

С.И.БАРАНОВ , В.А.СКЛЯРОВ

**Цифровые
устройства
на программируемых
БИС
с матричной
структурой**



Москва · Радио и связь · 1986

ББК 32.97

Б24

УДК 621.3.049.771.14-529:681.3

Баранов С. И., Скляр В. А.

Б24 Цифровые устройства на программируемых БИС с матричной структурой. — М.: Радио и связь, 1986. — 272 с.: ил.

Изложены принципы структурной организации и физического исполнения программируемых больших интегральных схем с матричной структурой—программируемых логических матриц, постоянных запоминающих устройств, программируемых мультиплексоров и др. Рассмотрены методы проектирования логических схем современных управляющих систем цифровой автоматики и вычислительной техники. Для инженерно-технических работников, специализирующихся в области проектирования систем цифровой автоматики и вычислительной техники.

Б 2403000000-109 120-86
046(01)-86

ББК 32.97

Рецензенты: член-кор. АН БССР *А. Д. Закревский* и д-р. техн. наук *А. А. Новиков*

Редакция литературы по электронной технике

© Издательство «Радио и связь», 1986

Предисловие

Успехи в области интегральной технологии, достигнутые за последние 8—10 лет, привели к созданию больших интегральных схем (БИС), содержащих десятки тысяч элементов на одном кристалле. Однако появление БИС породило очень серьезную проблему — «что положить на подложку» или, другими словами, каким образом реализовать устройство на схемах с таким колоссальным количеством элементов.

Первым и довольно естественным решением этой проблемы явилось изготовление так называемых заказных схем, разрабатываемых каждый раз специально для использования в конкретной аппаратуре. В то же время проектирование заказных БИС — весьма длительный и трудоемкий процесс, использующий сложные человеко-машинные системы автоматизированного проектирования. Поэтому разработка и изготовление заказных БИС могут быть экономически оправданны только при массовом производстве аппаратуры, в которой эти схемы применяются.

Хорошей альтернативой заказным БИС явились микропроцессорные наборы — совокупность больших интегральных схем, реализующих сложные функции цифровой аппаратуры. Из этих «кирпичей» достаточно просто строятся микрокомпьютеры (микро-ЭВМ), получившие исключительное развитие и нашедшие широкое применение в разнообразных системах управления.

Микропроцессор является универсальным устройством, способным реализовать любую логическую функцию. Однако программная реализация логики управления осуществляется сравнительно медленно, микропроцессор зачастую не способен обеспечить необходимое быстрое действие. В связи с этим в настоящее время широкое распространение получили программируемые БИС с матричной структурой, среди которых особое место занимают программируемые логические матрицы (ПЛМ) — большие интегральные схемы, сочетающие регулярность структуры полупроводникового запоминающего устройства (ЗУ) с универсальностью микропроцессора. ПЛМ обладает существенными преимуществами перед микропроцессором при реализации сложных алгоритмов управления, когда необходимо обеспечить

высокое быстродействие, но не требуется запоминания большой промежуточной информации.

В предлагаемой к изданию книге представлены методы проектирования цифровых устройств на основе широкого набора программируемых логических БИС с матричной структурой — программируемых логических матриц, постоянных запоминающих устройств, программируемых матриц вентиляей и программируемых мультиплексоров. Излагаются принципы структурной организации БИС с матричной структурой и методы проектирования на их основе логических схем цифровых управляющих систем. Упор сделан на методы, апробированные в практике инженерного проектирования и позволяющие строить управляющие устройства реальной сложности. Изложение сопровождается приме-рами.

ЭЛЕМЕНТЫ ТЕОРИИ БУЛЕВЫХ ФУНКЦИЙ

1.1. Булевы функции и их основные свойства

Булевой функцией (БФ) y называется функция, принимающая, как и ее аргументы x_1, \dots, x_L , два значения, которые обозначим 0 и 1. Число различных наборов значений аргументов БФ конечно, в силу чего любая БФ y может быть полностью задана таблицей с 2^L строками. В левой части таблицы перечисляются все наборы значений аргументов функции y , а в правой — значения этой функции на соответствующих наборах. В табл. 1.1 приведен пример табличного задания БФ $y(x_1, x_2, x_3)$.

Если наборам значений аргументов БФ y поставить в соответствие точки L -мерного пространства, то множество 2^L наборов определит множество вершин L -мерного единичного куба, которое образует область определения БФ, зависящей от L аргументов. Обозначим \circ (кружком) и \times (крестиком) вершины L -мерного куба, соответствующие наборам значений аргументов, на которых функция принимает значения 0 и 1. В результате получим геометрическое задание БФ. Пример такого задания для функции $y(x_1, x_2, x_3)$ из табл. 1.1 приведен на рис. 1.1.

Значения БФ могут быть заданы не на всех 2^L возможных наборах значений аргументов, и тогда на некоторых из них они не определены, что обозначается прочерком (—) в таблице в столбце значений функции. Такие функции называются неполностью определенными или частичными. Частичная БФ может быть доопределена путем подстановки на место прочерков различных комбинаций из нулей и единиц. Например, если выполнить различные доопределения БФ, заданной табл. 1.2, то можно получить четыре различные полностью определенные функции (табл. 1.3). Таким образом, если БФ не определена на k наборах значений аргументов, то путем ее всевозможных доопределений можно получить 2^k различных полностью определенных функций.

Полностью определенная БФ $y(x_1, \dots, x_{l-1}, x_l, x_{l+1}, \dots, x_L)$ существенно зависит от аргумента x_l , если выполняется соотношение $y(x_1, \dots, x_{l-1}, 0, x_{l+1}, \dots, x_L) \neq y(x_1, \dots, x_{l-1}, 1, x_{l+1}, \dots, x_L)$. В противном случае БФ не зависит от x_l и x_l является ее фиктивным аргументом. Например, для функции y_3 , заданной табл. 1.3, аргумент x_4 фиктивный.

Число различных БФ, зависящих от L аргументов, конечно и равно 2^{2^L} . Среди многообразия существующих БФ рассмотрим те, которые играют наиболее важную роль в приложениях и встречаются в последующих главах книги. При $L=0$ имеются лишь две БФ, совпадающие с константами 0 и 1. При $L=1$ нас будут интересовать функции y_5 и y_6 , определяемые табл. 1.4. Функция $y_5 = x_1$ называется функцией повторения, а $y_6 = \bar{x}_1$ (читается «не x_1 ») — функцией отрицания или инверсии. Среди 16 различных БФ двух аргументов выберем четыре функции y_7 — y_{10} , заданные табл. 1.5. Функция y_7 называется функцией

Таблица 1.1

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	1
1	0	0	1
0	1	1	0
1	1	0	0
1	0	1	0
1	1	1	1

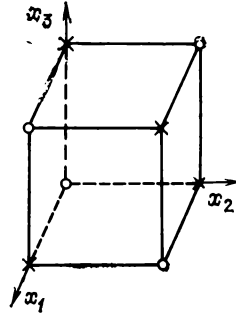


Рис. 1.1

Таблица 1.2

x_1	x_2	y
0	0	1
0	1	0
1	0	—
1	1	—

Таблица 1.3

x_1	x_2	y_1	y_2	y_3	y_4
0	0	1	1	1	1
0	1	0	0	0	0
1	0	0	0	1	1
1	1	0	1	0	1

дизъюнкции аргументов x_1 и x_2 (обозначается $y_7 = x_1 \vee x_2$), y_8 — функцией конъюнкции указанных аргументов (обозначается $y_8 = x_1 x_2$). Функция y_9 является функцией эквивалентности или равнозначности (записывается $y_9 = x_1 \sim x_2$), а y_{10} — функцией сложения по модулю 2 (записывается $y_{10} = x_1 \oplus x_2$).

Таблица 1.4

x_1	y_5	y_6
0	0	1
1	1	0

Таблица 1.5

x_1	x_2	y_7	y_8	y_9	y_{10}
0	0	0	0	1	0
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	1	1	0

Рассмотренные функции позволяют строить новые БФ путем перенумерации их аргументов и подстановки в функцию новых функций вместо их аргументов. Функцию y , полученную из функций y_1, \dots, y_n путем одно- или многократного применения этих правил, называют суперпозицией функций y_1, \dots, y_n .

Перечислим теперь основные свойства введенных БФ. Функции y_6, y_7, y_8 удовлетворяют следующим аксиомам:

$$\left. \begin{array}{l} x_1 \vee x_1 = x_1; \\ x_1 x_1 = x_1 \end{array} \right\} \text{— идемпотентный закон;}$$

$$\left. \begin{array}{l} x_1 \vee x_2 = x_2 \vee x_1; \\ x_1 x_2 = x_2 x_1 \end{array} \right\} \text{— коммутативный закон;}$$

$$\left. \begin{array}{l} (x_1 \vee x_2) \vee x_3 = x_1 \vee (x_2 \vee x_3); \\ (x_1 x_2) x_3 = x_1 (x_2 x_3) \end{array} \right\} \text{— ассоциативный закон;}$$

$$\left. \begin{array}{l} x_1 (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3; \\ x_1 \vee (x_2 x_3) = (x_1 \vee x_2) (x_1 \vee x_3) \end{array} \right\} \text{— дистрибутивный закон;}$$

$$\left. \begin{array}{l} x_1 \vee \bar{x}_1 = 1; \\ x_1 \bar{x}_1 = 0 \end{array} \right\} \text{— закон отрицания;}$$

$$\bar{\bar{x}}_1 = x_1 \text{— закон двойного отрицания;}$$

$$\left. \begin{array}{l} x_1 \vee x_2 = \overline{\bar{x}_1 \bar{x}_2}; \\ x_1 x_2 = \overline{\bar{x}_1 \vee \bar{x}_2} \end{array} \right\} \text{— закон двойственности} \\ \text{(правило де Моргана).}$$

Доказательство справедливости любого из этих законов легко выполняется путем сравнения таблиц для функций, записанных в левой и правой частях приведенных равенств.

Для функций y_9 и y_{10} справедливы ассоциативный и коммутативный законы, а также дистрибутивный закон относительно конъюнкции:

$$x_1 \oplus x_2 = x_2 \oplus x_1;$$

$$x_1 \sim x_2 = x_2 \sim x_1;$$

$$x_1 \oplus (x_2 \oplus x_3) = (x_1 \oplus x_2) \oplus x_3;$$

$$x_1 \sim (x_2 \sim x_3) = (x_1 \sim x_2) \sim x_3;$$

$$x_1 (x_2 \oplus x_3) = (x_1 x_2) \oplus (x_1 x_3);$$

$$x_1 (x_2 \sim x_3) = (x_1 x_2) \sim (x_1 x_3).$$

Выполняются также следующие очевидные соотношения:

$$x_1 \oplus x_1 = 0; \quad x_1 \oplus 0 = x_1; \quad x_1 \oplus 1 = \bar{x}_1; \quad x_1 \oplus \bar{x}_1 = 1;$$

$$x_1 \sim x_1 = 1; \quad x_1 \sim 0 = \bar{x}_1; \quad x_1 \sim 1 = x_1; \quad x_1 \sim \bar{x}_1 = 0.$$

Функции дизъюнкции, конъюнкции, эквивалентности и сложения по модулю 2 могут зависеть более чем от двух аргументов:

$$y_{11} = x_1 \vee x_2 \vee \dots \vee x_L = \bigvee_{l=1}^L x_l;$$

$$y_{12} = x_1 x_2 \dots x_L = \bigwedge_{l=1}^L x_l;$$

$$y_{13} = x_1 \sim x_2 \sim \dots \sim x_L;$$

$$y_{14} = x_1 \oplus x_2 \oplus \dots \oplus x_L.$$

Дадим несколько определений. Конъюнкцию назовем элементарной, если она содержит любое конечное множество попарно различных между собой букв, над частью которых (быть может, пустой или,

наоборот, содержащей все буквы) поставлены знаки отрицания. Ранг элементарной конъюнкции — это число входящих в нее аргументов. Дизъюнктивной нормальной формой (ДНФ) БФ называется дизъюнкция конечного множества попарно различных элементарных конъюнкций. ДНФ функции $y(x_1, \dots, x_l)$, все элементарные конъюнкции которой имеют ранг l , называется совершенной ДНФ этой функции. Любая БФ может быть представлена в совершенной ДНФ, для чего необходимо: выбрать в таблице задания функции все наборы значений аргументов, на которых она обращается в 1; выписать конъюнкции, соответствующие этим наборам значений аргументов (x_i вписывается в конъюнкцию без изменения, если его значение в l -м компоненте набора равно 1, и со знаком отрицания, если его значение в l -м компоненте набора равно 0); все полученные конъюнкции объединить знаком дизъюнкции. Например, совершенная ДНФ функции y , заданная табл. 1.1, запишется так:

$$y = \bar{x}_1 \bar{x}_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_2 x_3.$$

В заключение параграфа определим еще несколько понятий. Системе БФ $\{y_1, \dots, y_n\}$ назовем полной, или базисом, если любая БФ может быть представлена суперпозицией функций y_1, \dots, y_n . Минимальным базисом назовем такой, для которого удаление хотя бы одной из функций y_n , входящих в этот базис, превращает указанную систему БФ в неполную. Известно, например, что системы функций: отрицание и дизъюнкция, а также отрицание и конъюнкция — образуют минимальные базисы; функции отрицания, дизъюнкции и конъюнкции образуют избыточный (неминимальный) базис.

1.2. Минимизация булевых функций

Рассмотрим совершенную ДНФ функции y_4 , заданной табл. 1.3, $y_4 = \bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee x_1 x_2$. Преобразуем, ее используя ранее приведенные законы и тождества:

$$y_4 = \bar{x}_1 \bar{x}_2 \vee x_1 \bar{x}_2 \vee x_1 x_2 = \bar{x}_2 (\bar{x}_1 \vee x_1) \vee x_1 (x_2 \vee \bar{x}_2) = \bar{x}_2 \vee x_1.$$

Из примера видно, что последняя запись БФ y_4 значительно проще исходной.

Определим ряд понятий, необходимых для дальнейшего изложения.

Под длиной ДНФ будем понимать число образующих ее элементарных конъюнкций. ДНФ, имеющая наименьшую длину по сравнению со всеми другими ДНФ данной функции, называется кратчайшей ДНФ (КДНФ). ДНФ, содержащая наименьшее число букв по сравнению со всеми другими ДНФ данной функции, называется минимальной ДНФ (МДНФ).

Для упрощения булевых выражений могут быть использованы следующие тождества, доказательство которых очевидно:

$$\left. \begin{aligned} x_1 x_2 \vee x_1 \bar{x}_2 &= x_1; \\ (x_1 \vee x_2) (x_1 \vee \bar{x}_2) &= x_1 \end{aligned} \right\} \text{— правило склеивания;}$$

$$\left. \begin{aligned} x_1 \vee x_1 x_2 &= x_1; \\ x_1 (x_1 \vee x_2) &= x_1 \end{aligned} \right\} \text{— правило поглощения;}$$

$$x_1 \vee \bar{x}_1 x_2 = x_1 \vee x_2;$$

$$x_1 (\bar{x}_1 \vee x_2) = x_1 x_2;$$

$$x_1 x_3 \vee \bar{x}_1 x_2 \vee x_2 x_3 = x_1 x_3 \vee \bar{x}_1 x_2.$$

Вернемся к геометрическому представлению БФ. Последующее изложение будем иллюстрировать примером геометрической интерпре-

тации области определения функции трех переменных (число переменных, равное трем, выбрано лишь из соображений наглядности). Элементам куба поставим в соответствие элементарные конъюнкции различного ранга. На рис. 1.2 вершинам куба сопоставлены конъюнкции третьего ранга, ребрам — второго ранга, граням — первого ранга. При этом каждый геометрический эквивалент меньшей размерности покрывается соответствующими геометрическими эквивалентами большей размерности. По аналогии говорят, что конъюнкции большего ранга по-

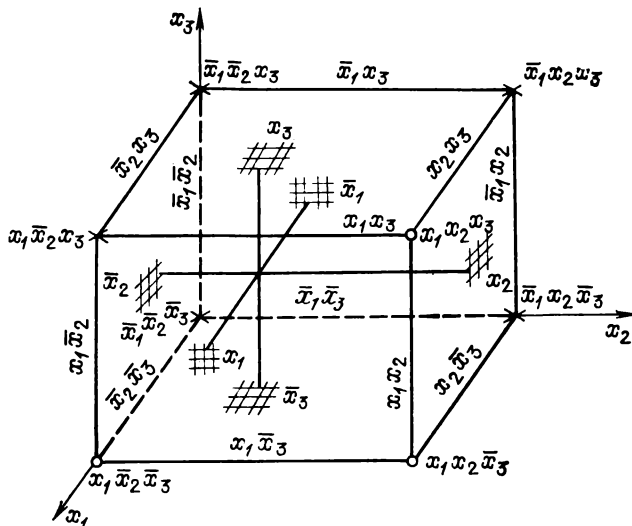


Рис. 1.2

крываются соответствующими конъюнкциями меньшего ранга. Подмножество вершин куба, соответствующих конъюнкции l -го ранга, называют интервалом l -го ранга; подмножество вершин куба, соответствующих наборам значений аргументов, на которых БФ y принимает значение единицы, обозначим через $y(1)$, а значение нуля — через $y(0)$. Тогда задание некоторой ДНФ функции y эквивалентно определению некоторого покрытия множества $y(1)$ интервалами, соответствующими элементарным конъюнкциям, входящим в ДНФ. Например, на рис. 1.2 вершины куба, соответствующие наборам значений аргументов 000, 001, 010, 011, 101, образуют множество $y(1)$.

Трехный вектор 01 , соответствующий конъюнкции \bar{x}_2x_3 , в котором на месте первого компонента записан прочерк ($-$), представляет собой интервал второго ранга. Указанному вектору соответствуют два двоичных вектора 001 и 101, т. е. на место прочерка может быть записан как 0, так и 1. Трехный вектор $0-$ —, соответствующий конъюнкции \bar{x}_1 , представляет собой интервал первого ранга. Вместо него могут быть записаны четыре двоичных вектора: 000, 001, 010, 011. Интервал J называют максимальным, если не существует интервала J' с рангом, меньшим, чем у J , и такого, что выполняется соотношение включения $J \subset J' \subset y(1)$.

ДНФ, соответствующая покрытию множества $y(1)$ всеми максимальными интервалами, называется сокращенной. МДНФ получается из

сокращенной ДНФ путем выбрасывания некоторых интервалов из покрытия множества $y(1)$ максимальными интервалами. Будем рассматривать задачу минимизации БФ как задачу о нахождении МДНФ либо близкой к ней формы. Заметим, что при минимизации изменяется лишь форма представления БФ, в то же время совпадают все значения исходной и полученной функций на наборах значений аргументов, на которых исходная функция определена.

Один из подходов к решению поставленной задачи минимизации БФ состоит в использовании так называемых минимизирующих карт

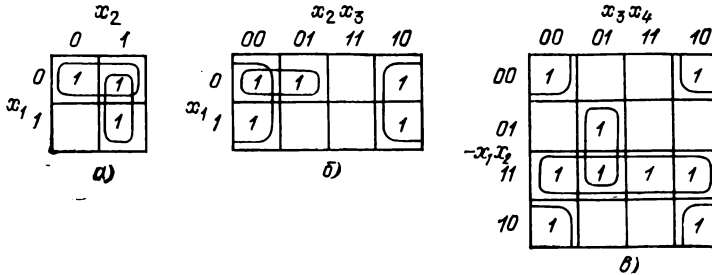


Рис. 1.3

Карно (Karnaugh), которые можно рассматривать как отображение на плоскость вершин L -мерного куба. Использование карт Карно позволяет легко выделять графически элементарные конъюнкции, для которых выполняются правила типа:

$$x_1 \vee x_1 x_2 = x_1; \quad x_1 \vee \bar{x}_1 x_2 = x_1 \vee x_2; \quad x_1 x_2 \vee x_1 \bar{x}_2 = x_1.$$

Эти правила лежат в основе упрощения булевых выражений. Карты Карно, предназначенные для минимизации БФ двух, трех и четырех аргументов, показаны соответственно на рис. 1.3, а, б, в¹. Каждая из четырех клеток на рис. 1.3, а соответствует одному из четырех возможных наборов значений двух аргументов x_1 и x_2 : 00, 01, 10, 11. Аналогичная карта для функции трех аргументов имеет восемь, а для функции четырех аргументов — 16 клеток. Карты построены так, что соседние клетки соответствуют наборам аргументов, различающихся лишь значением одного компонента. Заметим, что клетки на противоположных гранях карты также являются соседними.

Для представления БФ y на карте Карно необходимо записать единицы в клетки, соответствующие наборам значений аргументов, на которых функция y принимает значение 1. Если функция является полностью определенной, то оставшиеся клетки не заполняются. Клетки, в которых записаны единицы, образуют интервалы L -го ранга. Две любые соседние единицы в карте образуют интервал $(L-1)$ -го ранга. Указанному интервалу соответствует элементарная конъюнкция $(L-1)$ аргументов, причем в ней отсутствует та переменная, которая имеет различные значения для двух рассматриваемых единиц в клетках карты.

Например, карта Карно на рис. 1.3, а задает БФ $y = \bar{x}_1 x_2 \vee \bar{x}_1 \bar{x}_2 \vee x_1 x_2$. Двум верхним единицам соответствует элементарная конъюнкция \bar{x}_1 ,

¹ Наличие единиц в клетках карты Карно определяет задание булевых функций, рассматриваемых ниже.

поскольку для левой единицы $x_2=0$, а для правой — $x_2=1$. Двум правым единицам, также образующим интервал первого ранга, соответствует элементарная конъюнкция x_2 . Таким образом, функция y , заданная картой Карно на рис. 1.3,а, после минимизации примет вид $y = \bar{x}_1 \vee \bar{x}_2$. Четыре соседние единицы (см. рис. 1.3,б, в) образуют интервал ($L-2$)-го ранга. В соответствующей элементарной конъюнкции исключаются те переменные, которые для выделенного интервала не сохраняют постоянных значений.

По аналогии строятся интервалы ($L-3$)-го ранга, объединяющие восемь единиц карты Карно, интервалы ($L-4$)-го ранга, объединяющие ее 16 единиц, и т. д. Для того чтобы минимизировать некоторую БФ y , необходимо покрыть все единицы, записанные в клетках карты Карно, минимальным числом интервалов минимального ранга (максимальных интервалов). Упрощенная форма функции представляется в виде дизъюнкции элементарных конъюнкций, соответствующих полученным максимальным интервалам.

На рис. 1.3,б приведен пример минимизации БФ трех переменных

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3.$$

В результате минимизации получим $y = \bar{x}_3 \vee \bar{x}_1 \bar{x}_2$. Карта Карно на рис. 1.3,в задает БФ четырех переменных:

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4 \vee x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 \bar{x}_3 x_4.$$

После минимизации $y = \bar{x}_2 \bar{x}_4 \vee x_1 x_2 \vee x_2 \bar{x}_3 x_4$. Единичные значения БФ для выделенных максимальных интервалов обведены.

Карты Карно позволяют минимизировать не только полностью определенные, но и частичные БФ. В этом случае в некоторых их клетках будут записаны прочерки. В процессе упрощения БФ любую клетку, содержащую прочерк, можно считать либо единичной, либо нулевой, причем прочерк заменяется 1 лишь тогда, когда это позволяет либо получить покрытие множества $y(1)$ меньшим числом интервалов, либо сократить их ранг.

По аналогии можно строить карты Карно на пять и более переменных. Однако с возрастанием числа переменных они становятся сложными и ненаглядными. В силу этого на практике карты Карно применяются для минимизации БФ с числом аргументов не более шести.

Изложим еще один широко распространенный метод упрощения БФ, предложенный Квайном (Quine) и модифицированный Мак-Класки (Mc Cluskey). Метод Квайна — Мак-Класки предполагает, что минимизируемая БФ задана в совершенной ДНФ. Если это не так, то заданная ДНФ расширяется до совершенной путем многократного применения следующего правила: $a = x_i a \vee \bar{x}_i a$, где a — некоторая конъюнкция переменных x_1, \dots, x_L , в которую не входит x_i . Например, БФ $y = x_1 \vee \bar{x}_2 x_3$ расширяется до совершенной ДНФ следующим образом:

$$y = x_1 x_2 \vee x_1 \bar{x}_2 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_2 x_3 = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee \bar{x}_1 \bar{x}_2 x_3.$$

После замены нескольких одинаковых элементарных конъюнкций одной окончательно получим

$$y = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3.$$

Для минимизации БФ по методу Квайна все элементарные конъюнкции в записи ее совершенной ДНФ сравниваются попарно. Если две конъюнкции таковы, что имеют вид $a x_i$ и $a \bar{x}_i$, то вместо них выписывается единственная конъюнкция a ($L-1$)-го ранга. Конъюнкции

L -го ранга, для которых произошло склеивание, отмечаются. После построения всех конъюнкций $(L-1)$ -го ранга их вновь сравнивают попарно, в результате формируются конъюнкции $(L-2)$ -го ранга и отмечаются склеивающиеся конъюнкции $(L-1)$ -го ранга. Рассмотренные действия заканчиваются, когда на очередном этапе полученные конъюнкции l -го ранга уже не склеиваются между собой. Все неотмеченные элементарные конъюнкции называются простыми импликантами (им соответствуют все максимальные интервалы БФ).

Пусть задана БФ

$$y = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3,$$

в которой пять элементарных конъюнкций третьего ранга. Образует конъюнкции второго ранга: $x_1 x_2$, $x_1 x_3$, $x_1 \bar{x}_3$, $x_1 \bar{x}_2$, $\bar{x}_2 x_3$. Все элементарные конъюнкции третьего ранга оказались отмеченными. Далее находим конъюнкцию первого ранга: x_1 . Из неотмеченных конъюнкций $\bar{x}_2 x_3$ и x_1 формируется сокращенная ДНФ: $y = x_1 \vee \bar{x}_2 x_3$.

Для нахождения минимального покрытия максимальными интервалами необходимо произвести выбрасывание некоторых простых импликант. С этой целью строится так называемая импликантная матрица, число столбцов которой совпадает с числом элементарных конъюнкций в записи совершенной ДНФ минимизируемой БФ, а число строк равно числу полученных простых импликант. Каждому столбцу (строке) присписывается соответствующая элементарная конъюнкция из совершенной ДНФ (простая импликанта) минимизируемой функции. Если k -я простая импликанта является частью p -й элементарной конъюнкции, то на пересечении k -й строки и p -го столбца ставится метка. Искомое покрытие представляет собой множество интервалов, соответствующих тем строкам импликантной матрицы, которые покрывают метками все ее столбцы. В приведенном ранее примере минимальное покрытие включает обе элементарные конъюнкции $\bar{x}_2 x_3$ и x_1 и ДНФ $y = x_1 \vee \bar{x}_2 x_3$ является минимальной.

Рассмотрим еще один пример минимизации БФ:

$$y = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 x_3.$$

Сокращенная ДНФ этой функции представится в виде $y = x_1 x_2 \vee x_2 x_3 \vee \bar{x}_1 x_3$. Ее импликантная матрица показана в табл. 1.6; минимальная ДНФ функции y запишется в виде

$$y = x_1 x_2 \vee \bar{x}_1 x_3.$$

Таблица 1.6

	$x_1 x_2 x_3$	$x_1 x_2 \bar{x}_3$	$\bar{x}_1 x_2 x_3$	$\bar{x}_1 \bar{x}_2 x_3$
$x_1 x_2$	V	V		
$x_2 x_3$	V		V	
$\bar{x}_1 x_3$			V	V

Существуют различные алгоритмы нахождения минимального покрытия импликантной матрицы, рассмотренные, например, в [1].

Мак-Класки предложил модернизацию первой части метода Квайна, позволяющую существенно сократить число попарных сравнений различных элементарных конъюнкций. Согласно методу Мак-Класки элементарные конъюнкции записываются в виде L -разрядных троичных

в системе (1.1) каждая функция выражена через отрицание, дизъюнкцию и конъюнкцию. Если им сопоставить логические элементы НЕ, ИЛИ и И, то система (1.1) будет реализована комбинационной схемой. На рис. 1.5 представлена КС, реализующая в базисе И, ИЛИ, НЕ систему функций:

$$y_1 = x_1(x_2\bar{x}_3 \vee \bar{x}_2x_3) \vee \bar{x}_1\bar{x}_2;$$

$$y_2 = \bar{x}_1\bar{x}_2 \vee x_3.$$

Для одной и той же системы собственных функций может быть построено несколько КС различной сложности. Часто представляют

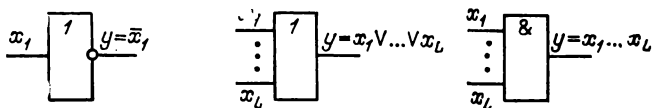


Рис. 1.4

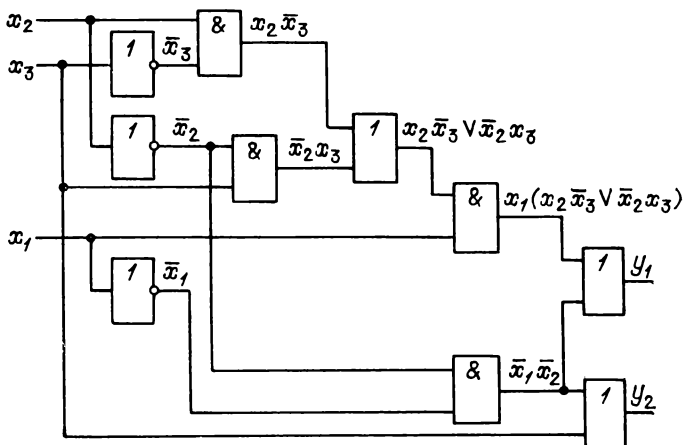


Рис. 1.5

интерес схемы, которые реализуют заданную систему собственных функций при минимально возможном числе логических элементов. Для построения минимальной схемы необходимо упростить систему БФ (1.1). Пусть $N=1$, т. е. комбинационная схема имеет один выход. Тогда для упрощения соответствующей функции могут быть использованы методы минимизации, рассмотренные в § 1.2. Дополнительное упрощение сложности КС можно получить с помощью операции вынесения за скобки.

При синтезе КС с N ($N > 1$) выходами можно минимизировать каждую из функций в системе (1.1), после чего построить соответствующую схему. Известно, однако, что совместная минимизация булевых функций приводит к лучшим реализациям. Существует много алгоритмов решения этой задачи, большинство из них реализовано на ЭВМ. Подробно алгоритмы совместной минимизации систем БФ рассмотрены в [1, 2].

АВТОМАТЫ И ГРАФ-СХЕМЫ

2.1. Абстрактные автоматы

Рассмотрим устройство S с одним входом и одним выходом (рис. 2.1), работающее в некотором идеализированном дискретном времени. Для этого на шкале времени выделим некоторые точки — последовательные моменты времени $t=0, 1, 2, \dots$ — и предположим, что в каждый из этих моментов на вход устройства поступает входной сигнал — одна из букв входного алфавита Z , а на его выходе появляется выходной сигнал — буква выходного алфавита W . Под алфавитом здесь понимается непустое множество попарно различных символов — букв этого алфавита. Конечная упорядоченная последовательность букв образует слово в данном алфавите. Пусть, например, $Z=\{z_1, z_2, z_3\}$, $W=\{\omega_1, \omega_2, \omega_3, \omega_4\}$, а само устройство таково, что в ответ на входное слово $z_2z_1z_1z_3z_2z_2$ на его выходе появляется выходное слово $\omega_1\omega_3\omega_4\omega_2\omega_1\omega_3$:

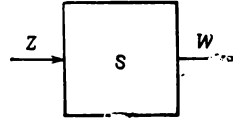


Рис. 2.1

t	0	1	2	3	4	5	
$z(t)$	z_2	z_1	z_1	z_3	z_2	z_2	(2.1)
$\omega(t)$	ω_1	ω_3	ω_4	ω_2	ω_1	ω_3	

Как видно из (2.1), реакция устройства на одну и ту же входную букву может быть различной. Для этого достаточно сравнить входные и выходные сигналы в моменты времени 1 и 2, 4 и 5. Другими словами, сигнал ω_t на выходе устройства в каждый момент времени t зависит не только от входного сигнала z_t в этот же момент t , но и от предыстории, т. е. в устройстве должна быть память о том, какое слово поступило на его вход до рассматриваемого момента времени. Математической моделью подобных устройств является абстрактный автомат, изучаемый в теории автоматов. Термин «абстрактный» используется в связи с идеализированным дискретным временем, а также потому, что в этой модели абстрагируются от реальной физической природы входных и выходных сигналов, рассматривая их просто как буквы некоторого алфавита.

