

УДК 681.3.001

ПРОЕКТИРОВАНИЕ АВТОМАТОВ УПРАВЛЕНИЯ НА ОСНОВЕ БИС

С.М.Ачасова, О.Л.Бандман

Развитие технологии БИС меняет не только внешний облик и технические параметры цифрового устройства, но и ставит по-новому задачи технического проектирования. Это касается, в основном, тех случаев, когда в БИС реализуются сложные логические функции или автоматы, заданные в виде булевых уравнений, таблиц переходов или граф-схем алгоритмов. Такие БИС применяются обычно в устройствах управления вычислительных или управляющих устройств, проектируются по специальным заказам и получили название "заказных" БИС.

Традиционные способы проектирования такой заказной БИС состоят из ряда сложных и трудоемких процедур:

- 1) преобразования заданного описания в булево выражение и приведения его к оптимальному в принятом базисе виду;
- 2) перехода к многоуровневой вентиляльной схеме (логической сети);
- 3) размещения вентиляей на площади кристалла и трассировки межвентильных соединений.

Известно много хорошо разработанных методов как для ручного, так и для автоматизированного выполнения этих процедур. Однако в целом такая методология, разработанная для логического проектирования на элементах малой интеграции, не учитывает основных особенностей групповой технологии и поэтому не дает оптимальных (с точки зрения общей стоимости) решений. Хотя критерий оптимизации в более или менее точном виде установить не удается, происходит постепенная переоценка требований к логическим схемам, реализуемым в виде БИС, которая выражается в следующем.

1. Стоимость соединений в БИС сравнима, а во многих случаях превосходит стоимость активных компонентов. В связи с этим критерий сложности по числу логических вентиляей на отражает общей стоимости изготовления. Очевидно, более реальным критерием является площадь кристалла, которую занимает схема.

2. Стоимость проектирования соизмерима, а для заказных БИС превосходит стоимость изготовления. Следовательно, оптимизация процедур проектирования в большей степени влияет на общую стоимость, чем минимизация логической схемы.

3. Условия тестирования должны учитываться на этапе проектирования, поскольку те ограничения на возможные отказы, которые приняты в большинстве методов генерации тестов (единичные отказы типа констант на входах и выходах) для БИС не всегда приемлемы.

Отсюда следует вывод, что проектирование БИС следует рассматривать как единую логико-топологическую задачу, в которой по заданному алгоритму функционирования схемы определяется её топологический рисунок на кристалле. Очевидно, решение этой задачи возможно, если процедура синтеза использует представление логических функций, отражающее геометрию схемы на плоскости. Таким является представление логических условий в д.н.ф., выраженных в матричной форме с последующим отображением матрицы на прямоугольную координатную сетку кристалла. Аналогичные представления функций положены в основу ряда методов логического синтеза в вычислительных средах [1,2,3]. Подобный прием применяется при проектировании автоматов управления микрокалькуляторов, реализуемых в виде МОП-БИС [4]. На этом же принципе основано создание так называемых "программируемых логических матриц", используемых для реализации микропрограммного управления в мини-машинах [5,6] и успешно вытесняющих заказные БИС [7].

При таком подходе из процесса проектирования исключаются этапы размещения вентиляей и трассировки связей между ними. Это существенно сокращает затраты на проектирование, а в случае программируемых логических матриц и на изготовление, так как переход от одной системы функций к другой при том же множестве переменных производится путем внесения изменений только в одну технологическую операцию.

Применение таких "матричных" методов до сих пор ограничивалось этапом проектирования, который соответствует переходу от булевых уравнений к реализующим их схемам. Все остальные этапы (минимизация числа состояний, кодирование состояний, минимизация функций переходов) производились традиционными способами, часто не соответствующими новым требованиям. В связи с этим возникла необходимость в создании полной системы методов синтеза с учетом требований реализации в виде БИС.

В настоящей статье дается теоретическое обоснование матричного метода, основанное на представлении автоматов в виде  $(0,1)$ -матриц, которые несут в себе информацию как о логическом его функционировании, так и схемотехнической реализации. Задача получения оптимального решения поставлена следующим образом: критерием оптимизации является площадь кристалла, необходимая для размещения схемы, а в качестве ограничения приняты способы выражения функций и отображения схемы на кристалл.

Предлагается метод получения оптимального в указанном смысле кода внутренних состояний, который использует аппарат алгебры разбиений на множестве состояний и представление множества импликант булевой функции в виде алгебраической структуры. На основе предложенного метода создана система алгоритмов и программы, позволяющих получить матрицы реализации автомата по заданным таблицам переходов или граф-схемам алгоритмов.

### § I. Матричный способ реализации автомата

Будем рассматривать 2 способа исходного задания автомата.

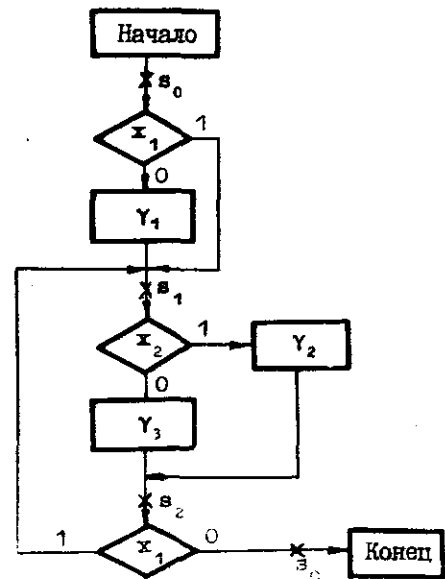
1. Классический способ задания в виде таблицы переходов и выходов (рис. I, а), который удобен для небольших полностью (или почти полностью) определенных автоматов.

2. Способ задания микропрограммы автоматов на языке граф-схем алгоритмов (ГСА) (рис. I, б), который применяется на практике при проектировании автоматов управления ЭМ и устройств промышленной автоматики. Этот способ может использоваться для автоматов большой размерности. Методы преобразования ГСА в переходы к автоматному заданию хорошо разработаны в [9].

Можно считать, что этим исчерпывается большая часть существующих форм представлений синхронных автоматов, поскольку первый способ включает в себя описания в виде графов состояний матриц переходов, второй - есть графическое выражение логических схем алгоритмов.

а)

	00	01	10	11	00	01	10	11
$s_0(00)$	$s_1$	$s_1$	$s_2$	$s_2$	$Y_1$	$Y_1$	$Y_3$	$Y_2$
$s_1(01)$	$s_2$	$s_2$	$s_2$	$s_2$	$Y_3$	$Y_2$	$Y_3$	$Y_2$
$s_2(1-)$	$s_0$	$s_0$	$s_2$	$s_2$	-	-	$Y_3$	$Y_2$



в)

Исходное состояние	Состояние перехода	Входное состояние	Выходное состояние
$s_2$	$s_0$	$\bar{x}_1$	-
$s_0$	$s_1$	$\bar{x}_1$	$Y_1$
$s_0$	$s_2$	$x_1 \bar{x}_2$	$Y_3$
$s_0$		$x_1 x_2$	$Y_2$
$s_1$		$\bar{x}_2$	$Y_3$
$s_1$		$x_2$	$Y_2$
$s_2$		$x_1 \bar{x}_2$	$Y_3$
$s_2$		$x_1 x_2$	$Y_2$

Рис. I. Способы исходного задания автомата

В любом случае задание автомата содержит в себе множества входных  $\Sigma = \{\sigma_1, \dots, \sigma_n\}$ , внутренних  $S = \{s_1, \dots, s_v\}$  и выходных  $\Gamma = \{\gamma_1, \dots, \gamma_w\}$  состояний, которые выражаются наборами значений соответствующих переменных  $X = \{x_1, \dots, x_n\}$ ,  $Q = \{q_1, \dots, q_p\}$  и  $Z = \{z_1, \dots, z_m\}$ . Функционирование автомата определяется отображениями  $\delta: \Sigma \times S \rightarrow S$  и  $\lambda: \Sigma \times S \rightarrow \Gamma$ . При задании автомата в виде таблицы переходов отображения  $\delta$  и  $\lambda$  задаются в табличном виде для всех значений множеств  $S$  и  $\Sigma$ . При задании в виде ГСА получение явной формы  $\delta$  и  $\lambda$  будем производить по [9] путем перехода к так называемой "обратной таблице переходов" (рис. 1в). В этой таблице каждая строка соответствует одному переходу автомата. В ней содержится информация, в какое состояние из какого переходит автомат при каких значениях входов и выходов. Таблица составляется по отмеченной ГСА. "Отметки" ставятся на входах вершин, следующих за операторными, и обозначают состояния автомата Мили. Значения входов при переходах соответствуют путям через условные вершины от отметки до отметки.

