\lambda \leqslant 2$ для реализации (11) на основе единственного ГЗ.

Так как среди 16 исходных ГЗ выявлено 8 пар подобных, то можно ограничиться рассмотрением только первых восьми $\text{ГЗ}_i$, $1 \leqslant i \leqslant 8$. Более того, так как матрица $\Phi$ не оказывает влияния на возможность существования пространственной стыковки, то из рассмотрения можно временно исключить $\text{ГЗ}_2$, $\text{ГЗ}_4$, $\text{ГЗ}_6$ и $\text{ГЗ}_8$ (эти ГЗ имеют те же возможности для пространственной стыковки, что и $\text{ГЗ}_1$, $\text{ГЗ}_3$, $\text{ГЗ}_5$ и $\text{ГЗ}_7$ соответственно).

В соответствии с теоремой 2 $\mathbf{L}^\lambda$-цепь при $\lambda \leqslant 2$ может быть построена только на основе $\text{ГЗ}_1$ (или $\text{ГЗ}_2$ и подобных им $\text{ГЗ}_{13}$ и $\text{ГЗ}_{14}$) и $\text{ГЗ}_7$ (или $\text{ГЗ}_8$ и подобных им $\text{ГЗ}_{11}$ и $\text{ГЗ}_{12}$).

Действительно, для ГЗ$_1$ $D_O D_I^{-1} = D_I D_O^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$, для ГЗ$_7$ $D_O D_I^{-1} = D_I D_O^{-1} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$, однако для ГЗ$_3$ $D_O D_I^{-1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} = D_I D_O^{-1}$ и для ГЗ$_5$ $D_O D_I^{-1} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix} \neq \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} = D_I D_O^{-1}$.

Все допустимые варианты пространственного построения $\mathbf{L}^\lambda$-цепи при $\lambda \leqslant 2$ (на основе ГЗ$_1$, ГЗ$_7$, ГЗ$_2$ и ГЗ$_8$) допускают единое таймирование в соответствии с теоремой 5.

Рассмотрим возможность построения $\mathbf{L}_*^\lambda$-цепи. В силу следствия из теоремы 4 существуют две возможности построения $\mathbf{L}_*^\lambda$-цепи: либо на основе ГЗ$_1$ (или ГЗ$_2$) и ГЗ$_7$ (или ГЗ$_8$), либо на основе ГЗ$_3$ (или ГЗ$_4$) и ГЗ$_5$ (или ГЗ$_6$). Проверка выполнения условий теоремы 4 показывает, что пространственное построение $\mathbf{L}_*^\lambda$-цепи возможно только на основе ГЗ$_3$ (или ГЗ$_4$) и ГЗ$_5$ (или ГЗ$_6$). Проверка выполнения условий теоремы 6 показывает, что для $\mathbf{L}_*^\lambda$-цепи на основе ГЗ$_3$ и ГЗ$_5$ и для $\mathbf{L}_*^\lambda$-цепи на основе ГЗ$_4$ и ГЗ$_6$ возможно и единое таймирование.

Таким образом, матричное представление областей ввода и вывода обеспечило быстрое нахождение подобных графов зависимостей из 16 первоначально полученных, получение аффинных операторов для построения состыкованных графов зависимостей на основе как единственного ГЗ, так и на основе двух различных ГЗ, а также нахождение единого временного режима работы всей вычислительной структуры.

## Литература


1. **Воеводин В.В.** Математические модели и методы в параллельных процессах. М.: Наука, 1986.
2. **Kung S.-Y.** VLSI Processor Arrays. Prentice-Hall Int., 1988.
3. **Quinton P., Robert Y.** Systolic algorithms and architectures. Prentice-Hall and Masson, 1989.
4. **Lengauer C.** Parallel Computing Technologies. Ed. N. Mirenkov. World Scientific, 1991. P. 32–46.
5. **Соболевский П.И., Лиходед Н.А.** Внешняя стыковка при реализации одношаговых итерационных процессов на систолических структурах. Мн., 1990 (Препринт / Ин-т математики АН Беларуси: 15(415)).
6. **Tiountchik A.A.** // Proc. Int. Workshop on Parallel Processing by Cellular Automata and Arrays – Parcella'96. Eds. R. Vollmar, W. Erhard, V. Jossifov. Berlin, 1996. P. 77–84.
7. **Karp R.M., Miller R.E., Winograd S.** The organization of computations for uniform recurrence equations // J. ACM. 1967. Vol. 14. P. 563–590.
8. **Лиходед Н.А., Соболевский П.И.** Стыковка базовых систолических вычислителей // Докл. АН БССР. 1989. № 5. С. 389–392.



*Тиунчик Александр Александрович*
*Институт математики НАН Беларуси*
*г. Минск, ул. Сурганова, 11*
*220072, Беларусь*
*e-mail: aat@im.bas-net.by*