

ОПТИЧЕСКИЕ ИНФОРМАЦИОННЫЕ  
ТЕХНОЛОГИИ И СИСТЕМЫ

УДК 621.382

С. В. Пискунов

(Новосибирск)

ПОСТРОЕНИЕ МНОГОСЛОЙНЫХ  
ЭЛЕКТРООПТИЧЕСКИХ СТРУКТУР

На базе модели распределенных вычислений, называемой алгоритмом параллельных подстановок, разработан метод перехода от двумерной вычислительной структуры к многослойной (трехмерной) структуре, реализующей, по сути, тот же алгоритм переработки информации. Показано, что при реализации трехмерной структуры в рамках электрооптической схемотехники метод обеспечивает возможность как выбирать сложность отдельной ячейки и топологию электрических связей между ячейками, расположенными в одном и том же слое, так и переводить существенную часть электрических связей между ячейками двумерной структуры в оптические межслойные связи трехмерной структуры. Изложение метода сопровождается примером построения четырехслойного параллельного сумматора многих чисел.

**Введение.** Вычислительные устройства с существенно однородной структурой, представляющие собой большие совокупности маломощных вычислителей (ячеек), работающих параллельно, широко распространены и применяются для выполнения алгоритмов распознавания образов, быстрой арифметики, матричной алгебры, преобразования сигналов и т. п. Они обладают высокой производительностью и технологичностью. Дальнейший прогресс в построении таких устройств связывают с использованием оптики. Оптические методы массовой обработки дискретной информации основаны на теневой логике [1], электрооптической схемотехнике [2, 3] и символических подстановках [4]. Если микроэлектронные устройства, как правило, двумерны, то применение оптики открывает возможность построения трехмерных устройств и, прежде всего, позволяет решить задачу массовых обменов информацией между ячейками устройства.

Среди перечисленных способов обработки информации с использованием оптики выделим электрооптическую схемотехнику, так как она позволяет органично сочетать микроэлектронику и оптику и в связи с этим открывает путь к построению трехмерной СБИС. Электрооптическая СБИС представляет собой многослойную структуру. Слои плоские, расположены один над другим, связи между слоями оптические, внутри слоев — электрические. Электрооптическая схемотехника, ориентированная на интегральное исполнение, разработана в [5, 6].

В данной работе предлагается метод отображения двумерной однородной вычислительной структуры в эквивалентную (выполняющую тот же алгоритм) трехмерную структуру. Этот метод (будем называть его методом расслоения) при реализации трехмерной структуры в рамках электрооптической схемотехники обеспечивает возможность как выбирать сложность отдельной ячейки и топологию электрических связей между ячейками, расположенными в одном и том же слое, так и, самое главное, переводить существенную часть

электрических связей между ячейками двумерной структуры в оптические межслойные трехмерной структуры.

Метод расслоения основан на модели распределенных вычислений, называемой алгоритмом параллельных подстановок [7—9]. В этой модели основным средством преобразования информации, представленной в виде конечного массива клеток  $W$ , характеризуемых именем из множества имен  $M$  и состоянием из конечного алфавита  $A$ , является параллельная подстановка  $\Theta$ . Она задается выражением  $S_1 * S_2 \rightarrow S_3$ , где  $S_i$  называются конфигурациями.  $S_i$  — отображение, ставящее в соответствие каждой конкретной клетке  $m^0 \in M$  некоторое множество клеток, обозначим его  $S_i(m^0)$ . При использовании  $m^0$  в качестве аргумента подстановка  $\Theta$  порождает микрооперацию, применение которой к  $W$  состоит в отыскании в нем объединения множеств  $S_1(m^0)$  и  $S_2(m^0)$  и, если таковое находится, в замене множества  $S_1(m^0)$  на множество  $S_3(m^0)$ . Алгоритм параллельных подстановок  $\Phi$  (АППФ) образует конечная совокупность параллельных подстановок. АППФ осуществляет итерационную процедуру преобразования клеточного массива. На каждом шаге итерации одновременно и повсеместно применяются все микрооперации, порожденные подстановками из АППФ. Если для каждой  $\Theta$  из АППФ справедливо условие, что в любой ее микрооперации для любого  $m^0$  множество клеток  $S_1(m^0)$  имеет точно такие имена, как и у клеток из множества  $S_3(m^0)$ , и наоборот, то АППФ называется микропрограммой  $\Phi$ , а составляющие ее параллельные подстановки — микрокомандами. В самой микрокоманде  $S_1 * S_2$  называется ее левой частью,  $S_3$  — правой,  $S_1$  — базой,  $S_2$  — контекстом.

При разработке метода расслоения будем использовать возможность построения сети автоматов, интерпретирующей параллельную микропрограмму. Техническую реализацию сети будем называть структурой.

1. Интерпретация параллельной микропрограммы сетью автоматов.  
1.1. Описание класса микропрограмм. Ограничим класс параллельных микропрограмм тем, который описывает преобразования информации в двумерных вычислительных структурах, так как именно эти структуры являются исходными при применении метода расслоения. Обозначим этот класс  $S_2$ .

Введем на плоскости левую систему координат  $(x, y)$ , разобьем плоскость на клетки с целочисленными координатами (такую плоскость будем обозначать  $D$ ).

Для  $S_2$  характерно использование в качестве имен клеток множества целочисленных координат клеток на плоскости:

$$M = \{(x, y) \mid x, y = 0, \pm 1, \pm 2, \dots\}.$$

Левые и правые части микрокоманд строятся с использованием конфигураций вида

$$S = \{(a_0, \varphi_0(x, y))(a_1, \varphi_1(x, y)) \dots (a_r, \varphi_r(x, y))\},$$

где  $a_i \in A$ ,  $x, y$  — целочисленные переменные;  $\varphi_i(x, y)$  — именуемые функции.

