УДК 681.142.353

ОПТИМАЛЬНЫЙ СИНТЕЗ АСИНХРОННЫХ АВТОМАТОВ В КРИОТРОННОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

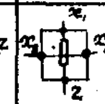
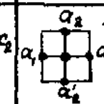
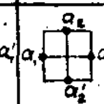
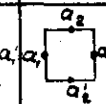
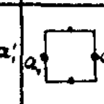
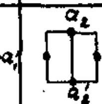
О.Л. Бандман

При постановке задачи синтеза автоматов в вычислительной среде выдвигается требование получения программ настройки, занимающих минимальную площадь [1]. В настоящей работе предлагается методика синтеза, удовлетворяющая этому требованию для большинства автоматов с минимизированным числом состояний, у которых на множестве состояний не существует разбиений со свойствами подстановки (SP - свойствами) [2] и, следовательно, такие автоматы не декомпозируются. Приводимая здесь методика не обеспечивает также минимальную площадь для автоматов со слабо определенными таблицами переходов, поскольку путем доопределения они могут быть приведены к декомпозируемым (например, многоразрядные однородные автоматы: регистры, счетчики и др.).

§ 1. Сложность реализации автоматов в криотронной вычислительной среде

Автомат реализуется в двумерной криотронной вычислительной среде, функционально-соединительный базис которой представлен в табл. 1.

Таблица 1

Обозначение	F_1	F_2	P	D	O	H	I
Схема							
Функция	$Z=x_2\cdot\bar{x}_1$	$Z=x_1\cdot\bar{x}_2$	$a_1=a_1'=a_2=a_2'$	$a_1=a_1'$ $a_2=a_2'$		$a_1=a_1'$	$a_2=a_2'$

Выбор такого базиса обусловлен следующими положениями:

1. Здесь достигается максимальная простота элемента (12 криотронов) при достаточно полном наборе функций.

2. Применимы принципы релейно-контактного синтеза, что позволяет легко получать программы настройки для дизъюнктивных нормальных форм (д.н.ф.) [3].

3. Каждый элементарный квадратик площади такой среды выполняет либо функцию одного криотрона, либо горизонтального и (или) вертикального соединений. Поэтому методы синтеза в криотронной среде с индивидуальной настройкой являются одновременно методами проектирования устройств для криотронной среды с фиксированной настройкой, а также основой для проектирования тонкопленочных схем.

Это последнее обстоятельство делает особенно важной задачу минимизации.

Основные принципы, которые положены в основу принятых методов, состоят в следующем:

1. Все переменные вводятся в парафазном коде.

2. Функции в д.н.ф. реализуются в прямоугольном поле среды (блоке), где каждому столбцу соответствует одна из переменных, а в каждой строке реализуется одна из конъюнкций. Поле среды справа и слева обрамляется столбцами "Р", которые объединяют элементарные конъюнкции в дизъюнкции [3].

3. Если в ряде строк или блоков реализуются взаимно ортогональные функции $f_1, \ldots, f_n$, то схема, соответствующая $f = f_1 \vee f_2 \vee \ldots f_n$ (объединение их в параллель), обладает памятью, т.е. образует многовходовой триггер с n устойчивыми состояниями.

Составление программы настройки не включает в себя процедур, связанных с логическими или вычислительными операциями, а основано только на преобразовании заданного представления функции к представлению в виде матриц инциденций подмножеств входных и внутренних переменных.

Пусть асинхронный автомат без выходов*) задан системой $\mathfrak{A} = \langle \Sigma, S, \delta \rangle$, где $\Sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_z\}$ – конечное множество входных состояний, определяемых наборами входных переменных из множества $X = \{x_1, \bar{x}_1, \ldots, x_n, \bar{x}_n\}$, причем

$$\sigma_i(X) = \tilde{x}_1 \cdot \tilde{x}_i \cdot \ldots \cdot \tilde{x}_n . \qquad (I)$$

$S = \{s_1, s_2, \ldots, s_v\}$ – конечное множество внутренних состояний автомата, определяемых наборами внутренних переменных из множества $Q = \{q_1, \bar{q}_1, \ldots, \bar{q}_p\}$, причем

$$s_k(Q) = \tilde{q}_1 \cdot q_2 \cdot \ldots \cdot \tilde{q}_p , \qquad (2)$$

где $\tilde{x}_i$ и $\tilde{q}_j$ обозначают либо x_i и q_j, либо $\bar{x}_i$ и $\bar{q}_j$.

Функция переходов δ есть отображение $[\Sigma \times S] \to S$, заданное таблицей (графом) переходов, в которой удовлетворяется следующее условие

$$\delta(s_j, \sigma_i) = \delta[\delta(s_j, \sigma_i), \sigma_i] . \qquad (3)$$

ОПРЕДЕЛЕНИЕ I. Множеством столбцовых переходов $\eta(\sigma_i) = \{s_j, s_k\}$ называется множество пар внутренних состояний, таких что $(s_j, s_k) \in \eta(\sigma_i)$, если $\delta(s_j, \sigma_i) = s_k$.

ОПРЕДЕЛЕНИЕ II. Множество переходов автомата $\eta = \bigcup_{i=1}^{z} \eta(\sigma_i)$.

ОПРЕДЕЛЕНИЕ III. Столбцовым разбиением $\pi(\sigma_i) = \{\mathcal{K}_{s_k}(\sigma_i)\}, (s_k \in S(\sigma_i))$ называется разбиение подмножества $S(\sigma_i) \subset S^*$, состоящего из состояний, расположенных в столбце σ_i таблицы переходов, такое что если $s_j \in \mathcal{K}_{s_k}(\sigma_i)$, то $\delta(s_j, \sigma_i) = s_k$.

*) Реализация функций выходов всегда может быть выполнена, как показано в [3,4].

Подмножества $\mathcal{K}_{s_k}(\sigma_i)$ в [5] называются $\mathcal{K}$ – множествами. Объединение всех $\mathcal{K}$ – множеств автомата обозначается

$$\pi = \bigcup_{i=1}^{z} \pi(\sigma_i) = \{\mathcal{K}_{s_k}(\sigma_i)\}, (s_k \in S;\ \sigma_i \in \Sigma).$$

На рис. I даны таблицы переходов и множества $\eta(\sigma_i)$ и $\pi(\sigma_i)$ для всех $\sigma_i \in \Sigma$.

δ	σ_1	σ_2	σ_3	σ_4
a	$\boxed{a}$	$\boxed{a}$	d	b
b	c	d	d	$\boxed{b}$
c	$\boxed{c}$	d	e	d
d	a	$\boxed{d}$	$\boxed{d}$	$\boxed{d}$
e	c	$\boxed{e}$	$\boxed{e}$	$\boxed{e}$

Рис. I

$$\eta(\sigma_1) = \{\overline{bc}; \overline{da}; \overline{ec}\},$$
$$\eta(\sigma_2) = \{\bar{a}; \overline{bd}; \overline{cd}; e\};$$
$$\eta(\sigma_3) = \{\overline{ad}; \overline{bd}; \overline{ce}\};$$
$$\eta(\sigma_4) = \{\overline{ab}; \overline{cd}, \bar{e}\}.$$

$$\pi(\sigma_1) = \{\overline{da}; \overline{bec}\},$$
$$\pi(\sigma_2) = \{\bar{a}; \overline{bcd}, \bar{e}\};$$
$$\pi(\sigma_3) = \{\overline{abd}, \overline{ce}\};$$
$$\pi(\sigma_4) = \{\overline{ab}; \overline{cd}; e\}.$$

Реализация асинхронного автомата в криотронной вычислительной среде сводится к реализации его функций перехода и к организации обратной связи. Поскольку необходимо обеспечить хранение результата, функции реализуются в виде взаимно ортогональных пар:

$$q_{j,(t+1)}(\mathcal{X},Q)=\bigvee_{s_k\in S_j'}\ \bigvee_{\delta(s_\ell\sigma_i)=s_k}\sigma_i(\mathcal{X})\cdot s_\ell(Q);$$

$$\bar{q}_{j,(t+1)}(\mathcal{X},Q)=\bigvee_{s_k\in S_j''}\ \bigvee_{\delta(s_\ell,\sigma_i)=s_k}\sigma_i(\mathcal{X})\cdot s_\ell(Q); \qquad (4)$$

где S_j' и S_j'' – подмножества S, у которых в кодирующем слове q_j = I и q_j = 0, соответственно.

