ОДНОРОДНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
(Вычислительные системы)
1981 год Выпуск 90

УДК 681.327

ОДНОРОДНАЯ ВЫЧИСЛИТЕЛЬНАЯ СТРУКТУРА ИЗ МНОГОЗНАЧНЫХ ЯЧЕЕК, РЕАЛИЗУЮЩАЯ АЛГОРИТМЫ ПАРАЛЛЕЛЬНЫХ ПОДСТАНОВОК

С.В. Пискунов

1. Необходимость решать все более сложные задачи приводит к тому, что в вычислительных устройствах возрастает число элементов, увеличиваются функциональные возможности элементов, усложняются связи элементов друг с другом. В связи с достижением очень высокой степени интеграции (близкой к предельной) при создании БИС и большой роли межсоединений все большее внимание в вычислительной технике уделяется устройствам, построенным на основе многозначных схем. Применение таких схем существенно увеличивает информационную плотность на единицу площади и, из-за возможности использовать те же самые сигнальные цепи для передачи нескольких уровней информации, в несколько раз снижает количество внешних выводов [1-3]. Это направление удачно сочетается с построением однородных вычислительных структур [4,5].

Статья посвящена вопросам реализации однородной параллельной структуры, конструктивная сложность ячеек которой сопоставима со сложностью ячеек вычислительных сред, а функциональные возможности с функциональными возможностями микропроцессоров. Это достигается при помощи микропрограммного представления процесса переработки информации в каждой ячейке структуры, сочетания частотного и пространственного кодирования символов микрокоманд, записанных в многозначном алфавите, и использования принципов построения многоустойчивых элементов при реализации ячеек структуры.

Отметим, что структура относится к классу устройств с множественным потоком команд и множественным потоком данных в соответствии с классификацией в [6].

Основой для микропрограммного представления процесса переработки информации в предлагаемой структуре служит алгоритмическая система - алгоритмы параллельных (обобщенных) подстановок [7,8]. Она базируется на понятиях клеточного автомата [9] и нормального алгоритма [10]. Приведем ее краткое описание.

2. Алгоритмы параллельных подстановок. Пусть A - конечный алфавит, M - множество имен с мощностью не более чем счетной. Клеткой m, которая находится в состоянии a, называется пара $(a,m) \in A \times M$. В алгоритмах параллельных подстановок основным объектом преобразования является клеточное множество W - конечная совокупность клеток, в которой нет ни одной пары клеток с одинаковыми именами. Через $K(A,M)$ обозначим множество всех клеточных множеств в (A, M). Для любого конечного k запишем множество пар $\{(a_1,\varphi_1(m)) \ldots (a_k,\varphi_k(m))\}$, где $a_i \in A$, $\varphi_i: M \xrightarrow{B} M$ такие, что для $i \neq j$, $i,j = 1,2,\ldots,k$ и $\forall m \in M$, $\varphi_i(m) \neq \varphi_j(m)$. Упорядочим элементы M: $\{m_1, m_2, \ldots, m_i, \ldots\}$. Взяв некоторое $m_i \in M$ и подставив вместо m во все функции φ_i, $i=1,2,\ldots,k$, получим клеточное множество $W_i = \{(a_1, \varphi_1(m_i)) \ldots (a_k, \varphi_k(m_i))\}$. Проделаем эту процедуру для всех $i = 1,2,\ldots,|M|$.

Конфигурацией S в $K(A,M)$ называется множество $\{W_i\}_{i=1,2,\ldots,|M|}$. Записывать конфигурацию будем как $S = \{(a_1,\varphi_1(m)) \ldots (a_k,\varphi_k(m))\}$, а конкретный ее элемент W_i как $S(m_i)$.

Пусть даны две конфигурации $S_1 = \{(a_1,\varphi_1(m)) \ldots (a_{n_1},\varphi_{n_1}(m))\}$ и $S_2 = \{(b_1,\psi_1(m)) \ldots (b_{n_2}, \psi_{n_2}(m))\}$, такие что для $\forall m \in M$, $\forall i,j$, $i = 1,\ldots,n_1$, $j = 1,\ldots,n_2$, $\varphi_i(m) \neq \psi_j(m)$. Тогда произведением S конфигураций S_1 и S_2 называется конфигурация вида $\{(a_1,\varphi_1(m)) \ldots (a_{n_1},\varphi_{n_1}(m)), (b_1,\psi_1(m)) \ldots (b_{n_2},\psi_{n_2}(m))\}$. Произведение S будем записывать как $S_1 * S_2$.