Абстрактный автомат определяется как шестикомпонентный кортеж $S=(A, Z, W, \delta, \lambda, a_1)$, у которого:

1) $A=\{a_1, \dots, a_m, \dots, a_m\}$ — множество состояний (алфавит состояний);

2) $Z=\{z_1, \dots, z_f, \dots, z_f\}$ — множество входных сигналов (входной алфавит);

3) $W=\{\omega_1, \dots, \omega_g, \dots, \omega_g\}$ — множество выходных сигналов (выходной алфавит);

4) функция переходов δ определяет состояние автомата¹ в следующий момент времени $t+1$ в зависимости от состояния автомата и входного сигнала в момент времени t . Другими словами, функция δ ставит в соответствие паре состояние — входной сигнал (a_m, z_f) со-

¹ В этом параграфе под термином «автомат» понимается абстрактный автомат.

стояние автомата a_s , в которое он переходит из a_m под действием сигнала z_f , т. е. $a_s = \delta(a_m, z_f)$;

5) функция выходов λ , вид которой приводит к разделению автоматов на два класса, получивших наибольшее распространение, — автоматы Мили (Mealy) и Мура (Moore). В автомате Мили функция λ ставит в соответствие паре состояние — входной сигнал (a_m, z_f) выходной сигнал w_g , т. е. $w_g = \lambda(a_m, z_f)$; в автомате Мура функция λ ставит в соответствие состоянию a_m выходной сигнал w_g , т. е. $w_g = \lambda(a_m)$;

6) a_1 — начальное состояние автомата ($a_1 \in A$), в котором он находится в момент времени $t=0$.

Абстрактный автомат (см. рис. 2.1) имеет один вход, один выход и работает в дискретном времени, принимаящем целые неотрицательные значения $t=0, 1, 2, \dots$. В любой момент t дискретного времени автомат находится в некотором состоянии $a(t) \in A$, причем в начальный момент времени ($t=0$) он установлен в начальное состояние $a(0) = a_1$. Будучи в момент времени t в состоянии $a(t)$, автомат способен воспринять на своем входе сигнал $z(t) \in Z$. В соответствии с функцией выходов в этот же момент времени он выдаст выходной сигнал $w(t) \in W$ и в следующий момент времени $(t+1)$ согласно функции переходов перейдет в состояние $a(t+1) \in A$. Если на вход автомата, установленного в начальное состояние a_1 , подавать некоторую последовательность входных сигналов $z(0), z(1), z(2), \dots$ — входное слово, то на его выходе будут формироваться выходные сигналы $w(0), w(1), w(2), \dots$ — выходное слово. Относя к каждому входному слову соответствующее ему выходное слово, получаем отображение, индуцированное абстрактным автоматом. Рассмотренные в гл. 1 комбинационные схемы (КС) также составляют подкласс автоматов, у которых выходные сигналы не зависят от предыстории и в каждый момент времени определяются лишь входными сигналами в этот же момент времени. На абстрактном уровне КС могут рассматриваться как автоматы с одним внутренним состоянием. Автоматы с числом внутренних состояний более одного составляют класс автоматов с памятью.

Автомат называется конечным, если конечны множества A, Z и W , и детерминированным, если в нем выполнены условия однозначности переходов: находясь в некотором состоянии, он не может перейти более чем в одно состояние под действием одного и того же входного сигнала. Автомат носит название инициального, если в нем выделено начальное состояние a_1 .

Автомат называется полностью определенным, если область определения функции δ совпадает с множеством всевозможных пар вида (a_m, z_f) , $a_m \in A, z_f \in Z$. Для автомата Мили область определения функции λ также совпадает с множеством всевозможных пар вида (a_m, z_f) , $a_m \in A, z_f \in Z$, а для автомата Мура — с множеством состояний $a_m \in A$. Автомат называется неполностью определенным или частичным, если либо функция δ определена не на всех парах $(a_m, z_f) \in A \times Z$, либо функция λ определена не на всех указанных парах в случае автомата Мили и на множестве не всех внутренних состояний для автомата Мура.

Для того чтобы задать автомат, необходимо описать все компоненты кортежа $S = (A, Z, W, \delta, \lambda, a_1)$. Среди многообразия различных способов задания автомата наибольшее распространение получили табличный и графический. В первом случае автомат Мили описывается с помощью двух таблиц. Одна из них (таблица переходов) задает функцию δ (табл. 2.1), вторая (таблица выходов) — функцию λ (табл. 2.2). Каждому столбцу табл. 2.1 и 2.2 поставлено в соответствие одно со-

стояние из множества A , каждой строке — один входной сигнал из множества Z . На пересечении столбца a_m и строки z_f в табл. 2.1 записывается состояние a_s , в которое должен перейти автомат из состояния a_m под действием входного сигнала z_f , т. е. $a_s = \delta(a_m, z_f)$. На пересечении столбца a_m и строки z_f в табл. 2.2 записывается выходной сигнал w_g , выдаваемый автоматом в состоянии a_m при поступлении на его вход сигнала z_f , т. е. $w_g = \lambda(a_m, z_f)$.

Таблица 2.1

	a_1	a_2	a_3	a_4
z_1	a_2	a_2	a_1	a_1
z_2	a_4	a_3	a_4	a_3

Таблица 2.2

	a_1	a_2	a_3	a_4
z_1	w_1	w_1	w_2	w_4
z_2	w_5	w_3	w_4	w_5

Для указанных таблиц $A = \{a_1, a_2, a_3, a_4\}$; $Z = \{z_1, z_2\}$. $W = \{w_1, w_2, w_3, w_4, w_5\}$. Автомат Мили может быть также задан одной совмещенной таблицей переходов и выходов (табл. 2.3), в которой каждый элемент a_s/w_g , записанный на пересечении столбца a_m и строки z_f , определяет-ся следующим образом:

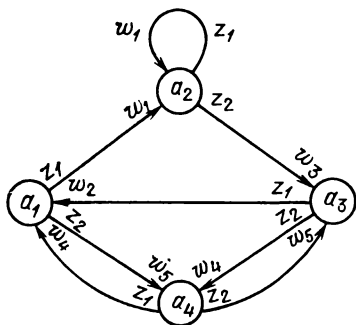
$$a_s = \delta(a_m, z_f); w_g = \lambda(a_m, z_f).$$

Таблица 2.3

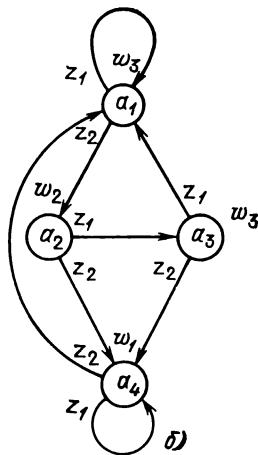
	a_1	a_2	a_3	a_4
z_1	a_2/w_1	a_2/w_1	a_1/w_2	a_1/w_4
z_2	a_4/w_5	a_3/w_3	a_4/w_4	a_3/w_5

Таблица 2.4

	w_3	w_2	w_3	w_1
	a_1	a_2	a_3	a_4
z_1	a_1	a_3	a_1	a_4
z_2	a_2	a_4	a_4	a_1



а)



б)

Рис. 2.2

Автомат Мура задается одной отмеченной таблицей переходов (табл. 2.4), в которой каждому столбцу приписаны не только состояние a_m , но еще и выходной сигнал w_g , соответствующий этому состоянию, где $w_g = \lambda(a_m)$. Для табл. 2.4 $A = \{a_1, a_2, a_3, a_4\}$; $Z = \{z_1, z_2\}$; $W = \{w_1, w_2, w_3\}$.

Для частичных автоматов Мили и Мура в рассмотренных таблицах на месте неопределенных состояний и выходных сигналов ставится прочерк.

Граф автомата — это ориентированный граф, вершины которого соответствуют состояниям, а дуги — переходам между ними. Дуга, направленная из вершины a_m в вершину a_s , задает переход в автомате из состояния a_m в состояние a_s . В начале этой дуги записывается входной сигнал $z_f \in Z$, вызывающий данный переход: $a_s = \delta(a_m, z_f)$. Для графа автомата Мили выходной сигнал $w_g \in W$, формируемый на переходе, записывается в конце дуги, а для автомата Мура — рядом с вершиной a_m , отмеченной состоянием a_m , в котором он формируется. Если переход в автомате из состояния a_m в состояние a_s производится под действием нескольких входных сигналов, то дуге графа, направленной из a_m в a_s , приписываются все эти входные и соответствующие выходные сигналы. Графы автоматов Мили и Мура, построенные по табл. 2.3 и 2.4, приведены соответственно на рис. 2.2, а, б.

2.2. Структурные автоматы

В отличие от абстрактного автомата, имеющего один вход и один выход, на которые поступают сигналы во входном $Z = \{z_1, \dots, z_F\}$ и выходном $W = \{w_1, \dots, w_G\}$ алфавитах, структурный автомат (рис. 2.3)

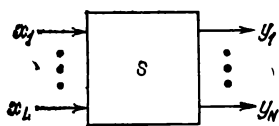
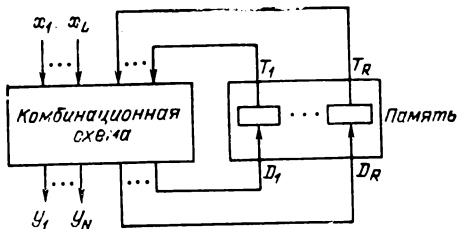


Рис. 2.3

Рис. 2.4. →



имеет L входных x_1, \dots, x_L и N выходных y_1, \dots, y_N полюсов, на каждом из которых сигнал может принимать два значения (0 или 1). В этом случае каждому входному сигналу z_f абстрактного автомата соответствует некоторый двоичный вектор (e_{f1}, \dots, e_{fL}) и $e_{fi} \in \{0, 1\}$. Для представления (кодирования) входных сигналов z_1, \dots, z_F абстрактного автомата различными двоичными векторами должно быть выполнено условие $L \geq \text{int} \log_2 F$, где $\text{int} \log_2 F$ означает ближайшее целое число, не меньшее $\log_2 F$.

Компоненты вектора (e_{f1}, \dots, e_{fL}) представляют собой набор значений переменных x_1, \dots, x_L , поставленный во взаимно-однозначное соответствие символу z_f алфавита Z . Предположим, что некоторая переменная x_i может принимать в этом наборе значения как 0, так и 1, т. е. символ z_f кодируется двумя двоичными векторами: $(e_{f1}, \dots, e_{fi-1}, 0, e_{fi+1}, \dots, e_{fL})$ и $(e_{f1}, \dots, e_{fi-1}, 1, e_{fi+1}, \dots, e_{fL})$. В этом случае, как и в гл. 1, будем считать, что z_f закодирован одним троичным вектором $(e_{f1}, \dots, e_{fi}, \dots, e_{fL})$, в котором $e_{fi} = -$. Тогда компоненты любого такого вектора могут принимать значения из множества $\{0, 1, -\}$, причем

если $e_{fi} = -$, то в соответствующем наборе значений переменных $x_1, \dots, x_l, \dots, x_L$ безразлично, равно ли x_l нулю или единице.

Точно так же выходному сигналу w_g ($g = \overline{1, G}$) абстрактного автомата соответствует некоторый двоичный вектор $(e_{g1}, \dots, e_{gn}, \dots, e_{gN})$ и $e_{gn} \in \{0, 1\}$. Для кодирования выходных сигналов w_1, \dots, w_a абстрактного автомата различными двоичными векторами должно быть выполнено условие $N \geq \text{intlog}_2 G$. Как и ранее, допустим, что рассмотренный вектор $(e_{g1}, \dots, e_{gn}, \dots, e_{gN})$ может быть троичным и у него $e_{gn} \in \{0, 1, -\}$, $n = \overline{1, N}$.

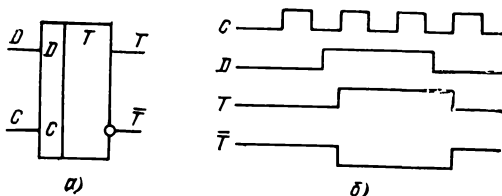


Рис. 2.5

Если абстрактный автомат был лишь математической моделью дискретного устройства, то в структурном автомате учитывается структура входных и выходных сигналов, а также его внутреннее устройство на уровне структурных схем. Структурным синтезом занимается структурная теория, основной задачей которой является построение логической схемы автомата из элементарных автоматов, принадлежащих к заранее заданному конечному числу типов.

На этапе структурного синтеза автомат принято представлять в виде двух частей — памяти и КС (рис. 2.4). Память автомата состоит из предварительно выбранных элементов памяти — элементарных полных автоматов Мура T_1, \dots, T_R . Не останавливаясь подробно на свойствах подобных автоматов, заметим, что на протяжении всей книги мы будем использовать один из самых простых и в то же время наиболее распространенный элемент памяти — D -триггер с информационным D и синхронизирующим C входами. Условное обозначение D -триггера и временная диаграмма его работы приведены на рис. 2.5, а, б соответственно. В D -триггере, как и во всех автоматах Мура, используемых в качестве элементов памяти, каждое состояние отмечено своим, отличным от других выходным сигналом, что позволяет отождествить его состояния с выходными сигналами. Другими словами, если триггер находится в состоянии 0 или 1, то с его выхода также снимается соответственно сигнал 0 или 1. Поэтому вместо отмеченной таблицы переходов в табл. 2.5 приведена просто таблица переходов D -триггера. В этой таблице вход C опущен, но следует иметь в виду, что изменение состояния D -триггера возможно лишь при наличии 1 на его синхронизирующем входе C .

Аналогично кодированию входных и выходных сигналов каждому состоянию a_m ($m = \overline{1, M}$) абстрактного автомата в структурном автомате поставим в соответствие двоичный вектор $(e_{m1}, \dots, e_{mr}, \dots, e_{mR})$, $e_{mr} \in \{0, 1\}$. Для кодирования состояний a_1, \dots, a_M абстрактного автомата различными двоичными векторами должно быть выполнено условие $R \geq \text{intlog}_2 M$. Как и ранее, будем допускать, что двоичный вектор $(e_{m1}, \dots, e_{mr}, \dots, e_{mR})$ может быть заменен троичным, у которого $e_{mr} \in \{0, 1, -\}$, $r = \overline{1, R}$. Закодируем состояния, входные и выходные

Таблица 2.5

D	T	
	0	1
0	0	0
1	1	1

Таблица 2.6

x_1	$T_1 T_2$			
	00	01	10	11
0	01/000	01/000	00/001	00/011
1	11/100	10/010	11/011	10/100

сигналы абстрактного автомата, заданного табл. 2.3, следующими кодами:

$$\begin{aligned}
 a_1 &\rightarrow (00); & z_1 &\rightarrow (0); & w_1 &\rightarrow (000); \\
 a_2 &\rightarrow (01); & z_2 &\rightarrow (1); & w_2 &\rightarrow (001); \\
 a_3 &\rightarrow (10); & & & w_3 &\rightarrow (010); \\
 a_4 &\rightarrow (11); & & & w_4 &\rightarrow (011); \\
 & & & & w_5 &\rightarrow (100).
 \end{aligned}$$

Таблица 2.7

x_1	T_1	T_2	D_1	D_2	y_1	y_2	y_3
0	0	0	0	1	0	0	0
0	0	1	0	1	0	0	0
0	1	0	0	0	0	0	1
0	1	1	0	0	0	1	1
1	0	0	1	1	1	0	0
1	0	1	1	0	0	1	0
1	1	0	1	1	0	1	1
1	1	1	1	0	1	0	0

Заменим в табл. 2.3 состояния, входные и выходные сигналы полученными кодами, в результате чего получим совмещенную таблицу переходов и выходов структурного автомата Мили (табл. 2.6). В этом автомате два элемента памяти (T_1, T_2), один входной (x_1) и три выходных (y_1, y_2, y_3) полюса.

Переходу абстрактного автомата из состояния a_m в состояние a_s под действием входного сигнала z_f с выдачей выходного сигнала w_g соответствует переход структурного автомата из состояния (e_{m1}, \dots, e_{mR}) в состояние (e_{s1}, \dots, e_{sR}) под действием входного сигнала $(e_{f1}, \dots$

$\dots, e_{fL})$ с выдачей выходного сигнала (e_{g1}, \dots, e_{gN}) . Изменение состояний элементов памяти на таком переходе происходит под действием сигналов D_1, \dots, D_R на входах памяти автомата (см. рис. 2.4). Таким образом, после выбора элементов памяти и кодирования символов алфавитов Z, W и A синтез структурного автомата сводится к синтезу КС, реализующей функции:

$$\begin{aligned}
 y_n &= y_n(T_1, \dots, T_R, x_1, \dots, x_L), \quad n = \overline{1, N}; \\
 D_r &= D_r(T_1, \dots, T_R, x_1, \dots, x_L), \quad r = \overline{1, R}.
 \end{aligned} \tag{2.2}$$

где T_1, \dots, T_R — функции обратной связи от памяти автомата к его КС, а D_1, \dots, D_R — функции возбуждения элементов памяти автомата.

Перейдем от табл. 2.6 к табл. 2.7. Последняя представляет систему БФ (2.2), записанную в привычном для нас табличном виде (см. § 1.1). Полученной табличной записи соответствует следующая ДНФ

функций D_1, D_2, y_1, y_2, y_3 :

$$D_1 = x_1 \bar{T}_1 \bar{T}_2 \vee x_1 \bar{T}_1 T_2 \vee x_1 T_1 \bar{T}_2 \vee x_1 T_1 T_2;$$

$$D_2 = \bar{x}_1 \bar{T}_1 \bar{T}_2 \vee \bar{x}_1 \bar{T}_1 T_2 \vee x_1 \bar{T}_1 \bar{T}_2 \vee x_1 T_1 \bar{T}_2;$$

$$y_1 = x_1 \bar{T}_1 \bar{T}_2 \vee x_1 T_1 T_2;$$

$$y_2 = \bar{x}_1 T_1 T_2 \vee x_1 \bar{T}_1 T_2 \vee x_1 T_1 \bar{T}_2;$$

$$y_3 = \bar{x}_1 T_1 \bar{T}_2 \vee \bar{x}_1 T_1 T_2 \vee x_1 T_1 \bar{T}_2.$$

(2.3)

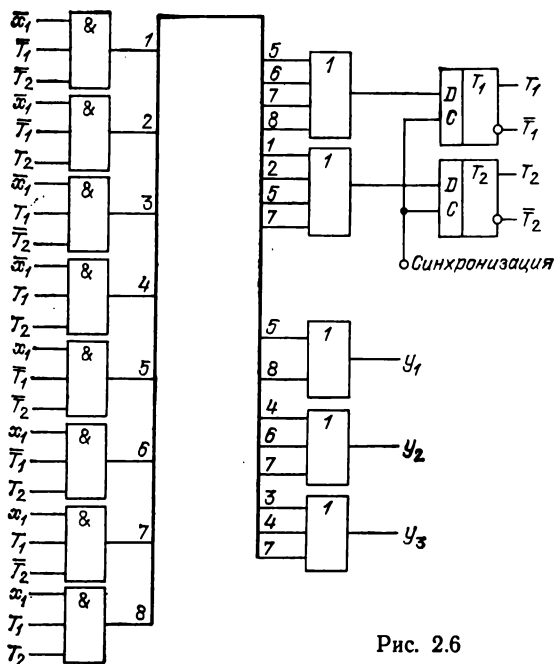


Рис. 2.6

Ясно, что табл. 2.7 можно не строить, а выражения (2.3) получать непосредственно по табл. 2.6. Не останавливаясь здесь на вопросах минимизации КС автомата, непосредственно по выражениям (2.3) построим его логическую схему из элементов И, ИЛИ, НЕ (рис. 2.6). Эта процедура построения логической схемы автомата носит название канонического метода его структурного синтеза.

Рассмотрим еще один метод, называемый графическим методом структурного синтеза. Представим автомат в виде графа. Так как при получении (2.3) используются конъюнкции входных переменных, соответствующих кодам входных сигналов, договоримся в начале дуг графа автомата вместо входных сигналов записывать сразу эти конъюнкции. Так, вместо сигналов 0 и 1 в начале дуг графа будут записаны соответственно \bar{x}_1 и x_1 . Кроме того, поскольку в табличных методах синтеза для получения ДНФ функций y_1, \dots, y_n ищутся только компоненты выходных наборов, принимающие значение 1, будем в конце дуг графа записывать именно эти компоненты. Так, вместо выходных наборов 010,

011 и 100 будем записывать соответственно y_2 , y_2y_3 и y_1 , а вместо выходного набора 000 ставить прочерк (—). На рис. 2.7 изображен граф автомата, заданного табл. 2.6.

Из таблицы переходов D -триггера (см. табл. 2.5) видно, что для перевода его в состояние $T_r=1$ функция возбуждения этого триггера должна быть равна единице. Тогда при синтезе автомата на D -триггерах все дуги графа, входящие в состояние, в коде которого $T_r=1$, отметим символом D_r . Так, на рис. 2.7 всем дугам, входящим в a_3 (его код 10), приспан символ D_1 . Обозначим через a_m конъюнкцию, соответствующую коду $K(a_m)$ состояния a_m . Например, если $K(a_1)=00$ и $K(a_2)=01$, то $a_1=\bar{T}_1\bar{T}_2$ и $a_2=\bar{T}_1T_2$. Таким образом, $a_m=1$ тогда и только тогда, когда автомат находится в состоянии a_m ¹.

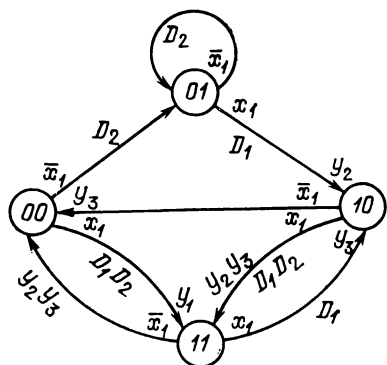


Рис. 2.7

Пусть автомат переходит из состояния a_m в a_s под действием входного сигнала X_f и на этом переходе $D_r=1$. Тогда в ДНФ функции D_r должна войти элементарная конъюнкция a_mX_f , которая равна единице

как раз тогда, когда автомат находится в состоянии a_m ($a_m=1$) и поступает сигнал $X_f=1$. В общем случае выражение для D_r получается как дизъюнкция всевозможных подобных конъюнкций, взятых по всем дугам, которым приспан символ D_r . Выражение для y_n ($n=1, N$) получается как дизъюнкция элементарных конъюнкций вида a_mX_f , где a_m — состояние, приспанное вершине, из которой выходит дуга, отмеченная символом y_n ; X_f — конъюнкция переменных x_1, \dots, x_L , записанная в начале этой дуги.

По графу на рис. 2.7 получаем ДНФ функций возбуждения и функций выходов, которая совпадает с (2.3).

В заключение параграфа отметим, что правильность функционирования автомата в значительной степени определяется правильностью выбора синхронизации переключения его элементов памяти.

2.3. Микропрограммы и граф-схемы

При описании работы широкого класса дискретных систем автоматики и вычислительной техники стало уже классическим их представление в виде композиции операционного (ОА) и управляющего (УА) автоматов (рис. 2.8). ОА выполняет преобразование входных слов информации (двоячных векторов), поступающих на его входы $\gamma_1, \dots, \gamma_f$. Такими словами могут быть, например, слагаемые для операции сложения, множимое и множитель для операции умножения, численное значение переменной (переменных) некоторой сложной функции, вычисляемой в ОА, и т. д. Результаты преобразования, реализуемого в ОА, формируются на выходах β_1, \dots, β_g . Задачей УА является выработка распределенной во времени последовательности управляющих сигналов, под воздействием которых в ОА выполняется некоторая операция.

¹ Здесь и далее под термином «автомат» понимается структурный автомат.

Элементарный неделимый акт обработки информации в ОА, происходящий в течение одного такта автоматного времени, назовем микрооперацией. Пусть $Y = \{y_1, \dots, y_N\}$ — множество микроопераций, реализуемых в ОА. Они возбуждаются сигналами y_1, \dots, y_N из УА, причем если должна быть выполнена микрооперация y_n , то $y_n = 1$ на выходе УА, если же микрооперация y_n не выполняется, то $y_n = 0$ ($n = \overline{1, N}$). Совокупность микроопераций, выполняемых одновременно за один такт автоматного времени, образуют микрокоманду. Для задания порядка следования микрокоманд вводятся специальные переменные,

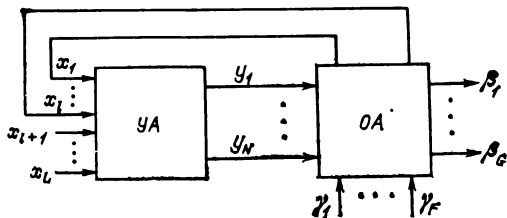


Рис. 2.8

называемые логическими условиями. Проверка значения логического условия в каждом такте работы УА позволяет определить микрокоманду, реализуемую в следующем такте.

Пусть Y_1, \dots, Y_T — микрокоманды, которые могут выполняться в ОА. Тогда последовательность выполнения микрокоманд определяется функциями перехода — булевыми функциями a_{ij} ($i, j = \overline{1, T}$), аргументами которых являются логические условия из множества $X = \{x_1, \dots, x_L\}$. Логическим условиям x_1, \dots, x_L ставятся во взаимно-однозначное соответствие входные переменные x_1, \dots, x_L УА (которые целесообразно обозначить теми же буквами). С каждой микрокомандой Y_i естественно связать множество функций перехода $\{a_{i1}, \dots, a_{iT}\}$ таких, что если $a_{ii} = 1$ после выполнения микрокоманды Y_i , то следующей будет выполняться микрокоманда Y_i . Множество функций перехода из одной и той же микрокоманды Y_i обладает свойствами ортогональности ($a_{ij}a_{it} = 0$ при $j \neq t$) и полноты ($\bigvee_{j=1}^T a_{ij} = 1$). Ортогональ-

ность говорит о том, что непосредственно после данной микрокоманды при определенных значениях логических условий может выполняться только одна микрокоманда, а полнота — что она выполнится обязательно.

Совокупность микрокоманд и функций перехода образует микропрограмму. Таким образом, для описания микропрограммы необходимо задать последовательность микрокоманд и функций перехода, определяющих порядок выполнения микрокоманд. Для описания микропрограммы мы будем использовать язык граф-схем алгоритмов (ГСА).

ГСА — это ориентированный связный граф, включающий вершины четырех типов: начальную, конечную, операторную и условную (рис. 2.9). Конечная, операторная и условная вершины имеют по одному входу, начальная вершина входов не имеет. У начальной и операторной вершин по одному выходу, у условной — два выхода, помеченных символами 1 и 0. Конечная вершина выходов не имеет.

ГСА удовлетворяет следующим условиям:

- 1) входы и выходы вершин соединяются друг с другом с помощью дуг, направленных всегда от выхода ко входу;
- 2) каждый выход соединен точно с одним входом;

- 3) любой вход соединяется, по крайней мере, с одним выходом;
 4) любая вершина ГСА лежит, по крайней мере, на одном пути из начальной вершины к конечной;
 5) в каждой условной вершине записывается один из элементов множества $X = \{x_1, \dots, x_L\}$ логических условий (разрешается в различных условных вершинах запись одинаковых элементов множества X);

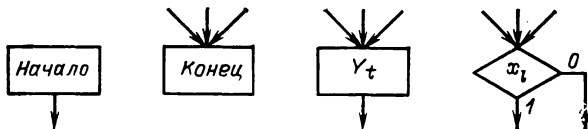


Рис. 2.9

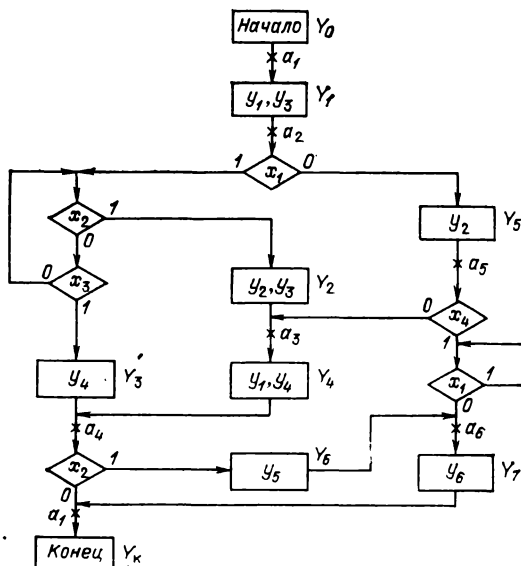


Рис. 2.10

6) один из выходов условной вершины может соединяться с ее входом, что недопустимо для операторной вершины; такие условные вершины будем иногда называть возвратными или ждущими;

7) в каждой операторной вершине записывается оператор (микрокоманда) Y_t — подмножество множества микроопераций $Y = \{y_1, \dots, y_n\}$; допускается $Y_t = \emptyset$; разрешается также запись в различных операторных вершинах одинаковых микрокоманд.

Пример ГСА показан на рис. 2.10. Процесс выполнения ГСА на произвольной последовательности всевозможных наборов значений логических условий x_1, \dots, x_L определен в [3], там же рассмотрены примеры описания на языке ГСА различных микропрограмм.

2.4. Микропрограммные автоматы и их синтез

Управляющий автомат, реализующий микропрограмму работы дискретного устройства, принято называть микропрограммным автоматом (МПА). Синтез МПА Мили по ГСА осуществляется в два этапа: получение отмеченной ГСА и построение графа автомата. На первом этапе выполняется отметка ГСА символами (метками) a_1, \dots, a_M по алгоритму Ф1:

1) символом a_1 отмечаются вход вершины, следующей за начальной, и вход конечной вершины;

2) входы всех вершин, следующих за операторными, кроме входа конечной вершины, отмечаются символами a_2, \dots, a_M ;

3) если вход вершины отмечается, то только одним символом;

4) входы различных вершин, за исключением конечной, отмечаются различными символами.

Применение первого этапа к ГСА на рис. 2.10 дает отмеченную ГСА (символы a_1, \dots, a_6).

Если идти от одной метки a_m к другой a_s в направлении ориентации дуг ГСА, выписывая содержимое лежащих на этом пути вершин, то каждому такому пути можно поставить в соответствие слово в алфавите $\{x_1^{e_1}, \dots, x_L^{e_L}, Y_1, \dots, Y_T\}$, где $e_l \in \{0, 1\}$ — символ, приписанный выходу условной вершины, через который проходит соответствующий путь: $x_l^0 = \overline{x_l}$; $x_l^1 = x_l$, $l = \overline{1, L}$. Если выписанное слово определяет путь из a_m в a_s , то будем ограничивать его слева и справа символами a_m и a_s . В дальнейшем нас будут интересовать слова вида:

$$a_m x_{m1}^{e_{m1}} \dots x_{mR}^{e_{mR}} Y_t a_s \quad (a_m, a_s \in \{a_1, \dots, a_M\}); \quad (2.4)$$

$$a_m x_{m1}^{e_{m1}} \dots x_{mR}^{e_{mR}} a_1 \quad (a_m \in \{a_1, \dots, a_M\}). \quad (2.5)$$

Путь вида (2.4) — это путь из одной отметки в другую, содержащий операторную вершину, а путь (2.5) — это путь в отметку a_1 без операторной вершины.

