АВТОМАТИЗАЦИЯ ПРОЕКТИРОВАНИЯ В МИКРОЭЛЕКТРОНИКЕ.
ТЕОРИЯ, МЕТОДЫ, АЛГОРИТМЫ
(Вычислительные системы)
1978 год Выпуск 77

УДК 681.3.001

ТЕОРЕТИКО-СТРУКТУРНЫЙ МЕТОД КОДИРОВАНИЯ МИКРОКОМАНД

С.М.Ачасова

I. Введение. При проектировании микропрограммного управления ЭВМ в виде программируемых логических матриц в качестве математической модели многовыходной двухуровневой схемы,реализующей систему булевых функций $F(X) = \{f_1(X), \ldots, f_m(X)\}, X = \{x_1, \ldots, x_n\}$ (логические преобразователи конечных автоматов, память микропрограмм, дешифраторы микрокоманд и т.д.),рассматривается (0,1)-матрица $M = \boxed{M_1}\boxed{M_2}$ размером $p(2n + m)$. Такая модель соответствует представлению функций системы $F(X)$ в д.н.ф. [1]. Строки матрицы M_1 соответствуют множеству $\Psi = \{\varphi_1, \ldots, \varphi_p\}$ элементарных конъюнкций, на каждой из которых хотя бы одна функция системы равна единице. В M_1 для каждой переменной $x_i \in X$ отведена пара столбцов для x_i и $\bar{x}_i$. Вхождение переменной в конъюнкцию отмечено единицей в соответствующем столбце матрицы M_1. Элемент m_{ij} матрицы M_2 определяет значение функции f_j на конъюнкции φ_i.

Если матрица M является моделью дешифратора микрокоманд, то в i-й строке M_1 записывается код i-й микрокоманды, а столбцы матрицы M_2 соответствуют микрооперациям, причем набор совместимых микроопераций для каждой микрокоманды определяется единицами в соответствующей строке матрицы M_2.

Одной из основных задач проектирования логических матриц является задача уменьшения числа строк матрицы M путем преобразования системы $F(X)$ в классе д.н.ф. [2,3]. В случае проектирования матрицы, реализующей дешифратор микрокоманд, эту задачу целесообразно решать, начиная с присвоения такого кода множеству микрокоманд Q, который позволил бы получить наименьшую величину матрицы M после всех возможных склеиваний ее строк.

Оригинальный подход к построению такого кода предложен в работе [4]. Он основан на представлении совокупности разбиений множества кодируемых объектов в виде алгебраической структуры. В [5] получены необходимые и достаточные условия для построения разбиений, образующих структуру.

Здесь задачи кодирования и преобразования системы булевых функций на основе теоретико-структурных свойств двоичного кода формулируются как единая задача проектирования логических матриц, а также дается алгоритм кодирования, использующий компактное представление алгебраической структуры.

2. Постановка задачи. В работе [5] показано, что совокупность всевозможных разбиений множества I интервалов наименьшего ранга n-мерного булева пространства (n-интервалов) на блоки мощностью, равной степени двойки, образует структуру $\theta = \{\theta_i^\nu\}$, $\nu = 0,\dots,n$, $i=1,\dots,C_n^\nu$. Каждое подмножество $\theta^\nu = \{\theta_i^\nu\}$, $\nu = const$, содержит разбиения на блоки мощностью 2^ν и называется ν-м уровнем структуры θ. Путем сложения и умножения разбиений [4, 5] одного уровня можно получить все множество θ, это значит, что множество разбиений какого-либо одного уровня, исключая нулевой и n-й, определяет структуру [5]. Совокупность блоков всех разбиений $\theta_i^\nu \in \theta^\nu$ является множеством всевозможных интервалов ранга $(n-\nu)$.

Решение задачи преобразования системы булевых функций $F(X)$ с целью уменьшения мощности множества конъюнкций Ψ, реализующего систему, так или иначе [2,3] связано с отысканием кратчайшего покрытия J множества n-интервалов $\hat{I}^1$, являющегося пересечением областей единичных значений некоторого подмножества функций $\{f_{i_1}, \dots, f_{i_k}\}$. Для построения такого покрытия понадобятся только те блоки $B_j \in \theta_i^\nu$, которые целиком состоят из элементов множества $\hat{I}^1$. После удаления из разбиений всех ненужных блоков мы получим совокупность разбиений $\varphi(\hat{I}^1)$ множества $\hat{I}^1$, названную в [2] усеченной структурой. Причем $\varphi(\hat{I}^1) = \cup\varphi_r$, где φ_r – структура с числом уровней меньшим, чем у θ, и разбиения ее построены на подмножестве n-интервалов $Y_r \subset \hat{I}^1$, которое является максимальным интервалом для $\hat{I}^1$. Таким образом, покрытие J строится из элементов множества $Y(\hat{I}^1) = \{Y_r\}$. Очевидно, что чем крупнее подмножества Y_r, т.е. чем меньше число компонентов составляет усеченную структуру, тем лучше по числу элементов будет покрытие J. А мощность множеств $\varphi(\hat{I}^1)$ и $Y(\hat{I}^1)$ зависит от того, насколько склеиваются n-интервалы из $\hat{I}^1$.