Составление программы настройки состоит в построении матрицы [3], в которой столбцам поставлено в соответствие множество $\mathcal{X}\cup Q\cup Q_{t+1}$, а строкам – $Q\cup\mathcal{D}_q$, где $\mathcal{D}_q$ – множество конъюнкций во всех выражениях (4). Сложность реализации равна размерности матрицы

$$N_I=(|\mathcal{D}_q|+2\rho)(2n+4\rho). \qquad (5)$$

Значение $|\mathcal{D}_q|$ является суммарной длиной д.н.ф. всех функций (4). Известно [6], что для почти всех функций от ($n+\rho$) переменных длина кратчайшей д.н.ф. оценивается следующим образом:

$$\mathcal{I}(\bar{f}(x^{n+\rho}))\geqslant\frac{c\cdot 2^{n+\rho}}{\log(n+\rho)\cdot\log\log(n+\rho)}.$$

Следовательно, суммарная длина д.н.ф. для функций (4) и их отрицаний

$$|\mathcal{D}_q|\geqslant 2\rho\,\frac{c\cdot 2^{n+\rho}}{\log(n+\rho)\cdot\log\log(n+\rho)}. \qquad (6)$$

Другой способ [4] предусматривает реализацию функций (4) в два каскада: в первом каскаде реализуется множество дизъюнкций

$$\mathcal{D}_{s_k}(\mathcal{X},Q)=\bigvee_{\delta(s_\ell,\sigma_i)=s_k}\sigma_i(\mathcal{X})\cdot s_\ell(Q);\quad (s_k\in S), \qquad (7)$$

во втором–

$$q_{j,(t+1)}(\mathcal{D}_{s_k})=\mathop{\&}_{s_k\in S_j''}\bar{\mathcal{D}}_{s_k};$$

$$\bar{q}_{j,(t+1)}(\mathcal{D}_{s_k})=\mathop{\&}_{s_k\in S_j'}\bar{\mathcal{D}}_{s_k};\quad (j=1,2,\dots,\rho). \qquad (8)$$

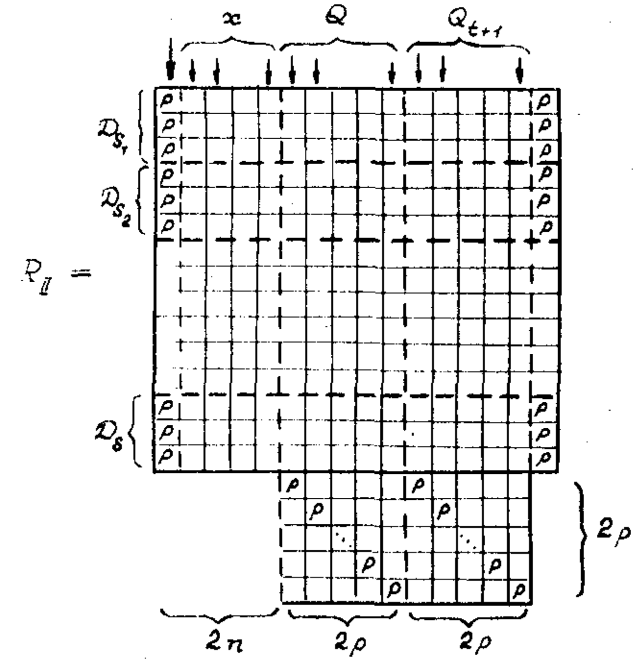


Рис. 2

Матрица – программа (рис.2) имеет строки, соответствующие множеству $Q\cup\mathcal{D}_s$, где $\mathcal{D}_s$ – множество конъюнкций в (7) для всех $s_k\in S$, и столбцы, соответствующие множеству $\mathcal{X}\cup Q\cup Q_{t+1}$. Функции (8) выполняются в столбцах Q_{t+1}. Сложность реализации

$$N=(|\mathcal{D}_s|+2\rho)(2n+4\rho). \qquad (9)$$

Здесь $|\mathcal{D}_s|$ – суммарная длина д.н.ф. (7), представляющих собой $v=|S|$ взаимно ортогональных функций от ($n+\rho$) переменных, и, следовательно,

$$|\mathcal{D}_s|\leqslant 2^{n+\rho} \qquad (10)$$

Сравнение (6) и (10) показывает, что для большинства автоматов $|\mathcal{D}_s|<|\mathcal{D}_q|$, и поэтому каскадную реализацию следует предпочитать непосредственной*).

---

*) Для среды с несимметричным функциональным базисом [2,3,4] такое утверждение не всегда правильно, т.к. там оценки (5) и (9) отличаются не только числом строк, но и числом столбцов.

Теперь можно конкретно поставить задачу оптимизации площади среды, необходимой для реализации автомата: найти такой способ преобразования таблицы переходов в матрицу - программу, при котором обеспечивается минимум числа строк. При этом можно утверждать, что прямоугольная программа такой структуры, когда столбцам соответствует множество $\mathcal{X} \cup Q \cup Q_{t+1}$ и число строк при этом минимизировано, является действительно минимальным для большинства автоматов, если исключить возможность их декомпозиции.

Это утверждение основано на следующих положениях.

I. В [3] показано, что меньшую сложность реализации логической функции, по сравнению с реализацией в виде кратчайшей д.н.ф., возможно получить только в случае, если она имеет разделительную декомпозицию вида

$$f(x_1, \bar{x}_1, \ldots, \bar{x}_n) = \varphi[f_1(x_{i_1}, \ldots, x_{i_k}), f_2(x_{i_{k+1}}, \ldots, x_{i_n})],$$

причем максимальное число дизъюнктивных членов в f_1 или f_2 на некоторое число меньше, чем длина кратчайшей д.н.ф. f. Легко подсчитать, что из общего числа $2^{2^{n+\rho}}$ функций от $n+\rho$ переменных только небольшая доля $n \cdot 2^{n+\rho}$ может иметь такую декомпозицию ($2^{n+\rho-1}$ способов разбиения ($n + \rho$) переменных на 2 группы и в каждом случае $2n$ способов замены переменных их отрицаниями).

Отсюда следует вывод, что в большинстве случаев целесообразно реализовать функции в виде кратчайших д.н.ф., то есть прямоугольными блоками с числом столбцов, равным числу переменных, и числом строк, равным числу элементарных конъюнкций.

2. На участке среды, где реализуется автомат, должно быть размещено $v = |S|$ прямоугольных блоков таким образом, чтобы каждый предыдущий соединялся $2(n + \rho)$ каналами с каждым последующим. Очевидно, что минимальная площадь для соединительных каналов соответствует тому случаю, когда они все прямые. Следовательно, каждой переменной $\tilde{x} \in \mathcal{X}$ и $\tilde{q} \in Q$ должен быть отведен один столбец или одна строка программы.

