АЛГОРИТМИЧЕСКИЙ АНАЛИЗ СТРУКТУРНОЙ ИНФОРМАЦИИ
(Вычислительные системы)
1985 год Выпуск II2

УДК 681.322.01

ПОСТРОЕНИЕ СЕТЕЙ АВТОМАТОВ, ИНТЕРПРЕТИРУЮЩИХ ПАРАЛЛЕЛЬНЫЕ МИКРОПРОГРАММЫ

С.В. Пискунов

Введение

В работе [1] предложена методика перехода от параллельного алгоритма решения заданного класса задач к структурной схеме устройства, реализующего этот алгоритм. Методика основана на формализованном переходе от параллельной микропрограммы, описывающей алгоритм, к интерпретирующей ее сети автоматов. Известны две процедуры такого перехода: общая [2], пригодная для построения как однородных, так и неоднородных сетей, и частная [3], ориентированная на построение однородных сетей. Предлагаемая в данной работе процедура построения сети автоматов, интерпретирующей параллельную микропрограмму, является процедурой общего типа основана на процедурах из [2,3], но в отличие от них позволяет строить более простые сети, т.е. сети, в которых каждый автомат в среднем имеет меньшее число входов-выходов. Целесообразность разработки процедуры вытекает из ориентации на реализацию проектируемых устройств в виде БИС (или СБИС), для которых одной из ключевых проблем является проблема межсоединений.

Приведем кратко перечень понятий, связанных с параллельными микропрограммами и их интерпретацией сетями автоматов, подробно эти понятия изложены в [3,4].

Параллельная микропрограмма Φ - это совокупность параллельных микрокоманд $\theta_1, \theta_2, \ldots, \theta_\nu$. Каждая микрокоманда θ представима в виде

$$\{(a_1, \varphi_1(m)) \ldots (a_l, \varphi_l(m))\} * \{(b_1, \varphi_1(m)) \ldots (b_s, \varphi_s(m))\} \rightarrow$$
$$\rightarrow \{(c_1, \varphi_1(m)) \ldots (c_l, \varphi_l(m))\},$$

где a_i $(1 \le i \le l)$, b_j $(1 \le j \le s)$, c_k $(1 \le k \le l)$ - символы из конечного алфавита A, а $\varphi_i(m)$ $(1 \le i \le l)$, $\psi_j(m)$ $(1 \le j \le s)$ - именующие функции, определенные на множестве имен M в общем случае бесконечном. Микропрограмма Φ преобразует состояния клеток из конечного множества клеток W, равного $\{(a_i, m_i)\}$, где a_i - состояние клетки $(a_i \in A)$, m_i - имя клетки $(m_i \in M)$. Если в записи любого W, преобразуемого микропрограммой Φ, использованы имена клеток из некоторого конечного множества $M' = \{m_1, m_2, \ldots, m_\mu\}$, являющегося подмножеством M, то для микропрограммы Φ может быть построена интерпретирующая ее сеть автоматов. В такой сети каждый автомат имеет имя из M', а множество внутренних состояний содержит алфавит A.

При построении сети автоматов мы будем широко использовать понятие микрооперации, поэтому определим его детально. Возьмем некоторое конкретное имя m_0 из M и, подставив его в именующие функции микрокоманды θ, получим выражение

$$\{(a_1, \varphi_1(m_0)) \ldots (a_l, \varphi_l(m_0))\} * \{(b_1, \psi_1(m_0)) \ldots (b_s, \psi_s(m_0))\} \to$$
$$\to \{(c_1, \varphi_1(m_0)) \ldots (c_l, \varphi_l(m_0))\}.$$

В том случае, когда

$$\varphi_1(m_0) = m_1^1 \ (m_1^1 \in M), \ldots, \varphi_l(m_0) = m_l^1 \ (m_l^1 \in M),$$

$$\psi_1(m_0) = m_1^2 \ (m_1^2 \in M), \ldots, \psi_s(m_0) = m_s^2 \ (m_s^2 \in M),$$

можно записать выражение

$$\{(a_1, m_1^1) \ldots (a_l, m_l^1)\} * \{(b_1, m_1^2) \ldots (b_s, m_s^2)\} \to \{(c_1, m_1^1) \ldots (c_l, m_l^1)\},$$

которое и называется микрооперацией. Другими словами, микрооперация (обозначается P) - это частный вариант микрокоманды, у которой все именующие функции - константы.