Процесс переработки клеточного множества $W \in K(A,M)$ заключается в применении к нему операций подстановок. Выражение вида $\Pi: S_1 * S_2 \to S_3$, где S_1, S_2, S_3 - конфигурации, $S_1 * S_2 = S$ - произведение конфигураций, называется подстановкой. Величина S называется левой частью, S_1 - базовой частью, S_2 - контекстом, S_3 - правой частью. Если существуют $m_1, m_2, \ldots, m_p \in M$ такие, что $S(m_1), S(m_2), \ldots, S(m_p) \subseteq W$, то подстановка считается применимой к W и результат применения подстановки есть клеточное множество $\{W\setminus \bigcup_{i=1}^{p} S_2(m_i)\} \cup \{\bigcup_{i=1}^{p} S_3(m_i)\}$. Если таких имен не существует, П неприменима к W.

Конечное множество Ф подстановок, записанных в произвольном порядке, называется с и с т е м о й параллельных п о д с т а н о в о к. Применение системы к $W \in K(A,M)$ задается следующей итерационной процедурой. Пусть клеточное множество W^{i-1} - результат (i-1)-й итерации. Если ни одна из подстановок неприменима к W^{i-1}, то W^{i-1} является результатом применения Ф к W. Если же какие-то подстановки алгоритма Ф применимы (будем обозначать их символами $П^1, П^2, \ldots, П^k$), то результатом применения Ф к W^{i-1} является клеточное множество

$$W^i = \{W^{i-1}\setminus \bigcup_{j=1}^{k} (\bigcup_{i=1}^{p^j} S_2^j(m_i^j))\} \cup \{\bigcup_{j=1}^{k} (\bigcup_{i=1}^{p^j} S_3^j(m_i^j))\}.$$

Оно служит исходным для следующей итерации. Система Ф вместе с такой процедурой применения называется а л г о р и т м о м п а р а л л е л ь н ы х п о д с т а н о в о к.

ПРИМЕР. Алгоритм сложения многих двоичных чисел. $A = \{0,I\}$, $M = N \times N$, где N - натуральный ряд. Клеточное множество W - это двумерная бинарная таблица, клетки которой пронумерованы в соот - ветствии с левой системой координат (i-абсцисса, j - ордината). В строках таблицы расположены двоичные слагаемые. Размеры таблицы определяются числом слагаемых и их разрядностью. Нижняя строка таблицы нулевая. Младшие разряды слагаемых записаны в столбце с $i = I$. Для функции $\varphi(i,j)$ из $M \to M$ вида $(i,j) \to (i+a,\ j+b)$, где a, b - некоторые целые константы, будем пользоваться сокращенной записью $(i+a,\ j+b)$. Введем функции (i,j), $(i,j+1)$, $(i,\ j-1)$, $(i+1,j)$, $(i+1,\ j-1)$. Алгоритм содержит подстановки:

$$П_1: \{(1,(i,j))(1,(i,j+1))(0,(i+1,j))\} * \{(0,(i+1,j-1))(0,(i,j-1))\} \to$$
$$\to \{(0,(i,j))(0,(i,j+1))(1,(i+1,j))\};$$

$$П_2: \{(1,(i,j))(0,(i,j+1))\} * \{(0,(i,j-1))\} \to \{(0,(i,j))(1,(i,j+1))\}.$$

В этом примере клеточные множества и элементы конфигураций можно изображать графически. На рис.I показаны конфигурации подстановок $П_1$ и $П_2$ и один шаг применения алгоритма.

