

УДК 681.3.001

ПРЕОБРАЗОВАНИЕ МИКРОПРОГРАММ ПРИ ВЛОЖЕНИИ ИХ  
В ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ МАТРИЦЫ

С.М.Ачасова

§1. Постановка задачи

Для реализации управляющих блоков вычислительных устройств широко используются программируемые логические матрицы. В них вкладываются микропрограммы, дешифраторы микрокоманд, преобразователи кодов, схемы принятия решений и т.д. При этом множество логических условий рассматривается как система булевых функций, представление которых принадлежит классу д.н.ф.

С точки зрения экономии оборудования наилучшим представлением системы булевых функций считается такое, которое позволяет вложить систему в программируемую логическую матрицу, имеющую минимальную площадь по сравнению со всеми другими матрицами, реализующими эту систему. При неизменном числе столбцов, равном суммарному числу входов и выходов, площадь программируемой логической матрицы изменяется с изменением числа строк. В связи с этим встает задача сжатия логических условий, для решения которой в данной статье предлагается эвристический алгоритм. В алгоритме используется представление множества совокупностей интервалов булева пространства в виде алгебраической структуры, а операции над интервалами производятся в рамках алгебры разбиений на множестве двоичных наборов [1].

Предлагаемый алгоритм, его модификация и алгоритмы, описанные в [1], объединены одной целью - проектирования программируемых логических матриц и одним математическим аппаратом.

Обозначим через  $F(X)$  систему булевых функций  $\{f_1(X), f_2(X), \dots, f_n(X)\}$ , заданных на множестве переменных  $X = \{x_1, \dots, x_n\}$ . Используя терминологию из [2],  $F$ -интервалом будем на-

зывать пару  $(u, v)$ , где  $u$  — интервал произвольного ранга булева пространства,  $v$  — подмножество функций системы  $F(X)$ , которые равны единице на этом интервале. Будем считать, что множество  $F$  — интервалов реализует систему  $F(X)$ , если оно включает в себя конституенты единицы всех функций системы  $F(X)$ .

Множество  $F$  — интервалов, реализующее систему  $F(X)$  и имеющее наименьшую мощность по сравнению со всеми другими множествами  $F$  — интервалов, реализующими систему  $F(X)$ , называется кратчайшим.

Считаем, что система  $F(X)$  задана множеством  $\Psi = \{F_1, \dots, F_p\}$   $F$ -интервалов, компоненты которых  $u$  являются интервалами ранга  $n$  ( $n$ -интервалами) булева пространства. Множество  $I^* = I \setminus I^{10}$ , где  $I$  — множество всех  $n$ -интервалов  $n$ -мерного булева пространства,  $I^{10}$  — множество компонент  $u$  и всех  $F$ -интервалов, принадлежащих множеству  $\Psi$ , является общей для всех функций системы  $F(X)$  областью неопределенных значений.

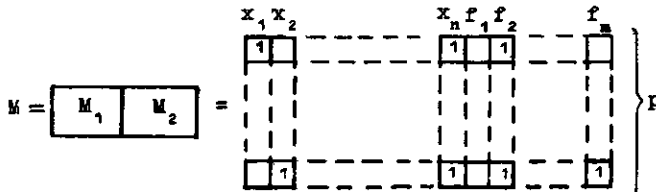


Рис. I

Такому заданию  $F(X)$  соответствует матричное представление системы, приведенное на рис. I. Матрица  $M$ , элементы которой могут иметь значения из множества  $\{0, 1\}$ , составлена из двух матриц  $M_1$  и  $M_2$ . В  $M_1$  записаны компоненты  $u$  и всех  $F$ -интервалов из множества  $\Psi$ , в  $M_2$  — соответствующие им характеристики  $v$ .

Задача преобразования  $F(X)$  состоит в том, чтобы по исходному множеству  $\Psi$  построить множество  $F$ -интервалов  $\Psi_{\approx}$ , реализующее  $F(X)$ , меньшее по мощности, чем  $\Psi$ , и, может быть, близкое к кратчайшему.

## §2. Эвристики в алгоритме преобразования $F(X)$

Обозначим через  $M_2$  матрицу, полученную из  $M_2$  вычеркиванием множества  $\alpha_1$  столбцов, соответствующих нулевым элементам  $i$ -й строки, и множества  $\beta_1^0$  строк, имеющих хотя бы один нуль на пересечении с оставшимися столбцами, а также множества  $\beta_1^*$  строк, имеющих на пересечении с оставшимися столбцами одна звездочка, т.е. без различия значения функций. Звездочки будут появляться в  $M_2$  в процессе работы алгоритма. Множество  $n$ -интервалов из  $M_1$ , соответствующих строкам  $M_2$ , объявляем областью единичных значений  $\hat{I}^1$  левой функции  $\hat{f}$ . Область нулевых значений  $\hat{I}^0$  составит множество  $n$ -интервалов, соответствующее  $\beta_1^0$ , и наконец,  $\hat{I}^* = \Gamma \setminus (\hat{I}^1 \cup \hat{I}^0)$ .

