

ОБОБЩЕНИЯ МАТЕМАТИЧЕСКОЙ МОДЕЛИ АЛГОРИТМА И ПРОЕКТИРОВАНИЕ СПЕЦПРОЦЕССОРОВ С ИСПОЛЬЗОВАНИЕМ ЕДИНОГО ТАЙМИРОВАНИЯ

3.1. СИСТЕМЫ РЕКУРРЕНТНЫХ УРАВНЕНИЙ

В главах 1 и 2 пособия рассматривались алгоритмы, заданные в виде рекуррентных уравнений вида (1.1). Далее будут рассматриваться алгоритмы более общего вида:

$$\begin{aligned} x^{(k)}(v) &= F_{\lambda}^{(k)}(x^{(1)}(v - \varphi^{(1,k)}), \dots, x^{(K)}(v - \varphi^{(K,k)})), \\ 1 \leq k \leq K, \quad 1 \leq \lambda \leq \Lambda, \quad K, \Lambda \in \mathbf{N}, \quad v \in V_{\lambda} \subset \mathbf{Z}^d. \end{aligned} \quad (3.1)$$

Обобщение заключается в том, что в системах рекуррентных уравнений (1.1) каждой переменной алгоритма $x^{(k)}$ соответствует единственный вектор зависимости $\varphi^{(k)}$. В системах рекуррентных уравнений (3.1) каждой переменной $x^{(k)}$ соответствует несколько векторов зависимостей $\varphi^{(k,j)}$, $1 \leq j \leq K$. В алгоритме (1.1) при вычислении значения всех переменных $x^{(j)}$ в точке v используется одно и то же значение $x^{(k)}(v - \varphi^{(k)})$ переменной $x^{(k)}$, а в алгоритме (3.1) – различные значения $x^{(k)}(v - \varphi^{(k,j)})$.

Множество $V = \bigcup_{\lambda=1}^{\Lambda} V_{\lambda}$ будем, как и раньше, называть *областью вычислений алгоритма* (3.1).

Операцией алгоритма (3.1) (*операцией типа λ*), приписанной точке $v \in V_{\lambda}$, назовем правило вычисления переменной $x^{(k)}$ в этой точке, т. е. операция алгоритма (3.1) определяется функцией $F_{\lambda}^{(k)}$. *Макрооперацией алгоритма* (3.1) (*макрооперацией типа λ*), приписанной точке $v \in V_{\lambda}$, назовем правило вычисления всех переменных $x^{(k)}$, $1 \leq k \leq K$, в этой точке, т. е. макрооперация – это множество операций $F_{\lambda}^{(k)}$, $1 \leq k \leq K$, приписанных точке $v \in V_{\lambda}$. Таким образом, каждой точке $v \in V_{\lambda}$ соответствует вычисление K переменных $x^{(k)}$, $1 \leq k \leq K$, определяемое макрооперацией. Другими словами, каждой точке

$v \in V_\lambda$ приписана одна макрооперация. Наличие вектора $\varphi^{(i,j)}$ в системе рекуррентных уравнений говорит о существовании информационной зависимости между переменными $x^{(i)}$ и $x^{(j)}$, т. е. означает, что результат операции, приписанной точке $v - \varphi^{(i,j)} \in V$, используется в качестве операнда в операции, приписанной точке $v \in V$. Множество всех векторов зависимостей $\varphi^{(i,j)}$, $1 \leq i, j \leq K$, входящих в систему рекуррентных уравнений (3.1), обозначим через Φ .

Если точка $v \in V$, а точка $v - \varphi^{(i,j)} \notin V$, то это означает, что значение переменной $x^{(i)}$ является *начальным данным* для вычисления переменной $x^{(j)}$ в точке $v \in V$. Если точка $v \in V$, а $v + \varphi \notin V$, то вершину $v \in V$ будем называть *граничной по направлению вектора φ* .

Поскольку в системе рекуррентных уравнений (3.1) функции $F_\lambda^{(k)}$, $1 \leq k \leq K$, не обязательно строго зависят от всех K переменных $x^{(1)}(v - \varphi^{(1,k)}), \dots, x^{(K)}(v - \varphi^{(K,k)})$, то для определенности введем в рассмотрение множество номеров переменных, которые используются в качестве аргументов для вычисления значения переменной $x^{(j)}$, $1 \leq j \leq K$, $I_j = \{i \in \mathbf{N} \mid \varphi^{(i,j)} \in \Phi\}$, и множество номеров переменных, использующих в качестве аргумента значения переменной $x^{(i)}$, $1 \leq i \leq K$, $J_i = \{j \in \mathbf{N} \mid \varphi^{(i,j)} \in \Phi\}$. Обозначим $I = \bigcup_{j=1}^K I_j$, $J = \bigcup_{i=1}^K J_i$. Заме-

тим, что множество J совпадает с множеством $\{1, \dots, K\}$, поскольку в каждой точке $v \in V$ вычисляются все K переменных алгоритма (3.1), однако множество I не всегда совпадает с множеством $\{1, \dots, K\}$, так как не обязательно все переменные алгоритма (3.1) используются в качестве аргументов при вычислении других переменных.

Граф зависимостей алгоритма (3.1) будем обозначать $G = (V, E)$, где V – область вычислений алгоритма (3.1), $E \subset V \times V$ – множество пар $(v - \varphi^{(i,j)}, v)$ информационно зависимых вершин. Каждый вектор зависимости $\varphi^{(i,j)} \in \Phi$ определяет дугу $(v - \varphi^{(i,j)}, v) \in E$, соединяющую вершины $v - \varphi^{(i,j)} \in V$ и $v \in V$. Если в вершине $v \in V_\lambda$ выполняется операция $F_\lambda^{(k)}$, то дуги, характеризующиеся векторами $\varphi^{(i,k)}$, будем называть *входящими* в вершину $v \in V_\lambda$, а дуги, характеризующиеся векторами $\varphi^{(k,j)}$, – *выходящими* из вершины $v \in V_\lambda$.