Соответствующие словам (2.4) и (2.5) пути в ГСА будем называть путями перехода. Для слова (2.4) допустим случай $R=0$, т. е. $a_m Y_t a_s$.

Каждому пути (2.4) или (2.5) можно поставить в соответствие конъюнкцию

$$X(a_m, a_s) = \bigwedge_{r=1}^R x_{mr}^{e_{mr}}$$

и записать их в виде:

$$a_m X(a_m, a_s) Y(a_m, a_s) a_s; \\ a_m X(a_m, a_1) a_1.$$

Здесь $Y(a_m, a_s) = Y_t$ (если $R=0$ — две операторные вершины следуют друг за другом, то $X(a_m, a_s) = 1$).

Сформированные конъюнкции $X(a_m, a_s)$ преобразуются по правилам булевой алгебры. Если в конъюнкцию $X(a_m, a_s)$ переменная x_{mr} входит как со знаком, так и без знака инверсии, то соответствующий этой конъюнкции путь перехода из a_m в a_s далее не рассматривается. Если в конъюнкцию $X(a_m, a_s)$ входит несколько переменных $x_{mr}^{e_{mr}}$, то в ней оставляют только одну такую букву.

Каждой отметке a_1, \dots, a_M поставим в соответствие вершину графа автомата Мили. Если в ГСА существует путь перехода из a_m в a_s вида (2.4) или (2.5), то на графе автомата вершина a_m соединяется дугой с вершиной a_s , направленной из a_m в a_s ; в начале дуги записывается конъюнкция $X(a_m, a_s)$, соответствующая этому пути, в конце дуги — множество микроопераций $Y(a_m, a_s)$ из операторной вершины, через которую проходит этот путь. Согласно ранее изложенному допустимо $Y(a_m, a_s) = \emptyset$. Если в ГСА из a_m в a_s имеется

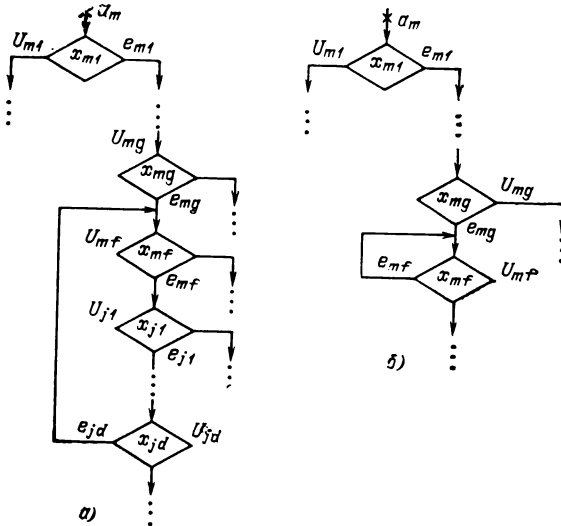


Рис. 2.11

несколько путей перехода, например пути i, j и k -й, то вершины a_m и a_s графа автомата соединяются одной дугой, направленной из a_m в a_s ; в начале дуги друг под другом записываются конъюнкции $X_i(a_m, a_s), X_j(a_m, a_s), X_k(a_m, a_s)$, в конце дуги — множества $Y_i(a_m, a_s), Y_j(a_m, a_s), Y_k(a_m, a_s)$.

Любой i -й переход в полученном графе автомата Мили из состояния a_m в состояние a_s (на графе существует дуга, направленная из a_m в a_s) осуществляется под действием такого набора значений переменных из множества $X = \{x_1, \dots, x_L\}$, на котором равна 1 конъюнкция $X_i(a_m, a_s)$. На i -м переходе формируются единичные значения выходных сигналов из множества $Y_i(a_m, a_s)$.

Если в отмеченной ГСА есть контуры, включающие только условные вершины, для некоторой метки a_m можно выписать слово вида

$$a_m x_{m1}^{e_{m1}} \dots x_{mg}^{e_{mg}} x_{mf}^{e_{mf}} x_{j1}^{e_{j1}} \dots x_{jd}^{e_{jd}} x_{mf}^{e_{mf}} \dots \quad (2.6)$$

(в общем случае оно бесконечной длины). Логические условия $x_{m1}, \dots, x_{mg}, x_{mf}, x_{j1}, \dots, x_{jd}, x_{mf}, \dots$ записаны соответственно в условных вершинах $U_{m1}, \dots, U_{mg}, U_{mf}, U_{j1}, \dots, U_{jd}, U_{mf}, \dots$ (рис. 2.11). Здесь вершина U_{mf} встретилась дважды, и если $d \neq 0$, то в рассматриваемом пути есть контур (рис. 2.11,а), если же $d=0$ — то ждущая вершина U_{mf} (рис. 2.11,б). Далее условную вершину U_{mf} будем назы-

вать вершиной входа в контур. Каждому слову вида (2.6) поставим в соответствие переход автомата из состояния a_m в то же состояние a_m под действием входного сигнала $X(a_m, a_m) = x_{m_1}^{e_m} \dots$

$\dots x_{m_g}^{e_{mg}} x_{m_f}^{e_{mf}} x_{j_1}^{e_{j1}} \dots x_{j_d}^{e_{jd}}$ с выдачей выходного сигнала $Y(a_m, a_m) = \bar{y}$.

Рассмотрим некоторые пути перехода в ГСА на рис. 2.10, отмеченной метками a_1, \dots, a_6 . Путь $a_1 Y_1 a_2$ имеет вид (2.4), причем $X(a_1, a_2) = 1$; $a_2 x_1 \bar{x}_2 x_3 Y_3 a_4$ — вид (2.4); $a_4 \bar{x}_2 a_1$ — вид (2.5); $a_2 x_1 \bar{x}_2 \bar{x}_3 a_2$ — вид (2.6) и $d \neq 0$; $a_5 x_4 x_1$ — вид (2.6) и $d = 0$. На рис. 2.12 изображен граф автомата Мили, синтезированного по ГСА на рис. 2.10.

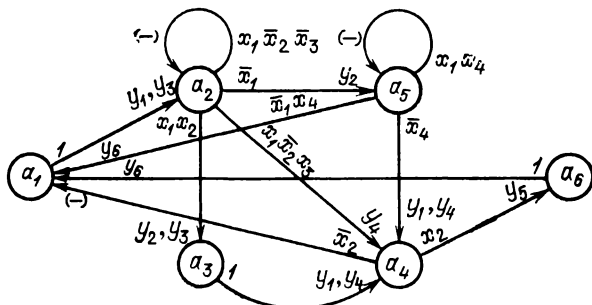


Рис. 2.12

Синтез МПА Мура по ГСА также состоит из двух этапов: получения отмеченной ГСА и построения графа автомата. На первом этапе начальная, конечная и операторные вершины отмечаются символами (метками) a_1, \dots, a_M по алгоритму Φ_2 :

- 1) символом a_1 отмечаются начальная и конечная вершины;
- 2) различные операторные вершины отмечаются различными символами;
- 3) все операторные вершины должны быть отмечены.

Применение первого этапа к ГСА на рис. 2.10 дает отмеченную ГСА, изображенную на рис. 2.13.

Для построения графа автомата Мура в ГСА находятся пути перехода вида

$$a_m x_{m_1}^{e_{m1}} \dots x_{m_R}^{e_{mR}} a_s \quad (a_m, a_s \in \{a_1, \dots, a_M\}). \quad (2.7)$$

Соответствующая этому пути конъюнкция $X(a_m, a_s) = \bigwedge_{r=1}^R x_{m_r}^{e_{m_r}}$ определяется так же, как и ранее. Если между операторными вершинами ГСА a_m и a_s лежит пустое множество условных вершин (вершина, отмеченная символом a_s , следует непосредственно за вершиной, отмеченной символом a_m), то $X(a_m, a_s) = 1$ и путь (2.7) превращается в путь $a_m a_s$. Пусть $Y(a_m) = Y_t$, где Y_t — микрокоманда, записанная в операторной вершине, отмеченной меткой a_m . Каждому символу a_1, \dots, a_M , полученному на первом этапе, поставим в соответствие вершину графа автомата Мура. Если в ГСА существует путь (2.7) из a_m в a_s , то в графе автомата вершина a_m соединяется дугой с вершиной a_s , направленной из a_m в a_s ; в начале дуги записывается конъюнкция $X(a_m, a_s)$, соответствующая этому пути. Рядом с вершиной a_m графа автомата записываются элементы множества $Y(a_m)$. Допу-

стимо $Y(a_m) = \emptyset$. Если в ГСА из a_m в a_s имеется несколько путей вида (2.7), например пути i, j и k -й, то вершины a_m и a_s графа автомата соединяются одной дугой, направленной из a_m в a_s , а в начале дуги друг под другом записываются конъюнкции $X_i(a_m, a_s)$, $X_j(a_m, a_s)$, $X_k(a_m, a_s)$.

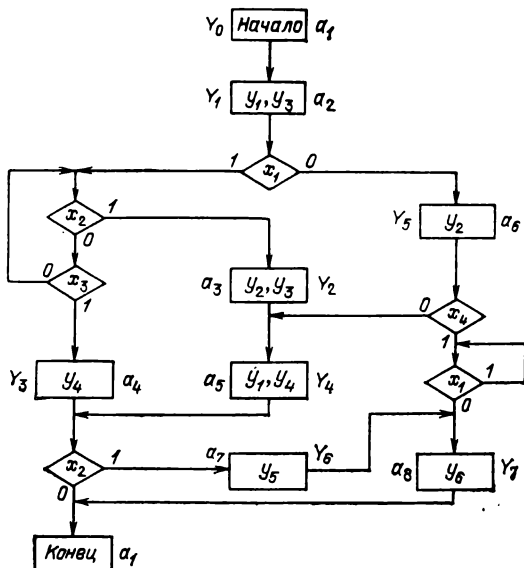


Рис. 2.13

Любой i -й переход в полученном графе автомата Мура из состояния a_m в состояние a_s (на графе существует дуга, направленная из a_m в a_s) осуществляется под действием такого набора значений переменных из множества X , для которого конъюнкция $X_i(a_m, a_s)$ равна 1. После выполнения перехода в состояние a_s формируются выходные сигналы из множества $Y(a_s)$.

При наличии в отмеченной ГСА контуров, включающих только условные вершины (например, в пути перехода из a_m), формируется конъюнкция $X(a_m, a_m)$ по правилам, изложенным ранее. В этом случае рассматривается переход из состояния a_m в то же состояние a_m под действием входного сигнала $X(a_m, a_m)$. Находясь в состоянии a_m , автомат постоянно выдает выходной сигнал $Y(a_m)$.

На рис. 2.14 изображен граф автомата Мура, синтезированный по ГСА на рис. 2.13.

При описании МПА с большим числом состояний и переходов наглядность графического представления теряется, поэтому оказывается более предпочтительным задавать граф автомата в виде списка — таблицы переходов МПА. Для автомата Мили такая таблица содержит четыре столбца: a_m и a_s — исходное состояние и состояние перехода; $X(a_m, a_s)$ — конъюнкция переменных из множества X , принимающая значение 1 на данном переходе; $Y(a_m, a_s)$ — подмножество выходных переменных, принимающих значение 1 на данном переходе. Каждая строка таблицы переходов МПА соответствует одному пути перехода, т. е. дуге графа автомата с одним входным и одним

выходным сигналами. В табл. 2.8 приведена таблица переходов, описывающая граф автомата Мили, изображенный на рис. 2.12. Очевидно, что таблицу переходов МПА целесообразно составлять непосредственно по отмеченной ГСА, записывая в нее все пути перехода, т. е. не нужно предварительно вычерчивать граф автомата.

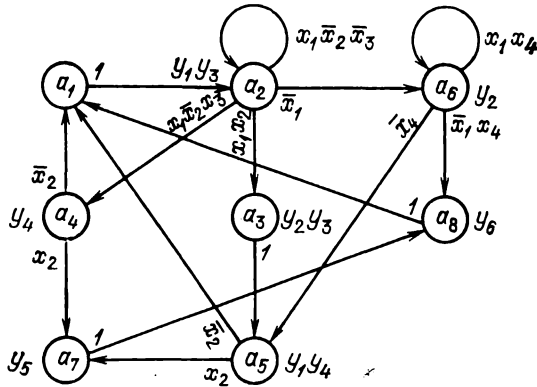


Рис. 2.14

Таблица 2.8

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$
a_1	a_2	1	$y_1 y_3$
a_2	a_2	$\overline{x_1 x_2 x_3}$	—
	a_3	$x_1 x_2$	$y_2 y_3$
	a_4	$x_1 \overline{x_2 x_3}$	y_4
	a_5	$\overline{x_1}$	y_2
a_3	a_4	1	$y_1 y_4$
a_4	a_1	$\overline{x_2}$	—
	a_6	x_2	y_6
a_5	a_1	$\overline{x_1 x_4}$	y_6
	a_4	$\overline{x_4}$	$y_1 y_4$
	a_5	$x_1 x_4$	—
a_6	a_1	1	y_6

Таблица 2.9

$a_m Y(a_m)$	a_s	$X(a_m, a_s)$
a_1	a_2	1
$a_2, y_1 y_3$	a_2	$\overline{x_1 x_2 x_3}$
	a_3	$x_1 x_2$
	a_4	$x_1 \overline{x_2 x_3}$
	a_6	$\overline{x_1}$
$a_3, y_2 y_3$	a_5	1
a_4, y_4	a_1	$\overline{x_2}$
	a_7	x_2
$a_5, y_1 y_4$	a_1	$\overline{x_2}$
	a_7	x_2
	a_6, y_2	a_6
a_6, y_2	a_6	$x_1 x_4$
	a_8	$\overline{x_1 x_4}$
a_7, y_5	a_8	1
a_8, y_6	a_1	1

Таблица переходов МПА Мура содержит три столбца. В ней выходные сигналы из множества $Y(a_m)$ записываются в столбце a_m рядом с состоянием, в котором они формируются. Таблица 2.9 представляет собой таблицу переходов МПА Мура, которая описывает граф автомата на рис. 2.14.

Расширением таблицы переходов МПА является его структурная таблица. Таблицы 2.10—2.11 — структурные таблицы МПА Мили и Мура, построенные соответственно по табл. 2.8 и 2.9. В этих таблицах три новых столбца: $K(a_m) = (e_{m1}, \dots, e_{mR})$ и $K(a_s) = (e_{s1}, \dots, e_{sR})$ — коды исходного состояния a_m и состояния перехода a_s ; $F(a_m, a_s)$ — множество компонентов обязательных (изменяющих состояния элементов памяти) функций возбуждения, вырабатываемых на переходе из a_m в a_s . Если элементами памяти МПА являются D -триггеры, то сигналы возбуждения, записываемые в столбце $F(a_m, a_s)$, должны быть сформированы лишь для тех D -триггеров, номера которых равны номерам разрядов (слева направо) кода $K(a_s)$ состояния a_s , принимающих значение 1. При построении структурной таблицы состояния МПА должны быть закодированы двоичными кодами длины R . Для автоматов, заданных табл. 2.10 и 2.11, состояния закодированы произвольным образом.

Так как структурная таблица есть граф МПА, заданный в виде списка, из нее можно получить выражения функций возбуждения и функций выходов аналогично тому, как это делалось при графическом методе структурного синтеза автоматов. Например, для y_1 и D_1 из табл. 2.10 имеем:

$$y_1 = \overline{T_1} \overline{T_2} \overline{T_3} \vee \overline{T_1} T_2 \overline{T_3} \vee T_1 \overline{T_2} \overline{T_3} \overline{x_4};$$

$$D_1 = \overline{T_1} \overline{T_2} T_3 \overline{x_1} \vee \overline{T_1} T_2 T_3 x_2 \vee T_1 \overline{T_2} \overline{T_3} x_1 x_4.$$

Таблица 2.10

a_n	$K(a_n)$	a_s	$K(a_s)$	$X(a_n, a_s)$	$Y(a_n, a_s)$	$F(a_n, a_s)$
a_1	000	a_2	001	1	$y_1 y_3$	D_3
a_2	011	a_2	001	$x_1 \overline{x_2} \overline{x_3}$	—	D_3
		a_3	010	$x_1 x_2$	$y_2 y_3$	D_2
		a_4	011	$x_1 \overline{x_2} x_3$	y_4	$D_2 D_3$
		a_5	100	$\overline{x_1}$	y_2	D_1
a_3	010	a_4	011	1	$y_1 y_4$	$D_2 D_3$
a_4	011	a_1	000	$\overline{x_2}$	—	—
		a_6	101	x_2	y_5	$D_1 D_3$
a_5	100	a_1	000	$\overline{x_1} x_4$	y_6	—
		a_4	011	$\overline{x_4}$	$y_1 y_4$	$D_2 D_3$
		a_5	100	$x_1 x_4$	—	D_1
a_6	101	a_1	000	1	y_6	—

Таблица 2.11

$a_n Y(a_m)$	$K(a_n)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$F(a_n, a_s)$
a_1	000	a_2	001	1	D_3
$a_2, y_1 y_3$	001	a_2	001	$\overline{x_1 x_2 x_3}$	D_3
		a_3	010	$x_1 x_2$	D_2
		a_4	011	$x_1 x_2 x_3$	$D_2 D_3$
		a_6	101	$\overline{x_1}$	$D_1 D_3$
$a_3, y_2 y_3$	010	a_5	100	1	D_1
a_4, y_4	011	a_1	000	$\overline{x_2}$	—
		a_7	110	x_2	$D_1 D_2$
$a_5, y_1 y_4$	100	a_1	000	$\overline{x_2}$	—
		a_7	110	x_2	$D_1 D_2$
a_6, y_2	101	a_5	100	$\overline{x_4}$	D_1
		a_6	101	$x_1 x_4$	$D_1 D_3$
		a_8	111	$\overline{x_1 x_4}$	$D_1 D_2 D_3$
a_7, y_5	110	a_8	111	1	$D_1 D_2 D_3$
a_8, y_6	111	a_1	000	1	—

Однако для встречающихся на практике автоматов подобные выражения очень сложны—они представляют собой системы из десятков и сотен функций от десятков и сотен переменных. В этом случае известные классические методы минимизации БФ оказываются неприемлемыми. Кроме того, практика показала, что основное снижение цены схемы происходит не за счет минимизации в классе нормальных форм, а в результате выделения ряда функций и подфункций, допускающих совместную минимизацию, и представления систем функций в виде декомпозиции.

Простейшим способом построения логической схемы МПА является совместная реализация функций возбуждения и функций выходов, записанных в одних и тех же строках структурной таблицы. Так, конъюнкция $T_1 T_2 T_3$ из первой строки табл. 2.10 может быть использована при построении функций y_1, y_3, D_3 , конъюнкция $T_1 T_2 T_3 x_1 \overline{x_2} x_3$ из четвертой строки — при построении y_4, D_2, D_3 и т. д.

2.5. Синтез М-автоматов

На практике часто целесообразно синтезировать МПА, у которых в любой строке столбца $X(a_m, a_s)$ таблицы переходов записано не более одной переменной из множества X (будем называть их М-автоматами). Ясно, что ранг любой конъюнкции в ДНФ функций

y_1, \dots, y_N и D_1, \dots, D_R не превышает $R+1$. Это свойство можно использовать при построении логической схемы МПА из элементов с ограниченным числом входов. Кроме того, поскольку любой переход в М-автомате существенно зависит не более чем от одной переменной из множества X , оказывается возможным упростить режимы синхронизации МПА. Ниже будет показано, что логическая схема М-автомата достаточно просто строится из элементов большой степени интеграции с матричной структурой.

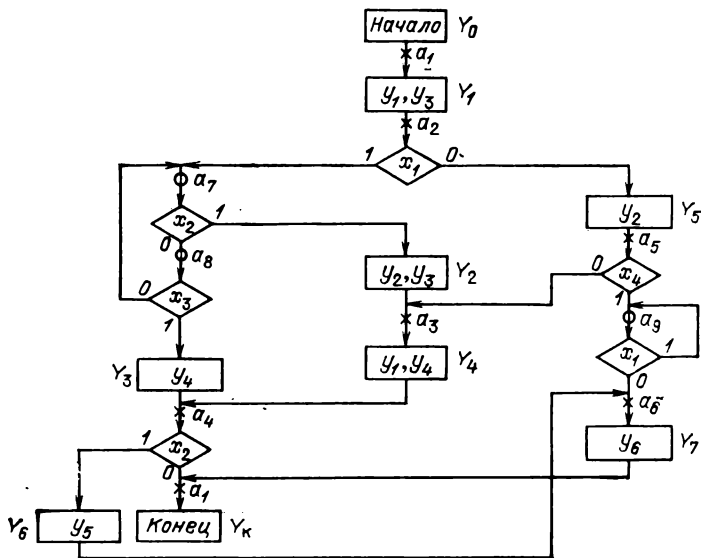


Рис. 2.15

Рассмотрим два алгоритма отметки ГСА (Φ_3 и Φ_4), позволяющие далее синтезировать М-автомат.

Алгоритм Φ_3 :

- 1) ГСА отмечается по алгоритму Φ_1 символами a_1, \dots, a_M ;
- 2) входы условных вершин, которые не отмечены в п. 1, отмечаются символами a_{M+1}, \dots, a_G ;
- 3) входы различных условных вершин отмечаются различными символами.

ГСА на рис. 2.10, отмеченная по алгоритму Φ_3 метками a_1, \dots, a_9 , изображена на рис. 2.15.

Далее нас будут интересовать пути перехода вида

$$a_m x_{mr}^e Y_i a_s \quad (a_m, a_s \in \{a_1, \dots, a_G\}), \quad (2.8)$$

для которых допустимо $Y_i = Y(a_m, a_s) = \emptyset$ (путь без операторной вершины), а также $r=0$ (путь без условной вершины).

Таким образом, возможны три варианта путей вида (2.8):

- 1) путь, содержащий одну условную и одну операторную вершины (например, путь $a_2 x_1 Y_6 a_5$ в ГСА на рис. 2.15);
- 2) путь, содержащий только одну условную вершину (например, $a_2 x_1 a_7$ в той же ГСА);

Таблица 2.12

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$
a_1	a_2	1	y_1, y_3
a_2	a_5	\bar{x}_1	y_2
	a_7	x_1	—
a_3	a_4	1	y_1, y_4
a_4	a_1	\bar{x}_2	—
	a_6	x_2	y_5
a_5	a_4	\bar{x}_4	y_1, y_4
	a_9	x_4	—
a_6	a_1	1	y_5
a_7	a_3	x_2	y_2, y_3
	a_8	\bar{x}_2	—
a_8	a_4	x_3	y_4
	a_7	\bar{x}_3	—
a_9	a_1	\bar{x}_1	y_5
	a_9	x_1	—

Таблица 2.13

$a_m Y(a_m)$	a_s	$X(a_m, a_s)$
a_1	a_2	1
$a_2, y_1 y_3$	a_6	\bar{x}_1
	a_9	x_1
$a_3, y_2 y_3$	a_5	1
a_4, y_4	a_1	\bar{x}_2
	a_7	x_2
$a_5, y_1 y_4$	a_1	\bar{x}_2
	a_7	x_2
a_6, y_2	a_5	\bar{x}_4
	a_{11}	x_4
a_7, y_5	a_8	1
a_8, y_6	a_1	1
a_9	a_3	x_2
	a_{10}	\bar{x}_2
a_{10}	a_4	x_3
	a_9	\bar{x}_3
a_{11}	a_5	\bar{x}_1
	a_{11}	x_1

3) входы различных условных вершин, для которых выполняется требование п. 2, отмечаются различными символами.

ГСА на рис. 2.10, отмеченная по алгоритму Φ_4 метками a_1, \dots, a_{11} , изображена на рис. 2.16. Для построения таблицы переходов M -автомата в ГСА находятся пути перехода вида

$$a_m x_{nr}^{e_m r} a_s \quad (a_m, a_s \in \{a_1, \dots, a_G\}). \quad (2.9)$$

При $r=0$ имеем путь перехода без условной вершины. В этом случае входной сигнал на переходе (a_m, a_s) — конъюнкция $X(a_m, a_s)$ равна 1.

Каждому пути вида (2.9) соответствует одна строка. Выходные переменные из множеств $Y(a_m)$ заносятся в первый столбец этой таблицы так, как показано в § 2.4. Число меток G в ГСА и число строк B в таблице переходов M -автомата, равное числу всех путей (2.9), определяются выражениями: $G=E+F(U)+1$; $B=E+E(U)+2F(U)+\rho$, где $F(U)$ — число таких условных вершин ГСА, в каж-

дую из которых ведет дуга с выхода хотя бы одной из других условных вершин; $E(U)$ — число операторных вершин, после которых следуют условные вершины; $\rho=1$, если после начальной следует операторная вершина, и $\rho=2$, если после начальной следует условная вершина.

Таблица 2.13 представляет собой таблицу переходов МПА, построенную по ГСА на рис. 2.16 после отметки последней по алгоритму Φ_4 . В дальнейшем при построении таблицы переходов МПА будем для краткости говорить, что соответствующий автомат синтезирован по алгоритму Φ_k , имея в виду, что при отметке ГСА использован алгоритм Φ_k .

2.6. Минимизация числа строк таблицы переходов автомата

В § 2.4 отмечалось, что простейший способ синтеза логической схемы МПА заключается в реализации элементарных конъюнкций переменных $T_1, \dots, T_R, x_1, \dots, x_L$, соответствующих строкам структурной таблицы, и использовании их при построении функций возбуждения и выходов. Ясно, что число таких элементарных конъюнкций равно числу строк B в таблице переходов МПА, поэтому, минимизируя B , можно сократить число элементарных конъюнкций и упростить, таким образом, логическую схему автомата. Ниже будет показана важность этой задачи при построении схем автоматов из элементов большой степени интеграции с матричной структурой.

Для сокращения числа B необходимо минимизировать число путей перехода в ГСА. Предварительно предположим, что в ГСА отсутствуют контуры, состоящие только из условных вершин, все входы которых при использовании алгоритма Φ_1 или Φ_2 не отмечены. Позже это ограничение будет снято. Отметим ГСА по алгоритму Φ_1 и будем рассматривать пути вида:

$$a_m x_{m1}^{e_{m1}} \dots x_{mR}^{e_{mR}} Y_i(a_{m1}, a_s) a_s \quad (a_m, a_s \in \{a_1, \dots, a_M\}); \quad (2.10)$$

$$a_m x_{m1}^{e_{m1}} \dots x_{mR}^{e_{mR}} a_s \quad (a_m, a_s \in \{a_1, \dots, a_M\}), \quad (2.11)$$

не проходящие через условные вершины, входы которых отмечены. Условимся также, что путь (2.11) не может вести в метку a_s , которой отмечен вход операторной вершины. Если возникнет необходимость подчеркнуть, что путь (2.10) или (2.11) проходит через условную вершину U_j ГСА, будем использовать запись

$$a_m X(a_m, U_j) U_j X(U_j, a_s) Y(a_m, a_s) a_s,$$

или

$$a_m X(a_m, U_j) U_j X(U_j, a_s) a_s.$$

Назначим каждой вершине U_j ГСА, вход которой не отмечен меткой, цену $C(U_j) = \psi_j \varphi_j - (\psi_j + \varphi_j)$, где ψ_j — число путей вида $a_m X(a_m, U_j) U_j$, ведущих ко входу вершины U_j из других меток ГСА; φ_j — число путей перехода вида $U_j X(U_j, a_s) Y(a_m, a_s) a_s$ или $U_j X(U_j, a_s) a_s$, ведущих из входа вершины U_j в другие метки ГСА. Значение $\psi_j \varphi_j$ определяет число путей вида (2.10) или (2.11), проходящих через условную вершину U_j . Отметим вход вершины U_j меткой $a_j \notin \{a_1, \dots, a_M\}$. Тогда значение $\psi_j + \varphi_j$ будет определять

суммарное число путей вида (2.10) и (2.11), ведущих в метку a_j из других меток и из метки a_i в другие метки. Цена $C(U_i)$ будет показывать, на сколько уменьшится, если $C(U_i) > 0$, или увеличится, если $C(U_i) < 0$, число путей перехода в ГСА при отметке входа вершины U_i меткой a_i .

Если отметить ГСА по алгоритму Φ_2 , то достаточно рассмотреть пути вида (2.11). Тогда в выражении $C(U_i) = \psi_i \varphi_i - (\psi_i + \varphi_i)$ (ψ_i — число путей вида $a_m X(a_m, U_i) U_i$; φ_i — число путей вида $U_i X(U_i, a_s) a_s$ и содержательный смысл цены $C(U_i)$ не изменяется.

Из сказанного ранее следует, что для решения задачи, поставленной в настоящем параграфе, необходимо отмечать метками входы условных вершин ГСА, для которых $C(U_i) > 0$. Обозначим через U^* множество условных вершин ГСА, входы которых не отмечены. Минимизацию числа путей перехода в ГСА будем выполнять по алгоритму Φ_5 :

1) ГСА отмечается символами a_1, \dots, a_M по алгоритму Φ_1 или Φ_2 ;

2) если $U^* = \emptyset$, осуществляется переход к п. 5, в противном случае — к п. 3;

3) ищется такая условная вершина $U_i \in U^*$, для которой цена $C(U_i)$ минимальна; если таких вершин несколько, то выбирается любая из них и осуществляется переход к п. 4;

4) если $C(U_i) > 0$, то вход вершины U_i отмечается меткой, из множества U^* удаляется вершина U_i и осуществляется переход к п. 2, если $C(U_i) \leq 0$ — то к п. 5;

5) конец.

На каждой итерации (исключая последнюю), включающей однократное выполнение пп. 2—4 алгоритма Φ_5 , в ГСА вводится одна новая метка. После отметки входа условной вершины U_i (т. е. после очередной итерации) необходимо пересчитывать цены для всех условных вершин из множества $U^* \setminus \{U_i\}$.

Рассмотрим пример. ГСА, изображенная на рис. 2.17, отмечена по алгоритму Φ_1 метками a_1, \dots, a_9 . Процесс выполнения алгоритма Φ_5 отражен в табл. 2.14, где в столбцах 1—3 указаны цены, вычисленные для условных вершин из множества U^* на итерациях 1—3. На первой итерации вводится метка a_{10} (ею отмечен вход вершины U_6), на второй — a_{11} (вход вершины U_3). Таблица 2.15 представляет собой таблицу переходов МПА, построенную по ГСА на рис. 2.17 с учетом использования алгоритма Φ_5 .

Рассмотрим теперь ГСА с контурами, состоящими только из условных вершин таких, что вход ни одной из этих вершин не отмечен. Например, ГСА, изображенная на рис. 2.18, отмечена по алгоритму Φ_2 метками a_1, \dots, a_7 и содержит контур, включающий условные вершины U_4 и U_5 . Вершина U_4 является вершиной входа в контур (см. § 2.4). Если определить цену $C(U_5)$, то $\psi_5 = 2$, $\varphi_5 = 5$ и $C(U_5) = 3 > 0^*$. Легко, однако, проверить, что отметка входа U_5 приведет к увеличению на три, а не к уменьшению числа путей перехода в ГСА. Это объясняется следующим. В ГСА, отмеченной метками a_1, \dots, a_7 , через нулевой выход условной вершины U_4 идут лишь пути перехода из вершин, отмеченных метками a_1 и a_4 , причем не существует пути перехода, проходящего одновременно через вход условной вершины U_5 и нулевой выход условной вершины U_4 . Если же отметить вход вершины U_5 меткой, то образуются сразу три таких пути, идущих в вершины, отмеченные метками a_3, a_5, a_7 . Эти пути учитываются при вычислении значения φ_5 .