Пара  $(a_i, \varphi_i(x, y))$  называется  $i$ -й компонентой конфигурации.

В  $S$  используются именуемые функции сдвига вида

$$\varphi_i : \langle x, y \rangle \rightarrow \langle x + \alpha_i, y + \beta_i \rangle,$$

где  $\alpha_i, \beta_i$  — целые числа. Функции будем записывать просто как  $\langle x + \alpha_i, y + \beta_i \rangle$ . Отметим, что в записи базовой части каждой микрокоманды содержится тождественная именуемая функция  $\langle x, y \rangle$ .

В качестве основного алфавита в  $S_2$  используется алфавит  $A = \{0, 1\}$ . Если  $A$  содержит другие символы, то они кодируются векторами из нулей и единиц. Наряду с  $A$ , введем два дополнительных символа:  $\#$  — пустой символ,  $*$  — общий (символ, представляющий любой символ алфавита  $A$ ).

В качестве клеточных массивов, перерабатываемых микропрограммами из  $C_2$ , выступают конечные подмножества клеток на плоскости  $D$ , в которые вписаны символы из  $A$ .

**Пример 1.** К классу  $C_2$  принадлежит микропрограмма  $\Phi_2$  параллельного суммирования многих чисел [7, 10];  $A = \{0, 1\}$ ,  $W_2$  — двумерная прямоугольная таблица размером  $\mu \times \nu$ , чьи клетки перенумерованы в соответствии с левой системой координат ( $x$  — абсцисса,  $y$  — ордината). Двоичные  $\mu$ -разрядные числа записаны в строки таблицы, младшие биты — в самый правый столбец таблицы:

$$\begin{aligned} \Theta_1 : & \{(1, \langle x, y \rangle)(1, \langle x, y + 1 \rangle)(0, \langle x + 1, y \rangle)\} * \\ & * \{(0, \langle x + 1, y - 1 \rangle)(0, \langle x, y - 1 \rangle)\} \rightarrow \\ & \rightarrow \{(0, \langle x, y \rangle)(0, \langle x, y + 1 \rangle)(1, \langle x + 1, y \rangle)\}; \\ \Theta_2 : & \{(1, \langle x, y \rangle)(0, \langle x, y + 1 \rangle)\} * \{(0, \langle x, y - 1 \rangle)\} \rightarrow \\ & \rightarrow \{(0, \langle x, y \rangle)(1, \langle x, y + 1 \rangle)\}. \end{aligned}$$

Возьмем некоторую  $S$  и подставим в нее вместо  $x, y$  координаты конкретной клетки  $\langle x^0, y^0 \rangle$ , получим элемент конфигурации. Выбор множества имен  $M$  и функций  $\varphi_i$  таков, что этот элемент может быть изображен графически.

**Пример 2.** При  $x = 2, y = 1$  конфигурация

$$S = \{(1, \langle x, y \rangle)(1, \langle x, y + 1 \rangle)(0, \langle x + 1, y \rangle)\}$$

порождает элемент

$$\{(1, \langle 2, 1 \rangle)(1, \langle 2, 2 \rangle)(0, \langle 3, 1 \rangle)\}.$$

Его графическое изображение показано на рис. 1.

Заменим в  $S$  символы алфавита  $A$  пустым символом. Выражение

$$S^* = \{(\#, \varphi_0(x, y))(\#, \varphi_1(x, y)) \dots (\#, \varphi_r(x, y))\}$$

называется шаблоном конфигурации.

Подставим в  $S^*$  конкретное имя клетки  $\langle x^0, y^0 \rangle$ , получим шаблон, который называется шаблоном элемента конфигурации. При его графическом изображении в клетки не вписываются никакие символы.

Аналогично определяются шаблоны левой и правой частей любой микрооперации, так как они являются элементами некоторых конфигураций.

Объединение (в теоретико-множественном смысле) шаблонов левой и правой частей микрооперации называется шаблоном микрооперации. При графическом изображении клетки, попавшие в шаблоны из правой части микрооперации, выделяются штриховкой.

Объединение (в теоретико-множественном смысле) шаблонов микроопераций  $\Theta_{i_1}, \dots, \Theta_{i_k}$  называется шаблоном набора  $\Theta_{i_1}, \dots, \Theta_{i_k}$ . При графическом изображении все клетки, попавшие в шаблон из правых частей микроопераций набора, выделяются штриховкой.

Центром шаблона любого вида, кроме  $S^*$ , называется клетка, установленная в соответствие компоненте с тождественной именуемой функцией одной из конфигураций, использованных при его построении.

Клеточный массив, в котором все символы алфавита заменены пустым символом, называется шаблоном клеточного массива  $W$  (обозначается  $W^p$ ).

1.2. Распределение множества микроопераций. Как отмечалось, параллельная микропрограмма интерпретируется сетью автоматов. При этом выполняется условие, что множество имен клеток, используемых в записи клеточных массивов, конечно.

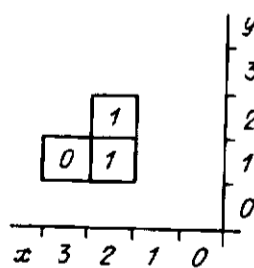


Рис. 1

Обозначим его  $M'$ . В нашем случае  $M' = \{(x, y) / u \leq x \leq u', v \leq y < v'\}$ , где  $u, u', v, v'$  — целые числа.