Введем в рассмотрение множество условий переходов  $\Phi = \{\varphi_1, \dots, \varphi_\mu\}$ . Каждое  $\varphi_i$  соответствует одному или группе полных состояний, приводящих автомат в одни и те же состояния перехода и выхода. В терминах "обратной таблицы переходов"  $\varphi_i$  соответствует строке таблицы. Условие перехода выражается одной конъюнкцией вида

$$\varphi_i = (X, Q) = \tilde{x}_{j_1} \cdot \dots \cdot \tilde{x}_{j_k} \cdot \bar{q}_{j_1} \cdot \dots \cdot \bar{q}_{j_k}, \quad (1)$$

где подмножества внутренних и входных переменных, а также прямой ( $\tilde{x}_i = x$ ) или инверсный ( $\tilde{x}_i = \bar{x}_i$ ) их вид определяются исходным и входным состояниями. Объединим входные и внутренние переменные и их инверсии в множество  $Y = \{x_1, \bar{x}_1, x_2, \bar{x}_2, \dots, q_1, \bar{q}_1, \dots, q_p, \bar{q}_p\} = \{y_1, y_2, \dots, y_{2n+2p}\}$ . Тогда уравнения переходов и выходов можно будет выразить в виде систем функций:

$$\theta_j(t+1) = \bigvee_{\varphi_i \in \Phi_j} \varphi_i(Y), \quad j = 1, 2, \dots, p, \quad (2)$$

$$z_j = \bigvee_{\varphi_i \in \Psi_j} \varphi_i(Y), \quad j = 1, 2, \dots, m. \quad (3)$$

Здесь  $\Phi_j \subset \Phi$  включает в себя все условия переходов, в результате которых внутренняя переменная  $q_j$  принимает значение "1". Аналогично  $\Psi_j \subset \Phi$  содержит условия переходов, при которых  $x_j = 1$ .

Введем в рассмотрение следующие матрицы:

1)  $M$  - прямоугольная (0,1)-матрица порядка  $[v \times (3p + 2n + m)]$ , которую будем рассматривать как клеточную матрицу

$$M = \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}$$

и называть "матрицей реализации автомата". Строки матрицы  $M$  соответствуют элементам множества  $\Phi$ ; столбцы - элементам  $Y$ ,  $Q$  и  $Z$ , образуя подматрицы  $M_1 = \|m_{ij}^1\|$ ,  $M_2 = \|m_{ij}^2\|$  и  $M_3 = \|m_{ij}^3\|$  соответственно

$$\left. \begin{aligned} m_{ij}^1 &= 1, & \text{если } y_j \in \varphi_i; \\ m_{ij}^2 &= 1, & \text{если } \varphi_i \in \Phi_j; \\ m_{ij}^3 &= 1, & \text{если } \varphi_i \in \Psi_j; \end{aligned} \right\} \quad (4)$$

в остальных случаях  $m_{ij} = 0$ .

2)  $Y$  - столбцовая матрица из элементов  $Y$ .

3)  $Q$  - строчная матрица из элементов  $Q$ .

4)  $Z$  - строчная матрица из элементов  $Z$ .

Тогда системы функций (2) и (3) выразятся в матричной форме:

$$Q = (\overline{M_1 \cdot Y})^T \cdot M_2; \quad (5)$$

$$Z = (\overline{M_1 \cdot Y})^T \cdot M_3. \quad (6)$$

Операция отрицания над матрицами обозначает инверсию всех её элементов, а умножение матриц производится по правилам матричного умножения с заменой арифметических операций одноименными логическими. Знак  $T$  обозначает транспонирование матрицы. Переход от абстрактного задания автомата к матричному чрезвычайно прост и может быть выполнен по следующим правилам.

I. Если автомат задан таблицей, матрицей или графом переходов, то построение  $M_1$  заключается в расстановке единиц в её строках в соответствии с кодами входного и внутреннего состояний для каждой клетки таблицы или дуги графа. Если автомат задан в виде ГСА, то каждой строке  $M_1$  ставится в соответствие путь через условные вершины. Единицы в этой строке расставляют-

ся на основании обозначений условий на дугах этого пути и кода внутреннего состояния, из которого путь исходит. Порядок расположения строк в  $M_1$  соответствует порядку возрастания номеров внутренних состояний, в которые переходит автомат при соответствующем  $\varphi_i$ . Внутри подмножеств  $\Phi$ , приводящих автомат в то же состояние, порядок строк - произвольный.

Переход	Условие перехода	Выходное состояние
$s_2 \rightarrow s_0$	$q_1 \bar{x}_1$	$\bar{z}_1 \bar{z}_2$
$s_0 \rightarrow s_1$	$\bar{q}_1 \bar{q}_2 \bar{x}_1$	$z_1 z_2$
$s_0 \rightarrow s_2$	$\bar{q}_1 \bar{q}_2 x_1 \bar{x}_2$	$\bar{z}_1 z_2$
$s_0 \rightarrow s_2$	$\bar{q}_1 \bar{q}_2 x_1 x_2$	$z_1 \bar{z}_2$
$s_1 \rightarrow s_2$	$\bar{q}_1 q_2 \bar{x}_1$	$\bar{z}_1 z_2$
$s_1 \rightarrow s_2$	$\bar{q}_1 \bar{q}_2 x_2$	$z_1 \bar{z}_2$
$s_2 \rightarrow s_2$	$q_1 x_1 \bar{x}_2$	$\bar{z}_1 z_2$
$s_2 \rightarrow s_2$	$q_1 x_1 x_2$	$z_1 \bar{z}_2$

а)

$$C_S = \begin{matrix} & q_1 & q_2 \\ s_0 & \begin{matrix} 0 & 0 \\ 0 & 1 \end{matrix} \\ s_1 & \begin{matrix} 1 & - \end{matrix} \\ s_2 & \begin{matrix} - & - \end{matrix} \end{matrix}$$

$$C_T = \begin{matrix} & z_1 & z_2 \\ Y_1 & \begin{matrix} 1 & 1 \\ 1 & 0 \\ 0 & 1 \end{matrix} \\ Y_2 & \\ Y_3 & \end{matrix}$$

б)

$$M = \begin{matrix} & x_1 & \bar{x}_1 & x_2 & \bar{x}_2 & q_1 & \bar{q}_1 & q_2 & \bar{q}_2 & q_1 & q_2 & z_1 & z_2 \\ \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} & \begin{matrix} \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} 1 \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} & \begin{matrix} \\ \\ \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{matrix} \end{matrix}$$

в)

Рис. 2. Условия переходов, кодирующие таблицы и матрица реализации автомата А

2. Матрица  $M_2$  может быть получена из кодирующей матрицы  $C_S$  путем повторения каждой её строки  $\nu_k$  раз, где  $\nu_k$  - число условий  $\varphi_i \in \Phi$ , приводящих автомат в состояние  $s_k$ .

3. Матрица  $M_3$  содержит строки кодирующей матрицы  $C_T$  и строится согласно (4).

Например, для автомата, таблица переходов и ГСА которого изображены на рис.1, множество условий перехода, кодирующие таблицы и матрица реализации приведены на рис.2.

Как нетрудно видеть, операция матричного умножения с последующей инверсией соответствует выражению функции в базисе ИЛИ-НЕ, и, следовательно дальнейшей её реализации в виде параллельного соединения транзисторов. Матрицу  $M$  можно рассматривать как условное отображение принципиальной схемы и топологического рисунка, причем распределение единиц и нулей в ней определяет расположение транзисторов в схеме. Переход от матрицы к принципиальной схеме, которая является одновременно эскизом топологии, осуществляется следующим образом.

