МАТЕМАТИЧЕСКОЕ И АРХИТЕКТУРНОЕ ОБЕСПЕЧЕНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ
(Вычислительные системы)
1985 год Выпуск 109

УДК 681.322

ПРИМЕНЕНИЕ АЛГОРИТМОВ ПАРАЛЛЕЛЬНЫХ ПОДСТАНОВОК ДЛЯ ОПИСАНИЯ СИСТОЛИЧЕСКИХ УСТРОЙСТВ

В.П.Маркова

Начиная с 80-х годов, в зарубежной литературе большое внимание уделяется разработке параллельных алгоритмов решения широкого круга задач в устройствах, названных систолическими [1-3].

Принципы построения систолических устройств: параллельность обработки информации, локальность, однородность (вернее, квазиоднородность), синхронность, перестраиваемость, децентрализация функций памяти, управления и преобразования не являются новыми в организации ЭВМ. Они полностью либо частично присутствуют в таких устройствах параллельного типа, как вычислительные среды [4], ассоциативные процессоры [5], системы из микропроцессоров [6], параллельные микропрограммные структуры [7,8] и т.д. Интерес к систолическим устройствам параллельной обработки информации связан с появлением больших интегральных схем, которые дали практическую возможность реализации моделей параллельных вычислений типа клеточного автомата [9].

Существует несколько моделей систолических устройств. Все они являются модификациями клеточного автомата и предназначены для разных целей. Так, например, модель в [10] используется для вложения циклических алгоритмов решения задач, модель в [11] - для верификации и т.д. В данной работе в качестве модели систолических устройств используются алгоритмы параллельных подстановок [7].

Алгоритмы параллельных подстановок служат основой для микропрограммного описания локальных параллельных преобразований информации в массивах данных. Объектом преобразования является клеточное множество, представляющее собой совокупность клеток W, характеризуемое именем и состоянием. Средством преобразования служит операция подстановки П: $S_1 * S_2 \to S_3$, где S_1 S_2 и S_3 - конфи-

гурации,т.е. способ задания структуры данных (имен клеток и их состояний), участвующих в операции П.Выполнение подстановки осуществляется над всеми клетками множества W и заключается в замене клеточного множества $S_1(m)$ на $S_3(m)$ при условии, что $S_1(m) \cup S_2(m) \subset W$,где m - имя клетки.

Неупорядоченное множество микрокоманд (стационарных подстановок) образует параллельную микропрограмму [8], которая от обычной отличается тем, что в ней все применимые микрокоманды выполняются одновременно и повсеместно. Известно, что параллельные микропрограммы интерпретируются сетями из автоматов. Это означает, что каждой клетке из множества W ставится в соответствие автомат с тем же именем и состоянием. Функционирование автоматов и структура их связей однозначно определяются по исходной микропрограмме. Если парралельная микропрограмма построена с использованием только локальных конфигураций [I2], то интерпретирующая ее структура - сеть синхронно действующих автоматов. Все автоматы сети работают параллельно и расположены в узлах двумерной целочисленной решетки, каждый автомат связан со своими ближайшими соседями. И если микропрограмма составлена таким образом, что информация в сети распространяется конвейерным способом, то автоматная реализация такой микропрограммы представляет собой систолическое устройство. И наоборот, вычислительный процесс в любом систолическом устройстве может быть записан в виде параллельной микропрограммы.

К настоящему времени на основе алгоритмов параллельных подстановок и аппарата сетей Петри [8] разработаны методы синтеза параллельных устройств. В [I3] приведены формальные процедуры перехода от параллельной микропрограммы к структурной схеме устройства. Ведутся исследования свойств асинхронной интерпретации параллельных микропрограмм [I4,I5], которые могут описывать функционирование другого класса устройств параллельного типа - волновых процессоров [I6]. Предложены методы синхронной и асинхронной композиций алгоритмов [I2]. Известно, что представление сложного алгоритма в виде композиции простых, каждый из которых вкладывается в отдельное систолическое устройство, является открытой проблемой в систолическом проектировании [2].

В данной работе на примере задачи вычисления свертки в линейном систолическом устройстве показано составление параллельной микропрограммы алгоритма, включение ее в композицию и переход от синхронной интерпретации микропрограммы к асинхронной.

§ I. Вложение алгоритмов решения задач в систолические устройства

I. Определение систолических устройств. Систолические устройства представляют собой сети локально связанных процессорных элементов (клеток), не обязательно одинаковых. Клетки в таких устройствах взаимодействуют синхронно и работают параллельно. Информационные потоки распространяются конвейерным способом, они могут двигаться в разных направлениях и с разными скоростями. Связь с внешней средой осуществляется только через граничные клетки. По структуре связей в сети систолические устройства делятся на деревья и одно- или двухмерные массивы. (В статье рассматриваются только систолические массивы, хотя алгоритмы параллельных подстановок могут применяться для описания вычислений и в систолических деревьях.)