В алгоритме преобразования системы булевых функций многократно выполняются две основные процедуры:

- 1) определение функции  $\hat{f}$ ,
- 2) построение покрытия множества  $n$ -интервалов  $\hat{I}^1$  возможно меньшим числом максимальных интервалов.

Найденные во второй процедуре максимальные интервалы вместе с подмножеством функций, определяемых множеством  $\alpha_1$ , составят компоненты  $u$  и  $v$   $F$ -интервалов, включаемых в решение  $\psi_\infty$ .

Определение функции  $\hat{f}$  естественно начать со строк из  $M_2$ , имеющих наименьшее количество единиц.

Вслед за заданием  $\hat{f}$  решаем вторую задачу, т.е. строим покрытие множества  $\hat{I}^1$  наименьшим числом максимальных интервалов. Эта задача в свою очередь раздваивается. Во-первых, возникает вопрос, каким образом найти максимальные интервалы, об этом пойдет речь в следующей части статьи, во-вторых, какие из них предпочесть другим.

При построении покрытия в его элементы требуется включить все  $n$ -интервалы, соответствующие строкам из  $M_2$ , равным  $i$ -й, и как можно больше остальных  $n$ -интервалов из  $\hat{I}^1$ . В согласии с этим будем рассматривать множество  $\hat{I}^1$  как объединение двух непересекающихся множеств  $\hat{I}^{11}$  и  $\hat{I}^{1*}$ . В  $\hat{I}^{11}$  войдут все  $n$ -интервалы, соответствующие строкам из  $M_2$ , равным  $i$ -й, в  $\hat{I}^{1*}$  войдут остальные  $n$ -интервалы из  $\hat{I}^1$ . Таким образом, мы будем строить покрытие множества  $\hat{I}^{11}$ , стараясь включить в него как можно больше элементов из  $\hat{I}^{1*}$ , и ни один элемент покрытия не будет включать в себя одни лишь  $n$ -интервалы из  $\hat{I}^{1*}$ .

Покрытие множества  $\hat{I}^{11}$  строим следующим образом. Один из  $n$ -интервалов, например,  $u \in \hat{I}^{11}$ , назначаем порождающим. Из всех

максимальных интервалов, включающих в себя  $u$ , выберем  $u_1$ , который содержит наибольшее число элементов из  $\hat{I}^{11}$ . Интервал  $u_1$  включаем в покрытие. Затем все  $n$ -интервалы из  $\hat{I}^1$ , оказавшиеся покрытыми, переносим в область  $\hat{I}^*$ , и в согласии с этим все единицы в  $M_2$ , соответствующие покрытым  $n$ -интервалам, заменяем на звездочки. Если  $\hat{I}^{11}$  оказывается пусто, то задаем новую рабочую функцию, иначе назначаем новый порождающий  $n$ -интервал и т.д.

Таким образом, включение на каждом шаге алгоритма в решение  $F$ -интервала, характеристика которого  $v$  покрывает максимально возможное число единиц в  $M_2$ , позволяет надеяться на то, что преобразованная система будет более или менее близка к кратчайшей форме.

### §3. Множество совокупностей интервалов булева пространства как структура

Представление совокупностей интервалов булева пространства в виде структуры и основные свойства такого представления даны в [1]. В этой части статьи кратко излагаются общие положения из [1] и перечисляются новые свойства представления, необходимые для обоснования алгоритма преобразования системы  $F(X)$ .

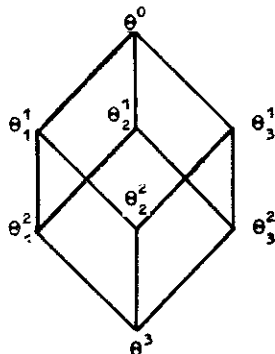
На множестве  $n$ -интервалов булева пространства  $I = \{I_1, \dots, I_{2^n}\}$  можно построить множество разбиений

$$\theta = \{\theta_1^v\}, \quad v = 0, 1, \dots, n, \quad i = 1, \dots, C_n^v.$$

Разбиение  $\theta_1^v$  содержит  $2^{n-v}$  подмножеств (блоков) по  $2^v$   $n$ -интервалов в каждом. При этом блок разбиения является интервалом ранга  $n-v$   $n$ -мерного булева пространства. Множество  $\theta$  является частично упорядоченным. Порядок, зафиксированный на нем, определяется отношением включения каждого блока меньшего разбиения, по крайней мере, в один блок большего разбиения. Частично упорядоченное множество  $\theta$  является структурой, так как всякое его двухэлементное подмножество имеет как точку верхнюю, так и точку нижнюю грани [3]:  $a+b = \text{sup}\{a,b\}$ ,  $a \cdot b = \text{inf}\{a,b\}$ ,  $a, b \in \theta$ . Блоки разбиения—суммы равны объединениям тех блоков слагаемых разбиений, пересечение которых непусто. В разбиении—произведении блоками являются непустые пересечения блоков перемножаемых разбиений.

ний. Для определенных таким образом операций сложения и умножения на множестве  $\Theta$  выполняются все законы булевой алгебры.

