ВЫЧИСЛИТЕЛЬНЫЕ СИСТЕМЫ
1971 год Выпуск 41

УДК 681.142.353

ОСНОВНЫЕ СВОЙСТВА ОДНОРОДНЫХ УСТРОЙСТВ С ПЕРЕМЕННОЙ СТРУКТУРОЙ НАСТРОЙКИ

А.И. Мишин

В работе рассматриваются свойства однородных устройств с переменной структурой настройки. Описываются различные способы кодирования и настройки и производится их сравнение. Отмечается возможность настройки устройства с переменной структурой с помощью автомата, реализованного в самом устройстве, а также возможность самовоспроизведения автоматов. Рассматривается применение устройств на примере решения лабиринтных задач.

1°. Однородное устройство с переменной структурой настройки представляет собой решетку, каждый элемент которой может обмениваться информацией только со смежными элементами. Элементы решетки - идентичные конечные автоматы; таблица состояний и правила, определяющие переходы из одного состояния в другое, являются общими для всех элементов. Элементы решетки позволяют изменять конфигурацию путей передачи управляющей информации и локализовать любую неисправность смежного элемента. Это достигается тем, что элементы решетки получают информацию не по принципу координатной выборки, когда отказ даже одной координатной шины делит структуру на две, не связанные между собой области, а по принципу близкодействия, когда при отказе какого-либо участка решетки работоспособность системы настройки не нарушается, так как имеется множество путей обхода неисправных элементов или их группы; управляющая информация передается в решетке последовательно от соседа к соседу вдоль цепочки (цепочек), образованной элементами решетки в процессе настройки. Описанное свойство однородного устройства будем называть живучестью.

Элемент однородного устройства с переменной структурой настройки по своим характеристикам и работе лежит где-то между элементом вычислительной среды с переменной структурой [I] и клеточным автоматом фон Неймана [2].

Блок-схему элемента решетки можно представить в виде двух элементов (рис.I): логического элемента (ЛЭ), реализующего набор логических и вычислительных функций, и управляющего элемента (УЭ), включающего в себя один или несколько разрядов многомерного регистра сдвига (например, двумерного), и блок памяти [3]. Тогда однородное устройство условно можно разделить на две решетки: логическую, состоящую из логических элементов, и управляющую (настроечную), состоящую из управляющих элементов. Каждая из решеток представляет собой однородное устройство, состоящее из однотипных элементов, жестко соединенных между собой. Логические и управляющие элементы могут обмениваться информацией как с одноименными элементами, так и между собой. Последнее имеет большое значение для ввода исходных данных и вывода из устройства результатов вычислений.

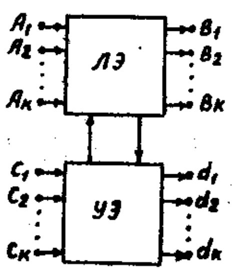


Рис. I

Логический элемент имеет k логических входов A_1, A_2, ..., A_k, на которые поступают сигналы с логических выходов B_1, B_2, ..., B_k смежных элементов решетки. Настройка ЛЭ на выполнение требуемой функции осуществляется выходными сигналами управляющего элемента.

Настройка однородного устройства сводится к выбору из всех возможных состояний устройства такого, которое определяет схему коммутации элементов решетки для выполнения заданной функции. Информация настройки поступает по управляющим входам C_1, C_2, ..., C_k с управляющих выходов D_1, D_2, ..., D_k смежных элементов и запоминается в ячейках памяти настраиваемых элементов. Сигналы с выходов ячеек памяти определяют одно из возможных состояний реализации логического и управляющего элементов.

2^0. Поставим в соответствие решетчатой структуре граф G так, чтобы его вершинам соответствовали элементы решетки, а ребрам – связи между элементами. При этом каждой вершине графа G поставим в соответствие набор функций, реализуемый соответствующим ей элементом решетки. Такой граф G называется графом решетчатой структуры. Каждой конкретной логической схеме, реализованной в решетке, соответствует некоторый подграф G^* графа G, причем каждой вершине этого подграфа соответствует набор функций, реализуемых соответствующим ей элементом решетки и определяемых состояниями соответствующих управляющих элементов.