В качестве примера продемонстрируем вычисление свертки на линейном систолическом устройстве. Пусть $X = (x_1, x_2, \ldots, x_n)$ - входная последовательность отсчетов, $W = (w_1, w_2, \ldots, w_k)$ - последовательность весов. Требуется вычислить свертку Y, представляющую собой последовательность $(y_1, y_2, \ldots, y_{n+1-k})$, компоненты которой определяются по формуле

$$y_i = \sum_{j=1}^{k} w_j x_{i+j-1} = \sum_{j=1}^{k} y_i^j, \qquad i = 1, 2, \ldots, n+1-k.$$

Существует несколько вариантов реализации алгоритма вычисления свертки на линейном систолическом устройстве. Здесь мы выбрали вариант, предложенный Кунгом [2], суть которого состоит в следующем. Входные отсчеты, разделенные друг от друга одним тактом, из источника поступают на крайнюю левую клетку устройства (см. рис.I,а и рис.2) и продвигаются в нем слева направо. Начальное значение первой компоненты свертки равное нулю, обозначим ее через y_1^0, поступает на крайнюю правую клетку устройства спустя (k-1) такт после начала его работы. Остальные компоненты $y_i^0 = 0$ вводятся в устройство через такт. Частичные суммы y_i^j формируются следующим образом. На каждом такте в j-ю клетку поступают $x_{вх} = x_i$ и $y_{вх} = y_i^{j-1}$ (см.рис.I,б) и вычисляется сумма

$$y_i^j = y_i^{j-1} + x_i w_j. \qquad (1)$$

Затем значение $x_{вых} = x_i$ передается в (j-1)-ю клетку систолического устройства, значение y_i^j - в (j+1)-ю клетку, где они участ-

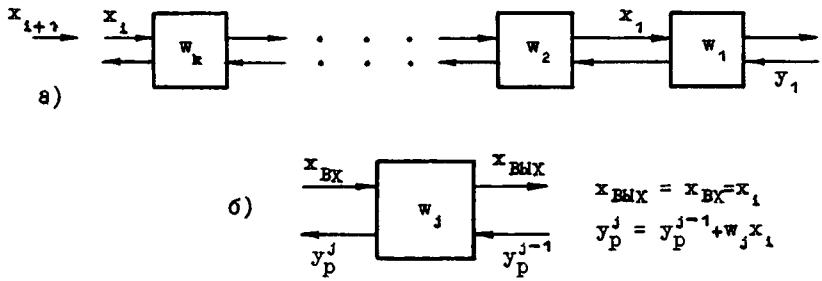

Рис.I. Систолическое устройство (а), клетка систолического устройства для вычисления свертки (б).

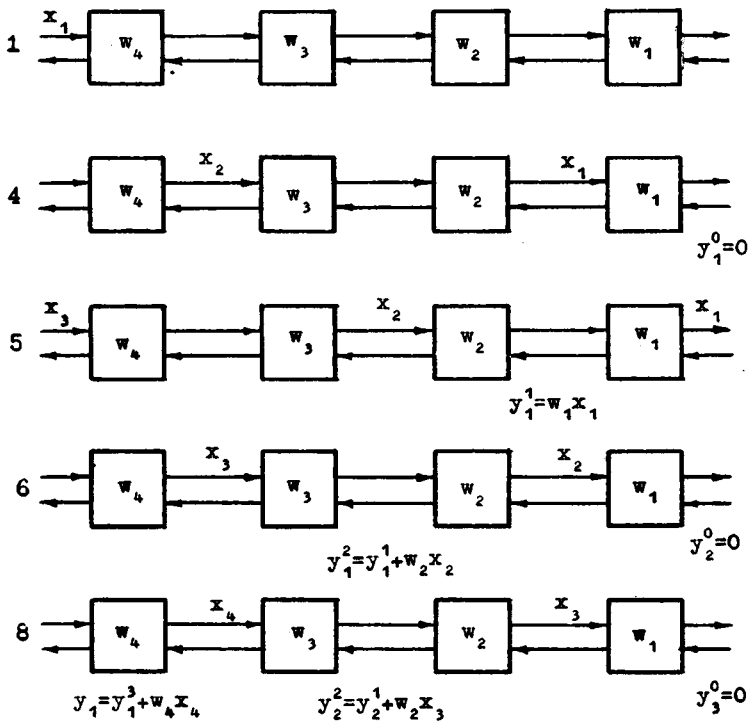

Рис.2. Временная диаграмма вычисления первого элемента свертки на систолическом устройстве для $k = 4$.

вуют в вычислении частичных сумм y_{i+1}^{j-1} и y_i^{j+1} соответственно. Значение первой компоненты свертки получается на 2k-м такте выполнения алгоритма. Вычисление последовательности Y продолжается до тех пор, пока все входные отсчеты не пройдут через клетки систолического устройства.

2. Вложение алгоритмов решения задач в систолические устройства. Процедура вложения алгоритмов в систолические устройства выполняется в два этапа. На первом этапе исходный алгоритм преобразуется в параллельный, на втором - по параллельному алгоритму определяются структура и функционирование систолического устройства в виде тройки параметров $\langle G, F, H \rangle$. Геометрия G задает местоположение каждой клетки сети координатами двухмерной целочисленной решетки. Функции F описывают поведение клеток. Синхронизация H определяет для каждой клетки время выполнения функций и время пересылки информации. Существует два подхода к построению процедуры отображения алгоритмов. Автором первого подхода является Молдован [10], второго - Квинтон [17].

