

ПАРАЛЛЕЛЬНЫЕ МИКРОПРОГРАММНЫЕ СТРУКТУРЫ
ДЛЯ ОРТОГОНАЛЬНЫХ ПРЕОБРАЗОВАНИЙ

В.П.Маркова

В статье на основе методов параллельного микропрограммирования [1] показан первый этап синтеза структуры спецпроцессора для ортогональных преобразований. Согласно методике структурного синтеза [2], первый этап включает анализ исходного алгоритма и технических ограничений на проектируемое устройство и составление параллельной микропрограммы. В качестве исходного алгоритма используется основной расчетный алгоритм спектрального анализа – вычисление ортогонального преобразования (прямого или обратного) функции, заданной на интервале длины N , и некоторой пороговой функции. Рассмотрены варианты микропрограммного описания данного алгоритма с различной степенью распараллеливания и приведены оценки временной сложности.

I. Основные понятия и определения

Наиболее полное и строгое изложение алгоритмов вычисления ортогональных преобразований и теории параллельного микропрограммирования можно найти в работах [1-3]. Ниже приведем некоторые понятия и определения, которые используются в данной статье.

I. Вычисление ортогональных преобразований функций. Пусть $f(x)$ – произвольная функция (в общем случае комплексная), заданная на интервале длины N , $N = k^2$. Разложение функции $f(x)$ по функциям полного ортогонального базиса Q называется ортогональным преобразованием (спектром) и обозначается $f^*(w)$. В матричной форме вычисление спектра записывается следующим образом:

$$f^* = Qf, \quad (1)$$

где f, f^* - векторы-столбцы функции $f(x)$ и ее спектра $f^*(w)$, Q - матрица преобразования порядка $N \times N$. Сложность вычисления спектра по соотношению (1) составляет величину $t = O(N^2)$.

На практике спектры функций вычисляются по алгоритмам быстрых преобразований (бп) [3]. Суть этих алгоритмов состоит в факторизации матрицы преобразования, т.е. в представлении ее в виде произведения n более простых матриц, состоящих большей частью из нулей. Независимо от способа факторизации матрицы преобразования, сложность вычисления спектра по алгоритму быстрых преобразований не превышает величину $t = O(N/2 \log_2 N)$.

Т а б л и ц а

Способ вычисления спектра	Алгоритм вычисления спектра	К-во вычислителей	Структура системы	Оценки временной сложности
Последовательный	Быстрое преобразование	1	-	$t_{бп} = O(N/2 \log_2 N)$ [3]
	Матричное умножение	1		$t = O(N^2)$ [3]
Параллельный	Быстрое преобразование	N	Линейная	$t = O(t_{бп} + t_{об}(N-1))$ [7]
			n-мерный куб	$t = O(t_{бп} + t_{об} \log_2 N)$ [7]
			Двумерная решетка	$t = O(t_{бп} + t_{об}(\sqrt{N}-1))$ [7]
	Матричное умножение	N	Линейная	$t = O(2N)$ [5,7]
			Древовидная	$t = O(\log_2 N + N)$ [1]

Дальнейшее стремление увеличить быстродействие алгоритмов вычисления спектров привело, с одной стороны, к разработке новых Фурье-подобных преобразований [4], с другой стороны, к созданию параллельных алгоритмов вычисления спектров [1,5-7]. В таблице приведены оценки сложности вычислений ортогональных преобразований по алгоритмам быстрых преобразований и матричного умножения на системе из N элементарных вычислителей с разной структурой связей ($t_{об}$ - время обмена информацией между вычислителями).

2. Параллельные микропрограммы. Массив перерабатываемой информации в любой микропрограмме представляется в виде множества пар (a_i, m_i) , называемых клетками, где $i = 0, 1, \dots, p$, m_i - имя клетки из массива имен M , a_i - ее состояние из алфавита A . Множество клеток $W = \{(a_i, m_i)\}$, в котором никакие две клетки не имеют одинаковых имен, называется клеточным множеством.

Элементарным преобразованием клеточного множества является микрокоманда (подстановка), заданная оператором вида

$$e: \{(a_i, \varphi_i(m))\} \rightarrow \{(b_j, \varphi_j(m))\} \rightarrow \{(c_i, \varphi'_i(m))\}, \quad (2)$$

