

## 2.5. СТЫКОВКА БАЗОВЫХ ВЫЧИСЛИТЕЛЬНЫХ ГРАФОВ

При построении параллельных форм сложных алгоритмов для реализации на матричных специализированных процессорах (МСП) естественно декомпозировать исходную задачу на простые подзадачи. Приведем формальный аппарат стыковки двух базовых вычислительных графов алгоритмов решения двух информационно связанных подзадач. Отображение состыкованных БВГ позволит получить параллельную реализацию исходного алгоритма на МСП.

Проблема состоит в следующем. После получения моделей отдельных МСП для подзадач их необходимо объединить в единую вычислительную систему, решающую исходную задачу. Место и время вывода промежуточных результатов из одного процессора может не соответствовать месту и времени ввода их во второй процессор. Возникает проблема стыковки МСП. Под стыковкой двух матричных специализированных процессоров понимается такое объединение их в единую вычислительную систему, при котором обмен данными выполняется только с помощью локальных связей, без каких-либо дополнительных устройств временного согласования ввода-вывода промежуточных результатов. Излагаемая далее теория позволяет учитывать условия стыковки на этапе синтеза моделей МСП.

Обозначим через  $G^\alpha = (V^\alpha, E^\alpha, T^\alpha, H^\alpha)$ ,  $\alpha = 1, 2$ , два базовых вычислительных графа:

$V^\alpha \subset \mathbf{Z}^d$ ,  $E^\alpha \subset \mathbf{Z}^d \times \mathbf{Z}^d$ , и дуги определяются множеством векторов зависимостей  $\Phi^\alpha = \{ \varphi_\alpha^{(k)}, 1 \leq k \leq K_\alpha \}$ ;

$t^\alpha(v) = \tau^\alpha \cdot (v - v_{\min}^\alpha) + t_0^\alpha$ ,  $\tau^\alpha \in K(G^\alpha)$ ,  $v \in V^\alpha$ , вершины  $v_{\min}^\alpha \in V^\alpha$  определяются условиями  $\tau^\alpha \cdot v_{\min}^\alpha = \min_{v \in V^\alpha} \tau^\alpha \cdot v$ ;

$H^\alpha = \{ h_{\varphi_\alpha^{(k)}} = \tau^\alpha \cdot \varphi_\alpha^{(k)}, \varphi_\alpha^{(k)} \in \Phi^\alpha, 1 \leq k \leq K_\alpha \}$  – множество задержек, а описание функционирования локальной памяти  $P_\lambda^\alpha(v)$ ,  $v \in V_\lambda^\alpha$ ,  $1 \leq \lambda \leq \Lambda$ ,  $\alpha = 1, 2$ , произведено согласно равенствам (2.4).

Пусть результаты вычислений в  $G^1$  являются входными данными для вычислений в  $G^2$ . Предположим, что вершины, реализующие вывод этих результатов из  $G^1$  и ввод их в  $G^2$ , расположены в граничных областях  $V^{out} \subseteq \partial(V^1)$  и  $V^{in} \subseteq \partial(V^2)$ , причем  $V^{out}$  и  $V^{in}$  принадлежат некоторым гиперплоскостям.

Под *стыковкой базовых вычислительных графов*  $G^1$  и  $G^2$  будем понимать построение нового кусочно-регулярного БВГ путем совмещения  $G^1$  и  $G^2$  в пространстве и согласования их работы по времени. *Пространственное совмещение (объединение)* означает размещение  $G^1$  и  $G^2$  в  $\mathbf{Z}^d$  так, чтобы  $V^1$  и  $V^2$  не пересекались, а дуги, соединяющие любую пару информационно связанных вершин  $v^{out} \in V^{out}$  и  $v^{in} \in V^{in}$ , и соответствующие им векторы были одинаковы. Временное согласование означает, что временные интервалы между выводом данных из  $G^1$  и их вводом в  $G^2$  одинаковы:

$$t^2(v^{in}) - t^1(v^{out}) = h, \quad (2.18)$$

где целое число  $h$  одно и то же для любой пары информационно связанных вершин  $v^{out} \in V^{out}$  и  $v^{in} \in V^{in}$ .

Формально реализацию пространственного совмещения  $G^1$  и  $G^2$  будем осуществлять аффинным преобразованием

$$L: v \rightarrow Av + b, \quad v \in V^2$$

(здесь  $v$  понимается как вектор-столбец) с целочисленными матрицей  $A$  и вектором-столбцом  $b$ .

**Теорема 2.4.** Пусть отображение  $L$  удовлетворяет условиям

а)  $A \cdot A' = A' \cdot A = E$  ( $E$  – единичная матрица, «'» – знак транспонирования),  $\det A = 1$ , т. е. матрица  $A$  задает собственное вращение,

б)  $V^1 \cap LV^2 = \emptyset$ ,

в)  $Lv^{in} - v^{out} = e$ ,

где  $LV^2 = \{\hat{v} \in \mathbf{Z}^d \mid \hat{v} = Lv, v \in V^2\}$ , а вектор  $e$  один и тот же для любой пары информационно связанных вершин  $v^{out} \in V^{out}$  и  $v^{in} \in V^{in}$ . Тогда отображение  $L$  осуществляет пространственное совмещение  $G^1$  и  $G^2$ .

В самом деле, совмещение  $G^1$  и  $G^2$  осуществляется преобразованием  $L$  без деформаций и отражений (условие а)), без пересечений (условие б)), с использованием постоянных связей (условие в)).

**Теорема 2.5.** Пусть базовые вычислительные графы  $G^1$  и  $G^2$  пространственно совмещены (т. е. существует преобразование  $L$ , удовлетворяющее условиям а)–в)). Тогда работа  $G^1$  и  $G^2$  согласована по времени (т. е. выполняется условие (2.18)) тогда и только тогда, когда

- 1)  $t_0^2 = t^1(v_{\min}^{out}) + h - A\tau^2 \cdot (v_{\min}^{out} + e - Lv_{\min}^2)$ ,  
 2)  $(\tau^1 - A\tau^2) \cdot (v^{out} - v_{\min}^{out}) = 0$  для всех  $v^{out} \in V^{out}$ ,  
 где вершина  $v_{\min}^{out} \in V^{out}$  определяется условием  $\tau^1 \cdot v_{\min}^{out} = \min_{v \in V^{out}} \tau^1 \cdot v$ .

**Доказательство.** Так как базовые вычислительные графы  $G^1$  и  $G^2$  пространственно совмещены, то для любой пары информационно связанных вершин  $v^{out} \in V^{out}$  и  $v^{in} \in V^{in}$  имеем:

$$\begin{aligned} t^2(v^{in}) &= \tau^2 \cdot (v^{in} - v_{\min}^2) + t_0^2 = A\tau^2 \cdot (Av^{in} - Av_{\min}^2) + t_0^2 = \\ &= A\tau^2 \cdot ((Av^{in} + b - e) + e - (Av_{\min}^2 + b)) + t_0^2 = A\tau^2 \cdot ((Lv^{in} - e) - \\ &\quad - Lv_{\min}^2 + e) + t_0^2 = A\tau^2 \cdot (v^{out} - Lv_{\min}^2 + e + v_{\min}^{out} - v_{\min}^{out}) + t_0^2 = \\ &\quad = A\tau^2 \cdot (v^{out} - v_{\min}^{out}) + A\tau^2 \cdot (v_{\min}^{out} + e - Lv_{\min}^2) + t_0^2; \\ t^1(v^{out}) &= \tau^1 \cdot (v^{out} - v_{\min}^1) + t_0^1 + \tau^1 \cdot v_{\min}^{out} - \tau^1 \cdot v_{\min}^{out} = \tau^1 \cdot (v^{out} - v_{\min}^{out}) + t^1(v_{\min}^{out}). \end{aligned}$$

Следовательно, условие временного согласования (2.18) в данном случае эквивалентно условию

$$\begin{aligned} A\tau^2 \cdot (v^{out} - v_{\min}^{out}) + A\tau^2 \cdot (v_{\min}^{out} + e - Lv_{\min}^2) + t_0^2 - \\ - \tau^1 \cdot (v^{out} - v_{\min}^{out}) - t^1(v_{\min}^{out}) &= h, \\ v^{out} \in V^{out}, h \geq 1, h \in \mathbf{Z}, \end{aligned}$$

которое можно переписать в виде

$$\begin{aligned} (\tau^1 - A\tau^2) \cdot (v^{out} - v_{\min}^{out}) &= \\ = t_0^2 + A\tau^2 \cdot (v_{\min}^{out} + e - Lv_{\min}^2) - t^1(v_{\min}^{out}) - h, &\quad (2.19) \\ v^{out} \in V^{out}, h \geq 1, h \in \mathbf{Z}. \end{aligned}$$

Пусть работа пространственно совмещенных базовых вычислительных графов  $G^1$  и  $G^2$  согласована по времени, т. е. выполняется условие (2.19). Положив в (2.19)  $v^{out} = v_{\min}^{out}$ , получим  $t_0^2 + A\tau^2 \cdot (v_{\min}^{out} + e - Lv_{\min}^2) - t^1(v_{\min}^{out}) - h = 0$ , откуда и следует равенство 1) теоремы ??

Обратно, пусть условия 1) и 2) теоремы 2.5 выполняются. Сложив равенства 1) и 2), получим равенство (2.19), которое в рассматриваемом случае пространственно совмещенных  $G^1$  и  $G^2$  эквивалентно условию временного согласования (2.18). Следовательно, работа  $G^1$  и  $G^2$  согласована по времени.

Доказательство теоремы 2.5 завершено.

**Замечание 2.2.** Если  $v_{\min}^2 \in V^{in}$ , то  $Lv_{\min}^2 = v_{\min}^{out} + e$  и условие 1) теоремы 2.5 упрощается:  $t_0^2 = t^1(v_{\min}^{out}) + h$ .

**Теорема 2.6.** Пусть базовые вычислительные графы  $G^1$  и  $G^2$  стыкованы, т. е. совмещены в пространстве и их работа согласована по времени,  $h = \tau^1 \cdot e$ . Тогда таймирование вершин пристыкованного графа  $G^2$  в зависимости от их расположения в области  $LV^2 = \{v \in \mathbf{Z}^d | v = L\hat{v}, \hat{v} \in V^2\}$  осуществляется с помощью функции

$$\begin{aligned} \hat{t}^2(v) = & A\tau^2 \cdot v + t_0^1 - \tau^1 \cdot v_{\min}^1 + \\ & + (\tau^1 - A\tau^2) \cdot (v_{\min}^{\text{out}} + e), v \in LV^2. \end{aligned} \quad (2.20)$$

**Доказательство.** Пусть  $v = L\hat{v}$ . Тогда  $\hat{t}^2(v) = t^2(\hat{v}) = \tau^2 \cdot (\hat{v} - v_{\min}^2) + t_0^2 = A\tau^2 \cdot (A\hat{v} - Av_{\min}^2 + b - b) + t_0^2 = A\tau^2 \cdot (v - Lv_{\min}^2) + t_0^2 = A\tau^2 \cdot v + t_0^2 - A\tau^2 \cdot Lv_{\min}^2$ . Согласно условию 1) теоремы ??,  $t_0^2 - A\tau^2 \cdot Lv_{\min}^2 = t^1(v_{\min}^{\text{out}}) + h - A\tau^2 \cdot (v_{\min}^{\text{out}} + e)$ , поэтому  $\hat{t}^2(v) = A\tau^2 \cdot v + t^1(v_{\min}^{\text{out}}) + h - A\tau^2 \cdot (v_{\min}^{\text{out}} + e) = A\tau^2 \cdot v + \tau^1 \cdot (v_{\min}^{\text{out}} - v_{\min}^1) + t_0^1 + \tau^1 \cdot e - A\tau^2 \cdot (v_{\min}^{\text{out}} + e) = A\tau^2 \cdot v + t_0^1 + \tau^1 \cdot (-v_{\min}^1) + \tau^1 \cdot v_{\min}^{\text{out}} + \tau^1 \cdot e - A\tau^2 \cdot (v_{\min}^{\text{out}} + e) = A\tau^2 \cdot v + t_0^1 - \tau^1 \cdot v_{\min}^1 + (\tau^1 - A\tau^2) \cdot (v_{\min}^{\text{out}} + e)$ . Доказательство теоремы 2.6 завершено.

**Следствие 2.1.** Пусть базовые вычислительные графы  $G^1$  и  $G^2$  стыкованы, т. е. совмещены в пространстве и их работа согласована по времени,  $h = \tau^1 \cdot e$ ,  $A\tau^2 = \tau^1$ . Тогда таймирование вершин пристыкованного графа  $G^2$  в области  $LV^2$  осуществляется с помощью функции  $\hat{t}^2(v) = t^1(v)$ .

В самом деле, положив в равенстве (2.20)  $A\tau^2 = \tau^1$ , получим  $\hat{t}^2(v) = \tau^1 \cdot v + t_0^1 - \tau^1 \cdot v_{\min}^1 + (\tau^1 - \tau^1) \cdot (v_{\min}^{\text{out}} + e) = t^1(v)$ . Таким образом, базовый вычислительный граф  $G = (V, E, T, H)$ , получаемый стыковкой  $G^1 = (V^1, E^1, T^1, H^1)$  и  $G^2 = (V^2, E^2, T^2, H^2)$ , задается следующим образом:

- 1)  $V = V^1 \cup LV^2$ ;
- 2)  $E = \{(v, v + \varphi), v \in V, \varphi \in \Phi\}$ ,  $\Phi = \Phi^1 \cup L\Phi^2 \cup \{e\}$ ;
- 3)  $T = \{t(v), v \in V\}$ ,  $t(v) = \begin{cases} t^1(v), & v \in V^1, \\ \hat{t}^2(v), & v \in LV^2; \end{cases}$
- 4)  $H$  – множество задержек  $\{h_\varphi, \varphi \in \Phi\}$ .

Описание функционирования  $P = P^1 \cup LP^2$ , где  $LP^2 = \{LP_\lambda^2(v), v \in LV_\lambda^2, 1 \leq \lambda \leq \Lambda\}$ , получается заменой в описании  $P_\lambda^2(v)$  векторов  $\varphi_2^{(k)}$  на их образы  $L\varphi_2^{(k)}$ .

## Набор базовых вычислительных графов для перемножения трех матриц

Пусть  $Y^2, X^2, X^1$  – квадратные матрицы порядка  $N$ . Задачу вычисления произведения трех матриц  $Y^3 = Y^2 \cdot X^2 \cdot X^1$  декомпозируем на две однотипные подзадачи  $X^3 = X^2 \cdot X^1$  и  $Y^3 = Y^2 \cdot Y^1$ , где  $Y^1 = X^3$ . Пусть произведение  $X^2 \cdot X^1$  реализуется на одном из базовых вычислительных графов  $G^\alpha$ ,  $1 \leq \alpha \leq 16$ , а произведение  $Y^2 \cdot Y^1$  – на одном из базовых вычислительных графов  $G^\beta$ ,  $1 \leq \beta \leq 16$ , приведенных в табл. 1.1. Наша ближайшая цель – получить БВГ для перемножения трех матриц стыковкой  $G^\alpha$  и  $G^\beta$ .

Легко проверить, что для каждого фиксированного числа  $\alpha$  стыковка  $G^\alpha$  и  $G^\beta$  возможна при значениях  $\beta = \beta_1$  и  $\beta = \beta_2$ , приведенных в табл. 2.4.