Процедура Молдована предназначена для вложения циклических алгоритмов, записанных в виде программ. Суть процедуры заключается в следующем. Сначала по программе исходного алгоритма выявляются векторы связей данных. Затем они модифицируются таким образом, чтобы полученный в результате параллельный алгоритм, во-первых, был эквивалентен исходному и, во-вторых, наилучшим образом удовлетворял требованиям реализации в систолических устройствах, одним из свойств которых является локальность связей. Формально такая модификация записывается в виде отображения:

$$T: \langle \mathcal{L}^n, R \rangle \to \langle \mathcal{L}_T^n, R_T \rangle,$$

где $\mathcal{L}^n, R$ и $\mathcal{L}_T^n, R_T$ - множество индексов и упорядоченность векторов связи данных последовательного и параллельного алгоритмов соответственно, n - количество циклов.

Параметры устройства G, F и H определяются непосредственно из параллельного алгоритма. Функции F выводятся из математических выражений алгоритма. Синхронизация сети связана с новым упорядочением векторов связей данных и задается его первыми k, $k<n$, компонентами, т.е. H: $\mathcal{L}^n \to \mathcal{L}_T^k$. Геометрия сети находится из отображения G: $\mathcal{L}^n \to \mathcal{L}_T^{n-k}$. При этом клетки систолического устройства ставятся в соответствие элементам из $\mathcal{L}_T^{n-k}$, связи между клетками определяются последними (n-k) компонентами модифицированных век-

торов связей данных. Если $n-k>2$, то выполняется дополнительное отображение $G': \mathcal{L}_T^{n-k} \to \mathcal{L}_T^2$.

Мы не будем также подробно описывать подход, предложенный Квинтон, а лишь укажем на его основное отличие от подхода Молдована. Оно состоит в исходном задании алгоритма. В процедуре Квинтон алгоритм задается в виде системы однородных рекуррентных уравнений. Такая форма представления алгоритма дает больше возможностей для его формального преобразования. Однако в этом случае возникают дополнительные задачи: приведение исходного алгоритма к системе однородных рекуррентных уравнений и нахождение условий, при которых эта система имеет последовательную схему решения.

На данный момент систолическое проектирование имеет ряд нерешенных проблем. К ним, исключая технологические, относятся проблема ввода-вывода информации, минимизация связей в сети, моделирование, представление исходного алгоритма в виде композиции простых, каждый из которых вкладывается в отдельное систолическое устройство, и т.д.

Ниже будет показано, что ряд перечисленных проблем, в частности, проблема ввода-вывода информации и композиция алгоритмов, могут быть решены методами параллельного микропрограммирования, математической основой которого являются алгоритмы параллельных подстановок и аппарат сетей Петри.

§ 2. Микропрограммные описания параллельных вычислений в систолических устройствах

Объектом преобразования любой микропрограммы является массив данных, записанный в виде клеточного множества. Клеточное множество - это конечная совокупность клеток $W = \{(a_k, m_k)\} \subseteq A \times M$, где A - алфавит состояний, W - множество имен клеток, $k = 1, 2, \ldots, p$, в котором никакие две клетки не имеют одинаковых имен. Понятие клетки имеет двойную интерпретацию. С одной стороны, клетка - это абстрактный элемент памяти, сопоставленный элементу перерабатываемого массива данных. С другой стороны, клетка представляет собой элемент структуры параллельного устройства. Отсюда, следуя определению систолического устройства, клетка - это пара координат $\langle i\ j \rangle$, $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, L$, двумерной целочисленной решетки, в которую вписан некоторый символ из алфавита A.

Для перечисления клеточных множеств используются конфигурации. Пусть векторы $(a_1,a_2,\ldots,a_p)$ и $(m_1,m_2,\ldots,m_p)$ - первая и вторая проекции множества W, $\varphi(m)$: M→M - именующая функция. Совокупность клеточных множеств, у которых первые проекции совпадают, а вторые задаются с помощью набора именующих функций, называется конфигурацией и обозначается через $S = \{(a_1,\varphi_1(m)),(a_2,\varphi_2(m)),\ldots,(a_p,\varphi_p(m))\}$. В конфигурации все именующие функции попарно различны. На практике в качестве именующих функций используются функции сдвига по осям координат. Задавая конкретное m_k из множества имен M, можно записать конкретное множество W из конфигурации S. Пусть, например, A={0,1}, M={⟨i,j⟩}, i=1,2,...,N, j=1,2,..., L. Множество $S = \{(1,\langle i,j\rangle),(0,\langle i,j\rangle)\}$, $i > 1$, (рис.3) - конфигурация с именующими функциями $\varphi_1(m) = i,j$ и $\varphi_2(m) = i,j$.