где $m \in M$, $a_i, b_j, c_i \in A$, $\varphi_i, \varphi_j, \varphi'_i$ - некоторые функции, определенные на множестве имен M . Микрокоманда состоит из двух частей: левой и правой. Первое множество левой части называется базовой частью микрокоманды, второе - контекстом. Применение микрокоманды к множеству W состоит в следующем. Для каждого m отыскивается множество клеток, определяемое левой частью микрокоманды. Если полученное множество для какого-либо m включено в W , то микрокоманда применима. Результатом применения микрокоманды к W является объединение множества клеток, соответствующих правой части микрокоманды, и множества W , из которого исключены клетки, определенные базовой частью микрокоманды. Если правая часть микрокоманды содержит символы переменных отношений и функций, принимающих значения из алфавита состояний A , то такая микрокоманда называется функциональной. Ее применение отличается от обычной (2) тем, что клетки базовой части заменяются на клетки, состояния которых вычисляются как функции состояний соседей. В работе мы будем пользоваться стационарными микрокомандами, т.е. микрокомандами, в которых для любых $m_i \in M$ $\varphi_i(m) = \varphi'_i(m)$. Тогда применение микрокоманды состоит в замене a_i на φ_i . Множество микрокоманд $\{e_i\}$, в котором никакие две микрокоманды не имеют одинаковых левых частей, называется параллельной микропрограммой. Выполнение микрокоманд микропрограммы производится одновременно над всеми клетками множества W . Микропрограмма применима к W , если к нему применима хотя бы одна микрокоманда. Если же ни одна микрокоманда к W неприменима, то W - результат применения микропрограммы.

Написание параллельной микропрограммы начинается с выбора клеточного множества, т.е. алфавита состояний и структуры имен.

Алфавит состояний A строится в процессе составления микропрограммы и состоит из двух частей: основной и вспомогательной. Основной алфавит представляет числовую информацию алгоритма, вспомогательный — управляющую. При написании микропрограммы к алфавиту могут быть добавлены символы переменных, функций, отношений и т.д. при условии, что эти символы принимают значения из алфавита A . Структура имен клеток M выбирается с учетом структуры обрабатываемых массивов информации и алгоритма их преобразования.

Прежде чем перейти к составлению параллельных микропрограмм исходного алгоритма, напомним его суть.

2. Основной расчетный алгоритм спектрального анализа (алгоритм A)

Пусть $f(X)$ — исходная функция. Суть алгоритма A состоит в вычислении спектра $f^*(W)$ и некоторой пороговой функции Λ по его коэффициентам. Для определенности считаем, что исходной функцией является функция k -значной логики от n переменных, определяемая отображением вида

$$f(X): X^n \rightarrow B^k,$$

где $B^k = \{0, 1, \dots, k-1\}$ — кольцо вычетов по $\text{mod } k$, $X^n = \{X_i\}$, $i = 0, 1, \dots, n-1$, — множество упорядоченных n -мерных наборов (векторов); $X_i = (\alpha_1 \alpha_2 \dots \alpha_m)$, $\alpha_m \in B^k$, $m = 1, 2, \dots, m$, представляет собой разложение числа i в k -значном коде. Множество векторов $\{X_i\}$ относительно сложения по $\text{mod } k$ образует абелеву группу.

В качестве ортогонального базиса будем использовать множество функций Виленькина-Крестенсона [3] и вычислять спектр функции (3) матричным умножением. Это означает, что коэффициенты спектра $f^*(W)$ определяются по соотношению

$$f^*(W_j) = \sum_{i=0}^{k^n-1} f(X_i) \varepsilon^{X_i(-W_j)} \quad \text{для всех } W_j \in W^n,$$

где $\varepsilon = \sqrt[k]{1}$, $f(X_i)$ — значение функции $f(X)$ на элементе X_i группы X^n , $W^n = \{W_j\}$, $j = 0, 1, \dots, n-1$, — область преобразования функции, тождественная X^n , $(-W_j)$ — дополнение элемента W_j , $X_i(-W_j)$ — скалярное произведение векторов с операцией сложения по $\text{mod } k$.

Пороговая функция Λ представляет собой вектор длины n , компоненты которого вычисляются по правилу

$$\lambda_j = \begin{cases} 1, & \text{если } f(W_j) = B, \\ 0 & \text{- в противном случае.} \end{cases} \quad (4)$$

3. Два варианта микропрограммного описания алгоритма А

Методы параллельного микропрограммирования позволяют представить исходный алгоритм в предельно-параллельном виде. Однако сохранить степень параллельности алгоритма в процессе проектирования устройства, как правило, не удается. Это объясняется тем, что при составлении микропрограммного описания алгоритма, которое в явной форме определяет структуру проектируемого устройства, приходится учитывать различного рода ограничения, которые либо возникают по ходу проектирования, либо заранее заданы. Например, ограничения на количество элементарных вычислителей, структуру связей между ними, форматы слов, объем подпрограмм и т.д.

Ниже приведены два варианта параллельной микропрограммы алгоритма А. Первый вариант микропрограммы составлен в предположении, что технические ограничения на проектируемое устройство отсутствуют. Второй вариант - написан с ограничением на количество элементарных вычислителей.

1. Первый вариант параллельной микропрограммы алгоритма А. Отсутствие технических ограничений на проектируемое устройство позволяет достичь предельной для исходного алгоритма параллельности. В этом случае микропрограммное описание алгоритма является первичным, оно определяет микроструктуру устройства и его основные параметры.