1. Столбцам  $M_1$  ставятся в соответствие вертикальные шины, по которым подаются переменные  $Y$  к затворам транзисторов.

2. Строкам  $M_1$  ставятся в соответствие пары горизонтальных шин ("земля", "выход"), к которым присоединяются стоки и истоки транзисторов, образующих вентиль "ИЛИ-НЕ", реализующий одно из условий перехода (для каждой пары строк шина "земля" может быть общей).

3. Транзисторы ставятся в местах, соответствующих единицам в  $M_1$ .

4. Часть принципиальной схемы, определяемая матрицами  $M_2$  и  $M_3$ , аналогична  $M_1$ , но повернута относительно нее на  $90^\circ$ , о чем символизирует знак транспонирования в формулах (5,6).

5. Построение схемы завершается обозначениями цепей питания, "земли" и тактирующих импульсов. Для автомата, приведенного на рис. 1 и 2, принципиальная схема изображена на рис. 3.

Площадь, необходимая для реализации автомата на кристалле, определяется размерами матрицы  $M$ . Выраженная в условных квадратах со стороной, равной шагу сетки на кристалле, она равна

$$L_A = (1,5\mu + c_1)(2n + 3,5\rho + 1,5m + c_2), \quad (7)$$

где  $\mu$  - число условий перехода,  $n$ ,  $\rho$  и  $m$  - числа входных, внутренних и выходных переменных,  $c_1$  и  $c_2$  - константы, не за-

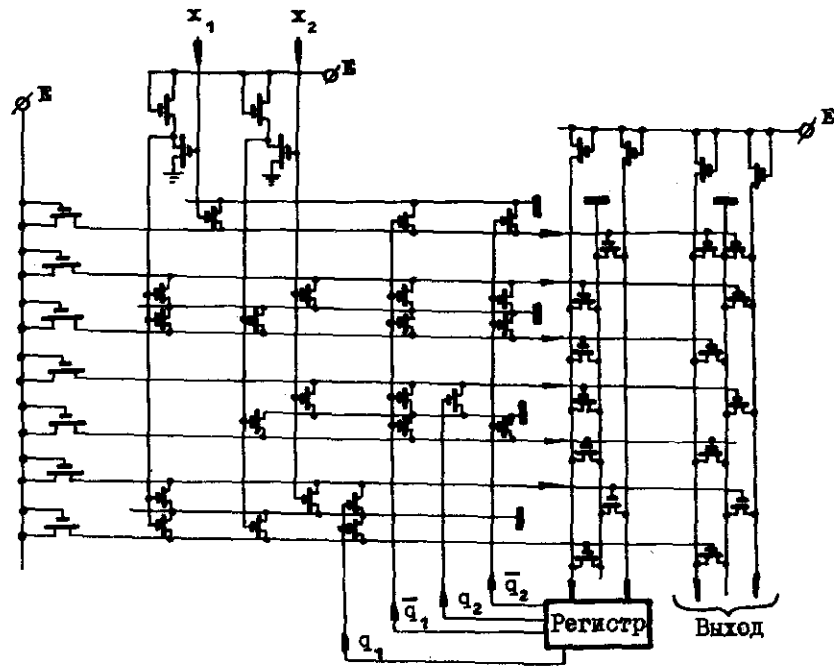


Рис. 3. Принципиальная схема автомата А

висящие от размеров автомата и учитывающие технологическое оформление схемы (инверторы, нагрузка, питание и т.д.).

Задачу оптимизации будем понимать как поиск таких представлений автомата, для которых матрица имеет наименьшие размеры. Если коды состояний автомата заданы, то эта задача сводится к определению кратчайших д.н.ф., выражающих условия перехода в одни и те же внутренние и выходные состояния. Если коды состояний автомата не заданы, то их следует составить таким образом, чтобы они позволили выразить условия переходов наименьшим числом интервалов в пространстве переменных и таким образом уменьшить число строк в матрице  $M$ .

## § 2. Математическая формулировка задачи оптимального кодирования

Будем предполагать, что код входных состояний  $C_2$  известен и что для каждого перехода функции входных переменных выражены в неизбыточной форме. Следовательно, функции (2) и (3) могут сокращаться за счет объединения условий переходов  $\varphi_i$  (I) только по внутренним переменным.

**ОПРЕДЕЛЕНИЕ 1.**  $K$  - множеством будем называть такое подмножество внутренних состояний, из которых при одинаковых значениях входов автомат переходит в одни и те же внутреннее и выходное состояния. Покрытие множества  $S$ , состоящее из всех  $K$ -множеств, будем обозначать  $P_{\pi} = \{K_1, \dots, K_m\}$ .

Очевидно, что сокращение д.н.ф. (2) и (3) возможно только путем склеивания внутренних состояний, принадлежащих одному и тому же  $K$ -множеству.

**ОПРЕДЕЛЕНИЕ 2.** Оптимальным кодом внутренних состояний автомата будем называть код, который обеспечивает покрытие  $K$ -множеств наименьшим суммарным числом интервалов и, следовательно, позволяет строить матрицу  $M$  с наименьшим числом строк  $\mu_{min}$ .

Задачу поиска оптимального кода будем формулировать в терминах алгебры разбиений, введенной в [10]. Операциями в этой алгебре являются:

1. умножение разбиений  $\theta = \theta_1 \cdot \theta_2$ , каждый блок произведения  $\theta$  является пересечением пары блоков из  $\theta_1$  и из  $\theta_2$ ;
2. сложение разбиений  $\theta = \theta_1 + \theta_2$ , каждый блок суммы  $\theta$  является объединением пары блоков из  $\theta_1$  и  $\theta_2$ , пересечение которых пусто.

Эти операции устанавливают на множестве разбиений следующее отношение порядка

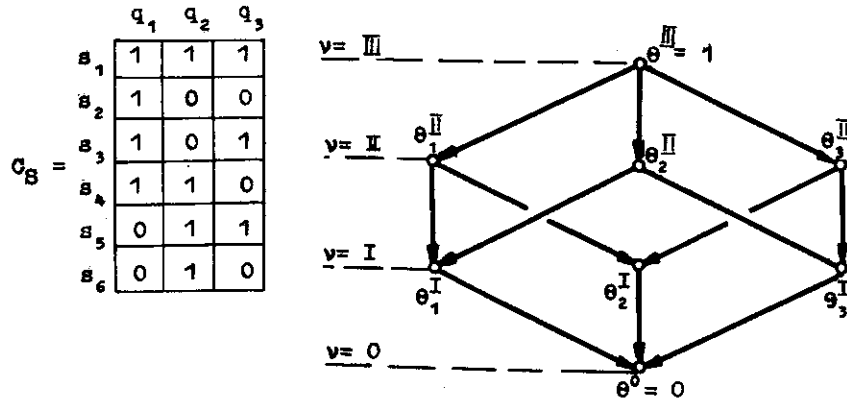
$$\begin{aligned} \theta_1 \leq \theta_2, & \text{ если } \theta_1 \cdot \theta_2 = \theta_1; \\ \theta_1 \geq \theta_2, & \text{ если } \theta_1 + \theta_2 = \theta_1. \end{aligned}$$

Существуют 2 тривиальных разбиения:  $\theta = \{\bar{s}_1, \bar{s}_2, \dots, \bar{s}_y\} = I$ , когда все состояния входят в один блок; и  $\theta = \{s_1, s_2, \dots, s_y\} = 0$ , когда в каждый блок входит только одно состояние.

Двоичный код  $C_3$  будем определять множеством двублочных разбиений  $T = \{\tau_1, \dots, \tau_p\}$  на множестве внутренних состояний  $S$ . Каждое  $\tau_j = \{S_j^0; S_j^1\}$  соответствует одной из внутренних пере-

менных  $q_j \in Q$ , причем  $S_j^0$  объединяет те состояния, в которых  $q_j = 0$ , а  $S_j^1$  — те состояния, в которых  $q_j = 1$ .