В виде (3.1) может быть представлен более широкий класс алгоритмов, чем в виде (1.1).

Методы проектирования спецпроцессоров для алгоритмов, задаваемых в виде (3.1), отличаются от рассмотренных во второй главе пособия.

Заметим, что более общий класс алгоритмов вида (3.1) часто без существенных ограничений можно свести к виду (1.1), увеличив число переменных. В этом случае при проектировании спецпроцессора, реализующего данный алгоритм, можно пользоваться методами, изложенными в первой и второй главах пособия. Однако при этом структура получаемой вычислительной системы будет более сложной. Покажем, как можно свести алгоритм вида (3.1) к алгоритму вида (1.1). Пусть переменная $x^{(k)}$, вычисляемая в вершине $v \in V_\lambda$ по правилу (3.1)

$$x^{(k)}(v) = F_\lambda^{(k)}(x^{(1)}(v - \varphi^{(1,k)}), \dots, x^{(K)}(v - \varphi^{(K,k)})), \quad (3.2)$$

используется для вычисления других переменных $x^{(j)}$, $j \in J_k = \{j \in \mathbf{N} \mid \varphi^{(k,j)} \in \Phi\}$. Тогда заменим в алгоритме (3.1) переменную $x^{(k)}$ на $|J_k|$ новых переменных $x^{(k,j)}$, $j \in J_k$, вычисляемых в вершине $v \in V_\lambda$ по правилу (1.1):

$$x^{(k,j)}(v) = F_\lambda^{(k)}(x^{(1,k)}(v - \varphi^{(1,k)}), \dots, x^{(K,k)}(v - \varphi^{(K,k)})), \quad (3.3)$$

$$j \in J_k.$$

Такое преобразование алгоритма не меняет область вычислений V и оставляет неизменным множество векторов зависимостей.

Пример 3.1. Рассмотрим алгоритм, заданный в виде (3.1):

$$\begin{aligned} x^{(1)}(i_1, i_2, i_3) &= x^{(1)}(i_1, i_2 - 1, i_3), \\ x^{(2)}(i_1, i_2, i_3) &= x^{(2)}(i_1 - 1, i_2, i_3), \\ x^{(3)}(i_1, i_2, i_3) &= x^{(1)}(i_1, i_2 - 3, i_3 + 1) + x^{(2)}(i_1, i_2, i_3 - 1) + \\ &\quad + x^{(3)}(i_1, i_2, i_3 - 1), \\ 1 \leq i_1, i_2, i_3 &\leq N. \end{aligned}$$

Здесь $\varphi^{(1,1)} = (0, 1, 0)$, $\varphi^{(1,3)} = (0, 3, -1)$, $\varphi^{(2,2)} = (1, 0, 0)$, $\varphi^{(2,3)} = (0, 0, 1)$, $\varphi^{(3,3)} = (0, 0, 1)$, $J_1 = \{1, 3\}$, $J_2 = \{2, 3\}$, $J_3 = \{3\}$.

Введя новые переменные, придем к алгоритму вида (1.1):

$$\begin{aligned}
x^{(1,1)}(i_1, i_2, i_3) &= x^{(1,1)}(i_1, i_2 - 1, i_3), \\
x^{(1,3)}(i_1, i_2, i_3) &= x^{(1,1)}(i_1, i_2 - 1, i_3), \\
x^{(2,2)}(i_1, i_2, i_3) &= x^{(2,2)}(i_1 - 1, i_2, i_3), \\
x^{(2,3)}(i_1, i_2, i_3) &= x^{(2,2)}(i_1 - 1, i_2, i_3), \\
x^{(3,3)}(i_1, i_2, i_3) &= x^{(1,3)}(i_1, i_2 - 3, i_3 + 1) + x^{(2,3)}(i_1, i_2, i_3 - 1) + \\
&\quad + x^{(3,3)}(i_1, i_2, i_3 - 1), \\
1 \leq i_1, i_2, i_3 &\leq N.
\end{aligned}$$

3.2. ТАЙМИРУЮЩАЯ ФУНКЦИЯ. ЛОКАЛЬНАЯ ПАМЯТЬ. БАЗОВЫЙ ВЫЧИСЛИТЕЛЬНЫЙ ГРАФ

Как и в первой главе пособия, введем в рассмотрение таймирующую функцию вида (2.1), удовлетворяющую условию $t(v) > t(v - \varphi^{(i,j)})$, $v, v - \varphi^{(i,j)} \in V$, $(v - \varphi^{(i,j)}, v) \in E$, $1 \leq i, j \leq K$, т. е. $\tau \in K(G)$.

Каждая вершина $v \in V$ графа $G = (V, E)$ алгоритма (3.1), размещенного в пространстве $\mathbf{Z}^d$, помечена двумя метками: макрооперацией вычисления значения переменных $x^{(k)}$, $1 \leq k \leq K$, в точке $v \in V$ и значением таймирующей функции $t(v)$, которое трактуется как номер такта, на котором по входящим в вершину дугам, соответствующим направлениям $\varphi^{(i,k)} \in \Phi$, поступают значения переменных алгоритма (3.1) $x^{(i)}(v - \varphi^{(i,k)})$, $1 \leq k \leq K$, $i \in I_k$, и осуществляется вычисление $x^{(k)}(v)$, $1 \leq k \leq K$. Введем понятие задержки вычисленного значения $x^{(k)}(v)$ в вершине v по выходящим из вершины v дугам на определенное число тактов.

Задержкой в вершине $v \in V$ по выходящей дуге $(v, v + \varphi^{(k,j)}) \in E$, $v + \varphi^{(k,j)} \in V$, определяемой вектором $\varphi^{(k,j)} \in \Phi$, $k \in I$, $j \in J_k$, будем называть целое положительное число

$$h_{\varphi^{(k,j)}}(v) = t(v + \varphi^{(k,j)}) - t(v) = \tau \cdot \varphi^{(k,j)}, \quad v \in V, \quad \varphi^{(k,j)} \in \Phi. \quad (3.4)$$

Поскольку задержка $h_{\varphi^{(k,j)}}(v)$, $v \in V$, не зависит от вершины v , то аналогично тому, как делалось в главах 1 и 2, задержку по выходящей

дуге $(v, v + \varphi^{(k,j)}) \in E$ для каждой вершины $v \in V$ будем обозначать $h_{\varphi^{(k,j)}}$.