3. Поскольку автомат не декомпозируется, то множество $\mathcal{D}_S$ невозможно подразделить хотя бы на два подмножества $\mathcal{D}_S'$ и $\mathcal{D}_S''$ таких, чтобы подмножества переменных $(\mathcal{X} \cup Q)'$ и $(\mathcal{X} \cup Q)''$, являющиеся

аргументами функций $\mathcal{D}_{s_k} \in \mathcal{D}_S'$ и $\mathcal{D}_{s_k} \in \mathcal{D}_S''$ соответственно, не пересекались, т.е. $(\mathcal{X} \cup Q)' \cap (\mathcal{X} \cup Q)'' = \varnothing$. Из этого следует, что не существует двух блоков, реализующих $\mathcal{D}_{s_k} \in \mathcal{D}_s$, которые можно было бы разместить друг относительно друга в направлении, перпендикулярном путям прохождения переменных.

Из этих трех положений вытекает, что поиск минимальных реализаций в вычислительной среде с приведенным базисом для большинства недекомпозируемых автоматов состоит в минимизации числа строк в матрице $\mathcal{R}$ (рис.2).

Естественным ходом решения такой задачи было бы написание по таблице переходов уравнений (7), поиск для них кратчайших д.н.ф. с одновременным преобразованием их к системе ортогональных функций. Однако при этом нарушался бы четвертый принцип, положенный в основу принятых методов, и потерялось бы достоинство простоты непосредственного преобразования таблицы переходов к программе настройки. Поэтому к методу предъявляется требование: найти такие системы подмножеств множества $(Q \cup \mathcal{X})$, матрица инциденций которых имела бы минимальное число строк.

рых имела бы минимальное число строк.

Безусловно, нет никакой гарантии того, что процедура поиска такой системы подмножеств для всех автоматов проще, чем поиск кратчайших форм. Однако эта процедура очень алгоритмична, удобна для поиска наилучших кодов [8] и легко видоизменяется для получения простых методов почти оптимального синтеза.

§ 2. Системы разбиений множеств состояний автомата

Рассматриваются автоматы, у которых все состояния кодируются двоичными кодами. Так, например, кодами входных и внутренних состояний автомата A_1 (рис.I) могут быть соответственно

$$C_\Sigma = \begin{array}{c} \\ \sigma_1 \\ \sigma_2 \\ \sigma_3 \\ \sigma_4 \end{array} \begin{array}{c} \begin{array}{cc} x_1 & x_2 \end{array} \\ \begin{array}{|c|c|} 0 & 0 \\ I & I \\ 0 & I \\ I & 0 \end{array} \end{array} \qquad C_S = \begin{array}{c} a \\ b \\ c \\ d \\ e \end{array} \begin{array}{|c|c|c|c|} I & I & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & I & I \\ I & 0 & 0 & I \\ 0 & I & I & I \end{array}$$

Каждому двоичному коду ставится в соответствие **с и с т е м а д в у б л о ч н ы х р а з б и е н и й м н о ж е с т в а с о с т о я н и й**, т.е. такое множество двублочных разбиений, произведение которых равно тривиальному нулевому разбиению[*)].

Будем обозначать системы двублочных разбиений множеств S и Σ через $T=\{\tau_1,\tau_2,\dots,\tau_\rho\}$ и $\Omega=\{\omega_1,\omega_2,\dots\omega_n\}$ соответственно. Так, например, коду C_Σ соответствует $\Omega=\{\omega_1,\omega_2\}$, где $\omega_1=\{\overline{\sigma_1\sigma_3};\ \overline{\sigma_2,\sigma_4}\}$, $\omega_2=\{\overline{\sigma_1,\sigma_4};\overline{\sigma_2,\sigma_3}\}$, а коду C_S – $T=\{\tau_1,\tau_2,\tau_3,\tau_4\}$, где $\tau_1=\{\overline{a,d};\ \overline{b,c,e}\}$, $\tau_2=\{\overline{a,e};\ \overline{b,c,d}\}$, $\tau_3=\{\overline{c,e};\ \overline{a,b,d}\}$, $\tau_4=\{\overline{c,d,e};\ \overline{a,b}\}$.

Условие $\prod_{j=1}^{\rho}\tau_j=0$ обеспечивает для каждого состояния наличие своего кодирующего слова, отличающегося от всех других хотя бы в одном разряде. Подмножество состояний, в которых j-я переменная кодируется единицей, будем обозначать S_j' или Σ_j', а нулем – S_j'' или Σ_j''. Тогда $\tau_j=\{S_j';S_j''\}$, а $\omega_j=\{\Sigma_j';\Sigma_j''\}$.

ОПРЕДЕЛЕНИЕ IV. **П о к р ы т и я м и** P_T и P_Ω называются множества всевозможных произведений двублочных разбиений T и Ω соответственно:

$$\left.\begin{aligned}&P_T=\{B_\ell\subset S\};\quad B_\ell\in\prod_{k=1}^{\nu}\tau_{j_k};\ (\nu,j_k=1,2,\dots,\rho),\\&P_\Omega=\{C_m\subset\Sigma\};\quad C_m\in\prod_{k=1}^{\mu}\omega_{i_k};\ (\mu,i_k=1,2,\dots,n).\end{aligned}\right\}\qquad(11)$$

Каждому $B_\ell\in P_T$ или $C_m\in P_\Omega$ соответствует $Q_{B_\ell}\subset Q$ или $\mathcal{X}_{C_m}\subset\mathcal{X}$. При этом $q_j\in Q_{B_\ell}$, если $B_\ell\subset S_j'$, и $\bar q_j\in Q_{B_\ell}$, если $B_\ell\subset S_j''$. Аналогично, $x_i\in\mathcal{X}_{C_m}$, если $C_m\subset\Sigma_i'$, и $\bar x_i\in\mathcal{X}_{C_m}$, если $C_m\subset\Sigma_i''$.

В соответствии с этим определим функции

---

*) Произведением разбиений называется такое разбиение, в котором в данном блоке могут находиться только те состояния, которые во всех сомножителях входят в один и тот же блок [2]. Тривиальными считаются разбиения вида $\tau=\{\overline{s_1};\overline{s_2};\dots;\overline{s_\nu}\}=0$ и $\tau=\{\overline{s_1,s_2,\dots,s_\nu}\}=1$.

$$\left.\begin{aligned}&B_\ell(Q)=\mathop{\&}_{\tilde q_j\in Q_{B_\ell}}\tilde q_j\,;\\&C_m(\mathcal{X})=\mathop{\&}_{\tilde x_i\in\mathcal{X}_{C_m}}\tilde x_i\,.\end{aligned}\right\}\qquad(12)$$

Так, например, для приведенных выше кодов C_S и C_Σ автомата A_1

$P_T=\{a;b,c;d;e;a,d;a,e;c,e;b,c;b,d;b,c,e;b,c,d;a,b,d\}$;

$P_\Omega=\{\sigma_1;\sigma_2;\sigma_3;\sigma_4;\ \sigma_1\sigma_3;\ \sigma_2,\sigma_4;\sigma_1,\sigma_4;\ \sigma_2,\sigma_3\}$,

$a,d(Q)=q_1;\quad b,c(Q)=\bar q_1\cdot\bar q_2;\quad e(Q)=\bar q_1\cdot q_2\cdot q_3$.

$\sigma_1,\sigma_3(\mathcal{X})=x_1;\quad \sigma_2\sigma_3(\mathcal{X})=\bar x_2;\quad \sigma_3(\mathcal{X})=x_1\cdot\bar x_2$

Криотронная реализация и условие (3) для таблицы переходов асинхронных автоматов предъявляют к системам двублочных разбиений определенные требования. Первая требует выполнения условий ортогональности функций $\mathcal{D}_{S_k}(Q,\mathcal{X})$ (обеспечивается специальным кодированием входных состояний), второе накладывает условие отсутствия гонок и состязаний (обеспечивается противогоночным кодированием внутренних состояний).