Для широкого класса алгоритмов (алгоритмов стационарных подстановок) [7] всегда возможно построение однородной сети конечных

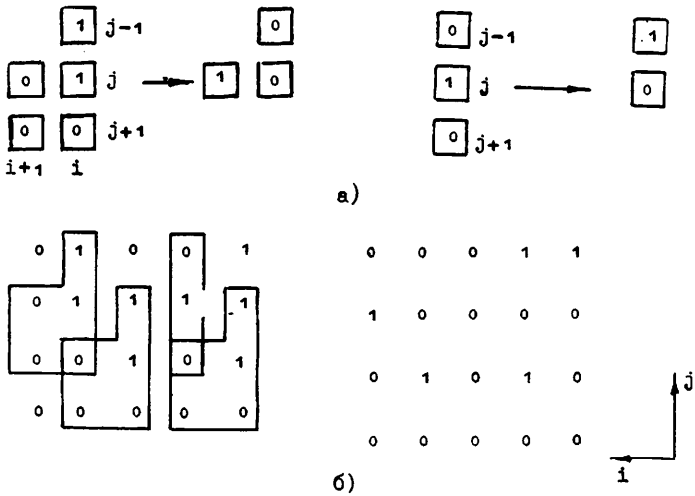

Рис. I

автоматов (ячеек), выполняющей либо отдельный алгоритм, либо некоторую совокупность алгоритмов параллельных подстановок. Выбирая тот или иной способ реализации сети, можно получать конкретные устройства. Например, в [II] описано специализированное устройство, реализующее алгоритм сложения. Оно представляет собою матрицу из одинаковых ячеек, содержащих комбинационную схему и двоичный элемент памяти, хранящий разряд числа. Соединение ячеек друг с другом определяется функциями, используемыми в записи Π_1 и Π_2, и показано на рис.2.

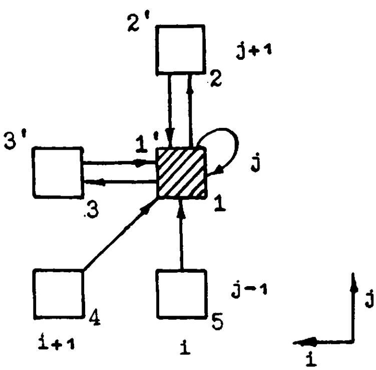

Рис. 2

Далее используется реализация ячейки сети в виде микропрограммного автомата. Для осуществления такой реализации необходимо выполнить три шага:

I) переход от алгоритма параллельных подстановок к микропрограмме;

2) кодирование микрокоманд;

3) вложение микропрограммы в ассоциативный блок ячейки.

В п.3 на примере рассматриваются эти шаги для видеоимпульсного кодирования информации и двоичных элементов памяти и логических схем. В последующих пунктах описывается новый подход к реализации второго и третьего шага, приводящий к цели, поставленной во введении.

3. **Микропрограммная реализация ячеек однородной структуры.** Ячейка однородного устройства состоит из ассоциативной матрицы, хранящей коды микрокоманд, коммутатора и элемента памяти, хранящего внутреннее состояние микропрограммного автомата.

Рассмотрим построение такой ячейки для приведенного выше примера алгоритма сложения многих чисел.

Первый шаг. Топология связей ячеек друг с другом соответствует рис. 2. По подстановкам $П_1$ и $П_2$ записываются две микрокоманды:

$$П'_1:\ II000 \rightarrow 00I$$
$$П'_2:\ I0■■0 \rightarrow 01\sqcup.$$

Символы в левых частях микрокоманд упорядочены в соответствии с нумерацией без штриха (см. рис.2), указывающей те ячейки, от которых выделенная ячейка получает информацию. Символы в правых частях микрокоманд упорядочены в соответствии с нумерацией со штрихом (см.рис.2), указывающей те ячейки, в которые выделенная записывает информацию. Состояния элементов памяти ячеек, не участвующих в записи некоторой команды подстановки, заменяются в левой части соответствующей микрокоманды символом ■ (маскирование), в правой - символом ␣ (пробел).

Второй шаг. Примем следующий вариант кодировки микрокоманд. Микрокоманде $П'_1$ сопоставляется код, содержащий две строки: $\boxed{II000 \mid 00I}$ - код собственно микрокоманды и $\boxed{00000 \mid III}$ - код маски. Для получения кода микрокоманды $П'_2$ производится доопределение (произвольное) разрядов, содержащих ■ и ␣, в первой строке. Код, например, может состоять из строк $\boxed{I0I00 \mid 0I0}$, $\boxed{00II0 \mid II0}$.

