

РОССИЙСКАЯ АКАДЕМИЯ НАУК
МЕТОДЫ ПОСТРОЕНИЯ 3-D ОПТИЧЕСКИХ
ИНТЕГРАЛЬНЫХ СХЕМ

УДК 621.382

С. В. Пискунов
(Новосибирск)

ПОСТРОЕНИЕ ПРЕДЕЛЬНО РАССЛОЕННЫХ
ЭЛЕКТРООПТИЧЕСКИХ КЛЕТОЧНЫХ УСТРОЙСТВ*

Предлагается преобразование двумерной клеточной структуры в трехмерную (многослойную) клеточную структуру. Преобразование основано на модели распределенных вычислений, называемой алгоритмом параллельных подстановок. Характерной чертой преобразования является то, что при схемотехнической реализации трехмерной структуры оно обеспечивает возможность построения предельно расслоенного электрооптического устройства. У этого устройства есть слой, состоящий из ячеек памяти с оптическими входами и выходами, другие слои состоят из повторяющихся простых электрооптических ячеек, все межслойные связи оптические.

Введение. Предлагается преобразование двумерной клеточной структуры в трехмерную. Исходная структура представляет собой совокупность однотипных клеток (конечных автоматов), объединенных в плоскую однородную сеть. Результирующая структура — это трехмерная сеть, состоящая из нескольких плоских однородных сетей (слоев). Обе структуры выполняют один и тот же параллельный алгоритм массовой обработки информации. Рассматривается случай, когда число слоев больше двух. Двухслойные структуры исследованы в [1].

Цель преобразования — обеспечить возможность построения электрооптического устройства с простой электрической топологией отдельного слоя и межслойными оптическими связями при схемотехнической реализации расслоенной сети.

Целесообразность преобразования обусловлена тем, что накоплен большой опыт создания двумерных структур, и оно открывает простой путь эффективно использовать этот опыт при построении трехмерных устройств.

Предлагаемое преобразование от преобразования в [2] отличается следующим. В [2] при расслоении выполнялся перенос части клеток плоской сети и их связей в третье измерение. В данном преобразовании сама клетка делится на части и выполняется перенос в третье измерение ее частей и их связей (внутриклеточных связей), а клетки сохраняют в расслоенной сети те же позиции, что и в исходной.

Электрооптическая реализация расслоенной сети опирается на схемотехнику, разработанную в [3—5]. Характерной чертой преобразования является то, что оно может приводить к построению предельно расслоенного электрооптического устройства, у которого есть слой, состоящий только из ячеек памяти с оптическими входами и выходами, а все другие слои состоят из повторяющихся электрооптических ячеек, предложенных в [3]. Следует

* Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (93-012-1000).

ожидать, что совместное применение этого преобразования и преобразования из [2] обеспечит глубокие трансформации исходной структуры для более полного использования возможностей, предоставляемых оптикой.

Преобразование основано на модели распределенных вычислений (алгоритм параллельных подстановок), представленной в [6—8].

В статье дается адаптированное к ее цели изложение основных понятий параллельного микропрограммирования (п. 1), причем основное внимание уделяется построению функционального преобразователя клетки (автомата), из экземпляров которой строится плоская сеть (п. 2). Это построение служит фундаментом для последующих разделов, в которых показано, как на основе метода декомпозиции функционального преобразователя (п. 3) построить расслоение плоской сети (п. 4) и затем выполнить электрооптическую реализацию расслоенной сети (п. 5).

1. Параллельная микропрограмма. Пусть A — конечный алфавит, M — множество имен с мощностью не более чем счетной.

Пара $(a, m^0) \in A \times M$ называется *клеткой*, символ a — *состоянием* клетки, m^0 — *именем* клетки.

Конечная совокупность клеток $W \subset A \times M$, в которой нет двух клеток с одинаковыми именами, называется *клеточным массивом*.

Если в качестве множества имен использовать множество векторов, задающих координаты клеток в целочисленной прямоугольной решетке (клеточном пространстве), клеточный массив может быть изображен графически: в виде подмножества клеток пространства, в которые вписаны символы из алфавита A . В дальнейшем изложении будем ориентироваться именно на такое множество имен и обозначать его N^k , где $N = \{1, 2, 3, \dots\}$, а k — это мерность пространства. Переменную, определенную на множестве имен, обозначим m .

Пусть дана совокупность функций $\{\varphi_0(m), \varphi_1(m), \dots, \varphi_n(m)\}$, отображающих $N^k \rightarrow N^k$ и таких что:

- 1) $\varphi_0(m) = m$;
- 2) для $i > 0$ $\varphi_i(m) = m + \omega_i$, где ω_i — вектор длины k , составленный из целых чисел, хотя бы одно из которых отлично от нуля;
- 3) $\varphi_0(m) \neq \varphi_1(m) \neq \dots \neq \varphi_n(m)$.