Для обеспечения ортогональности функций $\mathcal{D}_{S_k}(Q,\mathcal{X})$ (7) необходимо, чтобы $\sigma_i(\mathcal{X})$ (i = 1,2,..., z) удовлетворяли условиям

$$\sigma_i(\mathcal{X})\cdot\sigma_j(\mathcal{X})=0\,,\qquad(13)$$

$$\bigvee_{i=1}^{z}\sigma_i(\mathcal{X})=1\qquad(14)$$

Условие (13) выполняется автоматически, если входные состояния закодированы в соответствии с системой разбиений, т.е. если $\prod_{1}^{n}\omega_i=0$. Условие (14) выполняется автоматически только, если $n=\log_2 z$. При $n<\log_2 z$ условия ортогональности входных функций нарушаются. Например, если z = 6, а n = 3, то системе двублочных разбиений

$$\Omega = \begin{cases} \omega_1 = \{\overline{\sigma_1, \sigma_2, \sigma_3};\ \overline{\sigma_4, \sigma_5, \sigma_6}\}, \\ \omega_2 = \{\overline{\sigma_1, \sigma_3, \sigma_4};\ \overline{\sigma_2, \sigma_5, \sigma_6}\}, \\ \omega_3 = \{\overline{\sigma_1, \sigma_2, \sigma_5};\ \sigma_3, \sigma_4, \sigma_6\}. \end{cases}$$

соответствует код, в котором отсутствуют наборы 0II и I00 и условие (I4) не выполняется. Если же состояниям σ_2 и σ_4 поставить в соответствие по два набора, то $\sigma_2(X)$ и $\sigma_4(X)$ будут определяться не тремя, а двумя переменными:

$$\sigma_2(X) = x_1 \cdot \bar{x}_2 \cdot x_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 = x_1 \cdot \bar{x}_2;$$

$$\sigma_4(X) = \bar{x}_1 \cdot x_2 \cdot x_3 \vee \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 = \bar{x}_1 \cdot x_2;$$

и множество $\{\sigma_i(X)\}$ (i = I,2,..., z) будет удовлетворять (I3) и (I4).

В терминах систем двублочных разбиений условия ортогональности $\{\sigma_i(X)\}$ (i = I,2,..., z) формулируются следующим образом.

Система двублочных разбиений $\Omega = \{\omega_j\}$ (j = I,2,..., n) множества $\Sigma = \{\sigma_i\}$ (i = I,2,..., n) $(2^{n-1} < z \geq 2^n)$ должна содержать $(n-1)$ полных двублочных разбиений $\{\omega_1, \omega_2, \dots, \omega_{n-1}\}$ и одно неполное ω_n, таких что:

I. $\prod_{i=1}^{n-1} \omega_i = C_m \in P_\Omega$ содержит 2^{n-1} блока, из которых можно выделить $(z - 2^{n-1})$ парных блока, составляющих разбиение $C_m^* \subset C_m$ на множестве $\Sigma^* \subset \Sigma$;

2. ω_n – двублочное разбиение множества Σ, причем $C_m^* \times \omega_n = 0$.

Условие, которому должна удовлетворять система двублочных разбиений множества S для обеспечения отсутствия критических состяаний, основано на теоремах и определениях из [5] и [9] и формулируется следующим образом:

$$\forall \{\overline{s_e, s_k};\ \overline{s_m, s_n}\} \in \eta(\sigma_i)\Big|_{s_k \neq s_n}, \tag{I5}$$

$$\exists\, \tau_j \in T \Big|_{\tau_j \geq \{\overline{s_e, s_k};\ \overline{s_m, s_n}\}}.$$

§ 3. Минимальное число строк в матрице-программе настройки

ОПРЕДЕЛЕНИЕ 7. Кратчайшим столбцовым покрытием $\Phi_r(\sigma_i) = \{F_\ell(\sigma_i)\}$ $(\ell = 1, 2, \dots, \varphi_r(\sigma))$ множества $S(\sigma_i)$, соответствующим системе двублочных разбиений T, называется покрытие, удовлетворяющее следующим условиям:

$$\left.\begin{aligned} &1)\ \forall (s_n, s_m) \in F_\ell(\sigma_i): \delta(s_n, \sigma_i) = \delta(s_m, \sigma_i); \\ &2)\ \Phi_r(\sigma_i) \leq \pi(\sigma_i); \\ &3)\ \Phi_r(\sigma_i) \subset P_r; \end{aligned}\right\} \tag{I6}$$

4) любое другое $\Phi(\sigma_i)$, удовлетворяющее пп.I,2,3, имеет больше подмножеств, чем $\Phi_r(\sigma_i)$.

$$\bigcup_{i=1}^{z} \Phi_r(\sigma_i) = \Phi_r; \quad \sum_{i=1}^{z} \varphi_r(\sigma_i) = \varphi_r.$$

Например, для автомата A_1 (рис.I) и системы разбиений T, приведенной на стр. 9, столбцовые покрытия равны:

$$\Phi_r(\sigma_1) = \{\overline{a, d};\ \overline{b, e, c}\};$$

$$\Phi_r(\sigma_2) = \{\bar{a};\ \overline{b, d};\ \overline{c, d};\ \bar{e}\};$$

$$\Phi_r(\sigma_3) = \{\overline{c, e};\ \overline{a, b, d}\};$$

$$\Phi_r(\sigma_4) = \{\overline{a, b};\ \overline{c, d};\ \bar{e}\}.$$

ТЕОРЕМА I. Если ни в одной строке таблицы переходов не содержится одинаковых состояний, то есть

$$\delta(s_j, \sigma_i) \neq \delta(s_j, \sigma_k), \quad \begin{aligned} &j = 1, 2, \dots, \nu, \\ &i, k = 1, 2, \dots, z, \end{aligned} \tag{I7}$$

и система двублочных разбиений, в соответствии с которой закодированы внутренние состояния, удовлетворяет условиям (I5), то минимальное число строк в матрице R автомата равно $\varphi_r + 2\rho$.

ДОКАЗАТЕЛЬСТВО. Дизъюнктивные члены в (7) сгруппируем так, чтобы в каждую группу входили только те конъюнкции $\sigma_i(x)\cdot s_j(Q)$, в которых s_j образуют блоки $\Phi_\tau(\sigma_i)$. Обозначим полученные д.н.ф $F_e(x,Q)$. Тогда

$$F_e(x,Q)=\bigvee_{s_j\in F_e}\sigma_i(x)\cdot s_j(Q)=\sigma_i(x)\cdot\bigvee_{s_j\in F_e}s_j(Q), \tag{18}$$

$$(F_e\in\Phi_\tau(\sigma_i));$$

$$\mathcal{D}_{s_k}(x,Q)=\bigvee_{\delta(s_j\in F_e,\sigma_i)=s_k}F_e(x,Q). \tag{19}$$

Теорема будет доказана, если показать, что автомат может быть реализован так, что в матрице R каждому подмножеству конъюнкций $F_e(x,Q)$ соответствует только одна строка, и что меньшим числом конъюнкций выразить все переходы в автомате невозможно.