Пометим каждую вершину $v \in V$ множеством векторов задержек $H = \{H^{(k)} = (h_{\varphi^{(k,1)}}, \dots, h_{\varphi^{(k,K)}}), k \in I\}$. Вектор задержек является частично неопределенным. Если не существует дуги, выходящей из вершины $v \in V$ и характеризующейся вектором $\varphi^{(k,j_0)}$, $j_0 \notin J_k$, то в векторе $H^{(k)}$ на месте координаты j_0 будем ставить прочерк.

Множество значений таймирующей функции $t(v)$, соответствующих всем вершинам $v \in V$, обозначим $T = \{t(v), v \in V\}$.

Таким образом, получен базовый вычислительный граф $G = (V, E, T, H)$, где V, E – множество вершин и множество дуг графа алгоритма (3.1), вершины графа помечены выполняемыми операциями алгоритма, значениями таймирующих функций из множества $T = \{t(v), v \in V\}$ и векторами задержек из множества $H = \{H^{(k)}, k \in I\}$.

Опишем структуру процессорных элементов построенной виртуальной вычислительной системы, реализующей алгоритм (3.1).

Каждый $\Pi\Theta(v)$, $v \in V_\lambda$, $1 \leq \lambda \leq \Lambda$, имеет набор входов $\{I_{\varphi^{(i,k)}}^{(i,k)}, 1 \leq i \leq K, k \in I\}$ и набор выходов $\{O_{\varphi^{(k,j)}}^{(k,j)}, k \in I, j \in J_k\}$. Число входов равно числу операндов макрооперации $F_\lambda^{(k)}$, $1 \leq k \leq K$, по вычислению $x^{(1)}(v), \dots, x^{(K)}(v)$. Число выходов равно числу операций, использующих результат макрооперации $F_\lambda^{(k)}$, $1 \leq k \leq K$, в качестве операнда. Каждый $\Pi\Theta(v)$ БВГ выполняет только одно вычисление на такте $t(v)$. Для этого в $\Pi\Theta(v)$ имеется соответствующее устройство, позволяющее выполнить эту операцию над входными данными, поступившими в $\Pi\Theta(v)$ БВГ. Каждый $\Pi\Theta(v)$, $v \in V$, БВГ имеет K блоков линейно связанных ячеек локальной памяти. Из них $|I|$ блоков для хранения значений переменных $x^{(k)}$, $k \in I$, длина каждого такого блока — $\max_{j \in J_k} h_{\varphi^{(k,j)}}: \{R_\alpha^{(k)}(t), 1 \leq \alpha \leq \max_{j \in J_k} \{h_{\varphi^{(k,j)}}\}\}$, $k \in I$, и $K - |I|$ дополнительных ячеек памяти для хранения значений переменных $x^{(k)}$, $k \notin I$ (ни одна из переменных алгоритма (3.1) не использует значения переменных $x^{(k)}$, $k \notin I$, для вычисления своего значения): $R_1^{(k)}(t)$, $k \notin I$. Все ячейки памяти меняют свое содержимое в зависимости от номера такта t . В первую ячейку памяти в момент времени $t = t(v)$ записывается результат выполнения операции $F_\lambda^{(k)}$: $R_1^{(k)}(t) =$

$= F_{\lambda}^{(k)}(I_{\varphi^{(1,k)}}^{(1,k)}, \dots, I_{\varphi^{(K,k)}}^{(K,k)})$ (функции $F_{\lambda}^{(k)}$, $1 \leq k \leq K$, не обязательно строго зависят от всех K переменных). На каждом следующем такте происходит перезапись содержимого каждой ячейки памяти с номером $\alpha - 1$ в ячейку памяти с номером α , $2 \leq \alpha \leq \max_{j \in J_k} h_{\varphi^{(k,j)}}: R_{\alpha}^{(k)}(t + \alpha - 1) = R_{\alpha-1}^{(k)}(t + \alpha - 2)$, $1 \leq k \leq K$. На каждый выход $O_{\varphi^{(k,j)}}^{(k,j)}$, $k \in I$, $j \in J_k$, в момент времени $t(v) + h_{\varphi^{(k,j)}}$ подается содержимое ячейки памяти с номером $h_{\varphi^{(k,j)}}: O_{\varphi^{(k,j)}}^{(k,j)} = R_{h_{\varphi^{(k,j)}}}^{(k)}(t(v) + h_{\varphi^{(k,j)}} - 1)$.

Если для некоторого $k \in \{1, \dots, K\}$, $\max_{j \in J_k} h_{\varphi^{(k,j)}} = 1$, то содержимое ячейки памяти $R_1^{(k)}(t)$ в момент времени $t + 1$ подается на $|J_k|$ выходов $O_{\varphi^{(k,j)}}^{(k,j)}$, $j \in J_k$, $t = t(v)$.

Процессорные элементы стыкуются друг с другом по входным и выходным каналам таким образом, что выход $O_{\varphi^{(k,j)}}^{(k,j)}$ ПЭ(v), $v \in V$, совмещается со входом $I_{\varphi^{(k,j)}}^{(k,j)}$ ПЭ($v + \varphi^{(k,j)}$), $v + \varphi^{(k,j)} \in V$, $\varphi^{(k,j)} \in \Phi$. Во времени такая стыковка означает следующее: если в момент времени $t = t(v) + h_{\varphi^{(k,j)}}$ на выходе $O_{\varphi^{(k,j)}}^{(k,j)}$ ПЭ(v) появляется значение переменной $x^{(k)}(v)$, то в этот же момент времени t оно подается на вход $I_{\varphi^{(k,j)}}^{(k,j)}$ ПЭ($v + \varphi^{(k,j)}$).