Таблица 2.4

**Возможные варианты стыковки**

$\beta \backslash \alpha$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\beta_1$	3	4	11	12	3	4	11	12	1	2	9	10	1	2	9	10
$\beta_2$	13	14	5	6	13	14	5	6	15	16	7	8	15	16	7	8

В самом деле, непосредственной подстановкой в условия а) и б) теоремы 2.4 проверяется, что пространственное совмещение  $G^\alpha$  и  $G^\beta$  осуществляется преобразованием

$$L^{\alpha, \beta} : v \rightarrow A^{\alpha, \beta} v + b^{\alpha, \beta}, \quad v \in V,$$

где

$$A^{\alpha, \beta_1} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, \quad A^{\alpha, \beta_2} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix};$$

$$b^{\alpha, \beta_1} = \begin{cases} (0, 0, 2N + 1)^T, & 1 \leq \alpha \leq 8, \\ (0, 0, 1)^T, & 9 \leq \alpha \leq 16; \end{cases}$$

$$b^{\alpha, \beta_2} = \begin{cases} (N + 1, 0, N)^T, & 1 \leq \alpha \leq 8, \\ (N + 1, 0, -N)^T, & 9 \leq \alpha \leq 16. \end{cases}$$

При этом  $L^{\alpha, \beta} \varphi_\beta^{(1)} = \varphi_\alpha^{(3)}$ ,  $L^{\alpha, \beta} \varphi_\beta^{(2)} = \varphi_\alpha^{(2)}$ ,  $L^{\alpha, \beta} \varphi_\beta^{(3)} = \varphi_\alpha^{(1)}$ .

Проверим, например, выполнение условия а) для случая  $\alpha = 1$ ,  $\beta = 3$ . Информационно связанные вершины  $v^{out}$  и  $v^{in}$  имеют координаты:  $v^{out} = (i, j, N)$  (в этих вершинах завершается вычисление элементов  $x_{ij}^3$  матрицы  $X^3$ ) и  $v^{in} = (N, j, i)$  (в эти вершины подаются элементы  $y_{ij}^1 = x_{ij}^3$  матрицы  $Y^1$ ). Имеем:

$$\begin{aligned} & A^{1,3}v^{in} + b^{1,3} - v^{out} = \\ & = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \begin{pmatrix} N \\ j \\ i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 2N+1 \end{pmatrix} - \begin{pmatrix} i \\ j \\ N \end{pmatrix} = \\ & = \begin{pmatrix} i \\ j \\ -N \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 2N+1 \end{pmatrix} - \begin{pmatrix} i \\ j \\ N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \end{aligned}$$

Таким образом,  $L^{1,3}v^{in} - v^{out} = (0, 0, 1) = e_3$ ,  $\Phi = \Phi^1 = \{e_1, e_2, e_3\}$ ,  $H = H^1$ .

Согласование работы по времени пространственно состыкованных базовых вычислительных графов  $G^1$  и  $G^2$  достигается ограничением на параметры таймирующих функций

$$t_0^\beta = t_0^\alpha + h_3^\alpha N, \quad h_2^\beta = h_2^\alpha, \quad h_3^\beta = h_1^\alpha,$$

где, напомним,  $\tau^\alpha \cdot \varphi_\alpha^{(k)} = h_k^\alpha \geq 1$ ,  $\alpha = 1, 2, \dots, 16$ ,  $k = 1, 2, 3$ . Например, для случая  $\alpha = 1$ ,  $\beta = 3$  из теоремы 2.5 имеем:

$$\begin{aligned} & 1) \quad t_0^3 = t^1(v_{\min}^{out}(1, 1, N)) + \tau^1 \cdot e_3 - A^{1,3}\tau^3 \cdot (v_{\min}^{out}(1, 1, N) + e_3 - \\ & \quad - L^{1,3}v_{\min}^3(N, 1, 1)) = \tau^1 \cdot ((1, 1, N) - (1, 1, 1)) + t_0^1 + h_3^1 - (\tau_3^3, \tau_2^3, - \\ & \quad - \tau_1^3) \cdot ((1, 1, N) + (0, 0, 1) - (1, 1, N + 1)) = \tau^1 \cdot (0, 0, N - 1) + t_0^1 + h_3^1 - \\ & \quad - (\tau_3^3, \tau_2^3, -\tau_1^3) \cdot (0, 0, 0) = (N - 1)h_3^1 + t_0^1 + h_3^1 = t_0^1 + h_3^1 N; \\ & 2) \quad (\tau^1 - A^{1,3}\tau^3) \cdot (v^{out}(i, j, N) - v_{\min}^{out}(1, 1, N)) = ((\tau_1^1, \tau_2^1, \tau_3^1) - (\tau_3^3, \tau_2^3, - \\ & \quad - \tau_1^3)) \cdot (i - 1, j - 1, 0) = (\tau_1^1 - \tau_3^3)(i - 1) + (\tau_2^1 - \tau_2^3)(j - 1) = (h_1^1 - \\ & \quad - h_3^3)(i - 1) + (h_2^1 - h_2^3)(j - 1) = 0, \text{ если } h_3^3 = h_1^1, \quad h_2^3 = h_2^1. \end{aligned}$$

Из теоремы 2.6 и ограничений  $h_2^\beta = h_2^\alpha$ ,  $h_3^\beta = h_1^\alpha$  следует, что таймирование состыкованных базовых вычислительных графов осуществляется функцией

$$t^{\alpha,\beta}(v) = t^\alpha(v) + \begin{cases} 0, & v \in V, \\ (h_1^\beta - h_3^\alpha)\varphi_\alpha^3 \cdot (v - L^{\alpha,\beta}v_{\min}^\beta), & v \in L^{\alpha,\beta}V, \end{cases}$$

а при  $h_1^\beta = h_3^\alpha$ ,  $h_2^\beta = h_2^\alpha$ ,  $h_3^\beta = h_1^\alpha$  возможно единое таймирование  $t^{\alpha,\beta}(v) = t^\alpha(v)$ .

Например, при  $\alpha = 1$ ,  $\beta = 3$ ,  $v \in L^{\alpha,\beta}V$  из теоремы 2.6 имеем:

$$\begin{aligned} t^{1,3}(v) &= A^{1,3}\tau^3 \cdot v + t_0^1 - \tau_1^1 \cdot (-v_{\min}^1) + (\tau^1 + A^{1,3}\tau^3) \cdot (v_{\min}^{out}(1, 1, N) + \\ &+ e_3) = (\tau_3^3, \tau_2^3, -\tau_1^3) \cdot v + t_0^1 - \tau_1^1 \cdot (-v_{\min}^1) + (\tau_1^1 - \tau_3^3, \tau_2^1 - \tau_2^3, \tau_3^1 + \\ &+ \tau_1^3) \cdot (1, 1, N + 1) = ((h_1^1, h_2^1, h_3^1) + (0, 0, h_1^3 - h_3^1)) \cdot v + t_0^1 - \\ &- \tau^1 \cdot (-v_{\min}^1) + (0, 0, h_3^1 - h_1^1) \cdot L^{1,3}v_{\min}^3 = t^1(v) + (h_1^3 - h_3^1)e_3 \cdot v - \\ &- (h_1^3 - h_3^1)e_3 \cdot L^{1,3}v_{\min}^3 = t^1(v) + (h_1^3 - h_3^1)e_3 \cdot (v - L^{1,3}v_{\min}^3). \end{aligned}$$

Таким образом, стыковкой  $G^\alpha$  и  $G^\beta$  получен набор базовых вычислительных графов  $G^{\alpha,\beta} = (V^{\alpha,\beta}, E^{\alpha,\beta}, t^{\alpha,\beta}, H^{\alpha,\beta})$ . Здесь  $V^{\alpha,\beta} = V \cup L^{\alpha,\beta}V$ ,  $E^{\alpha,\beta}$  определяется множеством векторов  $\Phi^{\alpha,\beta} = \Phi^\alpha$ , так как  $L^{\alpha,\beta}\Phi^\beta = \Phi^\alpha$ ,  $t^{\alpha,\beta}(v) = t^\alpha(v)$ ,  $H^{\alpha,\beta} = H^\alpha$ . Множество  $P^{\alpha,\beta}$  состоит из функциональных описаний вершин четырех типов  $P_1^\alpha(v)$ ,  $v \in V_1^\alpha$ ,  $P_2^\alpha(v)$ ,  $v \in V_2^\alpha$ ,  $L^{\alpha,\beta}P_1^\beta$ ,  $v \in L^{\alpha,\beta}V_1^\beta$ ,  $L^{\alpha,\beta}P_2^\beta$ ,  $v \in L^{\alpha,\beta}V_2^\beta$ , где  $L^{\alpha,\beta}P_\lambda^\beta(v)$  получается из  $P_\lambda^\beta(v)$  заменой векторов  $\varphi_\beta^{(k)}$  на их образы  $L^{\alpha,\beta}\varphi_\beta^{(1)} = \varphi_\alpha^{(3)}$ ,  $L^{\alpha,\beta}\varphi_\beta^{(2)} = \varphi_\alpha^{(2)}$ ,  $L^{\alpha,\beta}\varphi_\beta^{(3)} = \varphi_\alpha^{(1)}$  и заменой параметров задержки  $h_1^\beta$  на  $h_3^\alpha$ ,  $h_2^\beta$  на  $h_2^\alpha$ ,  $h_3^\beta$  на  $h_1^\alpha$ .

### Пример проектирования плоского спецпроцессора для перемножения трех квадратных матриц

Воспользуемся полученными результатами для проектирования плоского (2-мерного) спецпроцессора для перемножения трех квадратных матриц  $Y^3 = Y^2 \cdot X^2 \cdot X^1$  порядка  $N$ . Для этого возьмем  $\alpha = 16$  и  $\beta = 10$  (см. табл. 1.1). Пространственное совмещение БВГ  $G^\alpha$  и БВГ  $G^\beta$  осуществляется аффинным преобразованием:  $L^{16,10}: v \rightarrow A^{16,10}v + b^{16,10}$ ,  $v \in V = \{(i_1, i_2, i_3) \in \mathbf{Z}^3 \mid 1 \leq i_1, i_2, i_3 \leq N\}$ :

$$A^{16,10} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}, b^{16,10} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Если обозначить

$$\begin{aligned} V_N &= \{(i_1, i_2, i_3) \in \mathbf{Z}^3 \mid 1 \leq i_1, i_2 \leq N, i_3 = N\}, \\ V_{\overline{N}} &= \{(i_1, i_2, i_3) \in \mathbf{Z}^3 \mid 1 \leq i_1, i_2 \leq N, 1 \leq i_3 < N\}, \end{aligned}$$

то «стыкованный» алгоритм перемножения трех квадратных матриц можно записать в виде системы рекуррентных уравнений (1.1):

$$\begin{aligned}
v \in V \cup L^{16,10}V &= \{ (i_1, i_2, i_3) \in \mathbf{Z}^3 \mid 1 \leq i_1, i_2 \leq N, \\
&\quad -(N-1) \leq i_3 \leq N \}; \\
x^{(1)}(v) &= \begin{cases} x^{(1)}(v + e_1), & v \in V, \\ x^{(2)}(v + e_2)x^{(3)}(v + e_3), & v \in L^{16,10}V_N, \\ x^{(2)}(v + e_2)x^{(3)}(v + e_3) + x^{(1)}(v + e_1), & v \in L^{16,10}V_{\overline{N}}; \end{cases} \quad (2.21) \\
x^{(2)}(v) &= x^{(2)}(v + e_2), \quad v \in V \cup L^{16,10}V; \\
x^{(3)}(v) &= \begin{cases} x^{(2)}(v + e_2)x^{(1)}(v + e_1), & v \in V_N, \\ x^{(2)}(v + e_2)x^{(1)}(v + e_1) + x^{(3)}(v + e_3), & v \in V_{\overline{N}}, \\ x^{(3)}(v + e_3), & v \in L^{16,10}V. \end{cases}
\end{aligned}$$

Результат перемножения матриц определяется равенствами:

$$y_{ij}^3 = x^{(1)}(1, j, -i + 1), \quad 1 \leq i, j \leq N.$$

Начальные значения, элементы перемножаемых матриц, присваиваются соответствующим переменным:

$$\begin{aligned}
x_{ij}^1 &= x^{(1)}(N + 1, j, N - i + 1), \\
x_{ij}^2 &= x^{(2)}(i, N + 1, N - j + 1), \\
y_{ij}^2 &= x^{(2)}(j, N + 1, -i + 1), \\
&\quad 1 \leq i, j \leq N.
\end{aligned}$$

Если «стыковочный» вектор  $e$  в теореме 2.4 положить равным  $\varphi_\alpha^{(3)} = \varphi_{16}^{(3)} = -e_3$ , то алгоритм (2.21) распараллеливается единой таймирующей функцией  $t^{(16,10)}(v) = t^{(16)}(v) = -(i_1 + i_2 + i_3) + 3N + 1$ ,  $v = (i_1, i_2, i_3) \in V \cup L^{(16,10)}V$ , с единичными задержками по всем направлениям, определяемым векторами зависимостей  $\Phi^{16} = \{-e_1, -e_2, -e_3\}$ .

Выберем в качестве оператора отображения  $\pi: \mathbf{Z}^3 \rightarrow \mathbf{Z}^2$  оператор проецирования на плоскость  $i_3 = 0$ .

Вычислим образы:

$$\begin{aligned}
\pi(-e_1) &= (-1, 0), \quad \pi(-e_2) = (0, -1), \quad \pi(-e_3) = (0, 0); \\
\pi L^{16,10}V_N &= \{ (x, y) \in \mathbf{Z}^2 \mid x = N, 1 \leq y \leq N \} = \tilde{V}_1; \\
\pi L^{16,10}V_{\overline{N}} &= \{ (x, y) \in \mathbf{Z}^2 \mid 1 \leq x < N, 1 \leq y \leq N \} = \tilde{V}_2; \\
\pi V_N = \pi V_{\overline{N}} &= \{ (x, y) \in \mathbf{Z}^2 \mid 1 \leq x, y \leq N \} = \tilde{V}_1 \cup \tilde{V}_2.
\end{aligned}$$

Итак, проектируемый матричный специализированный процессор содержит два типа процессорных элементов, расположенных соответственно в областях  $\tilde{V}_1$  и  $\tilde{V}_2$ . Каждый процессорный элемент работает в трех режимах, выполняя три типа операций, приписанных вершинам-прообразам.

Процессорные элементы  $\text{П}\tilde{\Theta}_1(\tilde{v})$ ,  $\tilde{v} \in \tilde{V}_1$ , выполняют операции алгоритма (2.21), соответствующие вершинам множеств  $V_N$ ,  $V_{\overline{N}}$  и  $L^{16,10}V_N$ , а процессорные элементы  $\text{П}\tilde{\Theta}_2(\tilde{v})$ ,  $\tilde{v} \in \tilde{V}_2$ , выполняют операции алгоритма (2.21), соответствующие вершинам множеств  $V_N$ ,  $V_{\overline{N}}$  и  $L^{16,10}V_{\overline{N}}$ . Таким образом, выполняется необходимое условие (2.15) введения управления. Для того чтобы пометить каждую вершину области  $V_N$  меткой  $m_1$ ,  $V_{\overline{N}}$  – меткой  $m_2$ , а вершины области  $L^{16,10}V_N$  и  $L^{16,10}V_{\overline{N}}$  – меткой  $m_3$ , дополним алгоритм перемножения трех квадратных матриц алгоритмом вида (2.16) транспортировки управляющих меток  $m(v) = m(v + e_2)$ ,  $v = (i_1, i_2, i_3) \in V \cup L^{16,10}V$ , присвоив переменной  $m$  в граничных точках начальные значения:

$$m(i_1, N + 1, i_3) = \begin{cases} m_1, & i_3 = N, 1 \leq i_1 \leq N, \\ m_2, & 1 \leq i_3 < N, 1 \leq i_1 \leq N, \\ m_3, & -(N - 1) \leq i_3 \leq 0, 1 \leq i_1 \leq N. \end{cases}$$

На рис. 2.6 изображен ввод начальных данных и управляющих меток, а также вывод результатов в БВГ алгоритма (2.21) при  $N = 3$ , а на рис. 2.7 – структурная схема матричного спецпроцессора при  $N = 3$ .