Прежде чем перейти к описанию предлагаемой процедуры, кратко изложим известную [2]. Она основана на получении совокупности всех микроопераций. Микрооперации строятся для всех микрокоманд исходной Φ и всех имен M'. Совокупность всех микроопераций обозначается Q. Для каждого автомата с именем $m_i \in M'$ строится сокращенная каноническая таблица (СК-таблица). Но прежде строится таблица 2 [2, с. 137]. Она состоит из двух частей: левой и правой. Левая часть содержит μ столбцов (по числу символов в M'), j-й столбец имеет имя m_j, правая - один столбец с именем m_i. Из мно-

Таблица 2 автомата p_3

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_3
1	1	0	0	*	*	*	1
*	*	1	*	1	0	*	0

СК-таблица автомата p_3

p_1	p_2	p_3	p_4	p_5	p_6	p_3
1	1	0	0	*	*	1
*	*	1	*	1	0	0

Рис. I

жества Q извлекаются все микрооперации $\{P_1,P_2,\ldots,P_z\}$ такие, что в каждой P_j:

$$\{(a_1^j,m_1^{1j})\ldots(a_{l_j}^j,m_{l_j}^{1j})\}*\{(b_1^j,m_1^{2j})\ldots(b_{s_j},m_{s_j}^{2j})\}\to$$

$$\to\{(c_1^j,m_1^{1j})\ldots(c_{l_j}^j,m_{l_j}^{1j})\}$$

есть пара (a_x^j,m_x^{1j}) такая, что $m_x^{1j}=m_i$. Эти микрооперации берутся по очереди. Пусть взята P_j, в таблицу 2 вписывается j-я строка: в столбец левой части, имя которого совпадает с именем m_y^{1j}, вписывается a_y^j $(y=1,\ldots,l_j)$, в столбец левой части, имя которого совпадает с именем m_p^{2j}, вписывается b_p^j $(p=1,\ldots,s_j)$, в столбец с именем m_i из правой части таблицы вписывается символ c_x^j из пары $(c_x^j\ m_x^{1j})$ в правой части микрооперации P_j. Остальные столбцы строки заполняются символом * , обозначающим любой символ из алфавита A. После заполнения из таблицы 2 вычеркиваются все столбцы, содержащие только символ *. Полученная таблица и является СК-таблицей автомата с именем m_i. При этом считается, что в столбцы с именами $m_1,m_2,\ldots,m_{i-1}$, $m_{i+1},\ldots,m_\mu$ записаны состояния входов с соответствующими именами автомата в момент времени t, в столбец с именем m_i – внутреннее состояние автомата в момент t, в столбец с именем m_i из правой части таблицы – внутреннее состояние автомата в момент $t+1$,кото-

p_1	p_2	p_3	p_4	p_1
1	1	0	0	0

p_1	p_2	p_3	p_4	p_2
1	1	0	0	0

p_3	p_5	p_6	p_5
1	1	0	0

p_3	p_5	p_6	p_7	p_6
1	1	0	*	1
*	*	1	0	0

p_6	p_7	p_7
1	0	1

Рис. 2

рое совпадает со значением выхода автомата. Сеть строится так, что m_j-й вход автомата m_i соединен с выходом m_j-го автомата.

ПРИМЕР I. Даны $M' = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$ и Q, содержащая три микрооперации:

$$\{(1,p_1)(1,p_2)(0,p_3)\} * \{(0,p_4)\} \to \{(0,p_1)(0,p_2)(1,p_3)\};$$
$$\{(1,p_3)(1,p_5)(0,p_6)\} * \to \{(0,p_3)(0,p_5)(1,p_6)\};$$
$$\{(1,p_6)(0,p_7)\} * \to \{(0,p_6)(1,p_7)\}.$$

Построить сеть автоматов, интерпретирующую Q.