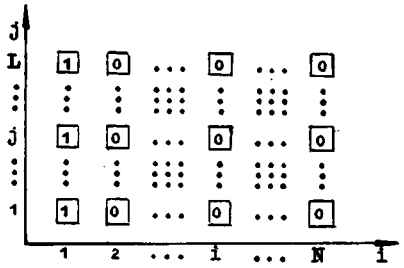

Рис.3

Элементарным преобразованием, выполняемым над клеточным множеством, является операция подстановки вида

$$П: S_1 * S_2 \to S_3.$$

Подстановка состоит из двух частей: левой $S_1 * S_2$ и правой S_3. Конфигурация S_1 называется базой, S_2 - **контекстом**, $S_1 * S_2$ - произведением базы и контекста. Если $Pr_1(S_1) = Pr_1(S_3)$, то подстановка называется стационарной или микрокомандой. Микрокоманда применима к клеточному множеству W, если в нем найдется хотя бы одна клетка (a_k,m_k) такая, что $S_1(m_k) * S_2(m_k) \subseteq W$. Результатом применения микрокоманды П является замена состояний клеток из базы $S_1(m_k)$ на состояния клеток из правой части $S_3(m,k)$. Контекст в условие применимости микрокоманды входит, но в процессе ее выполнения не меняется. Если левая часть микрокоманды содержит символы переменных, отношений и функций, принимающих значения из алфавита A, то такая микрокоманда называется функциональной. Ее применение от обычной отличается тем, что клетки базовой части заменяются на клетки, состояния которых вычисляются как функции состояний соседей. Например, микрокоманда $П: \{(a,\langle i,j\rangle)\} * \{(b,\langle i-1,j\rangle)\}\{(c,\langle i,j+1\rangle)\} \to \{(p,\langle i,j\rangle)\}$, где $p = a + bc$, является функциональной. Действительно, клетка с именем $\langle i,j\rangle$ получает информацию о состоянии соседей слева и сверху и меняет свое состояние с "a" на "a+bc".

Именующие функции в записи конфигураций микрокоманды определяют структуру связей и интерпретирующей ее сети. В нашем случае структура связей задана: каждая клетка систолического устройства соединена со своими ближайшими соседями. Согласно классификации конфигураций [I2], такому типу связей соответствуют так называемые локальные конфигурации. Микрокоманды, построенные из локальных конфигураций, обеспечивают конвейерное распространение информации в систолических устройствах.

Неупорядоченное множество микрокоманд $\Phi = \{\Pi_l : S_{l_1} * S_{l_2} \to S_{l_3}\}$, $l = 1,2,\ldots,\nu$, образует параллельную микропрограмму, описывающую преобразования информации в систолическом устройстве, если конфигурации, входящие в записи каждой микрокоманды, локальны и для любых l и j, $l \neq j$, имеем $S_{l_1} * S_{l_2} \neq S_{j_1} * S_{j_2}$. Микропрограмма Φ задает преобразование клеточного множества W следующей итерационной процедурой: на каждом l-м шаге все применимые микрокоманды выполняются одновременно и повсеместно над всеми клетками множества W^{l-1} (результат $(l-1)$-го шага). Полученный результат служит исходным множеством для следующего шага. Выполнение микрокоманд продолжается до тех пор, пока не получится клеточное множество W^j, к которому не применима ни одна микрокоманда. Оно и будет результатом выполнения микропрограммы.

Таким образом, параллельная микропрограмма описывает функционирование любого алгоритма решения задачи в систолических устройствах с максимальной степенью параллельности и содержит в явном виде всю информацию о структуре и функционировании систолического устройства. Для этого множеству M ставится в соответствие множество имен автоматов, алфавиту A – множество их внутренних состояний. Структура связей автоматов в сети отражена в именующих функциях, входящих в записи конфигураций системы микрокоманд. Функционирование автомата с именем m_k, т.е. его таблицы переходов и выходов строятся на основе микрокоманд, которые в базовых частях содержат клетки с именами m_k. Автоматы в интерпретирующей микропрограмму сети работают синхронно. Это означает, что смена состояний всех автоматов совершается только в моменты подачи тактового сигнала.

§ 3. Параллельная микропрограмма алгоритма вычисления свертки на линейных систолических устройствах

Известно, что составление параллельной микропрограммы начинается с выбора клеточных множеств, т.е. алфавита состояний и

структуры множества имен. Алфавит состояний строится в процессе написания микропрограммы и состоит из двух частей: основной и вспомогательной. Основной алфавит представляет числовую информа - цию 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 множества имен M выбирается с учетом структуры устройства, обрабатываемых массивов информации и алгоритма их образования.

Параллельная микропрограмма исходного алгоритма представляет собой объединение двух микропрограмм. Первая микропрограмма (Φ_1) описывает процедуру непосредственного вычисления свертки, как это делается в систолических устройствах, вторая - (Φ_2) - процедуру ввода-вывода информации. Обычно при написании микропрограммы явного разделения алгоритма на процедуры счета и ввода-вывода не существует. Здесь разделение делается намеренно, чтобы продемонстрировать возможности методов параллельного микропрограммирования для описания ввода-вывода информации, которые в систолическом проектировании отсутствуют.

I. Параллельная микропрограмма алгоритма вычисления свертки Φ_1. Согласно алгоритму вычисления свертки (см. § I), в j-й, j = I,2,..., n, клетке систолического устройства циркулирует следующая информация: хранится вес w_j, вводится значение отсчета x_i и вычисляется частичная сумма y_i^j (I) (см.рис.3). Следовательно, систолическое устройство можно представить в виде двухмерного массива $M_1 = \{\langle j, z \rangle\}$, z = I,2,3,4 (см.рис.4), в котором в клетках с именами $\langle j,1 \rangle$ хранятся веса w_j, в клетках $\langle j,2 \rangle$ - отсчеты x_i, в клетках $\langle j,3 \rangle$ - суммы y_i^j, в клетках с именами $\langle j,4 \rangle$ - признаки u. Признак u, $u \in \{0,1\}$, указывает на режим работы j-й клетки систолического устройства, а именно если клетка с именем $\langle j-1,4 \rangle$