Каждому массиву информации алгоритма А (исходному, промежуточному и результирующему) ставится в соответствие клеточное множество матричной структуры. Исходные данные по массивам размещаются следующим образом (см.рис.1). Множество элементов $f(X)$ и вектор значений функции $f(X)$ - в клетках с именами из $M_2 = \{ \langle -1, i \rangle \}$, $i=0, 1, \dots, N-1$, множество элементов W^a - в клетках массива $M_1 = \{ \langle j, -1 \rangle \}$, $j=0, 1, \dots, N-1$, величина ϵ хранится в клетках с именами из $M_3 = \{ \langle 1, j, 1 \rangle \}$. Исходный алгоритм выполняется в два этапа: на первом этапе в массиве $M_2 = \{ \langle 1, j, 2 \rangle \}$ вычисляется спектр, на втором этапе в массиве $M_3 = \{ \langle N, j \rangle \}$ - пороговая функция.

Вычисление спектра $f^*(W)$ осуществляется следующим образом. Сначала в клетки $\{ \langle a, \langle 1, j, 2 \rangle \} \}$ по горизонтальным шинам расс-

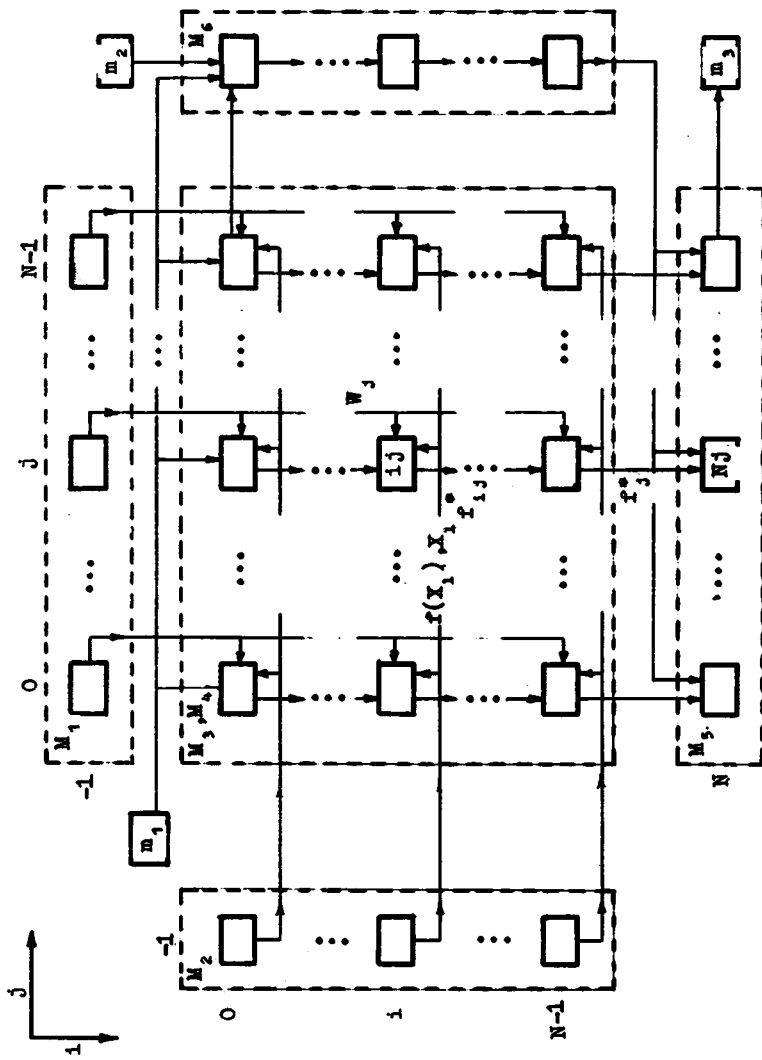


Рис.1. Структурная схема алгоритма А (вариант 1).

даются значения X_1 и $f(X_1)$ из массива M_2 , по вертикальным шинам - значения W_j из массива M_1 . Затем вычисляются произведения

$$f_{i,j}^* = f(X_1) \varepsilon^{X_1(-W_j)} \quad (5)$$

Момент окончания вычисления произведений (5) фиксируется по переходу клетки с именем $\langle 0, N-1, 2 \rangle$ из состояния α в состояние β и такое, что $\beta \in E^k$. (Действительно, для любого W_j произведение $X_0(-W_j) \equiv 1$, следовательно, $f_{0, N-1}^* = f(X_{0, N-1})$.) Далее определяются частичные суммы вида

$$F_{i,j}^* = f_{i,j}^* + F_{i-1,j}^* \quad (6)$$

где $F_{i-1,j}^* = \sum_{k=0}^{i-1} f_{k,j}^*$. Нахождение сумм (6) в микропрограмме связано с движением символа β в клетках массива $M_6 = \{ \langle i, N \rangle \}$ в сторону больших i . Появление символа β в клетке с именем $\langle N-1, N \rangle$ означает, что вычисление коэффициентов спектра закончено.