*) Здесь путь вида (2.6) в φ_5 учтен как один путь.

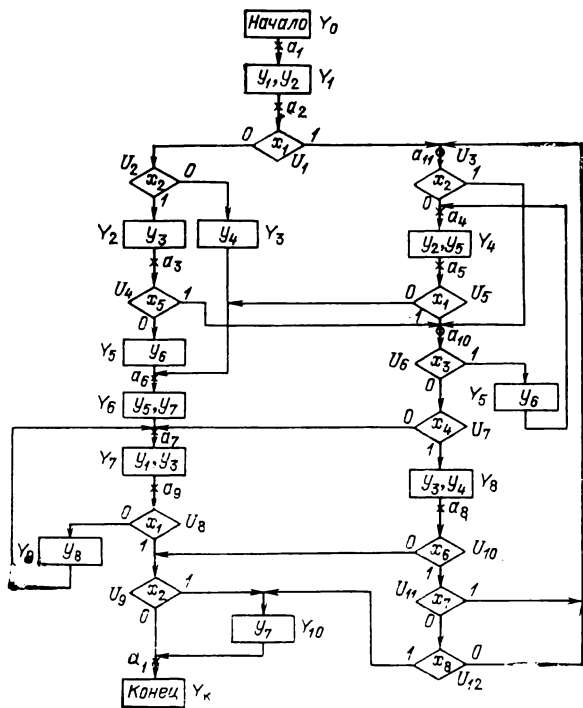


Рис. 2.17

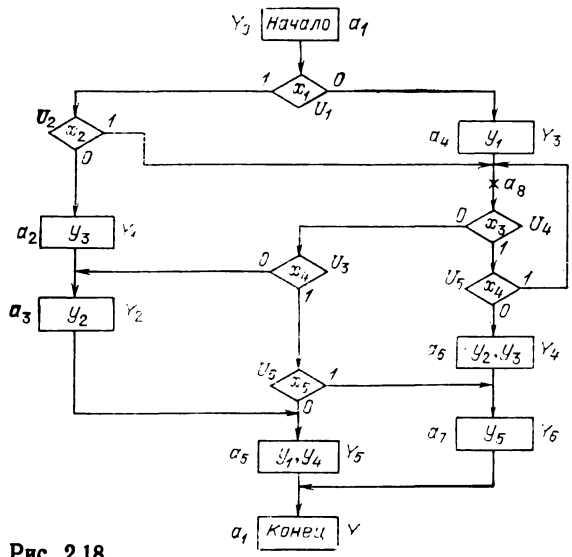


Рис. 2.18

Обозначим через U^* множество вершин, включенных в рассматриваемые контуры, за исключением вершин, являющихся входами контуров. Напомним еще раз, что если вход хотя бы одной условной вершины в контуре отмечен меткой, то такой контур далее не рассматривается, т. е. его вершины не включаются в множество U^* . Для каждой условной вершины $U_f \in U_h \cap U^*$ вычислим цену $C(U_f)$, подсчитав, на сколько изменится число путей перехода в ГСА до и после

Таблица 2.14

Итерация	1	2	3
$C(U_2)$	-1	-1	-1
$C(U_3)$	5	1	
$C(U_6)$	7		
$C(U_7)$	3	-1	-1
$C(U_9)$	0	0	0
$C(U_{11})$	-1	-1	-1
$C(U_{12})$	-1	-1	-1

Другими словами, подсчитываются все пути перехода через условную вершину U_f , вход которой не отмечен, затем отмечается вход U_f и подсчитывается суммарное число путей перехода, входящих в U_f и выходящих из него. Цена $C(U_f)$ есть разность в числе путей перехода до и после отметки входа U_f . Рассмотренный ранее алгоритм Φ_5 корректируется следующим образом. Если $U_f \in U^* \cap U^*$, то значение $C(U_f)$ определяется с помощью рассмотренных правил,

Таблица 2.15

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$
a_1	a_2	1	y_1, y_2	a_8	a_1	$\overline{x_2 x_6}$	—
a_2	a_3	$\overline{x_1 x_2}$	y_3	a_1	a_1	$\overline{x_2 x_6}$	y_7
	a_6	$\overline{x_1 x_2}$	y_4	a_1	a_1	$\overline{x_6 x_7 x_8}$	y_7
	a_{11}	x_1	—	a_{11}	a_{11}	$\overline{x_6 x_7}$	—
a_3	a_6	$\overline{x_5}$	y_6	a_9	a_1	$\overline{x_1 x_2}$	—
	a_{10}	x_5	—		a_1	a_1	$\overline{x_1 x_2}$
a_4	a_5	1	y_2, y_5		a_7	a_7	$\overline{x_1}$
a_5	a_7	$\overline{x_1}$	y_5, y_7	a_{10}	a_4	$\overline{x_3}$	y_6
	a_{10}	x_1	—		a_8	a_8	$\overline{x_3 x_4}$
a_6	a_7	1	y_5, y_7		a_9	a_9	$\overline{x_3 x_4}$
	a_9	1	y_1, y_3	a_{11}	a_5	$\overline{x_2}$	y_2, y_5
a_7	a_9	1	y_1, y_3		a_{10}	a_{10}	x_2

Таблица 2.16

Итерация	1	2
$C(U_1)$	-1	-1
$C(U_2)$	-1	-1
$C(U_3)$	1	-1
$C(U_4)$	3	
$C(U_5)$	-3	-1
$C(U_6)$	0	-1

если же $U_i \in U^*$, $U_i \notin U^k$, то $C(U_i) = \psi_i \varphi_i - (\psi_i + \varphi_i)$. В остальном этот алгоритм не изменяется.

В табл. 2.16 отражена последовательность выполнения алгоритма Φ_5 для ГСА, изображенной на рис. 2.18 и отмеченной предварительно по алгоритму Φ_2 символами a_1, \dots, a_7 . На первой итерации единственной новой меткой a_8 отмечается вход вершины U_4 . Таблица 2.17 представляет собой таблицу переходов МПА, построенную по ГСА на рис. 2.18 с учетом использования алгоритма Φ_5 .

Таблица 2.17

$a_m Y(a_m)$	a_s	$X(a_m, a_s)$
a_1	a_2	$\overline{x_1 x_2}$
	a_4	$\overline{x_1}$
	a_8	$x_1 x_2$
a_2, y_3	a_3	1
a_3, y_2	a_5	1
a_4, y_1	a_8	1
$a_5, y_1 y_4$	a_1	1
$a_6, y_2 y_3$	a_7	1
a_7, y_5	a_1	1
a_8	a_3	$\overline{x_3 x_4}$
	a_5	$\overline{x_3 x_4 x_5}$
	a_6	$\overline{x_3 x_4}$
	a_7	$\overline{x_3 x_4 x_5}$
	a_8	$x_3 x_4$

ГЛАВА ТРЕТЬЯ.

ПРОГРАММИРУЕМЫЕ ЛОГИЧЕСКИЕ УСТРОЙСТВА С МАТРИЧНОЙ СТРУКТУРОЙ

3.1. Матричная реализация булевых функций

В качестве функциональных узлов больших интегральных схем, ориентированных на реализацию булевых функций, широко используются так называемые матричные схемы. Матричная схема представляет собой сетку ортогональных проводников, в местах пересечения которых могут быть установлены полупроводниковые элементы с односторонней проводимостью (ЭОП) — диоды или транзисторы.

Рассмотрим матрицы M_1 и M_2 , изображенные соответственно на рис. 3.1, а и б. Матрица M_1 имеет шесть горизон-

тальных и четыре вертикальные шины. Каждая j -я входная шина этой матрицы ($j=1,3$) связана с двумя горизонтальными шинами M_1 (с одной непосредственно, а со второй — через инвертор). Способ включения ЭОП в местах пересечения шин матрицы M_1 позволяет реализовать на любом из ее выходов любую конъюнкцию ее входных переменных, взятых со знаком либо без знака инверсии. Так, в матрице

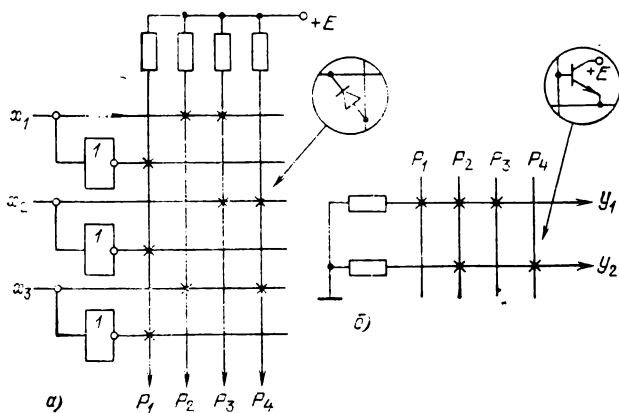


Рис. 3.1

на рис. 3.1,а $P_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3$; $P_2 = x_1 x_3$; $P_3 = x_1 x_2$; $P_4 = x_2 x_3$. Матрица M_2 (рис. 3.1,б) имеет четыре вертикальные и две горизонтальные шины. Способ включения ЭОП в местах пересечения шин M_2 позволяет реализовать на любом из ее выходов любую дизъюнкцию ее входных переменных. На рис. 3.1,б показан пример реализации элементарных дизъюнкций: $y_1 = P_1 \vee P_2 \vee P_3$; $y_2 = P_2 \vee P_4$. Если соединить матрицы M_1 и M_2 , изображенные на рис. 3.1,а и б, так, как это показано на рис. 3.2, то полученная схема будет реализовывать следующую систему булевых функций, представленную в ДНФ:

$$y_1 = P_1 \vee P_2 \vee P_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 \vee x_1 x_2;$$

$$y_2 = P_2 \vee P_4 = x_1 x_3 \vee x_2 x_3.$$

Аналогичным образом любая ДНФ системы θ булевых функций y_1, \dots, y_N входных переменных x_1, \dots, x_L может быть реализована двухуровневой матричной схемой, на первом уровне которой образуются различные элементарные конъюнкции, а на втором — дизъюнкции y_1, \dots, y_N соответствующих конъюнкций. В итоге построение схем с матричной структурой сводится к определению точек пересечения

шин, где должны быть включены ЭОП, и настройке матриц (установке ЭОП в найденных точках).

По способу программирования различают матрицы, настраиваемые (программируемые) на заводе-изготовителе, пользователем и репрограммируемые (многократно настраиваемые). Далее договоримся перед сокращенным наименованием устройств, построенных на основе матриц первого типа, добавлять букву М (от слова масочно-программируемые), второго типа — букву П (от слова программируемые пользователем) и третьего — букву Р (от слова репрограммируемые).

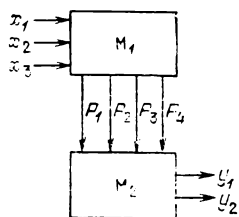


Рис. 3.2

Таблица 3.1

x_1	x_2	x_3	y_1	y_2
0	0	0	1	.
1	—	1	1	1
1	1	—	1	.
—	1	1	.	1

В М-матрицах соединение ЭОП с шинами осуществляется один раз с помощью специальных масок, используемых для металлизации определенных участков кристалла БИС. После изготовления БИС полученные соединения изменены быть не могут. П-матрицы поставляются потребителю ненастроенными и содержащими ЭОП в каждой точке пересечения их шин. Настройка сводится к удалению (отключению) некоторых ненужных ЭОП. Физически процесс настройки осуществляется различными способами, например путем пропускания серии импульсов тока достаточно большой амплитуды через соответствующий ЭОП и разрушения плавкой перемычки, включенной последовательно с этим ЭОП и соединяющей его с одной из шин в точке их пересечения. Р-матрицы позволяют осуществлять программирование неоднократно. Повторное программирование выполняется электрическим способом после стирания содержимого матриц под действием ультрафиолетового (иногда рентгеновского) облучения или электрическим способом отдельно для каждого ЭОП.

Сложность матричной реализации БФ принято оценивать суммарной информационной емкостью (площадью) матриц. В общем случае если в схеме, аналогичной рис. 3.2,

имеется L входов, N выходов и B вертикалей, то

$$S(M) = S(M_1) + S(M_2) = 2LB + BN \text{ [бит]}.$$

Для сокращения информационной емкости при реализации системы БФ необходимо представлять ее в ДНФ с минимальным числом различных элементарных конъюнкций, т. е. в кратчайшей ДНФ.

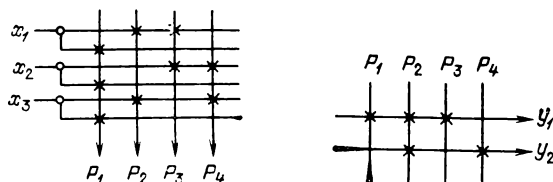


Рис. 3.3

Матрицы M_1 и M_2 при реализации системы БФ принято условно изображать в виде таблицы, столбцы которой отмечаются переменными x_1, \dots, x_L и функциями y_1, \dots, y_N . Каждой промежуточной шине P_1, \dots, P_B ставится в соответствие строка таблицы. На пересечении j -й строки и столбца x_l ($l = \overline{1, L}$) записываются:

1, если переменная x_l входит в j -ю элементарную конъюнкцию без инверсии;

0, если переменная x_l входит в j -ю элементарную конъюнкцию с инверсией;

прочерк (—), если переменная x_l не входит в j -ю элементарную конъюнкцию.

На пересечении j -й строки и столбца y_n ($n = \overline{1, N}$) записываются 1, если j -я элементарная конъюнкция входит в ДНФ функции y_n , и точка (.) — в противном случае. В табл. 3.1 приведено условное представление матриц из рис. 3.2. Для простоты будем изображать схемы на рис. 3.1 так, как это показано на рис. 3.3.

3.2. Программируемые логические матрицы

Программируемая логическая матрица (ПЛМ) представляет собой функциональный блок, созданный на базе полупроводниковой технологии и предназначенный для реализации логических схем цифровых систем. В зависимости от внутренней организации программируемые логические матрицы можно разделить на ПЛМ комбинационной логики и ПЛМ с памятью. Среди ПЛМ первого типа наибольшее распространение получили двухуровневые

(рис. 3.4,а), которые состоят из двух матриц M_1 и M_2 , образующих соответственно первый и второй уровни схемы. Матрица M_1 имеет s входов и q выходов. Она позволяет реализовать q элементарных конъюнкций P_1, \dots, P_q переменных x_1, \dots, x_s , поступающих на ее входы. Матрица M_2 имеет q входов и t выходов. Она позволяет реализовать t элементарных дизъюнкций y_1, \dots, y_t переменных P_1, \dots, P_q ,

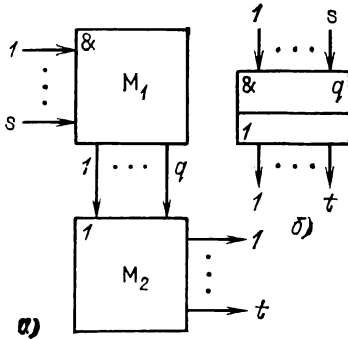


Рис. 3.4

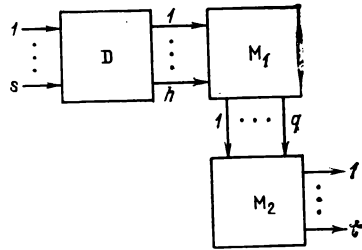


Рис. 3.5

поступающих на ее входы с выходов матрицы M_1 . Выходы матрицы M_1 , соединенные со входами матрицы M_2 , образуют промежуточные шины ПЛМ. ПЛМ с s входами, t выходами и q промежуточными шинами называется ПЛМ (s, t, q) . К выходам матрицы M_2 часто подключается слой программируемых инверторов, который условимся не выделять в отдельный третий уровень. Аналогичные инверторы иногда включаются между матрицами M_1 и M_2 .

На одной ПЛМ (s, t, q) может быть реализована система булевых функций $y_1(x_1, \dots, x_L), \dots, y_N(x_1, \dots, x_L)$, представленная в произвольной ДНФ, такая, что $L \leq s, N \leq t, B \leq q$, где B — число различных элементарных конъюнкций в системе. Внутренняя структура матриц M_1 и M_2 полностью совпадает со структурой аналогичных матриц M_1 и M_2 , рассмотренных в § 3.1. Договоримся в дальнейшем ПЛМ (s, t, q) изображать условно так, как это показано на рис. 3.4,б.

Разновидностью ПЛМ (s, t, q) является их усовершенствованный вариант: ПЛМ (z, q) . В ПЛМ (z, q) фиксируются лишь два параметра: суммарное число входов и выходов $z = s + t$ и число промежуточных шин q . Конкретные значения s и t могут выбираться произвольно при настройке ПЛМ (z, q) . Например, ПЛМ $(6, 10)$ путем соот-

ветствующей настройки может быть использована как ПЛМ (3, 3, 10), ПЛМ (5, 1, 10) и т. д.

В табл. 3.2 (строки 1—10) приведены отдельные параметры некоторых двухуровневых программируемых пользователем ПЛМ комбинационного типа, производимых отечественной и зарубежной промышленностью.

Таблица 3.2

№ п/п	Тип	Число входов	Число выходов	Число промежуточных шин	Число элементов памяти	Задержка, нс	Потребляемая мощность, мВт	Число выводов в корпусе
1	ППЛМ SN74 330 331 [9]	12	6	50	0	30	—	20
2	ППЛМ IM5200 [9]	14	8	48	0	100	90	24
3	МПЛМ MMI 5775A [11]	14	8	96	0	80	—	—
4	МПЛМ MMI 6775/75A [11]	14	8	96	0	80	—	—
5	МПЛМ MMI 6870/71 [11]	14	8	48	0	—	—	—
6	МПЛМ DM 8575/75A [11]	14	8	96	0	—	—	—
7	МПЛМ DM 7576 [11]	14	8	96	0	—	—	—
8	ППЛМ 93458/59 [9]	16	8	48	0	25	—	28
9	ППЛМ 82S100/101 [9]	16	8	48	0	50	600	28
10	ППЛМ 556PT1 [12]	16	8	48	0	50	1000	28
11	ППЛМД μ PB450 [13]*	24	16	72	16	250	—	48
12	ППЛМД ω PB450A [14, 15]	24	16	72	16	100	—	48
13	ППЛМД μ PB450A-1 [14, 15]	24	16	72	16	70	—	48
14	ППЛМ 82S104/105 [9]	16	8	48	6	90	600	28
15	МПЗУ 6275/6276 [7]	11	8	—	—	110	—	—
16	РПЗУ 7810 [7]	11	4	—	—	—	—	—
17	РПЗУ 10443 [7]	8	8	—	—	—	—	—
18	ППЗУ 556PT4 [12]	8	4	—	—	—	—	16
19	ППЗУ 556PT5 [12]	9	8	—	—	80	800	24
20	ППЗУ 556PT6 [12]	11	8	—	—	100	1000	24
21	ППМВ 82 S102/103 [9]	16	9	—	—	35	600	28
22	ППМ 29693 [9]	10	4	3**	—	—	—	20

* Авторы располагают лишь переводом статьи, выполненным во Всесоюзном центре переводов, и все параметры строки 11 взяты из этого перевода.

** Число управляющих входов ПМ.

Трехуровневые ПЛМ комбинационного типа (рис. 3.5), в отличие от двухуровневых, содержат дополнительный s -выходной блок D. Число выходов блока D равно числу h горизонтальных шин в матрице M_1 , и каждый i -й выход

соединен с i -й горизонтальной шиной этой матрицы ($i = \overline{1, h}$). Блок D может иметь самую различную внутреннюю структуру. Например, в [4] он представлен в виде матрицы M_0 , позволяющей образовать h элементарных дизъюнкций переменных x_1, \dots, x_s , поступающих на ее входы и взятых со знаком либо без знака инверсии.

На рис. 3.6 показана матрица M_0 , настроенная на реализацию функций трехвходового дешифратора с низким активным уровнем на его выходах (наличие сигнала индицируется низким уровнем напряжения на соответствующем выходе дешифратора, а отсутствие — высоким). Наиболее часто блок D состоит из набора $s/2$ двухвходовых полных дешифраторов. Договоримся далее трехуровневую ПЛМ с s входами, q промежуточными шинами и t выходами, первый уровень которой образует схема из двухвходовых дешифраторов, называть

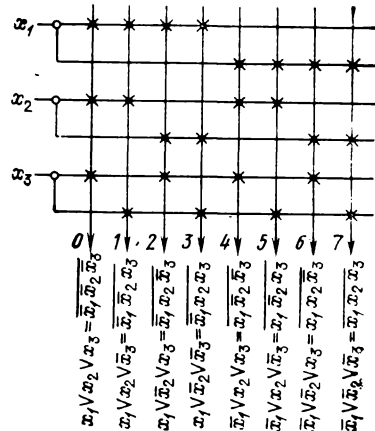


Рис. 3.6

ПЛМД (s, t, q). Для ПЛМД (s, t, q) блок D имеет s входов и $2s$ выходов ($h=2s$), а число горизонтальных шин матрицы M_1 кратно четырем. Входы каждого дешифратора $DC_1, \dots, DC_{s/2}$ являются входами блока D, а четыре выхода каждого j -го из них DC_j соединены с соответствующими четырьмя горизонтальными шинами j -й группы матрицы.

На рис. 3.7 выделена такая j -я группа из четырех входных шин матрицы M_1 , причем для удобства изображения эта матрица развернута на 90° так, что 16 горизонтальных линий являются промежуточными шинами ПЛМД. Таблица в правой части рис. 3.7 указывает функции, реализуемые на промежуточных шинах при соответствующем программировании (настройке) матрицы M_1 ПЛМД. Там же указаны наборы значений переменных x_1 и x_2 на входах дешифратора, при которых на соответствующей промежуточной шине будет сформирован единичный сигнал с высоким активным уровнем. В [4] показано, что использование ПЛМД позволяет по сравнению с ПЛМ сократить число требуемых промежуточных шин для реализации заданной системы

БФ. Использование ПЛМД для построения различных стандартных дискретных устройств будет показано в гл. 4. В качестве примера рассмотрим реализацию на ПЛМД следующей булевой функции:

$$y = \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 x_3 x_4. \quad (3.1)$$

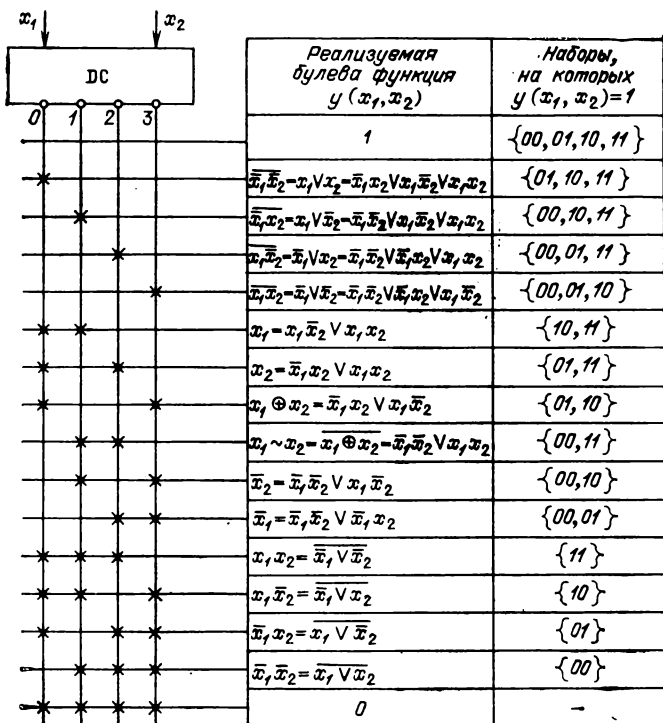


Рис. 3.7

Представим функцию (3.1) в виде $y = \bar{x}_1 \bar{x}_3 (\bar{x}_2 \bar{x}_4 \vee x_2 x_4) \vee x_1 x_3 (\bar{x}_2 \bar{x}_4 \vee x_2 x_4) = (\bar{x}_1 \bar{x}_3 \vee x_1 x_3) (\bar{x}_2 \bar{x}_4 \vee x_2 x_4)$, после чего она может быть реализована на одной ПЛМД (4, 1, 1) так, как это показано на рис. 3.8. Заметим, что ДНФ функции (3.1) не может быть минимизирована, и если $s=4$, $t=1$, то при использовании ПЛМ (s, t, q) необходимо, чтобы $q \geq 4$ и $S(M) = 36$. Из рис. 3.8 легко определить, что суммарная информационная емкость матриц M_1 и M_2 ПЛМД равна 9, что ровно в 4 раза меньше значения аналогичного параметра для ПЛМ. Дополнительное усложнение ПЛМД за счет

введения блока D на практике настолько незначительно, что его можно не учитывать. Таким образом, использование ПЛМД позволяет при такой же, как и у ПЛМ, суммарной информационной емкости матриц M_1 и M_2 реализовать более сложные системы БФ, т. е. можно говорить о том, что площадь матриц M_1 и M_2 используется более рационально. Последующие работы, направленные на повы-

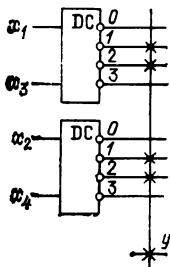


Рис. 3.8

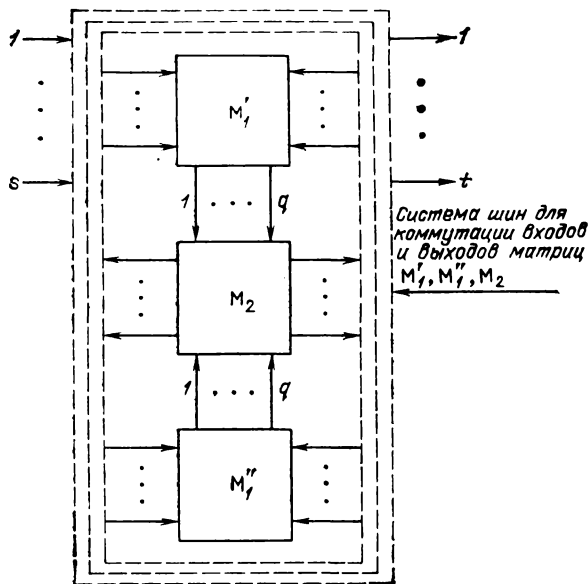


Рис. 3.9

шение эффективности использования площади матриц ПЛМ, привели к созданию структуры, изображенной на рис. 3.9 [5], которая имеет следующие особенности:

1. Матрица M_1 разделена на две части: M'_1 и M''_1 . Матрица M'_1 расположена над матрицей M_2 , а M''_1 — под ней. Это позволяет при необходимости разрезать промежуточные шины в матрице M_2 и реализовать на верхней и нижней частях одной и той же промежуточной шины различные элементарные конъюнкции входных переменных.

2. Входы матриц M'_1 и M''_1 расположены с двух сторон (справа и слева). Любая горизонтальная шина M'_1 (M''_1) разрезается в одном месте, и на одну ее часть подается переменная (или отрицание переменной) с левого входа матрицы, а на другую — с правого.

3. Выходы матрицы M_2 расположены с двух сторон (слева и справа). Любая горизонтальная шина M_2 разрезается в одном месте, и на одной ее части формируется значение функции для левого выхода матрицы, а на другой — для правого.

4. На кристалле БИС ПЛМ предусмотрена специальная система шин, позволяющая соединять выходы одной матрицы со входами другой. Выполнение разрезов шин и органи-

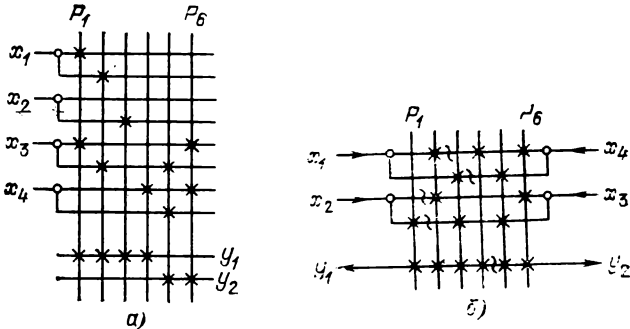


Рис. 3.10

зация необходимых связей между входами и выходами различных матриц осуществляются на этапе настройки ПЛМ на заводе-изготовителе.

Возможность сокращения площади матриц M_1 и M_2 при использовании ПЛМ со структурой, изображенной на рис. 3.9, покажем на примере из [6]. Рассматриваемые ПЛМ, для которых допускается разрезание шин, назовем ПЛМР. Пусть задана следующая система булевых функций:

$$y_1 = x_1 x_3 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 \vee x_4;$$

$$y_2 = \bar{x}_3 \bar{x}_4 \vee x_3 x_4.$$

Настроенная на реализацию этой системы ПЛМ (4, 2, 6) показана на рис. 3.10,а; для нее $S(M) = 60$. Та же самая система функций может быть реализована на ПЛМР так, как это показано на рис. 3.10,б. В ПЛМР на рис. 3.10,б использованы лишь одна верхняя матрица M'_1 и матрица M_2 ; точки разреза в M'_1 и M_2 указаны волнистыми линиями. На промежуточных шинах P_1, \dots, P_6 ПЛМР реализуются соответственно конъюнкции: $\bar{x}_2, x_1 x_3, \bar{x}_1 \bar{x}_3, x_4, \bar{x}_3 \bar{x}_4, x_3 x_4$. Суммарная информационная емкость матриц ПЛМР на рис. 3.10,б равна 30, что в 2 раза меньше значения $S(M)$ для ПЛМ на рис. 3.10,а.

Двухуровневая ПЛМ с памятью (рис. 3.11,а), в отличие от двухуровневой ПЛМ комбинационного типа, содержит внутренний r -разрядный регистр. Входы регистра соединяются с выходными шинами матрицы M_2 , а выходы — с входными шинами матрицы M_1 . Двухуровневую ПЛМ с памятью, имеющую s внешних входов, t внешних выходов, q промежуточных шин и r внутренних элементов памяти, будем называть ПЛМ (s, t, q, r) ; функциональное изображение такой ПЛМ показано на рис. 3.11,б.

Разновидностью ПЛМ (s, t, q, r) являются матричные БИС, у которых включение элементов памяти в обратную связь задается на этапе настройки. По аналогии с термином «программируемый инвертор» здесь целесообразно использовать термин «программируемый элемент памяти» (триггер), поскольку он либо включается в обратную связь с

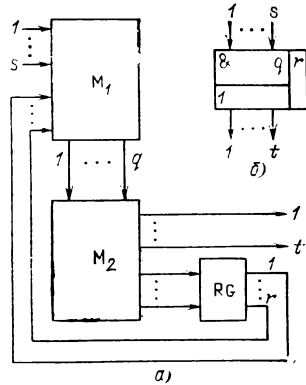


Рис. 3.11

выхода M_2 на вход M_1 , либо нет. В последнем случае выход M_2 будет непосредственно соединен со входом M_1 , в результате на соответствующей ПЛМ может быть реализована система БФ, представленная в форме, отличной от ДНФ.

Трехуровневые ПЛМ с памятью строятся по тому же принципу, что и трехуровневые ПЛМ комбинационного типа. В частности, если блок D трехуровневой ПЛМ (s, t, q, r) состоит из двухвходовых полных дешифраторов, то, как и ранее, будем использовать обозначение ПЛМД (s, t, q, r) .