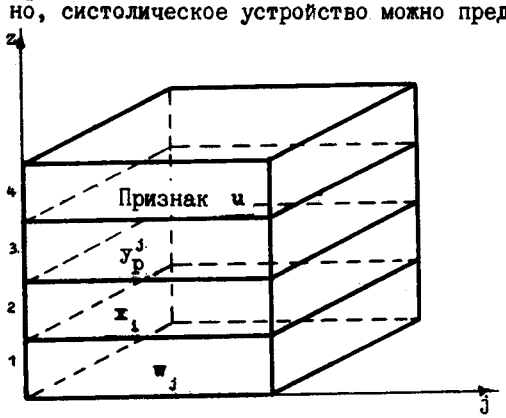

Рис.4

находится в состоянии "I", то клетка j устройства принимает от клетки j+1 отсчет i и вычисляет значение частичной суммы y_i^j (I), в противном случае она только принимает отсчет i .

Основной алфавит задается областью определения отсчетов и весов и областью значений частичных сумм, т.е. основной алфавит - множество комплексных чисел C заданной разрядности. Тогда для массива M_1 алфавит $A_1 = C \times (C \cup \{*\}) \times (C \cup \{*\}) \times \{*, 1, 0\}$, где $* \notin C$. Исходное клеточное множество имеет вид $w^0 = (w_j, \langle j, 1 \rangle) \cup \cup (*, \langle j, 2 \rangle) \cup (*, \langle j, 3 \rangle) \cup (0, \langle j, 4 \rangle)$. При написании микропрограммы переменные символы из основного алфавита обозначаются последними буквами латинского алфавита.

Микропрограмма Φ_1 состоит из двух микрокоманд: первая микрокоманда сдвигает отсчет x_i в сторону меньших j, вторая - сдвигает отсчет и вычисляет значение частичной суммы y_i^j:

$$\Pi_1: (*, \langle j, 2 \rangle) * (x, \langle j+1, 2 \rangle)(0, \langle j-1, 4 \rangle) \to (x, \langle j, 2 \rangle),$$

$$\Pi_2: (0, \langle j, 3 \rangle)(*, \langle j, 2 \rangle)(*, \langle j, 4 \rangle)(v, \langle j-1, 3 \rangle) * * (x, \langle j+1, 2 \rangle)(w, \langle j, 1 \rangle)(1, \langle j-1, 4 \rangle) \to \to (t, \langle j, 3 \rangle)(x, \langle j, 2 \rangle)(1, \langle j, 4 \rangle)(0, \langle j-1, 3 \rangle),$$

где $t = v + xw$.

2. Параллельная микропрограмма ввода-вывода информации Φ_2. Исходная информация, т.е. последовательность отсчетов $\bar{X}$, хранится в массиве $M_2 = \{\langle i \rangle\}$, i=1,2,...,n (см.рис.4), отсчет i последовательности вводится в граничную клетку массива M_2 под управлением клетки $(1, m_1)$ (микрокоманда Π_3). После этого выполняется сдвиг отсчета i в массиве M_1 (микрокоманда Π_1) и подготовка для ввода следующего отсчета, которая осуществляется сдвигом данных в массиве M_2 (микрокоманда Π_4). В микропрограмме начало формирования первой компоненты свертки, т.е. вычисление первой частичной суммы y_1^1 связано с движением признака "I" по массиву $M_3 = \{1\}$ *), l=1,2,...,k-1 (микропрограммы Π_5, Π_6). Окончание формирования y_1^1 фиксируется по изменению клеток с именами $\langle 1, 3 \rangle$ и $\langle 1, 4 \rangle$ с "L" на v, $v \in C$, и с "0" на "I" соответственно (микрокоманда Π_7).

*) Движение признака "I" вдоль массива M_3 начинается одновременно с вводом первого отсчета в клетку с именем $\langle 1, 2 \rangle$.

Предположим, что алгоритм вычисления свертки не требует хранения последовательности отсчетов после ее прохождения через клетки систолического устройства. Тогда в качестве стока последовательности x можно использовать одноклеточный массив $M_5 = \{ m_2 \}$. Запись отсчета i в клетку m_2 сопровождается изменением состояния клетки с именем $\langle 1,4 \rangle$ (микрокоманда Π_8). Смена состояний данной клетки разрешает вычисление следующей компоненты свертки. Через $2k$ тактов выполнения микропрограммы в клетке с именем $\langle 1,4 \rangle$ заканчивается вычисление первой компоненты свертки. Каждая компонента по мере формирования переписывается в правую крайнюю клетку массива $M_6 = \{\langle p \rangle\}$, $p=1,2,\ldots,n+1-k$, (микропрограмма Π_9). После записи очередной компоненты в массиве M_6 выполняется сдвиг данных (микропрограмма Π_{10}). (В исходном состоянии в клетках массива M_6 хранятся символы α, $\alpha \notin C$.)

