МАТЕМАТИЧЕСКОЕ И АРХИТЕКТУРНОЕ ОБЕСПЕЧЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
(Вычислительные системы)
1985 Выпуск 109

УДК 681.32:519.68

ПРОЕКТИРОВАНИЕ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ И ВЫЧИСЛИТЕЛЬНЫХ СТРУКТУР ДЛЯ БЫСТРОГО ОБРАЩЕНИЯ МАТРИЦ

С.Г.Седухин

Введение

В последние годы большое внимание уделяется проектированию вычислительных сетей для прямого осуществления некоторых матричных алгоритмов в аппаратуре на базе сверхбольших интегральных схем. Разработанные сети для этих алгоритмов часто являются асимптотически оптимальными для реализации в сверхбольших интегральных схемах [1]. Математической мерой сложности осуществления проекта в сверхбольших интегральных схемах является произведение AT^2, которое заключает в себе компромисс между стоимостью изготовления кристалла с площадью A и временем T выполнения кристаллом заданных вычислений. Основываясь на модели сверхбольших интегральных схем [1], в работе [2] было показано, что любой проект для умножения двух $(n \times n)$-матриц должен удовлетворять оценке $AT^2 \geq \Omega(n^4)$. Так как существует простое сведение матричного произведения к обращению треугольной матрицы, то полученная оценка будет справедливой и для задачи обращения неособенной треугольной $(n \times n)$-матрицы. Соответствующие, асимптотически оптимальные вычислительные сети, удовлетворяющие нижней оценке $AT^2 = O(n^4)$, были описаны в работе [3].

В настоящей работе, на основе разработанного ранее [4] систематического подхода, описываются возможные пространственно-временные выполнения алгоритма обращения плотной $(n \times n)$-матрицы на регулярных асинхронных вычислительных сетях. Некоторые из представленных сетей являются асимптотически оптимальными для реализации на базе существующих (планарных) и перспективных (трехмерных)

сверхбольших интегральных схем. Другие могут быть использованы при построении вычислительных систем на базе существующих быстродействующих ЭВМ.

§ I. Общая схема для обращения неособенной (n x n)-матрицы

Пусть $A = [a_{ij}]$ будет неособенной плотной $(n \times n)$-матрицей, которая разбита на $N \times N$ клеточных матриц, где $N = n/r$, r - натуральное число в диапазоне $[1, n/2]$. Пусть $A(I,J)$ обозначает (I,J)-ю клетку матрицы A для $I, J \in \{1,2,\ldots,N\}$, т.е.

$$A(I,J) = [a_{(I-1)r+p,(J-1)r+q}],\ 1 \le p,q \le r;$$

и пусть $A^{-1}(I,J)$ будет соответствующая (I,J)-я клетка обратной матрицы A^{-1}.

Нахождение обратной матрицы будем осуществлять методом обмена [5]. Метод основан на преобразовании системы уравнений $y = Ax$ в систему $x = A^{-1}y$ путем последовательной замены в системе $y = Ax$ строки i на столбец j. Клеточный вариант метода описывается системой рекуррентных соотношений, задающей множество вычислений для каждого шага $K = 1,2,\ldots,N$ и $1 \le I, J \le N$:

$$\left.\begin{aligned}
&A(K,K,K) = A^{-1}(K,K,K-1);\\
&A(K,J,K) = -A(K,K,K)A(K,J,K-1);\ 1 \le J \le N,\ J \ne K;\\
&A(I,K,K) = A(K,K,K)A(I,K,K-1),\ 1 \le I \le N,\ I \ne K;\\
&A(I,J,K) = A(I,J,K-1) + A(I,K,K)A(K,J,K),\ I \ne K,\ J \ne K;
\end{aligned}\right\} \quad (1)$$

где $A(I,J,K)$ является (I,J)-й клеткой матрицы A на K-м шаге вычислений; $A(I,J,0) = A(I,J)$ для $1 \le I,J \le N$. Ясно, что $A(I,J,N) = A^{-1}(I,J)$, и мы полагаем, что $A(I,J,N+1) = A(I,J,N)$ для $1 \le I,J \le N$.

Как следует из приведенных соотношений (I), на каждом K-м шаге вычислений для каждой пары (I,J) требуется выполнить макрооперацию обращения, либо произведения, либо накопления произведений клеточных $(r \times r)$-матриц. В свою очередь, каждая из этих макроопераций требует выполнения $O(r^3)$ арифметических операций типа умножения и сложения. Очевидно, что при $r = 1$ клеточный вариант алгоритма вырождается в числовой случай, при котором макроопераций над клеточными $(r \times r)$-матрицами заменяются на соответствующие арифметические операции над числами. Так как число макроопераций

для выполнения всего множества вычислений есть величина n^3, то операционная (арифметическая) сложность алгоритма составит величину $O(N^3r^3) = O(n^3)$. Ясно, что при последовательной реализации алгоритма временная сложность вычислений определится операционной сложностью метода.