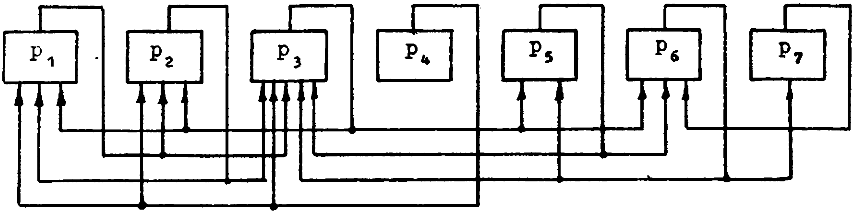

Рис. 3

На рис.I приведен пример построения таблицы 2 и СК-таблицы для автомата с именем p_3, СК-таблицы остальных автоматов приведены на рис.2. Сеть автоматов изображена на рис.3.

Новая процедура построения сети автоматов

К недостаткам процедур из [2,3] следует отнести дублирование выполнения микроопераций автоматами сети. Например, в примере I первую микрооперацию выполняют сразу три автомата p_1, p_2, p_3. В основу новой процедуры положено разделение всех автоматов сети на две группы: активные и пассивные. Активные автоматы выполняют некоторые подмножества микрокоманд, желательно непересекающиеся. Пассивные автоматы имеют простые таблицы переходов и выходов и с содержательной точки зрения осуществляют только функции хранения информации.

Общая блок-схема процедуры изображена на рис.4 и содержит четыре этапа.

Этап I. Построение множества микроопераций Q по исходным Φ и M'. Выполняется так, как описано во введении.

Этап 2. Заполнение таблиц I для автоматов с именами из M'. Макет таблицы I изображен на рис.5. Он состоит из двух частей: ле-

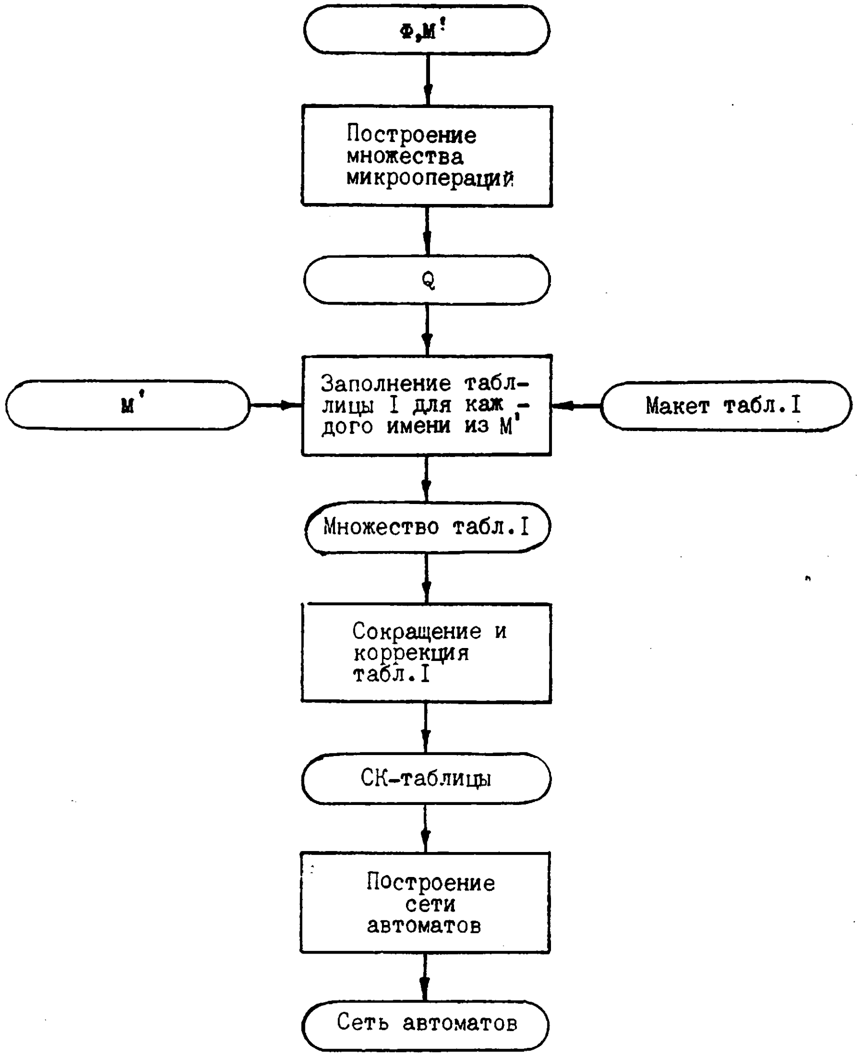

Рис.4. Блок-схема процедуры построения сети автоматов.