Таким образом, полное множество имен $M = \bigcup_{j=1}^{6} M_j$, где $M_4 = \{m_1\}$. Полный алфавит состояний $A = \bigcup_{j=1}^{6} A_j$, где $A_1 = C \times (C \cup \{*\}) * (C \cup \{*,\alpha\}) \times \{*,0,1\}$, $A_2 = A_6 = C \cup \{\alpha\}$, $A_3 = \{0\ 1\}$, $A_4 = \{1\}$, $A_5 = C \cup \{*\}$.

Микропрограмма Φ_2 включает следующие микрокоманды:

$$\Pi_3: (*,\langle 1,2 \rangle)(x,\langle 1(M_2)\rangle) * (1,m_1) \to (x,\langle 1,2\rangle)(\alpha,\langle 1(M_2)\rangle),$$

$$\Pi_4: (*,\langle i \rangle) * (x,\langle i+1 \rangle)(\alpha,\langle 1(M_2) \rangle) \to (x,\langle i \rangle),$$

$$\Pi_5: (0,\langle 1(M_3) \rangle) * (1,m_1) \to (1,\langle 1(M_3) \rangle),$$

$$\Pi_6: (0,\langle 1 \rangle) * (1,\langle 1-1 \rangle) \to (1,\langle 1 \rangle),$$

$$\Pi_7: (*,\langle 1,2 \rangle)(\alpha,\langle 1,3 \rangle)(0,\langle 1,4 \rangle) * (1,\langle k-1(M_3) \rangle) \times (w,\langle 1,1 \rangle)(x,\langle 2,2 \rangle) \to (x,\langle 1,2\rangle)(v,\langle 1,3\rangle)(1,\langle 1,4 \rangle),$$

где $v = xw$,

$$\Pi_8: (*,m_3)(1,\langle 1,4 \rangle) * (x,\langle 1,2 \rangle) \to (x,m_3)(0,\langle 1,4 \rangle),$$

$$\Pi_9: (\alpha,\langle 1(M_0) \rangle)(v,\langle k,3 \rangle) * \to (v,\langle 1\ (M_6) \rangle)(\alpha,\langle 1(M_0) \rangle),$$

$$\Pi_{10}: (z,\langle p \rangle)(v,\langle 1(M_6)\rangle) * (u,\langle p-1\rangle) \to (u,\langle p\rangle)(\alpha,\langle 1(M_6)\rangle).$$

Если длина систолического устройства меньше длины последовательности весов W, то для организации вычисления свертки вводится дополнительная конвейеризация [18].

§ 4. Включение в композицию параллельной микропрограммы алгоритма вычисления свертки на линейных систолических устройствах

Известно, что композиция - это способ составления сложной микропрограммы Φ из более простых $\Phi_1, \Phi_2, \ldots, \Phi_q$. Композицию принято представлять параллельной граф-схемой [8], которая от обычной граф-схемы отличается добавлением двух типов управляющих вершин: вершины параллельного ветвления и вершины объединения параллельных ветвей. Параллельная граф-схема полностью характеризует процесс выполнения микропрограммы Φ. Множеству операторных вершин ставится в соответствие множество микропрограмм $\{\Phi_i\}$, $i = 1, 2, \ldots, q$, входящих в композицию. Множество управляющих вершин определяет структуру композиции (порядок действий в параллельной граф-схеме), которая записывается в виде управляющего алгоритма. Поскольку время выполнения микропрограмм Φ_i не определено, то управляющий алгоритм составляется с помощью модели асинхронных взаимодействий - сети Петри. Написание такого алгоритма сводится к формальной процедуре замены вершин параллельной граф-схемы на соответствующие фрагменты сети Петри. Полученная таким образом сеть называется управляющей, а ее подстановочное описание - управляющей микропрограммой Φ_c. Алфавитом для Φ_c является множество $\{0, 1\}$, множеством имен - множество переходов сети, исключая переходы, соответствующие операторным вершинам.

Включение микропрограммы Φ_i в композицию означает ее модификацию, т.е. добавление к ней микрокоманд, воспринимающих сигнал о запуске в работу ("пуск") и вырабатывающих сигнал о получении результата ("стоп"). Микропрограмма, построенная в результате такой модификации, называется расширенной. Общий метод построения расширенных микропрограмм изложен в [8], в [12] предложены частные способы составления расширенных микропрограмм, сохраняя заданный тип конфигураций.

В написании расширенной микропрограммы участвуют по крайней мере две клетки $(0, r_i)$ и $(0, a_i)$ из множества W_c. Сигнал "пуск" состоит в том, что микропрограмма Φ_c изменяет состояние клетки $(0, r_i)$ на $(1, r_i)$, тем самым разрешая выполняться всем применимым микрокомандам. После выполнения всех микрокоманд из Φ_i микропрограмма Φ_c возвращает клетку $(1, r_i)$ в прежнее состояние и изменяет состояние клетки с именем a_i с "0" на "1", извещая таким образом управляющую микропрограмму о получении результата. Если мно-