Другими словами, при построении сети исходным является некоторый максимальный по мощности шаблон  $W^p$  с именами клеток из  $M'$ . Множеству имен клеток шаблона  $W^p$  ставится в соответствие множество имен автоматов. Примем, что клетке с именем  $\langle x^0, y^0 \rangle$  ставится в соответствие автомат с именем  $\langle x^0, y^0 \rangle$ .

Обозначим  $\Phi(x^0, y^0)$  — множество микроопераций, полученных из  $\Phi$ , при подстановке в ее микрокоманды координат  $x^0, y^0$ .

Объединение

$$F = \bigcup_{\langle x^0, y^0 \rangle \in M'} \Phi(x^0, y^0)$$

называется полным списком микроопераций.

Каждому автомату с именем  $\langle x^0, y^0 \rangle$  приписывается некоторый набор (возможно, пустой) микроопераций, обозначим его  $\Phi^*(x^0, y^0)$ . Микрооперации могут быть распределены среди автоматов любым образом при соблюдении единственного условия: каждая микрооперация должна быть приписана, по крайней мере, одному автомату, т. е.

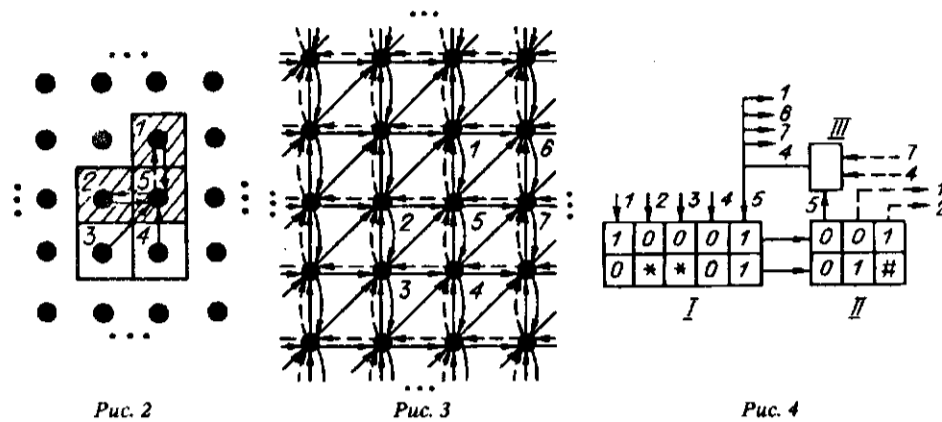
$$F = \bigcup_{\langle x^0, y^0 \rangle \in M'} \Phi^*(x^0, y^0).$$

Таблица переходов автомата и его входовыходные связи с соседними автоматами определяются набором микроопераций, которые к нему приписаны. Как это происходит, показано в примере 3.

Те автоматы, к которым не приписана ни одна микрооперация, — это ячейки памяти, хранящие информацию; автоматы, к которым приписаны непустые наборы микроопераций, — это функциональные ячейки, осуществляющие не только хранение, но и переработку информации. Ячейку будем изображать простым кружком, функциональную ячейку — зачерненным.

Отметим, что все последующие примеры будут базироваться на  $\Phi_2$  из примера 1.

**Пример 3.** Пусть каждая ячейка с именем из  $M'$  является функциональной. Каждой функциональной ячейке с именем  $\langle x^0, y^0 \rangle$  припишем набор микроопераций:  $\Theta_1(x^0, y^0), \Theta_2(x^0, y^0)$ . На рис. 2 показан шаблон набора микроопераций. Для удобства последующего изложения ячейки, установленные в соответствие клеткам шаблона, перенумерованы. Функциональная ячейка с именем  $\langle x^0, y^0 \rangle$  (будем называть такую ячейку центральной) имеет номер 5. Чтобы выполнить микрооперации  $\Theta_1(x^0, y^0), \Theta_2(x^0, y^0)$ , она должна получать информацию от всех ячеек, попавших в шаблон (связи показаны сплошными линиями), и записывать информацию в ячейки, попавшие в заштрихованную часть шаблона (связи показаны штриховыми линиями). Выбор класса  $S_2$  таков, что любой элемент конфигурации может быть получен из другого ее элемента сдвигом. Поэтому, как следствие этого факта, вся сеть (рис. 3) может быть получена повторением фрагмента сети (см. рис. 2) при совмещении его центральной ячейки со всеми ячейками с именами из  $M'$ . Структурная схема ячейки 5 показана на рис. 4, где  $I$  и  $II$  — блоки микропрограммной памяти,  $III$  — элемент внутренней памяти, хранящий перерабатываемую информацию, номера входов и выходов соответствуют номерам ячеек, с которыми связана ячейка 5. Работа ячейки 5 состоит в том, что при совпадении вектора данных, поступающего по входам  $I-5$ , с одной из строк блока  $I$  компоненты вектора из соответствующей строки блока  $II$  записываются в элементы памяти соседних ячеек согласно номерам выходов.



В [11] продемонстрировано, что, выбирая распределение функциональных ячеек в сети и затем распределение микроопераций среди них, можно найти вариант сети, в которой получается меньшее среднее число входовых связей у каждой ячейки, чем у ячеек в других вариантах сети. Но в целом вариации этого числа невелики, при этом сложность функциональной ячейки (число строк и столбцов в блоках I, II) возрастает. Более радикальным является структурно-временное преобразование сети, которое позволяет ограничить сложность функциональной ячейки и существенно изменить топологию сети, правда, ценой растяжки вычислительного процесса во времени.