§ 2. Представление множества вычислений в решетчатом пространстве

Первоочередным шагом разработанного ранее подхода к синтезу регулярных вычислительных сетей [4] является представление исходных рекуррентных соотношений в виде эквивалентной системы уравнений, задающей требуемые вычисления и явно определяющей непосредственные входные-выходные связи между ними в целочисленном решетчатом пространстве. Для рассматриваемого алгоритма область всех вычислений определяется индексным множеством:

$P = P_{IN} \cup P_{INT} \cup P_{OUT}$,

где $P_{IN} = \{(I,J,0)/ 1 \le I,J \le N\} \subseteq \mathbb{Z}^2$ определяет область входных вычислений; $P_{INT} = \{(I,J,K)/1 \le I,J,K \le N\} \subseteq \mathbb{Z}^3$ - область внутренних вычислений; а $P_{OUT} = \{(I,J,N+1)/1 \le I,J \le N\} \subseteq \mathbb{Z}^2$ задает область выходных вычислений (см.рис.I).

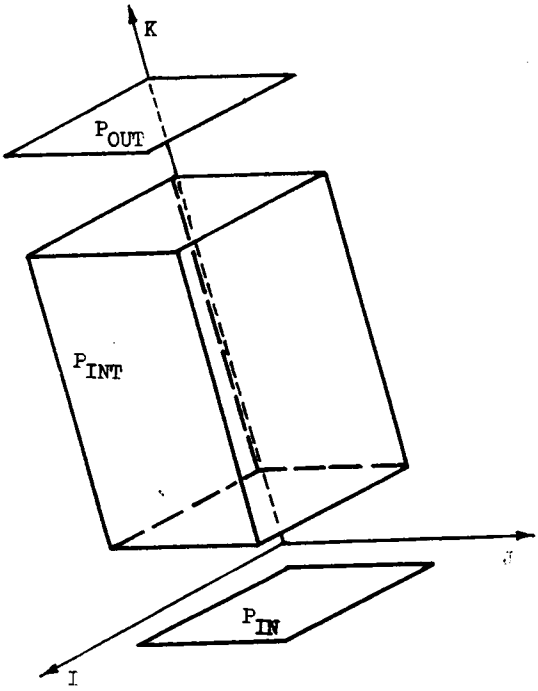

Рис. I

Множеству вычислений, заданному соотношением (I), соответствует система уравнений, определяющая обработку для каждой точ-

ки $p = (I,J,K) \in P_{INT}$ и задающая связь данной точки от своих непосредственных предшественников:

$$
\left.
\begin{aligned}
&X_I(p) = A(p-\theta_I);\\
&X_J(p) = A(p-\theta_J);\\
&X_K(p) = A(p-\theta_K);\\
&A(p) = \underline{\text{if}}\ I = J = K\ \underline{\text{then}}\ X_K^{-1}(p)\ \underline{\text{else}}\\
&\qquad \underline{\text{if}}\ I = K\ \underline{\text{then}}\{-X_J(p)\ X_K(p)\}\ \underline{\text{else}}\\
&\qquad\ \underline{\text{if}}\ J = K\ \underline{\text{then}}\{X_I(p)\ X_K(p)\}\ \underline{\text{else}}\\
&\qquad\qquad \{X_K(p) + X_J(p)X_I(p)\};
\end{aligned}
\right\} \qquad (2)
$$

где $\theta_I = (I-K,0,0)$, $\theta_J = (0,J-K,0)$, $\theta_K = (0,0,1) = e_3$ являются векторами непосредственной информационной зависимости точки $p \in P_{INT}$. Системе уравнений (2) соответствует граф непосредственной информационной зависимости вычислений $\Gamma = (P,\theta)$, где $\theta = \{\theta_I, \theta_J, \theta_K\}$, который изображен на рис.2 для случая $N = 4$.

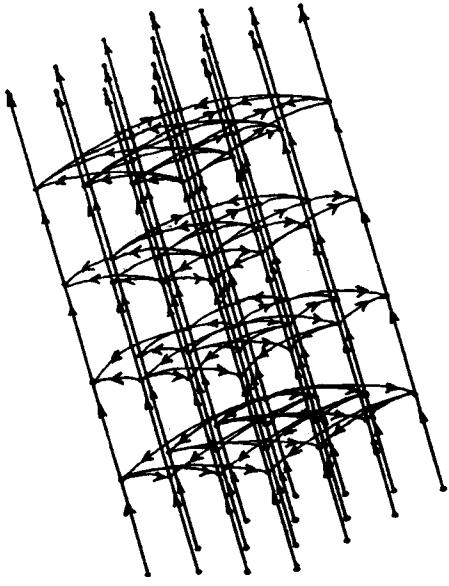

Рис. 2

Зависимость векторов θ_I и θ_J от координат точки $p = (I,J,K)$ задает трансляционную (глобальную) схему обмена (r x r)-матрицами между точками решетчатого пространства. Отметим, что векторы θ_I и θ_J в зависимости от значений координат I,J,K могут иметь как положительные, так и отрицательные направления. Переход к локальной схеме обменов [4] дает эквивалентную к (2) систему уравнений, свободную от трансляций:

$$
\left.
\begin{aligned}
X_I(p) = \underline{\text{if}}\ I = K\ \underline{\text{then}}\ X_K(p)\ \underline{\text{else}}\\
\underline{\text{if}}\ I-K = 1\ \underline{\text{then}}\ A(p-e_1)\ \underline{\text{else}}\\
\underline{\text{if}}\ K-1 = 1\ \underline{\text{then}}\ A(p+e_1)\ \underline{\text{else}}
\end{aligned}
\right|
$$

$$
\left.\begin{aligned}
X_I(p) &= \underline{\text{if}}\ I-K>1\ \underline{\text{then}}\ X_I(p-e_1)\ \underline{\text{else}}\ X_I(p+e_1);\\
X_J(p) &= \underline{\text{if}}\ J=K\ \underline{\text{then}}\ X_K(p)\ \underline{\text{else}}\\
&\quad \underline{\text{if}}\ J-K=1\ \underline{\text{then}}\ A(p-e_2)\ \underline{\text{else}}\\
&\quad \underline{\text{if}}\ K-J=1\ \underline{\text{then}}\ A(p+e_2)\ \underline{\text{else}}\\
&\quad \underline{\text{if}}\ J-K>1\ \underline{\text{then}}\ X_J(p-e_2)\ \underline{\text{else}}\ X_J(p+e_2);\\
X_K(p) &= A(p-e_3);\\
A(p) &= \underline{\text{if}}\ I=K=J\ \underline{\text{then}}\ X_K^{-1}(p)\ \underline{\text{else}}\\
&\quad \underline{\text{if}}\ I=K\ \underline{\text{then}}\ \{-X_J(p)X_K(p)\}\ \underline{\text{else}}\\
&\quad \underline{\text{if}}\ J=K\ \underline{\text{then}}\ \{X_I(p)X_K(p)\}\ \underline{\text{else}}\\
&\quad \{X_K(p)+X_J(p)\cdot X_I(p)\};
\end{aligned}\right\} \qquad (3)
$$

где $e_1=(1,0,0)$, $e_2=(0,1,0)$, $e_3=(0,0,1)$ являются векторами локальной информационной зависимости точки $p=(I,J,K)\in P_{INT}$.

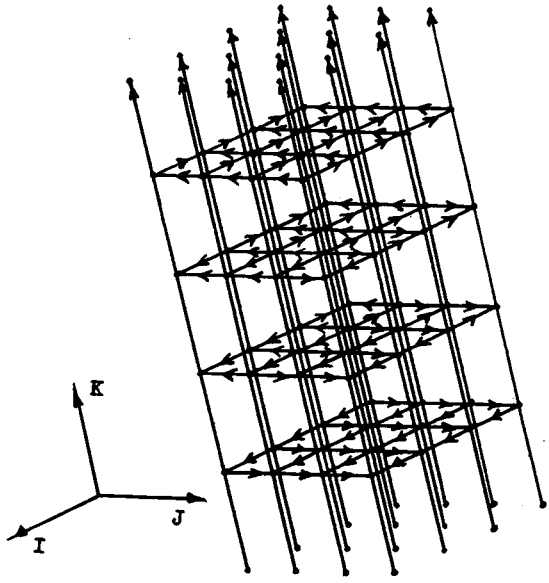

Рис.3

Приведенной системе уравнений (3) можно сопоставить граф локальной зависимости вычислений $\Gamma^*=(P,E(I,J,K))$, где множество

$$
\begin{aligned}
E(I,J,K) = \underline{\text{if}}\ I=J=K\ \underline{\text{then}}\ \{e_3\}\ \underline{\text{else}}\\
\underline{\text{if}}\ (I=K)\,\&\,(J<K)\ \underline{\text{then}}\ \{-e_2,e_3\}\ \underline{\text{else}}\\
\underline{\text{if}}\ (I=K)\,\&\,(J>K)\ \underline{\text{then}}\ \{e_2,e_3\}\ \underline{\text{else}}
\end{aligned}
$$

$$\underline{\text{if}}\ (I<K)\,\&\,(J=K)\ \underline{\text{then}}\ \{-e_1,e_3\}\ \underline{\text{else}}$$
$$\underline{\text{if}}\ (I>K)\,\&\,(J=K)\ \underline{\text{then}}\ \{e_1,e_3\}\ \underline{\text{else}}$$
$$\underline{\text{if}}\ (I<K)\,\&\,(J<K)\ \underline{\text{then}}\ \{-e_1,-e_2,e_3\}\ \underline{\text{else}}$$
$$\underline{\text{if}}\ (I<K)\,\&\,(J>K)\ \underline{\text{then}}\ \{-e_1,e_2,e_3\}\ \underline{\text{else}}$$
$$\underline{\text{if}}\ (I>K)\,\&\,(J<K)\ \underline{\text{then}}\ \{e_1,-e_2,e_3\}\ \underline{\text{else}}$$
$$\underline{\text{if}}\ (I>K)\,\&\,(J>K)\ \underline{\text{then}}\ \{e_1,e_2,e_3\}.$$

Граф локальной информационной зависимости вычислений Γ^* для случая $N=4$ показан на рис.3.

Следующим шагом является рассмотрение различных пространственно-временных реализаций алгоритма.