Приведем структурную схему процессорных элементов первого и второго типов и опишем функционирование их локальной памяти, которая представлена четырьмя ячейками для хранения переменных  $x^{(1)}$ ,  $x^{(2)}$ ,  $x^{(3)}$  и  $m$ . Процессорные элементы первого типа  $\text{П}\tilde{\Theta}_1(\tilde{v})$  расположены в области  $\tilde{V}_1 = \{ (x, y) \in \mathbf{Z}^2 \mid x = N, 1 \leq y \leq N \}$ , а процессорные элементы второго типа  $\text{П}\tilde{\Theta}_2(\tilde{v})$  – в области  $\tilde{V}_2 = \{ (x, y) \in \mathbf{Z}^2 \mid 1 \leq x < N, 1 \leq y \leq N \}$ . Оба типа процессорных элементов имеют по три входа  $I_{(-1,0)}$ ,  $I_{(0,-1)}$ ,  $I_{(0,-1)}^{(m)}$ , по три выхода  $O_{(-1,0)}$ ,  $O_{(0,-1)}$ ,  $O_{(0,-1)}^{(m)}$  и по четыре ячейки локальной памяти  $R^{(1)}$ ,  $R^{(2)}$ ,  $R^{(3)}$ ,  $R^{(m)}$  для задержки поступивших данных на один такт, в течение которого выполняются операции алгоритма.

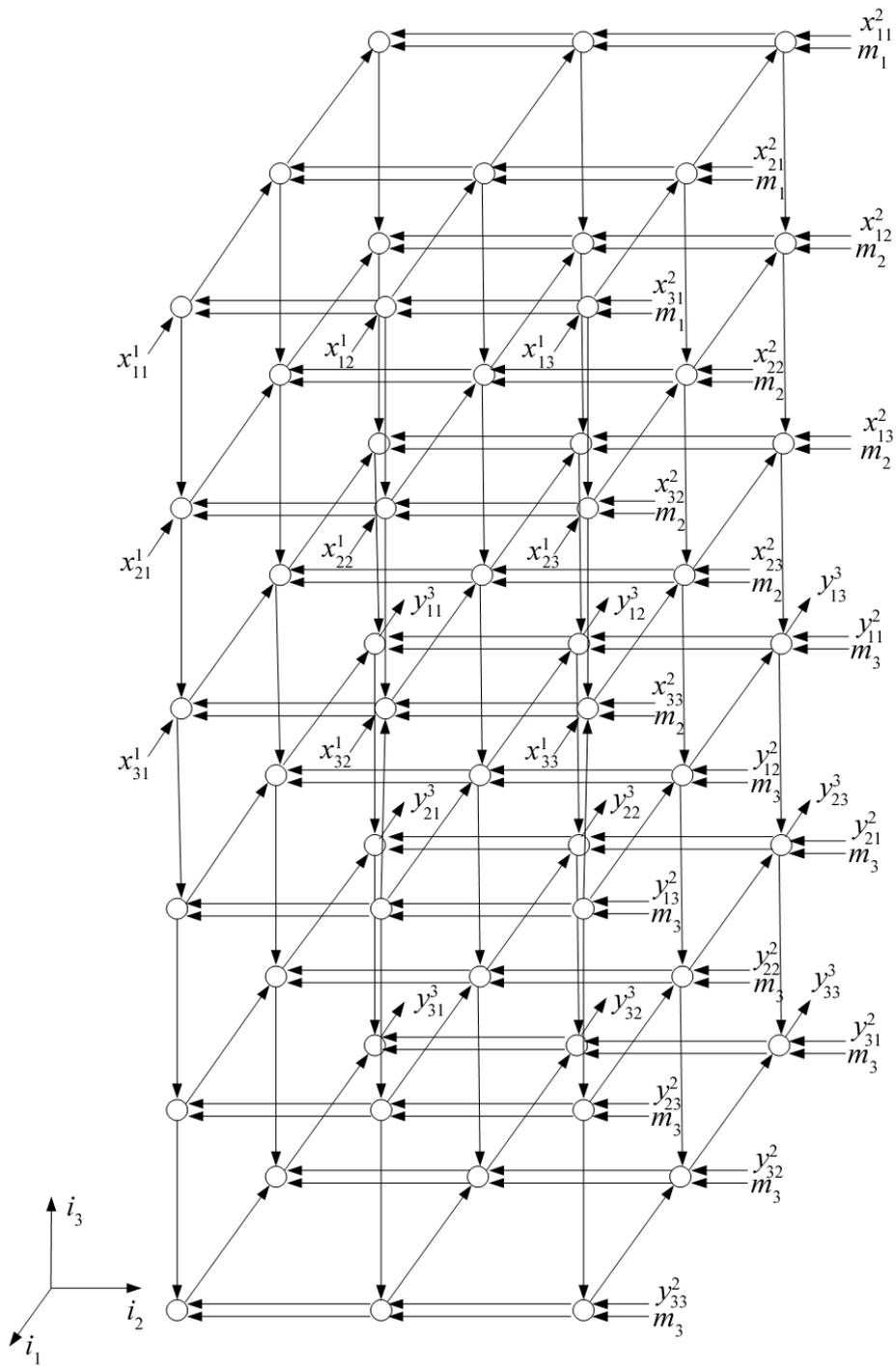


Рис. 2.6. Ввод начальных данных и управляющих меток, а также вывод результатов в БВГ алгоритма (2.21) при  $N = 3$

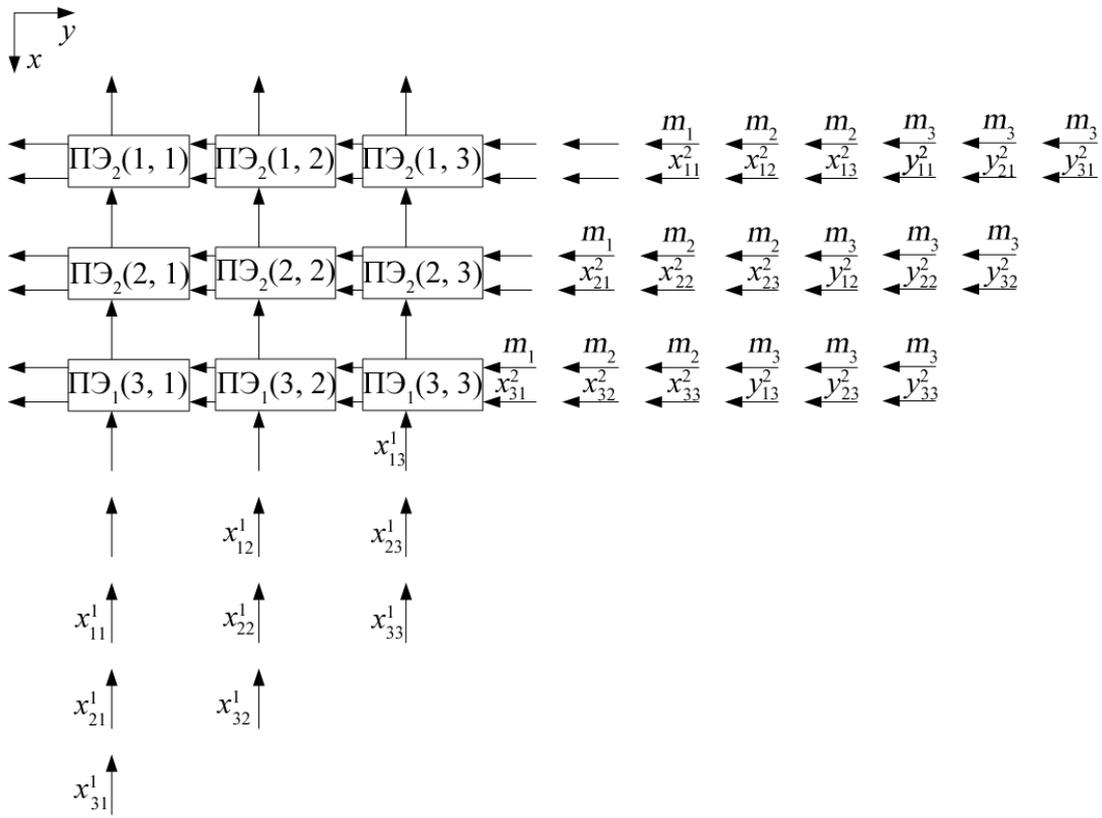
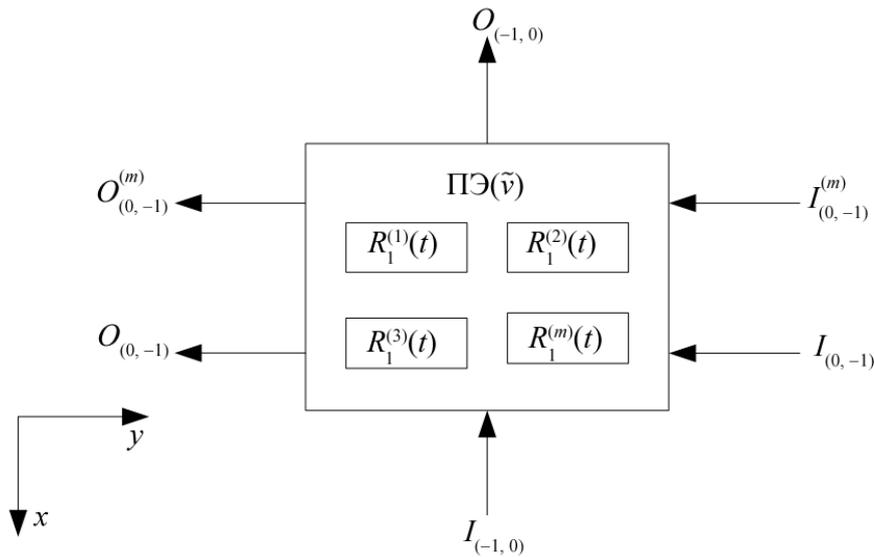


Рис. 2.7. Структурная схема спецпроцессора для перемножения трех квадратных матриц порядка  $N = 3$



Режим ввода начальных данных и управляющих меток в спецпроцессор и вывода результатов приведен в табл. 2.5.

Таблица 2.5

**Ввод начальных данных и управляющих меток и вывода результатов спецпроцессора для перемножения трех квадратных матриц**

Данные	$v_{in}$	$\text{ПЭ}(\pi(v_{in}))$	Порт	$t(v_{in})$	Изменение параметров
$m_1$	$(i_1, N, N)$	$\text{ПЭ}(i_1, N)$	$I_{(0,-1)}^{(m)}$	$N + 1 - i_1$	$1 \leq i_1 \leq N$
$m_2$	$(i_1, N, i_3)$	$\text{ПЭ}(i_1, N)$	$I_{(0,-1)}^{(m)}$	$2N + 1 - i_1 - i_3$	$1 \leq i_1 \leq N$ $1 \leq i_3 \leq N - 1$
$m_3$	$(i_1, N, i_3)$	$\text{ПЭ}(i_1, N)$	$I_{(0,-1)}^{(m)}$	$2N + 1 - i_1 - i_3$	$1 \leq i_1 \leq N$ $-(N - 1) \leq i_3 \leq 0$
$x_{\alpha\beta}^1$	$(N, \beta, N - \alpha + 1)$	$\text{ПЭ}(N, \beta)$	$I_{(-1,0)}$	$N + \alpha - \beta$	$1 \leq \alpha, \beta \leq N$
$x_{\alpha\beta}^2$	$(\alpha, N, N - \beta + 1)$	$\text{ПЭ}(\alpha, N)$	$I_{(0,-1)}$	$N + \beta - \alpha$	$1 \leq \alpha, \beta \leq N$
$y_{\alpha\beta}^2$	$(\beta, N, -\alpha + 1)$	$\text{ПЭ}(\beta, N)$	$I_{(0,-1)}$	$2N + \alpha - \beta$	$1 \leq \alpha, \beta \leq N$
Результат	$v_{out}$	$\text{ПЭ}(\pi(v_{out}))$	Порт	$t(v_{out})$	Изменение параметров
$y_{\alpha\beta}^3$	$(1, \beta, -\alpha + 1)$	$\text{ПЭ}(1, \beta)$	$O_{(-1,0)}$	$3N + \alpha - \beta - 1$	$1 \leq \alpha, \beta \leq N$

Описание функционирования локальной памяти  $\text{ПЭ}_1(\tilde{v})$ ,  $\tilde{v} \in \tilde{V}_1$ , в любой момент времени  $t$  имеет вид:

$$P_1(\tilde{v}) : \{ R_1^{(m)}(t) = I_{(0,-1)}^{(m)}, O_{(0,-1)}^{(m)} = R^{(m)}(t);$$

$$R^{(1)}(t) = \begin{cases} I_{(-1,0)}, & \text{если } I_{(0,-1)}^{(m)} = m_1 \vee m_2, \\ I_{(0,-1)} \cdot R^{(3)}(t-1), & \text{если } I_{(0,-1)}^{(m)} = m_3, \end{cases}$$

$$O_{(-1,0)} = R^{(1)}(t);$$

$$R^{(2)}(t) = I_{(0,-1)}, O_{(0,-1)} = R^{(2)}(t);$$

$$R^{(3)}(t) = \begin{cases} I_{(0,-1)} \cdot I_{(-1,0)}, & \text{если } I_{(0,-1)}^{(m)} = m_1, \\ I_{(0,-1)} \cdot I_{(-1,0)} + R^{(3)}(t-1), & \text{если } I_{(0,-1)}^{(m)} = m_2, \\ R^{(3)}(t-1), & \text{если } I_{(0,-1)}^{(m)} = m_3. \end{cases}$$

Описание функционирования локальной памяти  $\text{ПЭ}_2(\tilde{v})$ ,  $\tilde{v} \in \tilde{V}_2$ , в любой момент времени  $t$  имеет вид:

$$\begin{aligned}
P_2(\tilde{v}) : \{ & R_1^{(m)}(t) = I_{(0,-1)}^{(m)}, \quad O_{(0,-1)}^{(m)} = R^{(m)}(t); \\
& R^{(1)}(t) = \begin{cases} I_{(-1,0)}, & \text{если } I_{(0,-1)}^{(m)} = m_1 \vee m_2, \\ I_{(0,-1)} \cdot R^{(3)}(t-1) + I_{(-1,0)}, & \text{если } I_{(0,-1)}^{(m)} = m_3, \end{cases} \\
& O_{(-1,0)} = R^{(1)}(t); \\
& R^{(2)}(t) = I_{(0,-1)}, \quad O_{(0,-1)} = R^{(2)}(t); \\
& R^{(3)}(t) = \begin{cases} I_{(0,-1)} \cdot I_{(-1,0)}, & \text{если } I_{(0,-1)}^{(m)} = m_1, \\ I_{(0,-1)} \cdot I_{(-1,0)} + R^{(3)}(t-1), & \text{если } I_{(0,-1)}^{(m)} = m_2, \\ R^{(3)}(t-1), & \text{если } I_{(0,-1)}^{(m)} = m_3 \end{cases}.
\end{aligned}$$

Таким образом спроектированный спецпроцессор перемножает три квадратные матрицы порядка  $N$  за  $4N - 2$  такта.

**Замечание 2.3.** Если на входы  $I_{(-1,0)}$  ПЭ<sub>1</sub>( $\tilde{v}$ ) вслед за элементами  $x_{\alpha\beta}^1$ ,  $1 \leq \alpha, \beta \leq N$ , подавать нули, то можно в спроектированном спецпроцессоре ограничиться только ПЭ<sub>2</sub>( $\tilde{v}$ ), заменив ПЭ<sub>1</sub>( $\tilde{v}$ ) на ПЭ<sub>2</sub>( $\tilde{v}$ ).