Приведем описание функционирования локальной памяти ПЭ:

$$\begin{aligned} R_{\lambda}(v): \{ & R_1^{(k)}(t) = F_{\lambda}^{(k)}(I_{\varphi^{(1,k)}}^{(1,k)}, \dots, I_{\varphi^{(K,k)}}^{(K,k)}), \\ & R_{\alpha}^{(k)}(t + \alpha - 1) = R_{\alpha-1}^{(k)}(t + \alpha - 2), \quad 2 \leq \alpha \leq \max_{j \in J_k} h_{\varphi^{(k,j)}}, \\ & O_{\varphi^{(k,j)}}^{(k,j)} = R_{h_{\varphi^{(k,j)}}}^{(k)}(t(v) + h_{\varphi^{(k,j)}} - 1), \quad j \in J_k, \quad 1 \leq k \leq K \}, \end{aligned} \quad (3.5)$$

$$v \in V_{\lambda}, \quad 1 \leq \lambda \leq \Lambda.$$

3.3. ОТОБРАЖЕНИЕ В ПРОСТРАНСТВА МЕНЬШИХ РАЗМЕРНОСТЕЙ

Отображение d -мерных алгоритмов, заданных в виде (3.1), в r -мерную архитектуру будем задавать линейной функцией размещения $\pi: \mathbf{Z}^d \rightarrow \mathbf{Z}^r$, которая не порождает связей, зависящих от размерности задачи, т. е. $\|\pi(\varphi^{(i,j)})\|_{\infty} \leq \text{const}$. Выбранная функция размещения и таймирующая функция должны удовлетворять условию совместимости (2.8).

Теоремы, приведенные в п. 2.3, остаются справедливыми и для алгоритмов, заданных в виде (3.1).

Описание структуры процессорных элементов вычислительной системы производится так же, как и в первой главе пособия. Описание функционирования локальной памяти процессорного элемента $\text{ПЭ}(\tilde{v})$, $\tilde{v} = \pi(v)$, $v \in V$, вычислительной системы можно получить из описания функционирования локальной памяти $\text{ПЭ}(v)$ виртуальной вычислительной системы (БВГ) следующим образом. Каждый вектор $\varphi^{(i,j)}$, не принадлежащий ядру оператора π , т. е. $\pi(\varphi^{(i,j)}) \neq 0$, следует заменить на вектор $\pi(\varphi^{(i,j)})$. Если для некоторого $i_0 \in I$ вектор $\varphi^{(i_0,j)}$, $j \in J_{i_0}$, принадлежит ядру оператора π , т. е. $\pi(\varphi^{(i_0,j)}) = 0$, то следует заменить $I_{\varphi^{(i_0,j)}}^{(i_0,j)}$ на $R_{h_{\varphi^{(i_0,j)}}}^{(i_0)}(t-1)$. Если для некоторого $j_0 \in \{1, \dots, K\}$ вектор $\varphi^{(i,j_0)}$, $i \in I_{j_0}$ принадлежит ядру оператора π , т. е. $\pi(\varphi^{(i,j_0)}) = 0$, то равенство, соответствующее подаче результата вычислений на выход $O_{\varphi^{(i,j_0)}}^{(i,j_0)}$, следует опустить.

Если для алгоритма, заданного в виде системы рекуррентных уравнений вида (3.1), при отображении π выполняется необходимое условие введения управления (2.15), то введение управляющих меток следует производить таким же образом, как описано в п. 2.4.

3.4. ПРОЕКТИРОВАНИЕ МАТРИЧНЫХ ПРОЦЕССОРОВ НА СБИС С АРХИТЕКТУРОЙ КОЛЬЦА

Метод преобразования сложных алгоритмов к виду (1.1) путем увеличения числа переменных, описанный выше для алгоритмов (3.1), можно применить и для других классов алгоритмов вычислительной математики. Проиллюстрируем полезность такого преобразования на примере проектирования систолического массива для численного решения нестационарного уравнения теплопроводности.

При проектировании матричных процессоров на СБИС с архитектурой кольца, состоящего из фиксированного числа Δ процессорных элементов, обычно применяется ЛПарГПос стратегия пространственно-временного отображения исходного алгоритма, заданного в виде системы рекуррентных уравнений (1.1). Однако многие важные для при-

ложений задачи решаются с помощью алгоритмов, которые представлены в другом виде. Примером такого алгоритма является алгоритм численного решения нестационарного уравнения теплопроводности

$$\frac{\partial u}{\partial t} = a \frac{\partial^2 u}{\partial x^2} + f(x, t), \quad 0 < x < 1, \quad 0 < t < T,$$

с начальными $u(x, 0) = f^{(0)}(x)$, $0 \leq x \leq 1$, и краевыми $u(0, t) = f_0(t)$, $u(1, t) = f_1(t)$, $0 \leq t \leq 1$, условиями ($f_0(0) = f^{(0)}(0)$, $f_1(0) = f^{(0)}(1)$). Этот алгоритм получен с использованием двухслойной явной разностной схемы:

$$\begin{aligned} y(i, 0) &= f^{(0)}(ih), \quad 0 \leq i \leq N, \\ y(0, j) &= f_0(j\tau), \quad y(N, j) = f_1(j\tau), \quad 0 \leq j \leq M, \\ f_0(0) &= f^{(0)}(0), \quad f_1(0) = f^{(0)}(1), \\ y(i, j) &= (1 - \frac{2a\tau}{h^2})y(i, j-1) + \frac{a\tau}{h^2}(y(i+1, j-1) + \\ &+ y(i-1, j-1)) + f(i, j-1), \quad 1 \leq i \leq N-1, \quad 1 \leq j \leq M, \\ N \cdot h &= 1, \quad M \cdot \tau = T. \end{aligned} \tag{3.6}$$

В алгоритме при вычислении значения $y(v)$ в точке $v = (i, j) \in V^0 = \{(i, j) \in \mathbf{Z}^2 \mid 1 \leq i \leq N-1, 1 \leq j \leq M\}$ используются значения этой же переменной в трех соседних точках индексного пространства V^0 : $y(v - \varphi^{(1)})$, $y(v - \varphi^{(2)})$, $y(v - \varphi^{(3)})$, $\varphi^{(1)} = (-1, 1)$, $\varphi^{(2)} = (0, 1)$, $\varphi^{(3)} = (1, 1)$.