Всевозможные произведения  $\tau$  — разбиений  $\theta_i^y = \tau_{i_1} \cdot \tau_{i_2} \cdot \dots \cdot \tau_{i_k}$  ( $k = \rho - y$ ;  $y = 0, 1, 2, \dots, \rho$ ) составляют частично упорядоченное множество  $\theta = \{\theta_i^y\}$  ( $i = 1, 2, \dots, C_\rho^y$ ;  $y = 1, 2, \dots, \rho$ ), обладающее свойствами дистрибутивной структуры с дополнениями (булевой алгебры) [II]. Так, например, двоичный код  $C_3$  для автомата с шестью состояниями и порождаемая этим кодом структура  $\theta$  изображена на рис. 4.



$$\theta^{\text{III}} = \{\overline{s_1, s_2, s_3, s_4, s_5, s_6}\} = 1,$$

$$\theta_1^{\text{II}} = \tau_1 = \{\overline{s_1, s_2, s_3, s_4}; \overline{s_5, s_6}\};$$

$$\theta_2^{\text{II}} = \tau_2 = \{\overline{s_1, s_4, s_5, s_6}; \overline{s_2, s_3}\};$$

$$\theta_3^{\text{II}} = \tau_3 = \{\overline{s_1, s_3, s_5}; \overline{s_2, s_4, s_6}\};$$

$$\theta_1^{\text{I}} = \{\overline{s_1, s_4}; \overline{s_2, s_3}; \overline{s_5, s_6}\};$$

$$\theta_2^{\text{I}} = \{\overline{s_1, s_3}; \overline{s_2, s_4}; \overline{s_5}; \overline{s_6}\};$$

$$\theta_3^{\text{I}} = \{\overline{s_1, s_5}; \overline{s_4, s_6}; \overline{s_2}; \overline{s_3}\};$$

$$\theta^0 = \{\overline{s_1}; \overline{s_2}; \overline{s_3}; \overline{s_4}; \overline{s_5}; \overline{s_6}\}.$$

Рис. 4. Кодировочная таблица и порождаемая ею структура

Каждое разбиение  $\theta_j^y$  состоит из  $2^{\rho-y}$  блоков

$$\theta_j^y = \{B_1, \dots, B_{2^{\rho-y}}\}, \quad (8)$$

которые образуют в  $\rho$ -мерном пространстве внутренних переменных  $Q$   $y$ -мерные подкубы. Множество блоков всех разбиений булевой структуры

$$P_T = \{B_1, \dots, B_{3^p}\}$$

перечисляет всевозможные подмножества  $S$ , образующие интервалы в пространстве переменных  $Q$ . Следовательно, поиск оптимального кода можно свести к построению такой структуры  $\theta$ , блоки разбиений которой образуют покрытия  $K$ -множеств автомата наименьшей суммарной длины.

Перечислим свойства структуры  $\theta$ , необходимые для обоснования алгоритма определения оптимального кода. Для определенности будем считать, что  $|S| = v = 2^y$ .

1. Произведение всех двублочных разбиений равно тривиальному разбиению, каждый блок которого содержит точно по одному состоянию

$$\prod_{j=1}^{\rho} \tau_j = \{\overline{s_1}; \overline{s_2}; \dots; \overline{s_v}\} = 0. \quad (9)$$

Это разбиение является "нулевым" элементом структуры. Условие (9) обеспечивает различие кодирующих слов внутренних состояний автомата.

2. "Единицей" структуры является тривиальное разбиение, имеющее один блок

$$\theta^{\rho} = \{\overline{s_1, s_2, \dots, s_v}\} = 1. \quad (10)$$

3. Разбиения  $\theta_j^y$ , образованные произведением  $(\rho - y)$  и  $\tau$ -разбиений, составляют  $y$ -й уровень структуры  $\theta$

$$\theta_j^y = \tau_{i_1} \cdot \dots \cdot \tau_{i_{\rho-y}}, \quad (j = 1, 2, \dots, C_\rho^y).$$

4. Любое разбиение  $y$ -го уровня является произведением двух разбиений  $(y+1)$ -го уровня и в то же время суммой двух разбиений  $(y-1)$ -го уровня:

$$\theta_k^y = \theta_i^{y+1} \cdot \theta_j^{y+1} = \theta_m^{y-1} + \theta_n^{y-1} \quad (II)$$

( $k = \{1, 2, \dots, C_\rho^y\}$ , ( $i, j$ )  $\in \{1, 2, \dots, C_\rho^{y+1}\}$ , ( $m, n$ )  $\in \{1, 2, \dots, C_\rho^{y-1}\}$ ).

5. Геометрической интерпретацией структуры  $\theta$  является  $r$ -мерный куб. Вершинам куба поставлены в соответствие разбиения. На одной из главных диагоналей куба лежат  $\theta^0$  и  $\theta^p$ . Ребра куба соединяют вершины, связанные соотношениями (II), и направлены от вершин, соответствующих более крупным разбиениям к более мелким. Разбиения  $y$ -го уровня находятся на расстоянии  $y$  от нулевой вершины.

6. Для каждого  $\theta_j^y$  в множестве  $\theta$  существует другое (причем единственное) разбиение  $\theta_j^{p-y}$ , такое что

$$\left. \begin{aligned} \theta_j^y + \theta_j^{p-y} &= 1, \\ \theta_j^y \cdot \theta_j^{p-y} &= 0. \end{aligned} \right\} \quad (I2)$$

Разбиение  $\theta_j^{p-y}$  называется дополнением для  $\theta_j^y$  и обозначается  $\theta_j^y = \theta_j^{p-y}$ .

7. Множество двублочных разбиений  $T = \{\tau_1, \dots, \tau_\rho\}$  составляет  $(\rho - 1)$ -й уровень структуры

$$\tau_j = \theta_j^{p-1} \quad (j = 1, 2, \dots, \rho). \quad (I3)$$

Дополнениями к  $\tau$ -разбиениям являются разбиения первого уровня, блоки которых содержат по 2 состояния, и поэтому дальше мы их будем называть парами

$$\tau_j = \theta_j^{p-1} = \overline{\theta_{\rho-j}^1}. \quad (I4)$$

8. Каждое разбиение  $y$ -го уровня можно представить в виде суммы  $y$  разбиений I-го уровня

$$\sum_{k=1}^y \theta_{i_k}^I = \theta_j^y = \{B_1^y, \dots, B_\ell^y, \dots, B_{2^{p-y}}^y\}, \quad (I5)$$

где  $B_\ell^y$  содержит по  $2^y$  состояний, объединяя по  $2^{y-1}$  пар из разбиений  $\theta_{i_k}^I$ . Иными словами, подмножество  $\{\theta_{i_1}^I, \dots, \theta_{i_k}^I, \dots, \theta_{i_y}^I\} = \hat{\theta}^I$  порождает на каждом  $\theta_{i_k}^I \in \theta^I$  разбиение множества пар на  $2^{p-y}$  подмножеств  $\{C_1^{i_k}, \dots, C_k^{i_k}, \dots, C_{2^{p-y}}^{i_k}\}$ , которые мы будем называть коллекциями пар.

9. Каждую коллекцию  $C_k^j$  в разбиении  $\theta_{i_k}^I$  можно рассматривать как одно разбиение I-го уровня такой подструктуры, наибольшим элементом которой ("единицей") является соответствующий ей блок  $B_k^y \in \theta_j^y$ . А множество коллекций  $\{C_k^{i_1}, \dots, C_k^{i_2}, \dots, C_k^{i_y}\}$  из всех  $\theta_{i_k}^I \in \theta$ , объединяющих пары, включенные в один и тот же блок  $B_k^y$ , образуют первый уровень этой подструктуры.

10. Множество разбиений какого-либо одного уровня полностью определяет всю структуру.