Таким образом, каждой конкретной логической схеме соответствует определенная совокупность элементов, которую будем называть геометрической конфигурацией.

Для записи информации о геометрических конфигурациях могут быть применены следующие способы кодирования:

I) кодирование абсолютных значений координат элементов;

2) кодирование по принципу приращений (близкодействия)[4];

3) комбинированный.

Примером первого способа является хорошо известный метод записи декартовых координат узловых точек, а примером схемы, реализующей этот способ, может служить схема с двухкоординатной выборкой и двухкоординатным заданием кода адреса элемента. Очевидно, что в случае кодирования абсолютных значений координат одного элемента потребуется не менее $\log_2 mn$ двоичных разрядов, где $m \times n$ – число элементов в решетке.

Примером второго способа является такое кодирование, когда при наложении прямоугольной сетки, например, на кривую, точки кривой кодируются ближайшими к ним значениями в узлах сетки, причем положение каждой из них определяется относительно предшествующей. Рассмотрим процедуру кодирования кривой I,2,...,I0 в решетке, каждый элемент которой может обмениваться информацией с восемью соседними элементами (рис. 2а). Для этого занумеруем возможные направления передачи информации элементом цифрами от 0 до 7 в направлении против часовой стрелки, начиная, например, с горизонтального направления (рис. 2б). Тогда кривая I÷I0, запишется следующей последовательностью:

OOI OOO IIO IOI IOI OOO OOO OIO OII.

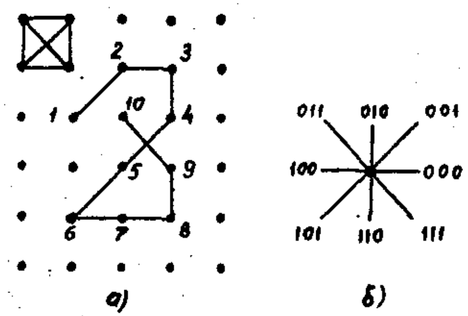


Рис. 2

Таким образом, при кодировании точек кривой по приращениям требуется не более $\log_2 k$ двоичных разрядов для каждой точки, где k определяется количеством соседних элементов (для решетки на рис. 2а $k = 8$).

Если граф G имеет гамильтонов путь, то для записи информации о графе достаточно $\alpha(\log_2 k + \log_2 q)$ двоичных разрядов, где α – число вершин в нем, q – число состояний реализации логического элемента.

Если граф не имеет гамильтонова пути, то для записи информации в сокращенной форме целесообразно выделить в нем такие подграфы (возможно, с преобразованием исходного графа), каждый из которых имеет гамильтонов путь. При этом нужно стремиться к тому, чтобы число различных гамильтоновых путей, а также общая длина путей, с помощью которых осуществляются переходы с одного пути на другой были минимальными. Это хотя и потребует дополнительных разрядов на кодирование путей перехода, однако общее число разрядов будет меньше, чем при кодировании абсолютных значений координат элементов, так как при записи информации о графе не нужно учитывать размеры решетки (решетка может быть сколь угодно большой).

Примером третьего способа является такое кодирование, когда одна часть точек геометрических конфигураций записывается по первому способу, а другая – по второму. Например, информация о путях, с помощью которых можно обойти вершины графа записывается по второму способу, а переходы с одного пути на другой – по первому. При этом количество двоичных разрядов, требуемых для записи всей информации о путях, будет равно

$$H=\sum_{z=1}^{z=\beta}\alpha_z \log_2 k+\beta\log_2\beta ,$$

где α_z – число вершин в z – ом пути,

β – число путей.

При минимизации этого выражения могут быть такие ситуации, когда возникает необходимость в преобразовании исходного графа, например, требуется ввести в граф дополнительные вершины и ребра с целью уменьшения числа путей.