1	2	...	μ	1	2	...	μ	1	2	...	μ	
m_1	m_2	...	m_μ	m_1	m_2	...	m_μ	m_1	m_2	...	m_μ	

Рис. 5

вой и правой, разделенных двойной вертикальной чертой. В левой части – 2μ столбцов, в правой части – μ столбцов. Имена столбцов в левой части таблицы – это имена входов автомата, имена столбцов в правой части – имена выходов автомата. Имена первых μ столбцов в левой части таблицы образуют первую группу входов (аналогичную входам M_1^{in} автомата в процедуре [3]), имена последующих μ столбцов образуют вторую группу входов (аналогичную входам M_2^{in} автомата в процедуре [3]). Группы входов разделены в макете волнистой вертикальной чертой. Кроме того, к правой части макета добавляется еще один столбец (помечен косыми крестами), играющий служебную роль в процедуре построения сети.

Заполнение всех таблиц I ведется одновременно на каждом шаге реализации этапа. Исходными данными для очередного (z-го) шага являются: Q_z – совокупность микроопераций и M_z' – множество имен автоматов. На первом шаге $Q_1 = Q$, $M_1' = M'$.

Пусть $(z-1)$-й шаг заполнения таблиц I сделан, сделаем z-й шаг. Возьмем некоторое имя m', принадлежащее M_z'. Выберем из Q_z микрооперации (их список U_z запишем $P_1, P_2, \dots, P_{q_z}$), такие что в правой части любой P_j:

$$\{(a_1^j, m_1^{1j}) \dots (a_{l_j}, m_{l_j}^{1j})\} \bullet \{(b_1^j, m_1^{2j}) \dots (b_{s_j}, m_{s_j}^{2j})\} \to$$

$$\to \{(c_1^j, m_1^{1j}) \dots (c_{l_j}, m_{l_j}^{1j})\}$$

есть пара (c_i^j, m_i^{1j}) такая, что $m_i^{1j} = m'$. Критерии выбора m' могут быть самыми разными. Например, среди всех имен из M_z' можно выбрать имя, для которого список микроопераций имеет самую большую мощность (если таких имен несколько, можно взять любое из них). Имя записывается в список активных автоматов.

Для каждой микрооперации из U_z в таблицу I автомата с именем m' вписывается строка. Выполним, например, эту запись для микрооперации P_j. В строке первые μ столбцов заполняются точно так же,как строка для такой же микрооперации в левой части таблицы 2. В следующие μ столбцов вписывается символ λ (новый символ, обозначающий состояние отключения). При заполнении столбцов строки правой части таблицы I в столбец, имя которого совпадает с m_1^{1j}, вписывается символ c_1^j и т.д., в столбец, имя которого совпадает с именем $m_{l_j}^{1j}$, - символ $c_{l_j}^j$; в незаполненные столбцы вписывается символ λ . В служебный столбец вписывается символ "+".

Кроме того, для каждой микрооперации из U_z составляется список имен клеток (из списка исключается имя m'), входящих в конфигурацию правой части микрооперации. Для P_j этот список имеет вид $m_1^{1j},\ldots,m_{i-1}^{1j},m_{i+1}^{1j},\ldots,m_{l_j}^{1j}$. Для каждого имени из списка в таблицу автомата с таким же именем вписывается строка. Например, для некоторого имени m_f^{1j} из списка для P_j в таблицу I автомата с именем m_f^{1j} вносится строка, в первые μ столбцов которой вписан символ $*$; во второй группе входов в столбец с именем m_i^{1j}, равным m' , вписан символ c_i^{1j}, в остальные столбцы - символ λ . Для строки правой части таблицы 2 в столбец с именем m_f^{1j} вписывается символ c_f^{1j}, в остальные столбцы вписывается символ λ .В служебный столбец вписывается символ "-". По исчерпании списков имен для всех микроопераций из U_z выполняются операции $Q_{z+1}=Q_z\setminus U_z$ и $M'_{z+1}=M'_z\setminus m'$, и осуществляется переход к следующему шагу. Второй этап завершается, когда множество Q_z делается пустым. Имена, не попавшие в список имен активных автоматов, вносятся в список имен пассивных автоматов.

