АРХИТЕКТУРА И МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ
(Вычислительные системы)
1984 год Выпуск 104

УДК 681.142.2

К ПРОБЛЕМЕ АВТОМАТИЗАЦИИ РАЗРАБОТКИ СИСТОЛИЧЕСКИХ СИСТЕМ

В.А.Бальковский

Исследования, связанные с разработкой систолических систем для тех или иных назначений активно ведутся с конца 70-х годов (см.,например, обзор [1]). Получаемые системы, как правило, являются ориентированными на ту или иную конкретную задачу, а их конструирование целиком зависит от искусства разработчика. Естественно желание выделить во всем многообразии методов разработки более или менее универсальные приемы и в конечном итоге создать процедуру достаточно общего назначения, позволяющую конструировать систолические системы по некоторым спецификациям автоматически или полуавтоматически, как это делается при синтезе параллельных программ [2].

В работах [3-5] предлагаются методы, позволяющие систематически формировать систолические системы по спецификациям в виде гнезд циклов и рекуррентных соотношений. В настоящей работе проблема автоматизации исследуется более детально: определяется более широкий класс входных спецификаций, описывается, как можно полностью автоматизировать ряд наиболее важных промежуточных шагов, более точно формулируются требования, предъявляемые к результирующим систолическим системам. Изложение иллюстрируется на простом примере.

I. Систолические системы

Принято считать, что не может быть дано строгого определения систолических систем, так же, например, как и не может быть дано математическое определение параллельного метода вычислений, или даже просто любого метода вычислений. Строгие формулировки резко

сужают класс специфицируемых систем, в их число не попадают те или иные, порой весьма изящные решения. Поэтому мы введем класс

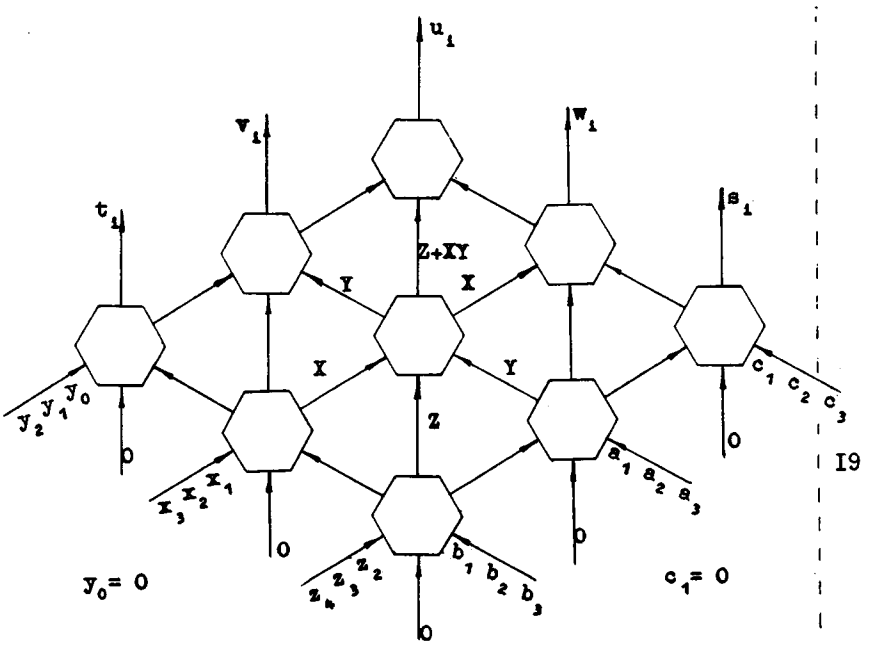

Рис.I

систем некоторым перечнем неформальных требований к ним, иллюстрируя эти требования, как и дальнейшее изложение, на примере системы для умножения трехдиагональных матриц (рис.I):

$$\begin{pmatrix} a_1 & b_1 & & & \\ c_2 & a_2 & b_2 & & 0 \\ & c_3 & a_3 & b_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & c_n & a_n \end{pmatrix} \begin{pmatrix} x_1 & y_1 & & & \\ z_2 & x_2 & y_2 & & 0 \\ & z_3 & x_3 & y_3 & \\ & & \ddots & \ddots & \ddots \\ 0 & & & z_n & x_n \end{pmatrix} =$$

$$= \begin{pmatrix} u_1 & v_1 & t_1 & & & & \\ w_2 & u_2 & v_2 & t_2 & & & \\ s_3 & w_3 & u_3 & v_3 & t_3 & & \\ & & \ddots & \ddots & \ddots & \ddots & \\ & & & & & s_n \quad w_n & u_n \end{pmatrix} .$$