**Замечание 2.4.** Если вместо элементов  $y_{\alpha\beta}^2$ ,  $1 \leq \alpha, \beta \leq N$ , на входы  $I_{(0,-1)}$  подавать элементы единичной матрицы, то спроектированный спецпроцессор выполнит произведение двух матриц  $X^2$  и  $X^1$  за  $4N - 2$  такта.

## 2.6. ПАРАЛЛЕЛЬНЫЕ ФОРМЫ ЗАДАННОЙ ШИРИНЫ

Число процессорных элементов матричного спецпроцессора на СБИС ограничено стоимостью изготовления, числом портов ввода (вывода) и некоторыми другими технологическими факторами. Поэтому необходимо уметь получать параллельные формы алгоритмов, пригодные для реализации на матричных систолических процессорах с фиксированным числом процессорных элементов. Возникает задача построения параллельных форм заданной (фиксированной) ширины.

Существуют две основные стратегии построения параллельных алгоритмов фиксированной ширины для реализации на матричных спецпроцессорах с заданным числом процессорных элементов: локально последовательная глобально параллельная (ЛПосГПар) и локально параллельная глобально последовательная (ЛПарГПос).

Согласно этим стратегиям, алгоритм (или граф зависимостей алгоритма) решения исходной задачи разбивается на части с последующим пространственно-временным отображением. В случае применения ЛПосГПар стратегии каждая полученная часть отображается в одну вершину вычислительного графа, в связи с чем число вершин совпадает с числом частей алгоритма. В случае применения ЛПарГПос стратегии полученные части последовательно отображаются на один и тот же вычислительный граф с заданным числом вершин. К достоинствам ЛПарГПос стратегии можно отнести то, что локальная память процессорных элементов соответствующего матричного спецпроцессора не зависит от того, на сколько частей разбивается граф алгоритма, а может зависеть только от размера задачи. К недостаткам относится наличие глобальных обратных связей, а также невозможность производить разбиения, в которых какие-либо две части графа алгоритма информационно взаимно зависимы друг от друга. ЛПосГПар стратегия позволяет для решения исходной задачи получать модели МСП без обратных связей, но требует дополнительной локальной памяти процессорных элементов.

Как и раньше будем предполагать, что исходный алгоритм задан в виде системы рекуррентных уравнений вида (1.1), а граф зависимостей  $G$  алгоритма строго направлен, т. е. что существует вектор  $\tau = (\tau_1, \dots, \tau_a)$ , образующий острые углы со всеми векторами из множества  $\Phi$ , характеризующего дуги графа:

$$\tau \cdot \varphi \geq 1, \varphi \in \Phi. \quad (2.22)$$

### **Применение локально параллельной глобально последовательной стратегии разбиения**

Предлагаемые в этом и следующем параграфе методы ориентированы на получение вычислительных графов, являющихся моделями наиболее удобных для практической реализации линейных (одномерных) матричных спецпроцессоров. Отметим, что в случае применения ЛПарГПос стратегии для получения модели линейного матричного спецпроцессора глобальные межсоединения возникают только между крайними процессорными элементами, поэтому от глобальности можно

избавиться путем такой укладки процессорных элементов на кристалле, при которой крайние элементы оказываются рядом.

Пусть  $\Delta$  — заданное целое положительное число (заданная ширина алгоритма),  $\pi: \mathbf{Z}^d \rightarrow \mathbf{Z}$  — невырожденное линейное отображение, удовлетворяющее условию

$$\pi(\varphi) \in \{(0), (1)\} \text{ для всех } \varphi \in \Phi, \quad (2.23)$$

причем существует хотя бы один вектор  $\varphi \in \Phi$ , для которого  $\pi(\varphi) = 1$ . Ненулевые элементы матрицы-строки  $\pi$  взаимно просты.

Обозначим

$$\begin{aligned} \delta_{\min} &= \min_{v \in V} \pi(v), \quad \delta_{\max} = \max_{v \in V} \pi(v), \\ \Gamma &= (\delta_{\max} - \delta_{\min} + 1) / \Delta. \end{aligned}$$

Не ограничивая общности, можно предполагать, что  $\Gamma$  — целое число. В противном случае следует расширить область вычислений  $V$  и положить

$$\Gamma = \lceil (\delta_{\max} - \delta_{\min} + 1) / \Delta \rceil, \text{ где } \lceil a \rceil = \min\{z \in \mathbf{Z} \mid z \geq a\}.$$

Разобьем множество вершин  $V$  на подмножества

$$U_\gamma = \{v \in V \mid \delta_{\min} + (\gamma - 1)\Delta \leq \pi(v) \leq \delta_{\min} + \gamma\Delta - 1\}, \quad \gamma = 1, 2, \dots, \Gamma.$$

Функцию размещения  $f: V \rightarrow \mathbf{Z}$ , задающую для каждой вершины  $v \in V$  местоположение вершины (процессорного элемента) вычислительного графа, выполняющей приписанное вершине  $v$  вычисление, определим следующим образом:

$$f(v) = \sum_{\delta=1}^{\Delta} \delta 1_{U^\delta}(v),$$

где

$$1_W(v) = \begin{cases} 1, & \text{если } v \in W, \\ 0, & \text{если } v \notin W; \end{cases}$$

$$U^\delta = \bigcup_{\gamma=1}^{\Gamma} U_\gamma^\delta;$$

$$U_\gamma^\delta = \{v \in U_\gamma \mid \pi(v) = \delta_{\min} + (\gamma - 1)\Delta + \delta - 1\}.$$

При пространственном отображении графа зависимостей  $G$  посредством функции  $f$  множество  $V$  отобразится во множество  $\{1, 2, \dots, \Delta\}$ , дуги  $(v, v + \varphi)$  отобразятся в дуги  $(f(v), f(v + \varphi))$ .

**Теорема 2.7.** Пусть  $v \in U_{\gamma_1}^{\delta_1}$ ,  $v + \varphi \in U_{\gamma_2}^{\delta_2}$ ,  $\varphi \in \Phi$ ,  $1 \leq \gamma_1, \gamma_2 \leq \Gamma$ ,  $1 \leq \delta_1, \delta_2 \leq \Delta$ . Тогда либо  $\gamma_1 = \gamma_2$ ,  $\delta_1 = \delta_2$ , либо  $\gamma_1 = \gamma_2$ ,  $\delta_2 = \delta_1 + 1$ , либо  $\gamma_2 = \gamma_1 + 1$ ,  $\delta_1 = \Delta$ ,  $\delta_2 = 1$ . Дуги  $(f(v), f(v + \varphi))$  определяются одномерными векторами с координатой

$$f(v + \varphi) - f(v) = \begin{cases} 0, & \text{если } \gamma_1 = \gamma_2, \delta_1 = \delta_2, \\ 1, & \text{если } \gamma_1 = \gamma_2, \delta_2 = \delta_1 + 1, \\ 1 - \Delta, & \text{если } \gamma_2 = \gamma_1 + 1, \delta_1 = \Delta, \delta_2 = 1. \end{cases} \quad (2.24)$$

**Доказательство.** Так как  $v \in U_{\gamma_1}^{\delta_1}$ ,  $v + \varphi \in U_{\gamma_2}^{\delta_2}$ , то

$$\pi(v) = \delta_{\min} + \delta_1 + (\gamma_1 - 1)\Delta - 1,$$

$$\pi(v + \varphi) = \delta_{\min} + \delta_2 + (\gamma_2 - 1)\Delta - 1,$$

$$\pi(\varphi) = \pi(v + \varphi) - \pi(v) = \delta_2 - \delta_1 + (\gamma_2 - \gamma_1)\Delta,$$

где  $|\delta_2 - \delta_1| \leq \Delta - 1$ .

Возможны два случая:

а)  $\pi(\varphi) = 0$ . Тогда  $\delta_2 - \delta_1 + (\gamma_2 - \gamma_1)\Delta = 0$ , что возможно только если  $\gamma_1 = \gamma_2$ ,  $\delta_1 = \delta_2$  (так как  $|\delta_2 - \delta_1| \leq \Delta - 1$ );

б)  $\pi(\varphi) = 1$ . Тогда  $\delta_2 - \delta_1 + (\gamma_2 - \gamma_1)\Delta = 1$ , поэтому либо  $\gamma_1 = \gamma_2$ ,  $\delta_2 = \delta_1 + 1$ , либо  $\gamma_2 = \gamma_1 + 1$ ,  $\delta_1 = \Delta$ ,  $\delta_2 = 1$ .

Формула (2.24) есть простое следствие определения функции размещения  $f$  и полученной связи  $\gamma_1$  и  $\gamma_2$ ,  $\delta_1$  и  $\delta_2$ . Теорема доказана.

Из теоремы 2.7 следует, что связи между вершинами вычислительного графа порождаются только дугами, соединяющими вершины множества  $U_{\gamma}^{\delta}$  и множества  $U_{\gamma}^{\delta+1}$ , а также вершины множества  $U_{\gamma}^{\Delta}$  и множества  $U_{\gamma+1}^1$ , причем в последнем случае соединения глобальные (длины  $\Delta - 1$ ).

Определим таймирующую функцию равенством

$$t(v) = \sum_{\gamma=1}^{\Gamma} (c_{\gamma} + \tau \cdot v) 1_{U_{\gamma}}(v), \quad (2.25)$$

где константы  $c_\gamma$ ,  $1 \leq \gamma \leq \Gamma$ , и направляющий вектор  $\tau$  выбираются из условий (2.22) и следующих неравенств:

$$t(v + \varphi) > t(v), \quad v, v + \varphi \in V, \quad \varphi \in \Phi, \quad (2.26)$$

что означает непротиворечивость таймирования вершин графа  $G$ :

$$t(v_1) \neq t(v_2), \quad \text{если } v_1, v_2 \in U_\gamma^\delta, \quad 1 \leq \gamma \leq \Gamma, \quad 1 \leq \delta \leq \Delta, \quad v_1 \neq v_2 \quad (2.27)$$

(всем вершинам из области  $U_\gamma^\delta$ , отображаемым в одну и ту же вершину  $(\delta)$  вычислительного графа, приписывается разное время выполнения соответствующих им операций);

$$\begin{aligned} t(v_2) > t(v_1), \quad \text{если } v_1 \in U_{\gamma_1}^\delta, \quad v_2 \in U_{\gamma_2}^\delta, \\ 1 \leq \gamma_1 < \gamma_2 \leq \Gamma, \quad 1 \leq \delta \leq \Delta \end{aligned} \quad (2.28)$$

(операции, соответствующие вершинам из  $U_{\gamma_1}^\delta$ , выполняются в вершине  $(\delta)$  вычислительного графа раньше операций, соответствующих вершинам из  $U_{\gamma_2}^\delta$ , если  $\gamma_1 < \gamma_2$ ).

**Теорема 2.8.** *Ограничение (2.26) на таймирующую функцию вида (2.25) выполняется тогда и только тогда, когда выполняется условие (2.22) и неравенства*

$$c_{\gamma+1} - c_\gamma \geq -x_\gamma, \quad 1 \leq \gamma \leq \Gamma - 1, \quad (2.29)$$

где

$$x_\gamma = \min_{\varphi \in \Phi_\gamma} \tau \cdot \varphi - 1, \quad \Phi_\gamma = \{ \varphi \in \Phi \mid \varphi = v_2 - v_1, \quad v_1 \in U_\gamma^\Delta, \quad v_2 \in U_{\gamma+1}^1 \}.$$

**Доказательство.** Если  $v$  и  $v + \varphi$  принадлежат какому-либо одному множеству  $U_\gamma$ , то неравенства (2.22) и (2.26) эквивалентны ввиду  $t(v + \varphi) - t(v) = \tau \cdot \varphi$ . Если  $v$  и  $v + \varphi$  принадлежат разным множествам, то  $v \in U_\gamma^\Delta$ ,  $v + \varphi \in U_{\gamma+1}^1$  (см. теорему 2.7),  $\varphi \in \Phi_\gamma$ , и следующие неравенства эквивалентны:  $t(v + \varphi) - t(v) \geq 1$ ,  $c_{\gamma+1} + \tau \cdot (\varphi + v) - (c_\gamma + \tau \cdot v) \geq 1$ ,  $c_{\gamma+1} - c_\gamma \geq 1 - \tau \cdot \varphi$ ,  $c_{\gamma+1} - c_\gamma \geq -x_\gamma$ . Теорема 2.8 доказана.

**Теорема 2.9.** *Ограничение (2.27) на таймирующую функцию вида (2.25) выполняется тогда и только тогда, когда*

$$\tau \cdot s \neq 0 \quad \text{для } s \in S = \bigcup_{\gamma=1}^{\Gamma} ((U_\gamma \ominus U_\gamma) \cap \text{Кер}\pi) \setminus \{0\}, \quad (2.30)$$

где  $U_\gamma \ominus U_\gamma = \{v \in Z^d \mid v = v_1 - v_2, v_1, v_2 \in U_\gamma\}$ .

**Доказательство.** Пусть выполняется условие (2.27). Тогда  $t(v_2) - t(v_1) = \tau \cdot (v_2 - v_1) \neq 0$  для таких  $v_1, v_2 \in U_\gamma^\delta$ ,  $1 \leq \gamma \leq \Gamma$ , для которых

$$\begin{aligned}\pi(v_2) &= \pi(v_1) = \delta_{\min} + (\gamma - 1)\Delta + \delta - 1, \\ \pi(v_2) - \pi(v_1) &= \pi(v_2 - v_1) = 0.\end{aligned}$$

Следовательно, условие  $\tau \cdot s \neq 0$  выполнено для  $s \in S$ .

Обратно, пусть выполнено условие (2.30). Тогда  $t(v_2) - t(v_1) = \tau \cdot (v_2 - v_1) \neq 0$  для  $v_1, v_2 \in U_\gamma^\delta$ , так как  $v_2 - v_1 \in U_\gamma \ominus U_\gamma$  и  $v_2 - v_1 \in \text{Ker}\pi$  ( $\pi(v_2 - v_1) = \pi(v_2) - \pi(v_1) = \delta_{\min} + (\gamma - 1)\Delta + \delta - 1 - (\delta_{\min} + (\gamma - 1)\Delta + \delta - 1) = 0$ ). Условие (2.27) выполнено. Теорема 2.9 доказана.

**Теорема 2.10.** Ограничение (2.28) на таймирующую функцию вида (2.25) выполняется тогда и только тогда, когда

$$c_{\gamma+1} - c_\gamma \geq y_\gamma, \quad 1 \leq \gamma \leq \Gamma - 1, \quad (2.31)$$

где

$$y_\gamma = \max_{1 \leq \delta \leq \Delta} \left\{ \max_{v \in U_\gamma^\delta} \tau \cdot v - \min_{v \in U_{\gamma+1}^\delta} \tau \cdot v \right\} + 1.$$

**Доказательство.** Пусть условие (2.28) выполняется. Тогда оно выполняется и для  $\gamma_2 = \gamma + 1$ ,  $\gamma_1 = \gamma$ ,  $1 \leq \gamma \leq \Gamma - 1$ , т. е.

$$t(v_2) - t(v_1) = c_{\gamma+1} + \tau \cdot v_2 - \tau \cdot v_1 - c_\gamma = c_{\gamma+1} - c_\gamma + \tau \cdot v_2 - \tau \cdot v_1 \geq 1$$

для всех  $v_1 \in U_\gamma^\delta$ ,  $v_2 \in U_{\gamma+1}^\delta$ , в частности, для  $v_1$  и  $v_2$ , таких, что

$$\tau \cdot v_1 - \tau \cdot v_2 = \max_{1 \leq \delta \leq \Delta} \left\{ \max_{v_1 \in U_\gamma^\delta} \tau \cdot v_1 - \min_{v_2 \in U_{\gamma+1}^\delta} \tau \cdot v_2 \right\} = y_\gamma - 1.$$

Следовательно,  $c_{\gamma+1} - c_\gamma \geq y_\gamma$ , условие (2.31) выполнено.