Этап 3. Сокращение и коррекция таблиц I. Сокращение таблиц I состоит в вычеркивании всех столбцов, содержащих только либо символ λ , либо символ $*$. Из одинаковых строк в таблице I оставляется только одна. Коррекция сокращенных таблиц I, содержащих в служебном столбце только символ "+", не производится. Коррекция сокращенных таблиц I, содержащих в служебном столбце символ "-", производится следующим образом. Вводится покомпонентная операция $\bigcirc$: $*\bigcirc a=a\bigcirc *=a$, $*\bigcirc *=*$, $\lambda\bigcirc\lambda=\lambda$, $a\bigcirc\lambda=\lambda\bigcirc a=a$, $a\bigcirc a=a$,где $a\in A$. Для строк, содержащих в одноименных столбцах различные символы из алфавита A , операция $\bigcirc$ не определена. Операция $\bigcirc$ при помощи скобок обобщается на две, три и т.д. строки. К двум, трем и т.д. строкам (вплоть до числа строк, совпадающего с числом строк в табл. I) применяется операция $\circ$. Все различные строки, полученные в результате применения операции $\circ$, присоединяются к таблице I.

Коррекция сокращенных таблиц I, содержащих в служебном столбце как символ "+", так и символ "-", осуществляется в два приема. Сперва для строк, помеченных служебным символом "-", осуществляется коррекция так, как описано выше. Присоединенные строки помечаются символом "-". Затем осуществляется коррекция, использующая строки, помеченные как символом "+", так и символом "-". Для любой таблицы I коррекция состоит в следующем. Будем обозначать строки, помеченные минусом, символом α с индексом. Покажем, как осуществляется коррекция для любой из них, например строки α_i: μ экземпляров строки записываются в виде таблицы 3, определяемой макетом таблицы I. В таблице 3 первые μ столбцов образуют матрицу T размерности $\mu \times \mu$, заполненную символом $*$. Вводятся списки $\Omega_1, \ldots, \Omega_\mu$, в исходном состоянии пустые. Берем по очереди строки, помеченные символом "+" (будем обозначать их символом β с индексом). Пусть мы уже взяли строки $\beta_1, \beta_2, \ldots, \beta_{j-1}$, возьмем теперь строку β_j. Если результат операции $\alpha_i \circ \beta_j$ определен, полученную строку вписываем в таблицу I, а в каждый Ω_t $(t=1,2,\ldots,\mu)$ вносим символ, записанный в t-й столбец строки β_j. Если результат операции не определен, берем следующую строку, помеченную символом "+", и т.д., пока не исчерпываются все строки. Когда они кончатся, в диагональ матрицы T вписываются элементы $*\backslash\Omega, *\backslash\Omega, \ldots, *\backslash\Omega_\mu$. Запись $*\backslash\Omega_i$ означает любой символ алфавита A, кроме символов входящих в список Ω_i. Строки (пусть это будут строки с номерами i_1, $i_2, \ldots, i_p$), в которых списки $\Omega_{i_1}, \Omega_{i_2}, \ldots, \Omega_{i_p}$ или содержит символ $*$, или содержит все символы из алфавита A, вычеркиваются из таблицы 3. Затем таблица 3 присоединяется к таблице I. В объединенной таблице из одинаковых строк оставляется только одна. Этап коррекции завершается, когда откорректированы все таблицы I.

Этап 4. Построение сети автоматов. Мы получили СК-таблицы автоматов. Для любого автомата m_i все его входы заданы в момент времени t. Выходы автомата, кроме выхода с именем m_i, также заданы в момент времени t (будем называть их выходами типа I). Выход m_i, совпадающий с внутренним состоянием автомата, задан в момент времени $t+1$ (будем называть его выходом типа 2). Автоматы с выходами такого типа являются С-автоматами [5], обладающими чертами как автомата Мура,так и автомата Мили одновременно. Отметим, что вход с именем m_i в первой группе входов автомата m_i тождествен внутреннему состоянию этого автомата в момент времени t.

Сеть строится так, что m_j-й вход первой группы входов автомата m_i соединен с выходом типа 2 автомата m_j (такая связь на рисунках проводится сплошной линией), а m_k-й вход второй группы входов автомата m_i соединен с m_j-м выходом типа I автомата с именем m_k (эта связь проводится штриховой линией).