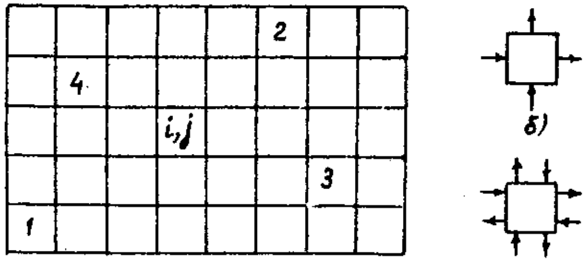
Из вышесказанного следует, что с целью более эффективного использования преимущества способа кодирования по приращениям нужно при размещении логических схем в решетке оптимизировать общую длину пути, с помощью которого можно обойти все элементы решетки, входящие в реализуемые схемы.

Вторым положительным качеством решетки, использующей способ кодирования по приращениям, является то, что однородное устройство обладает высокой живучестью. Последнее достигается передачей всей информации (управляющей и обрабатываемой) по принципу близкодействия, т.е. от элемента к элементу. Повышенная живучесть такой решетки иллюстрируется следующими примерами.

Пусть решетка на рис. 3а выполнена на базе элемента, который может принимать и передавать информацию по двум направлениям (рис. 3б). Подсчитаем полное число путей, с помощью которых можно передать информацию из элемента I с координатами (0,0) в элемент 2 с координатами (m , n). Для этого закодируем направления передачи информации элементом в двоичном алфавите: горизонтальному направлению поставим в соответствие цифру 0, а вертикальному – I. Тогда каждый путь, соединяющий элемент I с элементом 2, можно закодировать последовательностью из m нулей и n единиц. Полное число путей определяется числом перестановок из m нулей и n единиц и равно

$$N = \frac{(m+n)!}{m!\,n!}\ .$$

Если каждый элемент решетки на рис. 3а обладает свойством



а) в)

Рис. 3

обмена информацией со всеми соседними (рис. 3в), то число путей передачи информации из элемента 3 с координатами (i , j) в элемент 4 с координатами ($i+m$, $j+n$) за T тактов (считается, что передача информации от элемента к соседнему осуществляется за один такт) составит [4] :

$$N = \frac{[T!]^2}{S!\,K!\,(T-S!)!\,(T-K)!}\ ,$$

где

$$T = m + n + 2K\ ;$$
$$S = m + K\ ,\quad K = 0, 1, 2, \ldots$$

Третьим достоинством решетки, использующей способ кодирования по приращениям, является то, что для управления структурой не требуются мощные усилители формирователи, так как все сигналы передаются по принципу близкодействия. Это дает возможность осуществлять настройку (перестройку) элементов решетки с помощью автомата, реализованного в ней самой .

Для обеспечения этой возможности нужно часть логических выходов устройства соединить с частью его управляющих входов. Например, часть выходов логической решетки нужно соединить с управляющими входами элементов, расположенных на внешней стороне (грани) решетки.

Доказательство возможности настройки (перестройки) решетки с помощью автомата, реализованного в самой решетке, непосредственно следует из того факта, что настройка (перестройка) решетки может осуществляться по одной цепочке.

Однородное устройство, имеющее каналы для обмена информацией между логической и управляющей решетками, представляет собой универсальную вычислительную систему с программируемой структурой.

Относительно быстродействия устройств с переменной структурой настройки отметим следующее: количество тактов, требуемых для настройки однородного устройства, содержащего V элементов, зависит от числа одновременно настраиваемых цепочек элементов и составляет:

$$T = \frac{V}{p}\quad (\,p = 1, 2, 3, \ldots, 2\sqrt[z]{V}\,)\ ,$$

где z – размерность решетки.