Будем говорить, что подмножества  $\Theta^v \subset \Theta$  ( $v=0, 1, \dots, n$ ), содержащие разбиения на одинаковое число блоков, составляют  $v$ -й уровень структуры. Блоки разбиения  $v$ -го уровня - это все возможные интервалы ранга  $n-v$   $n$ -мерного булева пространства. Для обоснования алгоритма мы будем использовать тот факт, что из разбиений первого уровня структуры  $\Theta^1 = \{\Theta_1^1, \dots, \Theta_n^1\}$ , т.е. из разбиений на пары  $n$ -интервалов путем сложения их, можно получить разбиения на любые более крупные подмножества  $n$ -интервалов.



$$\Theta^0 = \{\overline{0}; \overline{1}; \overline{2}; \overline{3}; \overline{4}; \overline{5}; \overline{6}; \overline{7}\}.$$

$$\Theta_1^1 = \{\overline{0}, \overline{1}; \overline{2}, \overline{3}; \overline{4}, \overline{5}; \overline{6}, \overline{7}\},$$

$$\Theta_2^1 = \{\overline{0}, \overline{2}; \overline{1}, \overline{3}; \overline{4}, \overline{6}; \overline{5}, \overline{7}\},$$

$$\Theta_3^1 = \{\overline{0}, \overline{4}; \overline{1}, \overline{5}; \overline{2}, \overline{6}; \overline{3}, \overline{7}\}.$$

$$\Theta_1^2 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}; \overline{4}, \overline{5}, \overline{6}, \overline{7}\},$$

$$\Theta_2^2 = \{\overline{0}, \overline{1}, \overline{4}, \overline{5}; \overline{2}, \overline{3}, \overline{6}, \overline{7}\},$$

$$\Theta_3^2 = \{\overline{0}, \overline{2}, \overline{4}, \overline{6}; \overline{1}, \overline{3}, \overline{5}, \overline{7}\}.$$

$$\Theta^3 = \{\overline{0}, \overline{1}, \overline{2}, \overline{3}, \overline{4}, \overline{5}, \overline{6}, \overline{7}\}.$$

Рис. 2

Графом структуры  $\Theta$ , эквивалентной булевой алгебре, является  $n$ -мерный куб. На рис.2 дан трехмерный куб, интерпретирующий структуру  $\Theta = \{\Theta^0, \Theta^1, \Theta^2, \Theta^3\}$ . В записи разбиений в качестве имен  $n$ -интервалов используются десятичные аналоги их двоичных представлений.

В связи с задачей покрытия множества  $\Gamma^1$  максимальными интервалами, нас будут интересовать только те блоки разбиений в структуре  $\Theta$ , которые не содержат  $n$ -интервалов из области  $\Gamma^0$ , а состоят только из  $n$ -интервалов, принадлежащих множеству  $\Gamma^1 \cup \Gamma^*$ . После удаления из разбиений структуры  $\Theta$  блоков, содержащих хотя бы один элемент множества  $\Gamma^0$ , мы получим новое множество  $\tilde{\Theta}$  разбиений на множестве  $\Gamma^1 \cup \Gamma^*$ . Из куба, интерпретирующего структуру  $\Theta$ , выпадут вершины, которые представляют разбиения, оказавшиеся пустыми. Некоторое подмножество вершин куба окажется соответствующим множеству  $T$  наибольших разбиений. Блоки таких разбиений являются

максимальными интервалами на множестве  $\Gamma^1 \cup \Gamma^*$ . Из них будет состояться покрытие множества  $\Gamma^1$ .

Каждый блок любого разбиения  $\Theta_1^v$  из множества  $T$ , являющийся интервалом ранга  $n-v$   $n$ -мерного булева пространства, можно рассматривать теперь как единицу структуры  $\Theta_v$ , которой является множество разбиений множества  $n$ -интервалов, составляющих блок разбиения  $\Theta_1^v$ . Ни один блок, входящий в  $\Theta_\rho^v$  ( $\rho = 1, \dots, v$ ), не был отброшен при переходе от  $\Theta$  к  $\Theta$ , так как не содержал ни одного элемента из  $\Gamma^0$ . Разбиения, принадлежащие структурам  $\Theta_\mu$  и  $\Theta_\nu$ , могут иметь общие блоки. Каждая структура  $\Theta_\nu$ , как и  $\Theta$ , эквивалентна булевой алгебре.

Таким образом, структура  $\Theta$  превратилась в некоторое множество структур с меньшим числом уровней, и таких структур ровно столько, сколько максимальных интервалов можно выделить на множестве  $\Gamma^1 \cup \Gamma^*$ .

Это множество структур будем называть "усеченной структурой". Вычисляя максимальный интервал, мы каждый раз имеем дело с одной структурой в рамках усеченной структуры.