Из свойств (7-10) следует, что поиск  $\tau$ -разбиений, определяющих код, а также подмножеств состояний, образующих интервалы (подкубы) в пространстве внутренних переменных, можно свести к определению множества разбиений первого уровня. При этом пары состояний следует выбирать таким образом, чтобы они, объединяясь на следующих уровнях в четверки, восьмерки и т.д., обеспечивали кратчайшее покрытие  $K$ -множеств. Кроме того, надо быть уверенным, что полученное в результате принятой процедуры множество разбиений на пары действительно порождает булеву алгебру, т.е. удовлетворяет необходимым и достаточным условиям того, что  $\rho$  разбиений на пары составляют множество разбиений I-го уровня структуры, обладающей перечисленными свойствами. Чтобы установить эти условия, введем следующее

ОПРЕДЕЛЕНИЕ 3. Две пары  $B_p = \{s_\alpha, s_\beta\}$  и  $B_q = \{s_\gamma, s_\delta\}$  из одного и того же разбиения  $\theta_j^I$  будем называть сопряженными по  $\theta_i^I$ , если в  $\theta_i^I$  существуют  $B_k$  и  $B_\ell$  такие, что

$$\{B_k, B_\ell\} \cdot \{B_p, B_q\} = \{\bar{s}_\alpha; \bar{s}_\beta; \bar{s}_\gamma; \bar{s}_\delta\},$$

$$\{B_k; B_\ell\} + \{B_p; B_q\} = \overline{\{s_\alpha; s_\beta; s_\gamma; s_\delta\}}.$$

Например,  $(s_1, s_2)$  и  $(s_3, s_4)$  являются сопряженными по  $\theta_i^I$ , если в  $\theta_i^I$  существуют  $(s_1, s_3)$  и  $(s_2, s_4)$  или  $(s_1, s_4)$  и  $(s_2, s_3)$ .

Операцию определения сопряженной пары будем называть сопряжением и обозначать символом  $\#$  с указанием в скобках, по какому разбиению берется сопряжение

$$B_p = B_q \# (\theta_i^I).$$

Для операции сопряжения характерно следующее:

а) если  $B_\rho = B_\rho^* (\theta_i^I)$ , то  $B_\rho = B_\rho^* (\theta_i^I)$ ; (I6)

б) если  $B_\rho \in \theta_i^I$ , то  $B_\rho^* (\theta_i^I) = B_\rho$ ; (I7)

в) при многократном выполнении операции сопряжения порядок выбора разбиений, по которым она выполняется, не существен, т.е.

$$[B_\rho^* (\theta_i^I)]^* (\theta_j^I) = [B_\rho^* (\theta_j^I)]^* (\theta_i^I) = B_\rho^* (\theta_i^I, \theta_j^I). \quad (I8)$$

**ТЕОРЕМА I.** Для того чтобы множество  $\theta^I = \{\theta_1^I, \dots, \theta_\rho^I\}$  разбиений множества  $S$  на пары состояний было множеством разбиений I-го уровня булевой алгебры, необходимо и достаточно, чтобы каждая пара  $B_\rho \in \theta_j^I$  имела в том же разбиении по одной сопряженной паре  $B_\rho \in \theta_j^I$  по отношению к каждому  $\theta_i^I \in \theta^I$ .

**ДОКАЗАТЕЛЬСТВО.** Необходимость этого условия легко доказывается от обратного. Предположим, что пара  $(s_i, s_j) \in \theta_j^I$  не имеет сопряженной по  $\theta_i^I$ . Это значит, что если  $\{s_i, s_k; s_j, s_l\} \in \theta_i^I$ , то  $s_k$  и  $s_l$  в  $\theta_j^I$  находятся в разных парах, например,  $(s_k, s_n), (s_l, s_m)$ . Тогда  $\theta_j^I + \theta_i^I$  будет содержать блок  $B = \{s_k, s_l, s_m, s_n, s_i, s_j\}$ , имеющий  $|B| > 4$ . Следовательно,  $B$  не может быть разбиением второго уровня, что противоречит (II).

Чтобы доказать достаточность, построим структуру на множестве разбиений, равных всевозможным суммам по  $\nu$  из заданного множества  $\theta^I$  разбиений на пары:  $\sum_{k=1}^{\nu} \theta_{i_k}^I$  ( $\nu = 1, 2, \dots, \rho$ ;  $i_k \in \{1, 2, \dots, \rho/2\}$ ). Из условий теоремы непосредственно вытекает, что

1)  $\sum_{j=1}^{\rho} \theta_j^I = 1$ ;

2)  $\prod_{j=1}^{\rho} \theta_j^I = \theta$ ;

3) если  $\theta_j^\nu = \theta_{i_1}^I + \theta_{i_2}^I + \dots + \theta_{i_\nu}^I$ ,

то  $\theta_k^{\rho-\nu} = \theta_{i_{\nu+1}}^I + \dots + \theta_{i_\rho}^I$  ( $\{i_1, \dots, i_\rho\} = \{1, 2, \dots, \rho\}$ ) является его дополнением.

Поскольку эти 3 свойства определяют булеву алгебру [10], то теорема доказана.

**СЛЕДСТВИЕ.** Каждая пара  $B_\rho \in \hat{\theta}_i$ , принадлежащая коллекции  $\hat{\theta}_k^i$ , порожденной подмножеством  $\hat{\theta}^I$ , имеет в той же коллекции по одной сопряженной с ней паре по отношению к каждому  $\theta_{i_k}^I \in \hat{\theta}^I$ .

Это следствие непосредственно вытекает из свойства 9 (см. стр. 39) и позволяет для каждой выбранной пары  $B_\rho \in \theta_i^I$  однозначно определить все пары содержащей её коллекции, если известны все остальные (кроме  $\theta_i^I$ ) разбиения порождающего подмножества.

**ТЕОРЕМА 2.** Если  $\theta_j^I \in \hat{\theta}^I$ , то в нем не содержится ни одной такой пары, оба состояния которой входили бы в пары одной и той же коллекции, порождаемой  $\hat{\theta}^I$ .

**ДОКАЗАТЕЛЬСТВО.** Предположим обратное: пусть существует  $B_\rho \in \theta_j^I$ , которая покрывается одним из блоков  $B_k^\nu \in \theta_i^\nu$ , где  $\theta_i^\nu = \sum_{k=1}^{\nu} \theta_{i_k}^I$ . Тогда в произведении  $\theta_i^\nu \cdot \theta_j^I$  тоже должна существовать эта пара, что невозможно, так как  $(\theta_{i_1}^I + \dots + \theta_{i_\nu}^I) \cdot \theta_j^I = 0$ , поскольку каждое произведение  $\theta_{i_k}^I \cdot \theta_j^I = 0$ .

Введем определенный порядок нумерации пар в разбиении I-го уровня. В каждом  $\nu$ -м разбиении  $\theta_j^I$  будем нумеровать пары в порядке их принадлежности к коллекциям, порождаемым подмножеством  $\hat{\theta}$ , состоящим из первых  $\nu$  разбиений,  $\hat{\theta} = \{\theta_1^I, \dots, \theta_\nu^I\}$ . Первой парой  $B_{k_1}$  любой  $k$ -й коллекции  $C_k \in \theta_j^\nu$  будет пара с номером  $k_1 = (k-1)2^{\nu-1} + 1$ . Остальные пары  $B_{k_1+1}, \dots, B_{k_1+l}, B_{k_1+l+1}, \dots, B_{k_1+2^{\nu-1}}$  будем нумеровать так, чтобы между ними соблюдалось следующее соотношение:

$$B_{k_1+l} = B_{k_1+l-2^{l-1}}^* (\theta_i) \quad (l = 2, 3, \dots, 2^{\nu-1}; i = \lceil \log l \rceil). \quad (I9)$$

Применяя свойство (I7) операции сопряжения, нетрудно убедиться в том, что коллекция  $C_k \in \theta_j^\nu$ , построенная согласно (I9), удовлетворяет следствию из теоремы I. Если при этом первые па-