3°. Рассмотрим применение многомерного регистра сдвига для моделирования поведения животных в лабиринтах [6] . Блок-схема машины для решения лабиринтных задач изображена на рисунке 4, где цифрами обозначены: 1 – двумерный регистр сдвига, 2 – управляющая головка, предназначенная для выбора направления передачи информации в один из соседних элементов, а также для выработки "развертки опроса" состояний соседних элементов; 3 – читающая головка, предназначенная для ввода результата работы "развертки опроса" в блок 4 (функции читающей головки выполняет сигнал "единица", записанный в один из элементов двумерного регистра сдвига), 4 – управляющее устройство, предназначенное для анализа ситуаций и выбора стратегии. Пока нет информации о конкретной геометрической конфигурации, очевидно, что из элемента (i , j) можно попасть в каждый из четырех соседних: ($i-1$, j); ($i+1$, j); (i , $j-1$); (i , $j+1$). Переход автомата в один из соседних элементов будет определяться способом выбора направления передачи возбуждения в один из соседних элементов.

Если каждое направление передачи возбуждения фиксировать в ячейках памяти элементов решетки, то выбор каждого нового пути в лабиринте будет приводить к увеличению информации о лабиринте. Пусть двумерный регистр сдвига уже настроен на реализацию конкретного лабиринта. Тогда один из алгоритмов первоначального прохождения лабиринта может быть записан в виде:

$$[1 \to \gamma] \overset{1}{\downarrow} \overset{2}{\downarrow} W P \overset{1}{\uparrow} D R \overset{2}{\uparrow},$$

где $[1 \to \gamma]$ – оператор установки начальных значений (оператор записи "единицы" в элемент двумерного регистра сдвига, соответствующий входу в лабиринт); W – оператор выработки "развертки опроса" (в результате работы "развертки опроса" автомат с помощью читающей головки получает информацию о состояниях соседних элементов); P – оператор проверки возможности передачи возбуждения в соседний элемент; R – оператор проверки окончания прохождения лабиринта; D – оператор перемещения автомата в смежную клетку.

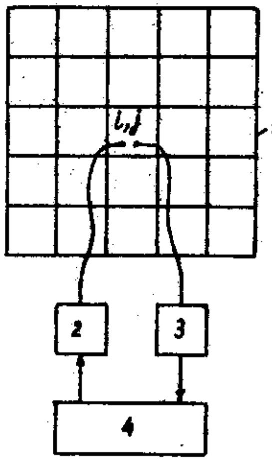


Рис. 4

Функционирование автомата в целом будет определяться видом оператора W, т.е. правилом выбора направления передач и возбуждения в соседний элемент. Некоторые операторы W описаны в работе [6]. Например, алгоритм ускоренного прохождения лабиринта после его изучения может быть сведен к исключению сторон элемента, которые автомат не переходил, а также тех сторон элемента, которые были пройдены в двух направлениях.

По приведенной схеме может осуществляться и автоматическое считывание графической информации, записанной в ячейках памяти элементов решетки, путем автоматического отслеживания считываемых кривых. Например, для автоматического кодирования и считывания некоторых геометрических конфигураций может быть использован способ обхода лабиринта по правилу "правой руки".

Небезынтересно заметить, что если каждый элемент с переменной структурой настройки снабдить приемником графической информации (например, фотодиодом) и источником светового излучения (например, излучающим диодом), то машина на рис. 4 будет представлять собой машину широкого назначения, которая может

быть использована, во-первых, для решения логических и вычислительных задач, во-вторых, для хранения и обработки графической информации и, в-третьих, для выполнения функций устройства ввода-вывода графической информации. Например, такое устройство может быть использовано в качестве читающего автомата, а также для квантования образа по плоскости с различной степенью квантования (см. рис. 5).