Определение пороговой функции Λ осуществляется в клетках $\{ \langle \alpha, \langle N, j \rangle \rangle \}$. Значения компонент λ_j вычисляются по соотношению (4). Состояние β клетки с именем $\langle N, N-1 \rangle$ такое, что $\beta \in \{0, 1\}$ служит признаком окончания вычисления функции Λ и переводит клетку с именем m_3 из нулевого состояния в единичное. Клетка $\langle 1, m_3 \rangle$ является сигналом завершения работы алгоритма. Оценка временной сложности алгоритма составляет величину $t = O(N)$.

Таким образом, полное множество имен $M = M_1 \cup M_2 \cup M_3 \cup M_4 \cup M_5 \cup M_6 \cup M_7$, где $M_7 = \{m_1, m_2, m_3\}$. Алфавит состояний клеток объединяет основной и вспомогательный. Основным алфавитом является декартово произведение $A_2 = \underbrace{E^k \times E^k \times \dots \times E^k}_{n \text{ раз}}$ (область значений эле-

ментов X_1 и W_j), множество комплексных чисел C заданной разрядности (область значения коэффициентов $f^*(W_j)$) и множество $D = \{0, 1\}$ (область значений компонент λ_j). Вспомогательным алфавитом служит множество $A_7 = \{1, 0, \beta\}$. Полным алфавитом состояний $A = A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup A_6 \cup A_7$, где $A_2 = A_1 \times C$, $A_3 = C$, $A_4 = CU \{a\}$, $A_5 = DU \{a\}$, $A_6 = \{a, \beta\}$. При написании микропрограммы конкретные символы из основного алфавита обозначаются начальными буквами латинского алфавита, переменные символы - последними буквами. Клетка $\langle 1, m_1 \rangle$ используется в качестве сигнала о начале работы алгоритма. Она вводится в контекст всех микропрограмм.

Микропрограмма алгоритма включает следующие микрокоманды.

1. Вычисление произведений (5):

$$e: \{(a, \langle i, j, 2 \rangle)\} \cdot \{(d, \langle i, j, 1 \rangle)\} \{(\langle x, x \rangle, \langle i, -1 \rangle)\} \{(\langle w, \langle -1, j \rangle)\} \{1, m_1\} \rightarrow \{(\langle w, \langle i, j, 2 \rangle)\},$$

где $w = fd^{xw}$.

2. Введение символа β в клетку с именем $\langle 0, N \rangle$:

$$e: \{(a, \langle 0, N \rangle)\} \cdot \{(w, \langle 0, N-1, 2 \rangle)\} \{1, m_1\} \{(\beta, m_2)\} \rightarrow \{(\beta, \langle 0, N \rangle)\},$$

где $w \in B^k$.

3. Вычисление частичных сумм (6):

$$e: \{(\langle w, \langle i, j, 2 \rangle)\} \cdot \{(t, \langle i-1, j, 2 \rangle)\} \{1, m_1\} \rightarrow \{(\langle u, \langle i, j, 2 \rangle)\},$$

где $u = w+t$.

4. Движение символа β вдоль строк массива M_2 :

$$e: \{(a, \langle i, N \rangle)\} \cdot \{(\beta, \langle i-1, N \rangle)\} \{1, m_1\} \rightarrow \{(\beta, \langle i, N \rangle)\}.$$

5. Вычисление компонент (4)

$$e: \{(a, \langle N, j \rangle)\} \cdot \{(y, \langle N-1, j, 2 \rangle)\} \{(\beta, \langle N-1, N \rangle)\} \{1, m_1\} \rightarrow \{(\langle b, \langle N, j \rangle)\},$$

где

$$b = \begin{cases} 1, & \text{если } y = \beta, \\ 0 - & \text{в противном случае.} \end{cases}$$

6. Формирование признака о завершении работы алгоритма:

$$e: \{1, m_1\} \{0, m_2\} \cdot \{(b, \langle N, N-1 \rangle)\} \rightarrow \{0, m_1\} \{1, m_2\},$$

где $b \in \{0, 1\}$.

2. Второй вариант параллельной микропрограммы алгоритма А.

Пусть задана однородная структура из M , $M \leq N^2$, элементарных вычислителей, расположенных в узлах двумерной решетки (рис.2), где $PL = M$. Требуется вложить в эту структуру микропрограммное описание алгоритма А.