Выражение

$$S(m) = \{(a_0, \varphi_0(m))(a_1, \varphi_1(m)) \dots (a_n, \varphi_n(m))\},$$

где $a_i \in A$, называется *конфигурацией*.

Функции $\varphi_0(m), \varphi_1(m), \dots, \varphi_n(m)$ называются *именующими*.

При подстановке в $S(m)$ вместо m конкретного $m^0 \in N^k$ получается клеточный массив. Так как в качестве именующих функций выбраны функции сдвига, все клеточные массивы, полученные из $S(m)$, имеют одно и то же геометрическое изображение и его можно считать образом самой конфигурации.

Преобразования исходного клеточного массива W^0 задаются *микрокомандами* вида

$$\Theta : S_1(m) * S_2(m) \rightarrow S_3(m),$$

где $S_1(m) = \{(a_i, \varphi_i(m))\}$, $S_2(m) = \{(b_j, \varphi_j(m))\}$, $S_3(m) = \{(c_i, \varphi_i(m))\}$ — конфигурации, объединение $S_1(m)$ и $S_2(m)$ — конфигурация, $i = 1, \dots, p$, $j = 1, \dots, q$, $a_i, b_j, c_i \in A$. Левая часть Θ состоит из базы $S_1(m)$ и контекста $S_2(m)$, правая часть Θ — это $S_3(m)$.

Так как левая и правая части микрокоманды Θ — это конфигурации, Θ может быть представлена графически в виде образов левой и правой частей, соединенных стрелкой. Применение Θ к массиву W^0 состоит в отыскании в

изображении W^0 всех вхождений образа левой части и замене в каждом из них вхождения образа базы образом правой части.

Пусть даны множество микрокоманд $\Phi = \{\Theta_1, \dots, \Theta_n\}$ и массив W^0 . Преобразование массива W^0 множеством Φ определяет синхронная итерационная процедура, в которой на каждом шаге применяются все микрокоманды. Если на каждом шаге в результате преобразования получается клеточный массив, Φ называется *параллельной микропрограммой*. Итерационные шаги продолжаются до тех пор, пока на очередном шаге в Φ может быть найдена хотя бы одна применимая микрокоманда, т. е. такая микрокоманда, образ левой части которой входит в изображение клеточного массива, полученного на предыдущем шаге.

Пример 1. Параллельная микропрограмма Φ , сложения многих целых положительных двоичных чисел.

$A = \{0, 1\}$, $M = N^2$, W^0 — это двумерная прямоугольная таблица, чьи клетки перенумерованы парами чисел в соответствии с левой системой координат (x — абсцисса, y — ордината), $m = \langle x, y \rangle$. Разряды двоичных слагаемых расположены в строках таблицы, нижняя строка нулевая, младшие разряды слагаемых занимают самый правый столбец таблицы. Микропрограмма состоит из двух микрокоманд:

$$\Phi = \left\{ \begin{array}{l} \Theta_1^a: \{(1, \langle x, y \rangle)(1, \langle x, y + 1 \rangle)(0, \langle x + 1, y \rangle)\} * \\ \quad \{(0, \langle x, y - 1 \rangle)(0, \langle x + 1, y - 1 \rangle)\} \rightarrow \\ \quad \{(0, \langle x, y \rangle)(0, \langle x, y + 1 \rangle)(1, \langle x + 1, y \rangle)\}; \\ \Theta_2^a: \{(1, \langle x, y \rangle)(0, \langle x, y + 1 \rangle)\} * \{(0, \langle x, y - 1 \rangle)\} \rightarrow \\ \quad \{(0, \langle x, y \rangle)(1, \langle x, y + 1 \rangle)\}. \end{array} \right.$$

Геометрические образы микрокоманд показаны на рис. 1, а. Один шаг преобразования исходной таблицы, содержащей слагаемые 9, 15, 5, приведен на рис. 1, б. После четырех шагов в верхней строке таблицы будет получена сумма, все остальные строки станут нулевыми.

2. Интерпретация параллельной микропрограммы сетью автоматов. Как известно [6, 8], параллельная микропрограмма $\Phi = \{\Theta_1, \dots, \Theta_n\}$ интерпретируется сетью автоматов. Каждой клетке с именем $m^0 \in M$ ставится в соответствие автомат с таким же именем и алфавитом внутренних состояний, равным A . Считается, что сеть интерпретирует параллельную микропрограмму, если и клетка, и соответствующий ей автомат, начав с одного и того же исходного состояния, далее переходят в совпадающие состояния на соответствующих шагах выполнения параллельной микропрограммы и работы сети.