§ 3. Пространственно-временные осуществления множества вычислений

В общем случае пространственно-временные осуществления алгоритма задаются отображениями:

$$\Phi_d:\ \Gamma^* \xrightarrow[s]{m^d(p)\ t^d(p)} S_d \times T_d \subseteq \mathbb{Z}^{d+1} = \mathbb{Z}^d \times \mathbb{Z},$$

где $d \in \{0,1,\ldots,\dim(P)\}$ – выбранная размерность физического пространства (сети) реализации вычислений $S_d \subseteq \mathbb{Z}^d$, $\dim(P)$ – размерность области вычислений алгоритма (для подавляющего числа практических алгоритмов $\dim(P) \le 3$); $m^d(p)$ – мультииндекс пространственной точки (одиночного вычислителя) из S_d, на которую отображается одна или множество вершин графа Γ^*; s – вектор направления проекции графа Γ^* на сеть S_d; $t^d(p)$ – порядковый номер (ранг) осуществления обработки в вершине $p \in P$ вычислителем $m^d(p) \in S_d$ из общей последовательности $t^d(p) \in \{0,1,\ldots,T_d\}$; T_d – заключительный номер в последовательности вычислений при реализации алгоритма в d-мерной сети.

Рассмотрим теперь три интересных для нас случая, когда $d = 1,2,3$.

1. Трехмерное осуществление алгоритма. Реализация множества вычислений при $d = \dim(P) = 3$ задается аффинным преобразованием Φ_3, которое непосредственно вкладывает граф Γ^* в структуру сети вычислителей $S_3 \subseteq \mathbb{Z}^3$, т.е. $m^3(p) \equiv p$. Таким образом, при трехмерном осуществлении алгоритма каждой вершине $p \in P_{INT}$ графа $\Gamma^* = (P,E)$ однозначно сопоставляется вычислитель с одноименными координатами, а векторам из E – коммуникационные каналы связи меж-

ду соседними вычислителями. Очевидно, что $\|S_3\| = \|P_{INT}\| = N \times N \times N$, т.е. сеть содержит N^3 ортогонально связанных вычислителей, каждый из которых срабатывает при получении всех необходимых для данного шага входных переменных $X_I(p)$, $X_J(p)$, $X_K(p)$. Найдем теперь ранжирующую функцию $t^3(p)$, которая определяет порядок прихода в вычислитель $m^3(p) \in S_3$ всех входных переменных. Заметим, что решетчатый граф Γ^* не является координатным [6], для которого ранжирующая функция находится сравнительно просто [7]. Более того, мы рассматриваем асинхронную реализацию алгоритма, требующую задания на множестве вычислений специфических условий выполнения.

Каждому шагу обработки $K \in \{1,2,\ldots,N\}$ соответствует подграф $\Gamma^*_K = (P^K, E(I, J, K))$, который является K-й плоскостью графа Γ^*. Нетрудно показать, что среди множества вершин P^K_{INT} имеется единственная минимальная вершина $p^{min}_K = (K,K,K)$, для которой $E(K, K, K) = e_3$, т.е. зависящая только от непосредственно предшествующей ей вершины p_{K-1} из P^{K-1}_{INT} или P_{IN} при $K = 1$. Вершина p^{min}_K называется начальной вершиной K-го шага обработки, так как только с нее может начаться процесс вычисления остальных вершин данного шага.

Отношение непосредственного предшествования вычислений структурно заданного ориентированными дугами графа Γ^*_K индуцирует на частично упорядоченном множестве вершин $P^K_{INT} \subseteq P$ отношение эквивалентности $\sim$, разбивающее множество P^K_{INT} на классы эквивалентности:

$$[P^K_{INT}]_\sim = \{[P^K_{INT}(\pi)]_\sim / 0 \le \pi \le \pi^{max}_K,\ \pi = |I-K| + |J-K|\},$$

где $[P^K_{INT}(\pi)]_\sim = \{p \sim q / p,q \in P^K_{INT},\ \pi(p^{min}_K, p) = \pi(p^{min}_K, q) = \pi\}$ является классом взаимно независимых вершин, отстоящих в графе Γ^*_K на одинаковом расстоянии (пути) $\pi = I,2,\ldots,\pi^{max}_K$ от начальной вершины $p^{min}_K = (K, K, K) \equiv [P^K_{INT}(0)]_\sim$. Расстояние $\pi^{max}_K = \pi(p^{min}_K, p^{max}_K)$, где p^{max}_K - вершина (вершины) из P^K_{INT}, максимально удаленная в Γ^*_K от вершины p^{min}_K. Ясно, что $[P^K_{INT}(\pi_1)]_\sim \cap [P^K_{INT}(\pi_2)]_\sim = \emptyset$ при $\pi_1 \neq \pi_2$, т.е. $[P^K_{INT}(\pi)]_\sim$ попарно не пересекаются. Кроме того,

$$\bigcup_{0 \le \pi \le \pi^{max}_K} [P^K_{INT}(\pi)]_\sim = P^K_{INT},$$

т.е. все вместе $[P^K_{INT}(\pi)]_\sim$ исчерпывают P^K_{INT}. Нетрудно показать, что для каждого шага обработки $K \in \{1,2,\ldots,N\}$ расстояние $\rho^{max}_K = \max[2(K-1), 2(N-K)]$, так как если N четно, то

$$p^{max}_K = \begin{cases} (N,N,K) & \text{при } 1 \le K \le N/2; \\ (1,1,K) & \text{при } N/2 < K \le N; \end{cases}$$