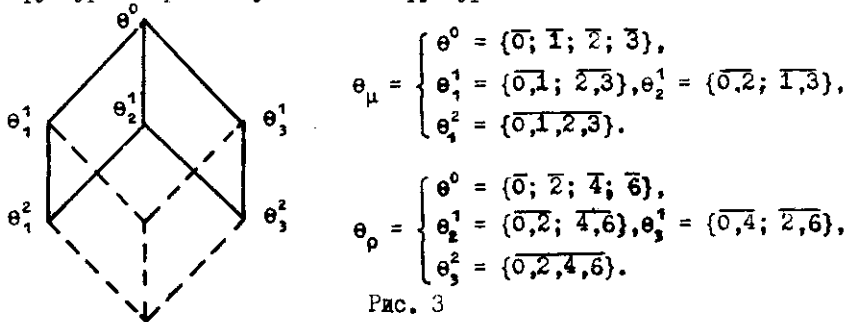


Рис. 3

Усеченная структура для трехмерного булева пространства и подмножеств  $\Gamma^1 \cup \Gamma^* = \{0,1,2,3,4,6\}$ ,  $\Gamma^0 = \{5,7\}$  дана на рис.3. Две структуры  $\Theta_\mu$  и  $\Theta_\rho$  составляют усеченную структуру.

Удобной формой представления структуры  $\Theta$  для разработки машинных алгоритмов является матрица-строка  $S = \|s_i\|$ ,  $i=1, \dots, 2^n$ . Каждый элемент  $s_i$  есть  $n$ -интервал, а порядок записи  $n$ -интервалов в  $S$  таков. В качестве  $s_1$  можно взять любой  $n$ -интервал. Порядок заполнения остальных  $2^n - 1$  клеток матрицы  $S$  носит рекурсивный характер. При каждой рекурсии число заполненных клеток удваивается. Первая рекурсия - это определение  $s_2$ , которым является  $n$ -интервал,

соседний к  $s_1$  по первой координате. Вторая рекурсия - определение  $s_2$  и  $s_4$ , как  $\alpha$ -интервалов, соседних к  $s_1$  и  $s_2$ , по второй координате и т.д. по формуле

$$s_i = s_{i-2^{k-1}} + (-1)^{x_{k-1}} \cdot 2^{k-1}, \quad (1)$$

где  $k-1$  - номер самого старшего разряда, равного единице, в двоичном представлении числа  $i$ ;  $k$  - номер рекурсии, т.е. номер координаты, по которой ищутся соседние  $\alpha$ -интервалы;  $x_{k-1}$  - значение  $(k-1)$ -го разряда в двоичном представлении  $\alpha$ -интервала, сосед которого по  $k$ -й координате вычисляется.

ПРИМЕР 1. Построим матрицу  $S$  для  $n = 4$  и  $s_1 = 2$ :

$$S = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|} \hline 2 & 3 & 0 & 1 & 6 & 7 & 4 & 5 & 10 & 11 & 8 & 9 & 14 & 15 & 12 & 13 \\ \hline \end{array}$$

Матрица  $S$  представляет структуру в том смысле, что, выбирая из нее элементы по формуле

$$\theta_k^1 = \{s_{i_1}, s_{(i_1+2^{k-1}) \cdot 1}\}, \quad i_1=1, 2, \dots, 2^{k-1}; \quad 1=1, 2, \dots, 2^{n-k}, \quad (2)$$

можно получить все разбиения  $\theta^1 = \{\theta_1^1, \dots, \theta_n^1\}$  первого уровня структуры  $\theta$ . Блоки этих разбиений объединяют  $\alpha$ -интервалы, соседние по координате, совпадающей с номером разбиения. Формула (2) непосредственно следует из способа построения матрицы  $S$ .

Перечислим свойства матрицы  $S$  как представления структуры  $\theta$ , а также усеченной структуры  $\tilde{\theta}$ .

1. Из способа построения матрицы  $S$ , т.е. из формулы (1) следует, что номер  $i$  элемента  $s_i$  матрицы  $S$  связан с цепочкой номеров координат, по которым рекурсивно определялись соседи, начиная с  $s_1$  следующим соотношением:

$$i_{10} = (\varepsilon_n \varepsilon_{n-1} \dots \varepsilon_1)_2 + 1. \quad (3)$$

Единичное или нулевое значение  $j$ -го разряда двоичного числа  $\varepsilon = \varepsilon_n \dots \varepsilon_1$  показывает, вычислялся ли сосед по  $j$ -й координате в цепочке вычислений  $i$ -го элемента матрицы  $S$  по формуле (1).

2. Из матрицы  $S$  можно выделить блок, принадлежащий разбиению  $\nu$ -го уровня структуры и содержащий  $\alpha$ -интервал  $s_1$ . Пусть разбиение из  $\theta^\nu$  равно сумме  $\nu$  разбиений первого уровня с номерами  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_\nu$ . В искомый блок войдут  $2^\nu$   $\alpha$ -интервалов. Номера элементов матрицы  $S$ , составляющих этот блок, найдутся по формуле

(3), где в разрядах двоичного числа  $g$  с номерами, не совпадающими с  $\varepsilon_1, \dots, \varepsilon_n$ , будут стоять нули, а  $h = \varepsilon_{q_1}, \dots, \varepsilon_{q_j}$  будет меняться от  $0 \dots 0$  до  $1 \dots 1$ .