Третий шаг. Ассоциативная матрица не содержит принципиальных отличий от устройств, описанных в [12]. Для рассматриваемого примера она содержит две строки. Каждая строка состоит из двух час -

тей: левой – для хранения кода левой части микрокоманды и правой – для хранения кода правой части микрокоманды. Каждый разряд ассоциативной матрицы содержит два элемента памяти: один – для хранения символа кода собственно микрокоманды, другой – символа кода маски. Код опроса образуют текущие внутренние состояния ячеек, упорядоченные в соответствии с нумерацией без штриха на рис.2. В ассоциативной матрице обеспечивается сравнение соответствующих разрядов кода опроса и кодов левых частей микрокоманд и операция ИЛИ над результатом сравнения и левой частью кода маски. Символы правой части микрокоманды, левая часть которой (с учетом кода маски) совпала с кодом опроса, через коммутатор в соответствии с нумерацией со штрихом записываются в элементы памяти ячеек. Разрешение записи обеспечивает символ I в соответствующем разряде правой части кода маски.

При увеличении числа микрокоманд растет число строк в матрице, при увеличении разрядности микрокоманды – число элементов в строке матрицы. Это означает, что структурная сложность ячейки (а значит, и устройства в целом) пропорциональна числу микрокоманд в алгоритме. Избежать увеличения сложности устройства при росте числа микрокоманд позволяют использование многозначного структурного алфавита и сочетание частотного и пространственного кодирования символов микрокоманд в оптическом диапазоне частот сигналов.

4. К о д и р о в а н и е м и к р о п р о г р а м м ы. Пусть $A = \{a_1, a_2, \dots, a_k\}$ – алфавит микрокоманд и $\mathfrak{A} = \{\Pi_1, \Pi_2, \dots, \Pi_n\}$ – некоторый алгоритм параллельных подстановок. Сделаем равными длины левых частей всех микрокоманд, полученных из $\mathfrak{A}$, при помощи символов $\blacksquare$. Аналогично поступим с правыми частями микрокоманд, используя символ $\sqcup$. Запишем микрокоманды алгоритма в виде табл.I, в которой $a_{ij} \in A \cup \blacksquare$, $b_{im} \in A \cup \sqcup$, $i = 1, \dots, n$, $j = 1, \dots, p$, $m = 1, \dots, q$.

Т а б л и ц а I

Π'_1	a_{11}	a_{12}	...	a_{1p}	b_{11}	b_{12}	...	b_{1q}
...		...				...		
Π'_i	a_{i1}	a_{i2}	...	a_{ip}	b_{i1}	b_{i2}	...	b_{iq}
...		...				...		
Π'_n	a_{n1}	a_{n2}	...	a_{np}	b_{n1}	b_{n2}	...	b_{nq}

Каждому символу $a_i \in A$ поставим во взаимно-однозначное соответствие оптический сигнал с частотой ω_i (параметр ω_i), символу $\sqcup$ – сигнал, являющийся суперпозицией сигналов с частотами $\omega_1, \omega_2, \dots, \omega_k$ (обозначается как c_ω).

Левой части табл.I ставится в соответствие некоторый сигнал (оптический луч), поперечное сечение которого представляет собой множество прямоугольных участков, организованных в столбцы и строки. Число строк равно n, столбцов – p. Строки отделены одна от другой зонами, в которых сигнал отсутствует. Прямоугольник с координатами (i,j) соответствует символу a_{ij} и несет сигнал с частотой ω_{ij}, если $a_{ij} \in A$, или сигнал c_ω, если $a_{ij} = \sqcup$.

Поставим во взаимно-однозначное соответствие каждому символу $a_s \in A$ периодическую (с периодом τ) функцию $m_s(t)$.

Для кодирования правых частей микрокоманд каждому символу b_{im} $(i = 1,\dots,n;\ m=1,\dots,q)$ ставится во взаимно-однозначное соответствие периодическая (с периодом $T = q\tau$) функция:

$$M_{im}(t+(m-1)\tau) = \begin{cases} m_s(t), \text{если } b_{im} = a_s \text{ и } (m-1)\tau \le t(\mathrm{mod}\ T) \le m\tau, \\ 0, \text{если } b_{im} = \sqcup \text{ или } (m-1)\tau > t(\mathrm{mod}\ T), \\ \qquad\qquad \text{или } m\tau \le t(\mathrm{mod}\ T). \end{cases}$$

Правой части микрокоманды Π'_i ставится в соответствие

$$M_i = \sum_{m=1}^{q} M_{im}(t + (m-1)\tau).$$

Чтобы получить в оптическом луче полную кодировку табл.I, амплитуда сигнала в каждой i-й строке поперечного сечения луча модулируется по закону M_i. Такое представление микропрограммы будем называть о б р а з о м а л г о р и т м а.