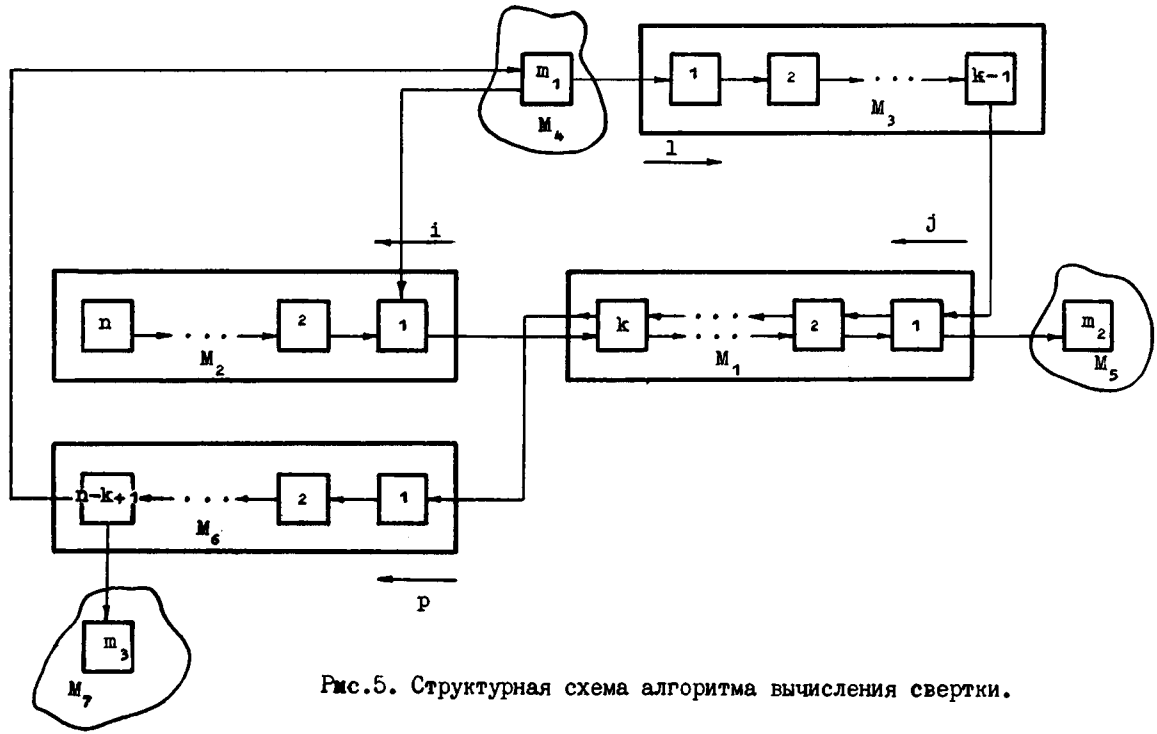

Рис.5. Структурная схема алгоритма вычисления свертки.

жество W_i не содержит клеток, которые можно было бы отождествить с клетками r_i и a_i из W_c, то их следует добавить к множеству W_i и написать микрокоманды, вырабатывающие сигнал "стоп", используя какой-либо признак получения результата.

В нашем случае клетка $\langle 1, r_i \rangle$, разрешающая выполнение микропрограммы алгоритма вычисления свертки, совмещается с управляющей клеткой $\langle 1, m_1 \rangle$ (см.рис.5). Для формирования сигнала о получении результата к исходному клеточному множеству добавляется клетка $\langle 0, m_3 \rangle$. Признаком окончания работы микропрограммы служит появление комплексного числа в крайней левой клетке массива M_6. Таким образом, для расширения исходной микропрограммы необходимо добавить одну микрокоманду

$$\Pi_{11}: \{\langle 1, m_1 \rangle\}\{\langle 0, m_4 \rangle\} * \{u, \langle n-k+1(M_6) \rangle\} \to \{\langle 0, m_1 \rangle\}\{\langle 1, m_3 \rangle\},$$

где $u \in C$.

§ 5. Асинхронная интерпретация параллельной микропрограммы алгоритма вычисления свертки

Асинхронная интерпретация параллельной микропрограммы представляет собой итерационную процедуру [I3,I4], на каждом шаге которой выполняется любое подмножество микрокоманд из числа применимых относительно некоторого подмножества клеток M^i, $i = 1,2,\ldots,k$. Поведение сети автоматов, соответствующей асинхронной интерпретации микропрограммы, имеет сложный характер. Взаимодействия между подмножествами не синхронизовано, т.е. каждый автомат из M^i воспринимает изменения на своих входах, когда они приходят от соседних ему автоматов из подмножества M^{i-1}. Внутри подмножеств автоматы работают синхронно. Здесь синхронность понимается в том смысле, что моменты срабатывания автоматов в подмножествах M^i отличаются друг от друга не более чем на некоторый заданный промежуток времени τ. Общий метод преобразования синхронной интерпретации параллельной микропрограммы в асинхронную описан в [I5]. Ниже продемонстрируем составление асинхронной интерпретации микропрограммы Φ_1 (см. § 3).

Согласно алгоритму вычисления свертки, каждая клетка устройства работает в двух режимах: если признак $u = 1$, $u \in \{0,1\}$, то клетка j принимает отсчет i и вычисляет значение частичной суммы (I), в противном случае она только принимает отсчет i.

Пусть $u=0$. Клетка с именем $\langle j,2 \rangle$ принимает отсчет i при условии готовности информации в соседней клетке слева. Условие готовности фиксируется по признаку "I", который вырабатывается в клетке с именем $\langle j+1,5 \rangle$. Запись отсчета i сопровождается сменой состояний клеток с именами $\langle j+1,5 \rangle$ и $\langle j,5 \rangle$ с "I" на "0" и с "0" на "I" соответственно. Пусть теперь $u=1$. Запись отсчета i и вычисление значения частичной суммы (I) выполняются в том случае, если клетки с именами $\langle j+1,5 \rangle$ и $\langle j-1,6 \rangle$ находятся в единичном состоянии. Признак "I" в клетке $\langle j-1,6 \rangle$ введен для синхронизации процесса вычисления.