Обратно, пусть выполнено условие (2.31).

а) Если  $\gamma_2 = \gamma_1 + 1 = \gamma + 1$ , то для  $v_1 \in U_\gamma^\delta$ ,  $v_2 \in U_{\gamma+1}^\delta$ , имеем  $t(v_2) - t(v_1) = c_{\gamma+1} - c_\gamma + \tau \cdot v_2 - \tau \cdot v_1 \geq y_\gamma + \tau \cdot v_2 - \tau \cdot v_1 = 1 + \max_{1 \leq \delta \leq \Delta} \left\{ \max_{v \in U_\gamma^\delta} \tau \cdot v - \min_{v \in U_{\gamma+1}^\delta} \tau \cdot v \right\} + \tau \cdot v_2 - \tau \cdot v_1 \geq 1 + \max_{v \in U_\gamma^\delta} \tau \cdot v - \min_{v \in U_{\gamma+1}^\delta} \tau \cdot v + \tau \cdot v_2 - \tau \cdot v_1 \geq 1$ , условие (2.28) выполнено.

б) Если  $\gamma_2 > \gamma_1 + 1$ , то выберем  $v^{(1)} \in U_{\gamma_1+1}^\delta$ ,  $v^{(2)} \in U_{\gamma_1+2}^\delta, \dots, v^{(\gamma_2-\gamma_1-1)} \in U_{\gamma_1+(\gamma_2-\gamma_1-1)}^\delta = U_{\gamma_2-1}^\delta$ . С учетом а) имеем  $t(v_2) > t(v^{(\gamma_2-\gamma_1-1)}) > \dots > t(v^{(1)}) > t(v_1)$ .

**Теорема 2.11.** При фиксированном  $\tau$  минимальная высота параллельной формы алгоритма достигается при  $c_1, c_2, \dots, c_\Gamma$ , задаваемых рекуррентными соотношениями

$$c_{\gamma+1} = c_\gamma + \max\{-x_\gamma, y_\gamma\}, \quad 1 \leq \gamma \leq \Gamma - 1, \quad (2.32)$$

$c_1$  выбирается произвольно.

**Доказательство.** Пусть  $|T| = \max_{v \in V} t(v) - \min_{v \in V} t(v) + 1$  — высота параллельной формы алгоритма. Имеем:

$$\begin{aligned} |T| &= \max_{v \in V} t(v) - \min_{v \in V} t(v) + 1 = \max_{v \in U_\Gamma} (c_\Gamma + \tau \cdot v) - \min_{v \in U_1} (c_1 + \tau \cdot v) + 1, \\ \min_{c_1, c_2, \dots, c_\Gamma} |T| &= \min_{c_1, c_2, \dots, c_\Gamma} (c_\Gamma - c_1) + \max_{v \in U_\Gamma} \tau \cdot v - \min_{v \in U_1} \tau \cdot v + 1 = \min_{c_1, c_2, \dots, c_\Gamma} ((c_\Gamma - \\ &- c_{\Gamma-1}) + (c_{\Gamma-1} - c_{\Gamma-2}) + \dots + (c_2 - c_1)) + \max_{v \in U_\Gamma} \tau \cdot v - \min_{v \in U_1} \tau \cdot v + 1. \end{aligned}$$

Так как  $c_1, c_2, \dots, c_\Gamma$  должны удовлетворять как неравенствам (2.29), так и неравенствам (2.31), то  $\min_{c_1, c_2, \dots, c_\Gamma} |T|$  достигается при  $c_{\gamma+1} - c_\gamma = \max\{-x_\gamma, y_\gamma\}$ .

Таким образом, параллельная форма алгоритма заданной ширины  $\Delta$  для реализации на линейном матричном спецпроцессоре может быть получена следующим образом.

1. Выбрать линейную функцию  $\pi$ , удовлетворяющую условию (2.23).
2. Задать число процессорных элементов  $\Delta$ , определить  $\Gamma$  и области  $U_\gamma, U_\gamma^\delta$ .
3. Выбрать направляющий вектор графа, удовлетворяющий условиям (2.22) и (2.30).
4. Задать константы  $c_1, c_2, \dots, c_\Gamma$  рекуррентными соотношениями (2.32).
5. Вычислить временные задержки  $h_\varphi(v) = t(v + \varphi) - t(v)$ ,  $v \in V$ ,  $\varphi \in \Phi$  ( $h_\varphi(v)$  — задержка в вершине  $v$  данного, пересылаемого в направлении вектора  $\varphi$ ):

$$h_\varphi(v) = c_\gamma + \tau \cdot (v + \varphi) - (c_\gamma + \tau \cdot v) = \tau \cdot \varphi,$$

если  $v, v + \varphi \in U_\gamma, 1 \leq \gamma \leq \Gamma$ ;

$$h_\varphi(v) = c_{\gamma+1} - c_\gamma + \tau \cdot \varphi,$$

если  $v \in U_\gamma, v + \varphi \in U_{\gamma+1}, 1 \leq \gamma \leq \Gamma - 1$ .

6. Получить параллельную форму алгоритма в виде вычислительного графа. Граф содержит заданное число  $\Delta$  вершин, связи между вершинами определяются одномерными векторами с координатами  $0, 1, -(\Delta - 1)$  (см. теорему 2.7). Задержка в вершине  $f(v)$  данного, пересылаемого в направлении одномерного вектора  $f(v + \varphi) - f(v)$ , определяется величиной  $h_\varphi(v)$ ; если  $f(v + \varphi) - f(v) = 0$ , то данное остается в вершине  $f(v)$  и будет участвовать в вычислениях через временной промежуток  $h_\varphi(v)$ .

7. Описать функционирование локальной памяти.

### Пример построения параллельной формы заданной ширины алгоритма Гаусса – Жордана для обращения матрицы

Пусть  $A = (a_{ij})$ ,  $1 \leq i, j \leq N$ , квадратная матрица порядка  $N$ . Воспользуемся следующим алгоритмом Гаусса – Жордана для получения обратной матрицы  $A^{-1} = X = (x_{ij})$ ,  $1 \leq i, j \leq N$ , записанным в виде системы рекуррентных уравнений вида (1.1), определив начальные значения переменной  $a$ :

$$\begin{aligned}
 a(i, j, 0) &= a_{ij}, \quad 1 \leq i, j \leq N; \\
 a(v) &= \begin{cases} c(v - e_1), & v \in V_6, \\ -c(v - e_1) \cdot b(v - e_2), & v \in V_5, \\ -c(v - e_1) \cdot b(v - e_2) + a(v - e_3), & v \in V_3; \end{cases} \\
 b(v) &= \begin{cases} a(v - e_3), & v \in V_1, \\ b(v - e_2), & v \in V_2 \cup V_3; \end{cases} \\
 c(v) &= \begin{cases} 1/b(v - e_2), & v \in V_4, \\ a(v - e_3)/b(v - e_2), & v \in V_2, \\ c(v - e_1), & v \in V_3 \cup V_5, \end{cases}
 \end{aligned} \tag{2.33}$$

где области  $V_\lambda$ ,  $1 \leq \lambda \leq 6$ , имеют вид:

$$V_1 = \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, j = k, k \leq i \leq N + k - 1, \},$$

$$V_2 = \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, i = k, k + 1 \leq j \leq N + k - 1, \},$$

$$V_3 = \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, k + 1 \leq i \leq N + k - 1, k + 1 \leq j \leq N + k - 1 \},$$

$$\begin{aligned}
V_4 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, i = k, j = N + k \}, \\
V_5 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, k + 1 \leq i \leq N + k - 1, j = N + k \}, \\
V_6 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, i = N + k, k + 1 \leq j \leq N + k \}.
\end{aligned}$$

Элементы обратной матрицы определяются как соответствующие значения переменной  $a$ :

$$x_{ij} = a(N + i, N + j, N), \quad 1 \leq i, j \leq N.$$

Граф зависимостей алгоритма (2.33) имеет шесть типов вершин в соответствии с количеством типов макроопераций. Вершины графа расположены в области  $V = \bigcup_{\lambda=1}^6 V_\lambda = \{ v(i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq \leq N, k \leq i, j \leq N + k, (i, j, k) \neq (N + k, k, k) \}$ , дуги характеризуются векторами  $e_1 = (1, 0, 0)$ ,  $e_2 = (0, 1, 0)$ ,  $e_3 = (0, 0, 1)$ .

Зададим отображение  $\pi : \mathbf{Z}^3 \rightarrow \mathbf{Z}^1$  матрицей  $(0 \ 0 \ 1)$ . Будем предполагать, что  $N$  делится на  $\Delta$ , т. е.  $N = \Gamma \Delta$ . В противном случае рассмотрим расширенную матрицу  $A'$  порядка  $N' = \lceil N/\Delta \rceil \Delta$ . Матрица  $A'$  получается присоединением к  $A$   $N' - N$  новых строк и столбцов. Новые элементы  $A'$  равны нулю, исключая диагональные, равные единице.

Так как при выбранном операторе отображения вершины различного типа графа зависимостей отображаются в одну и ту же вершину вычислительного графа, необходимо обеспечить вершины одного типа одинаковыми управляющими метками. С этой целью управляющие метки  $m_1, m_2, \dots, m_6$  присваиваются переменной  $m(v)$  в вершинах  $v(i, j, 1)$ ,  $1 \leq i, j \leq N + 1$ , за исключением вершины  $v(N + 1, 1, 1)$ , а затем пересылаются к другим вершинам по направлению вектора  $\varphi^{(m)} = e_1 + e_2 + e_3 = (1, 1, 1)$ :  $m(v) = m(v - \varphi^{(m)})$ ,  $v \in V$ .

Множества  $U_\gamma$  и  $U_\gamma^\delta$ ,  $1 \leq \gamma \leq \Gamma$ ,  $\Gamma = N/\Delta$ ,  $1 \leq \gamma \leq \Gamma$ , имеют следующий вид:

$$\begin{aligned}
U_\gamma &= \{ v(i, j, k) \in \mathbf{Z}^3 \mid 1 + (\gamma - 1)\Delta \leq k \leq \gamma\Delta, k \leq i, j \leq N + \\
&\quad + k, (i, j, k) \neq (N + k, k, k) \}; \\
U_\gamma^\delta &= \{ v(i, j, k) \in \mathbf{Z}^3 \mid (\gamma - 1)\Delta + \delta \leq i, j \leq (\gamma - 1)\Delta + \delta + \\
&\quad + N, (i, j, k) \neq (N + k, k, k), k = (\gamma - 1)\Delta + \delta \}.
\end{aligned}$$

На рис. 2.8 изображен граф зависимостей алгоритма (2.33) Гаусса–Жордана для обращения квадратной матрицы порядка  $N = 3$ .

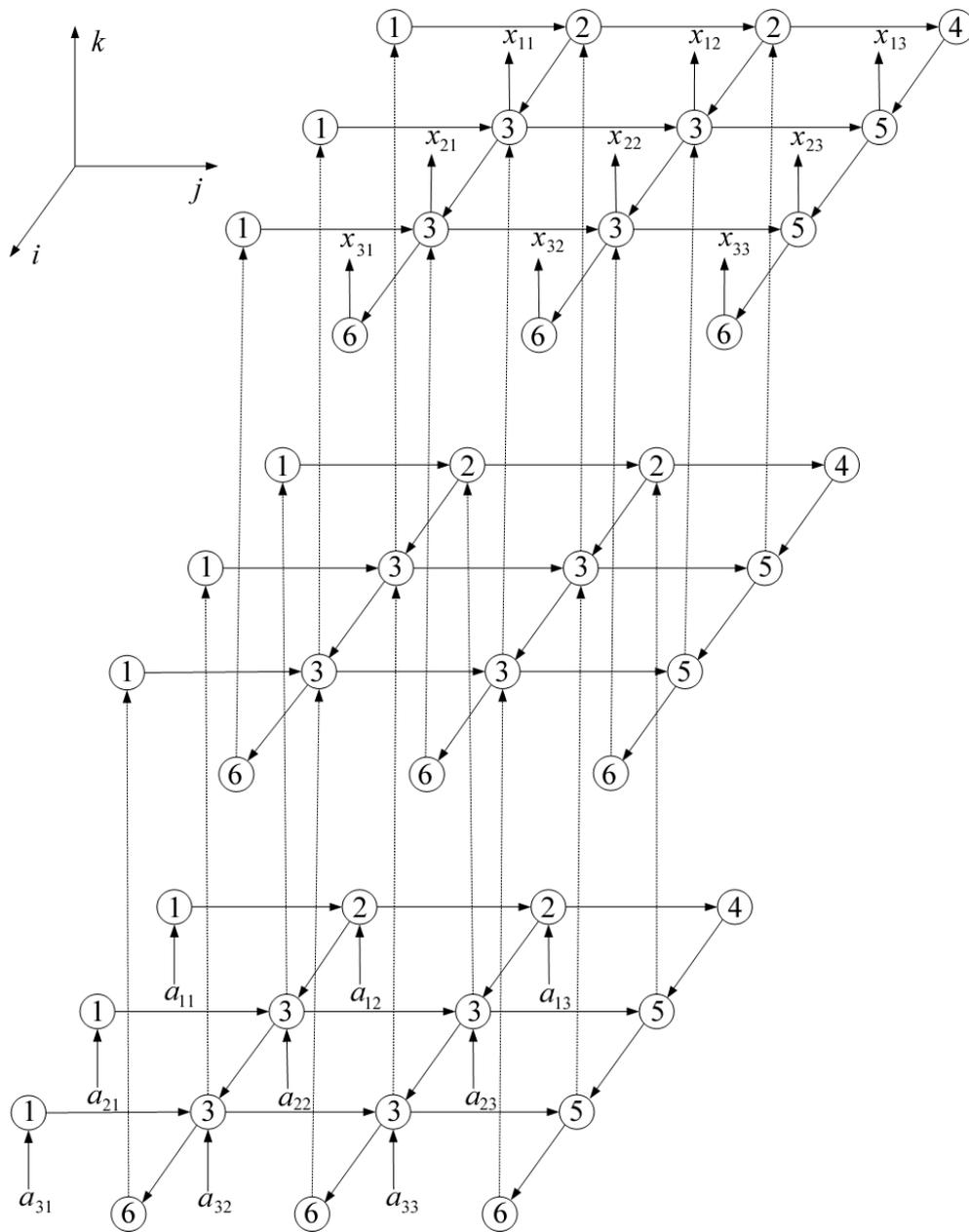


Рис. 2.8. Граф зависимостей алгоритма (2.33) при  $N = 3$ , вершины помечены номером соответствующей макрооперации

Так как  $U_{\gamma+1}$  может быть получено параллельным переносом множества  $U_\gamma$  на вектор  $(\Delta, \Delta, \Delta)$ , то

$$U_1 \ominus U_1 = U_2 \ominus U_2 = \dots = U_\Gamma \ominus U_\Gamma,$$

$$S = \{ (c_1, c_2, 0) \in \mathbf{Z}^3 \mid |c_1| \leq N, |c_2| \leq N, c_1^2 + c_2^2 \neq 0, c_1 \cdot c_2 \neq -N^2 \}.$$

Используя правило, сформулированное в теореме 2.9, находим, что векторы  $\tau = (1, N + 1, 1)$  и  $\tau = (N + 1, 1, 1)$  удовлетворяют условию (2.30). Далее будем считать, что  $\tau = (1, N + 1, 1)$ .