ры коллекций во всех разбиениях выбраны на основе условий теоремы 2, то такие разбиения составляют множество разбиений I-го уровня булевой структуры.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32		
1	2	3	4				5									6																	
2	1	2	3	4				5									6																
3	2	1	2			4			5									6															
4	2	1	2			4			5										6														
5	3			2	3					5										6													
6	3		1	2	3					5											6												
7		3	2		2						5											6											
8			3	2	1	2					5												6										
9	4							2	3	4													6										
10	4							1	2	3	4													6									
11		4						2	1	2		4													6								
12			4					2	1	2			4													6							
13				4				3			2	3														6							
14					4				3		1	2	3														6						
15						4				3	2	1	2															6					
16							4				3	2	1	2																		6	
17	5										2	3	4				5																
18	5										1	2	3	4			5																
19		5									2	1	2		4			5															
20			5									2	1	2		4			5														
21				5								3		2	3				5														
22					5							3		1	2	3				5													
23						5							3	2	1	2					5												
24							5							3	2	1	2					5											
25								5							4							2	3	4									
26									5							4							1	2	3	4							
27										5							4						2	1	2								
28																		4						2	1	2							
29																			4					3		2	3						
30																				4					3		1	2	3				
31																									3	2	1	2					
32																										3	2	1	2				

Рис. 5

Выбранный таким образом порядок нумерации пар устанавливает строгую зависимость по операции сопряжения между парами любого ( $\nu$ -го) разбиения I-го уровня. Эта зависимость наглядно изображается в виде матрицы  $R = \|r_{\alpha\beta}\|$  (рис. 5), в которой строкам и столбцам соответствуют номера пар в любом  $\nu$ -м разбиении, а элементами  $r_{\alpha\beta}$  являются номера разбиений  $\theta_j^I$ , по

которым сопряжены между собой  $\alpha$ -я и  $\beta$ -я пары в  $\theta_j^I$ , причем нижняя треугольная матрица ( $\alpha > \beta$ ) соответствует случаям  $\nu > i$ , верхняя треугольная матрица ( $\alpha < \beta$ ) - случаям  $\nu < i$ , а диагональные клетки ( $\alpha = \beta, \nu = i$ ) в соответствии с (I7) заполнены номером того разбиения, в котором определяются сопряженные пары -  $\theta_j^I$ .

Представленная в матрице зависимость между парами в разбиении  $\theta_j^I$  выражается следующими соотношениями:

$$B_j = \begin{cases} B_{j+2^{i-1}}^{\#}(\theta_i), & \text{если } \nu > i, j \pmod{2^i} \leq 2^{i-1}; & (20) \\ B_{j-2^{i-1}}^{\#}(\theta_i), & \text{если } \nu > i, j \pmod{2^i} > 2^{i-1}; & (21) \\ B_{j+2^{i-2}}^{\#}(\theta_i), & \text{если } \nu < i, j \pmod{2^{i-1}} \leq 2^{i-2}; & (22) \\ B_{j-2^{i-2}}^{\#}(\theta_i), & \text{если } \nu < i, j \pmod{2^{i-1}} > 2^{i-2}. & (23) \end{cases}$$

Введение строгого порядка нумерации пар позволяет определять блоки разбиений более высоких уровней без применения операции сложения разбиений. Это выполняется путем объединения пар с определенными номерами в каком-либо разбиении I-го уровня. Пусть, например, требуется найти разбиение  $\nu$ -го уровня  $\theta_j^{\nu} = \sum_{i=1}^{\nu} \theta_{i_j}$ , причем  $(i_1 < i_2 < \dots < i_{\nu})$ . Будем искать блоки  $\theta_{i_j}^{\nu}$  как объединения пар, принадлежащих коллекциям  $C_k \in \theta_{i_j}$ , порождаемым  $\hat{\theta} = \{\theta_{i_1}, \dots, \theta_{i_{\nu}}\}$  на разбиении  $\theta_{i_{\nu}}^I$ . Условия теоремы 2 требуют, чтобы в разных коллекциях не существовало пар, сопряженных по разбиениям порождающего множества  $\hat{\theta}$ . Поэтому первые пары коллекций  $\{C_1, \dots, C_k, \dots, C_{2^{\nu}-1}\}$  будем определять как множество пар, сопряженных по разбиениям, не принадлежащим  $\hat{\theta}$ , т.е. по  $\{\theta_{i_{\nu+1}}^I, \dots, \theta_{i_p}^I\}$ , используя рекуррентно формулы (21, 23):

$$B_{k_1} = \begin{cases} B_{k_1-2^{i-1}}^{\#}(\theta_{i_{\nu+k}}), & \text{если } i_{\nu} > i_{\nu+k}, i_{\nu+k} = \lfloor \log k_1 \rfloor; \\ B_{k_1+2^{i-1}}^{\#}(\theta_{i_{\nu+k}}), & \text{если } i_{\nu} < i_{\nu+k}, i_{\nu+k} = \lfloor \log k_1 \rfloor. \end{cases} \quad (24)$$

Все остальные пары  $k$ -й коллекции будут определяться таким же рекуррентным способом, но поскольку мы определяем коллекции в разбиении с наибольшим номером в  $\hat{v}$ , то используется только случай  $(i_j > i_j)$ .

$$B_{k_\ell} = B_{k_\ell - 2^{i-1}}^{\#}(\theta_{i_j}) \quad (25)$$

( $\ell = 2, 3, \dots, 2^{i-1}$ ;  $i_j$  - ближайшее к  $\lceil \log k_\ell \rceil$  целое число, принадлежащее  $\{i_1, \dots, i_v\}$ ).

Условия теоремы 1 и 2 и соотношения (20-23) позволяют получить оптимальный код внутренних состояний и матрицу реализации без построения булевой структуры в целом, ограничиваясь определением разбиений 1-го уровня.

Из общей постановки задачи синтеза имеет смысл выделить частный случай, характерный для слабо определенных автоматов, - когда попарные пересечения  $K$ -множеств пусты. Поскольку в этом случае интервалы, покрывающие  $K$ -множества, тоже не пересекаются, задача оптимального кодирования становится тривиальной и по существу сводится к присвоению каждому  $K_j \in K$  группы из  $2^{\lceil \log |K_j| \rceil}$  кодирующих слов, образующих интервал в  $p$ -мерном пространстве внутренних переменных. При этом для того, чтобы каждое  $K$ -множество покрывалось одним интервалом (матрица  $M$  была минимальна), необходимо выполнение следующих условий:

- а) самое большое  $K$ -множество должно иметь мощность не более чем  $2^{p-1}$ ;
- б) количество различных кодирующих слов должно быть достаточным для образования покрывающих интервалов для всех  $K$ -множеств, т.е.

$$\sum_{j=1}^{\infty} 2^{\lceil \log |K_j| \rceil} \leq 2^p \quad (26)$$

Если эти условия не удовлетворяются, то необходимо разбить некоторые из  $K$ -множеств на подмножества, мощность которых равна целым степеням двойки.

Задача определения покрытий  $K$ -множеств в случае, когда они не пересекаются, исчезает. Получение матрицы реализации сводится к расстановке единиц и нулей в её строках, соответствующих интервалам, покрывающим  $K$ -множества.

На основе приведенных в этом разделе соотношений, характеризующих свойства кода внутренних состояний с точки зрения оптимальности реализации, разработан комплекс алгоритмов и программ, включающих в себя весь процесс синтеза автомата матричным способом.

### § 3. Алгоритм оптимального синтеза

Алгоритм разработан для двух способов исходного задания автомата:

- 1) таблицей переходов и выходов с таблицами кодов входных  $C_{\Sigma}$  и выходных  $C_r$  состояний;
- 2) микропрограммным описанием на языке граф-схем алгоритмов с таблицей кодов микрокоманд.

Результатом выполнения алгоритма является оптимизированная матрица реализации автомата. Если необходимо, можно вывести из ЭВМ оптимальный код внутренних состояний.

Алгоритм состоит из 10 процедур, реализованных отдельными программами или наборами программ.

Процедура 1. По заданной ГСА построение обратной таблицы переходов, выделение из нее  $K$ -множеств и проверка их на пересечения.

Отмечаются все вершины ГСА, следующие за операторными. Эти вершины считаются "состояниями" автомата. Строится таблица, в которой в каждой строке записывается: 1) номер состояния перехода ( $s_k$ ); 2) номер исходного состояния ( $s_\ell$ ); 3) значения условий по каждому пути через условные вершины от  $s_\ell$  к  $s_k$ ; 4) состояние выхода. Составляется список  $K$ -множеств: в каждое  $K$ -множество вносятся исходные состояния, стоящие в строках таблицы со всеми остальными одинаковыми значениями. Каждое сформированное  $K$ -множество сравнивается со всеми уже полученными. Отмечается существование непустых пересечений.