Представим матрицу преобразования Q , которая строится в процессе вычисления спектра $f^*(w)$, в виде $n_x n_w$ матриц Q_0 порядка $P \times L$ (рис.3), где $n_x = \lfloor \frac{N}{P} \rfloor$, $n_w = \lfloor \frac{N}{L} \rfloor$. В соответствии с данным разбиением матрицы Q , примем следующий вариант организации вы-

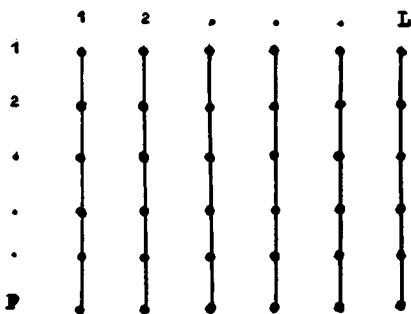


Рис.2

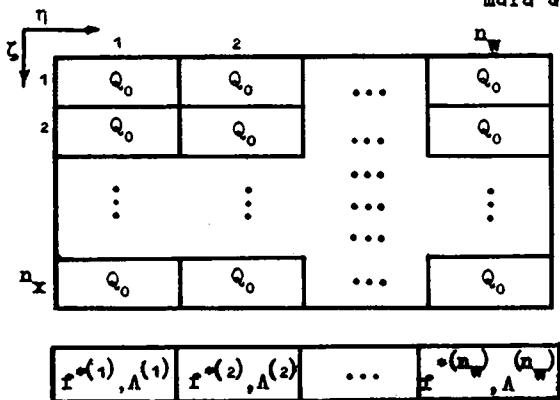


Рис.3

$f(z)$, $z = 1, 2, \dots, n_X$, каждое из которых записывается в соответствующий столбец из M_{10} . Элементы множества X^n вычисляются рекурсивно в процессе выполнения алгоритма в клетках с именами из $M_9 = \{ \langle i, -2, 1 \rangle \}$, $i = 0, 1, \dots, P-1$. (В исходном состоянии клетки с именами $\langle i, -2, 1 \rangle$ хранят элементы из $X^{(1)} = \{X_0, X_1, \dots, X_{P-1}\}$ и номер $z = 1$.) Элементы подмножества $W^{(\eta)}$ записываются в клетках массива $M_{12} = \{ \langle -1, j, 1 \rangle \}$, $j = 0, 1, \dots, L-1$, величина ϵ - в клетках массива $M_4 = \{ \langle i, j, 1 \rangle \}$.

В микропрограмме вычисление подмножества $f^{*(\eta)}$ организовано следующим образом. По сигналу, разрешающему выполнение η -го шага

числительного процесса алгоритма А. Спектр $f^{*(W)}$ формируется в виде n_W подмножеств $f^{*(\eta)}$ мощности L . Элементы подмножеств $f^{*(\eta)}$, $\eta = 1, 2, \dots, n_W$, вычисляются конвейерным способом. Для каждого подмножества $f^{*(\eta)}$ по соотношению (4) определяется функция $\Lambda^{(\eta)}$, и тогда пороговая функция $\Lambda = (\Lambda^{(1)} \Lambda^{(2)} \dots \Lambda^{(n_X)})$.

Опишем выполнение η -го шага алгоритма, т.е. вычисление подмножества коэффициентов $f^{*(\eta)}$ и пороговой функции $\Lambda^{(\eta)}$. Исходной информацией шага являются вектор значений функции $f(X)$, множество элементов X^n , подмножество элементов $W^{(\eta)}$ мощности L и величина ϵ . По массивам они размещаются следующим образом (см.рис.4). Множество значений функции $f(X)$ разбивается на подмножестве