Описание процедуры закончено.

ПРИМЕР 2. При помощи новой процедуры построить сеть автоматов для совокупности микроопераций из примера I.

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	
1	1	0	0	*	*	*	λ	λ	λ	λ	λ	λ	λ	0	0	1	λ	λ	λ	λ	+
*	*	1	*	1	0	*	λ	λ	λ	λ	λ	λ	λ	λ	λ	0	λ	0	1	λ	+

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	
*	*	*	*	*	*	*	λ	λ	0	λ	λ	λ	λ	0	λ	λ	λ	λ	λ	λ	−

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_1	p_2	p_3	p_4	p_5	p_6	p_7	
*	*	*	*	*	*	*	λ	λ	1	λ	λ	λ	λ	λ	λ	λ	λ	λ	1	λ	−
*	*	*	*	*	1	0	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ	0	1	+

Рис. 6

p_1	p_2	p_3	p_4	p_5	p_6	p_1	p_2	p_3	p_5	p_6
1	1	0	0	*	*	0	0	1	λ	λ
*	*	1	*	1	0	λ	λ	0	0	1

p_3	p_1
0	0

p_3	p_2
0	0

p_3	p_5
0	0

p_6	p_7	p_3	p_6	p_7
1	0	λ	0	1
*	*	1	1	λ

p_6	p_7
1	1

Рис. 7

На рис.6 приведены таблицы I для автоматов с именами p_3, p_1, p_6 (таблицы I для автоматов с именами p_2, p_5, p_7 строятся аналогично таблице I для автомата p_1). СК-таблицы всех автоматов приведены на рис.7. Сеть автоматов изображена на рис.8.

Сопоставим сети, полученные в примерах I,2. Если сложность каждого автомата оценивать произведением числа строк на число столбцов в его СК-таблице, то в среднем сложность каждого автомата, полученного и старой и новой процедурой, примерно одинакова. Но сопоставление

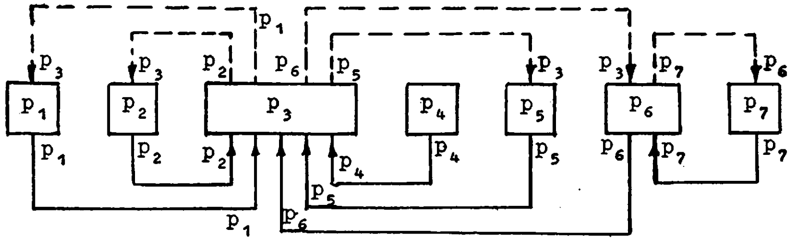

Рис. 8

рис.3,8 позволяет заключить, что сеть автоматов, рассматриваемая как граф, получается проще при применении новой процедуры.

Очевидно, что тенденция упрощения сети сохранится всегда и причины этому следующие.

Если автомат активный, то суммарное число входов-выходов определяется числом различных клеток во всех микрооперациях, в которые входит та клетка, которой сопоставлен данный активный автомат, и совпадает как для новой, так и для старой процедур.

Для пассивных автоматов, как правило, это не так. Действительно, пусть клетка, которой сопоставлен пассивный автомат, входит в несколько множеств (пусть их число k, $k \geq 1$) микроопераций для нескольких активных автоматов. Для такой клетки соответствующий автомат всегда имеет $2k$ входов-выходов. Для процедуры [2] автомат имеет такое число входов-выходов только в том случае, когда клетка входит в множество микроопераций, каждая из которых содержит лишь две клетки: ее саму и клетку, которой в новой процедуре сопоставляется активный автомат; в любом другом случае процедура [2] дает число входов-выходов больше чем $2k$. Образами пассивных автоматов служат автоматы p_1, p_2, p_5, p_7 в примере 2.