1.3. Структурно-временное преобразование сети. Выберем из  $\Phi^*(x^0, y^0)$  по представителю таких микроопераций, шаблоны которых не получаются один из другого при сдвиге до совпадения их центров. Сдвинем шаблоны представителей так, чтобы, во-первых, совпали их центры, а во-вторых, их общий центр совпал с клеткой с именем  $\langle x^0, y^0 \rangle$ . Получим шаблон, который будем называть опорным шаблоном набора  $\Phi^*(x^0, y^0)$ .

Пример 4. Зададим имена функциональных ячеек на плоскости двумя совокупностями координат:

$$\{\langle x^0, y^0 \rangle \mid x^0 = 2j, y^0 = 4l\}, \quad \{\langle x^0, y^0 \rangle \mid x^0 = 2j + 1, y^0 = 4l + 2\},$$

где  $j, l = 0, \pm 1, \pm 2, \dots$

Каждой функциональной ячейке с именем  $\langle x^0, y^0 \rangle$  припишем набор микроопераций  $\Phi_{\Sigma}^*(x^0, y^0)$ :

$$\Theta_1(x^0, y^0), \Theta_1(x^0, y^0 + 1), \Theta_1(x^0, y^0 + 2), \Theta_1(x^0, y^0 + 3), \\ \Theta_2(x^0, y^0), \Theta_2(x^0, y^0 + 1), \Theta_2(x^0, y^0 + 2), \Theta_2(x^0, y^0 + 3).$$

В качестве представителей можно взять, например,  $\Theta_1(x^0, y^0 + 1)$  и  $\Theta_2(x^0, y^0 + 3)$ . Опорный шаблон набора  $\Phi_{\Sigma}^*(x^0, y^0)$  совпадает с шаблоном, изображенным на рис. 2.

Разделим набор  $\Phi^*(x^0, y^0)$  на поднаборы  $\Phi_1^*(x_1^0, y_1^0), \Phi_2^*(x_2^0, y_2^0), \dots, \Phi_k^*(x_k^0, y_k^0)$ . В один поднабор  $\Phi_i^*(x_i^0, y_i^0)$  включаются микрооперации, шаблоны которых имеют один и тот же центр  $\langle x_i^0, y_i^0 \rangle$ .

Пример 5.  $\Phi_{\Sigma}^*(x^0, y^0)$  делится на поднаборы:

$$\Phi_{1\Sigma}^*(x^0, y^0) = \{\Theta_1(x^0, y^0), \Theta_2(x^0, y^0)\}, \\ \Phi_{2\Sigma}^*(x^0, y^0 + 1) = \{\Theta_1(x^0, y^0 + 1), \Theta_2(x^0, y^0 + 1)\}, \\ \Phi_{3\Sigma}^*(x^0, y^0 + 2) = \{\Theta_1(x^0, y^0 + 2), \Theta_2(x^0, y^0 + 2)\}, \\ \Phi_{4\Sigma}^*(x^0, y^0 + 3) = \{\Theta_1(x^0, y^0 + 3), \Theta_2(x^0, y^0 + 3)\}.$$

Структурно-временное преобразование сети, предлагаемое здесь, является некоторой модификацией  $ST$ -преобразования из [12]. Основу структурно-временного преобразования составляет переход от параллельного выполнения микроопераций набора  $\Phi^*(x^0, y^0)$  функциональной ячейкой с именем  $\langle x^0, y^0 \rangle$  к последовательному выполнению поднаборов набора  $\Phi^*(x^0, y^0)$  той же функциональной ячейкой. Внутри каждого поднабора микрооперации выполняются параллельно. В сети опорный шаблон определяет связи ячейки с именем  $\langle x^0, y^0 \rangle$  с соседями. Чтобы создать условия для выполнения функциональной ячейкой микроопераций очередного поднабора, необходимо выполнить сдвиг информации между ячейками сети, обеспечивающий совмещение центра поднабора с позицией этой функциональной ячейки. Для этого проводятся в сети такие связи между ячейками, которые создают в ней структуру двумерного регистра сдвига.

Предложенное преобразование так же, как и  $ST$ -преобразование, корректно тогда и только тогда, когда  $\Phi$  допускает ординарную асинхронную интерпретацию [12]. Корректность преобразования означает, что, как бы не выбиралась последовательность сдвигов информации в сети, результат выполнения  $\Phi$  будет таким же, как и в случае выполнения  $\Phi$  по правилам, описанным во введении. Отметим, что  $\Phi_{\Sigma}$  допускает ординарную асинхронную интерпретацию.

**Пример 6.** На рис. 5 изображена сеть, выполняющая микропрограмму  $\Phi_{\Sigma}$ , и такая, в которой реализовано последовательное выполнение поднаборов  $\Phi_{1\Sigma}^*$ ,  $\Phi_{2\Sigma}^*$ ,  $\Phi_{3\Sigma}^*$ ,  $\Phi_{4\Sigma}^*$  каждой функциональной ячейкой  $\langle x^0, y^0 \rangle$ . Связи функциональной ячейки с соседями задаются опорным шаблоном (см. рис. 2). Эти связи изображены прямыми линиями. Дугами показаны связи, обеспечивающие сдвиг информации в положительном и отрицательном направлениях вдоль оси  $y$ . При выполнении микропрограммы  $\Phi_{\Sigma}$  чередуются такты выполнения микроопераций поднаборов и такты сдвига информации:

$$\Phi_{1\Sigma}^* \downarrow \Phi_{2\Sigma}^* \downarrow \Phi_{3\Sigma}^* \downarrow \Phi_{4\Sigma}^* \uparrow \Phi_{3\Sigma}^* \uparrow \Phi_{2\Sigma}^* \uparrow \Phi_{1\Sigma}^* \downarrow \Phi_{2\Sigma}^* \downarrow \dots,$$

здесь  $\downarrow(\uparrow)$  — сдвиг информации на одну позицию в отрицательном (положительном) направлении вдоль оси  $y$ .