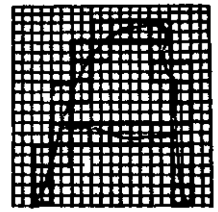


Рис. 5

4°. Отметим, что однородное устройство с переменной структурой настройки позволяет путем элементарных действий (подачей импульсов от генератора импульсов) воспроизвести требуемое количество однотипных автоматов по описанию исходного (по информации настройки одного автомата). Это означает, что автомат, реализованный в устройстве, способен создавать себе подобные, т.е. является самовоспроизводящимся. Процесс воспроизведения автоматов по описанию одного поясняется рис. 6, где кривая, показанная сплошной линией, характеризует исходный автомат, а кривая, показанная пунктирной линией, – автоматы, полученные в процессе размножения исходного. Из рис. 6 видно, что общая кривая характеризует периодический процесс и, следовательно, для ее представления могут быть использованы производящие функции*). Для рис. 6 производящая функция имеет вид

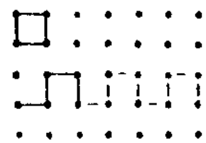


Рис. 6

$$\frac{x + 3x^3}{1 - x^4}$$

и описывает бесконечную последовательность 01030103... .

С помощью производящих функций могут быть описаны структурные схемы любых автоматов. При этом универсальной кривой является такая, которая проходит через все элементы, входящие в схему исходного автомата, и имеет вид волны прямоугольной фор-

*) Функция $f(x)$ называется производящей для последовательности чисел $a_0, a_1, a_2, \ldots, a_n$, если для значений x, лежащих в некоторой области, степенной ряд $a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n +$ $+ \ldots$ сходится к функции $f(x)$.

мы. Амплитуда волны в общем случае определяется размером решетки, требуемой для размещения исходного автомата. Например, если настройка однородного устройства осуществляется по одной цепочке, имеющей форму прямоугольной волны, а для размещения одного автомата требуется решетка размером $A \times B$ элементов, то амплитуда волны должна составлять A (или B).

Геометрически процесс воспроизведения автоматов может быть проиллюстрирован следующим образом. Пусть настройка решетки на схему исходного автомата была произведена по одной цепочке элементов, например, по цепочке I,2,..., α на рис. 7,а, а в последний элемент цепочки был записан код выбора направления следующего ($\alpha+1$) элемента. На рис. 7,б,в показаны цепочки в моменты времени $T+1$ и $T+2$. В момент времени T в решетке имелась только одна (начальная) цепочка $1 \div \alpha$. В момент $T+1$

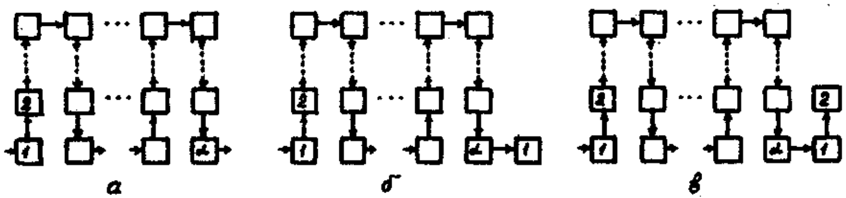


Рис. 7

последним элементом цепочки стал первый элемент исходной цепочки, а в момент $T+2$ – второй элемент. Вообще, в каждый дискретный момент времени длина исходной цепочки увеличивается на один элемент и в момент $T+z$ общее число цепочек будет равно $\frac{z}{\alpha}+1$.

При этом в результирующей цепочке I,2,.., α, I,2,...,α, I,2,... одноименные элементы содержат одну и ту же информацию. Например, последний элемент цепочки I,2,...,α,I на рис. 7, б является копией первого. Самовоспроизведение со стиранием показано на рис. 8.