В этой записи

$$1.\ t_i = b_i y_{i+1}, \quad i = 1,\ldots, n-2;$$

$$2.\ v_i = a_i y_i + b_i x_{i+1}, \quad i = 1,\ldots, n-1;$$

$$3.\ u_i = c_i y_{i-1} + (a_i x_i + b_i z_{i+1}); \quad i = 1,\ldots, n; \qquad (1)$$

$$4.\ w_i = c_i x_{i-1} + a_i z_i, \quad i = 2,\ldots, n;$$

$$5.\ s_i = c_i z_{i-1}, \quad i = 3,\ldots, n .$$

Система, производящая перемножение матриц, состоит из элементов, имеющих форму гексаэдра с тремя входами и тремя выходами (см. рис.I). Элемент X, поступающий слева, никак не изменяется и выходит справа. Аналогично "насквозь" справа-налево проходит элемент Y . А к элементу Z, поступающему снизу, добавляется произведение XY , и результат выходит по верхней стрелке. Легко проверить, что структура из описанных элементов, изображенная на рис.I, получая на вход шесть диагоналей аргументов, выдает на выходе пять диагоналей результата. (Предполагается, что в начальный момент на всех непомеченных стрелках находятся нули.)

Среди наиболее важных свойств, которым должна удовлетворять система, обычно называют следующие;

а) однородность и элементарность преобразователей;

б) регулярность и локальность связей между ними;

в) параллельность и конвейерность обработки;

г) синхронность взаимодействия;

д) возможность плотного расположения элементов и удобный ввод-вывод (входные и выходные полюса должны быть на периферии системы).

К этим требованиям мы добавим еще несколько менее обсуждаемых в литературе:

е) никакое данное дважды не вводится, не перевычисляется и не дублируется;

к) каждый элемент и каждое (промежуточное) данное на каждом такте участвуют в полезной (т.е. влияющей на выходной результат) операции;

з) структура системы инвариантна относительно размерности задачи.

Последний пункт требует пояснения. При обработке данных систолической системой можно выделить два типа размерностей задачи: главную и второстепенные. Неформально, главная размерность - это "длина" обрабатываемого потока данных, а второстепенная - его "ширина". Так, в нашем примере n является главной размерностью, а три - число диагоналей - второстепенной. Требование "з" может быть уточнено следующим образом: систолическая система должна быть одна и та же для задач, отличающихся только главной размерностью, и изменяться (например, достраиваться) регулярным образом при изменении второстепенной размерности. Легко проверит, что для умножения, например, пятидиагональных матриц годится система точно такой же структуры, что и на рис.I, но только размером 5x5. Таким образом, она удовлетворяет требованию "з", равно как и другим перечисленным требованиям.

2. Входные спецификации

Пусть заданы некоторые конечные множества: M - переменных с индексами, принадлежащими множеству целых чисел (допустимы также индексные выражения из арифметических действий), и F - множество операций. Пусть T - множество термов над $M \cup F$, построенных обычным образом (см., например, [2]). Символом $t(i_1,\ldots,i_m)$ обозначим терм, имеющий в качестве индексов своих переменных в точности множество $\{i_1,\ldots,i_m\}$. Соотношением над M,F назовем выражение вида:

$$x(i_1,\ldots,i_m,j_1,\ldots,j_k) = t(i_1,\ldots,i_m), \qquad (2)$$
$$i_s \in I_s,\ s = 1,\ldots,m;\ j_s \in J_s,\ s = 1,\ldots,k.$$

В этой записи $x \in M$, $t \in T$, а I_s, J_s - некоторые достаточно просто описываемые множества целых чисел.

В качестве входной спецификации систолической системы (СС), обозначаемой S(СС), будем использовать произвольное конечное множество соотношений с указанием массивов входных и выходных переменных. Соотношения могут быть рекурсивными. Считается, что S(СС) является правильной в следующем смысле.