Для этого дополним множество S автомата A до множества S^*, содержащего все 2^ρ состояний от ρ переменных, и обозначим новые состояния $s_j^*\in S^*\setminus S$ в произвольном порядке недостающими словами кода. Заполним добавленные к таблице переходов строки так, чтобы

$$\delta(s_j^*,\sigma_i)=\delta(s_k\in F_e,\sigma_i), \tag{20}$$

если $\quad S_j^*(Q)\leq F_e(Q)\Big|_{F_e\in\Phi_\tau(\sigma_i)}.$

Поскольку все добавленные к автомату A состояния недостижимы, то полученный новый автомат A^* эквивалентен A и мы имеем право заменить реализацию A реализацией A^*. Соответствующая A^* система разбиений $T^*=\{\sigma_1^*,\sigma_2^*,\ldots,\sigma_\rho^*\}$ такова, что

$$\forall\sigma_j\in T\quad\exists\sigma_j^*\in T^*\Big|_{\sigma_j^*\geq\sigma_j}. \tag{21}$$

Каждое столбцовое покрытие $\Phi_{\tau^*}^*(\sigma_i)\geq\Phi_\tau(\sigma_i)$ при этом же числе блоков, т.е.

$$\varphi_\tau^*=\varphi_\tau; \tag{22}$$

а каждое подмножество $F_e^*\in\Phi_{\tau^*}^*(\sigma_i)\supseteq F_e\in\Phi_\tau(q_i)$.

Согласно (20), каждый блок $F_e^*\in\Phi_\tau^*$ содержит все состояния из полного набора от ρ переменных, для которых $S_j^*(Q)\leq F_e(Q)$. Поэтому

$$\bigvee_{s_j^*\in F_e^*}s_j^*(Q)=\mathop{\&}_{\tilde q_k\in Q_{F_e}}\tilde q_k, \tag{23}$$

и, следовательно, $F_e^*(x,Q)$ реализуется в виде одной конъюнкции:

$$F_e^*(x,Q)=\sigma_i(x)\cdot\mathop{\&}_{\tilde q_k\in Q_{F_e}}\tilde q_k. \tag{24}$$

Теперь остается показать, что д.н.ф. (24) является кратчайшей формой, которой можно непротиворечиво реализовать все переходы в состояние s_k в автомате A. Из условия (17) теоремы I вытекает, что сокращение д.н.ф. (7) возможно только за счет преобразований внутри функции $\bigvee s_j(Q)\big|_{s_j\in\mathcal{K}_{s_k}(\sigma_i)}$. При этом если какие-либо два состояния s_n и s_m не входят в одно и то же подмножество $F_e\in\Phi_\tau$, то, согласно п.3(16), не существует ни одной переменной $\tilde q$, принадлежащей одновременно $s_n(Q)$ и $s_m(Q)$, и, следовательно, никакие тождественные преобразования между ними невозможны. Поэтому сокращения д. н. ф. возможны только внутри функций $\bigvee s_j(Q)\big|s_j\in B_e$. Кратчайшим покрытием S, удовлетворяющим обоим этим условиям, является, согласно определению (16), покрытие Φ_τ. И поскольку найдена кратчайшая (в виде одной конъюнкции) д.н.ф. для выражения каждого подмножества $F_e\in\Phi_\tau$, то φ_τ – минимальное число членов, необходимое для выражения всех переходов в автомате A.

Теорема I распространяется только на автоматы с полностью определенной таблицей переходов, удовлетворяющей условиям (17). Если условия (17) не выполняются, то автомат имеет одинаковые переходы при разных входных состояниях ("параллельные" переходы), наличие которых иногда позволяет улучшить оценку площади среды. Кроме того, если в таблице имеются неопределенные состояния, то их можно доопределить так, чтобы "параллельных" переходов стало больше. При этом минимум строк будет определяться не значением φ_τ, а другими характеристиками, зависящими также и от кода входных состояний.

ОПРЕДЕЛЕНИЕ VI. Покрытием $P_\pi(s_k)=\{\mathcal{Z}_{s_k}(C_m)\}$ на множестве $S_k\subset S$ называется множество пересечений подмножеств

$$\mathcal{K}_{s_k}(\sigma_i)\in\pi\,\big|_{\sigma_i\in C_m\in P_\Omega}, \qquad (25)$$

$$P_\pi(s_k)=\{\mathcal{Z}_{s_k}(C_m)\},\quad (s_i\, s_j)\in\mathcal{Z}_{s_k}(C_m),$$

если
$$(s_i\, s_j)\subset\bigcap_{\sigma_i\in C_m}\mathcal{K}_{s_k}(\sigma_i)$$

Множество пар $(\mathcal{Z}_{s_k}, C_m)$ образует множество $P_{\pi\Omega}(s_k)\subset[P_\pi(s_k)\times P_\Omega]$, причем $(\mathcal{Z}_{s_k}, C_m)\in P_{\pi\Omega}(s_k)$, если $\mathcal{Z}_{s_k}(C_m)\in P_\pi(s_k)$. (26)

ОПРЕДЕЛЕНИЕ VII. Кратчайшим покрытием $\Psi_{\tau\Omega}(s_k)=\{(\mathcal{H}_e(s_k); C_m)\}$ ($e=1,2,\dots,\varphi_{\tau\Omega}(s_k)$) на множестве $[S(s_k)\times\Sigma]$ называется подмножество $[P_\tau\times P_\Omega]$, удовлетворяющее следующим условиям:

1. $\Psi_{\tau\Omega}(s_k)\subset[P_\tau\times P_\Omega]$;
2. $\bigvee\left[(s_j,\sigma_i)\in(\mathcal{H}_e; C_m)\big|_{s_j\in\mathcal{H}_e;\,\sigma_i\in C_m}\right]: \delta(s_j\,\sigma_i)=s_k$; (27)
3. $\Psi_{\tau\Omega}(s_k)\leq P_{\pi\Omega}(s_k)$;
4. если существует другое покрытие $\Psi(s_k)$, удовлетворяющее п.п. 1,2,3 (27), то $|\Psi(s_k)|\geq\varphi_{\tau\Omega}(s_k)$.

Обозначим $\bigcup_{k=1}\Psi_{\tau\Omega}(s_k)=\Psi_{\tau\Omega}$, а $|\Psi_{\tau\Omega}|=\varphi_{\tau\Omega}$.

Например, для автомата A_1 (рис.1) и систем разбиений τ и Ω, приведенных на стр. 9, одно из наибольших покрытий
$\Psi_{\tau\Omega}=\{(\alpha,a,\sigma_1);(a,\sigma_2),(\gamma,b,\sigma_4),(b,e,c;\sigma_1);(c,\alpha,\sigma_2\,\sigma_4),(b;\sigma_2);(a,b,\alpha;\sigma_3),(e,\sigma_2\,\sigma_?)\}$

ТЕОРЕМА 2. Если множества входных и внутренних состояний автомата закодированы в соответствии с системами двублочных разбиений Ω и τ, удовлетворяющими условиям (13, 14) и (15) соответственно, то

наименьшее число строк в матрице R этого автомата равно $\varphi_{\tau\Omega}+2\rho$.

ДОКАЗАТЕЛЬСТВО. Общий ход доказательства аналогичен доказательству теоремы 1.