В табл. 3.2 (строки 11—14) приведены отдельные параметры некоторых ПЛМ с памятью, производимых зарубежной промышленностью. Сокращенные наименования ПЛМ были введены выше, например сокращение ППЛМД обозначает трехуровневую ПЛМ, программируемую пользователем, с блоком D из двухвходовых полных дешифраторов.

3.3. Постоянные запоминающие устройства

Постоянное запоминающее устройство (ПЗУ) может рассматриваться как двухуровневая ПЛМ, матрица M_1 в которой настроена на реализацию функций полного дешифратора (рис. 3.12). Схема полного дешифратора в ПЗУ про-

граммированию не подлежит, поэтому параметр q фиксирован: $q=2^s$. Настройка матрицы M_2 осуществляется так же, как настройка матрицы M_2 в ПЛМ. ПЗУ, имеющее s входов и t выходов, по аналогии с ПЛМ назовем ПЗУ (s, t). На одном ПЗУ (s, t) может быть реализована любая система из N булевых функций ($N \leq t$) L переменных ($L \leq s$). Заметим, что ПЗУ предназначено для реализации совершенных ДНФ булевых функций. Поэтому если заданная система представлена в другой форме, то предварительно она

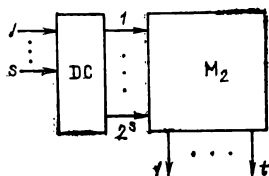


Рис. 3.12

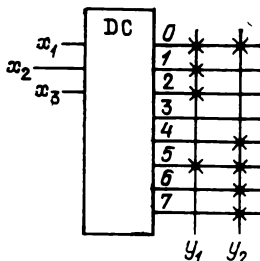


Рис. 3.13

должна быть преобразована к совершенной ДНФ. Например, функции

$$y_1 = \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 x_3;$$

$$y_2 = x_1 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3$$

после преобразования к совершенной ДНФ

$$y_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 x_3;$$

$$y_2 = x_1 \bar{x}_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3$$

могут быть реализованы на ПЗУ (3, 2) так, как это показано на рис. 3.13.

Указанное ранее отличие между ПЛМ и ПЗУ в организации доступа к матрице M_2 на первый взгляд может показаться непринципиальным. Однако если рассмотреть, например, ПЛМ (16, 8, 48) (см. табл. 3.2, строка 10), то ее информационная емкость $S(M) = 1920$. У ПЗУ (16, 8) с таким же числом входов и выходов $S(M) = 2^{16} \times 8 = 524\,288$. Развивающаяся стремительными темпами интегральная технология на сегодняшний день не позволяет создавать на одном кристалле БИС ПЗУ такого объема. В результате стандартные интегральные ПЛМ характеризуются значительно большим числом входов s и выходов t по сравнению с ПЗУ, реализованным на кристалле БИС такого же размера. Другими словами, функциональные свойства ПЛМ ориентированы на реализацию таких систем БФ, в которых

число переменных L и число функций N могут быть довольно большими, однако число термов $B \ll 2^L$. Системы функций, реализуемых на ПЗУ, должны характеризоваться меньшими значениями L и N , при этом ограничение на значение B можно практически не учитывать.

В табл. 3.2 (строки 15—20) приведены параметры некоторых ПЗУ, производимых отечественной и зарубежной промышленностью: масочно-программируемых (МПЗУ), программируемых пользователем (ППЗУ) и репрограммируемых (РПЗУ). Широкая номенклатура ПЗУ, выпускаемых за рубежом, представлена в [7].

3.4. Программируемые матрицы вентиляей

Простейший представитель матричных схем — программируемая матрица вентиляей (ПМВ) — состоит из единственной матрицы M_1 , аналогичной матрице M_1 ПЛИМ. На ее

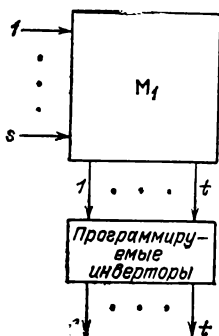


Рис. 3.14

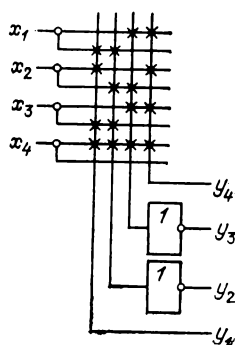


Рис. 3.15

выходах установлены программируемые инверторы, что позволяет на этапе настройки представлять выходные сигналы как в прямом, так и в инверсном виде (рис. 3.14). На любом из выходов программируемых инверторов могут быть реализованы: конъюнкция входных переменных, отрицание этой конъюнкции, дизъюнкция входных переменных, отрицание этой дизъюнкции. Пусть, например, на выходах ПМВ необходимо реализовать следующие функции:

$$y_1 = \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4; \quad y_3 = \bar{x}_1 \vee x_2 \vee \bar{x}_3 \vee \bar{x}_4;$$

$$y_2 = \overline{x_1 x_2 x_3 x_4}; \quad y_4 = \overline{x_1 x_2 x_3 x_4}.$$

Вспользуемся законом двойственности (правилом де-Моргана) и выполним преобразования:

$$y_3 = x_1 \overline{x_2} x_3 x_4; \quad y_4 = x_1 x_2 x_3 x_4.$$

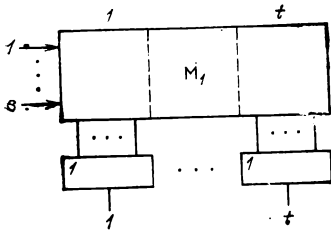


Рис. 3.16

После этого функции y_1, \dots, y_4 могут быть реализованы на одной ПМВ с четырьмя входами и четырьмя выходами так, как это показано на рис. 3.15.

Стандартную ПМВ, имеющую s входов и t выходов, назовем ПМВ (s, t) . На одной ПМВ (s, t) может быть реализовано не более t конъюнкций

(дизъюнкций) L переменных $L \leq s$. Каждая конъюнкция (дизъюнкция) может быть взята со знаком или без знака инверсии.

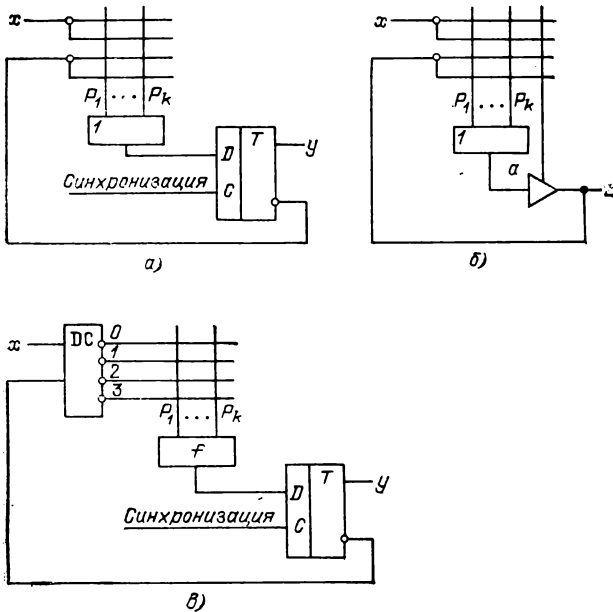


Рис. 3.17

В табл. 3.2 (строка 21) приведены параметры программируемой пользователем ПМВ (16, 9), производимой фирмой Signetics.

3.5. Программируемые матрицы логики

Программируемая матрица логики (ПМЛ) представляет собой матрицу M_1 , ко входам и выходам которой могут быть подключены различные логические и запоминающие элементы. ПМЛ с t выходными шинами может рассматриваться как набор из t ПМВ; к выходам каждой из них подключена логическая схема. На рис. 3.16 изображена про-

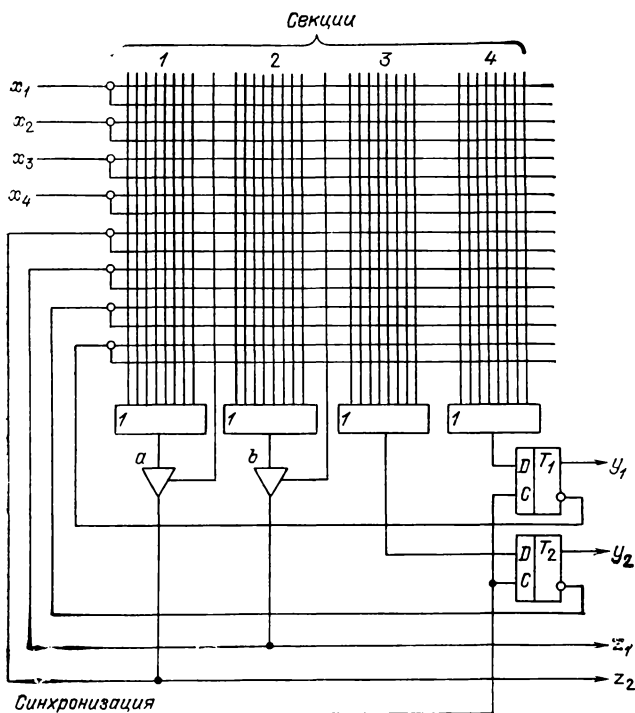


Рис. 3.18

стейшая ПМЛ, в которой матрица M_1 штриховыми линиями разделена на t секций. Выходы каждой секции подключены ко входам элемента ИЛИ, число которых равно t . К выходам секций матрицы M_1 могут подключаться более сложные логические схемы (их типовые структуры изображены на рис. 3.17). На рис. 3.17,а выходы i -й секции P_1, \dots, P_k соединены со входами элемента ИЛИ, выход которого подключен к D входу D -триггера. Сигнал с единичного плеча D -триггера передается на выход ПМЛ, а с нулевого — на

вход матрицы M_1 . Вход x является внешним входом матрицы M_1 . На рис. 3.17,б выход элемента ИЛИ соединен со входом управляемого элемента a , с помощью которого шина z может быть использована либо как выход ПМЛ, либо как вход ПМЛ. Управление элементом a осуществляется сигналами с выхода i -й секции. На рис. 3.17,в выходы i -й секции подключены ко входам логического элемента f , реализующего некоторую функцию (в ряде случаев довольно сложную). Сигналы с нулевого плеча D -триггера и с внешнего входа x поступают на входы дешифратора ДС, принцип использования которого сохраняется таким же, как и в ПЛМД. Схема на рис. 3.17,в предназначена для использования в арифметических устройствах [8].

На рис. 3.18 приведена структура ПМЛ [9], содержащей четыре секции. Выходы матриц в первых двух секциях подключены ко входам схемы, изображенной на рис. 3.17,б, а в последних двух — ко входам схемы, изображенной на рис. 3.17,а. Входы x_1, \dots, x_4 являются внешними входами ПМЛ, выходы y_1, y_2 — внешними выходами ПМЛ. Шины z_1 и z_2 в зависимости от сигналов управления, поданных на элементы a и b , могут быть использованы либо как вход-

Таблица 3.3

Тип	Число входов	Число выходов	N_1	N_2	Функция схемы на выходе секции
PAL 10H8	10	8	—	—	И—ИЛИ
PAL 12H6	12	6	—	—	И—ИЛИ
PAL 14H4	14	4	—	—	И—ИЛИ
PAL 16H2	16	2	—	—	И—ИЛИ
PAL 10L8	10	8	—	—	И—ИЛИ—НЕ
PAL 12L6	12	6	—	—	И—ИЛИ—НЕ
PAL 14L4	14	4	—	—	И—ИЛИ—НЕ
PAL 16L2	16	2	—	—	И—ИЛИ—НЕ
PAL 16C1	16	2	—	—	И—ИЛИ
					И—ИЛИ—НЕ
PAL 16L8	10	8	6	—	И—ИЛИ—НЕ
PAL 16R8	8	8	—	8	И—ИЛИ—НЕ
PAL 16R6	8	8	2	6	И—ИЛИ—НЕ
PAL 16R4	8	8	4	4	И—ИЛИ—НЕ
PAL 16X4	8	8	4	4	Сложная функция
PAL 16A4	8	8	4	4	То же

Примечание. N_1 и N_2 — число секций, выходы которых связаны со входами схем, изображенных на рис. 3.17, б и 3.17, а, в соответственно.

ные, либо как выходные. Запись информации в триггеры T_1 и T_2 на рис. 3.18 синхронизируется специальным сигналом, формируемым в виде ПМЛ.

В таблице 3.3 приведены данные о некоторых программируемых пользователем ПМЛ, производимых фирмой Monolithic Memories Inc. [8, 9]. Все они имеют потребляемую мощность 300 мВт, задержку 25 нс и 20 выводов.

3.6. Программируемые мультиплексоры

Программируемый мультиплексор (ПМ) состоит из t мультиплексоров W_1, \dots, W_t , имеющих общие управляющие входы $1, \dots, n$ (рис. 3.19). Каждый i -й мультиплексор

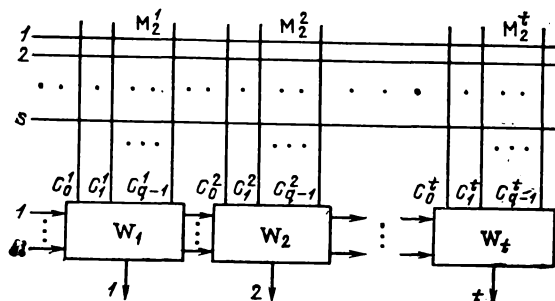


Рис. 3.19

W_i имеет свои информационные входы C_0^i, \dots, C_{q-1}^i , где $q = 2^n$. При подаче на управляющие входы i -го мультиплексора двоичного кода с десятичным эквивалентом j сигнал со входа C_j^i i -го мультиплексора (и только этот сигнал) поступает на его выход. Информационные входы C_0^i, \dots, C_{q-1}^i i -го мультиплексора W_i соединены с выходами матрицы M_2^i , имеющей ту же структуру, что и матрица M_2 в ПЛМ. Каждая матрица M_2^i ($i = \overline{1, t}$) имеет s входных и q выходных шин, причем одноименные входные шины всех матриц M_2^1, \dots, M_2^t объединены. Матрица M_2^i ($i = \overline{1, t}$) позволяет реализовать q элементарных дизъюнкций на выходах C_0^i, \dots, C_{q-1}^i переменных x_1, \dots, x_s , поступающих на ее входы. Формирование элементарных дизъюнкций осуществляется путем настройки матриц M_2^1, \dots, M_2^t по аналогии с настройкой матрицы M_2 ПЛМ. Структура матриц M_2^i ($i = \overline{1, t}$), вообще говоря, может быть различной. Например,

в [10] рассматриваются ПМ, для которых матрицы M_2 полностью подобны матрицам M_1 ПЛМ.

Стандартный ПМ, имеющий s информационных внешних входов, t внешних выходов и n управляющих внешних входов, назовем ПМ (s, t, n) .

В табл. 3.2 (строка 22) приведены параметры программируемого пользователем ПМ $(10, 4, 3)$, производимого фирмой Raytheon.

ГЛАВА ЧЕТВЕРТАЯ

СТАНДАРТНЫЕ ЦИФРОВЫЕ СХЕМЫ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ УСТРОЙСТВАХ

4.1. Шифраторы и дешифраторы

Шифраторы и дешифраторы широко используются в цифровых устройствах для преобразования двоичного кода в унитарный и унитарного в двоичный. Напомним, что в унитарном L -разрядном коде x_0, \dots, x_{L-1} лишь одна переменная x_l может принимать значение 1, а все оставшиеся — 0. Если $x_l = 1, x_0 = \dots = x_{l-1} = x_{l+1} = \dots = x_{L-1} = 0$, то код x_0, \dots, x_{L-1} представляет число, десятичный эквивалент которого равен l . Шифратором назовем комбинационную схему, реализующую преобразование унитарного кода x_0, \dots, x_{L-1} в двоичный код y_1, \dots, y_N . При $L=10, N=4$ функции шифратора запишутся в следующем виде:

$$y_1 = x_8 \vee x_9; \quad y_2 = x_4 \vee x_5 \vee x_6 \vee x_7; \quad (4.1)$$

$$y_3 = x_2 \vee x_3 \vee x_6 \vee x_7; \quad y_4 = x_1 \vee x_3 \vee x_5 \vee x_7 \vee x_9.$$

Здесь переменная y_1 представляет значение старшего разряда двоичного числа, а y_4 — младшего. Используем правило де-Моргана и преобразуем функции (4.1) к виду:

$$y_1 = \overline{x_8 x_9}; \quad y_2 = \overline{x_4 x_5 x_6 x_7}; \quad (4.2)$$

$$y_3 = \overline{x_2 x_3 x_6 x_7}; \quad y_4 = \overline{x_1 x_3 x_5 x_7 x_9}.$$

Шифраторы, характеризуемые другими значениями параметров L и N , описываются аналогичными функциями, поэтому ограничимся реализацией конкретного рассматриваемого шифратора на программируемых логических уст-

роЙствах (ПЛУ). Функции (4.2) могут быть непосредственно реализованы на программируемой матрице вентилей ПМВ (9, 4) так, как это показано на рис. 4.1. На рис. 4.2 показана реализация функций (4.1) на программируемой матрице логики. Два варианта построения шифратора на программируемой логической матрице рассмотрены на рис. 4.3: ПЛМ(9, 4, 9) на рис. 4.3,а реализует функции (4.1), а ПЛМ(9, 4, 4) на рис. 4.3,б — функции (4.2). Во

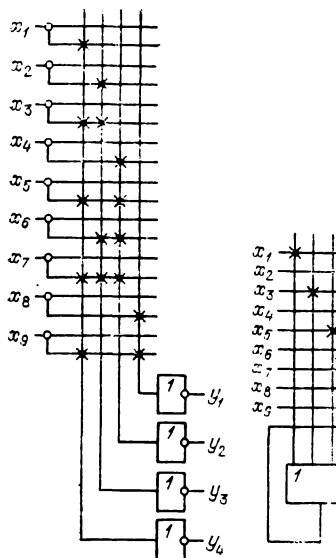


Рис. 4.1

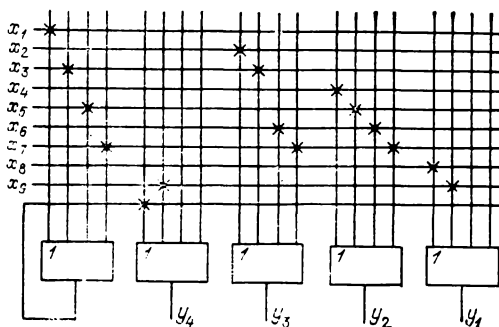


Рис. 4.2

втором случае используются программируемые инверторы на выходах ПЛМ. Слой программируемых инверторов на выходах имеется в большинстве ПЛМ, производимых в нашей стране и за рубежом (например, в отечественных ПЛМ 556РТ1). Если в используемых ПЛМ такие инверторы отсутствуют, то функции рассматриваемого шифратора могут быть записаны в следующем виде:

$$\begin{aligned}
 y_1 &= \overline{x_0 \vee x_1 \vee x_2 \vee x_3 \vee x_4 \vee x_5 \vee x_6 \vee x_7}; \\
 y_2 &= \overline{x_0 \vee x_1 \vee x_2 \vee x_3 \vee x_8 \vee x_9}; \\
 y_3 &= \overline{x_0 \vee x_1 \vee x_4 \vee x_5 \vee x_8 \vee x_9}; \\
 y_4 &= \overline{x_0 \vee x_2 \vee x_4 \vee x_6 \vee x_8};
 \end{aligned}$$

или после преобразования по правилу де-Моргана:

$$y_1 = \overline{x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7};$$

$$y_2 = \overline{x_0 x_1 x_2 x_3 x_8 x_9};$$

$$y_3 = \overline{x_0 x_1 x_4 x_5 x_8 x_9};$$

$$y_4 = \overline{x_0 x_2 x_4 x_8 x_9}.$$

(4.3)

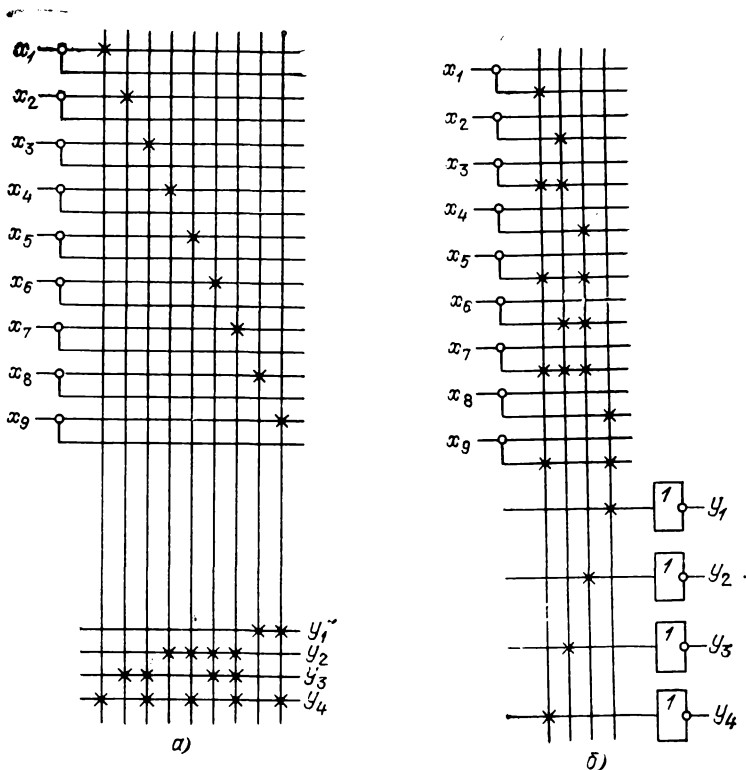


Рис. 4.3

Реализация функций (4.3) на ПЛМД(10, 4, 4) показана на рис. 4.4. Для настройки ПЛМД(10, 4, 4) использовался рис. 3.7.

На рис. 4.5 показана реализация функций (4.1) на программируемом мультиплексоре ПМ (9, 4, 2). При $v_1 = v_2 = 0$ ПМ на рис. 4.5 выполняет функции шифратора, а при $v_1 \vee v_2 = 1$ этот же ПМ может выполнять любые другие функции. Таким образом, при необходимости можно ис-

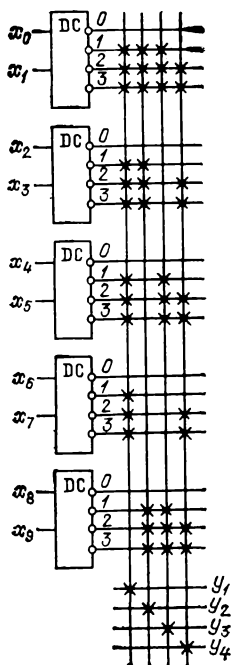


Рис. 4.4

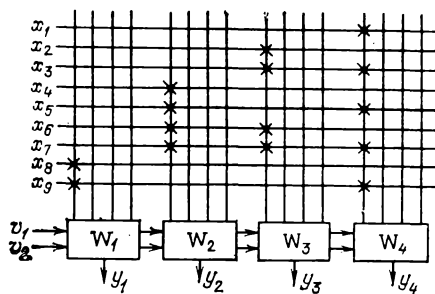


Рис. 4.5

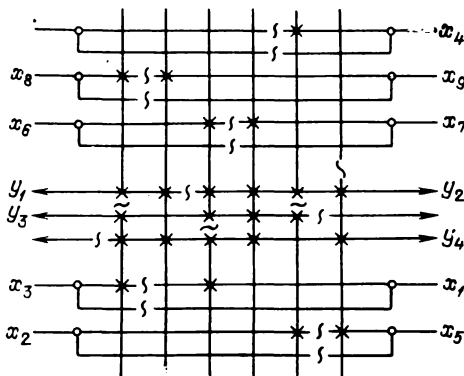


Рис. 4.6

пользовать не только внутреннюю, но и внешнюю настройку ПЛУ.

На рис. 4.6 показана одна из возможных реализаций функций (4.1) на ПЛМР. Сравнивая различные варианты построения шифратора на рис. 4.1—4.6, можно сделать вывод, что наименьшую суммарную информационную емкость имеют схемы, изображенные на рис. 4.1 (72 бит) и рис. 4.6 (78 бит).

Иногда возникает необходимость построения схем, преобразующих унитарный код в двоичный, причем если во входном коде возникла ошибка, выводящая его из класса унитарных кодов (например, появились две единицы или более), то на всех выходах рассматриваемой схемы устанавливаются нулевые сигналы. Булевы функции, описывающие функционирование такой схемы при $L=10$, $N=4$, заданы в табл. 4.1. На рис. 4.7 показана реализация этих функций на программируемой матрице логики.

Таблица 4.1

x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	y_1	y_2	y_3	y_4
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

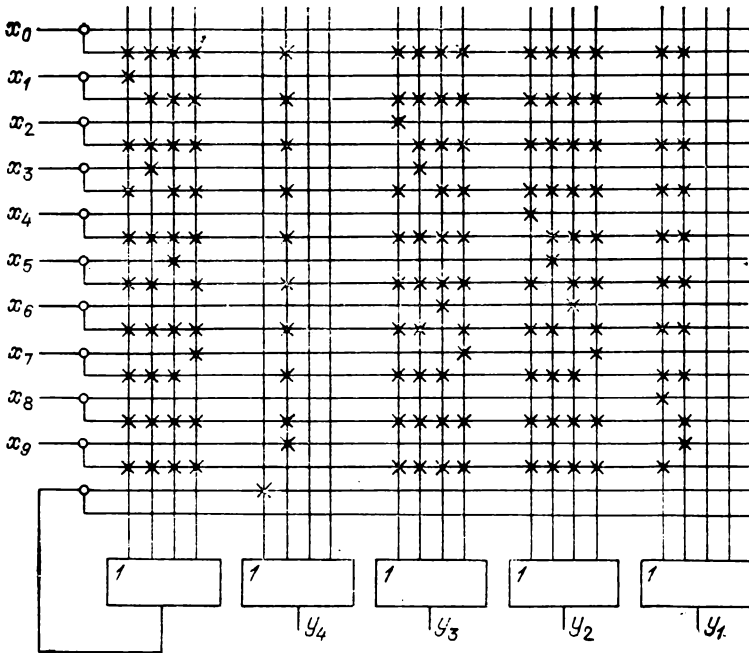


Рис. 4.7

Дешифратор выполняет функции, обратные шифратору, т. е. преобразует двоичный код в унитарный. Принципы построения дешифраторов с различным числом входов

..., v_s двоичного кода с десятичным эквивалентом l сигнал с информационного входа x_l (и только этот сигнал) передается на выход y . Такие схемы, которые принято называть мультиплексорами, находят широкое применение в самых разнообразных цифровых устройствах. Ограничимся рассмотрением принципов построения на ПЛУ конкретного мультиплексора, у которого $L=8$ и $s=3$. Функция y такого мультиплексора будет записана в следующем виде:

$$y = \bar{v}_1 \bar{v}_2 \bar{v}_3 x_0 \vee \bar{v}_1 \bar{v}_2 v_3 x_1 \vee \bar{v}_1 v_2 \bar{v}_3 x_2 \vee \bar{v}_1 v_2 v_3 x_3 \vee v_1 \bar{v}_2 \bar{v}_3 x_4 \vee v_1 \bar{v}_2 v_3 x_5 \vee v_1 v_2 \bar{v}_3 x_6 \vee v_1 v_2 v_3 x_7. \quad (4.4)$$

На рис. 4.9 показана реализация функций (4.4) на ПЛМ (11, 1, 8). Поскольку мультиплексоры относятся к

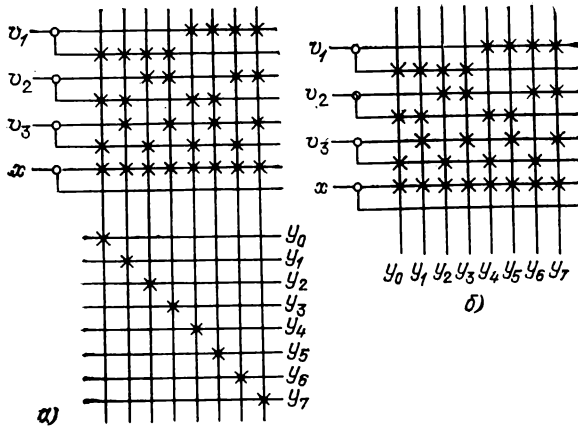


Рис. 4.10

схемам, имеющим большое число входов и единственный выход, для их построения целесообразно использовать ПЛМ (z, q), где $z=s+t$, перераспределяя значения s и t таким образом, чтобы $t=1$, а $s=z-1$.

Демультимплексор выполняет преобразования, обратные мультиплексору, и имеет единственный информационный вход x , управляющие входы v_1, \dots, v_s и выходы y_0, \dots, y_{N-1} (обычно $N=2^s$). При подаче на управляющие входы v_1, \dots, v_s двоичного кода с десятичным эквивалентом n сигнал со входа x поступает на выход y_n (и только на этот выход). Одновременно на всех остальных выходах $y_0, \dots, y_{n-1}, y_{n+1}, \dots, y_{N-1}$ устанавливаются нулевые сигналы. При $s=3$, $N=8$ поведение демультимплексора описывается следующи-

ми булевыми функциями:

$$\begin{aligned}
 y_0 &= \bar{v}_1 \bar{v}_2 \bar{v}_3 x; & y_1 &= \bar{v}_1 \bar{v}_2 v_3 x; \\
 y_2 &= \bar{v}_1 v_2 \bar{v}_3 x; & y_3 &= \bar{v}_1 v_2 v_3 x; \\
 y_4 &= v_1 \bar{v}_2 \bar{v}_3 x; & y_5 &= v_1 \bar{v}_2 v_3 x; \\
 y_6 &= v_1 v_2 \bar{v}_3 x; & y_7 &= v_1 v_2 v_3 x.
 \end{aligned}
 \tag{4.5}$$

На рис. 4.10,а показана реализация функций (4.5) на ПЛМ (4, 8, 8), а на рис. 4.10,б — на ПМВ (4, 8). С точки зрения суммарной информационной емкости матриц схема на ПМВ является более эффективной.

4.3. Схемы сравнения и контроля

Рассмотрим два целых двоичных числа A и B , и пусть $y(A=B)$, $y(A>B)$ и $y(A<B)$ — некоторые полностью определенные булевы функции, такие, что:

$$y(A=B) = \begin{cases} 1, & \text{если } A=B; \\ 0, & \text{если } A \neq B; \end{cases}$$

$$y(A>B) = \begin{cases} 1, & \text{если } A>B; \\ 0, & \text{если } A \leq B; \end{cases}$$

$$y(A<B) = \begin{cases} 1, & \text{если } A<B; \\ 0, & \text{если } A \geq B. \end{cases}$$

Схемой сравнения назовем схему, реализующую функции $y(A=B)$, $y(A>B)$, $y(A<B)$.