Теперь понятно, что, удачно кодируя множество микрокоманд $Q = \{q_1, \dots, q_p\}$, мы тем самым можем существенно уменьшить число строк матрицы M, представляющей дешифратор микрокоманд.

В работе [5] показано, что двузначный n-разрядный код $\varkappa$ множества объектов (в нашем случае множества микрокоманд Q) определяет n двублочных разбиений множества Q. В один блок j-го разбиения входят все $q_i \in Q$, имеющие в j-м разряде кодирующего слова единицу, в другой блок – все q_i, имеющие нуль в том же разряде. Множество таких разбиений $T = \{\tau_1, \dots, \tau_n\}$ можно рассматривать как $(n-1)$-й уровень структуры Q [5], который, по вышесказанному, определяет всю структуру целиком и, следовательно, мощность конкретной усеченной структуры тоже.

Максимальные подмножества микрокоманд, склейку которых допускает расстановка единиц в M_2, как и в [5], будем называть K-множествами. Каждое K-множество $K_i = \{q_{i_1}, \dots, q_{i_k}\}$ из множества $\Gamma = \{K_1, \dots, K_m\}$ объединяет микрокоманды, которые разрешают выполнение i-й микрооперации и, следовательно, отмечены единицами в i-м столбце матрицы M_2.

Теперь стало ясно, что задача построения кода есть задача синтеза такой структуры θ, в которой разбиения θ_i^{ν} построены на множестве Q, а совокупность усеченных структур для всех возможных пересечений K-множеств из Γ позволяет построить покрытие $\mathcal{P}$ множества Γ, минимальное среди всех возможных покрытий множества Γ подмножествами элементов из Q.

В этом случае, когда попарное пересечение множеств $K_i \in \Gamma$ пусто и выполнено условие [x)]

$$\sum_{i=1}^{m} 2^{]\log|K_i|[} \le 2^n$$

($]a[$ – ближайшее целое, большее a), задача синтеза θ становится наглядной и заключается в построении такой структуры, в которой каждая усеченная структура $\varphi(K_i)$ состоит из единственного компонента. Этим обеспечивается минимальная мощность покрытия $\mathcal{P}$

---

x) Если условие не выполнено, то некоторые из $K_i \in \Gamma$ расчленяются на подмножества K_i' и K_i'' и вместо Γ рассматривается новое множество Γ', которое получено из Γ заменой K_i на K_i' и K_i'', причем $|\Gamma'|$ минимально необходимая для выполнения условия.

множества Г, $|\mathcal{P}| = m$. Такая задача решается просто [5]. При непустом пересечении К-множеств получить точное решение задачи затруднительно из-за необходимости перебора большого числа вариантов. В следующем разделе дается простой эвристический алгоритм приближенного решения задачи.

3. Алгоритм кодирования. В основу алгоритма положен следующий эвристический прием. Соседние коды присваиваются таким парам микрокоманд, которые встречаются более, чем в одном К-множестве. Этот прием подсказан тем обстоятельством, что задача построения разбиений $T \subset \theta$, определяющих код, сводится к нахождению множества $\theta^1 = \{\theta_1^1, \ldots, \theta_n^1\}$ разбиений первого уровня структуры θ [5].

По К-множествам будем строить список всевозможных пар микрокоманд $P = \{p_1, p_2, \ldots, p_k\}$, целиком входящих хотя бы в одно К-множество. Каждой паре из P будем придавать вес, равный числу К-множеств, содержащих пару. Исходя из списка P будем строить множество разбиений θ^1. Для того чтобы изложить правила построения θ^1, необходимы следующие понятия.

Две пары $B_s = \{q_g, q_h\}$ и $B_t = \{q_k, q_l\}$ из одного и того же разбиения θ_j^1 называются сопряженными по $\theta_i^1 (B_s = B_t^{\#}(\theta_i^1))$, если в θ_i^1 есть две пары $\{q_g, q_k\}$ и $\{q_h, q_l\}$ или $\{q_h, q_l\}$ и $\{q_h, q_k\}$. Иными словами, из двух пар, входящих в θ_j^1 и сопряженных по θ_i^1, при перемножении разбиений θ_i^1 и θ_j^1 получаются четыре блока $\{\overline{q_g}; \overline{q_h}; \overline{q_k}; \overline{q_l}\} \subset \theta^0$, а при сложении θ_i^1 и θ_j^1 – блок $\{\overline{q_g, q_h, q_k, q_l}\} \subset \theta_r^2$.