Первые два свойства относятся как к структуре  $\Theta$ , так и к  $\tilde{\Theta}$ , остальные — только к усеченной структуре.

3. Подмножество разбиений  $R_i \in \Theta^1$  будем называть совокупностью несовместимых разбиений для некоторого назначенного  $n$ -интервала  $I_k$ , если в результате сложения разбиений этого подмножества в блоке, содержащем  $I_k$ , окажется хотя бы один  $n$ -интервал из  $I^0$ .

Заметим, что совокупность несовместимых разбиений может содержать только одно разбиение  $\theta_j^1 \in \Theta^1$ . Это тот случай, когда  $n$ -интервал, соседний назначенному, по  $j$ -й координате принадлежит множеству  $I^0$ .

В процессе построения матрицы  $Z$  можно найти совокупности несовместимых разбиений для  $n$ -интервала  $\varepsilon_1$ . По номеру элемента  $\varepsilon_i \in I^0$  в соответствии с формулой (3) определяем номера несовместимых разбиений, принадлежащих одной совокупности, как номера разрядов числа  $g$ , равных единице.

4. Определив некоторую совокупность  $R_i$ , мы в дальнейшем не будем рассматривать никакую другую  $R_j \supset R_i$ . Для этого в матрице  $Z$  ставим знак "пусто" в клетки с номерами  $j \geq 1$ , такие, которые должны быть заняты  $n$ -интервалами, входящими вместе с  $\varepsilon_1$  в блоки разбиений, равных всевозможным суммам, содержащим в качестве слагаемых несовместимые разбиения. При этом вычисления по формуле (3)  $\varepsilon_r = 1$ , если  $\theta_r^1 \in R_i$ , а совокупность остальных разрядов принимает всевозможные значения от  $0 \dots 0$  до  $1 \dots 1$ . Таким образом, мы исключаем из рассмотрения интервалы всех рангов, не вошедшие в усеченную структуру, отвечающую  $n$ -интервалу, назначенному в качестве  $\varepsilon_1$ .

ЗАМЕЧАНИЕ. Если совокупность несовместимых разбиений содержит одно разбиение  $\theta_j^1 \in \Theta^1$ , то очевидно, что согласно формуле (3) мы поставим знаки "пусто" в половину клеток матрицы  $Z$ . Если есть еще одна одноэлементная совокупность несовместимых разбиений, то нужно поставить знак "пусто" в половину оставшихся клеток и т.д. Видно, что наличие таких одноэлементных совокупностей сильно уменьшает количество вычислений при построении матрицы  $Z$ . Отсюда следует, что с целью уменьшения количества вычислений выгодно брать в качестве  $\varepsilon_1$   $n$ -интервал, имеющий наименьшее количество соседей среди всех  $n$ -интервалов, претендующих быть элементом  $\varepsilon_1$ .



Это замечание будет использовано в алгоритме преобразования для ускорения сходимости алгоритма.

5. Информацию о множестве совокупностей несовместимых разбиений  $R = \{R_i\}$  ( $i = 1, \dots, z$ ) помещаем в матрицу  $N$  размером  $z \cdot n$ . Столбцы этой матрицы соответствуют элементам множества  $\theta^1 = \{\theta_1^1, \dots, \dots, \theta_n^1\}$ , а число строк равно числу найденных совокупностей несовместимых разбиений для  $n$ -интервала  $\alpha_1$ . Элемент  $n_{ij} = 1$ , если разбиение  $\theta_j^1$  входит в  $i$ -ю совокупность несовместимых разбиений,  $n_{ij} = 0$  - в противном случае.

Из матрицы  $N$  мы найдем подмножества  $\bar{P}_k \subset \theta^1$  такие, что их дополнения  $P_k = \theta^1 \setminus \bar{P}_k$  не содержат ни одной совокупности несовместимых разбиений и определяют все максимальные интервалы, содержащие  $\alpha_1$ . Подмножества  $\bar{P}_k \subset \theta^1$  минимально возможной мощности определяем, отыскивая все безыбыточные покрытия матрицы  $N$  столбцами. Для нахождения безыбыточных покрытий используем алгоритм из [4].

ПРИМЕР 2. Построим матрицы  $S$  и  $N$  усеченной структуры для  $n$ -интервала  $\gamma$  и множества  $I^0 = \{0, 1, 2, 12, 14, 18\}$  при  $n = 5$ :

$$S = \begin{array}{|cccccc|cccc|cccc|} \hline 7 & 6 & 5 & 4 & 3 & || & 15 & 13 & 11 & || & 23 & 22 & 21 & 20 & | & 9 & || & 31 & | & 29 & | & 27 & || \\ \hline \end{array}$$

$$N = \begin{array}{|cc|cc|} \hline 1 & 1 & & \\ \hline & 1 & 1 & \\ \hline 1 & & & 1 \\ \hline \end{array}$$

Безыбыточные покрытия матрицы  $N$ :  $\{1, 3\}, \{1, 2\}, \{3, 4\}$ ; их дополнения до  $\{1, 2, 3, 4, 5\}$ :  $\{2, 4, 5\}, \{3, 4, 5\}, \{1, 2, 5\}$ .