Пусть значения разрядов чисел A и B представляются соответственно значениями переменных a_1, \dots, a_L и b_1, \dots, b_L . При $L=2$ булевы функции $y(A>B)$, $y(A<B)$, $y(A=B)$ могут быть заданы табл. 4.2. ДНФ этих функций представляется в виде:

Таблица 4.2

a_1	a_2	b_1	b_2	$y(A>B)$	$y(A<B)$	$y(A=B)$
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

$$\begin{aligned}
 y(A>B) &= \bar{a}_1 \bar{a}_2 \bar{b}_1 \bar{b}_2 \vee \bar{a}_1 \bar{a}_2 \bar{b}_1 b_2 \vee \bar{a}_1 \bar{a}_2 b_1 b_2 \vee \bar{a}_1 a_2 \bar{b}_1 \bar{b}_2 \vee \\
 &\vee \bar{a}_1 a_2 \bar{b}_1 b_2 \vee \bar{a}_1 a_2 b_1 \bar{b}_2; \\
 y(A<B) &= \bar{a}_1 \bar{a}_2 \bar{b}_1 b_2 \vee \bar{a}_1 \bar{a}_2 b_1 \bar{b}_2 \vee \bar{a}_1 a_2 b_1 b_2 \vee \bar{a}_1 a_2 b_1 \bar{b}_2 \vee \\
 &\vee \bar{a}_1 a_2 b_1 b_2 \vee \bar{a}_1 \bar{a}_2 b_1 b_2; \\
 y(A=B) &= \bar{a}_1 \bar{a}_2 \bar{b}_1 \bar{b}_2 \vee \bar{a}_1 \bar{a}_2 \bar{b}_1 b_2 \vee \bar{a}_1 \bar{a}_2 b_1 \bar{b}_2 \vee \bar{a}_1 a_2 b_1 b_2.
 \end{aligned}$$

На рис. 4.11, *а* и *б* показаны карты Карно для минимизации функций $y(A > B)$ и $y(A < B)$. ДНФ функции $y(A = B)$ не может быть минимизирована. В результате минимизации получим:

$$y(A > B) = a_1 \bar{b}_1 \vee a_2 \bar{b}_1 \bar{b}_2 \vee a_1 a_2 \bar{b}_2 = a_1 \bar{b}_1 \vee a_2 \bar{b}_2 (\bar{b}_1 \vee a_1);$$

$$y(A < B) = \bar{a}_1 b_1 \vee \bar{a}_1 \bar{a}_2 b_2 \vee \bar{a}_2 b_1 b_2 = \bar{a}_1 b_1 \vee \bar{a}_2 b_2 (\bar{a}_1 \vee b_1).$$

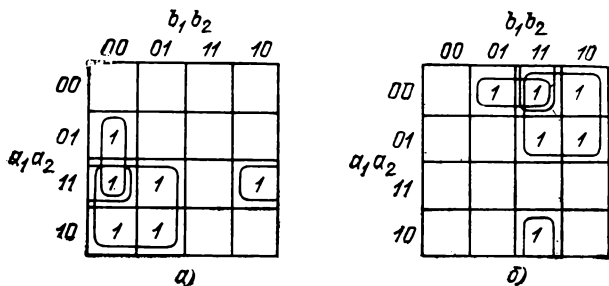


Рис. 4.11

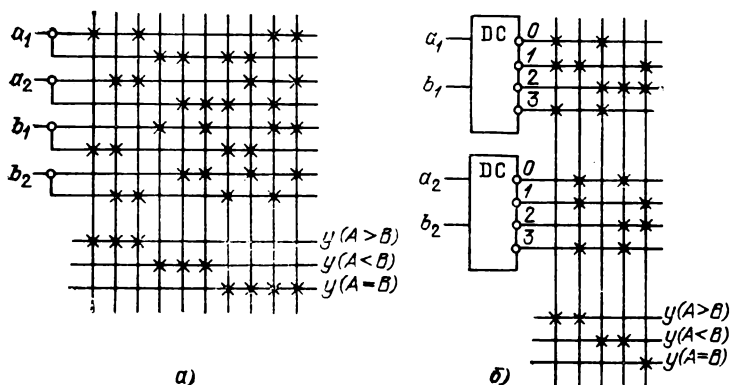


Рис. 4.12

На рис. 4.12, *а* показана реализация функций $y(A > B)$, $y(A < B)$, $y(A = B)$ на ПЛМ(4, 3, 10), а на рис. 4.12, *б* — на ПЛМД(4, 3, 5). Для реализации на ПЛМД(4, 3, 5) функция $y(A = B)$ представлена в виде

$$y(A = B) = \bar{a}_1 \bar{b}_1 (\bar{a}_2 \bar{b}_2 \vee a_2 b_2) \vee a_1 b_1 (\bar{a}_2 \bar{b}_2 \vee a_2 b_2) =$$

$$= (\bar{a}_1 \bar{b}_1 \vee a_1 b_1) (\bar{a}_2 \bar{b}_2 \vee a_2 b_2) = (a_1 \sim b_1) (a_2 \sim b_2).$$

Соотношение

$$y(A > B) = \overline{y(A < B) \vee y(A = B)} = \overline{y(A < B) \cdot y(A = B)}$$

позволяет сократить число используемых промежуточных шин ПЛМ и ПЛМД. Соответствующие реализации функций $y(A > B)$, $y(A < B)$ и $y(A = B)$ на ПЛМ(6, 3, 8) и ПЛМД(6, 3, 4) показаны на рис. 4.13, а и б. Таким же способом могут быть построены схемы для реализации функций:

$$y(A \geq B) = y(A > B) \vee y(A = B);$$

$$y(A \leq B) = y(A < B) \vee y(A = B).$$

На рис. 4.14 показана непосредственная реализация функций $y(A > B)$, $y(A < B)$, $y(A = B)$ из табл. 4.2 на ПЗУ(4, 3).

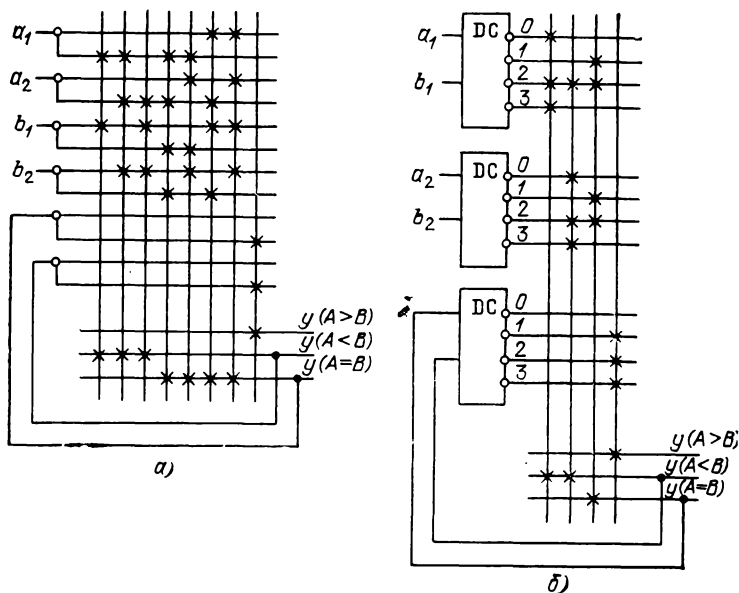


Рис. 4.13

Если $L > 2$, то функции $y(A > B)$, $y(A < B)$, $y(A = B)$ могут быть представлены в следующем общем виде:

$$y(A > B) = \overline{y(A < B)} \cdot \overline{y(A = B)};$$

$$y(A < B) = \bar{a}_1 b_1 \vee \bar{a}_2 b_2 (\bar{a}_1 \vee b_1) \vee \dots \vee \bar{a}_L b_L (\bar{a}_{L-1} \vee b_{L-1}) \dots$$

$$\dots (\bar{a}_1 \vee b_1);$$

$$y(A = B) = (a_1 \sim b_1) \cdot \dots \cdot (a_L \sim b_L).$$

Например, при $L=4$:

$$\begin{aligned}
 y(A > B) &= \overline{y(A < B)} \cdot \overline{y(A = B)}; \\
 y(A < B) &= \bar{a}_1 b_1 \vee \bar{a}_2 b_2 (\bar{a}_1 \vee b_1) \vee \bar{a}_3 b_3 (\bar{a}_2 \vee b_2) (\bar{a}_1 \vee b_1) \vee \\
 &\quad \vee \bar{a}_4 b_4 (\bar{a}_3 \vee b_3) (\bar{a}_2 \vee b_2) (\bar{a}_1 \vee b_1); \\
 y(A = B) &= (a_1 \sim b_1) (a_2 \sim b_2) (a_3 \sim b_3) (a_4 \sim b_4).
 \end{aligned}
 \tag{4.6}$$

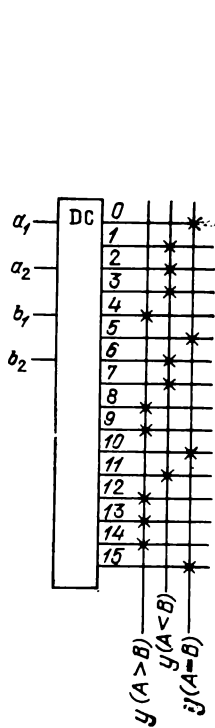


Рис. 4.14

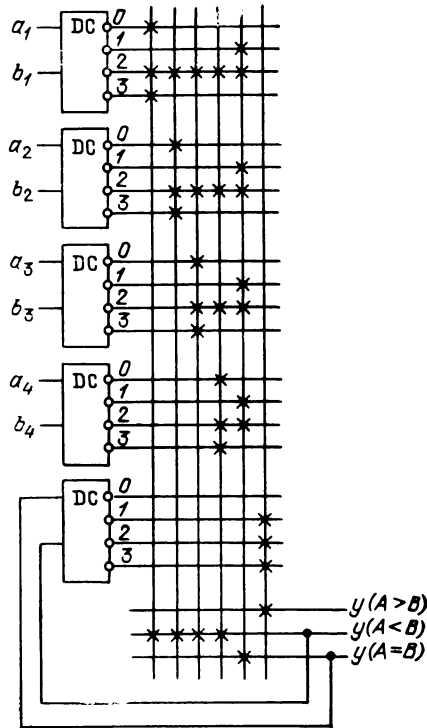


Рис. 4.15

Если реализовать функции (4.6) на ПЛМ, то $q=32$, если же реализовать их на ПЛМД, то $q=6$. Схема сравнения четырехразрядных двоичных чисел A и B на ПЛМД (10, 3, 6) показана на рис. 4.15. Суммарная информационная емкость матриц этой схемы (равная 138 бит) приблизительно в 5,3 раза меньше суммарной информационной емкости матриц (равной 736 бит) аналогичной схемы на ПЛМ (10, 3, 32) и приблизительно в 5,6 раза мень-

ше суммарной информационной емкости матриц аналогичной схемы на ПЗУ(8, 3) (равной 768 бит).

Рассмотрим L -разрядное двоичное число A и одноразрядное двоичное число B . Пусть $y(A)$ и $y(A, B)$ — некоторые полностью определенные БФ, такие, что $y(A) = 1$, если количество единиц в записи числа A — четное, и $y(A) = 0$ в противном случае; $y(A, B) = 1$, если суммарное количество единиц в записи чисел A и B — нечетное, и $y(A, B) = 0$ в противном случае. Схемы, реализующие функции $y(A)$ и $y(A, B)$, широко используются в схемах контроля цифровой аппаратуры. Пусть a_1, \dots, a_L — переменные, представляющие значения разрядов двоичного числа A ; b — переменная, представляющая значение единственного разряда двоичного числа B . Выберем для определенности $L = 8$. Тогда если

$$\begin{array}{lll} f_1 = a_1 \sim a_2; & f_3 = a_5 \sim a_6; & f_5 = f_1 \sim f_2; \\ f_2 = a_3 \sim a_4; & f_4 = a_7 \sim a_8; & f_6 = f_3 \sim f_4, \end{array}$$

то

$$y(A) = f_5 \sim f_6; \quad y(A, B) = y(A) \sim b.$$

Выполним необходимые подстановки:

$$\begin{aligned} y(A) &= f_5 \sim f_6 = (f_1 \sim f_2) \sim (f_3 \sim f_4) = (f_1 \sim f_2)(f_3 \sim f_4) \vee \\ &\vee (f_1 \oplus f_2)(f_3 \oplus f_4) = \{[(a_1 \sim a_2) \sim (a_3 \sim a_4)] [(a_5 \sim a_6) \sim \\ &\sim (a_7 \sim a_8)]\} \vee \{(a_1 \sim a_2) \oplus (a_3 \sim a_4) [(a_5 \sim a_6) \oplus (a_7 \sim a_8)]\} = \\ &= \{(a_1 \sim a_2)(a_3 \sim a_4) \vee (a_1 \oplus a_2)(a_3 \oplus a_4) [(a_5 \sim a_6)(a_7 \sim a_8) \vee \\ &\vee (a_5 \oplus a_6)(a_7 \oplus a_8)]\} \vee \{(a_1 \sim a_2)(a_3 \oplus a_4) \vee (a_1 \oplus a_2)(a_3 \sim a_4)\} \wedge \\ &\wedge \{(a_5 \sim a_6)(a_7 \oplus a_8) \vee (a_5 \oplus a_6)(a_7 \sim a_8)\} = (a_1 \sim a_2)(a_3 \sim \\ &\sim a_4)(a_5 \sim a_6)(a_7 \sim a_8) \vee (a_1 \sim a_2)(a_3 \sim a_4)(a_5 \oplus a_6)(a_7 \oplus a_8) \vee \\ &\vee (a_1 \oplus a_2)(a_3 \oplus a_4)(a_5 \sim a_6)(a_7 \sim a_8) \vee (a_1 \oplus a_2)(a_3 \oplus \\ &\oplus a_4)(a_5 \oplus a_6)(a_7 \oplus a_8) \vee (a_1 \sim a_2)(a_3 \oplus a_4)(a_5 \sim a_6)(a_7 \oplus a_8) \vee \\ &\vee (a_1 \sim a_2)(a_3 \oplus a_4)(a_5 \oplus a_6)(a_7 \sim a_8) \vee (a_1 \oplus a_2)(a_3 \sim a_4)(a_5 \sim \\ &\sim a_6)(a_7 \oplus a_8) \vee (a_1 \oplus a_2)(a_3 \sim a_4)(a_5 \oplus a_6)(a_7 \sim a_8). \end{aligned}$$

На рис. 4.16 показана реализация функций $y(A)$ и $y(A, B)$ на ПЛМД(10, 2, 9). Суммарная информационная емкость полученной схемы, равная 198 бит, более чем в 5 раз меньше суммарной информационной емкости аналогичной схемы на ПЗУ(9, 2) (равной 1024 бит). Используемая площадь матриц на рис. 4.16 может быть дополни-

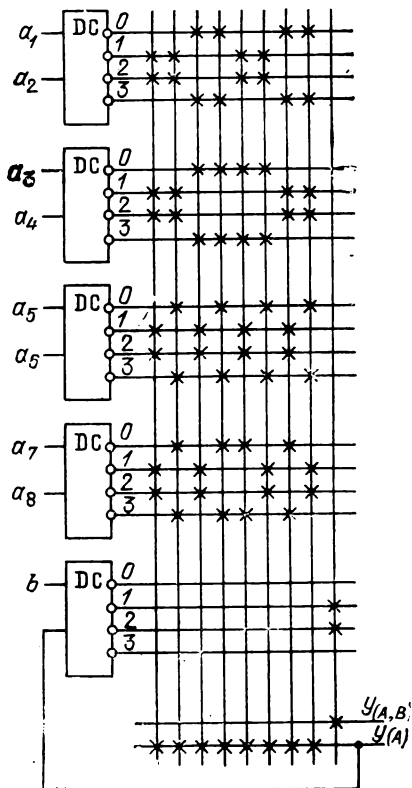


Рис. 4.16

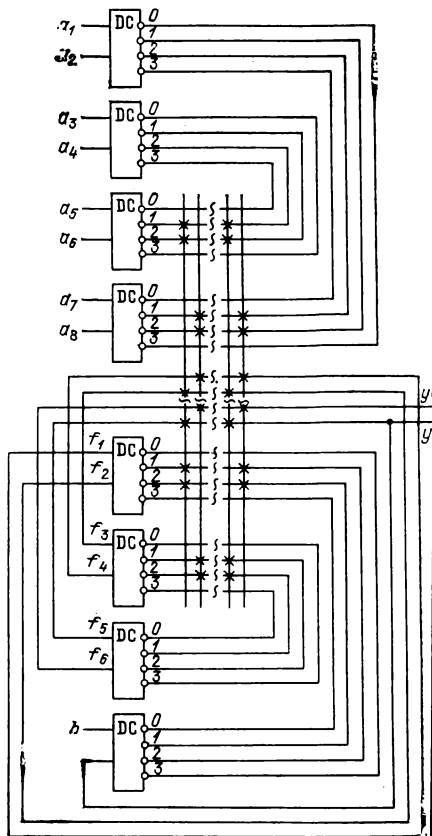


Рис. 4.17

тельно сокращена при применении принципа разрезания шин в ПЛМД (см. § 3.2); соответствующие ПЛМД назовем ПЛМДР. Реализация функций $y(A)$, $y(A, B)$ на ПЛМДР показана на рис. 4.17. В этом случае суммарная информационная емкость матриц равна всего 80 бит.

4.4. Схемы для арифметико-логических устройств

Пусть необходимо выполнить операцию $S=A+B$, где A и B — двоичные L -разрядные числа, разряды которых представляются соответственно значениями переменных a_1, \dots, a_L и b_1, \dots, b_L . Разряды двоичного числа S , равного

сумме чисел A и B , представляются значениями переменных s_1, \dots, s_N . Прежде чем изложить способы реализации сложения на ПЛУ, рассмотрим табл. 4.3, задающую функцию одноразрядного двоичного сумматора. В этой таблице a_l и b_l — одноразрядные слагаемые; c_{l-1} — перенос из предыдущего $(l-1)$ -го разряда сумматора; c_l — перенос в последующий $(l+1)$ -й разряд сумматора ($l = \overline{1, L}$); s_l — сумма слагаемых a_l и b_l с учетом переноса c_{l-1} из предыдущего $(l-1)$ -го разряда. При $l=1$ функции s_1, c_1 и \bar{c}_1 (функция \bar{c}_1 потребуется в дальнейшем) могут быть записаны в следующем виде:

Таблица 4.3

c_{l-1}	a_l	b_l	s_l	c_l
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

следующем виде:

$$s_1 = (a_1 \sim b_1) c_0 \vee (a_1 \oplus b_1) \bar{c}_0;$$

$$c_1 = a_1 b_1 \vee (a_1 \vee b_1) c_0;$$

$$\bar{c}_1 = \bar{a}_1 \bar{b}_1 \vee (\bar{a}_1 \vee \bar{b}_1) \bar{c}_0.$$

Аналогично для произвольного l ($l \in \{1, \dots, L\}$):

$$s_l = (a_l \sim b_l) c_{l-1} \vee (a_l \oplus b_l) \bar{c}_{l-1};$$

$$c_l = a_l b_l \vee (a_l \vee b_l) c_{l-1};$$

$$\bar{c}_l = \bar{a}_l \bar{b}_l \vee (\bar{a}_l \vee \bar{b}_l) \bar{c}_{l-1}. \quad (4.7)$$

Путем несложных преобразований из множества аргументов любой функции (4.7) можно исключить все переменные c_1, \dots, c_L . Например, при $L=3$:

$$s_2 = (a_2 \sim b_2) a_1 b_1 \vee (a_2 \sim b_2) (a_1 \vee b_1) c_0 \vee$$

$$\vee (a_2 \oplus b_2) \bar{a}_1 \bar{b}_1 \vee (a_2 \oplus b_2) (\bar{a}_1 \vee \bar{b}_1) \bar{c}_0;$$

$$c_2 = a_2 b_2 \vee (a_2 \vee b_2) a_1 b_1 \vee (a_2 \vee b_2) (a_1 \vee b_1) c_0;$$

$$\bar{c}_2 = \bar{a}_2 \bar{b}_2 \vee (\bar{a}_2 \vee \bar{b}_2) \bar{a}_1 \bar{b}_1 \vee (\bar{a}_2 \vee \bar{b}_2) (\bar{a}_1 \vee \bar{b}_1) \bar{c}_0;$$

$$s_3 = (a_3 \sim b_3) a_2 b_2 \vee (a_3 \sim b_3) (a_2 \vee b_2) a_1 b_1 \vee$$

$$\vee (a_3 \sim b_3) (a_2 \vee b_2) (a_1 \vee b_1) c_0 \vee (a_3 \oplus b_3) \bar{a}_2 \bar{b}_2 \vee$$

$$\vee (a_3 \oplus b_3) (\bar{a}_2 \vee \bar{b}_2) \bar{a}_1 \bar{b}_1 \vee (a_3 \oplus b_3) (\bar{a}_2 \vee \bar{b}_2) (\bar{a}_1 \vee \bar{b}_1) \bar{c}_0;$$

$$c_3 = a_3 b_3 \vee (a_3 \vee b_3) a_2 b_2 \vee (a_3 \vee b_3) (a_2 \vee b_2) a_1 b_1 \vee$$

$$\vee (a_3 \vee b_3) (a_2 \vee b_2) (a_1 \vee b_1) c_0;$$

$$\bar{c}_3 = \bar{a}_3 \bar{b}_3 \vee (\bar{a}_3 \vee \bar{b}_3) \bar{a}_2 \bar{b}_2 \vee (\bar{a}_3 \vee \bar{b}_3) (\bar{a}_2 \vee \bar{b}_2) \bar{a}_1 \bar{b}_1 \vee$$

$$\vee (\bar{a}_3 \vee \bar{b}_3) (\bar{a}_2 \vee \bar{b}_2) (\bar{a}_1 \vee \bar{b}_1) \bar{c}_0.$$

Тогда аргументами функций (4.7) будут лишь переменные $a_1, \dots, a_L, b_1, \dots, b_L, c_0$, и эти функции просто реализуются на ПЛМ или ПЛМД. Пример построения трехразрядного двоичного сумматора на ПЛМД (8, 4, 16) показан на рис. 4.18. На этом рисунке c_3 — перенос из третьего разряда. Суммарная информационная емкость матриц равна 320 бит, что значительно меньше емкости матриц для ПЛМ и ПЗУ. Заметим, что преимущества использования ПЛМД по сравнению с ПЛМ и ПЗУ существенно возрастают при увеличении L .

Перейдем к построению на ПЛУ схемы, реализующей операцию $S=A-B$, где A и B — положительные числа; переменные a_L и b_L представляют значения знакового разряда. Поскольку числа A и B — положительные, то $a_L = -b_L = 0$. Для выполнения операции вычитания сформируем в ПЛУ дополнительный код числа $-B$, а результат S будем также представлять в дополнительном коде. Напомним, что дополнительный код положительного двоичного числа совпадает с прямым кодом этого числа. Для образования дополнительного кода отрицательного двоичного числа необходимо в знаковом разряде поставить единицу, во всех цифровых разрядах единицы заменить нулями, а нули — единицами и к младшему разряду прибавить единицу.

Чтобы реализовать операцию добавления единицы к младшему разряду, положим $c_0=1$, операции же замены единиц нулями и нулей единицами осуществим заменой вычитаемого $B=b_1 \dots b_L$ на $\bar{B}=\bar{b}_1, \dots, \bar{b}_L$. После этого операцию вычитания $S=A-B$ можно заменить следующей операцией: $S=A+\bar{B}+1$. Пусть $L=3$. С учетом того, что $c_0=1$, $a_i \sim \bar{b}_i = a_i \oplus b_i$, $a_i \oplus \bar{b}_i = a_i \sim b_i$, функции s_1, s_2, s_3 запишутся в виде:

$$\begin{aligned} s_1 &= a_1 \oplus b_1; \\ s_2 &= (a_2 \oplus b_2) a_1 \bar{b}_1 \vee (a_2 \oplus b_2) (a_1 \vee \bar{b}_1) \vee (a_2 \sim b_2) \bar{a}_1 b_1; \quad (4.8) \\ s_3 &= (a_3 \oplus a_3) a_2 \bar{b}_2 \vee (a_3 \oplus b_3) (a_2 \vee \bar{b}_2) a_1 \bar{b}_1 \vee (a_3 \oplus b_3) \wedge \\ &\wedge (a_2 \vee \bar{b}_2) (a_1 \vee \bar{b}_1) \vee (a_3 \sim b_3) \bar{a}_2 b_2 \vee (\bar{a}_3 \sim b_3) (\bar{a}_2 \vee b_2) \bar{a}_1 b_1. \end{aligned}$$

Функции (4.8) могут быть реализованы на ПЛМД (6, 3, 9) так, как это показано на рис. 4.19. Здесь s_3 — знаковый разряд результата; s_2 и s_1 — его старший и младший разряды. Схемы, позволяющие выполнять арифметические операции других типов, строятся аналогичным образом.

Рассмотрим теперь правила построения на ПЛУ схем для выполнения логических операций. Пусть заданы, на-

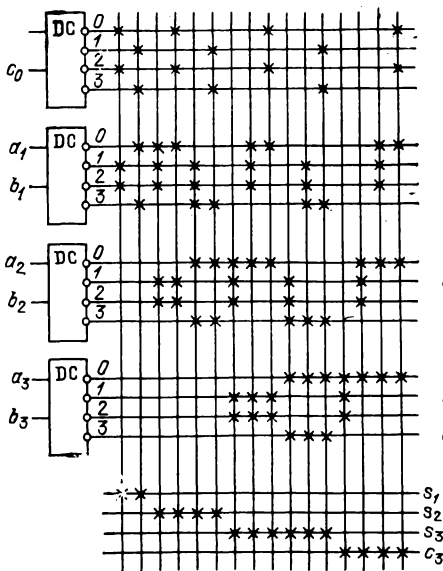


Рис. 4.18

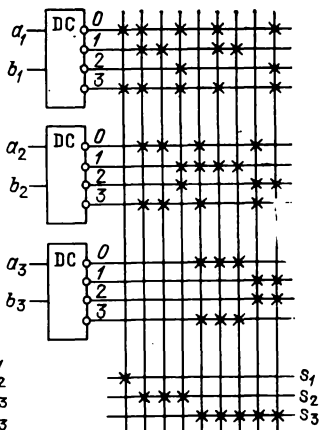


Рис. 4.19

пример, трехразрядные двоичные числа A и B , значения разрядов которых представляются соответственно значениями переменных a_1, a_2, a_3 и b_1, b_2, b_3 . Необходимо найти значения разрядов s_1^i, s_2^i, s_3^i двоичного числа S^i ($i=\overline{0,3}$), которые вычисляются по формулам:

$$\begin{aligned}
 s_1^0 &= a_1 \oplus b_1; & s_2^0 &= a_2 \oplus b_2; & s_3^0 &= a_3 \oplus b_3; \\
 s_1^1 &= a_1 b_1; & s_2^1 &= a_2 b_2; & s_3^1 &= a_3 b_3; \\
 s_1^2 &= a_1 \vee b_1; & s_2^2 &= a_2 \vee b_2; & s_3^2 &= a_3 \vee b_3; \\
 s_1^3 &= a_1 \sim b_1; & s_2^3 &= a_2 \sim b_2; & s_3^3 &= a_3 \sim b_3.
 \end{aligned}
 \tag{4.9}$$

Схемы, реализующие функции (4.9), широко используются в различных арифметико-логических устройствах. Закодируем значения переменной i двоичным двухразрядным кодом $v_1 v_2$ так, что при $i=0$ $v_1=v_2=0$; при $i=1$ $v_1=0, v_2=1$; при $i=2$ $v_1=1, v_2=0$; при $i=3$ $v_1=v_2=1$. На рис. 4.20 изображена схема на ПЛМД (8, 3, 12), реализующая все рассмотренные преобразования. Изменив значения v_1 и v_2 на входах ПЛМД, можно осуществить внешнюю настройку этой схемы на вычисление значений одного

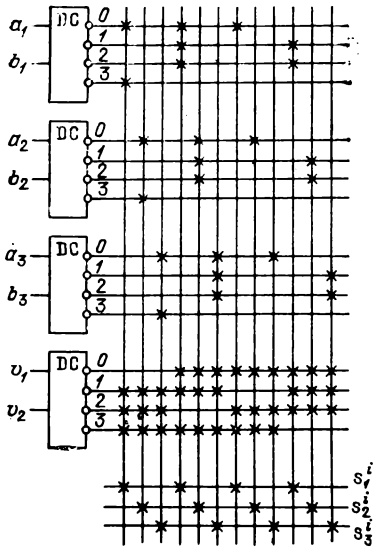


Рис. 4.20

регистре записано двоичное число A , значения разрядов которого представлены значениями переменных a_L, \dots, a_1 (a_L — старший, а a_1 — младший разряд). Тогда если в этом регистре обеспечивается сдвиг числа A вправо, то в следующем момент $t+1$ переменные a_1, \dots, a_L примут значения:

$$a_L := 0; a_{L-1} := a_L; a_{L-2} := a_{L-1}; \dots; a_1 := a_2, \quad (4.10)$$

а если влево, то:

$$a_L := a_{L-1}; a_{L-1} := a_{L-2}; a_{L-2} := a_{L-3}; \dots; a_1 := 0. \quad (4.11)$$

Часто в сдвигающих регистрах обеспечивается циклический сдвиг, в котором при сдвиге вправо $a_L := a_1$, а при сдвиге влево $a_1 := a_L$.

Цифровые устройства рассмотренного типа достаточно просто реализуются на ПЛМ с памятью. На рис. 4.21 приведена схема универсального регистра на ПЛМ (6, 4, 12, 4), для которого $L=4$. Такой регистр обеспечивает хранение четырехразрядного двоичного числа, поступающего на внешние входы D_1, \dots, D_4 . Запись этого числа во внутренний регистр RG ПЛМ (6, 4, 12, 4) осуществляется при $v_1=1, v_2=R=0$ и при наличии сигнала синхронизации на

из чисел, S^0, S^1, S^2 или S^3 . Схемы, позволяющие выполнить логические операции других типов, строятся аналогичным образом.