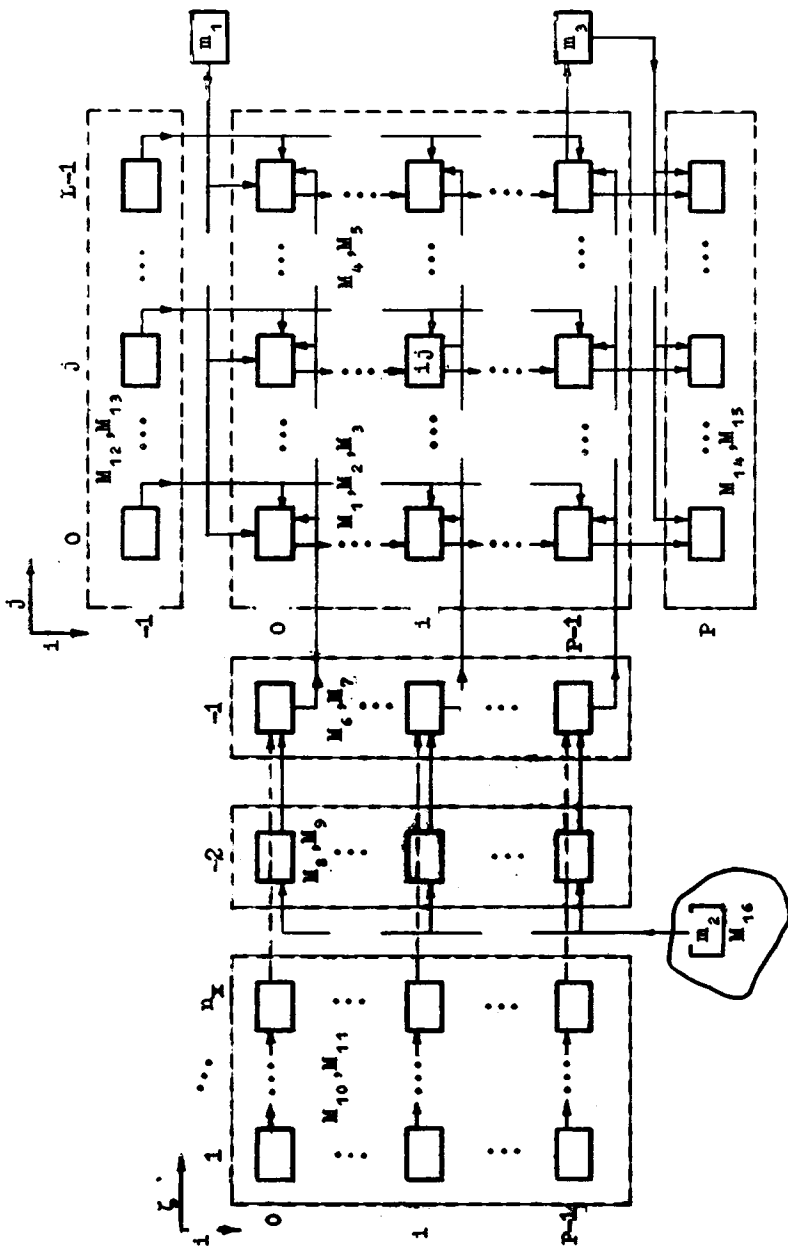


Рис.4. Структурная схема алгоритма А (вариант 2).

алгоритма, в массив $M_c = \{ \langle i, -1, 1 \rangle \}$ переписываются элементы подмножеств $X^{(1)}$ и $f^{(1)}$. Затем в клетки с именами из $M_3 = \{ \langle i, j, 3 \rangle \}$ рассылаются значения элементов $X_1^{(1)}$, $f^{(1)}(X_1)$, $w_j^{(\eta)}$ и вычисляются произведения

$$f_{ij}^{*(\eta_1)} = f^{(1)}(X_1) \in X_1^{(1)} (-w_j^{(\eta)}) \quad (7)$$

Одновременно с вычислением произведений (7) в массивах M_8 и $M_{10} = \{ \langle i, \zeta, 1 \rangle \}$ формируются подмножества $X^{(2)}$ и $f^{(2)}$. Элементы подмножества $X^{(2)}$ определяются по соотношению $X_1^{(2)} = (X_1^{(1)} \oplus \oplus X_P) \bmod N$. Элементы подмножества $f^{(2)}$ получают сдвигом содержимого клеток с именами из M_{10} в сторону меньших ζ .

Окончание вычисления произведений (7) фиксируется по переходу клетки с именем $\langle M_2(i, 0, 2) \rangle$ из нулевого состояния в единичное. Появление клетки $(1, \langle M_2(i, 0, 2) \rangle)$ служит сигналом начала вычисления частичных сумм (4) и записи значений $X_1^{(2)}$ и $f^{(2)}(X_1)$ в соответствующие строки массива M_c . После того, как произведения $f_{ij}^{*(\eta_1)}$ под управлением клетки $\langle \beta, m_1 \rangle$ попадут на $(i+1)$ -ю строку массива M_3 , в клетках с именами $\langle i, j, 3 \rangle$ вычисляются произведения $f_{ij}^{*(\eta_2)}$. Первая частичная сумма вида

$$f_j^{*(\eta_1)} = \sum_{i=0}^{P-1} f_{ij}^{*(\eta_1)}$$

появляется в клетках с именами $\langle P, j, 1 \rangle$ через P тактов. Клетка $\langle 1, m_3 \rangle$ является признаком окончания построения подмножества $f^{*(\eta)}$. Выполнение η -го шага алгоритма заканчивается вычислением пороговой функции $A^{(\eta)}$ в клетках с именами из $M_{15} = \{ \langle P, j, 2 \rangle \}$. Оценка временной сложности алгоритма составляет величину

$$t = O(n_w(P+n_X-1)) \quad (8)$$

при полной конвейеризации вычислительного процесса $t = O(P + (n_X-1)n_w)$.

Таким образом, полное множество имен $M = \bigcup_{i=1}^{17} M_i$, где $M_3 = \{ \langle i, j, 5 \rangle \}$, $M_7 = \{ \langle i, -1, 2 \rangle \}$, $M_9 = \{ \langle i, -2, 2 \rangle \}$, $M_{11} = \{ \langle i, \zeta, 2 \rangle \}$, $M_{13} = \{ \langle -1, j, 2 \rangle \}$, $M_{16} = \{ \langle P, j, 1 \rangle \}$, $M_{16} = \{ m_2 \}$,

$M_{17} = \{m_1, m_2\}$. Полный алфавит состояний $A = \bigcup_{i=1}^{17} A_i$. Здесь $A_1 = A_3 = A_{14} = \emptyset$, $A_2 = \{0, I, \beta\}$, $A_4 = A_5 = R$, $A_{10} = E^k$, $A_7 = A_9 = A_{11} = A_{13} = A_{17} = \{0, 1\}$, $A_{15} = \{a, 0, 1\}$, $A_{16} = \underbrace{E^k \times E^k \times \dots \times E^k}_{n \text{ раз}}$, $A_6 = A_{12} = A_{16} \times R$, где R - множество натуральных чисел.