Модификацию такого алгоритма выполним, используя следующее утверждение.

Пусть вычисление переменной y в точке v индексного пространства $V^0 \subset \mathbf{Z}^n$ использует значения, присвоенные этой переменной в k ($k \geq 2$) других точках индексного пространства V^0 :

$$\begin{aligned} y(v) &= F(y(v - \varphi^{(1)}), y(v - \varphi^{(2)}), \dots, y(v - \varphi^{(k)})), \\ v, v - \varphi^{(i)} &\in V^0, \quad 1 \leq i \leq k. \end{aligned}$$

Тогда введение k новых переменных $y^{(1)}, y^{(2)}, \dots, y^{(k)}$, вычисляемых по одному и тому же правилу

$$\begin{aligned} y^{(i)}(v) &= F(y^{(1)}(v - \varphi^{(1)}), y^{(2)}(v - \varphi^{(2)}), \dots, y^{(k)}(v - \varphi^{(k)})), \\ v, v - \varphi^{(i)} &\in V^0, \quad 1 \leq i \leq k, \end{aligned}$$

приводит к алгоритму для вычисления $y(v) = y^{(1)}(v) = y^{(2)}(v) = \dots = y^{(k)}(v)$ вида (1.1).

Следуя этому утверждению, из алгоритма (3.6) получим новый алгоритм вида (1.1) численного решения нестационарного уравнения теплопроводности:

$$\begin{aligned} y^{(k)}(v) &= (1 - \frac{2a\tau}{h^2})y^{(2)}(v - \varphi^{(2)}) + \frac{a\tau}{h^2}(y^{(1)}(v - \varphi^{(1)}) + \\ &+ y^{(3)}(v - \varphi^{(3)})) + f(v - \varphi^{(2)}), \quad v, v - \varphi^{(k)} \in V^0, \quad 1 \leq k \leq 3, \\ y^{(2)}(i, 0) &= f^{(0)}(ih), \quad 0 \leq i \leq N, \\ y^{(3)}(0, j) &= f_0(j\tau), \quad y^{(1)}(N, j) = f_1(j\tau), \quad 0 \leq j \leq M, \\ y^{(3)}(0, 0) &= y^{(2)}(0, 0), \quad y^{(1)}(N, 0) = y^{(2)}(N, 0). \end{aligned} \quad (3.7)$$

Применим теперь к алгоритму (3.6) ЛПарГПос стратегию пространственно-временного отображения.

Функцию размещения по Δ процессорам операций, приписанных в соответствии с определением (3.7) каждой точке (i, j) области вычислений, определим следующим образом:

$$\hat{\pi}(i, j) = (j - 1) \bmod \Delta + 1, \quad 1 \leq j \leq M,$$

$$M = \Gamma \cdot \Delta, \quad \Gamma \in \mathbf{N}, \quad (i, j) \in V^0.$$

При таком размещении операций по процессорам ввод граничных значений требуется выполнять в каждом процессорном элементе ПЭ(1), ПЭ(2), ..., ПЭ(Δ). Чтобы локализовать ввод граничных значений в одном «крайнем» процессорном элементе ПЭ(1), расширим область вычислений. В добавленных точках

$$v \in V_{\gamma}^{-} = \{(i, j) \in \mathbf{Z}^2 \mid 2 - \Delta \leq i \leq 0, \quad 1 + (\gamma - 1)\Delta \leq j \leq \gamma\Delta - 1\},$$

$$v \in V_{\gamma}^{+} = \{(i, j) \in \mathbf{Z}^2 \mid N \leq i \leq N - 2 + \Delta,$$

$$1 + (\gamma - 1)\Delta \leq j \leq \gamma\Delta - 1\}, \quad 1 \leq \gamma \leq \Gamma,$$

доопределим алгоритм (3.7) операциями «переприсваивания»: $y^{(1)}(v) = y^{(1)}(v - \varphi^{(1)})$, $v \in V_{\gamma}^{+}$, $y^{(3)}(v) = y^{(3)}(v - \varphi^{(3)})$, $v \in V_{\gamma}^{-}$, и осуществим транспортировку граничных значений от точек $(i, 1 + (\gamma - 1)\Delta)$, $1 \leq \gamma \leq \Gamma$, $-(\Delta - 2) \leq i \leq 0$, $N \leq i \leq N + \Delta - 2$, отображаемых функцией $\hat{\pi}$ в ПЭ(1), к точкам, где эти значения используются в качестве аргументов.

Обозначим:

$$\begin{aligned}
V_\gamma^0 &= \{(i, j) \in \mathbf{Z}^2 \mid 1 \leq i \leq N-1, 1 + (\gamma-1)\Delta \leq j \leq \gamma\Delta\}; \\
V_\gamma &= V_\gamma^- \cup V_\gamma^0 \cup V_\gamma^+ = \{(i, j) \in \mathbf{Z}^2 \mid 2 - \Delta \leq i \leq N - 2 + \Delta, \\
&\quad 1 + (\gamma-1)\Delta \leq j \leq \gamma\Delta\}, \quad 1 \leq \gamma \leq \Gamma; \\
V &= \bigcup_{\lambda=1}^{\Gamma} V_\lambda = \bigcup_{\lambda=1}^{\Gamma} V_\lambda^- \cup V_\lambda^0 \cup V_\lambda^+ = \{(i, j) \in \mathbf{Z}^2 \mid -(\Delta-2) \leq i \leq \\
&\quad \leq N + \Delta - 2, 1 \leq j \leq M\}.
\end{aligned}$$