Множество пар $C = \{B_{i_1}, \ldots, B_{i_k}\} \subset \theta_i^1$ называется коллекцией, если есть такое подмножество $\{\theta_{j_1}^1, \ldots, \theta_{j_\nu}^1\}$, что $\bigcup_{l=1}^{k} B_{i_l}$ является блоком разбиения $(\theta_{j_1}^1 + \ldots + \theta_{j_\nu}^1) \in \theta^\nu$.

В [5] доказано, что для того чтобы множество разбиений $\{\theta_1^1, \ldots, \theta_n^1\}$ на пары некоторого множества объектов Q было первым уровнем структуры, необходимо и достаточно, чтобы в каждом разбиении θ_i^1 любая пара имела сопряженную по любому другому разбиению θ_j^1 из множества $\{\theta_1^1, \ldots, \theta_n^1\}$.

Это утверждение лежит в основе следующей процедуры построения из списка P множества θ^1.

Разбиения θ_i^1 строятся в порядке возрастания их номеров. В разбиении θ_i^1 будем рассматривать коллекции, порожденные совокупностью всех уже построенных разбиений, т.е. $\{\theta_1^1, \ldots, \theta_i^1\}$. Считаем, что θ_1^1 состоит из 2^{n-1} коллекций (здесь коллекция – это блок, т.е. пара микрокоманд). В θ_2^1 имеется 2^{n-2} коллекций по две пары в каждой, в θ_i^1 – 2^{n-1} коллекций по 2^{i-1} пар. В качестве первой пары любой коллекции будем выбирать пару $p_t \in P$ по правилам:

I) в пары любой коллекции уже построенных разбиений не входят вместе оба элемента p_t (по теореме 2 из [5]);

2) пара p_t имеет наибольший вес по сравнению с другими парами из P, для которых выполнено правило I.

Остальные пары коллекции строятся по формуле [5]

$$B_{k_l} = B^{*}_{k_l - 2^{i-1}} (\theta_j^1),$$

где $k_l = l+(k-1)2^k$, $l = 2,3,\ldots,2^{i-1}$, k – номер коллекции в разбиении θ_i^1, $j =]\log l[- 1$.

Получив все разбиения θ_i^1, находим двублочные разбиения

$$\tau_i = \sum_{\substack{j=1 \\ j \neq i}}^{n} \theta_j^1, \quad i = 1,\ldots,n.$$

Реализация описанной процедуры построения множества $T=\{\tau_i\}$ на ЭВМ таким образом, как предложено в [5], требует для записи множества θ^1 $2^n \cdot n$ ячеек памяти. В условиях нехватки памяти лучше использовать компактное представление структуры θ в виде матрицы - строки S размером 2^n [2]. Элементы множества кодируемых объектов, разбиения которого образуют структуру, расположены в клетках строки S в таком порядке, что, выбирая их по формуле

$$\theta_i^1=\{s_{k \cdot l}, s_{(k+2^{i-1}) \cdot l}\}, \; i=1,\ldots,n, \; k=1,\ldots,2^{i-1}, \; l=1,\ldots,2^{n-1}, \qquad (1)$$

мы получаем все разбиения первого уровня структуры θ, l – номер коллекции в i-м разбиении на пары, k – номер пары в коллекции.

При таком представлении структуры задача кодирования сводится к выбору первых пар коллекций, записи их в матрице S в клетках, определяемых формулой (I), и переупорядочению элементов множества θ в строке S . Все это делается в соответствии со списком взвешенных пар P и правилам построения разбиений $\theta_i^1 \in \theta^1$.

Алгоритм кодирования, использующий компактное представление θ, состоит из следующих процедур.

I. Составляется список взвешенных пар микрокоманд P по K - множествам.

2. В S записываются 2^{n-1} пар из P по правилам I,2 так, что каждые 2 элемента s_{2k-1} и s_{2k} $(k = 1,\dots,n/2)$ образуют пару.

3. В P выбирается по правилам I,2 пара p_t в качестве первой пары 1-й коллекции i-го разбиения. Записываем ее в S в согласии с формулой (I) и переупорядочиваем элементы из θ в строке S таким образом, чтобы каждая пара ее элементов, начиная с первого, была коллекцией разбиения θ_1^1,каждые две пары,начиная с первой, - коллекцией из θ_2^1 и т.д.,наконец,каждое множество из 2^{i-1} пар - коллекцией разбиения θ_i^1 . Эта процедура выполняется до тех пор, пока не будут записаны все коллекции всех разбиений $\theta_i^1 \in \theta^1$.

4. Элементам множества θ, записанным в клетках S , начиная с первой клетки, присваиваются кодирующие слова в лексикографическом порядке.