а при нечетном N;

$$p^{max}_K = \begin{cases} (N,N,K) & \text{при } 1 \le K < (N+1)/2; \\ \{(1,1,K);(1,N,K);(N,1,K);(N,N,K)\} & \text{при } K = (N+1)/2; \\ (1,1,K) & \text{при } (N+1)/2 < K \le N. \end{cases}$$

Таким образом, для всего частично упорядоченного множества вычислений, заданного графом Γ^*, существует единственный минимальный элемент $p^{min}_1 = (1,1,1) \in P^1_{INT}$ и единственный максимальный элемент $P^{max}_N = (1,1,N) \in P^N_{INT}$. Элементы каждого множества $[P^K_{INT}]_\sim$ для $K \in \{1,2,\ldots,N\}$ линейно упорядочены, т.е.

$$p^{min}_K \equiv [P^K_{INT}(0)]_\sim <\cdot [P^K_{INT}(1)]_\sim <\cdot \ldots <\cdot [P^K_{INT}(\rho^{max}_K)]_\sim \equiv p^{max}_K,$$

где $<\cdot$ – двоичное иррефлексивное асимметричное транзитивное отношение предшествования. Кроме того, линейно упорядочена последовательность рекуррентных шагов вычислений, т.е. $P^1_{INT} <\cdot P^2_{INT} <\cdot \ldots <\cdot P^N_{INT}$.

Порядок, но не время, выполнения вершин $p_K \in P^K_{INT}$ задается ранжирующей функцией $t^3(p_K) \in \mathbb{Z}$, отражающей последовательность формирования потоков через вершину p_K. Заметим, что величина $t^3(p_k)$ характеризует время только для модели синхронных вычислений. Для модели асинхронных вычислений время $t \in \mathbb{R}$. Очевидно, что если $t^3(p^{min}_K)$ является рангом минимальной вершины, то $t^3(p_K) = t^3(p^{min}_K) + \pi(p^{min}_K, p_K)$. Можно показать, что при $t^3(p^{min}_1) = 0$ величина $t^3(p^{min}_K) = 3(K-1)$. Следовательно, $t^3(p_K) = 3(K-1) + |I-K| + |J-K|$ и все множество вычислений завершится на ранге $T_3(N) = t^3(p^{max}_N) = 5(N-1)$, так как $p^{max}_N = (1,1,N)$. Ясно, что величина $T_3(N) = \pi(p^{min}_1, p^{max}_N)$, т.е. равна длине критического пути в графе Γ^*. Так как выполнение каждой вершины требует $O(r^3)$ арифметических операций, то величина $T_3(N)\ O(r^3) = O(n^3/N^2)$ определит

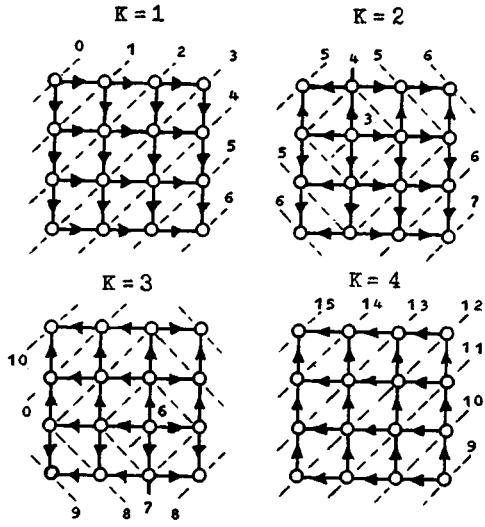

Рис. 4

общее **время завершения** вычислений на сети S_3. На рис.4 пунктирными линиями показаны значения ранжирующей функции для случая $N = 4$.

Определим теперь период вычислений $D_3(N)$ как время начала обработки сетью S_3 нового набора исходных данных. Нетрудно показать, что $D_3(N) = O(r^3)$ так,что M задач решается за время $T_3(N,M)=T_3(N)+(M-1)D_3(N)= = O(n^3/N^2 + Mn^3/N^3)$. Таким образом, в общем случае справедлива

ТЕОРЕМА I. Обращение (nxn)-матрицы, разбитой NxN клеток, методом обмена может быть осуществлено трехмерной сетью из NxNxN ортогонально связанных вычислителей, с локальной памятью каждого $O(r^2)$ слов, за время $O(n^3/N^2)$ и периодом вычислений $D_3(N) = O(r^3)$, где $r = n/N$.