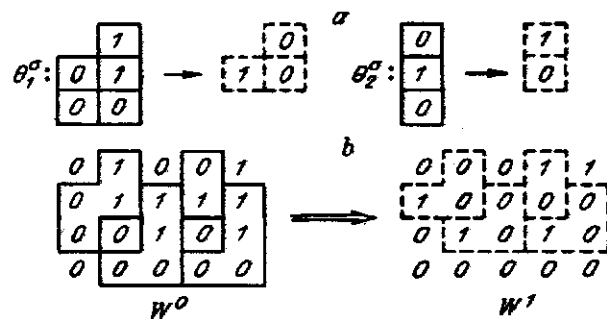


Рис. 1

Построим такую сеть. Составим два списка именуемых функций J^1 и J^2 . Список J^1 (J^2) — это перечень всех различных именуемых функций, использованных в записи левых (правых) частей микрокоманд микропрограммы Φ . Число функций в списке J^1 обозначим n , число функций в списке J^2 — k . Перенумеруем функции списка J^1 числами от 1 до n , а функции списка J^2 числами от 1' до k' . При построении сети решаются две задачи: выполняются межавтоматные связи и строится структурная схема каждого автомата.

Для каждого автомата с именем m^0 межавтоматные связи строятся на основании списков именуемых функций. Список J^1 , в функции которого вместо переменной m подставлено имя m^0 перечисляет имена автоматов-соседей, у которых выходы ячеек памяти, хранящих внутренние состояния, соединены со входами 1, ..., n автомата с именем m^0 в соответствии с номерами функций в списке J^1 . Список J^2 после выполнения замены m на m^0 перечисляет имена автоматов-соседей, чьи входы записи ячеек памяти соединены с выходами 1', ..., k' автомата с именем m^0 в соответствии с номерами функций в списке J^2 .

Как и принято, будем считать, что структурная схема каждого автомата состоит из двух блоков: ячейки памяти, предназначенной для хранения состояния автомата, и функционального преобразователя. Преобразователь выполним в виде микропрограммной памяти с ассоциативным механизмом выбора исполняемых микроопераций. Эта память содержит две таблицы: T^1 , в которую вносится список левых частей микроопераций, и T^2 , в которую вносится список правых частей микроопераций. Для заполнения таблиц используется текст микропрограммы Φ . Строки таблиц помечены именами микрокоманд $\Theta_1, \dots, \Theta_n$, столбцы — номерами функций из соответствующих списков J^1 и J^2 . Каждый столбец таблицы T^1 (T^2) имеет вход (выход) с таким же, как у самого столбца, номером. Если в левой части некоторой микрокоманды Θ_i присутствует компонента $(a, \varphi(m))$ и $\varphi(m)$ имеет номер j , в таблице T^1 на пересечении строки Θ_i и столбца с номером j ставится символ a . Если в Θ_i отсутствует компонента с функцией $\varphi(m) \in J^1$, имеющей номер j , в T^1 на пересечении строки Θ_i и столбца с номером j ставится символ $*$. Этот символ означает символ маскирования. Аналогичные действия выполняются при заполнении таблицы T^2 , но только символ $*$ заменяется символом $\#$, означающим пустой символ.

Сеть работает синхронно. Каждый ее автомат на первом такте выполняет ассоциативный поиск в своей таблице T^1 строки, совпадающей с вектором, поступающим на входы 1, ..., n , с точностью до символов маскирования, а на втором такте, если такая строка находится (пусть это будет строка Θ_i), выдает на выходы 1', ..., k' символы, записанные в строку Θ_i таблицы T^2 . Если таких строк несколько, автомат выдает на каждый выход объединение (в теоретическом множественном смысле) символов из соответствующих позиций этих строк. Такое действие корректно, так как параллельная микропрограмма непротиворечива [6], и, следовательно, в одинаковых позициях строк символы, отличные от пустых, совпадают. Наличие на некотором выходе пустого символа означает, что в связанную с этим выходом ячейку памяти запись нового состояния не производится.

Пример 2. Построение сети, интерпретирующей параллельную микропрограмму Φ .

$$J^1 = \{ \langle x, y + 1 \rangle, \langle x + 1, y \rangle, \langle x + 1, y - 1 \rangle, \langle x, y - 1 \rangle, \langle x, y \rangle \};$$

$$J^2 = \{ \langle x, y + 1 \rangle, \langle x + 1, y \rangle, \langle x, y \rangle \}.$$

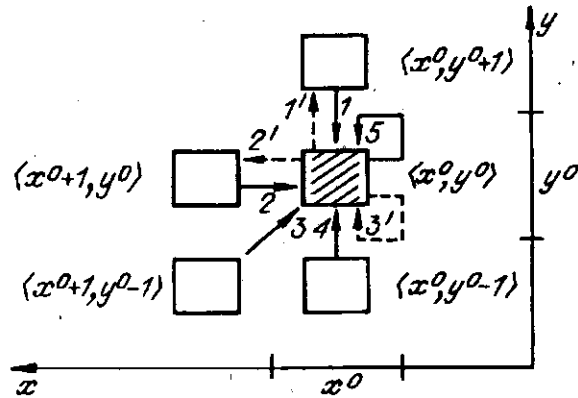


Рис. 2

Межавтоматные связи автомата с именем $\langle x^0, y^0 \rangle$ изображены на рис. 2. На рис. 3 приведена структурная схема автомата, состоящего из функционального преобразователя (таблицы T_σ^i и T_σ^j) и ячейки памяти P . Сеть, интерпретирующая Φ_σ , получается итеративным повторением фрагмента (см. рис. 2) так, чтобы его заштрихованная клетка совпадала с клетками с целочисленными координатами плоскости (x, y) . При этом у каждого автомата сети получают дополнительные входы и выходы, изображенные на рис. 3 без номеров. Далее эту сеть будем называть сумматором многих чисел, а ее автомат — разрядом.