Вычислим константы  $c_1, c_2, \dots, c_\Gamma$  и временные задержки:

$$\begin{aligned} y_\gamma &= \max_{1 \leq \delta \leq \Delta} \{ \max_{v \in U_\gamma^\delta} \tau \cdot v - \min_{v \in U_{\gamma+1}^\delta} \tau \cdot v \} + 1 = \max_{1 \leq \delta \leq \Delta} \{ (\gamma - 1)\Delta + \delta + N + \\ &+ ((\gamma - 1)\Delta + \delta + N)(N + 1) + (\gamma - 1)\Delta + \delta - (\gamma\Delta + \delta) - (\gamma\Delta + \\ &+ \delta)(N + 1) - (\gamma\Delta + \delta) \} + 1 = (N - \Delta)(N + 2) - \Delta + 1; \end{aligned}$$

$$x_\gamma = \min_{\varphi \in \Phi_\gamma} \tau \cdot \varphi - 1 = \min_{e_3=(0,0,1), \varphi^{(m)}=(1,1,1)} \tau \cdot \varphi - 1 = 0;$$

$$\begin{aligned} c_{\gamma+1} &= c_\gamma + \max\{-x_\gamma, y_\gamma\} = c_\gamma + (N - \Delta)(N + 2) - \Delta + 1 = c_1 + \\ &+ \gamma((N - \Delta)(N + 2) - \Delta + 1). \end{aligned}$$

Так как  $v_{\min}(1, 1, 1) \in U_1$ , то из равенства  $t(v_{\min}) = c_1 + N + 3 = 1$  получим  $c_1 = -(N + 2)$ , следовательно,  $c_\gamma = (\gamma - 1)((N - \Delta)(N + 2) - \Delta + 1) - N - 2$ .

Если  $v, v + \varphi \in U_\gamma$ ,  $1 \leq \gamma \leq \Gamma$ , то  $h_\varphi(v) = \tau \cdot \varphi$  и  $h_{e_1}(v) = \tau \cdot e_1 = 1$ ,  $h_{e_2}(v) = \tau \cdot e_2 = N + 1$ ,  $h_{e_3}(v) = \tau \cdot e_3 = 1$ ,  $h_{\varphi^{(m)}}(v) = \tau \cdot \varphi^{(m)} = N + 3$ .

Если  $v \in U_\gamma$ ,  $v + \varphi \in U_{\gamma+1}$ ,  $1 \leq \gamma \leq \Gamma - 1$ , то  $h_{e_3}(v) = c_{\gamma+1} - c_\gamma + \tau \cdot e_3 = (N - \Delta)(N + 2) - \Delta + 2$ ,  $h_{\varphi^{(m)}}(v) = c_{\gamma+1} - c_\gamma + \tau \cdot \varphi^{(m)} = (N - \Delta)(N + 2) - \Delta + 1 + N + 3 = (N - \Delta)(N + 3) + 4$ .

Каждый процессорный элемент  $\Pi\mathfrak{E}(\tilde{v}) \in V^\pi$ , помимо двух входов  $I^{(a)}$ ,  $I^{(m)}$  и двух выходов  $O^{(a)}$ ,  $O^{(m)}$ , имеет арифметическое устройство, необходимое для выполнения операций алгоритма над входными данными, содержит четыре блока ячеек локальной памяти для задержки вычисленных значений переменных алгоритма  $a, b, c, m$ . Содержимое ячеек  $R^{(a)}(t)$ ;  $R_1^{(b)}(t)$ ,  $R_2^{(b)}(t), \dots, R_{N+1}^{(b)}(t)$ ;  $R^{(c)}(t)$ ;  $R_1^{(m)}(t), \dots, R_{N+3}^{(m)}(t)$  меняется в зависимости от момента времени  $t$ , определяемого таймирующей функцией

$$\begin{aligned} t(v) &= \sum_{\gamma=1}^{\Gamma} [(\gamma - 1)((N - \Delta)(N + 2) - \Delta + 1) - N - 2 + \tau \cdot v] 1_{U_\gamma}(v), \\ &v \in V, \tau = (1, N + 1, 1). \end{aligned}$$

Приведем описание функционирования локальной памяти ПЭ( $\tilde{v}$ )  $\in V^\pi$ :

$$R_1^{(m)}(t) = I^{(m)}, R_\alpha^{(m)}(t + \alpha - 1) = R_{\alpha-1}^{(m)}(t + \alpha - 2), 2 \leq \alpha \leq N + 3;$$

$$R^{(a)}(t) = \begin{cases} R^{(c)}(t - 1), & \text{если } I^{(m)} = m_6, \\ -R^{(c)}(t - 1)R_{N+1}^{(b)}(t - 1), & \text{если } I^{(m)} = m_5, \\ -R^{(c)}(t - 1)R_{N+1}^{(b)}(t - 1) + I^{(a)}, & \text{если } I^{(m)} = m_3; \end{cases}$$

$$R_1^{(b)}(t) = \begin{cases} I^{(a)}, & \text{если } I^{(m)} = m_1, \\ R_{N+1}^{(b)}(t - 1), & \text{если } I^{(m)} = m_2 \vee m_3; \end{cases}$$

$$R_\alpha^{(b)}(t + \alpha - 1) = R_{\alpha-1}^{(b)}(t + \alpha - 2), \text{ если } 2 \leq \alpha \leq N + 1;$$

$$R^{(c)}(t) = \begin{cases} 1/R_{N+1}^{(b)}(t - 1), & \text{если } I^{(m)} = m_4, \\ I^{(a)}/R_{N+1}^{(b)}(t - 1), & \text{если } I^{(m)} = m_2, \\ R^{(c)}(t - 1), & \text{если } I^{(m)} = m_3 \vee m_5; \end{cases}$$

$$O^{(m)} = R_{N+3}^{(m)}(t + N + 2), O^{(a)} = R^{(a)}(t).$$

Данные с выходов  $O^{(a)}$  и  $O^{(m)}$  ПЭ( $\Delta$ ) при  $\Delta < N$  с дополнительной задержкой  $c_{\gamma+1} - c_\gamma = (N - \Delta)(N + 2) - \Delta + 1$  тактов по обратной связи поступают на входы  $I^{(a)}$  и  $I^{(m)}$  первого ПЭ(1).

Режим ввода начальных данных и управляющих меток приведен в табл. 2.6.

Таблица 2.6

**Ввод начальных данных, управляющих меток и вывода результатов спецпроцессора для получения обратной матрицы с помощью алгоритма Гаусса – Жордана**

Данные	Время ввода	Изменение параметров	Вход ПЭ(1)
$a_{ij}$	$i + (N + 1)(j - 1)$	$1 \leq i, j \leq N$	$I^{(a)}$
$m_1$	$i$	$1 \leq i \leq N$	$I^{(m)}$
$m_2$	$1 + (N + 1)(j - 1)$	$2 \leq j \leq N$	$I^{(m)}$
$m_3$	$i + (N + 1)(j - 1)$	$2 \leq i, j \leq N$	$I^{(m)}$
$m_4$	$(N + 1)^2 - N$	—	$I^{(m)}$
$m_5$	$i + (N + 1)N$	$2 \leq i \leq N$	$I^{(m)}$
$m_6$	$(N + 1)j$	$2 \leq j \leq N + 1$	$I^{(m)}$

Результаты вычислений  $x_{ij}$  снимаются с выхода  $O^{(a)}$  последнего ПЭ( $\Delta$ ) в момент времени  $t(i + N, j + N, N) = (\Gamma - 1)((N - \Delta)(N + 2) - \Delta + 1) - N - 2 + i + N + (N + 1)(j + N) + N$ . Полное время обращения квадратной матрицы порядка  $N$  на спроектированном спецпроцессоре, состоящем из  $\Delta$  линейно связанных процессорных элементов, составляет  $t(2N, 2N, N) = (N/\Delta)(N + 1)^2 + (\Delta - 1)(N + 3)$  тактов.

На рис. 2.9 изображен вычислительный граф, представляющий параллельную форму заданной ширины  $\Delta$  алгоритма (2.33).

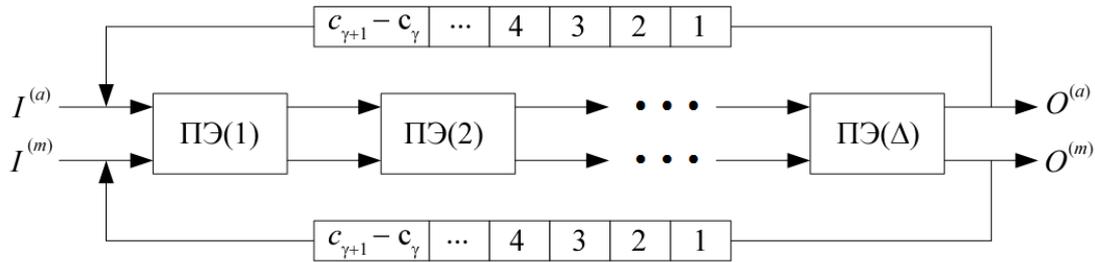


Рис. 2.9. Вычислительный граф, представляющий параллельную форму заданной ширины  $\Delta$  для обращения матриц

### Применение локально последовательной глобально параллельной стратегии разбиения

Пусть  $\Gamma$  – заданное целое положительное число,  $\pi: \mathbf{Z}^d \rightarrow \mathbf{Z}$  – линейное отображение, удовлетворяющее условию

$$\pi(\varphi) \in \{(-1), (0), (1)\} \text{ для всех } \varphi \in \Phi. \quad (2.34)$$

Обозначим, как и в предыдущем параграфе

$$\delta_{\min} = \min_{v \in V} \pi(v),$$

$$\Delta = (\max_{v \in V} \pi(v) - \delta_{\min} + 1)/\Gamma.$$

Не ограничивая общности, можно предполагать, что  $\Delta$  – целое число. В противном случае следует расширить область вычислений  $V$  и положить

$$\Delta = \lceil (\max_{v \in V} \pi(v) - \delta_{\min} + 1)/\Gamma \rceil.$$

Разобьем множество вершин  $V$  на подмножества:

$$U_\gamma = \{v \in V \mid \delta_{\min} + (\gamma - 1)\Delta \leq \pi(v) \leq \delta_{\min} + \gamma\Delta - 1\};$$

$$U_\gamma^\delta = \{v \in U_\gamma \mid \pi(v) = \delta_{\min} + (\gamma - 1)\Delta + \delta - 1\}, \quad 1 \leq \delta \leq \Delta, \quad 1 \leq \gamma \leq \Gamma.$$

Функцию размещения зададим с помощью нелинейного отображения  $f: V \rightarrow Z$ :

$$f(v) = \sum_{\gamma=1}^{\Gamma} \gamma 1_{U_\gamma}(v),$$

где, напомним,

$$1_W(v) = \begin{cases} 1, & \text{если } v \in W, \\ 0, & \text{если } v \notin W. \end{cases}$$

Функция  $f(v)$  отображает множество  $V$  в множество  $\{1, 2, \dots, \Gamma\}$ . Следовательно, вычислительный граф будет иметь  $\Gamma$  вершин. Дуги  $(v, v + \varphi)$  отображаются в дуги  $(f(v), f(v + \varphi))$ .

**Теорема 2.12.** Пусть  $v \in U_{\gamma_1}^{\delta_1}$ ,  $(v + \varphi) \in U_{\gamma_2}^{\delta_2}$ ,  $1 \leq \gamma_1, \gamma_2, \leq \Gamma$ . Тогда  $|\gamma_2 - \gamma_1| \leq 1$  и

$$f(v + \varphi) - f(v) = \begin{cases} -1, & \text{если } \gamma_2 - \gamma_1 = -1, \delta_1 = 1, \delta_2 = \Delta, \\ 0, & \text{если } \gamma_2 - \gamma_1 = 0, \delta_2 - \delta_1 = 0, 1, -1, \\ 1, & \text{если } \gamma_2 - \gamma_1 = 1, \delta_1 = \Delta, \delta_2 = 1. \end{cases} \quad (2.35)$$

**Доказательство.** Пусть  $v \in U_{\gamma_1}^{\delta_1}$ ,  $(v + \varphi) \in U_{\gamma_2}^{\delta_2}$ ,  $1 \leq \gamma_1, \gamma_2, \leq \Gamma$ ,  $1 \leq \delta_1, \delta_2 \leq \Delta$ , т. е.  $\pi(v) = \delta_{\min} + (\gamma_1 - 1)\Delta + \delta_1 - 1$ ,  $\pi(v + \varphi) = \delta_{\min} + (\gamma_2 - 1)\Delta + \delta_2 - 1$ . Тогда  $\pi(v + \varphi) - \pi(v) = (\gamma_2 - \gamma_1)\Delta + \delta_2 - \delta_1$ , причем  $|\delta_2 - \delta_1| \leq \Delta - 1$ .

Возможны три случая:

1)  $\pi(\varphi) = 0$ . Тогда  $(\gamma_2 - \gamma_1)\Delta + \delta_2 - \delta_1 = 0$ , что возможно только в случае  $\gamma_2 - \gamma_1 = 0$ ,  $\delta_2 - \delta_1 = 0$ ;

2)  $\pi(\varphi) = -1$ . Тогда  $(\gamma_2 - \gamma_1)\Delta + \delta_2 - \delta_1 = -1$ , и либо  $\gamma_2 - \gamma_1 = 0$ ,  $\delta_2 = \delta_1 - 1$ , либо  $\gamma_2 - \gamma_1 = -1$ ,  $\delta_1 = 1$ ,  $\delta_2 = \Delta$ ;

3)  $\pi(\varphi) = 1$ . Тогда  $(\gamma_2 - \gamma_1)\Delta + \delta_2 - \delta_1 = 1$ , и либо  $\gamma_2 - \gamma_1 = 0$ ,  $\delta_2 = \delta_1 + 1$ , либо  $\gamma_2 - \gamma_1 = 1$ ,  $\delta_1 = \Delta$ ,  $\delta_2 = 1$ .

Во всех трех случаях  $|\gamma_2 - \gamma_1| \leq 1$ . Формула (2.35) непосредственно вытекает из определения функции размещения  $f$  и полученной связи между  $\gamma_1$  и  $\gamma_2$ ,  $\delta_1$  и  $\delta_2$ . Теорема 2.12 доказана.

Из теоремы 2.12 следует, что связи между вершинами вычислительного графа порождаются только дугами, соединяющими вершины множества  $U_{\gamma+1}^1$  с вершинами множества  $U_{\gamma}^{\Delta}$ .

Таймирующую функцию  $t: V \rightarrow Z_+$  определим равенством

$$t(v) = \sum_{\gamma=1}^{\Gamma} (c_{\gamma} + \tau \cdot v) 1_{U_{\gamma}}(v), \quad (2.36)$$

где  $c_{\gamma}$ ,  $1 \leq \gamma \leq \Gamma$ , — константы.

Потребуем, чтобы таймирующая функция удовлетворяла ограничениям:

$$t(v + \varphi) > t(v), \quad v, v + \varphi \in V, \quad \varphi \in \Phi \quad (2.37)$$

(условие непротиворечивости таймирования графа  $G$ );

$$t(v_1) \neq t(v_2), \quad \text{если } v_1, v_2 \in U_{\gamma}, \quad 1 \leq \gamma \leq \Gamma, \quad v_1 \neq v_2 \quad (2.38)$$

(всем вершинам из области  $U_{\gamma}$ , отображаемым в одну и ту же вершину вычислительного графа, приписывается разное значение таймирующей функции).

Условие (2.38) можно переписать в виде

$$\tau \cdot (v_1 - v_2) \neq 0, \quad \text{если } v_1, v_2 \in U_{\gamma}, \quad 1 \leq \gamma \leq \Gamma, \quad v_1 \neq v_2. \quad (2.39)$$

Вектор  $\tau$ , удовлетворяющий ограничениям (2.22) и (2.39), определяет порядок, в котором операции, соответствующие вершинам из  $U_{\gamma}$ , выполняются в вершине  $\gamma$  вычислительного графа.