1. Допускается, что некоторые из индексов i_s левой части (2) могут отсутствовать. Например, (2) может иметь вид $x = a(i), i = 1,\ldots,10$. Тогда интерпретация (2) должна быть корректной в том смысле, что для всех значений отсутствующих индексов значение терма t - одно и то же. В нашем случае $\forall i,j\ (a(i)=a(j))$.

2. Естественно, при произвольном указании входных и выходных массивов не гарантируется, что последние можно вычислить из первых с использованием данных соотношений. Поэтому этапу синтеза системы должен предшествовать этап анализа S(CC), направленный на выяснение разрешимости задачи. Для такого рода анализа есть развитая техника, в том числе и в случае рекурсивных соотношений. Хотя понятно, что в общем случае, при произвольно задаваемых индексных множествах, эта проблема неразрешима.

Обычно спецификация получается из описания метода решения вполне естественно, так как с методом, как правило, связываются некоторые формулы для массовой обработки данных. Так, в случае нашего примера, спецификацией является совокупность термов (1). Но S (CC) может быть получена и другим способом. Из гнезд циклов, рассматриваемых в [3], спецификация получается обычным выписыванием всех термов, которые вычисляет отдельная итерация. Увязанными друг с другом они оказываются автоматически, через индексы. Спецификация может быть получена также на выходе некоторой системы планирования вычислений, например, на основе вычислительных моделей [6] над массивами переменных.

Этапу построения систолической системы может предшествовать также оптимизация спецификации. Традиционное преобразование оптимизации заключается в экономии вычислений путем выделения в разных термах общих фрагментов. Однако его нужно применять осторожно. Такого рода экономия может повлечь за собой появление глобальных путей передачи данных, т.е. нарушение условия "б" для системы.

3. Таблица обращения переменных

Входная спецификация несет достаточно богатую информацию о задаче и по ней уже можно построить без особых затруднений некоторую систему. Для этого нужно каждый терм S(CC) "реализовать" в виде преобразователей и связей. Для нашего примера такая система показана на рис.2. Однако она не удовлетворяет многим из требований "а"-"з". В частности, не удовлетворяется требование "е": как-

дый из входных потоков требуется в 3-х экземплярах. Но можно попытаться избежать дублирования, установив очередность при передаче общих данных от одного исполнительного устройства к другому. При этом можно следить за тем, чтобы данные активно использовались на каждом такте, а также за выполнением других условий для систолических систем. Все перечисленное удобно делать на так называемой таблице обращения переменных, изображающей режим потребления и вырабатывания переменных исполнительными устройствами.

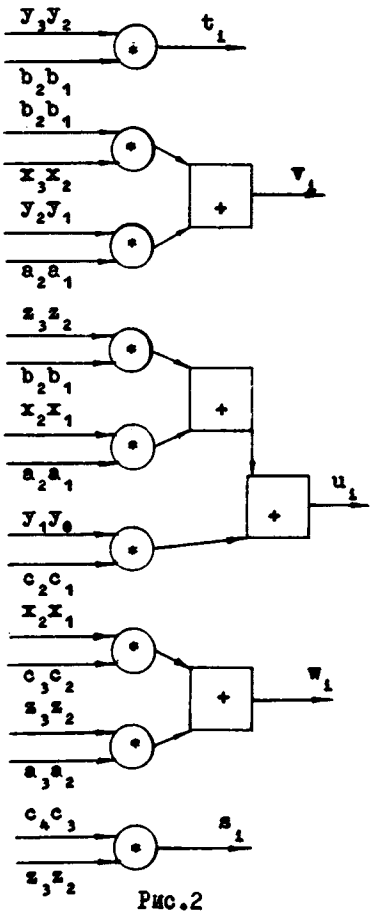

Рис.2

Три версии таких таблиц для нашего примера изображено на рис.3. Прокомментируем их.

Прежде всего, при выборе режима обращения у нас имеется одна дополнительная возможность - выбор режима ввода или вывода переменных. Таблица, изображенная на рис.3,а, составлена в том предположении, что мы зафиксировали режим вывода, когда не позднее чем на k-м такте выводятся t_{k-1}, u_k, v_k, s_{k+1}, w_{k+1}. В таблице вертикальную колонку занимают соотношения спецификации, по горизонтали отложены номера тактов. Таблица составляется следующим образом. Если на k-м такте мы хотим иметь результат первого терма, т.е. t_{k-1}, это значит, что на такт раньше мы должны иметь его аргументы b_{k-1}, y_k. Другими словами, для первого терма требуется два потока переменных b_k и y_k, причем b_k требуется на k-м такте, а y_k - на (k-1)-м.