Проиллюстрируем все процедуры алгоритма на следующем примере. Закодируем множество микрокоманд $\{q_1,\dots,q_8\}$, заданное матрицей M_2 при $n = 3$, $m = 5$.

M_2 =

q_1	1				
q_2	1	1		1	
q_3			1		1
q_4		1	1		
q_5	1	1		1	
q_6	1				
q_7		1			
q_8					1

В соответствии с расстановкой единиц в M_2 записываем K-множества микрокоманд: $K_1=\{q_1,q_2,q_5,q_6\}$,$K_2=\{q_2,q_4,q_5,q_7\}$,$K_3=\{q_3,q_4\}$, $K_4= \{q_2,q_5\}$, $K_5 = \{q_3,q_8\}$. Согласно процедуре I, строим множество взвешенных пар микрокоманд:

$$P = \{q_1,q_2;q_1,q_5;q_1,q_6;q_2,q_5(3);q_2,q_6;q_5,q_6;q_2,q_4;q_2,q_7;$$
$$q_4,q_5;q_4,q_7;q_5,q_7;q_3,q_4;q_3q_8\}.$$

В скобках после пары $q_2 q_5$ указан ее вес, остальные пары имеют вес I. В результате выполнения процедуры 2 получаем

$S =$

1	2	3	4	5	6	7	8
q_2	q_5	q_1	q_6	q_4	q_7	q_3	q_8

Это значит, что $\theta_1^1 = \{\overline{q_2, q_5}; \overline{q_1, q_6}; \overline{q_4, q_7}; \overline{q_3, q_8}\}$. Выполняем процедуру 3.

а) $i = 2,\ l = 1,\ p_t = \{q_1, q_2\}$,

$S =$

1	2	3	4	5	6	7	8
q_1	q_6	q_2	q_5	q_4	q_7	q_3	q_8

б) $i = 2,\ l = 2,\ p_t = \{q_3, q_4\}$,

$S =$

1	2	3	4	5	6	7	8
q_1	q_6	q_2	q_5	q_3	q_8	q_4	q_7

Получено разбиение $\theta_2^1 = \{\overline{q_1, q_2}; \overline{q_6, q_5}; \overline{q_3, q_4}; \overline{q_8, q_7}\}$.

в) $i = 3,\ l = 1,\ p_t = \{q_2, q_4\}$,

$S =$

1	2	3	4	5	6	7	8
q_2	q_5	q_1	q_6	q_4	q_7	q_3	q_8

$$\theta_3^1 = \{\overline{q_2, q_4}; \overline{q_5, q_7}; \overline{q_1, q_3}; \overline{q_6, q_8}\}.$$

Процедура 4 дает следующий код: $q_2 = 000$, $q_5 = 00I$, $q_1 = 0I0$, $q_6 = 0II$, $q_4 = I00$, $q_7 = I0I$, $q_3 = II0$, $q_8 = III$. В результате получаем покрытие К-множеств, совпадающее в данном случае с множеством Г :

$$\mathcal{P} = \{\{q_1, q_2, q_5, q_6\}, \{q_2, q_4, q_5, q_7\}, \{q_3, q_8\}, \{q_3, q_4\}, \{q_2, q_5\}\}.$$

А матрица М после склеек микрокоманд имеет вид

М =

x	$\bar{x}_1$	x_2	$\bar{x}_2$	x_3	$\bar{x}_3$	f_1	f_2	f_3	f_4	f_5
	1					1				
			1				1			
1					1			1		
	1		1						1	
1		1								1

В заключение заметим, что предложенный алгоритм, дающий выигрыш по памяти по сравнению с алгоритмом из [5], по времени выполнения на ЭВМ не хуже алгоритма из [5].

Литература

I. АЧАСОВА С.М., БАНДМАН О.Л. Матричный метод синтеза комбинационных схем и логических преобразователей конечных автоматов. -"Изв. АН СССР. Тех. кибернетика", 1975, № 6, с. 99-106.

2. АЧАСОВА С.М. Преобразование микропрограмм при вложении их в программируемые логические матрицы. -В кн.: Вопросы теории и построения вычислительных систем (Вычислительные системы, вып. 73.) Новосибирск, 1978, с. 66-79.

3. Синтез асинхронных автоматов на ЭВМ. Под ред. А.Д.Закревского. Минск, "Наука и техника", 1975.

4. BANDMAN O.L. State-Assignment of Sequential Machines for LSI Implementation.-IN: Discrete Systems. Proc.of International Symposium IFAC, October,1974. Riga, p.68-80.

5. АЧАСОВА С.М., БАНДМАН О.Л. Проектирование автоматов управления на основе БИС. -В кн.: Вычислительные системы. Вып.64.Автоматизация проектирования в микроэлектронике. Теория, методы, алгоритмы. 1975, с. 26-52.

Поступила в ред.-изд.отд.
19 октября 1978 года