4.5. Регистры и счетчики

Регистры относятся к устройствам, наиболее широко используемым в цифровой аппаратуре для запоминания и хранения двоичных слов информации. Среди разнообразных регистров различного назначения большое распространение получили сдвигающие регистры. Пусть в некоторый момент t дискретного времени в сдвигающем

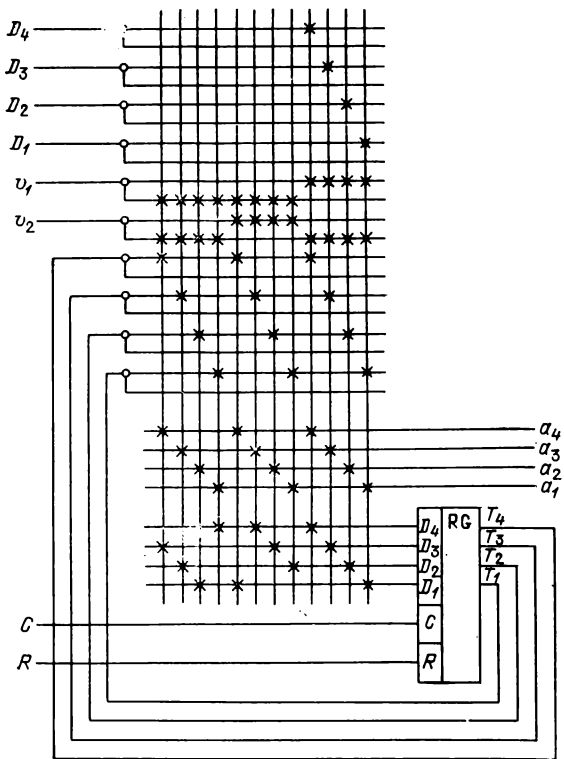


Рис. 4.21

внешнем входе C . Внешний вход R служит для асинхронной установки внутреннего регистра RG ПЛМ (6, 4, 12, 4) в состояние 0. При $v_1=v_2=R=0$ каждый импульс синхронизации, поступающий на внешний вход C , обеспечивает циклический сдвиг содержимого регистра RG на один разряд вправо, а при $v_1=R=0$ и $v_2=1$ — на один разряд влево. Значения разрядов числа A , записанного в регистре, формируются на выходах a_1, \dots, a_4 . Аналогичным образом могут быть построены регистры со сдвигом на большее число разрядов. Например циклическому сдвигу в регистре (вправо или влево) на два разряда соответствуют выражения:

$$\begin{aligned}
 a_L &:= a_2; & a_{L-1} &:= a_1; & a_{L-2} &:= a_L; & \dots; & a_1 &:= a_3; \\
 a_L &:= a_{L-2}; & a_{L-1} &:= a_{L-3}; & a_{L-2} &:= a_{L-4}; & \dots; & a_1 &:= a_{L-1}
 \end{aligned}$$

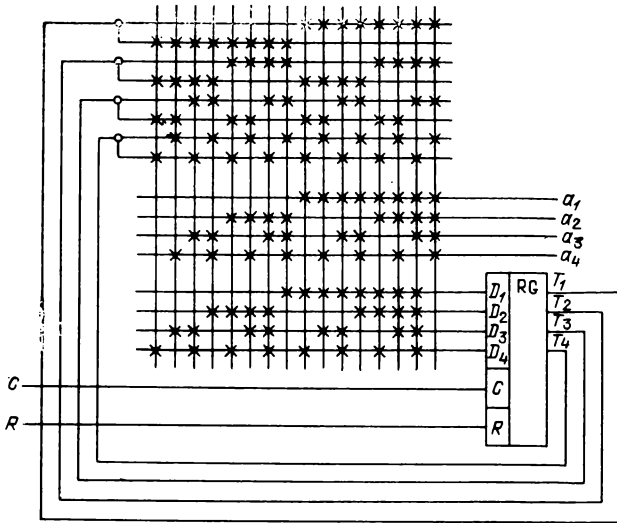


Рис. 4.22

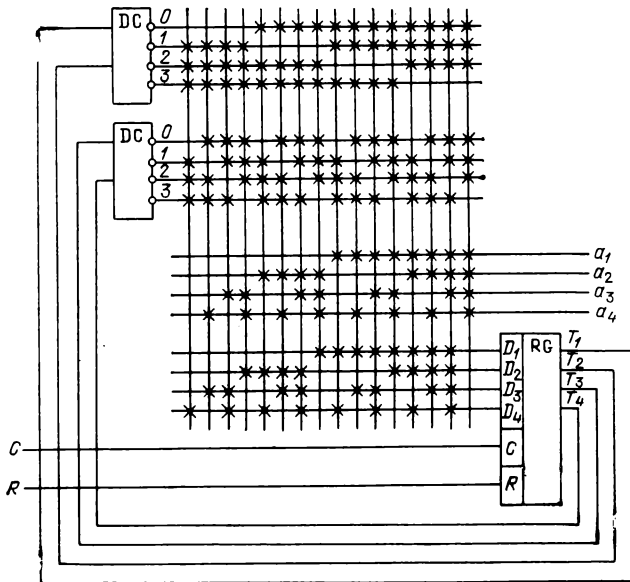


Рис. 4.23

Двоичные счетчики также относятся к распространенным цифровым устройствам. Они предназначены для выполнения операций вида $A := A + 1$ и $A := A - 1$, где A — двоичное число, записанное в счетчике. Другими словами, в каждый последующий момент времени $t+1$ обеспечивается либо увеличение на 1, либо уменьшение на 1 значения числа, содержащегося в счетчике в момент времени t . В первом случае счетчик называется суммирующим, во втором — вычитающим, если же реализуются оба преобразования $A := A + 1$ и $A := A - 1$, то реверсивным.

На рис. 4.22 показана схема четырехразрядного суммирующего счетчика, построенная на ПЛМ (0, 4, 16, 4), а на рис. 4.23 — аналогичная схема, построенная на ПЛМД (0, 4, 16, 4). Операции счета в этих схемах реализуются при $R=0$ и поступлении синхронизирующего импульса на внешний вход S . Содержимое счетчика формируется на выходах a_1, \dots, a_4 . Аналогичным образом могут быть построены вычитающие и реверсивные двоичные счетчики.

Таблица 4.4

a_4	a_3	a_2	a_1	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	1	1	1	0	1	1	1
0	0	0	1	0	1	0	0	1	0	0
0	0	1	0	1	0	1	1	1	0	1
0	0	1	1	1	0	1	1	0	1	1
0	1	0	0	0	1	1	1	0	1	0
0	1	0	1	1	1	0	1	0	1	1
0	1	1	0	1	1	0	1	1	1	1
0	1	1	1	1	0	1	0	0	1	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

На рис. 4.24 изображена схема суммирующего десятичного счетчика с цифровой индикацией, построенного на ПЛМ (5, 7, 14, 4). При $v=1, R=0$ с приходом синхронизирующего импульса двоичное число с внешних входов D_1, \dots, D_4 заносится во внутренний регистр RG ПЛМ (5, 7, 14, 4). При $v=0, R=0$ реализуется операция счета. Двоичный код, записанный во внутреннем регистре RG, преобразуется в двухзначный семиразрядный код для управления семисегментным цифровым индикатором. В результате двоичное число A , хранимое в счетчике, отобра-

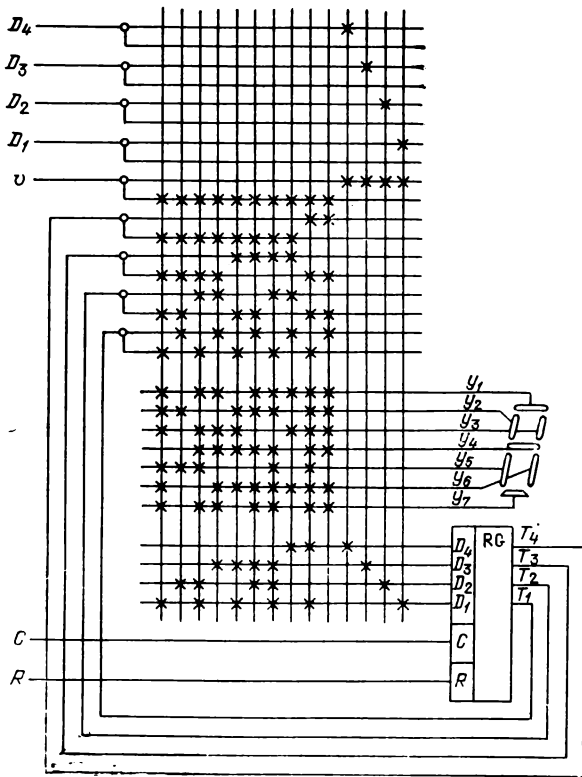


Рис. 4.24

жается в виде десятичного числа на экране этого индикатора. Пусть значения разрядов числа представляются значениями переменных a_1, \dots, a_4 , тогда функции y_1, \dots, y_7 (см. рис. 4.24) являются булевыми функциями аргументов a_1, \dots, a_4 , заданными табл. 4.4. Эти функции не определены на двоичных наборах значений переменных с десятичными эквивалентами 10, ..., 15. Минимизация функций, заданных табл. 4.4, и функций управления переключением элементов памяти внутреннего регистра RG ПЛМ (5, 7, 14, 4) не производилась.

**СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ
НА ПРОГРАММИРУЕМЫХ
ЛОГИЧЕСКИХ УСТРОЙСТВАХ**

5.1. Тривиальная реализация

Рассмотрим систему θ из N булевых функций L переменных, описывающую поведение некоторой комбинационной схемы. Пусть эта система представлена в дизъюнктивной нормальной форме (ДНФ), в которой число попарно неравных элементарных конъюнкций равно B . Опишем способы реализации системы θ на различных программируемых логических устройствах (ПЛУ).

Если выполняются условия $L \leq s$, $N \leq t$, $B \leq q$, то система θ может быть тривиально реализована на одной ПЛМ (s, t, q). Например, если θ представлена следующими булевыми функциями y_1, \dots, y_4 :

$$\begin{aligned} y_1 &= \overline{x_1} \overline{x_2} \vee \overline{x_1} x_3 x_4 \vee x_1 x_2 x_3 x_4; \\ y_2 &= \overline{x_1} x_2 \vee x_1 x_2 x_3 x_4 \vee \overline{x_2} x_4; \\ y_3 &= \overline{x_1} \overline{x_2} \vee \overline{x_1} x_3 x_4 \vee x_1 x_2 x_3 \overline{x_4}; \\ y_4 &= \overline{x_2} x_4 \vee x_1 x_3 x_4 \vee x_1 x_2 x_3 x_4, \end{aligned} \tag{5.1}$$

то она может быть реализована на одной ПЛМ (4, 4, 5), программирование которой показано в табл. 5.1.

Таблица 5.1

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
0	0	—	—	1	1	1	.
0	—	0	1	1	.	1	1
1	1	1	1	1	1	.	1
—	0	—	0	.	1	.	1
1	1	1	0	.	.	1	.

Таблица 5.2

Адрес	y_1	y_2	y_3	y_4
0	1	1	1	1
1	1	1	1	1
2	1	1	1	1
3	1	1	1	.
5	1	.	1	1
8	.	1	.	1
10	.	1	.	1
14	.	.	1	.
15	1	1	.	1

Для тривиальной реализации системы θ на ПЗУ (s, t) достаточно выполнение двух условий: $L \leq s, N \leq t$, однако в этом случае соответствующие БФ должны быть представлены в совершенной ДНФ. В нашем примере система (5.1) расширяется до следующей совершенной ДНФ:

$$\begin{aligned}
 y_1 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \\
 &\quad \vee \overline{x_1 x_2 x_3 x_4}; \\
 y_2 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \\
 &\quad \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}; \\
 y_3 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \\
 &\quad \vee \overline{x_1 x_2 x_3 x_4}; \\
 y_4 &= \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4} \vee \\
 &\quad \vee \overline{x_1 x_2 x_3 x_4} \vee \overline{x_1 x_2 x_3 x_4}.
 \end{aligned} \tag{5.2}$$

Функции (5.2) могут быть реализованы на одном ПЗУ (4, 4), программирование которого показано в табл. 5.2, где использовано упрощенное изображение ПЗУ (матрица M_2 представлена в том же виде, что и матрица M_2 ПЛМ, а номера выходов дешифратора, связанные со входами этой матрицы, указаны в столбце «Адрес»). Для уменьшения громоздкости табл. 5.2 из нее исключены строки, содержащие в столбцах y_1, \dots, y_4 только точки. Договоримся далее изображать ПЗУ в таком же виде.

В тех случаях, когда схема, заданная системой θ , должна быть построена на ПЛМ (z, q), тривиальная реализация соответствующих БФ возможна при выполнении условий: $N+L \leq z; B \leq q$. После распределения входных (x_1, \dots, x_L) и выходных (y_1, \dots, y_N) переменных между z шинами ПЛМ (z, q) она программируется так же, как и ПЛМ (s, t, q).

Если выполняются условия: $L \leq s, B \leq q$, но $N > t$, т. е. выходов ПЛМ (s, t, q) либо ПЗУ (s, t) не хватает для назначения на них всех выходных переменных y_1, \dots, y_N , система θ может быть тривиально реализована на нескольких ПЛМ (s, t, q) либо ПЗУ (s, t) за счет расширения ПЛМ или ПЗУ по выходам. В этом случае одноименные входные шины ПЛМ или ПЗУ, число которых равно k_1 , объединяются между собой так, что число входов k_1 элементов становится равным s (здесь $k_1 = \lceil N/t \rceil$). На s входов k_1 элементов назначаются входные переменные из множества $\{x_1, \dots, x_L\}$, $L \leq s$.

Значения выходных переменных из множества $\{y_1, \dots, y_N\}$ формируются на выходах k_1 элементов, число этих выходов $k_1 t \geq N$.

Распределение выходных переменных y_1, \dots, y_N между k_1 элементами осуществляется произвольным образом. Программирование k_1 ПЛМ (s, t, q) либо ПЗУ (s, t) выполняется так же, как программирование одной ПЛМ $(s, k_1 t, q)$ или ПЗУ $(s, k_1 t)$.

Пусть, например, систему (5.1) необходимо реализовать на ПЛМ $(4, 2, 5)$ и ПЗУ $(4, 2)$. Тогда $k_1=2$; программирование ПЛМ $(4, 2, 5)$ приведено в табл. 5.3 и табл. 5.4, ПЗУ $(4, 2)$ — в табл. 5.5 и табл. 5.6.

Если выполняются условия $L \leq s$, $N \leq t$, но $B > q$, т. е. не хватает промежуточных шин в ПЛМ (s, t, q) , система θ может быть тривиально реализована на нескольких ПЛМ (s, t, q) за счет их расширения по промежуточным шинам. В этом случае одноименные входы и выходы k_2 ПЛМ (s, t, q) объединяются¹ между собой так, что число входов и выходов k_2 ПЛМ равно соответственно s и t (здесь $k_2 = \lceil B/q \rceil$). На L входов каждой ПЛМ (s, t, q) , $L \leq s$, подаются значения входных переменных из множества $\{x_1, \dots, x_L\}$, на N выходах этих ПЛМ, $N \leq t$, формируются значения выходных переменных из множества $\{y_1, \dots, y_N\}$. Число же промежуточных шин в группе из k_2 ПЛМ (s, t, q) с учетом того, что одноименные входные и выходные шины во всех ПЛМ объединены, определяется значением $k_2 q \geq B$.

Поэтому дальнейшая настройка ПЛМ (s, t, q) может выполняться обычным способом и k_2 ПЛМ (s, t, q) программируются так же, как одна ПЛМ $(s, t, k_2 q)$. Распределение элементарных конъюнкций системы θ для их реализации на промежуточных шинах различных ПЛМ осуществляется произвольным образом. Предположим, что систему (5.1) необходимо реализовать на ПЛМ $(4, 4, 3)$. Тогда $k_2=2$; программирование ПЛМ $(4, 4, 3)$ приведено в табл. 5.7 и 5.8.

Если $L \leq s$, но $N > t$, $B > q$, т. е. в ПЛМ (s, t, q) не хватает как выходных, так и промежуточных шин, то может быть выполнено одновременное расширение ПЛМ $(s,$

¹ Если несколько выходов объединяется между собой, то предполагается, что в точке соединения реализуется функция «монтажное ИЛИ». В тех случаях, когда для используемых элементов такая реализация недопустима, выходы объединяются на дополнительных схемах ИЛИ. В качестве таких схем могут быть использованы элементы малой степени интеграции, ПЛМ, ПМВ, ПМЛ и др.

Таблица 5.3

x_1	x_2	x_3	x_4	y_1	y_2
0	0	—	—	1	1
0	—	0	1	1	.
1	1	1	1	1	1
—	0	—	0	.	1

Таблица 5.4

x_1	x_2	x_3	x_4	y_3	y_4
0	0	—	—	1	.
0	—	0	1	1	1
1	1	1	1	.	1
—	0	—	0	.	1
1	1	1	0	1	.

Таблица 5.5

Адрес	y_1	y_2
0	1	1
1	1	1
2	1	1
3	1	1
5	1	.
8	.	1
10	.	1
15	1	1

Таблица 5.6

Адрес	y_3	y_4
0	1	1
1	1	1
2	1	1
3	1	.
5	1	1
8	.	1
10	.	1
14	1	.
15	.	1

t, q) по выходам и промежуточным шинам. Например, сначала расширяются ПЛМ (s, t, q) по выходам, а затем группа из k_1 ПЛМ (s, t, q) по промежуточным шинам или, наоборот, сначала расширяются ПЛМ (s, t, q) по промежуточным шинам, а затем группа из k_2 ПЛМ (s, t, q) по выходам. После расширения ПЛМ (s, t, q) по выходам и промежуточным шинам их число будет равно $k_1 k_2$. В этом случае $k_1 k_2$ ПЛМ (s, t, q) программируются так же, как одна ПЛМ $(s, k_1 t, k_2 q)$. Заметим, что такая тривиальная реализация обычно является в значительной степени избыточной, что, например, показано в [1]. Применяя специальные приемы [1, 17], можно сократить чис-

Таблица 5.7

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
0	0	—	—	1	1	1	.
0	—	0	1	1	.	1	1
1	1	1	1	1	1	.	1

Таблица 5.8

x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4
—	0	—	0	.	1	.	1
1	1	1	0	.	.	1	.

ло элементов одноуровневой схемы. Методы, используемые для этих целей, представляют собой частный случай более общих методов, излагаемых в следующем параграфе.

Если для некоторой системы θ и ПЛМ (s, t, q) либо ПЗУ (s, t) выполняется условие $L > s$, то в общем случае реализовать соответствующие БФ тривиальными способами не удастся. Ниже будут рассмотрены простые для понимания и доступные для практического использования методы и алгоритмы решения этой задачи.

5.2. Одноуровневые схемы

Обозначим через L^i число переменных в записи i -й элементарной конъюнкции заданной системы θ булевых функций, и пусть $L^{\max} = \max_i L^i$. В [1, 17] показано, что если выполняется условие $L^{\max} \leq s$, то система θ всегда может быть реализована одноуровневой схемой из ПЛМ (s, t, q) или ПЗУ (s, t) . Под одноуровневой понимается такая схема, в которой сигналы от ее входов x_1, \dots, x_L проходят на выходы y_1, \dots, y_N не более чем через один элемент. Пусть, например, задана система θ полностью определенных БФ:

$$\begin{aligned}
 y_1 &= x_3 \overline{x_8} x_{10} \vee \overline{x_3} x_8 \overline{x_{10}}; \\
 y_2 &= \overline{x_6} x_{10} x_{11} \vee x_6 x_7 x_9 x_{10} \overline{x_{11}} \vee x_6 x_{10} x_{11}; \\
 y_3 &= x_6 x_7 \overline{x_9} x_{10} \overline{x_{11}} \vee x_6 x_{10} x_{11}; \\
 y_4 &= x_2 \overline{x_8} x_{10} \vee \overline{x_2} x_3 x_5 \overline{x_8} x_{10} \vee x_3 x_8 \overline{x_{10}}; \\
 y_5 &= x_1 \overline{x_4} x_6 x_{10} \overline{x_{11}} \vee x_1 x_4 \overline{x_6} x_{10} \overline{x_{11}}; \\
 y_6 &= x_3 \overline{x_8} x_{10} \vee \overline{x_3} x_8 \overline{x_{10}} \vee x_2 \overline{x_8} x_5 x_8 \overline{x_{10}}; \\
 y_7 &= x_1 \overline{x_4} x_6 x_{10} \overline{x_{11}} \vee \overline{x_6} x_{10} x_{11} \vee x_6 x_7 \overline{x_9} x_{10} \overline{x_{11}} \vee \overline{x_1} x_6 x_{10} \overline{x_{11}}; \\
 y_8 &= x_1 \overline{x_4} x_6 x_{10} \overline{x_{11}}.
 \end{aligned} \tag{5.3}$$

Для функций (5.3) $L=11$, $N=8$, $B=10$, $L^{\max}=5$. Эти функции могут быть реализованы одноуровневой схемой из ПЛМ либо ПЗУ с числом входов $s \geq L^{\max}=5$.

Рассмотрим систему θ булевых функций с параметрами L, N, B, L^{\max} и ПЛМ (s, t, q) , для которых выполняются условия $L > s$, $N > t$, $B > q$, $L^{\max} \leq s$. Необходимо построить из ПЛМ (s, t, q) одноуровневую схему, реализующую заданную систему θ . Обозначим через $E = \{e_1, \dots, e_B\}$ множество различных элементарных конъюнкций (термов) рассматриваемой системы θ , $X(e_b)$ — множест-

во переменных, входящих в терм e_b , и пусть $|Y(e_b)| \leq t$, $b = \overline{1, B}$, где $Y(e_b)$ — множество функций, в которые входит терм e_b . Другими словами, будем рассматривать такие системы θ , в которых каждый терм e_b входит не более чем в t функций. Задачу синтеза одноуровневой схемы, реализующей систему θ на ПЛМ (s, t, q) , сформулируем следующим образом: на множестве E термов системы θ , удовлетворяющих ограничениям $|Y(e_b)| \leq t$, $L^b \leq s$, найти минимальное по числу блоков разбиение $\pi_E = \{E^1, \dots, E^u\}$ такое, что для каждого блока E^u этого разбиения должны быть справедливы:

$$|X(E^u)| \leq s; \quad (5.4)$$

$$|Y(E^u)| \leq t; \quad (5.5)$$

$$|E^u| \leq q. \quad (5.6)$$

В (5.4) $|X(E^u)|$ — число различных входных переменных во всех термах блока E^u ; в (5.5) $|Y(E^u)|$ — число функций, в которые входят термы блока E^u :

$$X(E^u) = \bigcup_{e_b \in E^u} X(e_b);$$

$$Y(E^u) = \bigcup_{e_b \in E^u} Y(e_b).$$

Рассмотрим приближенный последовательный алгоритм поиска разбиения π_E термов системы θ [18]. Изложение будем сопровождать примером синтеза одноуровневой схемы, реализующей булевой функции (5.3), на ПЛМ (5, 3, 5). Для системы (5.3) построим табл. 5.9, в которой каждому терму e_b поставлены в соответствие его множество $X(e_b)$ входных переменных и множество $Y(e_b)$ функций, в которые входит этот терм. Из табл. 5.9 сразу видно, что не любые сочетания термов могут быть реализованы в одной ПЛМ. Например, если блок E^u включает термы e_4, e_9 , то нарушается ограничение (5.4), поскольку $X(E^u) = \{x_1, x_3, x_6, x_8, x_{10}, x_{11}\}$. Точно так же включение в один блок E^p термов e_2 и e_6 приводит к нарушению ограничения (5.5), так как в этом случае $Y(E^p) = \{y_1, y_2, y_6, y_7\}$, хотя соотношение (5.4) выполняется: $X(E^p) = \{x_3, x_6, x_8, x_{10}, x_{11}\}$.

В последовательном алгоритме будем использовать три основные оценки:

$c(e_i, e_j)$ — близость двух термов e_i и e_j ;

$c(e_i, E^u)$ — близость терма e_i к блоку E^u ;

$W(e_i, E^u)$ — весовая функция терма e_i по отношению к блоку E^u . Здесь e_i — терм, не вошедший в блок E^u ($e_i \in E \setminus E^u$), а $W(e_i, E^u)$ является критерием выбора очередного терма для включения в блок E^u разбиения $\mathcal{L}E$.

Близость двух термов будем определять выражением

$$c(e_i, e_j) = k_1 |X(e_i) \cap X(e_j)| + k_2 |Y(e_i) \cap Y(e_j)|. \quad (5.7)$$

Таблица 5.9

e_b	$X(e_b)$	$Y(e_b)$
e_1	x_3, x_8, x_{10}	y_1, y_4, y_6
e_2	x_3, x_8, x_{10}	y_1, y_6
e_3	$x_2, x_3, x_5, x_8, x_{10}$	y_4, y_6
e_4	x_3, x_8, x_{10}	y_4
e_5	$x_1, x_4, x_6, x_{10}, x_{11}$	y_5, y_7, y_8
e_6	x_6, x_{10}, x_{11}	y_2, y_7
e_7	$x_6, x_7, x_9, x_{10}, x_{11}$	y_2, y_3, y_7
e_8	x_6, x_{10}, x_{11}	y_2, y_3
e_9	x_1, x_6, x_{10}, x_{11}	y_7
e_{10}	$x_1, x_4, x_6, x_{10}, x_{11}$	y_5

Здесь k_1, k_2 — весовые коэффициенты. Соотношение этих коэффициентов зависит от степени значимости ограничений (5.4), (5.5), т. е. от соотношений параметров L и N системы θ , определяющих число входных и выходных переменных, и параметров s и t , определяющих число входов и выходов в ПЛМ (s, t, q). В примере положим $k_1 = k_2 = 1$. Так,

$$\begin{aligned} c(e_1, e_3) &= |\{x_3, x_8, x_{10}\} \cap \{x_2, x_3, x_5, x_8, x_{10}\}| + \\ &+ |\{y_1, y_4, y_6\} \cap \{y_4, y_6\}| = |\{x_3, x_8, x_{10}\}| + \\ &+ |\{y_4, y_6\}| = 3 + 2 = 5. \end{aligned}$$

Другими словами, в первой и третьей строках табл. 5.9 встречаются три одинаковые входные и две выходные переменные.

Величину близости e_i к блоку E^u определим выражением

$$c(e_i, E^u) = \sum_{e_j \in E^u} c(e_i, e_j). \quad (5.8)$$

Наконец, для весовой функции $W(e_i, E^u)$ используем соотношение

$$W(e_i, E^u) = c(e_i, E^u) - c[e_i, E \setminus (E^1 \cup \dots \cup E^u)], \quad (5.9)$$

определяемое как разность между близостями терма e_i к блоку E^u и к множеству $E \setminus (E^1 \cup \dots \cup E^u)$, не вошедших в уже сформированные ранее блоки E^1, \dots, E^u термов.

В алгоритме поиска разбиения λ_E формирование блока E^u ($u=1, U$) осуществляется в два этапа. На первом этапе в блок E^u включается опорный элемент — терм e_i , для которого

$$c(e_i, E' \setminus \{e_i\}) = \max_j c(e_j, E' \setminus \{e_j\}),$$

где $E' = E \setminus (E^1 \cup \dots \cup E^{u-1})$ — множество еще не распределенных термов; $e_i, e_j \in E'$, а $c(e_j, E' \setminus \{e_j\})$ определяется по формуле (5.8). Таким образом, в качестве опорного (первого) элемента блока E^u выбирается терм, имеющий наибольшую близость ко всем еще не распределенным термам, т. е. термам, не вошедшим в блоки E^1, \dots, E^{u-1} .

Второй этап является многошаговым. На каждом шаге в E^u по описанному ниже правилу добавляется очередной терм. Процесс формирования блока E^u заканчивается если все термы уже распределены по блокам, т. е. $E \setminus (E^1 \cup \dots \cup E^u) = \emptyset$, или если в блок E^u нельзя добавить ни одного терма без нарушения какого-либо из ограничений (5.4) — (5.6).

Правило добавления очередного терма в блок E^u сводится к следующему. Пусть после i -го шага второго этапа в блоке E^u вошли термы $E^u = \{e_\alpha, e_\beta, \dots, e_\omega\}$. На $(i+1)$ -м шаге по формуле (5.9) определяется весовая функция $W(e_t, E^u)$ для каждого из еще нераспределенных термов $e_t \in E \setminus (E^1 \cup \dots \cup E^u)$, у которого $c(e_t, E^u) \neq 0$. Из всех таких термов выбирается терм e_j , для которого $W(e_j, E^u) = \max_t W(e_t, E^u)$, если его включение в блок E^u не нарушает какого-либо из ограничений (5.4) — (5.6). Если таких термов несколько, то из них выбирается терм e_t с $\max_t c(e_t, E^u)$, если и эти оценки одинаковы для нескольких термов, в блок E^u включается любой из них.

Если терм e_j с наибольшей весовой функцией $W(e_j, E^u)$ нельзя включить в блок из-за ограничений (5.4) — (5.6), то в порядке убывания весовой функции рассматриваются еще не включенные в блоки термы с ненулевой близостью к блоку E^u . Среди них выбирается первый, не нарушающий ограничений (5.4) — (5.6). Затем в

том же порядке просматриваются термы с нулевой близостью к блоку E^u , т. е. $c(e_i, E^u) = 0$.

Таким образом, последовательный алгоритм поиска разбиения π_E состоит из нескольких шагов.

1. Построить матрицу близости всех пар термов множества E , на пересечении i -й строки и j -го столбца которой записать функцию близости $c(e_i, e_j)$, $e_i, e_j \in E$.

2. Положить $u=1$, $E'=E$.

3. Выбрать опорный элемент для формирования u -го блока E^u разбиения π_E согласно описанному выше правилу.

4. Сформировать u -й блок E^u разбиения π_E .

5. Если $E' \setminus E^u = \emptyset$, то перейти к п. 8, иначе — к п. 6.

6. Положить $E' := E' \setminus E^u$ и $u := u + 1$.

7. Вычислить функции $c(e_i, E' \setminus \{e_i\})$ для всех $e_i \in E'$ и перейти к п. 3.

8. Конец.

Проиллюстрируем алгоритм поиска разбиения π_E на примере системы θ булевых функций (5.3) при использовании ПЛМ (5, 3, 5). Исходные данные для вычисления оценок $c(e_i, e_j)$, $c(e_i, E^u)$ и $W(e_i, E^u)$ приведены в табл. 5.9, матрица близости термов с учетом $k_1 = k_2 = 1$ — в табл. 5.10.

Таблица 5.10

	e_1	e_2	e_3	e_4	e_5	e_6	e_7	e_8	e_9	e_{10}
e_1	—	5	5	4	1	1	1	1	1	1
e_2	5	—	4	3	1	1	1	1	1	1
e_3	5	4	—	4	1	1	1	1	1	1
e_4	4	3	4	—	1	1	1	1	1	1
e_5	1	1	1	1	—	4	4	3	4	4
e_6	1	1	1	1	4	—	5	4	4	3
e_7	1	1	1	1	4	5	—	5	4	3
e_8	1	1	1	1	3	4	5	—	3	3
e_9	1	1	1	1	4	4	4	3	—	3
e_{10}	1	1	1	1	4	3	3	3	3	—

Последовательность формирования блоков разбиения π_E представлена в табл. 5.11. В столбе E этой таблицы перечислены все термы $e_b \in E$. В столбцах $c(e_i, E')$ записаны величины близости каждого терма со всеми остальными, еще не распределенными. Согласно алгоритму они используются для выбора опорных элементов при

Таблица 5.11

E	$c(e_i, E^i)$	$W(e_i, E^1)$				$c(e_i, E^i)$	$W(e_i, E^2)$					$c(e_i, E^i)$	$W(e_i, E^3)$					
		1	2	3	4		1	2	3	4	5		1	2	3	4		
e_1	20	-18	-	-	-	17	\oplus	+	+	+	+							
e_2	18	-16	-	-	-	15	-5	\oplus	+	+	+							
e_3	19	-17	-	-	-	16	-6	2	\oplus	+	+							
e_4	17	-15	-	-	-	14	-6	0	8	\oplus	+							
e_5	23	-15	-	-	-	12	-10	-	-	-	-	8	\oplus	+	+	+		
e_6	24	-14	-6	\oplus	+													
e_7	25	\oplus	+	+	+													
e_8	22	-12	\oplus	+	+													
e_9	22	-14	-	-	-	11	-9	-	-	-	-	7	1	\oplus	+	+		
e_{10}	20	-14	-	-	-	11	-9	-	-	-	-	7	1	7	\oplus	+		
X		$x_6, x_7, x_9, x_{10}, x_{11}$					$x_3, x_8, x_{10}, x_2, x_5$						$x_1, x_4, x_6, x_{10}, x_{11}$					
Y		y_2, y_3, y_7					y_1, y_4, y_6						y_5, y_7, y_8					

формировании очередного блока разбиения. Величины $W(e_i, E^u)$ используются для включения очередного термина в блок. Плюс в строке, относящейся к формированию u -го блока разбиения, означает, что терм, соответствующий этой строке, включен в u -й блок. Например, терм e_6 включен в первый блок разбиения на третьем шаге формирования этого блока. Минус в строке означает, что соответствующий терм нельзя включить в формируемый блок из-за нарушения ограничений (5.4)—(5.6). Так терм e_1 нельзя включить в блок E^1 на втором шаге, поскольку нарушаются оба ограничения (5.4) и (5.5). Для облегчения контроля за выполнением ограничений (5.4)—(5.6) в нижних двух строках табл. 5.11 записаны переменные термов формируемого блока, а также функции, в которые входят эти термы.