Покажем, как свернуть представление табл.I в поперечном сечении луча, чтобы одна строка сечения представляла несколько микрокоманд подстановки.

Алгоритм $\{\Pi'_1, \Pi'_2, \dots, \Pi'_n\}$ разбивается на несколько блоков (например, l). Разбиение на блоки имеет вид: $\Pi'^{1}_1, \dots, \Pi'^{1}_{d_1}, \quad \Pi'^{2}_1, \dots$ $\dots, \Pi'^{2}_{d_2}, \Pi'^{3}_1, \dots, \Pi'^{l}_1, \dots, \Pi'^{l}_{d_l}$. Принадлежность блоку указывается верхним индексом. Микрокоманды внутри каждого блока нумеруются числами $1, 2, \dots, d_z$ $(z = 1, \dots, l)$. Максимальное d_z будем обозначать буквой d. Поперечное сечение луча содержит p столбцов и d

строк. Всем подстановкам с номером j, $j = 1,\dots,d$, из всех l блоков отводится одна строка из сечения луча. В этой строке i-й прямоугольник отводится для представления всех символов $a^1_{ji}, a^2_{ji}, \dots, a^l_{ji}$ табл.1. Осуществляется это заданием поля излучения в каждой точке прямоугольника в виде суперпозиции сигналов с частотами $\Omega_1 + \omega^1_{ji}, \dots, \Omega_l + \omega^l_{ji}$. Параметры, характеризующие блоки и сим-

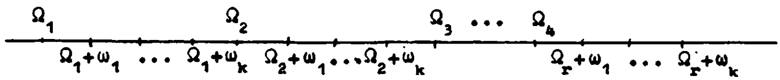

Рис. 3

волы алфавита,могут быть выбраны так,как показано на рис.3.Так же, как и в первом случае,каждый из сигналов модулирован по закону, представляющему правую часть соответствующей микрокоманды.

5. В л о ж е н и е м и к р о п р о г р а м м ы в а с с о ц и а т и в н ы й б л о к я ч е й к и. Оптический образ алгоритма формируется в некотором блоке излучения (в принципе,одном для всей однородной структуры) при помощи источников света, имеющих необходимый набор частот,и электрически управляемых транспарантов, задающих двумерную структуру луча и соответствующую модуляцию. На каждую ячейку структуры образ алгоритма поступает из блока излучения, например, через мультипликатор.

Детальное описание блок-схемы ячейки приведено в [13].Ячейка содержит ассоциативный блок, коммутатор и элемент памяти. В ассоциативном блоке можно выделить три подблока: перестраиваемый фильтр,устройство его перестройки и детектор. Фильтр представляет собой пластину сегнетоэлектрика с напыленными на нее р парами прозрачных электродов (перестраиваемый фильтр Фабри-Перо). На фильтр проектируется оптический образ алгоритма. Расположение электродов