Процедура 2. Составление списка взвешенных пар состояний по таблице переходов.

Для каждой пары строк таблицы переходов, соответствующих исходным состояниям ( $s_\ell, s_k$ ), определяется число столбцов, в которых в этих строках стоят одинаковые состояния переходов и выходов. Это число называется весом пары состояний  $\alpha(s_\ell, s_k)$ .

Составляется список взвешенных пар, в котором пары располагаются в порядке убывания их весов:

$$P = \{p_1, \dots, p_i\} \quad (\alpha(p_1) \geq \alpha(p_2) \geq \dots).$$

Процедура 3. Составление списка взвешенных пар по  $K$ -множествам.

Каждое  $K$ -множество  $K_i$  ( $i = 1, \dots, \infty$ ) мощности  $|K_i|$  порождает  $C_{|K_i|}^2$  пар состояний. Вес каждой пары определяется числом  $K$ -множеств, включающих эту пару. В списке пар состояний  $P$  пары расположены в порядке убывания их весов.

Процедура 4. Построение множества разбиений  $I$ -го уровня булевой структуры.

Вычисляется число внутренних переменных, равное числу разбиений первого уровня  $\rho = \lceil \log |S| \rceil$ . Если  $|S| < 2^\rho$ , то множество  $S$  дополняется фиктивными состояниями  $\{s_{|S|+1}, \dots, s_{2^\rho}\}$ .

Каждое разбиение  $\theta_j^I$  составляется из  $2^{\rho-\nu}$  коллекций  $\{C_1, \dots, C_k, \dots, C_{2^{\rho-\nu}}\}$ , порождаемых подмножеством уже построенных разбиений  $\hat{\theta} = \{\hat{\theta}_1, \dots, \hat{\theta}_\nu\}$ . Первая пара  $k$ -й коллекции  $C_k = \{B_{k_1}, \dots, B_{k_\nu}\}$  выбирается из списка взвешенных пар по следующим признакам: она должна обладать наибольшим весом, и, согласно теореме 2, её состояния не должны принадлежать парам одной и той же коллекции разбиения  $\theta_{j-1}^I$ . Если в списках  $P$  нет подходящей пары, то она составляется произвольно. Каждая следующая пара  $B_{k_\nu} \in C_k$  определяется по рекуррентной формуле (20), причем операция сопряжения выполняется путем сравнения состояний в парах.

Поскольку в разбиении  $\theta_j^I$  ( $\nu=1$ ) каждая коллекция состоит только из одной пары, то оно определяется как множество, состоящее из  $2^{\rho-1}$  не пересекающихся между собой первых пар из списка  $P$ , полученного с помощью процедуры 2. Вошедшие в  $\theta_j^I$  пары из списка вычеркиваются. Во втором разбиении  $\theta_2^I$  коллекции состоят из двух пар  $\{B_{k_1}, B_{k_2}\}$ ;  $B_{k_1}$  выбирается из оставшегося списка как первая не пересекающаяся с уже вошедшими в  $\theta_2^I$ ;  $B_{k_2} = B_{k_1}^{\#}(\theta_1)$  согласно (20). В третьем разбиении коллекции состоят из четырех пар, причем выбор каждой первой требует проверки, чтобы её состояния не входили в одну и ту же коллекцию  $\theta_2^I$ . Остальные пары:

$$B_{k_2} = B_{k_1}^{\#}(\theta_1),$$

$$B_{k_3} = B_{k_1}^{\#}(\theta_2),$$

$$B_{k_4} = B_{k_2}^{\#}(\theta_2).$$

Процедура 5. Выделение  $K$ -множества из таблицы переходов.

В каждом столбце таблицы переходов выделяются группы одинаковых значений состояний переходов и выходов. Исходные состояния автомата, соответствующие этим переходам, объединяются в  $K$ -множество.

Процедура 6. Определение блоков разбиений  $\nu$ -го уровня.

Каждому  $i$ -му разбиению  $\theta_i^I$  ставится в соответствие одно сочетание  $\nu$  чисел  $\{i_1, i_2, \dots, i_\nu\}$  из  $\{1, 2, \dots, \rho\}$  такое, что  $\theta_i^\nu = \sum_{k=1}^{\nu} \theta_{i_k}$ . Блоки  $B_k \in \theta_i^\nu$  определяются как объединения пар коллекций, порождаемых  $\hat{\theta} = \{\theta_{i_1}^I, \dots, \theta_{i_\nu}^I\}$  на разбиении с наибольшим индексом  $\theta_{i_\nu}$  ( $i_1 < i_2, \dots < i_\nu$ ). Первые пары этих коллекций  $\{B_{k_1} \in C_k\}$  ( $k = 1, \dots, 2^{\rho-\nu}$ ) определяются согласно (24), а все остальные — по (25).

Процедура 7. Определение покрытия  $K$ -множества.

$K$ -множество  $K_j$  сравнивается со всеми блоками разбиений булевой структуры, имеющих не более  $2^{\lceil \log |K_j| \rceil - 1}$  состояний. Перебор блоков начинается с наибольших, т.е. с блоков разбиений  $(\lceil \log |K_j| \rceil - 1)$ -го уровня. Каждый определяемый с помощью процедуры 6 блок  $B_k$  сравнивается с  $K_j$ . Если  $B_k \in K_j$  или  $B_k$  отличается от  $K_j$  только фиктивными состояниями, то он является покрывающим интервалом и определяет одну строку матрицы реализации автомата, которая составляется с помощью процедуры 10. Если  $K_j$  покрыто  $B_k$  не полностью, т.е.  $K_j' = K_j \setminus B_k \neq \emptyset$ , то к  $K_j'$  вновь применяется процедура 7 и так до тех пор, пока  $K_j$  не окажется полностью покрытым.

Процедура 8. Определение кодирующей таблицы.

Двублочные разбиения, определяющие код внутренних состояний, вычисляются как суммы разбиений  $I$ -го уровня по (15).

Процедура 9. Построение кодирующей таблицы внутренних состояний в случае непересекающихся  $K$ -множеств.

Рассматриваются все  $K$ -множества, начиная с больших по мощности. Если  $|K_j| = 2^i$  ( $i$  - целое число), то оно оставляется без изменения; если  $|K_j| < 2^{\lceil \log |K_j| \rceil}$ , то оно дополняется фиктивными состояниями; если фиктивных состояний не хватает, то оно разбивается на подмножества, мощности которых равны 2. Каждому полученному таким способом  $K$ -множеству  $K_j$  приписывается  $2^{\lceil \log |K_j| \rceil}$   $\rho$ -разрядных двоичных чисел, располагаемых в естественном порядке.

Процедура 10. Определение строки матрицы реализации.

Для каждого интервала  $B_k$  покрытия  $K$ -множества в прямой (если автомат задан таблицей переходов) или обратной (если автомат задан в виде ГСА) таблице переходов находятся соответствующие ему входные и выходные состояния. Строка матрицы  $M$  составляется из двух частей. В первой части записываются код входного состояния и код интервала  $B_k$ . Каждой переменной в этой части отводится по два разряда: (1,0) - если переменная в кодирующем слове равна 1; (0,1) - если переменная равна 0; (0,0) - если переменная отсутствует. Во второй части строки записываются обычные коды состояний перехода и выхода.

Блок-схема алгоритма изображена на рис.6. В прямоугольниках показаны номера процедур и результаты их выполнения. Связи между прямоугольниками соответствуют направлениям передачи информации. Условия выбора путей в схеме записаны в ромбах. Входные и выходные данные в овалах.

Алгоритм построен с учетом экономии оперативной памяти машины. В памяти хранятся только исходные данные, список пар и множество разбиений первого уровня. Ни  $K$ -множества, ни блоки разбиений структуры (для  $\nu > 1$ ) в память не заносятся. Каждое  $K$ -множество  $K_j$  формируется исходя из задания и обрабатывается до получения соответствующих ему строк матрицы  $M$ . При этом блоки разбиений структуры, среди которых ищутся элементы покрытия, каждый раз составляются путем объединения пар состояний. Такой подход позволяет обрабатывать автоматы достаточно большой размерности.