Дополнительной иллюстрацией сказанному является нижеследующий

ПРИМЕР 3. Даны $A = \{a_0, a_1\}$, $M' = \{1, 2, \ldots, n\}$, где n - некоторое фиксированное число, и Φ, содержащая одну микрокоманду

$$\{(a_1, i-1)(a_0, i)(a_1, i+1)\} * \{(a_1, i+2)\} \to \{(a_0, i-1)(a_1, i)(a_0, i+1)\}.$$

Надо построить сеть автоматов. Множество микроопераций имеет вид:

$$\{(a_1, \underline{\underline{1}})(a_0, 2)(a_1, 3)\} * \{(a_1, 4)\} \to \{(a_0, 1)(a_1, 2)(a_0, 3)\};$$

$$\{(a_1, 2)(a_0, 3)(a_1, \underline{\underline{4}})\} * \{(a_1, 5)\} \to \{(a_0, 2)(a_1, 3)(a_0, 4)\};$$

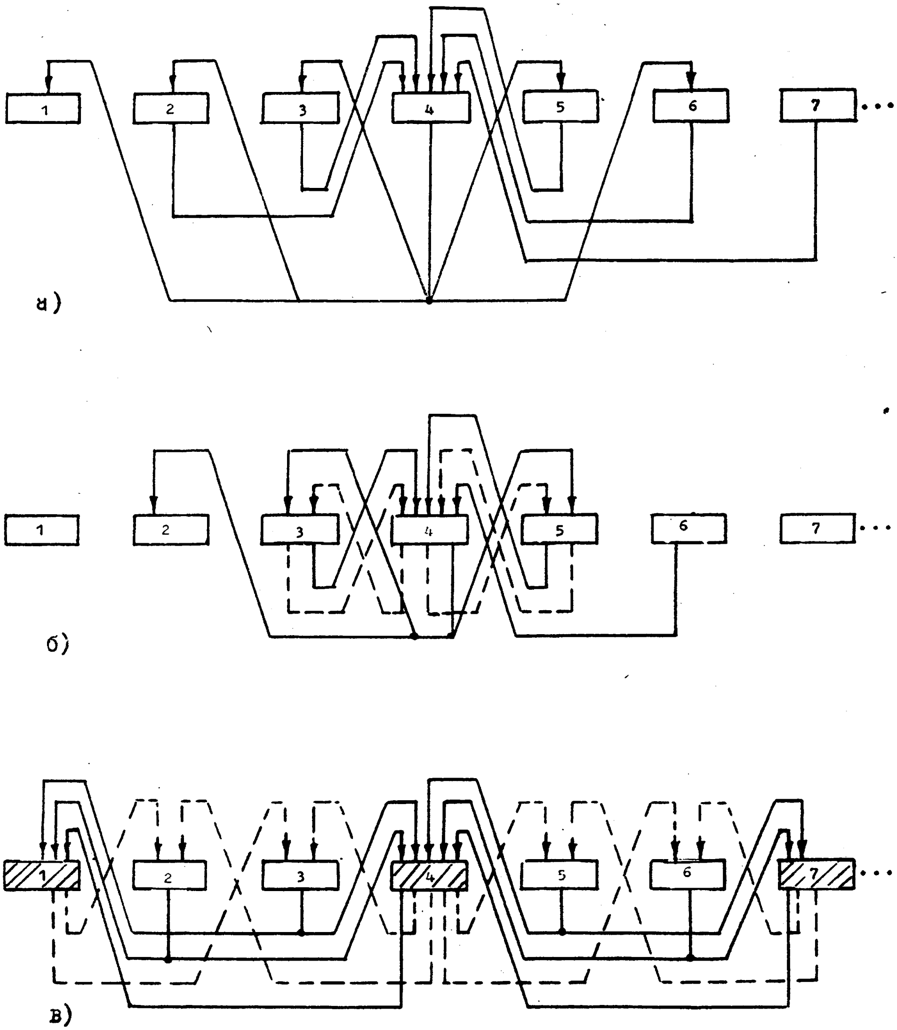

Рис.9. Фрагменты сетей автоматов, полученных известными (а,б) и новой (в) процедурами.

$$\{(a_1,3)(a_0,\underline{\underline{4}})(a_1,5)\} * \{(a_1,6)\} \to \{(a_0,3)(a_1,4)(a_0,5)\};$$
$$\{(a_1,\underline{\underline{4}})(a_0,5)(a_1,6)\} * \{(a_1,7)\} \to \{(a_0,4)(a_1,5)(a_0,6)\};$$
$$\{(a_1,5)(a_0,6)(a_1,\underline{\underline{7}})\} * \{(a_1,8)\} \to \{(a_0,5)(a_1,6)(a_0,7)\};$$
$$\{(a_1,6)(a_0,\underline{\underline{7}})(a_1,8)\} * \{(a_1,9)\} \to \{(a_0,6)(a_1,7)(a_0,8)\};$$
$$\ldots \qquad \ldots \qquad \ldots$$
$$\{(a_1,n-3)(a_0,n-2)(a_1,n-1)\} * \{(a_1,n)\} \to \{(a_0,n-3)(a_1,n-2)(a_0,n-1)\}.$$