По той же причине для получения результата v_k второго терма требуется a_k, y_k, b_k - на (k-2)-м такте и x_k - на (k-3)-м и

а)

	K - 4	K - 3	K - 2	K - 1	K
I				b_{k-1} y_k	b_k
II		x_k	$b_k a_k y_k x_{k+1}$		
III	z_k	$b_k a_k x_k c_k$	y_k		
IV		$a_k z_k$ c_k	x_k		
V			c_k	z_k	

I

б)

	K - 6	K - 5	K - 4	K - 3	K - 2
I			y_k b_k		
II		x_k b_k		$a_k y_k$	
III	z_k b_k		a_k x_k	c_k	y_k
IV		$a_k z_k$	c_k	x_k	
V		c_k	z_k		

I9

в)

	K - 4	K - 3	K - 2	K - 1	K
I				y_k	b_k
II		x_k	a_k y_k	b_k	
III	a_k x_k c_k	z_k y_k	b_k		
IV		a_k z_k c_k	x_k		
V			c_k	z_k	

Рис.3. Таблицы обращения переменных.

38

т.д. Таблица, построенная из этих соображений, изображена полностью на рис.3,а.

Нарушение требования "е" видно на этой таблице еще более отчетливо. Так, например, a_k требуется на (k-3)-м такте сразу для 3-го и 4-го термов. Переменная x_k для 2-го и 3-го и т.д. Наша очередная задача заключается в том, чтобы так передвинуть выписанные в таблице аргументы для термов, чтобы каждый из них фигурировал на каждом такте только один раз и передавался от терма к терму без пропусков тактов, т.е. без задержек. Это нужно для обеспечения условия "к". При этом для обеспечения этого же условия пары аргументов: y_k, b_k - для первого терма, a_k, y_k и x_k, b_k - для второго терма и т.д. нельзя разбивать. Их можно двигать только "монолитно", парами.

Как легко видеть, указанная задача является типичной задачей теории расписаний, и может быть вполне решена машинными методами. При этом можно оптимизировать желаемые нами критерии, например, общее время задержки системы. Однако задача решается достаточно эффективно и простейшими эвристическими приемами. Рассмотрим один из них - систематическую "разводку" переменных при движении по таблице в каком-либо направлении. Возьмем направление слева-направо. Левый крайний элемент z_k относится к третьему терму. Он не конкурирует ни с каким другим элементом, и его естественно оставить на месте. На следующем такте в нем нуждается 4-й терм, а оставшийся 5-й терм нуждается в z_k через такт, поэтому мы, для целей сохранения условия "к", сдвинем пару аргументов c_k, z_k 5-го терма влево. Далее рассмотрим парный к z_k аргумент b_k и проделаем с ним ту же процедуру. Пару аргументов первого терма y_k, b_k также придется сдвинуть на один такт влево. Далее рассмотрим a_k, которое требуется для 3-го и 4-го термов. Поскольку аргумент z_k 4-го терма уже фиксирован, то вместе с ним фиксируется и a_k, поэтому мы вынуждены сдвинуть пару аргументов a_k, x_k 3-го терма на такт вправо. Но при этом, как и при любом сдвиге вправо, всю картинку следует сдвинуть на такт влево, т.е. z_k окажется в столбце (k-5)-го такта, b_k, a_k - в столбце k-4 и т.д. Иначе выходной результат u_k будет получен не на k-м, как требовалось, а на (k+1)-м такте. Продолжая аналогичные действия, придем к таблице, изображенной на рис.3,б. В ней уже можно увидеть некоторую регулярность структуры передачи данных.

Другой систематический процесс исключения конфликтов - сверху-вниз, приводит к таблице, изображенной на рис.3,в. В нем нам не удалось удовлетворить одно из требований - переменные b_k и z_k на последнем этапе передаются через такт, следовательно, не избежать нарушения условия "х".

4. Синтез систолической системы

После того, как получена некоторая приемлемая таблица обращения переменных, по ней довольно просто строится систолическая система. Для этого достаточно соединить в нужном порядке части, соответствующие термам спецификации. Для нашего примера они изображены на рис.2. Для таблицы рис.3,б результирующая система изображена на рис.4.