**Теорема 2.13** *Ограничение (2.37) на таймирующую функцию вида (2.36) выполняется тогда и только тогда, когда выполняется условие (2.22) и неравенства*

$$-p_{\gamma} \leq c_{\gamma+1} - c_{\gamma} \leq q_{\gamma}, \quad 1 \leq \gamma \leq \Gamma, \quad (2.40)$$

где  $p_{\gamma} = \min_{\varphi \in \Phi_{\gamma}} \tau \cdot \varphi - 1$ ,  $q_{\gamma} = \min_{\varphi \in \Psi_{\gamma+1}} \tau \cdot \varphi - 1$ ;

$$\Phi_{\gamma} = \{ \varphi \in \Phi \mid \varphi = v_2 - v_1, \quad v_1 \in U_{\gamma}^{\Delta}, \quad v_2 \in U_{\gamma+1}^1, \};$$

$$\Psi_{\gamma+1} = \{ \varphi \in \Phi \mid \varphi = v_2 - v_1, \quad v_1 \in U_{\gamma+1}^1, \quad v_2 \in U_{\gamma}^{\Delta} \}.$$

**Доказательство.** Если  $v, v + \varphi \in U_{\gamma}$ , то  $t(v + \varphi) - t(v) = c_{\gamma} + \tau \cdot (v + \varphi) - (c_{\gamma} + \tau \cdot v) = \tau \cdot \varphi$ . Таким образом, условия (2.22) и (2.37) для данного случая эквивалентны.

Если  $v$  и  $v + \varphi$  принадлежат разным множествам  $U_\gamma$ , то возможны два случая:  $\varphi \in \Phi_\gamma$  либо  $\varphi \in \Psi_{\gamma+1}$ , причем  $\gamma$  может быть любым из множества  $\{1, 2, \dots, \Gamma - 1\}$ . В первом случае неравенство (2.37) эквивалентно неравенству  $c_{\gamma+1} + \tau \cdot (v + \varphi) - (c_\gamma + \tau \cdot v) \geq 1$  или  $c_{\gamma+1} - c_\gamma + \tau \cdot \varphi \geq 1$ , из которого следует

$$c_{\gamma+1} - c_\gamma \geq -p_\gamma, \quad \varphi \in \Phi_\gamma. \quad (2.41)$$

Во втором случае неравенство (2.37) эквивалентно неравенству  $c_\gamma + \tau \cdot (v + \varphi) - (c_{\gamma+1} + \tau \cdot v) \geq 1$  или  $c_{\gamma+1} - c_\gamma - \tau \cdot \varphi \leq -1$ , из которого следует

$$c_{\gamma+1} - c_\gamma \leq q_\gamma, \quad \varphi \in \Psi_{\gamma+1}. \quad (2.42)$$

Таким образом, если верно неравенство (2.37), то из неравенств (2.41) и (2.42) следует (2.40), если верно двойное неравенство (2.40), то из (2.41) и (2.42) следует (2.37).

Теорема 2.13 доказана.

**Замечание 2.5.** *Так как  $-p_\gamma \leq 0$ , а  $q_\gamma \geq 0$ , то всегда можно выбрать  $c_1 = c_2 = \dots = c_\Gamma$ , однако такой выбор не всегда приводит к параллельным формам минимальной высоты.*

Пусть таймирующая функция имеет вид (2.36). Вершина, для которой в графе алгоритма  $G$  нет предшествующей, единственна и принадлежит области  $U_{\gamma_0}$ , а вершина, для которой в графе нет последующей, также единственна и принадлежит области  $U_{\gamma_1}$ . Тогда справедлива следующая теорема.

**Теорема 2.14.** *При фиксированном  $\tau$  минимальная высота параллельной формы алгоритма достигается при  $c_1, c_2, \dots, c_\Gamma$ , задаваемых рекуррентными соотношениями*

$$c_{\gamma+1} = c_\gamma - p_\gamma, \quad 1 \leq \gamma \leq \Gamma - 1, \quad (2.43)$$

если  $\gamma_1 > \gamma_0$ , и соотношениями

$$c_{\gamma+1} = c_\gamma + q_\gamma, \quad 1 \leq \gamma \leq \Gamma - 1, \quad (2.44)$$

если  $\gamma_1 < \gamma_0$ ,  $c_1$  выбирается произвольно.

**Доказательство.** Пусть  $|T| = \max_{v \in U_{\gamma_1}} t(v) - \min_{v \in U_{\gamma_0}} t(v) + 1$  высота параллельной формы.

В случае  $\gamma_1 > \gamma_0$  имеем:

$$\begin{aligned} \min_{c_1, \dots, c_\Gamma} |T| &= \min_{c_1, \dots, c_\Gamma} (\max_{v \in U_{\gamma_1}} (c_{\gamma_1} + \tau \cdot v) - \min_{v \in U_{\gamma_0}} (c_{\gamma_0} + \tau \cdot v)) + 1 = \min_{c_1, \dots, c_\Gamma} (c_{\gamma_1} - \\ &- c_{\gamma_0}) + \max_{v \in U_{\gamma_1}} \tau \cdot v - \min_{v \in U_{\gamma_0}} \tau \cdot v + 1 = \min_{c_1, \dots, c_\Gamma} ((c_{\gamma_1} - c_{\gamma_1-1}) + (c_{\gamma_1-1} - \\ &- c_{\gamma_1-2}) + \dots + (c_{\gamma_0+1} - c_{\gamma_0})) + \max_{v \in U_{\gamma_1}} \tau \cdot v - \min_{v \in U_{\gamma_0}} \tau \cdot v + 1. \end{aligned}$$

Отсюда и из условия (2.40) следует, что  $\min_{c_1, \dots, c_\Gamma} |T|$  достигается при выполнении равенства (2.43).

В случае  $\gamma_1 < \gamma_0$

$$\begin{aligned} \min_{c_1, \dots, c_\Gamma} |T| &= \min_{c_1, \dots, c_\Gamma} ((c_{\gamma_1} - c_{\gamma_1+1}) + (c_{\gamma_1+1} - c_{\gamma_1+2}) + \dots + (c_{\gamma_0-1} - c_{\gamma_0})) + \\ &+ \max_{v \in U_{\gamma_1}} \tau \cdot v - \min_{v \in U_{\gamma_0}} \tau \cdot v + 1. \end{aligned}$$

Отсюда и из условия (2.40)  $c_\gamma - c_{\gamma+1} \geq -q_\gamma$  следует, что  $\min_{c_1, \dots, c_\Gamma} |T|$  достигается при выполнении равенства (2.44).

Теорема 2.14 доказана.

Таким образом, параллельная форма алгоритма заданной ширины  $\Gamma$  для реализации на линейном матричном спецпроцессоре может быть получена следующим образом.

1. Выбрать линейную функцию  $\pi$ , удовлетворяющую условию (2.34).
2. Задать  $\Gamma$ , определить  $\Delta$  и найти области  $U_\gamma, U_\gamma^\delta$ .
3. Выбрать направляющий вектор графа  $\tau$ , удовлетворяющий условиям (2.22) и (2.39).
4. Задать константы  $c_1, c_2, \dots, c_\Gamma$  рекуррентными соотношениями (2.43) или (2.44) в зависимости от положения вершины, для которой в графе зависимостей нет предшествующей.

5. Вычислить временные задержки  $h_\varphi(v) = t(v + \varphi) - t(v)$ ,  $v \in V$ ,  $\varphi \in \Phi$ :

$$h_\varphi(v) = c_\gamma + \tau \cdot (v + \varphi) - (c_\gamma + \tau \cdot v) = \tau \cdot \varphi,$$

если  $v, v + \varphi \in U_\gamma, 1 \leq \gamma \leq \Gamma$ ;

$$h_\varphi(v) = c_{\gamma+1} - c_\gamma + \tau \cdot \varphi, \text{ если } v \in U_\gamma, v + \varphi \in U_{\gamma+1}, 1 \leq \gamma \leq \Gamma - 1;$$

$$h_\varphi(v) = c_\gamma - c_{\gamma+1} + \tau \cdot \varphi, \text{ если } v \in U_{\gamma+1}, v + \varphi \in U_\gamma, 1 \leq \gamma \leq \Gamma - 1.$$

6. Получить параллельную форму алгоритма в виде вычислительно-го графа. Граф содержит заданное число  $\Gamma$  вершин, связи между которыми определяются одномерными векторами с координатами 0, 1, -1 (см. теорему 2.12).

**Замечание 2.6.** В ЛПосГПар стратегии области  $U_\gamma^1$  и  $U_\gamma^\Delta$  являются «особыми», что порождает необходимость в дополнительных управляющих метках  $n_0, n_1, n_\Delta$  ( $\Omega(n_1) = U_\gamma^1, \Omega(n_\Delta) = U_\gamma^\Delta, \Omega(n_0) = U_\gamma \setminus (U_\gamma^1 \cup U_\gamma^\Delta)$ ).

**Пример построения параллельной формы алгоритма  
заданной ширины для для решения систем  
линейных алгебраических уравнений**

Необходимо получить параллельную форму фиксированной ширины  $\Gamma$  ( $1 < \Gamma < N$ ) для решения системы  $N$  линейных алгебраических уравнений с  $N$  неизвестными и  $M$  правыми частями  $A_1 X = A_2$ , где  $A_1 = (a_{ij}), 1 \leq i, j \leq N, A_2 = (a_{ij}), 1 \leq i \leq N, N + 1 \leq j \leq N + M, X = (x_{ij}), 1 \leq i \leq N, 1 \leq j \leq M$ . Для решения этой системы воспользуемся алгоритмом Гаусса – Жордана без выбора ведущего элемента:

$$a(i, j, 0) = a_{ij}, 1 \leq i \leq N, 1 \leq j \leq N + M;$$

$$k = 1, 2, \dots, N:$$

$$1) b(i, k, k) = a(i, k, k - 1), i = k, k + 1, \dots, N + k - 1;$$

$$2) \begin{cases} c(k, j, k) = a(k, j, k - 1)/b(k, j - 1, k), \\ b(k, j, k) = b(k, j - 1, k), \end{cases}$$

$$j = k + 1, k + 2, \dots, N + M;$$

$$3) \begin{cases} a(i, j, k) = a(i, j, k - 1) - b(i, j - 1, k) \cdot c(i - 1, j, k), \\ c(i, j, k) = c(i - 1, j, k), \\ b(i, j, k) = b(i, j - 1, k), \end{cases}$$

$$i = k + 1, k + 2, \dots, N + k - 1, j = k + 1, k + 2, \dots, N + M;$$

$$4) a(N + k, j, k) = c(N + k - 1, j, k), j = k + 1, k + 2, \dots, N + M;$$

$$x_{ij} = a(N + i, N + j, N), 1 \leq i \leq N, 1 \leq j \leq M.$$

Граф зависимостей алгоритма изображен на рис. 2.10. Граф имеет вершины четырех типов, расположенные в области  $V = \bigcup_{\lambda=1}^4 V_\lambda = \{(i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, k \leq i \leq N + k, k \leq j \leq N +$

$+ M, (i, j, k) \neq (N + k, k, k) \}$ . Дуги графа характеризуются векторами  $e_1 = (1, 0, 0)$ ,  $e_2 = (0, 1, 0)$ ,  $e_3 = (0, 0, 1)$ .

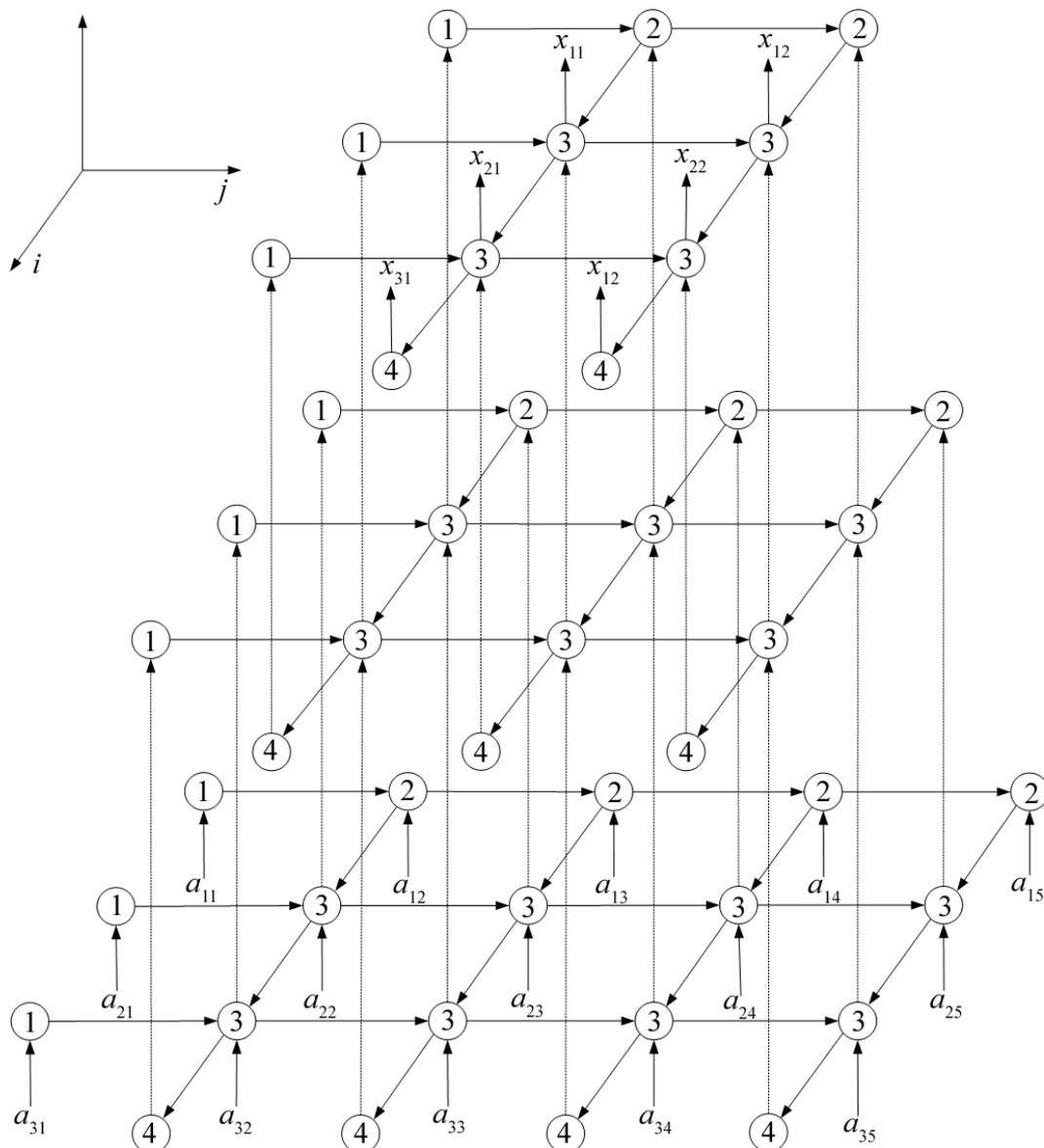


Рис. 2.10. Граф зависимостей алгоритма (2.45) при  $N = 3$ ,  $M = 2$ , вершины помечены номером соответствующей базовой операции

Обозначим.

$$\begin{aligned}
 V_1 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k = j \leq N, k \leq i \leq k + N - 1 \}, \\
 V_2 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k = i \leq N, k + 1 \leq j \leq N + M \}, \\
 V_3 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, k + 1 \leq i \leq k + N - 1, k + 1 \leq j \leq \\
 &\leq N + M \}, \\
 V_4 &= \{ (i, j, k) \in \mathbf{Z}^3 \mid 1 \leq k \leq N, i = N + k, k + 1 \leq j \leq N + M \},
 \end{aligned}$$