Следовательно, при неограниченном параллелизме, т.е. когда $r = I$ и $N = n$, сеть из nxnxn вычислителей обратит (n xn)-матрицу за время $O(n)$ с периодом $D_3(n) = O(1)$. Каждый вычислитель сети требует при этом $O(1)$ регистров для хранения, обработки и обмена данными. Время решения на такой трехмерной сети m задач определяется величиной $T_3(n,m) = O(n+m)$. Так как размер каждого вычислителя сети не зависит от размера решаемой задачи (порядка исходной матрицы), то физический размер (объем) всей сети $A_3(n)$ можно измерять в числе вычислителей, т.е. $A_3(n) = O(n^3)$. Тогда критерий эффективности, связывающий физический размер трехмерного кристалла сверхбольшой интегральной схемы, время и период вычислений, оценится величиной $A_3(n)T_3(n)D_3(n) = O(n^4)$, т.е. сеть оп-

тимальна для осуществления в аппаратуре на базе перспективных трехмерных сверхбольших интегральных схем [8]. Планарность существующей технологии сверхбольших интегральных схем требует отображения множества вычислений алгоритма на плоские сети.

2. Двухмерная реализация множества вычислений. Рассмотрим теперь отображение Φ_2, проецирующее трехмерный решетчатый граф Γ^* на плоскую сеть S_2 по некоторому направлению $s = (I, J, K)$. Среди множества возможных направлений проекций интересными будут только те, которые обеспечивают $\min_s \{\|S_2^s\|/T_2(N) = T_3(N)\}$, где $\|S_2^s\|$ - физический размер двухмерной сети, получаемой при отображении графа Γ^* по направлению s. Так как Γ^* представляет собой куб, то $\min \{\|S_2^s\|\} = N \times N$ обеспечивается только для направлений $s_I = e_1$, $s_J = e_2$, $s_K = e_3$. Однако так как вычисления рекуррентно связаны по K, то только для $s_K = e_3$ выполняется условие $T_2(N) = T_3(N)$. При такой двухмерной реализации множества вычислений на каждый (I,J)-й вычислитель сети $(1 \le I, J \le N)$ отображаются N вершин (I,J,K) графа Γ^* для $K = 1,2,\ldots, N$. Ясно, что локальная память каждого вычислителя должна составлять при этом $O(r^2) = O(n^2/N^2)$ слов. Ранжирующая функция $t^2(p_k) = t^3(p_K)$, т.е. $T_2(N) = T_3(N)$. Освобождение вычислителей сети S_2, реализующих некоторый шаг обработки, используется, как только это становится возможным, для осуществления последующих шагов вычислений. Для рассматриваемой реализации алгоритма характерно наличие N волн вычислений, каждая из которых связана с выполнением отдельного рекуррентного шага обработки. Положения фронтов волн для случая $N = 4$ показаны на рис.4 пунктирными линиями.

Так как вершины p_1^{min} и p_N^{max} реализуются в сети одним вычислителем с координатами $(1,1) \in S_2$, то период вычислений $D_2(N) = T_2(N) + 1 = 3N - 4$, т.е. M задач решается на сети за время $T_2(N,M) = O(MNr^3) = O(Mn^3/N^2)$. Ввод и вывод данных может быть эффективно осуществлен по дополнительным диагональным каналам (см. рис.5). При вводе данных (I,J)-й вычислитель сети S_2 на ранге $t^2(p_1) = I + J - 2$ принимает по диагональному каналу (I,J)-ю клетку исходной матрицы A от соседнего $(I-1, J-1)$-го вычислителя или извне и обрабатывает ее. На всех последующих рангах (I,J)-й вычислитель превращается в транзитный узел по данному диагональному направлению. Заметим, что ввод $(r \times r)$ клеток матрицы может осуществляться последовательно, так как дальнейшая обработка тре-