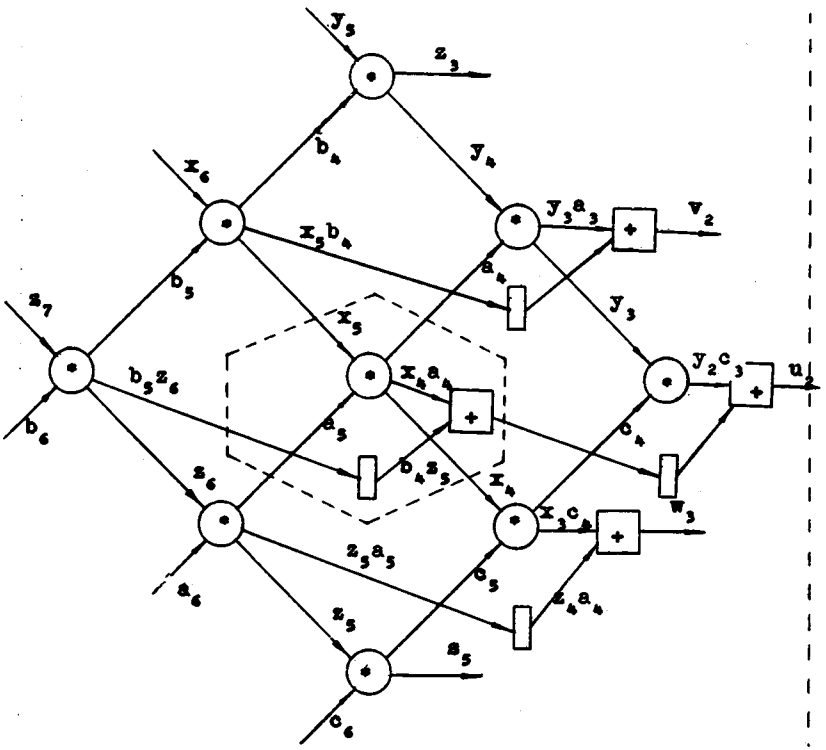

Рис.4

Однако она не вполне удовлетворяет требованиям к систолической системе: в ней присутствуют элементы трех, а это значит многих типов, и,самое главное, в нее пришлось ввести элемент задержки на один такт, изображенный на рис.4 знаком □ . Нарушается условие "к". Поэтому требуется дальнейшая работа над построенной системой - ее анализ в целях нахождения подходящего базиса,быть может,из несколько более сложных элементов, но превращающего систему в более однородную и структурно простую.

Один из традиционных способов упрощения любой системы - нахождение в ней каких-либо регулярностей и повторений и выделение повторяющейся части в отдельный блок. Это обычная техника макроопределений. Уже в нашем случае можно заметить в структуре системы повторение части, обведенной пунктирной линией. В случае сис -

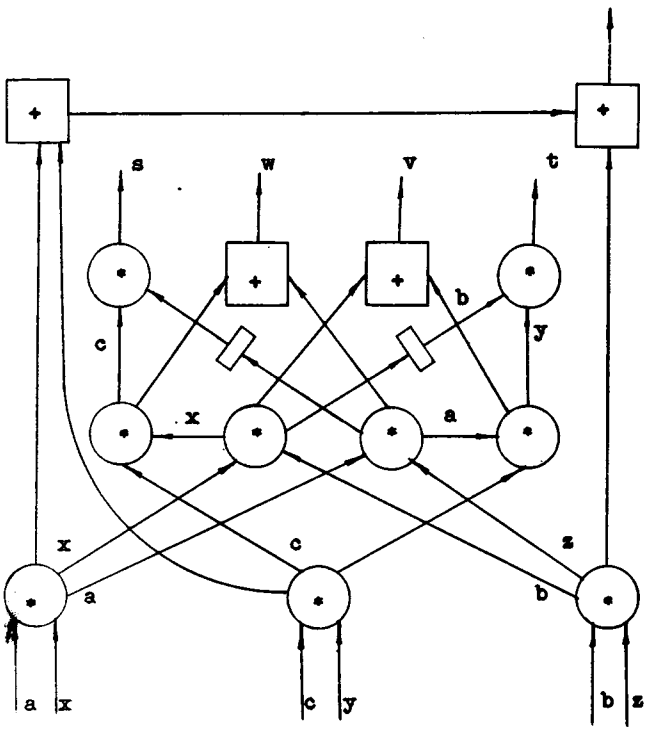

Рис.5

темы для умножения t-диагональных матриц при большом t эта регулярность была бы еще более заметна. Поэтому естественно попытаться выделить ее в отдельный элемент. Сделав это, мы можем убедиться, что он выполняет в точности ту же функцию, что и элементы на рис.I. А система после соответствующей замены выявленного фрагмента на этот элемент превращается в систолическую систему, изобра - женную на нем.