Перепишем алгоритм в виде системы рекуррентных уравнений (1.1):

$$\begin{aligned}
b(v) &= \begin{cases} a(v - e_3), & v \in V_1, \\ b(v - e_2), & v \in V_2 \cup V_3; \end{cases} \\
a(v) &= \begin{cases} a(v - e_3) - b(v - e_2) \cdot c(v - e_1), & v \in V_3, \\ c(v - e_1), & v \in V_4; \end{cases} \\
c(v) &= \begin{cases} a(v - e_3)/b(v - e_2), & v \in V_2, \\ c(v - e_1), & v \in V_3. \end{cases}
\end{aligned} \tag{2.45}$$

Определим линейное отображение  $\pi: \mathbf{Z}^3 \rightarrow \mathbf{Z}^1$  графа зависимостей на ось аппликат  $\pi(i, j, k) = k$ . Так как при выбранном операторе  $V_1^\pi = V_2^\pi = V_3^\pi = V_4^\pi = \{ (k) \in \mathbf{Z}^1 \mid 1 \leq k \leq N \}$ , необходимо введение четырех управляющих меток  $m_1, m_2, m_3, m_4$ . Метки будем присваивать в качестве начальных значений переменной  $m(v)$  в вершинах  $v(i, j, 1)$ ,  $1 \leq i \leq N + 1$ ,  $1 \leq j \leq N + M$ ,  $j \neq 1$ , если  $i = N + 1$ , и пересылать к остальным вершинам области их влияния  $\Omega(m_\lambda) = V_\lambda$ ,  $1 \leq \lambda \leq 4$ , по направлению вектора  $\varphi^{(m)} = e_1 + e_2 + e_3$ . Таким образом, алгоритм (2.45) следует дополнить уравнением

$$m(v) = m(v - e_1 - e_2 - e_3), \quad v \in V.$$

Далее, следуя ЛПосГПар стратегии, разобьем множество вершин графа зависимостей на подмножества:

$$U_\gamma = \{ (i, j, k) \in V \mid 1 + (\gamma - 1)\Delta \leq k \leq \gamma\Delta \}, \quad 1 \leq \gamma \leq \Gamma;$$

$$U_\gamma^\delta = \{ (i, j, k) \in U_\gamma \mid k = (\gamma - 1)\Delta + \delta \}, \quad 1 \leq \gamma \leq \Gamma, \quad 1 \leq \delta \leq \Delta.$$

Будем предполагать, что  $\Delta = N/\Gamma$  — целое число. В противном случае следует рассмотреть расширенную матрицу  $A'$  порядка  $N' = \lceil N/\Gamma \rceil \Gamma$ . Матрица  $A'$  получается добавлением  $N' - N$  новых строк и столбцов к матрице  $A$ . Новые элементы матрицы  $A'$  равны нулю, за исключением диагональных, равных единице.

Выбор вектора  $\tau = (\tau_1, \tau_2, \tau_3)$ , удовлетворяющего условиям  $\tau \cdot \varphi \geq 1$ ,  $\varphi \in \Phi$ , и  $\tau \cdot (v_2 - v_1) \neq 0$ ,  $v_1, v_2 \in U_\gamma$ , сводится к решению системы неравенств

$$\tau_1(i_2 - i_1) + \tau_2(j_2 - j_1) + \tau_3(k_2 - k_1) \neq 0, \quad \tau_1 \geq 1, \quad \tau_2 \geq 1, \quad \tau_3 \geq 1,$$

которые должны выполняться для любых точек  $v_1(i_1, j_1, k_1)$ ,  $v_2(i_2, j_2, k_2) \in U_\gamma^* = \{(i, j, k) \in \mathbf{Z}^3 \mid 1 + (\gamma - 1)\Delta \leq k \leq \gamma\Delta, k \leq i \leq N + k, k \leq j \leq N + M\}$ . ( $U_\gamma^*$  отличается от  $U_\gamma$  только одной добавленной (для упрощения выкладок) точкой  $(N + k, k, k)$  в каждом  $k$ -м слое.)

Из оценок

$$\begin{aligned} |k_2 - k_1| &\leq \Delta - 1, \\ |j_2 - j_1| &\leq N + M - (1 + (\gamma - 1)\Delta), \\ |i_2 - i_1| &\leq N + \Delta - 1 \end{aligned}$$

следует, что задача свелась к поиску плоскостей с нормальным вектором  $\tau$ , проходящих через начало координат и не содержащих точек множества  $C = \{(c_1, c_2, c_3) \in \mathbf{Z}^3 \mid |c_1| \leq N + \Delta - 1, |c_2| \leq N + M - (1 + (\gamma - 1)\Delta), |c_3| \leq \Delta - 1\}$ :

$$\tau_1 c_1 + \tau_2 c_2 + \tau_3 c_3 \neq 0, (c_1, c_2, c_3) \in C.$$

Так как область  $C$  есть параллелепипед:

$$C = \{(c_1, c_2, c_3) \in \mathbf{Z}^3 \mid |c_1| \leq J_1 - 1, |c_2| \leq J_2 - 1, |c_3| \leq J_3 - 1\},$$

то можно воспользоваться теоремой 2.3, полагая  $\tau_{\alpha_1} = 1$ ,  $\tau_{\alpha_2} = J_{\alpha_1}$ ,  $\tau_{\alpha_3} = J_{\alpha_1} \cdot J_{\alpha_2}$ , где  $\alpha_1, \alpha_2, \alpha_3$  — любая перестановка чисел 1, 2, 3. Выберем вариант  $\alpha_1 = 3, \alpha_2 = 1, \alpha_3 = 2$ . Получим  $\tau_3 = 1$ ,  $\tau_1 = J_3 = \Delta$ ,  $\tau_2 = J_3 \cdot J_1 = \Delta(N + \Delta)$ . Мы нашли вектор  $\tau = (\Delta, \Delta(N + \Delta), 1)$ , удовлетворяющий условиям (2.22) и (2.39).

Для определения таймирующей функции необходимо еще найти константы  $c_\gamma$ ,  $1 \leq \gamma \leq \Gamma$ , из рекуррентных соотношений (2.43). Учитывая, что  $\Psi_{\gamma+1} = \emptyset$ ,  $\Phi_\gamma = \{e_3, e_1 + e_2 + e_3\}$ , а  $p_\gamma = \min_{\varphi \in \Phi_\gamma} \tau \cdot \varphi - 1 = \tau_3 - 1 = 0$ , имеем  $c_\gamma = c_1$ ,  $1 \leq \gamma \leq \Gamma$ . Таким образом, получим таймирующую функцию (2.36):

$$t(v) = \tau \cdot v + c_1 = \Delta i + \Delta(N + \Delta)j + k + c_1, v = (i, j, k) \in V.$$

В графе зависимостей вершина  $v_{\min} = (1, 1, 1)$  не имеет предшествующих. Полагая  $t(v_{\min}) = 1$ , получим  $c_1 = -\Delta - \Delta(N + \Delta)$ . Окончательно имеем  $t(v) = \Delta(i - 1) + \Delta(N + \Delta)(j - 1) + k$ ,  $v = (i, j, k) \in V$ .

Полное время решения задачи:

$$|T| = t(2N, N + M, N) = \Delta(2N - 1) + \Delta(N + \Delta)(N + M - 1) + N.$$

Временные задержки:

$$h_{e_1} = \Delta;$$

$$h_{e_2} = \Delta(N + \Delta);$$

$$h_{e_3} = 1,$$

$$h_{e_1+e_2+e_3} = \Delta(N + \Delta + 1) + 1.$$

Вычислительный граф содержит  $\Gamma$  вершин – процессорных элементов  $\text{ПЭ}(\gamma)$ ,  $1 \leq \gamma \leq \Gamma$ . Каждый процессорный элемент имеет три входа  $I^{(a)}$ ,  $I^{(m)}$ ,  $I^{(n)}$ , два выхода  $O^{(a)}$ ,  $O^{(m)}$ , необходимое устройство для выполнения операций алгоритма и четыре блока ячеек локальной памяти для хранения значений  $a$ ,  $b$ ,  $c$  и  $m$ . Содержимое ячеек  $R_1^{(a)}(t)$ ;  $R_1^{(c)}(t), \dots, R_{\Delta}^{(c)}(t)$ ;  $R_1^{(b)}(t), \dots, R_{\Delta(N+\Delta)}^{(b)}(t)$ ;  $R_1^{(m)}(t), \dots, R_{\Delta(N+\Delta+1)+1}^{(m)}(t)$  меняется в зависимости от момента времени  $t$ , определяемого таймирующей функцией  $t(v) = \Delta(i-1) + \Delta(N+\Delta)(j-1) + k$ ,  $v = (i, j, k) \in V$ , от поступившей в этот момент времени на вход  $I^{(m)}$  управляющей метки и от поступившей в этот момент времени на дополнительный вход  $I^{(n)}$  управляющей метки  $n_0$  либо  $n_1$  в соответствии с замечанием 2.6,  $\Omega(n_1) = U_{\gamma}^1$ ,  $\Omega(n_0) = U_{\gamma} \setminus U_{\gamma}^1$ .

Приведем описание функционирования локальной памяти  $\text{ПЭ}(\gamma)$ ,  $1 \leq \gamma \leq \Gamma$ :

$$R_1^{(m)}(t) = \begin{cases} I^{(m)}, & \text{если } I^{(n)} = n_1, \\ R_{\Delta(N+\Delta+1)+1}^{(m)}(t-1), & \text{если } I^{(n)} = n_0; \end{cases}$$

$$R_{\alpha}^{(m)}(t + \alpha - 1) = R_{\alpha-1}^{(m)}(t + \alpha - 2), \quad 2 \leq \alpha \leq \Delta(N + \Delta + 1) + 1;$$

$$O^{(m)} = R_{\Delta(N+\Delta+1)+1}^{(m)}(t + \Delta(N + \Delta + 1));$$

$$R_1^{(b)}(t) = \begin{cases} I^{(a)}, & \text{если } I^{(m)} = m_1, \quad I^{(n)} = n_1, \\ R_1^{(a)}(t-1), & \text{если } I^{(m)} = m_1, \quad I^{(n)} = n_0, \\ R_{\Delta(N+\Delta)}^{(b)}(t-1), & \text{если } I^{(m)} = m_2 \vee m_3; \end{cases}$$

$$R_{\alpha}^{(b)}(t + \alpha - 1) = R_{\alpha-1}^{(b)}(t + \alpha - 2), \quad 2 \leq \alpha \leq \Delta(N + \Delta);$$

$$R_1^{(c)}(t) = \begin{cases} I^{(a)}/R_{\Delta(N+\Delta)}^{(b)}(t-1), & \text{если } I^{(m)} = m_2, \quad I^{(n)} = n_1, \\ R_1^{(a)}(t-1)/R_{\Delta(N+\Delta)}^{(b)}(t-1), & \text{если } I^{(m)} = m_2, \quad I^{(n)} = n_0, \\ R_{\Delta}^{(c)}(t-1), & \text{если } I^{(m)} = m_3; \end{cases}$$

$$R_1^{(a)}(t) = \begin{cases} I^{(a)} - R_{\Delta(N+\Delta)}^{(b)}(t-1)R_{\Delta}^{(c)}(t-1), & \text{если } I^{(m)} = m_3, \\ & I^{(n)} = n_1, \\ R_1^{(a)}(t-1) - R_{\Delta(N+\Delta)}^{(b)}(t-1)R_{\Delta}^{(c)}(t-1), & \text{если } I^{(m)} = m_3, \\ & I^{(n)} = n_0, \\ R_{\Delta}^{(c)}(t-1), & \text{если } I^{(m)} = m_4; \end{cases}$$

$$O^{(a)} = R_1^{(a)}(t).$$

Режим ввода начальных данных и вывода результатов спроектированного линейного спецпроцессора с заданным числом процессорных элементов для решения систем линейных алгебраических уравнений с несколькими правыми частями приведен в табл. 2.7.

Таблица 2.7

**Ввод начальных данных и вывод результатов спецпроцессора заданной ширины для решения систем линейных алгебраических уравнений**

Данные	$v_{in}$	ПЭ( $\gamma$ )	Порт	$t(v_{in})$	Изменение параметров
$m_1$	$(i_1, 1, 1)$	ПЭ(1)	$I^{(m)}$	$\Delta(i-1) + 1$	$1 \leq i \leq N$
$m_2$	$(1, j, 1)$	ПЭ(1)	$I^{(m)}$	$\Delta(N + \Delta) \times$ $\times(j-1) + 1$	$2 \leq j \leq N + M$
$m_3$	$(N + 1, j, 1)$	ПЭ(1)	$I^{(m)}$	$\Delta(N + \Delta) \times$ $\times(j-1) +$ $+ \Delta N + 1$	$2 \leq j \leq N + M$
$m_4$	$(N + 1, j, 1)$	ПЭ(1)	$I^{(m)}$	$\Delta(N + \Delta)j -$ $-\Delta^2 + 1$	$2 \leq j \leq N + M$
$n_0$	$(i, j, (\gamma - 1)\Delta + \delta)$	ПЭ( $\gamma$ )	$I^{(n)}$	$\Delta(i-1) +$ $+ \Delta(N + \Delta) \times$ $\times(j-1) +$ $+ \gamma - 1) \Delta +$ $+ \delta + 1$	$2 \leq \delta \leq \Delta,$ $1 \leq \gamma \leq \Gamma,$ $(\gamma - 1)\Delta + \delta \leq$ $\leq i \leq$ $\leq N + (\gamma - 1)\Delta +$ $+ \delta, (\gamma - 1)\Delta + \delta \leq$ $\leq j \leq$ $\leq N + M$

Данные	$v_{out}$	ПЭ( $\gamma$ )	Порт	$t(v_{out})$	Изменение параметров
$n_1$	$(i, j, (\gamma - 1)\Delta + 1)$	ПЭ( $\gamma$ )	$I^{(n)}$	$\Delta(i - 1) + \Delta(N + \Delta) \times (j - 1) + (\gamma - 1)\Delta + 2$	$1 \leq \gamma \leq \Gamma,$ $(\gamma - 1)\Delta + 1 \leq i \leq N + (\gamma - 1)\Delta + 1,$ $(\gamma - 1)\Delta + 1 \leq j \leq N + M,$ $j \neq (\gamma - 1)\Delta + 1,$ если $i = N + (\gamma - 1) \times \Delta + 1$
$a_{ij}$	$(i, j, 1)$	ПЭ(1)	$I^{(a)}$	$\Delta(i - 1) + \Delta(N + \Delta) \times (j - 1) + 1$	$1 \leq i \leq N,$ $1 \leq j \leq N + M$
$x_{ij}$	$(N + i, N + j, N)$	ПЭ( $\Gamma$ )	$O^{(a)}$	$\Delta(N + i - 1) + \Delta(N + \Delta) \times (N + j - 1) + N$	$1 \leq i \leq N,$ $1 \leq j \leq M$

На рис. 2.11 изображен вычислительный граф, представляющий параллельную форму заданной ширины  $\Delta$  алгоритма (2.45).

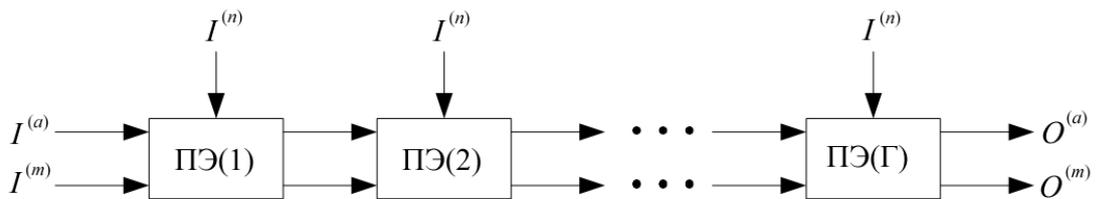


Рис. 2.11. Вычислительный граф, представляющий параллельную форму заданной ширины  $\Gamma$  для решения систем линейных алгебраических уравнений