Будем называть сеть *однослойной*, если в алгоритме, который она интерпретирует, используются имена из множества $M = N^2$ и все блоки структурной схемы автомата с именем $\langle x^0, y^0 \rangle$ занимают некоторую площадку, которой могут быть приписаны координаты (x^0, y^0) в плоскости (x, y) . В том случае, когда блоки структурной схемы автомата занимают несколько площадок, расположенных одна под другой, в целой группе плоскостей, сеть будем называть *многослойной*.

Далее покажем, как из однослойной сети может быть получена многослойная сеть.

3. Декомпозиция функционального преобразователя автомата сети. *Фрагментом* таблицы T^i (T^j) называется непустое подмножество столбцов таблицы T^i (T^j).

Пусть T_1^i, \dots, T_h^i — фрагменты таблицы T^i .

Совокупность $\{T_1^i, \dots, T_h^i\}$ называется *разбиением* таблицы T^i , если:

1) для любых i, j ($i \neq j, i, j \leq h$) справедливо, что $T_i^i \cap T_j^i = \emptyset$;

2) $\bigcup_{i=1}^h T_i^i = T^i$.

Аналогично определяется разбиение на фрагменты таблицы T^j .

Пусть таблица T^i разбита на h фрагментов: $T_1^i = \{i_1, \dots, i_p\}, \dots, T_h^i = \{i_q, \dots, i_n\}$, а таблица T^j — на d фрагментов: $T_1^j = \{j_1, \dots, j_u\}, \dots, T_d^j = \{j_v, \dots, j_k\}$. В скобках указаны номера столбцов, вошедших в соответствующий фрагмент. Исходя из такого разбиения таблиц T^i, T^j , выполним декомпозицию функционального преобразователя

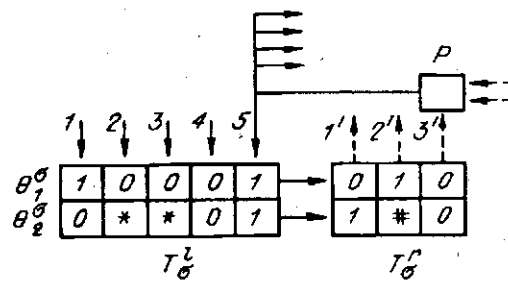


Рис. 3

ля автомата сети так, как это показано на рис. 4. Получается функциональный преобразователь, состоящий из блоков B_1^1, \dots, B_h^1 и B_1^2, \dots, B_d^2 , каждый из которых, в свою очередь, является функциональным преобразователем. Таблицы, хранящие фрагменты T_1^1, \dots, T_h^1 и образующие левые части преобразователей B_1^1, \dots, B_h^1 , так же, как и таблицы, хранящие фрагменты T_1^2, \dots, T_d^2 и образующие правые части преобразователей B_1^2, \dots, B_d^2 , просто аналогичны соответствующим таблицам T^1 и T^2 . Правые части преобразователей B_1^1, \dots, B_h^1 образованы разрядами блоков B_1^3, \dots, B_h^3 , каждый блок выполняет функцию И от своих входов. Внешний вход $\gamma_i, i = 1, \dots, \nu$, поставлен в соответствие Θ_i -й микрооперации, и на него поступает константа 1. Левые части преобразователей B_1^2, \dots, B_d^2 образованы разрядами блоков B_1^4, \dots, B_d^4 . Эти блоки можно назвать демультиплексорами: каждый из них превращает один входной сигнал в d сигналов, обеспечивающих одновременное считывание d строк из таблиц, хранящих фрагменты T_1^2, \dots, T_d^2 .

Функциональные преобразователи (исходный и изображенный на рис. 4) различаются нумерацией входов (выходов) и тем, что в преобразователе каждая строка левой и правой частей исходного преобразователя разделена на части. Однако с логической точки зрения они эквивалентны, так как:

- 1) и в том и в другом случае сохраняется соответствие между номером входа (выхода) функционального преобразователя и именем автомата-соседа;
- 2) части каждой строки в блоках B_1^1, \dots, B_h^1 склеиваются с помощью схемы И в ту же строку, что и в таблице T^1 ;
- 3) из блоков B_1^2, \dots, B_d^2 с помощью демультиплексора может быть считана (по частям) та же самая строка, что и из таблицы T^2 .