#### §4. Основной алгоритм преобразования $F(X)$

Преобразуемая система булевых функций задается матрицей  $M$  размером  $p \cdot (n+m)$ . Каждый элемент этой матрицы равен единице или нулю. Результатом выполнения алгоритма является матрица  $\Phi = \begin{array}{|c|c|} \hline \Phi_1 & \Phi_2 \\ \hline \end{array}$  размером  $p \cdot (2n+m)$ , строки которой представляют множество

$F$ -интервалов, реализующее систему  $F(X)$ . Элементы матрицы  $\Phi_1$  могут принимать значения из множества  $\{1, 0^*\}$ , а элементы  $\Phi_2$  - из  $\{1, 0\}$ . Считаем, что в начале выполнения алгоритма матрица  $\Phi$  имеет пустое множество строк.

Основной алгоритм преобразования  $F(X)$ , использующий эвристики, о которых шла речь в §2, и свойства усеченной структуры, носит итеративный характер. Считаем, что, выполнив процедуры алгоритма с  $I$  по  $\gamma$ , мы получим новую систему  $F^*(X)$ , которая от первоначальной отличается областями единичных значений функций системы. Изменение вышеупомянутых областей в процессе выполнения итеративно-

го шага фиксируется в матрице  $\tilde{M}_2$ , которая первоначально равна матрице  $M_2$ . По выполнении итеративного шага в  $\tilde{M}_2$  вносятся все изменения, которые претерпела  $M_2$ . Применяем затем процедуры 1-7 к новой системе. В результате  $\nu$ -кратного выполнения этих процедур алгоритма получаем матрицу  $\Phi_\nu$  с числом строк  $r_{\Phi_\nu}$ , и в матрице  $M$  оказывается  $r_\nu$  строк непокрытых построенными  $F$ -интервалами. На этом этапе вычислительного процесса система  $F(x)$  может быть реализована матрицей  $\hat{F}_\nu$ , имеющей  $r_{\Phi_\nu} + r_\nu$  строк. Итеративный вычислительный процесс считается закончившимся по выполнению одного из двух условий:

1) ни один из оставшихся непокрытыми  $n$ -интервалов не склеивается с каким-либо другим непокрытым  $n$ -интервалом для всех рабочих функций  $\hat{f}$ , заданных на  $\nu$ -м шаге итеративного процесса;

2)  $r_{\Phi_\nu} \leq r'$ , где  $r'$  - некоторое наперед заданное число, может быть, равное числу строк стандартной ПЛМ или имеющее другой смысл.

Алгоритм имеет параметр  $\mu = \{1, \dots, m\}$ . Смысл этого параметра состоит в том, что он задает наибольшее число функций системы, для которых ищутся общие интервалы. Изменяя  $\mu$ , можно полнее использовать возможности преобразования для разных подмножеств функций системы  $F(x)$ .

В каждой  $\nu$ -й итерации алгоритма выполняются процедуры.

1. В  $m$  отыскивается строка, содержащая в части  $M_2$  меньше единиц, чем все остальные строки. Пусть это  $i$ -я строка, и она содержит  $m_i$  единиц. Если  $m_i > \mu$ , то выполняется процедура 8. Иначе задается матрица  $M_{21}$  и, тем самым, множества  $\hat{I}^{11}$ ,  $\hat{I}^{1*}$ ,  $\hat{I}^0$ ,  $\hat{f}^*$ , определяющие  $\hat{f}$ .

2. Среди элементов множества  $\hat{I}^{11}$  в согласии с замечанием из §3 отыскиваем  $n$ -интервал, имеющий наименьшее число соседей в рамках функции  $\hat{f}$ . Этот  $n$ -интервал объявляется порождающим и берется в качестве  $\alpha$ , при построении в следующей процедуре матрицы  $S$ .

3. Для порождающего  $n$ -интервала по формуле (1) и в согласии со свойствами 3-5 (стр. 73-74) усеченной структуры строятся матрицы  $S$  и  $K$ .

4. Отыскиваются все избыточные покрытия  $P_k$  матрицы  $N$  и дополнения к ним  $R_k$ .

5. Для каждого дополнения  $R_k$  по формуле (3) и свойству 2 находится максимальный интервал. Если оказалось, что ни в одном мак-

симальном интервале не содержится ни одного элемента из  $\hat{I}^1$ , кроме порождающего, то выполняется процедура 7. В противном случае в качестве компоненты  $\alpha$   $F$ -интервала с характеристикой, определяемой 1-й строкой матрицы  $M_2$ , берется максимальный интервал, который покрывает наибольшее число единиц в  $M_2$ . Сформированный  $F$ -интервал заносится в  $\Phi_\nu$ , а покрытие  $\mu$ -интервалы исключаются из  $\hat{I}^1$ , и за счет них расширяется множество  $\hat{I}^*$ . В  $\hat{M}_2$  покрытые единицы заменяются звездочками.

6. Если  $r_{\Phi_\nu} + p_\nu \leq p'$ , то в  $\Phi$  записываются непокрытые строки из  $M$  и считается, что  $\Phi$  построена, решение получено. В противном случае выполняется процедура 7.