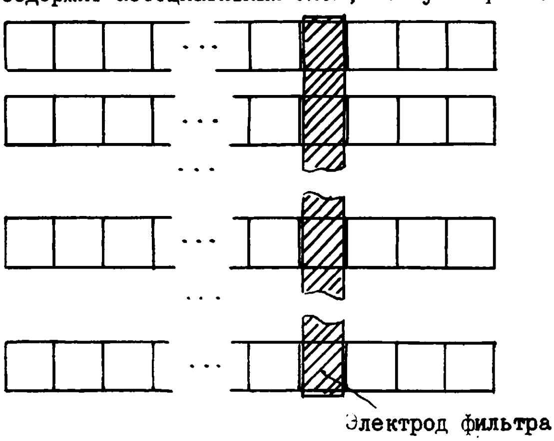

Рис. 4

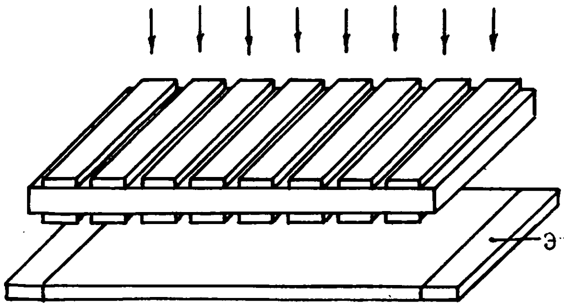

Рис. 5

по отношению к строкам сечения луча показано на рис.4.Электроды подключены к выходам устройства перестройки. Детектор - это поперечный фоторезистор. Взаимное расположение фильтра и детектора показано на рис. 5. Символом Э обозначены электроды фоторезистора.

Взаимодействие всех трех подблоков происходит следующим образом. Каждому символу $a_i \in A$ в устройстве перестройки сопоставляется напряжение u_i. Так же, как и в п.3, состояния элементов памяти ячеек-соседей образуют код опроса $a_{i_1} a_{i_2} \dots a_{i_p}$ $(a_{i_j} \in A)$. Этому коду в устройстве перестройки сопоставляется вектор напряжений $(u_{i_1}, u_{i_2}, \dots, u_{i_p})$. Каждому номеру блока алгоритма i в устройстве перестройки сопоставляется напряжение U_i. Номер блока алгоритма может поступать на каждую ячейку из некоторого внешнего для однородной структуры устройства настройки. С учетом номера блока устройство перестройки вырабатывает вектор напряжений $(U_i + u_{i_1}, U_i + u_{j_2}, \dots, U_i + u_{i_p})$. Компоненты этого вектора поступают на соответствующие пары электродов. Напряжения U_i и u_j и частоты Ω_i и ω_j подобраны так, что если на k-й паре электродов присутствует напряжение $U_i + u_j$, то сигналы прямоугольников, несущие частоты $\Omega_i + \omega_j$, k-го столбца сечения луча проходят через фильтр. Это значит, что если код опроса, представленный вектором напряжений, совпадает с кодом левой части некоторой микрокоманды Π_i', представленным строкой прямоугольников сечения луча, то под фильтром на поверхности фоторезистора образуется светящаяся полоса, соединяющая его электроды.

ЗАМЕЧАНИЕ. В том случае, если на практике возникнут разрывы в прохождении сигналов поля излучения в промежутках между электродами фильтра, структура луча может быть откорректирована наложением на стыки соседних прямоугольников новых прямоугольников,несущих сигнал с таким спектром, что этот сигнал проходит через фильтр независимо от потенциала на электродах.

Модуляция светящейся полосы по закону $M_{П_1}$ приводит к соответствующим изменениям потенциала в нагрузке фоторезистора. Эти изменения представляют код правой части микрокоманды и через коммутатор записываются в элементы памяти соседних ячеек. Такие черты ячейки, как наличие фильтра, управляемого по обратной связи, источник спектра, определяющий состояния ячейки, позволяют заключить, что она относится к классу частотно-гармонических элементов [14]. Принципиальным отличием использования источника спектра в рассматриваемом случае является то, что он несет алгоритм переработки информации однородной структурой. Важно также отметить "векторность" предложенной ячейки: она меняет не только свое состояние, но и состояния некоторых соседних ячеек, в свою очередь фильтр ячейки управляется сигналами не только от собственной памяти ячейки, но и от памяти некоторых соседних. Это делает ячейку типичным представителем класса ячеек однородных структур.

6. Вариант кодирования микропрограммы, снижающий требования к техническим параметрам ассоциативного блока ячейки. При частотном кодировании фильтр ячейки должен обладать способностью различать $(l \times k)$ частот. Это приводит к тому, что при увеличении числа символов в A (k) и числа блоков (l) число составляющих в спектре луча быстро растет, а значит, растут требования к качественным показателям блоков ячейки (особенно фильтра и фоторезистора). В связи с этим определенный интерес представляет вариант кодирования, в котором без увеличения числа параметров Ω, сопоставляемых блокам, в одну строку сечения луча проектируется в несколько раз большее число команд, чем число блоков. В этом варианте используется пространственное кодирование подблоков, на которые дополнительно разбиваются блоки алгоритма. Содержательно этот вариант не сложнее варианта из п. 4, но более громоздок в изложении, поэтому рассмотрение проведем на примере, из которого будут ясны принципы построения кодирования.