Таким образом, множество имен для асинхронной интерпретации микропрограммы Φ_1 получается добавлением к исходному множеству $M = \bigcup_{i=1}^{4} M_i$ множеств $M_5 = \{\langle j,5 \rangle\}$ и $M_6 = \{\langle j,6 \rangle\}$, $j = 1,2,\ldots$ ..., k, клетки которых используются для выработки синхронизирующих признаков, т.е. $M' = M \cup M_5 \cup M_6$. Соответственно $A' = A \cup A_5 \cup A_6$, где $A_5 = A_6 = \{0,1\}$. Асинхронная интерпретация микропрограммы алгоритма вычисления свертки включает следующие микрокоманды:

Π_1': $\{*, \langle j,2 \rangle\}\{1, \langle j+1,5 \rangle\}\{0, \langle j,5 \rangle\} * \{x, \langle j+1,2 \rangle\}\{0, \langle j+1,4 \rangle\} \rightarrow$
$\rightarrow \{x, \langle j,2 \rangle\}\{0, \langle j+1,5 \rangle\}\{1, \langle j,5 \rangle\}$,

Π_2': $\{0, \langle j,3 \rangle\}\{*, \langle j,2 \rangle\}\{0, \langle j,5 \rangle\}\{0, \langle j,6 \rangle\}\{v, \langle j-1,3 \rangle\}\{1, \langle j+1,5 \rangle\} \times \{1, \langle j-1,6 \rangle\} * \{x, \langle j+1,2 \rangle\}\{w, \langle j,1 \rangle\}\{1, \langle j+1,4 \rangle\} \rightarrow$
$\rightarrow \{u, \langle j,3 \rangle\} \times \{x, \langle j,2 \rangle\}\{1, \langle j,5 \rangle\}\{1, \langle j,6 \rangle\}\{0, \langle j-1,3 \rangle\} \times$
$\times \{0, \langle j+1,5 \rangle\}\{0, \langle j-1,6 \rangle\}$,

где $u = v + xw$.

Литература

1. KUNG H.T., LEISERSOW C.E. Algorithmes for Processor Arrays. - In: Introduction to VLSI Systems. Eds.Meed C., Consey L.Reading, MA: Addison-Wesley,1980,p.271-292.

2. KUNG H.T. Why Systolic Architecture? - Computer,1982,v.15, N 1,p.37-46.

3. Систолическая накачка - новый подход к обработке данных.- Электроника,1982,т.55, № II, с.5-7.

4. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные вычислительные системы высокой производительности. -Новосибирск: Наука,1966.-308 с.

5. YAN S.S., TUNG H.S. Associative Processor Architecture. A survey.-Computing Surveys,1977,v.9,N 1,p.3-28.

6. Однородные вычислительные системы на базе микропроцессорных БИС/Бандман О.Л., Евреинов Э.В.,Корнеев В.В.,Хорошевский В.Г. - В кн.: Вопросы теории и построения вычислительных систем (Вычислительные системы, вып.70). Новосибирск,1977,с.3-28.

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

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

9. НЕЙМАН Дж.,фон. Теория самовоспроизводящихся автоматов. - М.: Мир, 1971.- 384 с.

10. МОЛДОВАН Д.И. О разработке алгоритмов для систолических матриц СБИС.-ТИИЭР,1983,т.71,№ 1,с.140-149.

11. RAMI M.G., WERNER R.C. A Mathematical Model for the Verification of Systolic Networks.- SIAM Journal on Computing, 1984, v.13, N 4,August,p.541-564.

12. ПИСКУНОВ С.В. Асинхронная композиция параллельных микро - программ. - В кн.: Архитектура и математическое обеспечение вычислительных систем (Вычислительные системы,вып.104). Новосибирск , 1984,с.30-46.

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

14. АЧАСОВА С.М. Анализ асинхронной интерпретации параллельных микропрограмм.- В кн.: Однородные вычислительные системы из микро-ЭВМ (Вычислительные системы,вып.97).Новосибирск,1983, с.28-52.

15. БАНДМАН О.Л. Асинхронная интерпретация параллельных микропрограмм.- Кибернетика,1984,№ 2,с.14-20.

16. Wavefront Array Processor: Language, Architecture and Applications /Kung S.Y., Arun K.S., Gal-Ezez Ron J., Bhaskaz Rao D.V. - IEEE Trabsactions on Computers,1982,v.C-31, N 11,p.1054-1066.

17. QUINTOW P. The Systematic Design of Systolic Arrays. Centre de Rennes IRISA. Institut National de Recherche en Informatique et en Automatique. Rapports de Recherche N 216,- 36 p.

18. МАРКОВА В.П. Параллельные микропрограммные структуры для ортогональных преобразований.- В кн.: Архитектура и математичес - кое обеспечение вычислительных систем (Вычислительные системы,вып. 104). Новосибирск,1984,с.61-74.

Поступила в ред.-изд.отд.
11 июня 1985 года