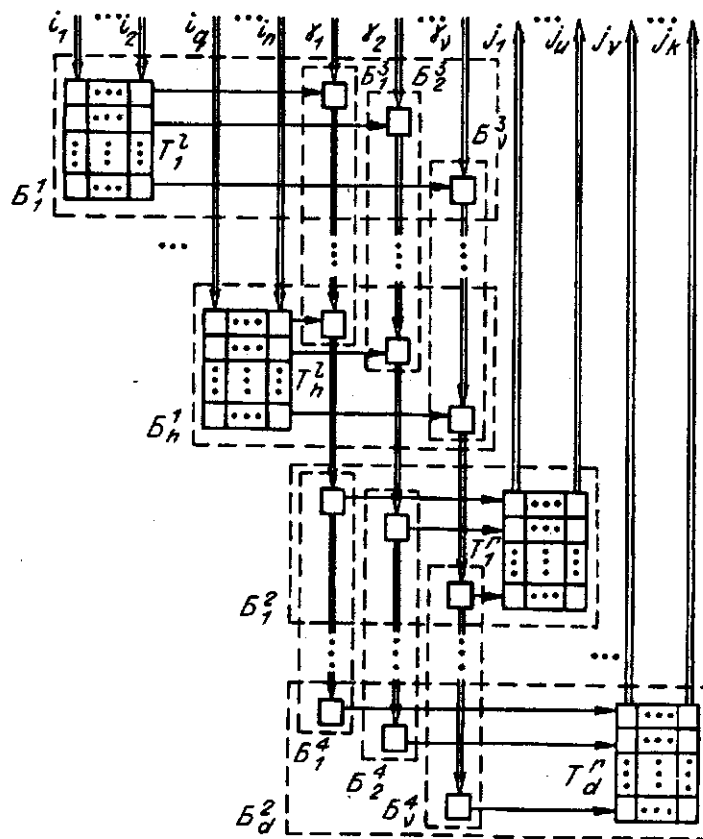


Рис. 4

4. Расслоение плоской сети.

На предложенном способе декомпозиции основан прием расслоения исходной плоской сети. Блоки, из которых в результате декомпозиции оказывается построен каждый ее автомат, распределяются между слоями следующим образом. Ячейка памяти располагается в нулевом слое, блок B_1^1 — в первом слое, ..., блок B_h^1 — в h -м слое, блок B_1^2 — в $(h+1)$ -м слое, ..., блок B_d^2 — в $(h+d)$ -м слое. При таком распределении блоков все связи, изображенные на рис. 4 одинарными линиями, оказываются внутрислойными связями, а связи, изображенные двойными линиями, — межслойными.

При выполнении декомпозиции таблицы T^1 и T^r были разделены на фрагменты по столбцам. Конечно, это не единственный способ декомпозиции. Аналогичное разделение можно провести по строкам, соблюдая лишь одно дополнительное условие: разделение таблиц T^1 и T^r должно проводиться согласованно, т. е. для каждого фрагмента в T^1 должен найтись фрагмент в T^r , имеющий такие же имена строк. Более того, можно проводить декомпозицию, выполняя совместное разбиение по столбцам и строкам.

Пример 3. На рис. 5 показан функциональный преобразователь для разряда с именем (x^0, y^0) сумматора многих чисел, полученный в результате предельного (вплоть до отдельной клетки) разбиения исходных таблиц T_o^1 , T_o^r . Расслоенный сумматор содержит 14 слоев. Номера слоев функционального преобразователя указаны на рис. 5 слева. Сделаем некоторые пояснения к рисунку. Входной сигнал (вход 1) с первого слоя на шестой слой должен проходить «сквозь» элементы первого слоя. Это означает, что элементы первого слоя не только выполняют свою функцию, но и пропускают входной сигнал. Аналогично действуют элементы четвертого и пятого (входы 4 и 5), девятого и одиннадцатого (выходы 1', 3') слоев.

При разбиении таблицы T^1 (T^r) могут быть получены одинаковые фрагменты. В этом случае разные слои, построенные на основе совпадающих фрагментов, могут быть склеены в один слой.

Пример 4. На рис. 5 такими слоями являются слои 4, 7 и 5, 8, полученные при разбиении таблицы T_o^1 , и слои 11, 13, полученные при разбиении таблицы T_o^r . Функциональный преобразователь со склеенными слоями показан на рис. 6. На этом рисунке зачерненным квадратиком изображена схема ИЛИ.

Строго говоря, сеть со склеенными слоями уже не является предельно расслоенной, но мы будем считать ее таковой, потому что, как увидим далее, при очевидном уменьшении числа слоев топология склеенных слоев практически не отличается от топологии остальных слоев.

Используя описанные приемы расслоения, при электрооптической реализации расслоенной сети можно получить электрооптическую структуру с очень простой топологией отдельного слоя. Покажем это.