Нетрудно выписать СК-таблицы автоматов сети, полученные в соответствии с процедурами [2,3], тем более, что за исключением нескольких крайних автоматов, все остальные имеют и одинаковые СК-таблицы, и одинаковые связи с соседними автоматами. Считая, что такая работа проделана, изобразим связи автомата с соседями для каждой из процедур. На рис.9,а изображены связи автомата с соседями, полученные по процедуре [2], на рис.9,б – по процедуре [3].

Более детально проведем построение сети для новой процедуры. В качестве клеток, которым сопоставим активные автоматы, выберем клетки с именами I,4,7,I0,... (эти клетки подчеркнуты в перечне микроопераций двойной чертой).

$i = 1$

1	2	3	4	1	2	3
a_1	a_0	a_1	a_1	a_0	a_1	a_0

$i = 2, 5, 8, 11, \ldots$

$i-1$	$i+2$	i
a_0	λ	a_0
a_1	λ	a_1
λ	a_0	a_0
a_0	a_0	a_0

$i = 3, 6, 9, 12, \ldots$

$i-2$	$i+1$	i
a_0	λ	a_0
λ	a_1	a_1
λ	a_0	a_0
a_0	a_0	a_0

$i = 4, 7, 10, 13, \ldots$

$i-2$	$i-1$	i	$i+1$	$i+2$	$i+3$	$i-2$	$i-1$	i	$i+1$	$i+2$
a_1	a_0	a_1	a_1	*	*	a_0	a_1	a_0	λ	λ
*	a_1	a_0	a_1	a_1	*	λ	a_0	a_1	a_0	λ
*	*	a_1	a_0	a_1	a_1	λ	λ	a_0	a_1	a_0

Рис. I0

СК-таблицы автомата приведены на рис.I0. На этом рисунке i – это имя автомата. Фрагмент сети изображен на рис.9,в.

Используя рис.9, можно сопоставить все три процедуры. Мы видим, что процедуры [2,3] порождают однородные сети. Каждый автомат сети и в том, и в другом случае имеет одно и то же число входов-выходов, но для процедуры [3] число соседей у каждого автомата меньше чем для процедуры [2]. Новая процедура порождает квазиоднородную сеть, т.е. сеть, состоящую из итеративно повторяющихся фрагментов, содержащих автоматы двух типов. Активные автоматы (заштрихованы) имеют такое же число входов-выходов, что и автоматы в двух других сетях. Пассивные автоматы (не заштрихованы) имеют по два входа и два выхода. В среднем число входов-выходов у каждого автомата в этой сети на треть меньше, чем в сетях, изображенных на рис. 9,а,б.

Заключение

В работе предложена процедура, обеспечивающая формализованный переход от микропрограммного описания алгоритма решения некоторой задачи к логической структуре выполняющего этот алгоритм устройства и учитывающая требования современной элементной базы. Процедура может быть использована при автоматизации этапа структурного синтеза вычислительного устройства.

Литература

1. БАНДМАН О.Л., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. Применение методов параллельного программирования для синтеза структуры специализированных вычислителей. -Новосибирск, 1983. - 31 с. (Препринт/Институт математики СО АН СССР: №35 (ОВС-18)).

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

3. БАНДМАН О.Л., ПИСКУНОВ С.В., СЕРГЕЕВ С.Н. Синтез параллельных микропрограммных структур. -Кибернетика, 1981, №5, с. 48-54.

4. Методы параллельного программирования /Анишев П.А., Ачасова С.М., Бандман О.Л. и др. Под ред. О.Л.Бандман. - Новосибирск: Наука, 1981. - 180 с.

5. БАРАНОВ С.И. Синтез микропрограммных автоматов.-Л.: Энергия, 1974. - 216 с.

Поступила в ред.-изд.отд.
19 сентября 1985 года