Сначала покажем, что автомат A можно реализовать так, что каждому элементу $\mathcal{H}_e$ покрытия $\Psi_{\tau\Omega}$ соответствует только одна строка в матрице R. Для этого заменим автомат A эквивалентным ему автоматом A^* с числом внутренних состояний 2^ρ и таблицей переходов, построенной в соответствии с (20). Покрытие $\Psi^*_{\tau\Omega}$ содержит столько же подмножеств, сколько $\Psi_{\tau\Omega}$, т.е.

$$\Psi^*_{\tau^*\Omega}=\Psi_{\tau\Omega}. \qquad (28)$$

Каждому подмножеству $\mathcal{H}_e^*\in\Psi^*_{\tau\Omega}$ поставим в соответствие дизъюнктивную форму

$$\mathcal{H}_e^*(\mathcal{X},Q)=\bigvee_{(\sigma_i\, s_j^*)\in\mathcal{H}_e^*}\sigma_i(\mathcal{X})\cdot s_j(Q)=\bigvee_{\sigma_i\in C_m}\sigma_i(\mathcal{X})\bigvee_{s_j\in B_m}s_j^*(Q). \qquad (29)$$

Поскольку для Σ выполняются условия (13) и (14) и $\mathcal{H}_e^*$ вследствие (20) содержит все состояния из полного набора, для которых $s_j^*(Q)\leq\mathcal{H}_e(Q)$, то обе дизъюнкции (29) могут быть выражены в виде одной конъюнкции (12), т.е.

$$\mathcal{H}_e^*(\mathcal{X},Q)=\mathop{\&}_{\tilde{x}_i\in\mathcal{X}_{C_m}}\tilde{x}_i\cdot\mathop{\&}_{\tilde{q}_k\in Q_{B_m}}\tilde{q}_k. \qquad (30)$$

Теперь стоит показать, что никакие другие группировки конъюнкций в (5) не дадут более коротких д.н.ф., чем

$$\mathcal{D}^*_{s_k}(\mathcal{X},Q)\doteq\bigvee_{\mathcal{H}_e^*\in\Psi^*_{\tau^*\Omega(s_n)}}\mathcal{H}_e^*(\mathcal{X},Q). \qquad (31)$$

Это следует непосредственно из того, что согласно п.4 определения (27) $\Psi_{\tau\Omega}$ является кратчайшим покрытием, объединяющим в своих блоках полные состояния (s_j,σ_i), для которых между соответствующими конъюнкциями возможны какие-либо тождественные преобразования. Поскольку для каждого подмножества этого покрытия найдена кратчайшая форма в виде одной конъюнкции, то $\alpha_{S\,min} = \Psi_{T\Omega}$, а полное число строк в R равно $\Psi_{T\Omega} + 2\rho$.

Алгоритмы определения кратчайших покрытий Φ_T и $\Psi_{T\Omega}$ даны в приложении I.

§ 4. Построение минимальной программы настройки

Если таблица переходов автомата, подлежащего синтезу, удовлетворяет условию (I7) теоремы I, то для построения оптимальной программы настройки необходимо определить покрытие Φ_T, если же эти условия не удовлетворяются, то определяется покрытие Ψ_{TQ}. Если соответствующее покрытие известно, то построение матрицы-программы R сводится к расстановке состояний настройки в прямоугольную таблицу (рис.2) в соответствии со следующими правилами.

В матрице $R=[r_{ij}]$ выделяются 4 подматрицы: $R_{D(x\cup Q)}$; $R_{DQ_{t+1}}$; R_{QQ} и $R_{QQ_{t+1}}$, лежащие на пересечениях строк и столбцов соответствующих их индексам. Если знаком "$\sim$" обозначить соответствие строки или столбца R какой-либо переменной или функции, то

I. $r_{ij} \in R_{D(x\cup Q)} = F_1$; если $i \sim \mathcal{H}_e \in \Psi_{TQ}$;

$j \sim (\tilde{x}_k \vee \tilde{q}_k) \in \mathcal{H}_e$.

2. $r_{ij} \in R_{DQ_{t+1}} = F_2$, если $i \sim \mathcal{H}_e \in \Psi_{TQ}(s_k)$;

$$\begin{cases} j \sim q_j, & (s_k \in S_j'), \\ j \sim \bar{q}_j, & (s_k \in S_j'') \end{cases}$$

$$\left.\begin{array}{l} 3.\ r_{ij} \in R_{QQ} = P \\ 4.\ r_{ij} \in R_{QQ_{t+1}} = P \end{array}\right\} \text{ если } i = j.$$

5. Во всех остальных случаях $r_{ij} = \mathcal{D}$.

Матрица-программа обрамляется двумя столбцами с элементами "$\bar{P}$", объединяющими справа и слева строки, соответствующие $\bar{\mathcal{D}}_S$. Кроме того, парами элементов P объединяются в параллель столбцы Q_{t+1} и Q (рис.2). Пример синтеза автомата дан в приложении 2.

§ 5. Методы синтеза, близкие к оптимальным

Поскольку алгоритмы поиска кратчайших покрытий Φ_T и $\Psi_{T\Omega}$ достаточно громоздки, часто бывает выгодно пользоваться более простой методикой, которая приводит к программам, близким к минимальным.

I. Для автоматов, у которых $v > r$, даже если их таблицы переходов не удовлетворяют условиям (I8), можно рекомендовать строить матрицу - программу R на основе покрытия Φ_T. При этом можно ограничиться поиском не наибольшего покрытия Φ_T, удовлетворяющего условиям I,2,3 (I7), а близкого к нему, путем применения алгоритма I (приложение I) без самого трудоемкого пункта 4.

2. Для автоматов, у которых $r \gg v$ близкие к оптимальным реализации можно получить, если строить R на основе покрытия $\Psi_{\eta\Omega} = \{(G_e, C_m)\}$, где $G_e \in \eta(\sigma_i)$, $\sigma_i \in C_m$. Построение покрытия $\Psi_{\eta\Omega}$ можно выполнить следующим образом.

а) По таблице переходов составить множество переходов

$$\eta = \bigcup_{i=1}^{r} \eta(\sigma_i) = \{G_e(\sigma_i)\}.$$

б) Образовать покрытие P_η множества Σ:

$$P_\eta = \{\mathcal{J}_e \subset \Sigma \mid_{(\sigma_n, \sigma_m) \in \mathcal{J}_e}, \text{ если } G_e(\sigma_n) = G_e(\sigma_m)\}.$$

в) Для каждого $\mathcal{J}_e \in P_\eta$ найти его кратчайшее или близкое к нему покрытие $P_{\mathcal{J}_e}$ подмножества $C_m \subset P_\Omega$;

г) $\Psi_{\eta\Omega} = \{(G_e, C_m) \mid_{G_e \in \eta(\sigma_i)}\}$; $\sigma_i \in C_m \mid C_m \in P_{\mathcal{J}_e}$.

3. Для автоматов с неполностью определенной таблицей переходов теоремы I и 2 не справедливы. Получение минимального числа строк в R связано с дополнительным перебором, поскольку различные способы доопределения дадут в результате различные кратчайшие покрытия $\Psi_{T\Omega}$.

Однако для получения программ, близких к оптимальным, можно рекомендовать дополнять неполные столбцовые разбиения до полных, не изменяя числа блоков в них. Это эквивалентно доопределению столбцов таблицы переходов неустойчивыми состояниями из подмножества состояний этого столбца.

§ 6. Кодирование внутренних состояний

Обычно при кодировании состояний асинхронных автоматов главной задачей является определение такого кода, который обеспечивал бы отсутствие критических состояний и при этом был по возможности менее избыточным [5,9]. Критерием избыточности считается число внутренних переменных автомата. При реализации автомата в вычислительной среде избыточность должна определяться сложностью реализации автомата в среде. В связи с этим возникает задача исследования соотношения этих двух критериев и выяснения необходимости построения алгоритмов определения кодов с меньшей избыточностью.

Существует два типа алгоритмов [5,9] определения противогоночных кодов:

I) алгоритмы, основанные на рассмотрении множества состязающихся пар переходов:

$$\Theta_1 = \left\{ (\overline{s_e, s_k}\,;\, \overline{s_m, s_n}) \Big|_{s_k \neq s_e,\,(s_e, s_k), (s_m, s_n) \in \eta(\sigma_i)} \right\}, \qquad (32)$$
$$\forall \sigma_i \in \Sigma .$$

2) алгоритмы, основанные на рассмотрении множества пар $\mathcal{K}$ - множеств:

$$\Theta_2 = \left\{ (\mathcal{K}_{s_k}(\sigma_i), \mathcal{K}_{s_e}(\sigma_i)) \Big|_{s_k \ne s_e} \right\} \quad \forall \sigma_i \in \Sigma \qquad (33)$$

Алгоритмы второго типа менее сложны, однако не гарантируют получение кода с минимальным числом внутренних переменных.