Получим модифицированный алгоритм численного решения нестационарного уравнения теплопроводности:

$$\begin{aligned}
y^{(k)}(v) &= \left(1 - \frac{2a\tau}{h^2}\right) y^{(2)}(v - \varphi^{(2)}) + \frac{a\tau}{h^2} \left(y^{(1)}(v - \varphi^{(1)}) + \right. \\
&\quad \left. + y^{(3)}(v - \varphi^{(3)}) \right) + f(v - \varphi^{(2)}), \quad v \in V_\gamma^0, \\
y^{(1)}(v) &= y^{(1)}(v - \varphi^{(1)}), \quad v \in V_\gamma^+, \\
y^{(3)}(v) &= y^{(3)}(v - \varphi^{(3)}), \quad v \in V_\gamma^-, \\
y^{(2)}(i, 0) &= f^{(0)}(ih), \\
y^{(1)}(N + j - 1 - (\gamma-1)\Delta, (\gamma-1)\Delta) &= f_1(j\tau), \\
y^{(1)}(N, 0) &= y^{(2)}(N, 0), \\
y^{(3)}(1 - j + (\gamma-1)\Delta, (\gamma-1)\Delta) &= f_0(j\tau), \\
y^{(3)}(0, 0) &= y^{(2)}(0, 0), \\
1 \leq k \leq 3, \quad 0 \leq i \leq N, \quad 1 + (\gamma-1)\Delta \leq j \leq \gamma\Delta, \quad 1 \leq \gamma \leq \Gamma,
\end{aligned} \tag{3.8}$$

где $\Phi = \{\varphi^{(1)} = (-1, 1), \varphi^{(2)} = (0, 1), \varphi^{(3)} = (1, 1)\}$ – множество векторов зависимостей.

Теперь вершины ввода граничных значений отобразятся в $\Pi\Theta(1)$:

$$\hat{\pi}(-j+2+(\gamma-1)\Delta, 1+(\gamma-1)\Delta) = \hat{\pi}(N+j-2-(\gamma-1)\Delta, 1+(\gamma-1)\Delta) = 1.$$

Граф зависимостей $G = (V, E)$ алгоритма (3.8), где $E = \{(v_1, v_2) \in V \times V, v_2 = v_1 + \varphi, \varphi \in \Phi\}$, является строго направленным, так как конус допустимых направлений графа G $K(G) = \{\tau \in \mathbf{Z}^2 \mid \tau \cdot \varphi > 0, \forall \varphi \in \Phi\}$ не пуст. Принадлежность вектора $\tau = (\tau_1, \tau_2)$ конусу $K(G)$ определяется неравенством $\tau_2 \geq |\tau_1| + 1$.

Зададим невырожденное отображение $\pi : \mathbf{Z}^2 \rightarrow \mathbf{Z}^1$, удовлетворяющее условию $\pi(\varphi) \in \{(0), (1)\}$, $\varphi \in \Phi$, обеспечивающее локальность и

отсутствие обратных связей между процессорными элементами (за исключением связей между крайними процессорными элементами $\Pi\Theta(\Delta)$ и $\Pi\Theta(1)$). Этому условию удовлетворяет только оператор $\pi(i, j) = (j)$, $(i, j) \in \mathbf{Z}^2$. При таком выборе оператора область вычислений $V^0 = \bigcup_{\gamma=1}^{\Gamma} V_{\gamma}^0$ и области $V^- = \bigcup_{\gamma=1}^{\Gamma} V_{\gamma}^-$, $V^+ = \bigcup_{\gamma=1}^{\Gamma} V_{\gamma}^+$, в которых осуществляется транспортировка граничных значений, отображаются в одно и то же множество $\pi(V) = \{(j) \in \mathbf{Z}^1 \mid 1 \leq j \leq M\}$.

Так как область V состоит из объединения областей V^- , V^+ , V^0 , в каждой из которых вершинам соответствуют различные типы операций, то при размещении $\hat{\pi}$ этих операций по процессорам каждый процессорный элемент будет выполнять операции трех типов. Для того чтобы различать эти операции при их выполнении, введем в алгоритм управляющую переменную m , вычисляемую рекуррентно, с начальными значениями-метками m_{-1} , m_0 , m_{+1} для вершин из множеств V^- , V^0 , V^+ соответственно. Транспортировку этих меток будем осуществлять согласно следующему алгоритму:

$$\begin{aligned} m(v) &= m(v - \varphi^{(2)}), \quad v \in V, \\ m(i, 0) &= m_{-1}, \quad -\Delta + 2 \leq i \leq 0, \\ m(i, 0) &= m_0, \quad 1 \leq i \leq N - 1, \\ m(i, 0) &= m_{+1}, \quad N \leq i \leq N + \Delta - 2. \end{aligned}$$

Заметим, что введение управляющей переменной не расширило индексного пространства V и не сузило конус допустимых направлений $K(G)$.

Следуя ЛПарГПос стратегии, определим множества

$$\begin{aligned} V_{\gamma}^{\delta} &= \{v \in V_{\gamma} \mid \pi(v) = j = (\gamma - 1)\Delta + \delta\}, \\ V^{\delta} &= \bigcup_{\gamma=1}^{\Gamma} V_{\gamma}^{\delta}, \quad 1 \leq \gamma \leq \Gamma, \quad 1 \leq \delta \leq \Delta, \end{aligned}$$

и кусочно-аффинную таймирующую функцию

$$\begin{aligned} t(v) &= \sum_{\gamma=1}^{\Gamma} (c_{\gamma} + \tau \cdot v) 1_{V_{\gamma}}(v), \quad v \in V, \\ \text{где } 1_{V_{\gamma}}(v) &= \begin{cases} 1, & v \in V_{\gamma}, \\ 0, & v \notin V_{\gamma}. \end{cases} \end{aligned}$$