Пусть $A = \{0,1\}$ и будем считать, что любое двоичное слово длины десять может быть левой частью микрокоманды подстановки. Будем обозначать левую часть буквой W. Всего возможных вариантов будет 2^{10}. Разобьем это множество на четыре блока. Пусть $W = a_{i_1} \ldots a_{i_{10}}$, тогда в блок I зачислим все W, у которых $a_{i_1} a_{i_2} =$

$= 00$, в блок П – все W, у которых $a_{i_1} a_{i_2} = 0I$, в блок Ш – все W, у которых $a_{i_1} a_{i_2} = I0$, в блок IУ – все W, у которых $a_{i_1} a_{i_2} = II$. В свою очередь каждый блок разобьем на четыре подблока. В первый подблок включим W, у которых $a_{i_3} a_{i_4} = 00$, во второй – W, у которых $a_{i_3} a_{i_4} = 0I$, в третий – W, у которых $a_{i_3} a_{i_4} = I0$, в четвертый – W, у которых $a_{i_3} a_{i_4} = II$.

Сопоставим первому блоку параметр Ω_1, второму – Ω_2, третьему – Ω_3, четвертому – Ω_4. Первому подблоку первого блока поставим в соответствие набор параметров $\emptyset\emptyset\emptyset\emptyset\emptyset\,\Omega_1$, второму – $\emptyset\emptyset\emptyset\emptyset\Omega_1\emptyset$, третьему – $\emptyset\emptyset\Omega_1\,\Omega_1\emptyset\emptyset$, четвертому – $\Omega_1\Omega_1\,\emptyset\emptyset\emptyset\emptyset$. Наборы эти выбираются так, чтобы были они попарно-ортогональны. Аналогично делается сопоставление для подблоков i-го блока, только символ Ω_1 заменяется на Ω_i ($i = 2,3,4$). (Символ $\emptyset$ означает пустой параметр.)

В одну строку поля излучения попадает шестнадцать подстановок. Начальные отрезки слов, образующих левые части W этих подстановок, имеют вид:

0000...,	0I00...,	I000...,	II00...,
000I...,	0I0I...,	I00I...,	II0I...,
00I0...,	0II0...,	I0I0...,	III0...,
00II...,	0III...,	I0II...,	IIII...

Таких групп для данного примера 64. Вместо точек в каждой группе ставится шестиразрядный вектор из 0 и I, один и тот же для всей группы, так, чтобы по группам были разнесены все 2^{10} векторов. Поставим в соответствие символу 0 параметр ω_0, символу I – параметр ω_1. Будем обозначать символом ω_δ, где $\delta = 0, I$, любой из параметров ω_0, ω_1.

Строка поля излучения имеет шесть прямоугольных участков. В каждом из них поле излучения задается суперпозицией сигналов с частотами, указанными в табл.2. Параметр ω_δ в каждом столбце один и тот же, но в различных столбцах могут быть разными. Всего в поле излучения будет 64 строки, в каждой строке – 16 наборов. В каждой строке элемент вида $\Omega_i + \omega_i$ ($i = I,2,3,4$) и только он модулирован по закону, кодирующему правую часть соответствующей микрокоманды.

7. Использование многозначного структурного алфавита, пространственное и частотное кодирование информации позволяет построить однородные вычислительные устройства, структурная слож-