Оба типа алгоритмов предусматривают три следующих основных этапа.

1. Построение множества Θ_1 или Θ_2.
2. Построение избыточной системы двублочных разбиений

$$T_{max} = \{ \tau_1, \tau_2, \ldots, \tau_{\rho_{max}} \}$$

на множестве S.

3. Определение кратчайшего $T_{min} \subset T_{max}$ или близкого к кратчайшему $T' \subset T_{max}$ покрытия множества Q_1 или Q_2. Это приводит к получению минимального $\rho_{min} = |T_{min}|$ или почти минимального $\rho' = |T'|$ количества внутренних переменных.

Если применяются алгоритмы кодирования I-го типа (основанные на множестве состязающихся пар переходов), при реализации автоматов в вычислительной среде ρ_{min} не всегда обеспечивает N_{min}. Определение такой системы разбиений $\hat{T}$, которая обеспечила бы минимальную площадь среды, связано со значительным усложнением третьего этапа алгоритма. Здесь необходим перебор всех покрытий $T_i \subset T_{max}$ множества Q_1. При этом для каждого T_i следует определять $\Psi_{T_i \Omega}$ и подсчитывать $N(T_i)$, затем выбирать $T_i = \hat{T}$.

Если применяются алгоритмы кодирования 2-го типа (основанные на множестве пар $\mathcal{K}$ - множеств), то кратчайшее покрытие $\Phi_T = \pi$, а $\Psi_{T\Omega} = P_{\pi\Omega}$, и поэтому $\hat{T} = T_{min}$. Алгоритмы построения минимальных программ при кодировании по $\mathcal{K}$ - множествам предельно упрощаются, если построение матрицы-программы ведется на основе $\Phi_T = \pi$. Оно сводится, по существу, к размещению элементов F и P в прямоугольной таблице размерами $(|\pi|+2\rho)(2n+4\rho)$ в соответствии с матрицами инциденций подмножеств $Q_{B_e}\big|_{B_e \in \pi(\sigma_i)}$ и $\mathcal{X}_{C_m}\big|_{C_m = \sigma_i}$ в множествах Q и $\mathcal{X}$ соответственно.

Из этого следует сделать вывод, что если при синтезе автоматов можно ограничиться получением программ, близких к минимальным, то целесообразно применять кодирование, основанное на $\mathcal{K}$ - множествах.

ПРИЛОЖЕНИЕ

Алгоритмы определения кратчайших покрытий Φ_T и $\Psi_{T\Omega}$

I. Алгоритм определения Φ_T

Исходными являются таблица переходов автомата и системы двублочных разбиений T и Ω, соответствующие кодам внутренних и входных состояний. Конечным-покрытие Φ_T, удовлетворяющее условиям (16).

I. По таблице переходов строятся разбиения

$$\pi(\sigma_i) = \{ \mathcal{K}_{s_k}(\sigma_i) \}, \qquad (i = 1, 2, \ldots, z).$$

2. По системе двублочных разбиений T строится покрытие $P_T = \{B_m\}$ согласно (II).

3. Для каждого $\mathcal{K}_{s_k}(\sigma_i) \in \pi$ составляется подмножество $P_T(\sigma_i, s_k) \subset P_T$, такое что

а) $B_m \in P_T(\sigma_i, s_k)$, если $\mathcal{K}_{s_k}(\sigma_i) \geq B_m$ и

б) $B_n \in P_T(\sigma_i, s_k) \big|_{B_n \supset B_m}$.

4. Для каждого $\mathcal{K}_{s_k}(\sigma_i) \in \pi\ (s_k \in S)$ определяется кратчайшее его покрытие подмножествами $B_m \in P_T(\sigma_i, s_k)$ путем применения методики, аналогичной определению кратчайших д. н. ф. [7]

а) Составляется матрица $M_\varphi(s_k, \sigma_i)\ (s_k \in S, \sigma_i \in \Sigma)$, где строки соответствуют элементам S, а столбцы – подмножествам $P_T(s_k, \sigma_i)$:

$$m_{ij} = 1, \quad \text{если} \quad s_i \in B_j \big|_{B_j \in P_T(s_k, \sigma_i)},$$

$$m_{ij} = 0, \quad \text{если} \quad s_i \notin B_j \big|_{B_j \in P_T(s_k, \sigma_i)}.$$

Матрица $M_\varphi(s_k, \sigma_i)$ сокращается путем вычеркивания всех покрывающих строк.

б) Составляется конъюнктивная нормальная форма из элементов столбцовой матрицы, равной булеву матричному произведению

$$M'_\varphi(s_k, \sigma_i) \cdot \hat{P}_T(s_k, \sigma_i),$$

где $M'_\varphi(s_k, \sigma_i)$ – сокращенная матрица $M_\varphi(s_k, \sigma_i)$, а $\hat{P}(s_k, \sigma_i)$ – столбцовая матрица из элементов $P_T(s_k, \sigma_i)$.

в) Конъюнктивная нормальная форма преобразуется в д. н. ф. Любой из самых коротких членов может быть выбран в качестве подмножества $\{P_e(\sigma_i, s_k)\} \subset P_T(\sigma_i)$, составляющего наибольшее покрытие $\mathcal{K}_{s_k}(\sigma_i)$.

2. Алгоритм определения $\Psi'_{T\Omega}$

Исходными являются таблица переходов и системы двублочных

разбиений T и Ω, конечным – покрытие $\Psi_{T\Omega}$, удовлетворяющее условиям (27).

1. По таблице переходов строится множество $\pi = \{\pi(\sigma_1), \pi(\sigma_2), \ldots, \pi(\sigma_z)\}$.

2. По заданным T и Ω строятся покрытия $P_T = \{B_n\}$ и $P_\Omega = \{C_m\}$ согласно (II).

3. Для каждого $s_k \in S'$ образуется $P_{\pi_{s_k}}$ (25) и $P_{\pi\Omega}(s_k)$ (26).

4. Для каждого $s_k \in S$ из множества $[P_T \times P_\Omega]$ выделяется подмножество $P_{T\Omega}(s_k) = \{(B_n, C_m)\}$ такое, что

а) $\forall (B_n, C_m) \in P_{T\Omega}(s_k) \exists (\mathcal{Z}_j, C_m) \in P_{\pi\Omega}(s_k) \big|\ \mathcal{Z}_j \geq B_n$,

б) $\exists (B_p, C_q) \in P_{T\Omega}(s_k) \big|\ (B_p \supset B_n;\ C_q \supset C_m)$.

5. Для каждого $P_{\pi\Omega}(s_k)\ (s_k \in S)$ определяется кратчайшее его покрытие подмножествами $(B_n, C_m) \in P_{T\Omega}(s_k)$ путем применения методики, аналогичной изложенной в [7]:

а) Составляется матрица $M_\psi(s_k)(s_k \in S)$, где строки соответствуют элементам $[S \times \Sigma]_{s_k}$, а столбцы – подмножествам $P_{T\Omega}(s_k)$.

$$m_{ij} = 1, \quad \text{если} \quad (s_p, \sigma_q) \in (B_n, C_m) \big|_{(B_n, C_m) \in P_{T\Omega}(s_k)};$$

$$m_{ij} = 0, \quad \text{если} \quad s_p \notin B_n,\ \sigma_q \notin C_m.$$

б) Матрица $M_\psi(s_k)$ сокращается путем вычеркивания всех покрывающих строк. Составляется конъюнктивная нормальная форма (к.н.ф) из элементов столбцовой матрицы, равной булеву матричному произведению

$$M'_\psi(s_k) \cdot \hat{P}_{T\Omega}(s_k),$$

где $M'_\psi(s_k)$ – сокращенная матрица $M_\psi(s_k)$ и $\hat{P}_{T\Omega}(s_k)$ – столбцовая матрица из элементов $P_{T\Omega}(s_k)$.