7. Если  $\hat{I}^{11} \neq \emptyset$ , то выполняется процедура 2. В противном случае и при условии, что просмотрены все строки  $M_2$ , содержащие не менее одной и не более  $\mu$  единиц, проверяется первое условие окончания итеративного процесса, т.е. увеличилось ли число строк в  $\Phi_\nu$  по сравнению с  $\Phi_{\nu-1}$ . Если нет, то матрица  $\Phi_\nu$  расширяется за счет непокрытых строк из  $M$ ; таким образом,  $\Phi$  построена, решение получено. В противном случае выполняется процедура 8.

8. Матрица  $M_2$  заменяется на  $\hat{M}_2$ , выполняется следующая  $(\nu+1)$ -я итерация, т.е. происходит переход к процедуре I.

#### §5. Две модификации основного алгоритма

Основной алгоритм (будем называть его алгоритмом А) требует построения матрицы-строки  $S$  размером  $2^n$ . При  $n$ , больших относительно памяти машины, используемой для реализации алгоритма А, для преобразования  $F(x)$  может быть использована модификация В алгоритма А.

А л г о р и т м В отличается от А тем, что в нем вместо  $S$  строится матрица  $S'$  размером  $2 \cdot p$ . Матрица  $S'$  слита по сравнению с  $S$  за счет клеток со знаком "пусто", а также за счет  $n$ -интервалов из множества  $\hat{I}^*$ . Элементы  $a'_{i1}$  ( $i = 1, \dots, p$ ) равны расположенным по порядку в  $S$   $n$ -интервалам из  $\hat{I}^1$ ,  $a'_{2i}$  равны номерам соответствующих элементов матрицы  $S$ .

Построение матрицы  $S'$ , как и  $S$ , ведется по формуле (I) с той лишь разницей, что  $(k-1)$  — это номер самого младшего разряда, равного единице, в двоичном представлении  $i$ . Это соответствует тому, что вычисление по цепочке координат, связывающей  $a'_{i1}$  с  $a'_{i+1}$ , идет не от координат с младшими номерами к старшим, как при пост-

роения  $S$ , а наоборот. Очевидно, что результаты двух способов построения совпадают. Однако, при построении  $S'$  знаки "пусто" не ставятся в ее клетки, и значит не отбрасываются подмножества разбиений из  $\theta^1$ , включающие в себя найденные совокупности несовместимых разбиений. Поэтому, прежде чем вычислять очередной элемент матрицы  $S'$ , нужно проверить не включает ли в себя подмножество разбиений из  $\theta^1$ , соответствующее цепочке координат, уже найденные  $v_j \in V$  ( $j=1, \dots$ ), и отбросить это подмножество в случае положительного исхода проверки. В связи с этим на выполнение алгоритма  $B$  может быть затрачено времени больше по сравнению с алгоритмом  $A$ .

Вторая модификация основного алгоритма - алгоритм  $C$  - отличается от  $A$  и  $B$  тем, что матрица  $S$  не строится вовсе, а сразу находятся совокупности несовместимых разбиений. Делается это так. Порождающий  $n$ -интервал поразрядно складывается по модулю два последовательно с каждым элементом множества  $\hat{I}^0$ . Результат этой операции показывает, по каким координатам не могут быть склеены  $n$ -интервалы, т.е. мы получаем совокупность несовместимых разбиений.

ПРИМЕР 3.  $01100$  порождающий,  $00010 \in \hat{I}^0$ ,  $01100 \oplus 00010 = 01110$ . Результат поразрядного сложения  $n$ -интервалов по модулю два означает, что интервал  $0***0$  включает в себя оба  $n$ -интервала, а из этого в свою очередь следует, что  $\{\theta_2^1, \theta_3^1, \theta_4^1\}$  - есть совокупность несовместимых разбиений.

При построении матрицы  $N$  в алгоритме  $C$  так же, как и в  $B$ , надо выявлять подмножества разбиений первого уровня структуры, покрывающие совокупности несовместимых разбиений, и отбрасывать их.

Относительно того, какой из алгоритмов  $A, B, C$  удобно использовать в зависимости от имеющихся ресурсов времени и памяти ЭВМ, а также от параметров задачи, заметим следующее.

Время выполнения алгоритма  $C$  пропорционально величине  $pr_{\Phi}$ , т.е.  $T_C \sim pr_{\Phi}$ . Эта величина определяется тем, что для построения каждого  $F$ -интервала, включаемого в  $V_{\approx}$ , а всего их  $r_{\Phi}$ , нужно несколько раз просмотреть все строки матрицы  $M$ . Для нахождения каждого  $F$ -интервала, включаемого в  $V_{\approx}$ , по алгоритму  $B$  нужно вычислить самое большое  $2^{n-1}$   $n$ -интервалов, т.е.  $T_B \sim 2^{n-1} r_{\Phi}$ . При  $r \ll 2^n$  имеет место соотношение  $T_C < T_A < T_B$ . В случае, когда  $r \approx 2^n$ ,  $T_A < T_B < T_C$ . Что касается объема памяти, необходимого для выполнения алгоритмов  $A, B$  и  $C$  на ЭВМ, то  $V_C < V_B < V_A$ . Очевидно, что

алгоритм С может работать с системами функций, заданными интервалами любой размерности.