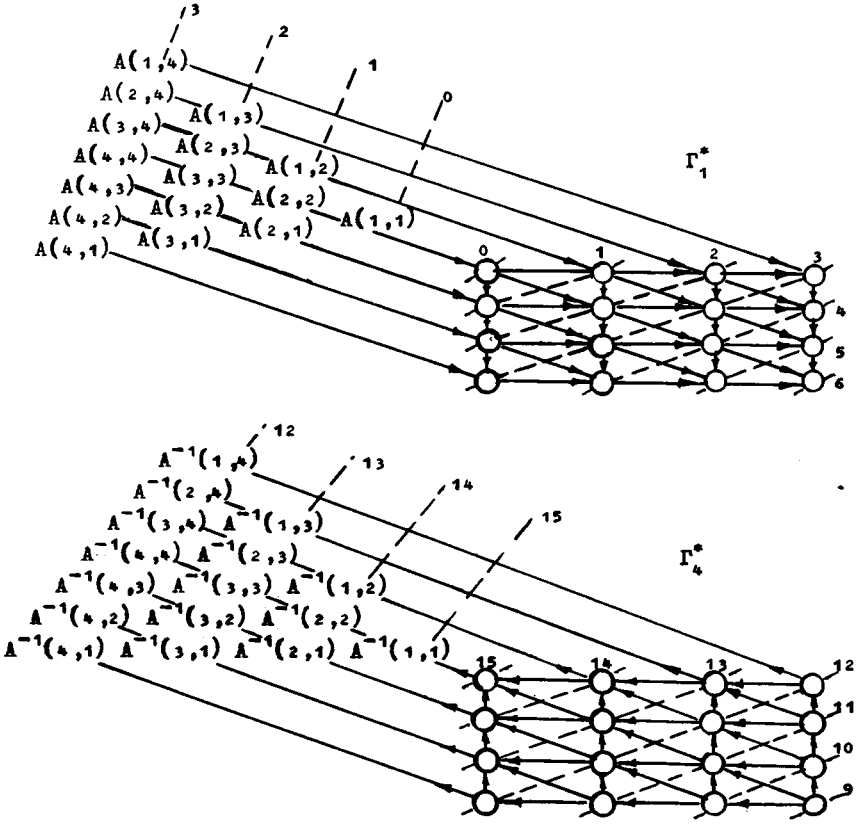

Рис. 5

бует $O(r^3)$ времени. Таким образом, ввод данных выполняется на фоне основной обработки, затрагивающей ортогональные каналы связи. При выводе результирующих клеточных матриц каждый (I,J)-й вычислитель является транзитным узлом вплоть до ранга $t^2(p_N) = 3N - (I+J+3)$, на котором он определяет клетку $A(I,J,N) = A^{-1}(I,J)$, посылая ее в диагональный канал связи. Справедлива

ТЕОРЕМА 2. Обращение $(n \times n)$-матрицы, разбитой $N \times N$ клеток $(N = n/r)$ может быть осуществлено сетью из $N \times N$ гексагона-

льносвязанных вычислителей, с локальной памятью каждого $O(n^2/N^2)$ слов, за время $O(n^3/N^2)$, включающее ввод/вывод данных, с периодом вычислений $D_2(N) = O(n^3/N^2)$.

Таким образом, при неограниченном параллелизме ($N = n$) сеть из $n \times n$ гексагонально связанных вычислителей обратит $(n \times n)$-матрицу за время $T_2(N) = O(n)$ с периодом вычислений $D_2(n) = O(n)$. Как и выше, каждый вычислитель сети для хранения, обработки и обмена данными должен иметь $O(1)$ регистров. Время обращения m матриц составит в этом случае величину $T_2(n,m) = O(nm)$.

Оценим теперь данный проект для случая неограниченного параллелизма на оптимальность осуществления в аппаратуре на базе сверхбольших интегральных схем. Согласно теоретически полученной оценке [2], для планарных сверхбольших интегральных схем оптимальным считается проект, для которого $A_2(n)\,T_2^2(n) = O(n^4)$. Так как в нашем случае $A_2(n) = \|S_2\| = O(n^2)$, $T_2(n) = D_2(n)$, то критерий $A(n)D(n)T(n)$ эквивалентен критерию $A(n)T^2(n)$. Следовательно, для нашего проекта $A_2(n)T_2^2(n) = O(n^4)$, что указывает на оптимальность сети для осуществления на сверхбольших интегральных схемах.

Необходимо отметить, что, используя относительную независимость вычислений от места осуществления на сети, можно применять другие стратегии организации обработки. Например, сдвигая получаемые на каждом K-м шаге результаты $A(I,J,K)$ в вычислитель с координатами $(I-K, J-K) \bmod N$, получаем сеть, которая будет ориентирована в одном (положительном) направлении. В этом случае обеспечивается специализация функций вычислителей в сети, т.е. вычислитель с координатами $(1,1)$ всегда будет выполнять обращение $(r \times r)$-матрицы (деление для числового варианта алгоритма), вычислители первой строки $(1,I)$ и первого столбца $(I,1)$ сети $(1 \le I \le N)$ - осуществлять произведение, а все остальные - матричную операцию накопления. Однако время выполнения алгоритма в данном случае будет больше, так как величина $T_2(N) = 6(N-1)$.

Заметим также, что при трехмерной и двумерной реализациях алгоритма приведенная выше система уравнений (3) фактически является программой, которая описывает последовательность действий одиночного вычислителя в трехмерной или двумерной сети. При этом функции по приему соответствующих данных задаются явно, а функции по передаче данных - неявно.

3. Одномерная реализация алгоритма. Одномерное осуществление множества вычислений может быть задано многими способами в зависимости от выбранного направления проецирования последовательности подграфов $\Gamma^*_K = (P^K, E(I, J, K))$ для $K = 1,2,\ldots, N$ на линейную сеть вычислителей. Так же, как и выше, нас интересует такое направление проекции s, которое обеспечивает $\min_s \{\| S_1 \|\}$. Нетрудно показать, что для данного алгоритма такими направлениями будут $s_I = (1,0)$ и $s_J = (0,1)$, отображающие графы Γ^*_K на линейную сеть $\| S_1 \| = N$ по строкам или по столбцам соответственно. Проецирование графов по направлению $s_{IJ} = (1,1)$ определяет сеть $\| S_1 \| = 2N-1$.