в) К.н.ф. преобразуется в д.н.ф. Любой из самых коротких членов д.н.ф. может быть выбран в качестве $\Psi_{T\Omega}(s_k)$.

П р и м е р. Построить оптимальную программу настройки для автомата, заданного таблицей переходов.

$\delta =$	σ_1	σ_2	σ_3	σ_4
a	$\boxed{a}$	c	d	d
b	a	c	$\boxed{b}$	$\boxed{b}$
c	a	$\boxed{c}$	b	b
d	a	c	$\boxed{d}$	$\boxed{d}$
e	a	$\boxed{e}$	d	$\boxed{e}$
g	$\boxed{g}$	e	d	e

Входные и внутренние состояния автомата закодированы в соответствии с системами двублочных разбиений :

$$\Omega = \begin{cases} \omega_1 = \{\overline{\sigma_1, \sigma_2};\ \overline{\sigma_3, \sigma_4}\}, \\ \omega_2 = \{\overline{\sigma_1, \sigma_3};\ \overline{\sigma_2, \sigma_4}\}. \end{cases}$$

$$T = \begin{cases} \tau_1 = \{\overline{a, c, d};\ \overline{b, e, g}\}, \\ \tau_2 = \{\overline{a, d, e, g};\ \overline{b, c}\}, \\ \tau_3 = \{\overline{a, b, c, e};\ \overline{d, g}\}. \end{cases}$$

1) Столбцовые разбиения.

$$\pi(\sigma_1) = \{\overline{b, c, d, e, a};\ \overline{g}\};$$
$$\pi(\sigma_2) = \{\overline{a, b, d, c};\ \overline{g, e}\}.$$
$$\pi(\sigma_3) = \{\overline{a, e, g, d},\ \overline{c, b}\};$$
$$\pi(\sigma_4) = \{\overline{a, d},\ \overline{c, b};\ \overline{g, e}\}.$$

2) $P_T = \{a, c, d;\ b, e, g;\ a, d, e, g,\ b, c;\ a, b, c, e,\ d, g,\ a, d, e, g;\ a, c;\ b, e;\ a, e;\ a, b,\ c,\ d;\ e,\ g\}.$

$P_\Omega = \{\sigma_1, \sigma_2;\ \sigma_3, \sigma_4;\ \sigma_1, \sigma_3;\ \sigma_2, \sigma_4;\ \sigma_1;\ \sigma_2;\ \sigma_3,\ \sigma_4\}$

3) $P_{\pi\Omega}(a) = \{(b, c, d, e, a;\ \sigma_1)\},$

$P_{\pi\Omega}(b) = \{(c, b, \sigma_3), (c, b, \sigma_4);\ (c, b; \sigma_3\ \sigma_4)\}$

$P_{\pi\Omega}(c) = \{(a, b, d, c;\ \sigma_2)\};$

$P_{\pi\Omega}(d) = \{(a, e, g, d, \sigma_1), (a, d;\ \sigma_3, \sigma_4)\};$

$P_{\pi\Omega}(e) = \{(g, e;\ \sigma_2, \sigma_4)\};$

$P_{\pi\Omega}(g) = \{(g;\ \sigma_1)\}.$

4) $P_{T\Omega}(a) = \{(a, c, d;\ \sigma_1); (a, b, c, e,\ \sigma_1)\};$

$P_{T\Omega}(b) = \{(c, b;\ \sigma_3, \sigma_4)\};$

$P_{T\Omega}(c) = \{(a, c, d;\ \sigma_2);\ (b, c;\ \sigma_2)\};$

$P_{T\Omega}(d) = \{(a, e, g, d;\ \sigma_1); (a, d;\ \sigma_3, \sigma_4)\};$

$P_{T\Omega}(e) = \{(g, e;\ \sigma_2, \sigma_4)\};$

$P_{T\Omega}(g) = \{(g;\ \sigma_1)\}.$

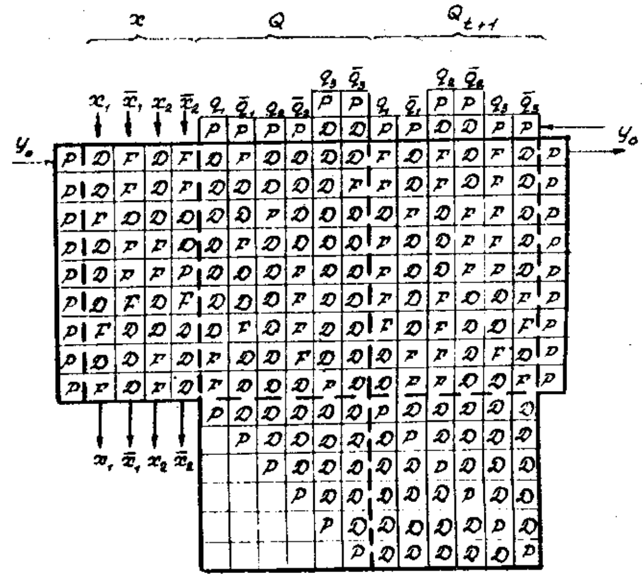


Рис. 3

5) В этом простом примере все покрытия $P_{T\Omega}(S_\kappa)$ являются кратчайшими покрытиями $P_{\Pi\Omega}(S_\kappa)$, т.е.

$$P_{T\Omega}(S_\kappa) = \Psi_{T\Omega}(S_\kappa); \quad \Psi_{T\Omega} = \bigcup_{\kappa=1}^{\upsilon} \Psi_{T\Omega}(S_\kappa); \quad \Psi_{T\Omega} = 9.$$

6) Матрица R строится в соответствии с правилами §4 (рис.3). Сложность реализации автомата равна, согласно (9),

$$N = 15 \times 16.$$

ЛИТЕРАТУРА

1. Э.В. ЕВРЕИНОВ, Ю.Г. КОСАРЕВ. Однородные вычислительные системы высокой производительности, Новосибирск, 1966.
2. J.HARTMANIS, R.STEARNS. Algebraic structure of Theory of Sequential Machines. Prentice Hall Inc.,1966.
3. О.Л. БАНДМАН. Реализация автоматов в криотронной вычислительной среде.—Труды I Всесоюзной конференции по вычислительным системам. Новосибирск, 1968, вып.3.
4. О.Л. БАНДМАН. Реализация автоматов в криотронной вычислительной среде по заданному графу состояний.— Вычислительные системы, Новосибирск, 1969, вып.33.
5. J.H.TRACEY. Internal State Assignments for Asynchronous Sequential Machines. IEEE Transactions, EC-15, N 4, August 1966.
6. В.В. ГЛАГОЛЕВ. Верхняя оценка сложности минимальной д.н.ф. для почти всех функций алгебры и логики.-Дискретный анализ, Новосибирск, 1965, вып.5, стр.3 - 9.
7. E.J.McCLUSKEY. Minimization of Boolean Functions. Bell Sys. Tech. vol. 35, pp. 1417-1444. November 1956.
8. А.Д. ЗАКРЕВСКИЙ, А.Е. ЯНКОВСКАЯ. Практические алгоритмы кодирования внутренних состояний асинхронных автоматов.- Автоматика и вычислительная техника, 1969, №3, 15-21.

Поступила в редакцию
5.I.1970 г.