Из сказанного следует, что количество двоичных разрядов для записи ряда однотипных автоматов, составляет:

$$H_2 = H_1 + \log_2 N_1 = \alpha_1(\log_2 k + \log_2 q) + \log_2 N_1,$$

где H_1 определяет количество информации, требуемой для записи одного (исходного) автомата, а $\log_2 N_1$ – информацию, необходимую для получения N_1 автоматов. При этом, если N_1 –чис-

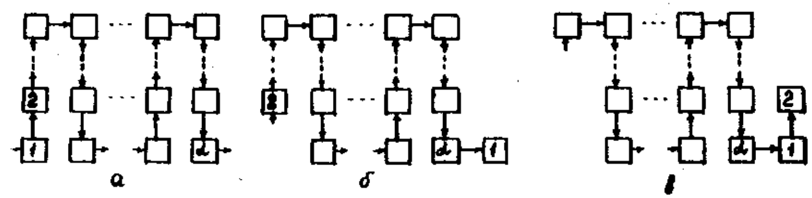


Рис. 8

ло неопределенное, то $\log_2 N_1 = 0$, т.е. для получения автоматов, количество которых точно не определено, не требуется информации; количество автоматов, создающихся в процессе воспроизведения, определяется только временем и размерами решетки.

Если и исходный автомат можно представить в виде ряда однотипных автоматов, то:

$$H_1 = H_0 + \log_2 N_0 = \alpha_0(\log_2 k + \log_2 q) + \log_2 N_0 \text{ бит.}$$

Таким образом, возможность воспроизведения автоматов по описанию одного автомата позволяет упростить программирование и сократить количество информации, требуемой для настройки однородного устройства на эти автоматы. Например, количество информации, требуемой для настройки однородного устройства на схему универсального ($n+2^n, 1$) – полюсника, реализующего любую функцию алгебры логики от n переменных, составляет:

$$H = (2^0 + 2^1 + 2^2 + \ldots + 2^{n-1})(\log_2 k + \log_2 q) + 1 + 2 + 3 + \ldots + n =$$

$$= 2^n(\log_2 k + \log_2 q) + \frac{n(n+1)}{2} \text{ бит}$$

и $H \approx c \cdot 2^n$ при больших n, где c определяется характеристиками одного элемента решетки.

Возможность воспроизведения автоматов является важным качеством устройства с настраиваемой структурой, так как многие практические задачи могут быть решены путем массового распараллеливания вычислительного процесса, для реализации которого требуется большое количество одинаковых вычислительных устройств.

З а к л ю ч е н и е

Применение однородных устройств с настраиваемой структурой в качестве основы для построения специализированной аппаратуры дискретной переработки информации будет экономически целесообразным при достаточно высокой живучести однородного устройства. Это требование вытекает из того, что при современном уровне обработки пластин получить полностью работоспособную решетку с большим количеством элементов практически невозможно. Так, вероятность того, что на пластине будет хотя бы один элемент решетки неисправен, равна

$$p = 1 - e^{-k \Delta S N C},$$

где ΔS - число дефектов, приходящихся на I мм^2 площади пластины; k - площадь одного элемента решетки; N - число элементов решетки на пластине; C - коэффициент поражаемости элемента решетки.

Очевидно, для того чтобы технико-экономические показатели однородного устройства были высокими, необходимо, чтобы некоторое число любых неисправностей в решетке не приводило к потере работоспособности всего однородного устройства.

Л и т е р а т у р а

I. Э.В. ЕВРЕИНОВ, Ю.Г. КОСАРЕВ. Однородные универсальные вычислительные системы высокой производительности. "Наука", СО, Новосибирск, 1966.

2. М. АРБИБ. Мозг, машина и математика. Изд-во "Наука", 1968.

3. Е.И. БЕЛЯЕВ, А.И. МИШИН, В.Г. ХРУЩЕВ. Элементы вычислительной и запоминающей среды с переменной структурой настройки на МОП-транзисторах. -Вычислительные системы, Новосибирск, "Наука" СО, 1969, вып. 33.

4. В.Г. ПОЛЯКОВ. Следящая развертка и сокращенные описания изображения. Современные проблемы машинного анализа биологических структур, "Наука", М., 1970.

5. Н.Я. ВИЛЕНКИН. Комбинаторика, Изд-во "Наука", 1969.

6. М.Г. ГААЗЕ-РАПОПОРТ. Автоматы и живые организмы, 1961.

Поступила в редакцию
16. У. 1970 г.