Чтобы получить законченное представление о сети, интерпретирующей  $\Phi_{\Sigma}$ , нужно рассмотреть устройство ее граничных частей: верхней и нижней. В этой сети по сравнению с сетью на рис. 3 добавлен один ряд ячеек сверху и три ряда ячеек снизу, исходное состояние элементов памяти которых нулевое. Их добавка обеспечивает выполнение сдвигов информации вдоль оси  $y$ . Ряды ячеек перенумеруем сверху вниз числами 1, 2, ...,  $v+4$  и расположим на плоскости так, чтобы 3-й ряд обязательно содержал функциональные ячейки.

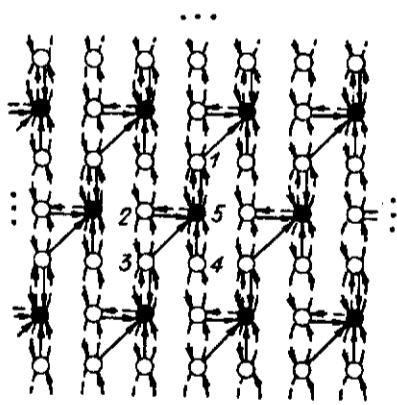


Рис. 5

Принудительно сделаем ячейки рядов 1, 2,  $v+4$  ячейками памяти. Отметим также, что в соответствии с правилами распределения микроопераций функциональные ячейки 3-го ряда выполняют только микрооперации поднабора  $\Phi_{1\Sigma}^*$ , а функциональные ячейки 5-го ряда — микрооперации поднаборов  $\Phi_{1\Sigma}^*$ ,  $\Phi_{2\Sigma}^*$ ,  $\Phi_{3\Sigma}^*$ .

Сделаем одно замечание. В данном конкретном примере распределение функциональных ячеек таково, что приписанные им микрооперации могут выполняться в произвольном порядке даже в том случае, когда у них отсутствуют контекстные части. Это наблюдение позволяет упростить сеть, убрав в ней все связи, такие, как связи ячеек с номерами 3, 5 и 4, 5 (см.

рис. 5), и исключив из блока  $I$  (см. рис. 4) третий и четвертый столбцы.

2. Расслоение двумерных вычислительных структур. 2.1. Разбиение множества клеток. Возьмем множество клеток плоскости  $D$ .

Разбиением [13]  $R$  плоскости  $D$  называется множество подмножеств  $D_k$  в  $D$  таких, что каждая клетка с именем  $\langle x^0, y^0 \rangle \in D$  принадлежит только одному из подмножеств  $D_k$ .

Подмножества  $D_k$  будем называть элементами разбиения.

Среди всех возможных вариантов разбиений выделим такие, элементы которых задаются шаблоном конфигурации:

$$\{(\#, \langle x, y \rangle)(\#, \langle x + \alpha_1, y + \beta_1 \rangle) \dots (\#, \langle x + \alpha_n, y + \beta_n \rangle)\}.$$

Этот шаблон назовем шаблоном разбиения и обозначим  $\bar{S}$ .

Чтобы получить элемент разбиения, нужно задать в  $\bar{S}$  конкретные значения параметров  $\alpha_i, \beta_i, h$ , а вместо  $x, y$  подставить координаты  $x^0, y^0$  некоторой клетки.

**Пример 7.** Выберем на плоскости  $D$  левую систему координат. Зададим шаблон

$$\bar{S} = \{(\#, \langle x, y \rangle)(\#, \langle x, y - 1 \rangle)(\#, \langle x + 1, y \rangle)(\#, \langle x + 1, y + 1 \rangle)\}$$

и две совокупности координат клеток:

$$\{\langle x^0, y^0 \rangle \mid x^0 = 2j, y^0 = 4l\}, \{\langle x^0, y^0 \rangle \mid x^0 = 2j + 1, y^0 = 4l + 2\},$$

где  $j, l = 0, \pm 1, \pm 2, \dots$ .

Разбиение  $R$  плоскости  $D$  показано на рис. 6.

2.2. Расслоение множества клеток. По разбиению  $R$  плоскости  $D$  можно построить отношение эквивалентности  $E(R)$  [13]:  $\langle x^0, y^0 \rangle E(R) \langle x^{0'}, y^{0'} \rangle$  означает, что клетки с именами  $\langle x^0, y^0 \rangle$  и  $\langle x^{0'}, y^{0'} \rangle$  лежат в одном и том же элементе  $D_k$ .

Обозначим  $D/E(R)$  — фактор-множество, состоящее из классов эквивалентности  $D_k$  отношения  $E(R)$ . Введем отображение  $\delta_1$ , которое каждому элементу  $D_k$  из  $D/E(R)$  взаимнооднозначно ставит в соответствие клетку с целочисленными координатами  $\langle X_k, Y_k \rangle$  в некоторой другой плоскости  $\bar{D}$ , разбитой на клетки.

В каждом  $D_k$  перенумеруем в каком-то порядке составляющие его клетки числами  $1, 2, \dots, n_k$ . Максимальное  $n_k$  обозначим  $n_{\max}$ . Введем координату  $z$ , ортогональную плоскости  $\bar{D}$ . Заменим  $\bar{D}$  ее дубликатами, расположенными вдоль  $z$ . Общее количество дубликатов равно  $n_{\max}$ . Координаты клеток  $l$ -й плоскости-дубликата задаются тройкой  $\langle X, Y, l \rangle$ .