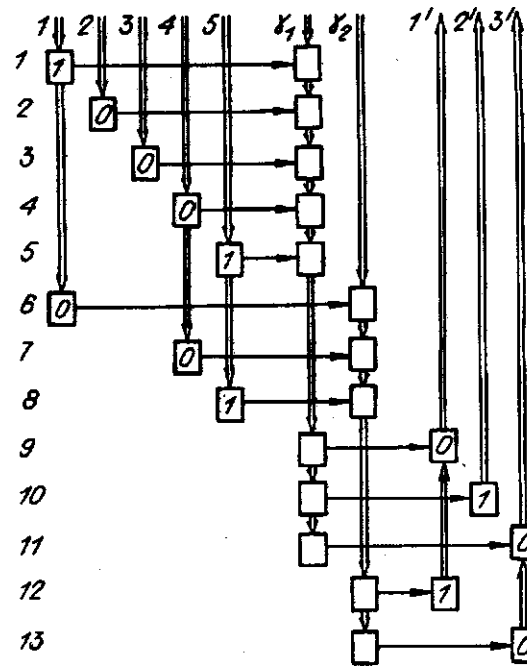


Рис. 5

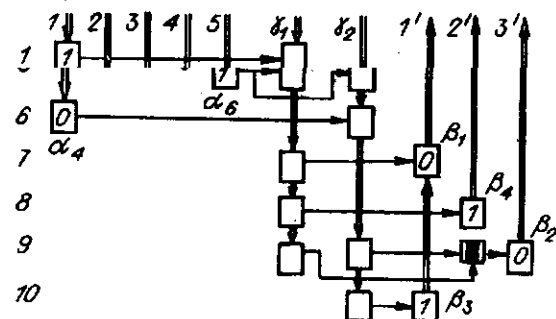


Рис. 6

5. Электрооптическая реализация предельно расслоенной сети. Принцип, положенный в основу электрооптической реализации предельно расслоенной сети, очень прост: все межслойные связи делаются оптическими, внутрислойные — электрическими. Выполнение электрооптической реализации предельно расслоенной сети покажем на примере сумматора многих чисел, имеющего у каждого своего разряда функциональный преобразователь, изображенный на рис. 6. Будем ориентироваться на реализацию межслойных связей с использованием потенциальных электрооптических ячеек [3—5]. Электрооптическую ячейку будем изображать в виде двух квадратов, соединенных линией, причем зачерненный квадрат — это фотоприемник, а заштрихованный — модулятор.

Пример 5. Возьмем 11 слоев. Разделим плоскость каждого слоя на прямоугольные площадки-клетки одинаковой величины. Каждому разряду с именем (x^0, y^0) поставим в соответствие 11 расположенных одна под другой клеток, которым припишем одинаковые целочисленные координаты (x^0, y^0) в соответствии с левой системой координат. Выделим в каждой клетке семь

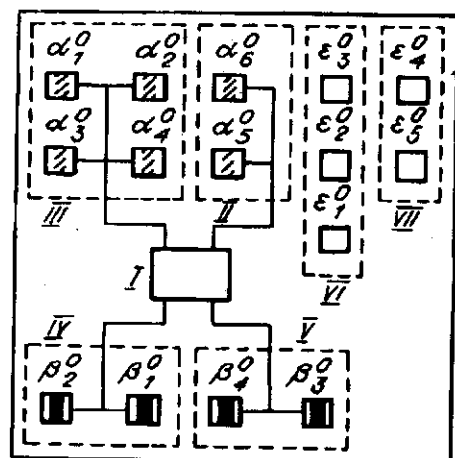


Рис. 7

прямоугольных площадок-фрагментов, обозначенных римскими цифрами от I до VII. На фрагментах с номерами II—VII площадками-квадратиками изобразим позиции с номерами $\epsilon_1 + \epsilon_5$ и номерами $\alpha_1 + \alpha_5$, $\beta_1 + \beta_4$, совпадающими с номерами элементов на рис. 6. Введем обозначение: позицию z_i в j -м слое будем записывать как z_i^j . Положение позиций в клетках нулевого слоя показано на рис. 7. Фрагмент I предназначен для размещения ячейки памяти: он используется только в нулевом слое. Примем «позиционную» кодировку символов алфавита $\{0, 1\}$ и номеров входов $\{\gamma_1, \gamma_2\}$, т. е. для их представления будем отводить определенные позиции в каждой

клетке. Для представления символа 0 отведем позиции фрагментов III и IV, для представления символа 1 — позиции фрагментов II, V, для представления номера γ_1 — позиции фрагмента VI, для представления номера γ_2 — позиции фрагмента VII.

Разделим каждую клетку плоскости на две части (A и B) так, чтобы в A оказались фрагменты II, III, VI, VII, а в B — IV, V. Расположим источники света таким образом, чтобы на клетки в части A световой поток падал «сверху» (в направлении от нулевого слоя к десятому), а в части B — в противоположном направлении.