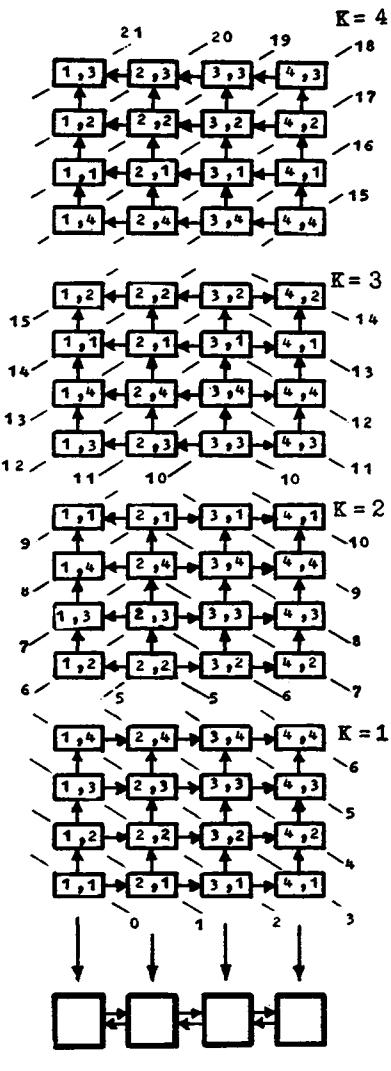

Рис.6

При отображении последовательности графов вдоль направления s_I или s_J, каждый вычислитель должен обработать N^2 вершин графа Γ^*. Ранжирующая функция при отображении, например вдоль s_I, будет иметь вид: $t^1(p_K) = (N+1)(K-1) + |K-1| + |[N \oplus (J-K+1)] \bmod N-1|$. Значения этой функции показаны пунктирными линиями на рис.6 для случая $N = 4$. Локальная память каждого вычислителя сети составляет при этом $O(Nr^2) = O(n^2/N)$ слов. Так как $p_N^{max} = (1, N-1, N)$ (см.рис.6), то $T_1(N) = N^2+2N-3$, т.е. время одномерной реализации алгоритма, с учетом веса матричной операции, составит ве-

личину $T_1(N)$ $O(r^3) = O(N^2r^3) = O(n^3/N)$. Нетрудно убедиться, что $D_1(N) = T_1(N) + 1 = O(n^3/N)$. Таким образом, справедлива

ТЕОРЕМА 3. Обращение (nxn)-матрицы, разбитой NxN клеток (N=n/r), может быть осуществлено сетью из N линейно связанных вычислителей, с локальной памятью каждого $O(n^2/N)$ слов, за время $O(n^3/N)$ и периодом вычислений $D_1(N)=O(n^3/N)$.

Заметим, что в данном случае последовательный ввод/вывод (nxn)-матрицы в сеть не ухудшит приведенных временных оценок. Таким образом, при неограниченном параллелизме (N = n) сеть из n линейно связанных вычислителей осуществит обращение (nxn)-матрицы за время $T_1(n) = O(n^2)$, а m таких матриц – за время $T_1(n,m) = O(mn^2)$. Локальная память каждого вычислителя сети составит при этом O(n) слов.

Следует отметить, что, хотя последний проект и не является оптимальным для сверхбольших интегральных схем, его использование наиболее реалистично в современных условиях создания сетей из относительно небольшого числа (порядка 10) существующих быстродействующих ЭВМ типа АР-120, "Электроника-МТ-70М" или А-12 [9,10].

Литература

1. THOMPSON C.D. Area-Time Complexity for VLSI.- In:Proc.11th Annu.ACM Symp.on Theory of Comput., Atlanta, Georgia,1979,p.81-88.

2. SAVAGE J.E. Area-Time Tradeoffs for Matrix Multiplication and Related Problems in VLSI Models.- J.of Comput.and Syst.Science,1981,N 22,p.230-242.

3. PREPARATA F.P., VUILLEMIN J. Optimal Integrated - Circuit Implementation of Triangular Matrix Inversion. - In: Proc.Parallel Processing Conf.,1980,p.211-216.

4. СЕДУХИН С.Г. Систематический подход к проектированию вычислительных структур на базе СБИС.-Новосибирск.Б.и.,1985.- 41 с. (Препринт/ ВЦ СО АН СССР: № 589).

5. ГИЛЬБЕРТ А. Как работать с матрицами.- М.: Статистика, 1981.-158 с.

6. ВОЕВОДИН В.В., КРАСНОВ С.А. Математические вопросы проектирования систолических массивов. - М.,Б.и.,1985. - 26 с. (Препринт /Отдел вычислительной математики АН СССР: № 80).

7. QUINTON P. The Systematic Design of Systolic Arrays. - IRISA Internal Report, N 216, INRIA, Transe,1983.- 35 p.

8. ROSENBERG A.L. Three-Dimensional VLSI: A Case Study.- J. ACM,1983,v.30,N 3,p.397-416.

9. ТАЛОВ И.А. и др. Мини- и микро-ЭВМ и спецпроцессоры семейства "Электроника". - В кн.: Тез.докл.Всесоюз. конф. "Диалог - 82". Пущино, 1982.

10. Высокопроизводительный периферийный векторный процессор А-12 /Бродский И.И., Козлачков В.А., Коршевер И.И. и др. - Автометрия, 1984, № 4, с.29-35.

Поступила в ред.-изд.отд.
28 июня 1985 года