Введем отображение  $\delta_2$ : для каждого  $D_k$  входящую в него клетку с номером  $l$  совместим с клеткой  $\langle X_k, Y_k, l \rangle$ , клетку с номером  $2$  — с клеткой  $\langle X_k, Y_k, 2 \rangle$  и т. д., клетку с номером  $n_k$  — с клеткой  $\langle X_k, Y_k, n_k \rangle$ .

Результат выполнения отображения  $\delta_2$  называется расслоением множества клеток плоскости  $D$ .

Пусть задан шаблон  $W^p$  некоторого клеточного массива  $W$ .

Множество  $\pi = \{D_{k_1}, \dots, D_{k_r}\}$  называется мощением шаблона  $W^p$ , если:

- 1)  $D_{k_j} \in R$ ;
- 2)  $\bigcup_{j=1}^r D_{k_j} \supseteq W^p$ ;

3)  $\pi$  минимально (из  $\pi$  нельзя выбросить ни один элемент, не нарушив условия 2).

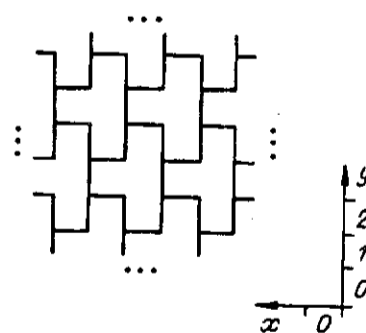


Рис. 6

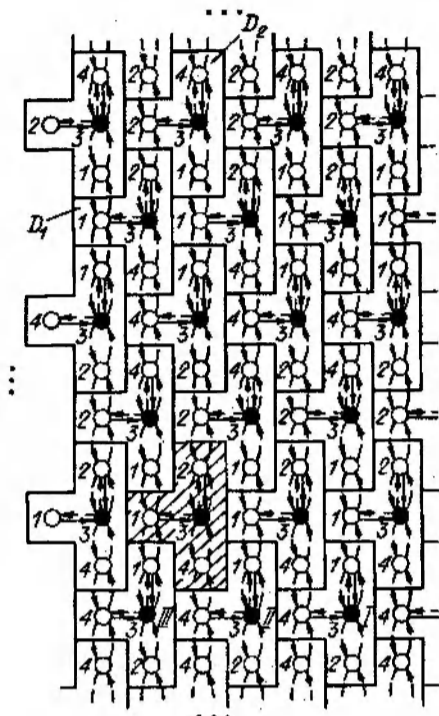


Рис. 7

ячейками, попавшими в один элемент мощения, назовем их вертикальными межслойными. Второй класс образуют связи между одноименными ячейками в разных элементах мощения, их назовем внутрислойными. Третий класс образуют связи между ячейками с разными именами, расположенными в разных элементах мощения, их назовем косыми межслойными.

**Пример 8.** На рис. 7 показаны мощение и сеть, интерпретирующая  $\Phi_2$  и изображенная с учетом замечания, сделанного в примере 6. Цифрами указаны номера ячеек и клеток в каждом элементе мощения. Связи между ячейками 1—4, попавшими в элемент  $D_1$ , являются вертикальными межслойными. Связи между ячейками с номерами 2 элементов  $D_1$  и  $D_2$  внутрислойные, косые межслойные связи в сети отсутствуют.

**2.4. Электрооптическая реализация расслоенной сети автоматов.** Все ячейки некоторого элемента мощения  $D_k$  расположены в разных слоях одна под другой, т. е. имеют одни и те же горизонтальные координаты  $(X_k, Y_k)$ . Чтобы сделать последующее изложение достаточно конкретным, будем ориентироваться на реализацию межслойных связей с использованием потенциальных электрооптических ячеек [5]. Для выполнения вертикальной межслойной связи, идущей из ячейки с номером  $i$  в ячейку с номером  $j$  ( $i < j$ ), на выходе ячейки с координатами  $(X_i, Y_k)$ , расположенной в  $i$ -м слое, устанавливается модулятор, а на входе ячейки с координатами  $(X_k, Y_k)$ , расположенной в  $j$ -м слое, — фотоприемник. Во всех слоях, расположенных между  $i$ -м и  $j$ -м, если они есть, делаются прозрачные окна так, чтобы свет, падающий сверху на модулятор, мог пройти на фотоприемник. Если  $i > j$ , то модулятор и фотоприемник меняются местами, а также меняется на противоположное направление светового потока.

Внутрислойная связь двух ячеек с одинаковым номером  $i$ , входящих в разные элементы мощения  $D_k, D_q$ , — обычная электрическая связь между ячейками с координатами  $(X_k, Y_k), (X_q, Y_q)$ , расположенными в  $i$ -м слое.

Косая межслойная связь, идущая из ячейки с номером  $i$  элемента  $D_k$  в ячейку с номером  $j$  ( $i < j$ ) элемента  $D_q$ , частично выполняется как оптическая,

Применение  $\delta_2$  к  $\pi$  приводит к расслоению множества клеток, составляющих  $\pi$ , а значит, к расслоению множества клеток шаблона  $W^p$ .

**2.3. Отображение двумерной сети в трехмерную.** При построении отображения исходным объектом является совместное изображение шаблона  $W^p$  и двумерной сети, интерпретирующей некоторую параллельную микропрограмму  $\Phi$ . Отображение выполняется в два шага.

На первом шаге выбирается один или несколько шаблонов разбиения и выполняется мощение шаблона  $W^p$ . Клетки каждого элемента мощения нумеруются. Номер клетки в элементе мощения присваивается установленной в соответствие этой клетке ячейке сети.