Поместим в позиции $\alpha_1^0 + \alpha_6^0$ модуляторы, в позиции $\beta_1^0 + \beta_4^0$ фотоприемники и соединим их с нулевыми и единичными входами и выходами ячейки памяти так, как показано на рис. 7. Единичный вход у ячейки памяти находится внизу справа, единичный выход — вверху справа. При итеративном повторении на плоскости клетки (см. рис. 7) получается топология нулевого слоя.

Позиция, в которую не помещен ни модулятор, ни фотоприемник, — это окно в слое: оно всегда прозрачно. Далее на рисунках окна изображаться не будут.

Покажем, как блоки функционального преобразователя реализуются клетками электрооптического устройства. На рис. 6 положение блоков не указано, чтобы не загромождать рисунок, но их расположение легко себе представить, сопоставив рис. 4 и 6.

Поместим в позиции $\epsilon_1^j + \epsilon_5^j$ слоев с номерами $j = 1, \dots, 5$ модуляторы. В каждом слое соединим их в один «трехлепестковый» модулятор. Получим реализацию блока B_1^3 . Действительно, свет сквозь столбик из трехлепестковых модуляторов на ниже расположенные слои проходит только тогда, когда все они прозрачны. Аналогично может быть выполнен блок B_2^3 из модуляторов, помещенных в позиции ϵ_4^j и ϵ_5^j слоев с $j = 4, 5, 6$.

Расположив в позициях $\epsilon_1^7, \epsilon_2^8, \epsilon_3^9, (\epsilon_4^9, \epsilon_5^{10})$ фотоприемники, реализуем блок B_1^4 (B_2^4), который размножает выход блока B_1^3 (B_2^3) в трех (двух) слоях.

Выполним реализацию любого из блоков B_j^3 , например B_1^3 . Функция этого блока состоит в том, что в нем сравнивается состояние ячейки памяти первого соседа (его имя $(x^0, y^0 + 1)$) с символом 1 алфавита A, и, если сравнение успешно, на вход первого разряда блока B_1^3 поступает разрешающий сигнал. Мы реализуем эту функцию, если в позицию α_5^1 клетки с координатами $(x^0, y^0 + 1)$ поместим фотоприемник и затем объединим его и трехлепестковый модулятор из клетки с координатами (x^0, y^0) первого слоя в электрооптическую ячейку.

Введем обозначение: электрооптическую ячейку, расположенную в u -м слое ($1 \leq u \leq 10$) и образующуюся при соединении фотоприемника, размещенного в позиции z_u^0 клетки с координатами (x_1^0, y_1^0) , с модулятором, размещенным в позиции s_u^0 клетки с координатами (x_2^0, y_2^0) , будем описывать выражением

$$z_u^0(x_1^0, y_1^0) \Rightarrow s_u^0(x_2^0, y_2^0). \quad (1)$$

Если слева от стрелки расположено несколько элементов, соединенных друг с другом, их перечни будем заключать в квадратные скобки.

Обозначим: $H^0(x^0, y^0)$ — трехлепестковый модулятор, $V^0(x^0, y^0)$ — двухлепестковый модулятор, $F^0(x^0, y^0)$ — пятилепестковый модулятор клетки с координатами (x^0, y^0) u -го слоя, (составлены соответственно из модуляторов, размещенных в позициях фрагмента VI, фрагмента VII и объединения фрагментов VI и VII).

Выражение $\alpha_5^1(x^0, y^0 + 1) \Rightarrow H^1(x^0, y^0)$ при замене x^0 и y^0 числами 1, 2, 3, ... задает топологию первого слоя.

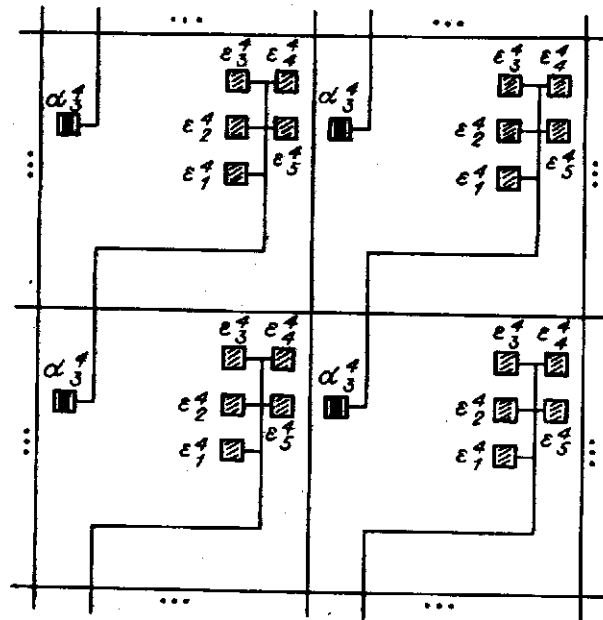


Рис. 8

Рассуждая сходным образом, можно выполнить размещение модуляторов и фотоприемников в других слоях устройства и реализовать остальные блоки функционального преобразователя.