Все три алгоритма могут работать с системами, функции которых имеют несовпадающие области неопределенных значений, т.е. в исходной  $M$  элементы  $M_2$  могут иметь значения из множества  $\{0, 1, \dots\}$ . В этом случае в алгоритм А после шага 5 надо ввести следующую процедуру.

5а. Проверяется возможность расширения характеристики  $\nu$  формируемого  $F$ -интервала для каждого максимального интервала, претендующего стать компонентом  $u$ , за счет функций, равных единице или не определенных на этом интервале. Окончательно, в качестве  $u$  берется тот максимальный интервал, который с соответствующей характеристикой  $\nu$  покрывает наибольшее количество единиц в  $M_2$ .

## §6. Заключение

Умозрительное сравнение алгоритмов В и С с известными алгоритмами преобразования систем функций [2, 5, 6] позволяет сделать следующие выводы \*).

Объем памяти, необходимый для выполнения алгоритма из [2] пропорционален величине  $\frac{p(p+m+n)}{k}$ , в то время как  $v_B \sim \frac{p(m+n)}{k}$ , где  $k$  — разрядность ячейки ЭВМ, реализующей алгоритмы.

При  $p \ll 2^n$  алгоритм В уступает алгоритму из [2] по быстродействию, так как время выполнения алгоритма из [2] пропорционально  $pp_\Phi$ . Однако алгоритм С по быстродействию не хуже алгоритма из [2].

Что касается алгоритма из [6], то, судя по примерам, приведенным в [6], время его выполнения больше времени выполнения алгоритмов В и С, но из-за более тщательного выбора максимальных интервалов, включаемых в решение, алгоритм из [6] даст в некоторых случаях решения лучшего качества, чем В и С. Заметим, однако, что результат преобразования алгоритмом В системы функций, приведенной в качестве примера в [6], оказался не хуже чем в [6].

Относительно способа нахождения максимальных интервалов, изложенного в [5], заметим следующее. Подмножества  $p$ -интервалов, по которым пересекаются максимальные интервалы, в [5] находятся заново каждый раз, в то время как свойства матрицы  $Z$  позволяют вы-

\* ) В [2] нет оценок сложности алгоритма, приведенных здесь. Эти оценки получены автором в результате рассмотрения алгоритма.

числить каждый  $n$ -интервал и проверять его принадлежность множеству  $\hat{I}^0$  только один раз.

Т а б л и ц а

Номер сис-темы	n	m	p	Р <sub>ф</sub> алгоритма В	Р <sub>ф</sub> алгоритма из [2]	Время выполнения алгоритма В	Назначение систем функций
1	10	7	198	119	-	1 40"	микропрограммы преобразова- тель кодов
2	10	7	297	181	-	2 20"	
3	12	8	256	223	-	4 10"	
4	9	3	33	13	12	1 12"	} функции управления
5	8	6	224	19	14	1 03"	
6	9	1	54	10	9	1 07"	

Алгоритм В реализован на ФОРТРАНе в системе "Дубна" для БЭСМ-6. Собственно программа занимает 8700 ячеек памяти. О быстроте действия алгоритма и о результатах его работы в сравнении с алгоритмом из [2] можно судить по данным, приведенным в таблице \*).

Из таблицы видно, что качество решения, полученного алгоритмом В, немного хуже чем алгоритмом из [2]. Дело в том, что в [2] проводится дополнительная коррекция полученного решения, состоящая в том, чтобы найти и отбросить интервалы, покрываемые другими для одной функции или некоторого подмножества функций.

#### Л и т е р а т у р а

1. АЧАСОВА С.М., БАНДМАН О.Л. Проектирование автоматов управления на основе БИС. - В кн.: Автоматизация проектирования в микроэлектронике. Теория. Методы. Алгоритмы. (Вычислительные системы. Вып. 64.) Новосибирск, 1975, с. 26-52.
2. Синтез асинхронных автоматов на ЭВМ. Под ред. А.Д.Закревского. Минск, "Наука и техника", 1975.
3. СКОРНИКОВ Л.А. Элементы теории структур. М., "Наука", 1970.
4. DAS S.R., KHANNA N.S. Clause-Column Table Approach for Generating All the Prime Implicants of Switching Functions. - "IEEE Trans. Computers", 1972, v.C-21, N 11, p.1239-1247.
5. ПРОЗДОВ Е.А. Оптимизация структур цифровых автоматов. М., "Сов. радио", 1975.
6. HONG S.J., CAIN R.G., OSTAPKO D.L. MINI: A Heuristic Approach for Logic Minimization. - "IBM Journal of research and development", 1974, v.18, N 5, p.443-458.

Поступила в ред.-изд.отд.  
31 января 1976 года

\* ) К сожалению, время преобразования алгоритмом из [2] указанных в таблице систем функций автору неизвестно.