На втором шаге производится расслоение мощения шаблона  $W^p$ . При расслоении мощения ячейки сети переходят в те слои, в которые переходят установленные в соответствие им клетки. Связи между ячейками в расслоенной сети можно разделить на три класса. Первый класс образуют связи между



а частично — как электрическая. Ее оптическая часть строится так, как если бы ячейки с номерами  $i$  и  $j$  были расположены в  $D_k$ , т. е. фотоприемник оказывается расположенным в ячейке с координатами  $\langle X_k, Y_k \rangle$   $j$ -го слоя, поэтому затем он соединяется электрической связью со входом ячейки с координатами  $\langle X_q, Y_q \rangle$  этого же слоя.

**Пример 9.** Модулятор будем изображать заштрихованным кружком, фотоприемник — зачерненным, окно в слое — обычным, световой поток — двойной линией, электрические связи — штриховыми линиями. Ячейки структуры будем изображать квадратиками. Электрооптическая реализация сети (см. рис. 7) представлена на рис. 8, 9.

На рис. 8 показаны электрические связи ячеек памяти первого слоя. Ячейки первого слоя — это те ячейки, которые имеют номер 1 на рис. 7. Их координаты в слое совпадают с координатами элементов мощения при перехо-

на рис. 9 показаны оптические каналы, реализующие межслойные вертикальные связи ячеек из заштрихованного элемента мощения (см. рис. 7). Конечно, выходы электронных схем (изображены штриховыми прямоугольниками) должны быть соединены электрическими связями с модуляторами, а входы схем — с фотоприемниками. Мы не показали эти связи по двум причинам: первая — это внутренние связи ячеек, а не межъячеечные; вторая — они существенно зависят от технических решений, использованных при построении электронных схем. Однако независимо от технической реализации можно видеть, что 60 % связей исходной сети в расслоенной структуре превращаются в оптические связи, топология межъячеечных связей в слоях очень проста и, более того, у первого, второго и четвертого слоев одна и та же.

**2.5. Некоторые рекомендации по построению расслоенной структуры.** Отметим те факты, которые влияют на параметры расслоенной структуры, и дадим некоторые рекомендации по ее построению.

Максимальное число клеток в элементе разбиения определяет число слоев в структуре. Поэтому очевидно, что это число не должно быть слишком большим.

Чтобы расслоенная структура была квазиоднородной (состоящей из небольшого числа повторяющихся фрагментов), целесообразно получать разбиение с использованием единственного варианта шаблона (как в примере 7) или некоторого фиксированного набора

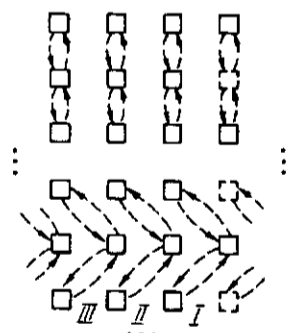


Рис. 8

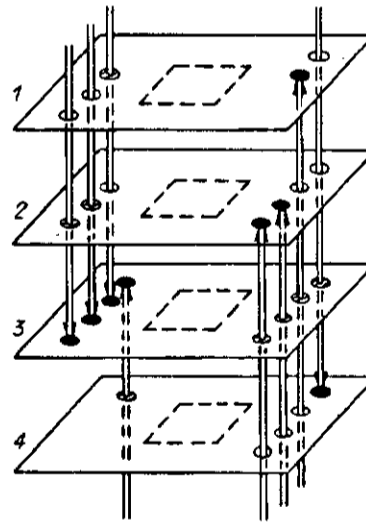


Рис. 9

вариантов шаблона, а функциональные ячейки выбирать так, чтобы была итеративная повторяемость в их расположении.

Последующие рекомендации относятся к рациональному распределению микроопераций среди функциональных ячеек. Для того чтобы их сделать, введем некоторые понятия.

Две клетки с именами  $\langle x_1, y_1 \rangle$ ,  $\langle x_2, y_2 \rangle$  находятся в отношении " $<$ ":  $\langle x_1, y_1 \rangle < \langle x_2, y_2 \rangle$ , если для их координат выполняется условие: либо  $(x_1 < x_2 \ \& \ y_1 \leq y_2)$ , либо  $(x_1 \leq x_2 \ \& \ y_1 < y_2)$ .

Совокупность центров  $\{\langle x_1, y_1 \rangle, \langle x_2, y_2 \rangle, \dots, \langle x_k, y_k \rangle\}$  поднаборов некоторого набора микроопераций образует монотонную последовательность, если любая пара имен клеток совокупности находится в отношении " $<$ ". Члены монотонной последовательности, упорядоченные в соответствии с отношением " $<$ ", будем нумеровать от 1 до  $k$ .

Два набора  $\Phi^*(x^0, y^0)$  и  $\Phi^*(x^{0'}, y^{0'})$  называются однотипными, если шаблон любой микрооперации одного набора может быть получен сдвигом из шаблона некоторой микрооперации другого набора.

Расстояние от центра поднабора с координатами  $\langle x_i, y_i \rangle$  некоторого набора до функциональной ячейки, которой набор приписан, определяется как  $(x^0 - x_i, y^0 - y_i)$ .

При распределении микроопераций нужно стремиться к тому, чтобы: 1) все наборы микроопераций были однотипными и равномоощными; 2) каждый набор был разделен на одно и то же число поднаборов; 3) внутри каждого набора центры поднаборов образовывали монотонную последовательность; 4) все поднаборы с одинаковыми номерами из разных наборов находились на одном и том же расстоянии от соответствующих функциональных ячеек.