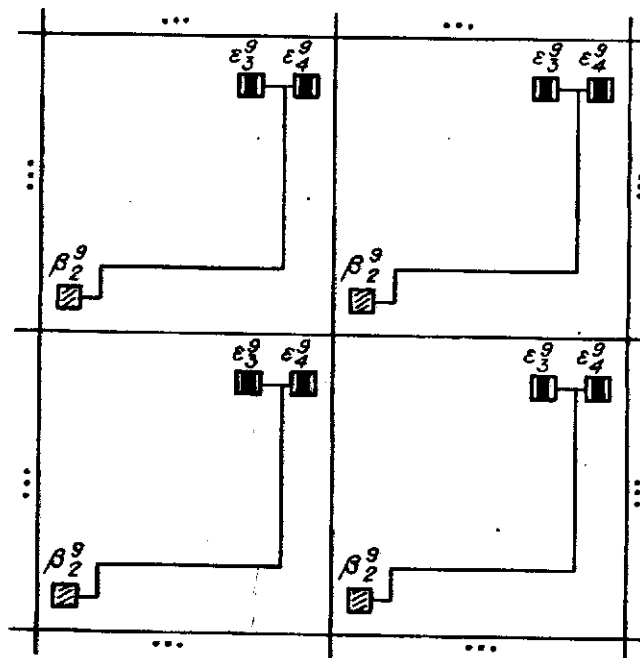


Рис. 9

Заменяя по очереди в (1) левую часть фотоприемниками $\alpha_1^2(x^0 + 1, y^0)$, $\alpha_2^3(x^0 + 1, y^0 - 1)$, $\alpha_3^4(x^0, y^0 - 1)$, $\alpha_4^5(x^0, y^0)$, $\alpha_5^6(x^0, y^0 + 1)$, $\epsilon_1^7(x^0, y^0)$, $\epsilon_2^8(x^0, y^0)$, $[\epsilon_3^9(x^0, y^0), \epsilon_4^9(x^0, y^0)]$, $\epsilon_5^{10}(x^0, y^0)$, а правую часть соответственно модуляторами $H^2(x^0, y^0)$, $H^3(x^0, y^0)$, $F^4(x^0, y^0)$, $F^5(x^0, y^0)$, $V^6(x^0, y^0)$, $\beta_1^7(x^0, y^0 + 1)$, $\beta_2^8(x^0 + 1, y^0)$, $\beta_3^9(x^0, y^0)$, $\beta_4^{10}(x^0, y^0 + 1)$, можно получить топологию слоев во второго по десятый.

На рис. 8 показано расположение на плоскости электрооптических ячеек четвертого, а на рис. 9 — девятого слоев устройства. Разветвление выхода элемента α_3 (см. рис. 6) осуществляется в электрооптической ячейке четвертого слоя пятилепестковым модулятором, схема ИЛИ реализуется при соединении пары фотоприемников в электрооптической ячейке девятого слоя.

Итак, слои получаются простыми, но их может быть очень много. Возникает вопрос: как, если потребуются, сократить число слоев? Рецепт таков: нужно склеивать слои, превращая межслойные оптические связи во внутрислойные электрические (обратное преобразование). Например, чтобы склеить первый и второй слои (см. рис. 6), нужно фотоприемник из позиции $\alpha_1^2(x^0 + 1, y^0)$ переместить в такую же позицию первого слоя и соединить с фотоприемником в позиции $\alpha_3^1(x^0, y^0 + 1)$ посредством конъюнкции. Новый слой описывается выражением

$$\alpha_3^1(x^0, y^0 + 1) \& \alpha_1^2(x^0 + 1, y^0) \Rightarrow H^1(x^0, y^0).$$

Достоинством такого способа уменьшения числа слоев является свобода выбора склеиваемых слоев, что позволяет широко варьировать топологию слоя.

СПИСОК ЛИТЕРАТУРЫ

1. Бандман О. Л. Электрооптическая реализация клеточных структур обработки данных // Автометрия.—1992.—№ 1.
2. Пискунов С. В. Построение многослойных электрооптических структур // Автометрия.—1992.—№ 4.
3. Косцов Э. Г., Мишин А. И. Особенности построения оптических ЦВМ // Микроэлектроника.—1977.—6, вып. 2.
4. Егоров В. М., Косцов Э. Г. Перспективы создания оптических цифровых высокопроизводительных вычислительных устройств // Автометрия.—1985.—№ 1.
5. Egorov V. M., Kostsov E. G. Integral optical digital computers // Appl. Opt.—1990.—29, N 8.—P. 1178.
6. Методы параллельного микропрограммирования /Под ред. О. Л. Бандман.—Новосибирск: Наука, 1981.
7. 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.
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.

Поступила в редакцию 4 февраля 1993 г.