Если проделать аналогичную процедуру с таблицей на рис.3,в , то получим систему, изображенную на рис.5. Она имеет те же недо - статки, что и система рис.4, а,кроме того,у нее, с первого взгляда, нарушено условие локальности связей, она не столь регулярна и изящна, как типичные систолические системы, но зато у нее меньше элементов задержки и на два такта меньше общее время задержки. А локальности связей можно добиться, расположив элементы системы в трех плоскостях.

5. Заключительные замечания

Описанная общая схема метода на всех этапах допускает боль - шую свободу выбора при ее уточнении и конкретизации.Необходимость делать проектировщику тот или иной выбор появляется, когда форми - руемый на каком-то этапе объект допускает несколько представлений, имеется более одного пути дальнейшей реализации стратегии синтеза, не могут быть удовлетворены все требования "а"-"з" на проектируе - мую систолическую систему, и нужно пожертвовать какими-то из них в пользу других; либо если получающаяся в итоге система не удов - летворяет проектировщика по каким-то другим, не формализуемым и не учитываемым стратегией критериям.

Неоднозначность представления обнаруживается уже при выборе исходной спецификации. Одна и та же массовая проблема может быть описана несколькими эквивалентными системами соотношений, разли - чающимися входными потоками переменных, составом операций,их после - довательностью. Так,например, спецификация

$$x_i = f(y_i, g(y_i)),$$

$$z_i = h(g(y_i)), \qquad i = 1, \ldots, n,$$

эквивалентна следующей:

$$x_i = f(y_i, v_i),$$

$$z_i = h(v_i),$$

$$v_i = g(y_i), \quad i = 1, \ldots, n .$$

В последней экономится одно вычисление подтерма $g(y_i)$, но появляется дополнительный массив v, которым придется дублировать или передавать между термами $f(\ldots)$ и $h(\ldots)$, что вызовет их последовательное выполнение и увеличение задержки всей системы.

Неоднозначность может следовать из каких-либо специальных соотношений. Так, в спецификации (I) третий терм на основании закона ассоциативности мог бы быть записан в виде $(c_i y_{i-1} + a_i x_i) + b_i z_{i+1}$. Это повлияло бы на заполнение таблицы обращения переменных и, возможно, на структуру разрабатываемой системы. Можно было бы избежать неоднозначности на этапе спецификации проблемы, введя некоторую каноническую форму спецификации. Но представляется, что этого делать пока не следует, оставив проектировщику определенную свободу выбора и возможность проигрывания различных вариантов.

Другой пример неоднозначного представления появляется при выборе таблицы. Номера соотношений спецификации могут быть расставлены в инициальную колонку произвольно. Из требований локальности связей можно рекомендовать такой их порядок, при котором выполняется своеобразное свойство связности общих данных: используемые термами (соотношениями) общие массивы располагаются вплотную между собой, без промежутков между ними в виде табличных строк, не содержащих переменных с этим именем. Таблица рис.3,а удовлетворяет этому условию: массивы x и a используются термами со 2-го по 4-й, y и b – с I-го по 3-й, z и c – с 3-го по 5-й. Таким образом, все они занимают сплошные полосы строк. Естественно, такого расположения можно добиться не всегда. Например, для спецификации

$$u_i = f(x_i, y_i),$$

$$v_i = f(y_i, z_i),$$

$$w_i = f(z_i, x_i)$$

это невозможно. Какая-то из переменных, например x, окажется "размазанной". Из табл.I видно, что x относится к I-му и 3-му тер-

Таблица I

1	...	x_i y_i	u_i	...
2	...	y_i z_i	v_i	...
3	...	z_i x_i	w_i	...