Емкостная функция сложности алгоритма (необходимое количество ячеек памяти) составляется из размеров следующих массивов:

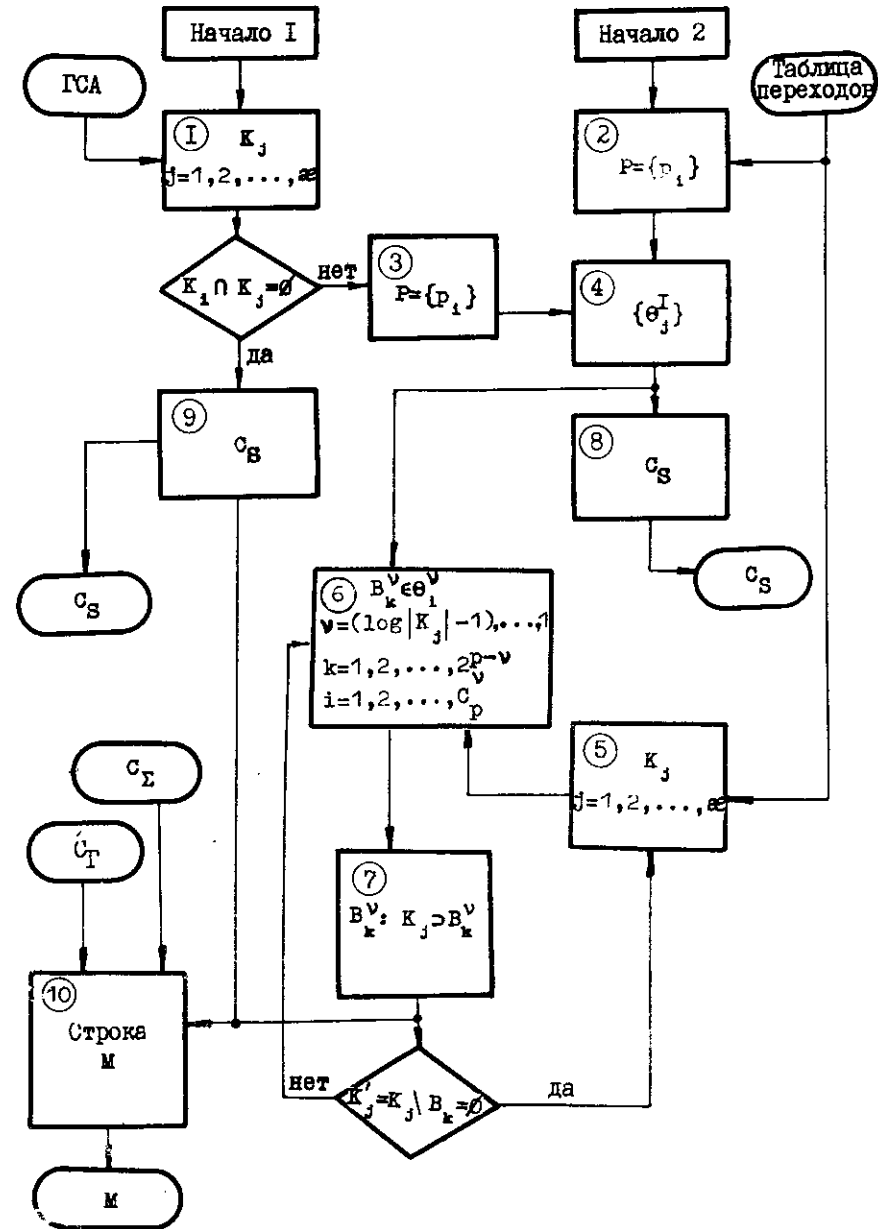


Рис. 6. Блок-схема алгоритма синтеза

таблицы кодов входных состояний  $2^n \cdot n$  ;  
 таблицы кодов внутренних состояний  $2^p \cdot p$  ;  
 таблицы кодов выходных состояний  $2^m \cdot m$  ;  
 таблицы переходов и выходов или ГСА  $2^{p+n+1}$  или  $4 \times$  (число вершин в ГСА);

множества разбиений I-го уровня структуры  $2^p \cdot p$  ;  
 списка взвешенных пар  $2^{2p-1}$  ;  
 нескольких рабочих массивов в  $2^{p+2}$ .

Здесь  $n$ ,  $p$  и  $m$  - количества входных, внутренних и выходных переменных, соответственно.

Суммируя объемы массивов и отбрасывая малые слагаемые, получим

$$V \approx 2^{2p-1} + 2^{p+n+1}.$$

Так, например, если в распоряжении программиста имеется 32 000 ячеек памяти, то он может синтезировать автомат с числом входных, внутренних и выходных состояний соответственно 32-256-256 или I28-64-256 и т.д.

Временную функцию сложности оценим необходимым количеством операций типа сравнения числа с числом. Распределение количества операций по процедурам приведено в таблице.

Т а б л и ц а

№ процедуры	Количество операций в процедуре
1	$2^{2p+n+m}$
2	$2^{2(p+n)}$
3	$2^{2(p+n)}$
4	$2^{2p}$
5	$2^{2(p+n)+m}$
6	$3^p - 2^p$
7	$(3^p - 2^p) \cdot 2^{p+n}$
8	$2^{p \cdot p}$
9	$2^{p \cdot p}$
10	$2^{p-1}$

Суммируя операции по всем процедурам и опуская малые слагаемые, получим

$$T \approx 2^{p+n} (2^{p+n+m} + 3^p).$$

Таким образом, предложены методы логического проектирования автоматов с учетом требований, предъявляемых БИС-технологией. Разработанная на основе этих методов система алгоритмов и программ предназначена для проектирования устройств управления микрокалькуляторов, микропроцессоров и мини-ЭВМ, реализуемых на основе БИС. Отдельные алгоритмы и программы могут применяться также для реализации любых сложных логических схем, таких как декодеры микрокоманд, преобразователи кодов, схемы принятия решений и т.д. Методика и алгоритмы кодирования состояний применимы для кодирования микрокоманд при проектировании микропроцессоров.

Эффективность работы системы в целом и отдельных алгоритмов проверена на реальных примерах. Система реализована на ЭВМ БЭСМ-6 (язык ФОРТРАН), содержит 8000 команд.

#### Л и т е р а т у р а

1. ЕВРЕЙНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, 1966.

2. БАНДМАН О.Л. Синтез автоматов в криотронной вычислительной среде. - "Автоматика и вычислительная техника", 1972, №5, с. 20-30.

3. МЕЛИХОВ А.Н., ТОПОЛЬСКИЙ И.Г. Об одном методе синтеза автоматов в вычислительной среде. - В кн.: Цифровые модели и интегрирующие структуры. Таганрог, 1970, с. 265-275.

4. РУВИНСКИЙ Б.И. Реализация микропрограммного автомата на БИС. - "Электронная техника, сер. 3, Микроэлектроника". Вып. 5(55), 1974.

5. ГОРМАН, КАУФМАН. Путь к созданию новых систем на основе мини-ЭВМ. - "Электроника", 1973, т.46, № 12, с.44-55.

6. GORMAN K. The programmable logic array: a new approach to microprogramming. - "END", November 20, 1973, p.68-75

7. СИДЕРИС. Ведущие полупроводниковые фирмы США свертывают производство заказных ИС. - "Электроника", 1974, т. 47, №1, с. 88-91.

8. BANDMAN O.I. State-Assignment of Sequential Machines for LSI Implementation. - In: Discrete Systems. Proc. of International Symposium IFAC, October, 1974. Riga, 1974.

9. БАРАНОВ С.И. Синтез микропрограммных автоматов. "Энергия", 1974.

10. HARTMANIS J., STEARNS R. Algebraic Structure Theory of Sequential Machines. New York, 1966.

11. СКОРНЯКОВ Л.А. Элементы теории структур. М., "Наука", 1970.

Поступила в ред.-изд.отд.

13 января 1975 года