Т а б л и ц а 2

N	1	2	3	4	5	6
1	ω_δ	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_1+\omega_\delta$
2	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_1+\omega_\delta$	ω_δ
3	ω_δ	ω_δ	$\Omega_1+\omega_\delta$	$\Omega_1+\omega_\delta$	ω_δ	ω_δ
4	$\Omega_1+\omega_\delta$	$\Omega_1+\omega_\delta$	ω_δ	ω_δ	ω_δ	ω_δ
5	ω_δ	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_2+\omega_\delta$
6	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_2+\omega_\delta$	ω_δ
7	ω_δ	ω_δ	$\Omega_2+\omega_\delta$	$\Omega_2+\omega_\delta$	ω_δ	ω_δ
8	$\Omega_2+\omega_\delta$	$\Omega_2+\omega_\delta$	ω_δ	ω_δ	ω_δ	ω_δ
9	ω_δ	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_3+\omega_\delta$
10	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_3+\omega_\delta$	ω_δ
11	ω_δ	ω_δ	$\Omega_3+\omega_\delta$	$\Omega_3+\omega_\delta$	ω_δ	ω_δ
12	$\Omega_3+\omega_\delta$	$\Omega_3+\omega_\delta$	ω_δ	ω_δ	ω_δ	ω_δ
13	ω_δ	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_4+\omega_\delta$
14	ω_δ	ω_δ	ω_δ	ω_δ	$\Omega_4+\omega_\delta$	ω_δ
15	ω_δ	ω_δ	$\Omega_4+\omega_\delta$	$\Omega_4+\omega_\delta$	ω_δ	ω_δ
16	$\Omega_4+\omega_\delta$	$\Omega_4+\omega_\delta$	ω_δ	ω_δ	ω_δ	ω_δ

ность ячеек которых не зависит от числа микрокоманд в алгоритме. От этого числа зависят геометрические размеры некоторых блоков ячейки. Каждая ячейка структуры может осуществлять выбор и выполнение любой из сотен (в пределе тысяч) одновременно поданных на нее микрокоманд. Например, элемент, построенный для варианта кодирования из п.6, имеет фильтр с шестью парами электродов и числом состояний фильтра равным десяти. Этот элемент может осуществлять выбор и выполнение любой из 1024 микрокоманд.

Л и т е р а т у р а

1. Электроника № 22, 1976, с. 3-4.
2. Электроника № 4, 1977, с. 95.

3. Реализация многозначных структур автоматики, под ред. Ракова М.А. - Киев: Наукова думка, 1976. - 350 с.

4. КМЕТЬ А.Б. О логических сетях из универсальных элементов. - В кн.: Многозначные машины и системы. Киев, 1976, с.51-66.

5. О многозначных логических сетях /Бобров А.Е., Кметь А.Б., Костянко И.Ф., Раков М.А., Угоров Б.Н. Там же, с. 21-51.

6. ГОЛОВКИН Б.А. Методы и средства параллельной обработки информации. -В кн.: Итоги науки и техники. Серия: Теория вероятностей. Математическая статистика. Теоретическая кибернетика.М. 1979, с. 85-171.

7. КОРНЕВ Ю.Н., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. Алгоритмы обобщенных подстановок и их интерпретация сетями автоматов и однородными машинами. -Изв. АН СССР. Тех. кибернетика, 1971, №6, с.131-142.

8. КОРНЕВ Ю.Н., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. Вопросы построения алгоритмов обобщенных подстановок с выделенным контекстом. -В кн.: Вычислительные системы. Вып. 47, Новосибирск,1971, с.117-129.

9. YAMADA H., AMOROSO S., Tesselation Automata. - Information and Control. 1969, v.14, N 3, p. 299-317.

10. МАРКОВ А.А. Теория алгоритмов. Т.42. - М.: Изд. АН СССР, 1954. -375 с. (Труды мат. ин-та АН ССР).

11. А.с. № 436350 (СССР). Двоичный накопительный сумматор /Ю.Н.Корнев, С.В.Пискунов, С.Н.Сергеев. - Опубл. в Б.И.,1974,№26, с. 192.

12. ГУРЬЕВ А.Ю., МЕТРИК Л.М., ХАВКИН В.Е. Ассоциативные запоминающие устройства на полупроводниковых запоминающих элементах. -М.: ЦНИИЭлектроника, 1975. - 42 с. (Обзоры по электронной технике. Сер. 3 "Микроэлектроника", вып. 4 (322)).

13. А.с. № 664168(СССР). Вычислительная однородная структура /Ю.Н.Корнев, С.В.Пискунов. - Опубл. в Б.И.,1979, № 19, с. 184.

14. СИГОРСКИЙ В.П., СИТНИКОВ Л.С., УТЯКОВ Л.Л. Многоустойчивые элементы дискретной техники. -М.: Энергия, 1966. - 360 с.

Поступила в ред.-изд.отд.
16 июня 1981 года