Микропрограмма η -го шага алгоритма включает следующие микрокоманды. (Для простоты в контекстах микрокоманд опущена управляющая информация.)

1. Запись значений $x_1^{(\zeta)}$ и $f^{(\zeta)}(x_1)$ в клетки с именами из M_6 :

е: $\{((\langle 0, 0, 0 \rangle, \langle i, -1, 1 \rangle)) \{ (1, \langle i, -1, 2 \rangle) \} \{ (0, \langle i, -2, 2 \rangle) \} \{ (0, \langle m_{11}(i, 1, 2) \rangle) \} \rightarrow \{ (\langle x, \zeta \rangle, \langle i, -2, 1 \rangle) \} \{ (f, \langle m_{11}(i, 1, 1) \rangle) \} \rightarrow \{ (\langle x, f, \zeta \rangle, \langle i, -1, 1 \rangle) \} \{ (1, \langle i, -2, 2 \rangle) \} \{ (1, \langle m_{11}(i, 1, 2) \rangle) \}$.

2. Формирование значений $x_1^{(\zeta+1)}$ и $f^{(\zeta+1)}(x_1)$ в клетках с именами из M_7 и M_8 :

е: $\{ (\langle x, \zeta \rangle, \langle i, -2, 1 \rangle) \} \{ (1, \langle i, -2, 2 \rangle) \} \rightarrow \{ \bar{x}, m_2 \} \rightarrow \{ (\langle \bar{x}, \bar{\zeta} \rangle, \langle i, -2, 1 \rangle) \} \{ (0, \langle i, -2, 2 \rangle) \}$,

где

$$\begin{aligned} \bar{x} &= (x \oplus \bar{x}) \bmod N, \\ \bar{\zeta} &= (\zeta + 1) \bmod P; \end{aligned} \quad (9)$$

е: $\{ (s, \langle i, \zeta, 1 \rangle) \} \{ (1, \langle i, \zeta, 2 \rangle) \} \{ (f, \langle i, \zeta+1, 1 \rangle) \} \{ (0, \langle i, \zeta+1, 2 \rangle) \} \rightarrow \{ (f, \langle i, \zeta, 1 \rangle) \} \{ (0, \langle i, \zeta, 2 \rangle) \} \{ (0, \langle i, \zeta+1, 1 \rangle) \} \{ (1, \langle i, \zeta+1, 2 \rangle) \}$.

3. Вычисление произведений (5):

е: $\{ (0, \langle i, j, 2 \rangle) \} \{ (0, \langle i, j, 3 \rangle) \} \rightarrow \{ (w, \eta, \langle -1, j, 1 \rangle) \} \{ (0, \langle -1, j, 2 \rangle) \} \{ (\langle x, f, \zeta \rangle, \langle i, -1, 1 \rangle) \} \{ (0, \langle i, -1, 2 \rangle) \} \{ (a, \langle i, j, 1 \rangle) \} \{ (\zeta, \langle i, j, 4 \rangle) \} \{ (\eta, \langle i, j, 5 \rangle) \} \rightarrow \{ (1, \langle i, j, 2 \rangle) \} \{ (u, \langle i, j, 3 \rangle) \}$,

где $w = fd^{xw}$.

4. Формирование сигнала о разрешении записи значений $x_1^{(\zeta+1)}$ и $f^{(\zeta+1)}(x_1)$ в клетки с именами из M_8 :

е: $\{ (0, \langle i, -1, 2 \rangle) \} \rightarrow \{ (1, \langle i, 0, 2 \rangle) \} \{ (\zeta, \langle i, 0, 4 \rangle) \} \rightarrow \{ (c, \langle i, -1, 2 \rangle) \}$,

где

$$c = \begin{cases} 1, & \text{если } \zeta < n_x, \\ 0 & \text{в противном случае.} \end{cases}$$

5. Введение признака β в клетки с именами $\langle 0, j, 2 \rangle$:

$$e: \{(1, \langle i, j, 2 \rangle)\} \rightarrow \{\beta, m_1\} \rightarrow \{(\beta, \langle i, j, 2 \rangle)\}.$$