Таблица 2

1	...	y_i	x_i	u_i	...
2	...	z_i	y_i	v_i	...
3	...	x_i	z_i	w_i	...

му, т.е. непрерывный интервал I-3 разделен второй строкой. Однако если представить таблицу в форме цилиндра со склеенными верхней и нижней линиями, то I-й и 3-й термы окажутся соседними - смежными. Результат разводки и направления передачи переменных даны в табл.2. Цилиндрическая форма таблицы определяет и форму "платы", на которой расположится будущая систолическая система. Таким образом, выбор порядка соотношений в таблице обращения переменных существенно влияет на топологию разрабатываемой системы.

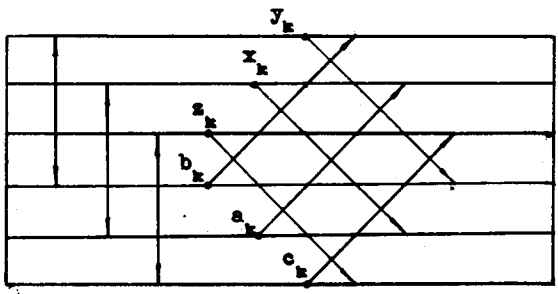

Рис.6

Пример неоднозначности при выборе пути на последующем этапе разработки дает выбор стратегии разводки переменных в таблице обращения. Но в ряде случаев удается вырабатывать довольно универсальные рекомендации, уменьшающие неоднозначность. Например, если таблица обладает свойством связности, то удобно производить разводку переменной x_k, начиная с одной стороны занимаемой ею в таблице полосы к другой. Для таблицы рис.3,а схематически это показано на рис.6. При указанном режиме разводки каждый очередной преемник каждой переменной определяется однозначно, позиции переменных в таблице устанавливаются форсированным образом и в широком ряде случаев арифметических соотношений результирующая таблица оказывается независима от порядка расстановки скобок в исходных термах спецификации. Это наблюдается, например, в случае перемножения ленточных матриц.

В процессе разработки может обнаружиться, что требования "а"-"в" в совокупности с пожеланиями пользователя не могут быть удовлетворены. Например, легко заметить, что если для системы выпол-

няется требование "е", то для каждого потока данных x в каждый момент времени в системе будет находиться только один элемент этого потока с фиксированным индексом - x_i : требование "е" делает невозможным его дублирование. Если при этом несколько элементов пользуются потоком данных x, то они с необходимостью должны быть выстроены в системе последовательно с конвейерной передачей элементов потока x между ними. И общее время обработки одной порции информации будет определяться уже не только и не столько сложностью обрабатываемых выражений, сколько объемом общих используемых данных. Но пользователь может потребовать минимальность задержки результирующей систолической системы. В этом случае конвейерной организации естественно предпочесть параллельную. Но тогда мы сталкиваемся с необходимостью нарушения требования "е": либо нужно будет вводить несколько экземпляров массива x, либо придется его размножить в системе.

Подводя итог, можно сделать заключение, что человеческий компонент при проектировании систолических систем занимает на данном этапе разработки методов автоматизированного проектирования пока ведущее место.

Литература

1. Distribuovane a paralelne systemy /Gruska J., Privara I., Ružička P. a kol. Výskumne prace. Výskumne vypočtove stredisko.- Bratislava,1983.- 250 p.

2. Элементы параллельного программирования /Вальковский В.А., Котов В.Е.,Марчук А.Г.,Миренков Н.Н. - М.: Радио и связь,1983. - 240 с.

3. MOLDOVAN D.I. On the design of algorithms for VLSI systolic arrays.-Proc.of the IEEE,1983,v.71,M 1,p.5-9.

4. QUINTON P. The systematic design of systolic arrays. -INRIA,1983.- 35 p. (Inst.Nat. de Rech. en Inform. et en Autom., Rapport de Recherche, N 216, France).

5. LISPER B. Description and synthesis of systolic arrays. T.R. TRITA-NA-8318. (The Royal Inst.of Technol.,Stockholm,Sweden), 1983.- 66 p.

6. МАЛЫШКИН В.Э. Синтез параллельных программ на вычислительных моделях с массивами. I. Формальная модель. - Новосибирск,1982. - 29 с. (Препринт: ВЦ СО АН СССР, № 379).

Поступила в ред.-изд.отд.
10 сентября 1984 года