Выполнение рекомендаций приводит к тому, что в структуре все функциональные ячейки имеют одинаковую сложность и легко организуется последовательность сдвигов информации, исключая пропуски микроопераций (см. пример 7).

Далее, при выполнении мощения сети нужно стремиться к тому, чтобы было максимальное совпадение опорного шаблона каждой функциональной ячейки с некоторым элементом мощения. Выполнение этой рекомендации обеспечивает перевод максимального процента связей двумерной сети в оптические связи трехмерной структуры.

И наконец, отметим последний фактор, влияющий на параметры расщепленной структуры: выбирая нумерацию ячеек, попадающих в один элемент разбиения, можно управлять распределением функциональных ячеек по слоям и типом связей между ячейками из разных элементов разбиения. В примере 8 нумерация выбрана так, что все функциональные ячейки попадают в один слой, а все связи между ячейками, принадлежащими разным элементам разбиения, электрические внутрислойные.

**Заключение.** За счет эксплуатации третьего измерения может быть построено вычислительное электрооптическое устройство, состоящее из нескольких слоев. Каждый слой — это микроэлектронная СБИС, которая может передавать оптические сигналы на другие слои. Возможность задавать число слоев в устройстве и выбирать в достаточно широких пределах вариант реализации связей (оптический или электрический) между ячейками устройства позволяет сделать топологию каждого слоя очень простой.

У такого устройства по сравнению с двумерным устройством, выполняющим такой же алгоритм, появляются следующие преимущества. В нем существенно уменьшен объем электрических межсоединений. Процесс создания таких устройств может оказаться более простым и надежным, так как слои могут быть спроектированы и изготовлены независимо один от другого и объединены в единое устройство только при сборке.

#### СПИСОК ЛИТЕРАТУРЫ

1. Tanida J., Inchioka Y. Optical-logic-array processor using shadowgrams // JOSA.—1985.—2, N 8.—P. 1245.

2. Косцов Э. Г., Мишин А. И. Особенности построения оптических ЦВМ // Микроэлектроника.—1977.—6, вып. 2.
3. Egorov V. M., Koszov E. G. Integral optical digital computers // Appl. Opt.—1990.—29, N 8.—P. 1178.
4. Brenner K.-H., Huang A., Streibl N. Digital optical computing with symbolic substitution // Appl. Opt.—1986.—25, N 19.—P. 3054.
5. Егоров В. М., Косцов Э. Г. Перспективы создания цифровых высокопроизводительных вычислительных устройств // Автометрия.—1985.—№ 1.
6. Егоров В. М., Косцов Э. Г. Микроэлектронные оптические цифровые вычислительные устройства // Автометрия.—1989.—№ 3.
7. Методы параллельного микропрограммирования /Под ред. О. Л. Бандман.—Новосибирск: Наука, 1981.
8. Bandman O. L., Piskunov S. V. Parallel substitution algorithm as a model for distributed computations // J. New Gener. Comput. Syst.—1991.—4, N 1.—P. 3.
9. Bandman O. L., Piskunov S. V. Parallel microprogramming as a tool for multi-microprocessor system // Lect. Notes in Comp. Sci.—Berlin a. o., 1989.—Vol. 342.—P. 57.
10. А. с. 436350 СССР. Двоичный сумматор /Ю. Н. Корнев, С. В. Пискунов, С. Н. Сергеев.—Заявл. 14.06.71; Опубл. 15.07.74; Бюл. № 26.
11. Маркова В. П., Пискунов С. В. Универсальная процедура построения структурных схем специализированных параллельных вычислителей // Математическое и архитектурное обеспечение параллельных вычислений.—Новосибирск: ВЦ СО АН СССР, 1989.
12. Bandman O. L. Space-time transformation of cellular computations // Parallel Computing Technologies: Proc. of the Internat. Conf. /Ed. N. Mirenkov.—Novosibirsk, USSR.—1991.—P. 235.
13. Биркгоф Г., Барти Т. Современная прикладная алгебра.—М.: Мир, 1976.

Поступила в редакцию 28 февраля 1992 г.

УДК 535.854

Х. Брандт, Ю. Н. Солодкин

(Новосибирск)

#### О ВОЗМОЖНОСТИ КОРРЕКЦИИ РЕЗУЛЬТАТА В ЦЕЛОЧИСЛЕННЫХ ИНТЕРФЕРОМЕТРАХ

Проведен анализ метода определения полной разности фаз, основанного на решении систем сравнений целых чисел для случая, когда численные значения интерференционных полос имеют общий делитель. Показано, что возможны не только получение достоверного значения полной разности фаз, но и коррекция результата измерения. Приведен критерий необходимости коррекции. Рассмотрены условия, при которых коррекция возможна.

В [1] показано, что математический аппарат арифметики вычетов (арифметики в остаточных классах) адекватен задаче, решаемой в интерферометрии. Эта задача сводится к определению полной разности фаз световых волн, выраженной в числе интерференционных полос. Подсчет числа полос во времени или в пространстве требует знания точки отсчета или «нулевой» полосы, определения знака при суммировании полос, фиксации экстремальных или нулевых значений яркости для выделения периодов. В итоге для получения разности фаз оказываются необходимыми существенная априорная информация и выполнение трудоемких, практически не поддающихся автоматизации операций. Особенно эти проблемы характерны для голографической интерферометрии, в которой расшифровка интерферограмм невозможна иначе, чем в интерактивном режиме [2].

Арифметика вычетов оперирует сравнениями вида

$$X \equiv b(\text{mod } m), \quad (1)$$

где  $X$  — все целые числа, которые при делении на  $m$  дают тот же остаток, что и  $b$ . Они образуют класс чисел, сравнимых по модулю  $m$ , а каждое число класса