6. Вычисление частичных сумм (6):

$$e: \{(1, \langle i, j, 2 \rangle)\} \{(\bar{u}, \langle i, j, 3 \rangle)\} \{(\zeta, \langle i-1, j, 4 \rangle)\} \{(\eta, \langle i-1, j, 5 \rangle)\} \{(\beta, \langle i-1, j, 2 \rangle)\} \{(\bar{t}, \langle i-1, j, 2 \rangle)\} \rightarrow \{(\zeta, \langle i, j, 4 \rangle)\} \{(\eta, \langle i, j, 5 \rangle)\} \rightarrow \{(\beta, \langle i, j, 2 \rangle)\} \{(\bar{y}, \langle i, j, 3 \rangle)\} \{(\bar{o}, \langle i-1, j, 2 \rangle)\} \{(\bar{\zeta}, \langle i-1, j, 4 \rangle)\} \{(\bar{\eta}, \langle i-1, j, 5 \rangle)\},$$

где $y = t + u$, $\bar{\zeta}$ вычисляется по соотношению (9),

$$\bar{\eta} = \begin{cases} \eta, & \text{если } \zeta < n_x, \\ \eta + 1 & \text{в противном случае.} \end{cases} \quad (10)$$

7. Запись частичных сумм в клетки с именами из $M_{1,4}$:

$$e: \{(0, \langle P, j, 1 \rangle)\} \{(\beta, \langle P-1, j, 2 \rangle)\} \{(\bar{y}, \langle P-1, j, 3 \rangle)\} \{(\zeta, \langle P-1, j, 4 \rangle)\} \{(\eta, \langle P-1, j, 5 \rangle)\} \rightarrow \{(\bar{y}, \langle P, j, 1 \rangle)\} \{(\bar{o}, \langle P-1, j, 2 \rangle)\} \{(\bar{o}, \langle P-1, j, 3 \rangle)\} \{(\bar{\zeta}, \langle P-1, j, 4 \rangle)\} \{(\bar{\eta}, \langle P-1, j, 5 \rangle)\},$$

где значения $\bar{\zeta}$ и $\bar{\eta}$ вычисляются по соотношениям (9) и (10).

8. Формирование признака α завершения вычисления подмножества $r^*(\eta)$:

$$e: \{(0, m_1)\} \rightarrow \{(\zeta, \langle P-1, L-1, 4 \rangle)\} \rightarrow \{\alpha, m_1\},$$

где

$$\alpha = \begin{cases} 1, & \text{если } \zeta = n_x, \\ 0 & \text{в противном случае.} \end{cases}$$

9. Вычисление компонент вектора:

$$e: \{(a, \langle P, j, 2 \rangle)\} \rightarrow \{(\bar{y}, \langle P, j, 1 \rangle)\} \{1, m_1\} \rightarrow \{(b, \langle P, j, 2 \rangle)\},$$

где

$$b = \begin{cases} 1, & \text{если } y = \bar{v}, \\ 0 & \text{в противном случае.} \end{cases}$$

На примере одного из алгоритмов спектрального анализа (алгоритма А) показано, что микропрограммный способ синтеза однородных структур позволяет представить исходный алгоритм с различной степенью параллельности. Придавая конкретные значения параметрам структуры (P и L) в соотношении (9), можно получить известные оценки временной сложности выполнения алгоритма. Действительно, при $P = L = 1$ (вырожденный случай однородной системы) оценка слож-

ности составляет величину $t = O(N)^2$. При $P = 1, L = N$ (линейная структура) - $t = O(2N)$. При $P = L = N$ (первый вариант микропрограммного описания алгоритма) - величина $t = O(N)$.

Л и т е р а т у р а

1. Методы параллельного микропрограммирования /Под ред. Бандман О.Л. - Новосибирск: Наука, 1981. - 180 с.

2. Применение методов параллельного микропрограммирования для синтеза структуры специализированных вычислителей /Бандман О.Л., Пискунов С.В., Сергеев С.Н. - Новосибирск, 1983. - 32 с. (Препринт/Институт математики СО АН СССР, ОВС-18).

3. ТРАХТМАН А.М., ТРАХТМАН В.А. Основы теории дискретных сигналов на конечных интервалах. - М.: Сов. радио, 1975. - 208 с.

4. ЛАБУНЦ В.Г. Фурье-подобные преобразования. - В кн.: Применение ортогональных методов при обработке сигналов и анализе систем. Межвузовск. сборник. Свердловск, 1981, с. 4-14.

5. МИШИН А.И., СЕДУХИН С.Г. Однородные вычислительные системы и параллельные вычисления. - АИГ, 1981, с. 20-24.

6. СЕРГЕЕВ С.Н. Структурное проектирование специализированных параллельных микропрограммных устройств. - В кн.: Однородные вычислительные системы (Вычислительные системы, вып. 90). Новосибирск, 1982, с. 64-77.

7. СЕРГЕЕВ С.Н. Синтез микропрограммных автоматов: Автореф. дис. на соиск. учен. степени канд. техн. наук. - Новосибирск, 1979.

Поступила в ред.-изд. отд.

8 октября 1984 года