Константы c_γ и вектор τ выберем из условий $0 < |\tau_1| \leq \tau_2 - 1$ и требования минимальности времени (количества тактов), необходимого для параллельной реализации алгоритма (3.8). Это время определяется функцией

$$T(\tau) = \max_{v \in V} t(v) - \min_{v \in V} t(v) + \max_{1 \leq r \leq 3} \tau \cdot \varphi^{(r)}, \quad \varphi^{(r)} \in \Phi.$$

При фиксированном $\tau \in K(G)$ $\min_{c_1, \dots, c_\Gamma} T(\tau)$ достигается при выборе констант $c_1, c_2, \dots, c_\Gamma$ из рекуррентного соотношения (2.32):

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

где $x_\gamma = \min_{\varphi \in \Phi_\gamma} \tau \cdot \varphi - 1$, $\Phi_\gamma = \Phi$, $y_\gamma = \max_{1 \leq \delta \leq \Delta} \{\max_{v \in V_\gamma^\delta} \tau \cdot v - \min_{v \in V_{\gamma+1}^\delta} \tau \cdot v\} + 1$.

В данном случае

$$-x_\gamma = |\tau_1| - \tau_2 + 1, \quad y_\gamma = |\tau_1|(N + 2\Delta - 4) - \tau_2\Delta + 1,$$

$$c_\gamma = c_1 + (\gamma - 1) \max\{|\tau_1| - \tau_2 + 1, |\tau_1|(N + 2\Delta - 4) - \tau_2\Delta + 1\}, \quad 1 \leq \gamma \leq \Gamma,$$

$$T(\tau_1, \tau_2) = (\Gamma - 1) \max\{|\tau_1| - \tau_2 + 1, |\tau_1|(N + 2\Delta - 4) - \tau_2\Delta + 1\} + M\tau_2 + |\tau_1|(N + \Delta - 2).$$

Функция $T(\tau_1, \tau_2)$ с учетом ограничения $0 < |\tau_1| \leq \tau_2 - 1$ принимает минимальное значение при выборе $|\tau_1| = 1$, $\tau_2 = 2$:

$$\min_{0 < |\tau_1| \leq \tau_2 - 1} T(\tau_1, \tau_2) = T(1, 2) = M \left(\frac{N - 3}{\Delta} + 2 \right) + \Delta + 1.$$

Положив для определенности $c_1 = \Delta - 3$, $\tau_1 = 1$, получим следующее утверждение.

Таймирующая функция

$$t(i, j) = \sum_{\gamma=1}^{\Gamma} (i + 2j + \Delta - 3 + (\gamma - 1)(N - 3)) 1_{V_\gamma}(i, j), \quad (i, j) \in V,$$

в условиях ЛПарГПос стратегии обеспечивает построение параллельной формы алгоритма (3.8) с заданной шириной Δ и на Δ локально-связанных процессорах $(\Pi\Theta(\delta), 1 \leq \delta \leq \Delta)$ позволяет решить задачу

за время $T(1, 2) = M \left(\frac{N-3}{\Delta} + 2 \right) + \Delta + 1$ тактов. Наименьшее значение $T(1, 2) = 2(\sqrt{M(N-3)} + M) + 1$ достигается при $\Delta = \sqrt{M(N-3)}$ ($\sqrt{M(N-3)} \in \mathbf{N}$).

Для определения объема локальной памяти процессорных элементов вычислим задержки $h_{\varphi^{(k)}}(v)$ в вершинах $v \in V$ по дугам $(v, v + \varphi^{(k)}) \in E$, $\varphi^{(k)} \in \Phi$, $1 \leq k \leq 3$.

Если $v, v + \varphi \in V_\gamma$, $1 \leq \gamma \leq \Gamma$, то $h_\varphi(v) = c_\gamma + \tau(v + \varphi) - (c_\gamma + \tau \cdot v) = \tau \cdot \varphi$. В этом случае $h_{\varphi^{(1)}}(v) = 1$, $h_{\varphi^{(2)}}(v) = 2$, $h_{\varphi^{(3)}}(v) = 3$.

Если $v \in V_\gamma$, $v + \varphi \in V_{\gamma+1}$, $1 \leq \gamma \leq \Gamma - 1$, то $h_\varphi(v) = c_{\gamma+1} - c_\gamma + \tau \cdot \varphi$. В этом случае $h_{\varphi^{(1)}}(v) = N - 2$, $h_{\varphi^{(2)}}(v) = N - 1$, $h_{\varphi^{(3)}}(v) = N$.

Процессы приема, вычисления и передачи информации, происходящие в процессорном элементе, представим в виде описания функционирования локальной памяти процессорного элемента. Локальная память процессорных элементов в проектируемом спецпроцессоре не зависит ни от параметров задачи M и N , ни от числа Δ процессорных элементов. В каждом процессорном элементе (рис. 3.1), кроме арифметико-логического устройства (АЛУ), имеются два блока ячеек локальной памяти $R_1^{(y)}$, $R_2^{(y)}$, $R_3^{(y)}$; $R_1^{(m)}$, $R_2^{(m)}$. Количество ячеек в блоке определяется максимальной величиной задержки данных ($y^{(k)}$ и m) в процессорном элементе. Каждый процессорный элемент, кроме четырех входов $I^{(1)}$, $I^{(2)}$, $I^{(3)}$, $I^{(m)}$ и четырех выходов $O^{(1)}$, $O^{(2)}$, $O^{(3)}$, $O^{(m)}$, предназначенных для передачи промежуточных результатов вычислений в соседний процессорный элемент, имеет один дополнительный вход $I^{(f)}$ для ввода значений функции f и один дополнительный выход $O^{(y)}$ для вывода промежуточных значений y . Содержимое ячеек памяти в каждый момент времени t определяется следующим алгоритмом:

$$R_1^{(y)}(t) = \begin{cases} I^{(3)}, & I^{(m)} = m_{-1}, \\ (1 - 2\alpha)I^{(2)} + \alpha(I^{(1)} + I^{(3)}) + I^{(f)}, & I^{(m)} = m_0, \\ I^{(1)}, & I^{(m)} = m_{+1}; \end{cases}$$

$$R_2^{(y)}(t+1) = R_1^{(y)}(t), \quad R_3^{(y)}(t+2) = R_2^{(y)}(t+1);$$

$$O^{(1)} = R_3^{(1)}(t+2), \quad O^{(2)} = R_2^{(2)}(t+1), \quad O^{(3)} = O^{(y)} = R_1^{(3)}(t);$$

$$R_1^{(m)}(t) = I^{(m)}, \quad R_2^{(m)}(t+1) = R_1^{(m)}(t), \quad O^{(m)} = R_2^{(m)}(t);$$

$$\alpha = aN^2T/M.$$

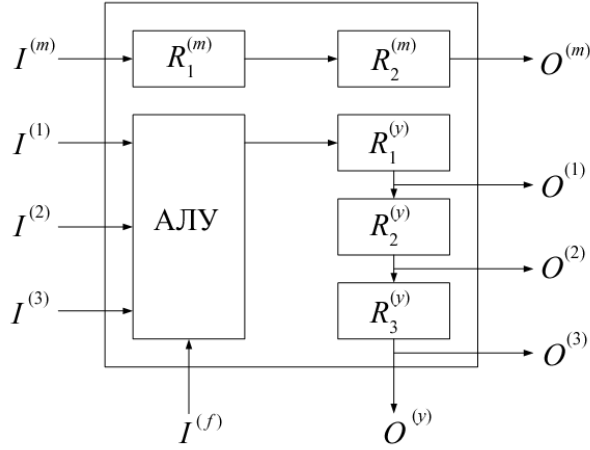


Рис. 3.1. Структурная схема процессорного элемента

Наличие (в соответствии с ЛПарГПос стратегией) обратной связи ПЭ(Δ) с ПЭ(1) предполагает дополнительную задержку передаваемых данных на $c_{\gamma+1} - c_{\gamma} = N - 3$ такта. Ввод данных в спецпроцессор представлен в табл 3.1.

Таблица 3.1

Ввод начальных данных и управляющих меток в спецпроцессор для численного решения нестационарного уравнения теплопроводности

Данные	ПЭ	Порт	Время (такт)	Диапазон изменения параметров
$f^{(0)}\left(\frac{i}{N}\right)$	ПЭ(1)	$I^{(2)}$	$i + \Delta - 1$	$1 \leq i \leq N - 1$
$f_0\left(j\frac{T}{M}\right)$	ПЭ(1)	$I^{(3)}$	$-j + \Delta +$ $+ (\gamma - 1)(N + 3\Delta - 3)$	$(\gamma - 1)\Delta \leq j \leq \gamma\Delta - 1,$ $1 \leq \gamma \leq \Gamma$
$f_1\left(j\frac{T}{M}\right)$	ПЭ(1)	$I^{(1)}$	$j + \gamma(N + \Delta - 3) + 1$	$(\gamma - 1)\Delta \leq j \leq \gamma\Delta - 1,$ $1 \leq \gamma \leq \Gamma$
$f(i, j - 1)$ $j = (\gamma - 1)\Delta + \delta$	ПЭ(δ)	$I^{(f)}$	$i + 2j + \Delta - 3 +$ $+ (\gamma - 1)(N - 3)$	$1 \leq i \leq N - 1,$ $1 \leq \delta \leq \Delta, 1 \leq \gamma \leq \Gamma$
m_{-1}	ПЭ(1)	$I^{(m)}$	$i + \Delta - 1$	$-(\Delta - 2) \leq i \leq 0$
m_0	ПЭ(1)	$I^{(m)}$	$i + \Delta - 1$	$1 \leq i \leq N - 1$
m_{+1}	ПЭ(1)	$I^{(m)}$	$i + \Delta - 1$	$N \leq i \leq N + \Delta - 2$

Таким образом, спроектирован систолический массив (рис. 3.2) с архитектурой кольца, состоящего из заданного числа Δ однотипных процессорных элементов.

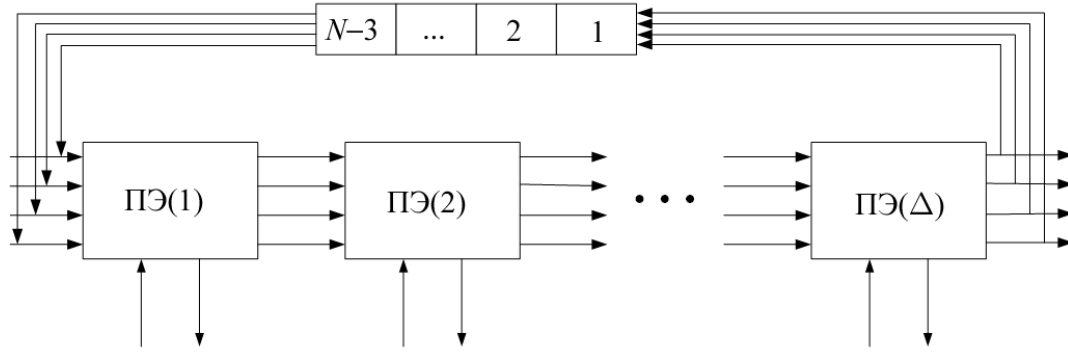


Рис. 3.2. Структурная схема спроектированного спецпроцессора

Массив предназначен для численного решения явным методом сеток нестационарного уравнения теплопроводности. Локальная память процессорных элементов не зависит ни от параметров N и M , определяющих число узлов сетки, ни от числа процессоров Δ .

Время решения задачи определяется функцией $\frac{M(N-3)}{\Delta} + \Delta + 2M + 1$, которая принимает наименьшее значение при $\Delta = \sqrt{M(N-3)}$ (в предположении, что $\sqrt{M(N-3)}$ — целое число).