В рассматриваемом примере разбиение π_E содержит три блока:

$$\begin{aligned}
 E^1 &= \{e_6, e_7, e_8\}; X(E^1) = \{x_6, x_7, x_9, x_{10}, x_{11}\}; \\
 Y(E^1) &= \{y_2, y_3, y_7\}; \\
 E^2 &= \{e_1, e_2, e_3, e_4\}; X(E^2) = \{x_2, x_3, x_5, x_8, x_{10}\}; \\
 Y(E^2) &= \{y_1, y_4, y_6\}; \\
 E^3 &= \{e_5, e_9, e_{10}\}; X(E^3) = \{x_1, x_4, x_6, x_{10}, x_{11}\}; \\
 Y(E^3) &= \{y_5, y_7, y_8\}.
 \end{aligned}$$

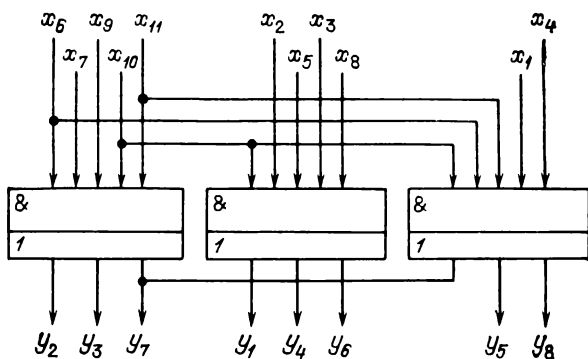


Рис. 5.1

Построенная по этим выражениям логическая схема приведена на рис. 5.1, таблицы программирования ПЛМ₁, ПЛМ₂ и ПЛМ₃ — в табл. 5.12, 5.13 и 5.14 соответственно.

Рассмотрим теперь возможности усовершенствования изложенного алгоритма, позволяющие полнее учитывать

Таблица 5.12

x_6	x_7	x_9	x_{10}	x_{11}	y_2	y_3	y_7
1	1	0	1	0	1	1	1
1	—	—	1	1	1	1	.
0	—	—	1	1	1	.	1

Таблица 5.13

x_2	x_3	x_5	x_8	x_{10}	y_1	y_4	y_6
—	1	—	0	0	1	1	1
—	0	—	1	0	1	.	1
0	0	1	0	0	.	1	1
—	1	—	1	0	.	1	.

Таблица 5.14

x_1	x_4	x_8	x_{10}	x_{11}	y_5	y_7	y_8
1	0	0	1	0	1	1	1
0	—	0	1	0	.	1	.
1	1	0	1	0	1	.	.

особенности реализуемой системы θ булевых функций с учетом заданных параметров s , t и q используемых ПЛМ.

Предположим, что для заданной системы θ и ПЛМ (s, t, q) выполняются условия $L \leq s$, $N > t$, $B > q$. Тогда

в выражении (5.7) целесообразно назначить $k_1 = 0$, $k_2 = 1$, поскольку ограничения вида (5.4) можно не учитывать. В остальном алгоритм нахождения разбиения π_E не изменяется; он позволяет строить более экономные по чис-

лу ПЛМ комбинационные схемы по сравнению с теми, которые синтезировались в § 5.1 с использованием тривиальных методов расширения ПЛМ по выходам и промежуточным шинам.

Если заданы система θ и ПЛМ (s, t, q) , причем ограничение $|Y(e_b)| \leq t$ для некоторых $b \in \{1, \dots, B\}$ не выполняется, то изложенный алгоритм может использоваться

Таблица 5.15

Адрес	y_2	y_3	y_7
3	1	.	1
7	1	.	1
11	1	.	1
15	1	.	1
19	1	1	.
23	1	1	.
26	1	1	1
27	1	1	.
31	1	1	.

Таблица 5.16

Адрес	y_1	y_4	y_6
2	1	.	1
4	.	1	1
6	1	.	1
8	1	1	1
10	.	1	.
12	1	1	1
14	.	1	.
18	1	.	1
22	1	.	1
24	1	1	1
26	.	1	.
28	1	1	1
30	.	1	.

Таблица 5.17

Адрес	y_6	y_7	y_8
2	.	1	.
10	.	1	.
18	1	1	1
26	1	.	.

совместно с рассмотренным в § 5.1 тривиальным методом расширения ПЛМ по выходам. Разделим множество E термов системы θ на классы E_1, \dots, E_l такие, что в i -й

класс E_i попадают только те термы, для которых выполняется условие $t(i-1) < |Y(e_b)| \leq t \cdot i$. Если в некоторый блок E^u на очередном шаге может быть включен терм $e_b \in E_i$, т. е. соответствующие оценки для этого терма принимают наилучшие значения, то ограничение (5.5) заменяется менее жестким: $|Y(E^u)| \leq t \cdot i$. Далее u -й блок реализуется на i ПЛМ, для которых выполнено расширение по выходам. Другие пути решения аналогичной задачи рассмотрены в [19, 20].

Качество решения задачи синтеза одноуровневой комбинационной схемы с помощью рассмотренного приближенного последовательного алгоритма можно повысить,

разумно выбирая значения весовых коэффициентов k_1 и k_2 . В [19, 20] показано, что целесообразно придавать этим коэффициентам различные значения, которые вычисляются на каждом шаге алгоритма.

Если в качестве элементов комбинационной схемы должны быть использованы ПЗУ (s, t), то методы синтеза практически не изменяются, не нужно лишь учитывать ограничения (5.6). Пусть, например, систему θ (5.3) необходимо реализовать на ПЗУ (5,3). Тогда общая структура полученной логической схемы будет такой же, как и на рис. 5.1, при условии, что ПЛМ (5, 3, 5) в этой схеме заменены на ПЗУ (5,3). Таблицы программирования ПЗУ₁, ПЗУ₂ и ПЗУ₃ приведены соответственно в табл. 5.15, 5.16 и 5.17. Булевы функции, реализуемые каждым ПЗУ, предварительно расширены до совершенной ДНФ.

5.3. Многоуровневые схемы

Необходимость в построении многоуровневых комбинационных схем из ПЛМ (s, t, q) возникает в тех случаях, когда выполняется условие $L^{\max} > s$. Многоуровневой схема названа потому, что сигналы от ее входов x_1, \dots, x_L проходят на выходы y_1, \dots, y_N более чем через один элемент. В настоящее время существует большое число различных приближенных алгоритмов синтеза таких схем [1, 17—27], которые в той или иной степени сводятся к декомпозиции систем булевых функций. Так, в [21] рассмотрен метод простой декомпозиции, основанный на использовании дизъюнктивного разложения Шеннона для БФ, представленных в совершенной ДНФ.

Идея дизъюнктивного разложения состоит в следующем. Пусть задана БФ

$$y(x_1, x_2, x_3, x_4) = \bar{x}_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee \\ \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4.$$

Эта функция может быть разложена, например, по переменной x_1 и представлена в виде

$$y(x_1, x_2, x_3, x_4) = \bar{x}_1 y(0, x_2, x_3, x_4) \vee \\ \vee x_1 y(1, x_2, x_3, x_4),$$

где функции $y(0, x_2, x_3, x_4)$ и $y(1, x_2, x_3, x_4)$, называемые коэффициентами разложения, получаются из $y(x_1, x_2, x_3, x_4)$ заменой переменной x_1 константами 0 и 1 соответственно. Таким образом,

$$y(x_1, x_2, x_3, x_4) = \bar{x}_1(\bar{x}_2x_3x_4 \vee \bar{x}_2\bar{x}_3\bar{x}_4 \vee x_2x_3\bar{x}_4 \vee x_2\bar{x}_3x_4) \vee \vee x_1(\bar{x}_2x_3\bar{x}_4 \vee \bar{x}_2\bar{x}_3x_4).$$

Точно так же может быть выполнено разложение заданной функции $y(x_1, x_2, x_3, x_4)$ одновременно по переменным x_1 и x_2 :

$$y(x_1, x_2, x_3, x_4) = \bar{x}_1\bar{x}_2y(0, 0, x_3, x_4) \vee \vee \bar{x}_1x_2y(0, 1, x_3, x_4) \vee x_1\bar{x}_2y(1, 0, x_3, x_4) \vee \vee x_1x_2y(1, 1, x_3, x_4), \quad (5.10)$$

Таблица 5.18

x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_1	y_2
0	0	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	1
0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	0	1
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0	1
0	1	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	0	0	0	1	1	0
1	0	0	0	0	1	0	1	0
1	0	0	0	1	0	0	1	0
1	0	0	1	0	0	0	1	0
1	0	1	0	0	0	0	1	0
1	1	0	0	0	0	0	1	0
0	1	0	0	0	0	1	1	0
0	1	0	0	0	1	0	1	0
0	1	0	0	1	0	0	1	0
0	1	0	1	0	0	0	1	0
0	1	1	0	0	0	0	1	0
0	0	1	0	0	0	1	1	0
0	0	1	0	0	1	0	1	0
0	0	1	0	1	0	0	1	0
0	0	1	1	0	0	0	1	0
0	0	0	0	1	0	1	1	0
0	0	0	0	1	1	0	1	0
0	0	0	0	0	1	1	1	0
1	1	1	1	1	1	1	1	1

где коэффициенты разложения определяются выражениями:

$$y(0, 0, x_3, x_4) = x_3x_4 \vee \bar{x}_3\bar{x}_4;$$

$$y(0, 1, x_3, x_4) = x_3\bar{x}_4 \vee \bar{x}_3x_4;$$

$$y(1, 0, x_3, x_4) = x_3\bar{x}_4 \vee \bar{x}_3x_4;$$

$$y(1, 1, x_3, x_4) = 0.$$

Рассмотрим любую конъюнкцию из функции (5.10), например $\bar{x}_1\bar{x}_2y(0, 0, x_3, x_4)$. В ней три буквы: x_1, x_2 и $y(0, 0, x_3, x_4)$, т. е. на одну букву меньше, чем в любой конъюнкции исходной функции $y(x_1, x_2, x_3, x_4)$. В выражении для $y(0, 0, x_3, x_4)$ две буквы, что также меньше, чем в любой конъюнкции $y(x_1, x_2, x_3, x_4)$. Таким образом, использование дизъюнктивного разложения позволяет выразить одни БФ через другие, зависящие от меньшего числа переменных.

Рассмотрим систему θ из N булевых функций L переменных, для которой $L^{\max} > s$, и покажем, как реализовать ее на ПЛМ (s, t, q) по методу из [21]. Изложение будем иллюстрировать на примере реализации системы

θ полностью определенных БФ, заданных табл. 5.18, на ПЛМ (5, 2, 15). Функции y_1 и y_2 из этой таблицы имеют следующий содержательный смысл: $y_1=0$, $y_2=1$, если и только если в двоичном семиразрядном коде, представленном значениями переменных x_1, \dots, x_7 , содержится ровно одна единица; $y_1=1$, $y_2=0$, если и только если в указанном коде содержатся ровно две единицы; $y_1=y_2=1$, тогда и только тогда, когда $x_1=\dots=x_7=0$ или $x_1=\dots=\dots=x_7=1$. Во всех остальных случаях $y_1=y_2=0$.

Выполним дизъюнктивное разложение функций из системы θ по i переменным, т. е. представим каждую из них в виде

$$\begin{aligned}
 y_n(x_1, \dots, x_L) = & \bar{x}_1 \dots \bar{x}_{i-1} \bar{x}_i y_n(0, \dots, 0, 0, x_{i+1}, \dots, x_L) \vee \\
 & \vee \bar{x}_1 \dots \bar{x}_{i-1} x_i y_n(0, \dots, 0, 1, x_{i+1}, \dots, x_L) \vee \\
 & \vee \bar{x}_1 \dots x_{i-1} \bar{x}_i y_n(0, \dots, 1, 0, x_{i+1}, \dots, x_L) \vee \\
 & \vee \bar{x}_1 \dots x_{i-1} x_i y_n(0, \dots, 1, 1, x_{i+1}, \dots, x_L) \vee \dots \vee \\
 & \vee x_1 \dots x_{i-1} x_i y_n(1, \dots, 1, 1, x_{i+1}, \dots, x_L).
 \end{aligned}$$

При выборе i необходимо стремиться к тому, чтобы число t^* используемых выходов ПЛМ (s, t, q) было бы как можно ближе к t . Если потребовать, чтобы при дизъюнктивном разложении функций по i переменным использовалось не менее двух выходов ПЛМ ($t^* \geq 2$), то должно выполняться неравенство

$$t^* = \left\lceil \frac{s-i}{2^i} \right\rceil \geq 2. \quad (5.11)$$

где $[a]$ — ближайшее целое число, не большее a . Из неравенства (5.11) можно получить

$$1 \leq i \leq [\log_2(s) - 1]. \quad (5.12)$$

Для нашего примера $s=5$, $t^*=2$, $i=1$, т. е. необходимо выполнять разложение функций рассматриваемой системы по одной переменной.

В [21] показано, что число уровней r в многоуровневой схеме, построенной при использовании метода дизъюнктивного разложения БФ, можно определить из выражения

$$r = \left\lceil \frac{L-s}{i} \right\rceil + 1. \quad (5.13)$$

В нашем примере $L=7$, $s=5$, $i=1$ и $r=3$. ПЛМ последних $r-1$ уровней, считая от входов схемы, могут быть запрограммированы единообразно. В этом случае на первые i входов подаются значения переменных x_{j1}, \dots, x_{ji} , по ко-

торым выполнено разложение. На остальные 2^{it^*} входов подаются значения коэффициентов разложения с выходов ПЛМ предшествующего уровня. Способ программирования ПЛМ при $t^*=2, i=2$ иллюстрируется рис. 5.2. Программирование ПЛМ первого уровня выполняется обычным способом после минимизации реализуемых ими БФ.

Общий алгоритм синтеза многоуровневой комбинационной схемы включает следующие этапы.

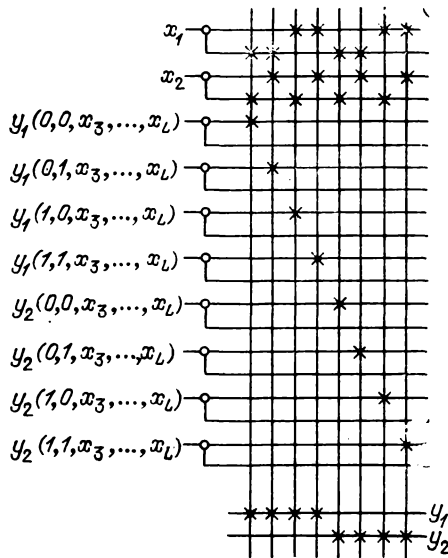


Рис. 5.2

1. По формулам (5.11)–(5.13) определяются значения t^* — число используемых выходов ПЛМ; i — число переменных, по которым выполняется разложение, и r — число уровней в схеме. Для нашего примера уже известно, что $t^*=2, i=1$ и $r=3$.

2. Значения каждой из функций системы θ сводятся в двумерную таблицу, у которой 2^{L-s} строк и 2^s столбцов. Каждая строка задает булеву функцию (коэффициент разложения), зависящую от s переменных: $y_n(\sigma_1, \dots, \sigma_{L-s}, x_{L-s+1}, \dots, x_L)$, где $\sigma_1, \dots, \sigma_{L-s}$ — константы, равные 0 или 1. Так, табл. 5.19 и 5.20 задают значения булевых функций y_1, y_2 на наборах значений переменных x_1, \dots, x_7 , указанных в табл. 5.18. Наборы значений переменных $x_1,$

Таблица 5.19

x_3	0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 1
x_4	0 0 0 0 1 0 0 0 0 1 1 1 1 0 0 1
x_5	0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1
x_6	0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1
x_7	0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 1 1
x_1 x_2	
0 0	1 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0
0 1	0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 0	0 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
1 1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

x_2 отмечают строки, а переменных x_3, \dots, x_7 — столбцы этих таблиц. Например, из табл. 5.19 можно записать $y(1, 1, x_3, x_4, x_5, x_6, x_7) = \bar{x}_3 \bar{x}_4 \bar{x}_5 \bar{x}_6 \bar{x}_7 \vee x_3 x_4 x_5 x_6 x_7$.

3. Для каждой из функций системы θ реализуются все коэффициенты разложения s переменных. В таблицах, соответствующих разным функциям системы θ , и даже в одной и той же таблице могут быть одинаковые или инверс-

Таблица 5.20

x_3	0 0 0 0 0 1 1 1 1 1 0 0 0 0 0 0 1
x_4	0 0 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1
x_5	0 0 0 1 0 0 0 0 1 0 0 0 1 1 1 0 1
x_6	0 0 1 0 0 0 0 1 0 0 0 1 0 0 1 1 1
x_7	0 1 0 0 0 0 1 0 0 0 1 0 0 1 0 1 1
x_1 x_2	
0 0	1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 0	1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

ные по отношению друг к другу строки. Наличие таких строк и строк, соответствующих константам и функциям, существенно зависящим от одной переменной, позволяет сократить число ПЛМ на первом уровне. В нашем примере на ПЛМ первого уровня необходимо реализовать следую-

щие функции (коэффициенты разложения):

$$y_1^{00} = y_1(0, 0, x_3, \dots, x_7); \quad y_2^{00} = y_2(0, 0, x_3, \dots, x_7);$$

$$y_1^{01} = y_1(0, 1, x_3, \dots, x_7); \quad y_2^{01} = y_2(0, 1, x_3, \dots, x_7);$$

$$y_1^{10} = y_1(1, 0, x_3, \dots, x_7); \quad y_2^{10} = y_2(1, 0, x_3, \dots, x_7);$$

$$y_1^{11} = y_1(1, 1, x_3, \dots, x_7); \quad y_2^{11} = y_2(1, 1, x_3, \dots, x_7).$$

Учитывая то, что $y_1^{10} = y_1^{01}$, $y_2^{10} = y_2^{01}$, значения y_1^{10} и y_1^{01} , y_2^{10} и y_2^{01} могут быть сформированы на одном выходе, в силу чего первый уровень схемы будет содержать три ПЛМ (5, 2, 15). Таблицы 5.21—5.23 представляют собой таблицы настройки этих ПЛМ.

Таблица 5.21

x_3	x_4	x_5	x_6	x_7	y_1^{00}	y_2^{11}
0	0	0	0	0	1	.
1	0	0	0	1	1	.
1	0	0	1	0	1	.
1	0	1	0	0	1	.
1	1	0	0	0	1	.
0	1	0	0	1	1	.
0	1	0	1	0	1	.
0	1	1	0	0	1	.
0	0	1	0	1	1	.
0	0	1	1	0	1	.
0	0	0	1	1	1	.
1	1	1	1	1	.	1

Таблица 5.22

x_3	x_4	x_5	x_6	x_7	$y_1^{01} = y_1^{10}$	y_1^{11}
0	0	0	0	0	.	1
0	0	0	0	1	1	.
0	0	0	1	0	1	.
0	0	1	0	0	1	.
0	1	0	0	0	1	.
0	1	0	0	0	1	.
1	0	0	0	0	1	.
1	1	1	1	1	.	1

Таблица 5.23

x_3	x_4	x_5	x_6	x_7	y_2^{00}	$y_2^{01} = y_2^{10}$
0	0	0	0	0	1	1
0	0	0	0	1	1	.
0	0	0	1	0	1	.
0	0	1	0	0	1	.
0	1	0	0	0	1	.
1	0	0	0	0	1	.

4. Назначается $j=0$.

5. Увеличивается j на единицу, т. е. $j := j+1$.

6. Реализуются булевы функции $s+j_i$ переменных. Для этого на первые i входов ПЛМ, реализующей t^* функций $s+j_i$ аргументов, подаются значения переменных, по которым выполняется разложение. Остальные 2^{i-t^*} входов со-

единяются с выходами ПЛМ, реализующих функции $s + (j-1)i$ переменных. В нашем примере при $j=1$ ПЛМ второго уровня программируются согласно табл. 5.24, 5.25. Например, первый вход ПЛМ₁ соответствует переменной x_2 , по которой выполнено разложение, на остальные входы ПЛМ₁ поступают значения коэффициентов разложения y_1^{00} , y_1^{01} , y_1^{10} , y_1^{11} с выходов ПЛМ первого уровня. Если учесть, что $y_1^{01}=y_1^{10}$, $y_2^{01}=y_2^{10}$, число использованных входных и промежуточных шин в этих ПЛМ может быть сокращено и их следует запрограммировать так, как это указано в табл. 5.26, 5.27.

Таблица 5.24

x_2	y_1^{00}	y_1^{01}	y_1^{10}	y_1^{11}	y_1^0	y_1^1
0	1	—	—	—	1	.
1	—	1	—	—	1	.
0	—	—	1	—	.	1
1	—	—	—	1	.	1

Таблица 5.25

x_2	y_2^{00}	y_2^{01}	y_2^{10}	y_2^{11}	y_2^0	y_2^1
0	1	—	—	—	1	.
1	—	1	—	—	1	.
0	—	—	1	—	.	1
1	—	—	—	1	.	1

Таблица 5.26

x_2	y_1^{00}	$y_1^{01}=y_1^{10}$	y_1^{11}	y_1^0	y_1^1
0	1	—	—	1	.
1	—	1	—	1	.
0	—	1	—	.	1
1	—	—	1	.	1

Таблица 5.27

x_2	y_2^{00}	$y_2^{01}=y_2^{10}$	y_2^{11}	y_2^0	y_2^1
0	1	—	—	1	.
1	—	1	—	1	.
0	—	1	—	.	1
1	—	—	1	.	1

7. Если $j < r-1$, переходят к п. 5, иначе — к п. 8. В нашем примере при $j=2$ единственная ПЛМ третьего уровня будет запрограммирована в соответствии с табл. 5.28.

8. Конец.

В тех случаях, когда при построении многоуровневой схемы в некоторой ПЛМ (s, t, q) будет недостаточно выходных или промежуточных шин, используются соответствующие тривиальные методы расширения ПЛМ.

Таблица 5.28

x_1	y^0	y^1	y_2^0	y_2^1	y_1	y_2
0	1	—	—	—	1	.
1	—	1	—	—	1	.
0	—	—	1	—	.	1
1	—	—	—	1	.	1

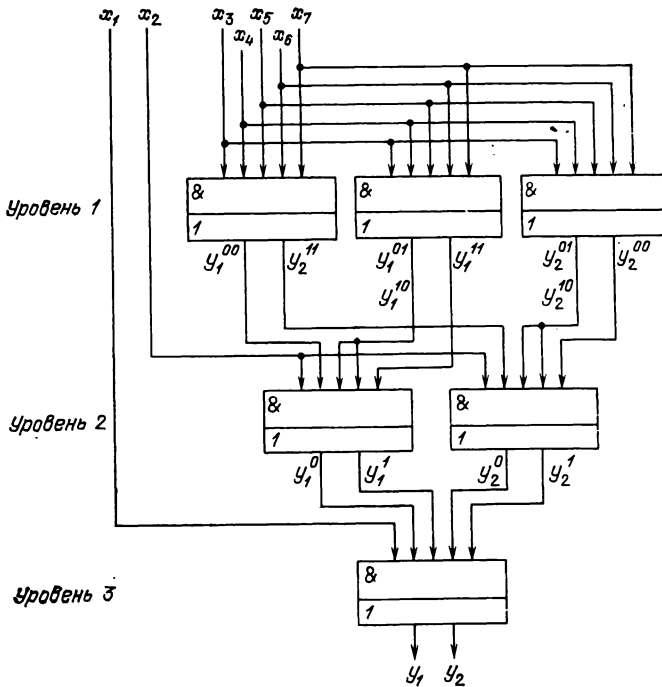


Рис. 5.3

Продолжим рассмотрение примера реализации на ПЛМ (5, 2, 15) системы функций, заданных табл. 5.18. Напомним, что программирование трех ПЛМ первого уровня выполнено в табл. 5.21—5.23, двух ПЛМ второго уровня — в табл. 5.24, 5.25 и одной ПЛМ третьего уровня — в табл. 5.28. На рис. 5.3 изображена логическая схема, реализующая заданную систему булевых функций. В этой схеме:

$$y_1^0 = y_1(0, x_2, \dots, x_7); \quad y_2^0 = y_2(0, x_2, \dots, x_7);$$

$$y_1^1 = y_1(1, x_2, \dots, x_7); \quad y_2^1 = y_2(1, x_2, \dots, x_7).$$

Поскольку $y_1^{10} = y_1^{01}$, $y_2^{10} = y_2^{01}$, оказывается возможным сократить в схеме на рис. 5.3 число уровней и ПЛМ (5, 2, 15) на единицу. Построенная для этого случая логическая схема показана на рис. 5.4. Таблицы 5.29 и 5.30 содержат информацию для программирования ПЛМ второго уровня в схеме на рис. 5.4.

Заметим, что ПЛМ последних $r-1$ уровней в многоуровневой схеме реализуют функции мультиплексирования. В § 4.2 показаны различные способы построения мультиплексоров на программируемых логических устройствах (ПЛУ). Соответствующим образом настроенные ПЛУ могут использоваться вместо ПЛМ на последних $r-1$ уровнях в рассматриваемых многоуровневых схемах.

Таблица 5.29

x_1	x_2	y_1^{00}	$y_1^{01}=y_1^{10}$	y_1^{11}	y_1
0	0	1	—	—	1
0	1	—	1	—	1
1	0	—	1	—	1
1	1	—	—	1	1

Таблица 5.30

x_1	x_2	y_2^{00}	$y_2^{01}=y_2^{10}$	y_2^{11}	y_2
0	0	1	—	—	1
0	1	—	1	—	1
1	0	—	1	—	1
1	1	—	—	1	1

Если в качестве элементов многоуровневой схемы должны быть использованы ПЗУ (s, t), описанный выше метод практически не меняется.

Качество решения задачи синтеза комбинационной схемы рассмотренным методом в значительной степени определяется выбором переменных, по которым выполняется

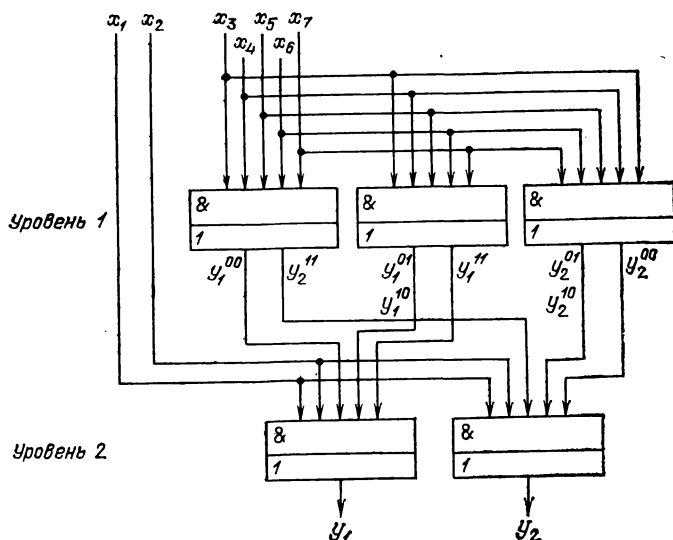


Рис. 5.4

разложение. Для различных вариантов выбора переменных может быть получено различное число совпадающих (в том числе с точностью до инверсии), константных, либо существенно зависящих от одной переменной коэффициентов разложения. Реализация таких коэффициентов может быть выполнена на различном числе элементов. Вполне очевидно, что лучшей будет та реализация, которая требует меньшего числа элементов. Точное решение этой задачи связано с существенным перебором. Для получения приближенного решения может быть проанализировано несколько вариантов разложения по различным переменным и среди них выбран наилучший. При этом чем больше вариантов будет проанализировано, тем качественнее получится решение, однако затраченное время также будет возрастать. Другими словами, за выигрыш в качестве решения необходимо платить проигрышем во времени.

Рассмотрим еще один довольно простой метод построения многоуровневых схем, изложенный в [1]. Пусть некоторая система θ булевых функций, заданных в совершенной ДНФ, представлена в табличном виде. В этом случае во всех столбцах x_1, \dots, x_L таблицы будут содержаться лишь нули и единицы, а любые две строки для столбцов x_1, \dots, x_L будут взаимно ортогональны. Поделим столбцы x_1, \dots, x_L вертикальными линиями на полосы $\omega_1, \dots, \omega_k$ в каждой из которых, кроме, быть может, последней, содержится ровно s столбцов и один и тот же столбец не может быть включен в различные полосы. Другими словами, если X_1, \dots, X_k — множества переменных, приписанных столбцам полос $\omega_1, \dots, \omega_k$, то $\pi_x = \{X_1, \dots, X_k\}$ — разбиение на множестве $X = \{x_1, \dots, x_L\}$ и $K = \lfloor L/s \rfloor$.

Обозначим через n_i число различных строк в i -й полосе, тогда, закодировав указанные строки различными двоичными кодами, получим минимальную длину кода $l_i = \lceil \log_2 n_i \rceil$, и если $l_i < s$, то s выделенных переменных рассматриваемых функций можно заменить меньшим числом l_i переменных. Запишем вместо строк в каждой полосе их коды, в результате чего получим новую таблицу, представляющую ту же самую систему БФ. Если для полученной

(новой) таблицы выполняется условие $L_n = \sum_{i=1}^K l_i \leq s$, то

функции системы θ реализуются на ПЛМ с помощью методов, изложенных в § 5.2. В тех случаях, когда $L_n > s$, рассмотренные действия повторяются. Если на некотором шаге повторения число столбцов во вновь полученной таб-

лице не меньше, чем в предыдущей, то рассматриваемая задача указанным методом не может быть решена. Таким образом, либо на некотором шаге задача будет решена, либо фиксируется факт невозможности ее решения.

Качество решения рассматриваемой задачи определяется тем выше, чем больше совпадающих строк будет в каждой выделенной полосе. В этом случае сокращаются зна-

Таблица 5.31

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	y_1	y_2	y_3
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
0	0	0	0	0	0	0	0	0	0	0	1	0	1	1
0	0	0	0	0	0	0	0	0	0	1	0	0	1	1
0	0	0	0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	0	0	0	1	0	0	0	0	1	1
0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	0	1	0	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	1	0	0	0	0	0	0	0	0	0	1	1
0	0	1	0	0	0	0	0	0	0	0	0	0	1	1
0	1	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1	1	1	0	0	1

чения n_i и l_i , что способствует минимизации величины L_n . Решение этой задачи связано с нахождением «хорошего» разбиения π_x и требует существенного перебора различных вариантов. Для приближенного нахождения разбиения π_x может быть применен тот же прием, что и при выборе переменных, по которым производится разложение в предыдущем методе.

Рассмотрим пример реализации системы θ булевых функций, заданных табл. 5.31, на ПЛМ (6, 3, 8). Штриховой линией столбцы x_1, \dots, x_{12} в табл. 5.31 поделены на две полосы и $\pi_x = x_1, x_2, x_3, x_4, x_5, x_6; x_7, x_8, x_9, x_{10}, x_{11}, x_{12}$. В каждой полосе будет по восемь различных строк, для кодирования которых может быть использован трехразрядный код. Таблицы 5.32 и 5.33 представляют собой таблицы, в которых выполнено кодирование различных строк в каждой полосе. Фактически табл. 5.32 и 5.33 задают систему

булевых функций z_1, \dots, z_6 , которые необходимо реализовать на ПЛМ (6, 3, 8) первого уровня. Ясно, что для реализации этих функций потребуются две ПЛМ (6, 3, 8). Таблица 5.34 представляет собой таблицу, в которой выходные переменные y_1, \dots, y_3 являются функциями кодирующих переменных z_1, \dots, z_6 . Таблица 5.35 задает те же самые функции (см. табл. 5.34) после их минимизации.

Таблица 5.32

x_1	x_2	x_3	x_4	x_5	x_6	z_1	z_2	z_3
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	1
1	0	0	0	0	0	1	1	0
1	1	1	1	1	1	1	1	1

Таблица 5.33

x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	z_4	z_5	z_6
0	0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0	1
0	0	0	0	1	0	0	1	0
0	0	0	1	0	0	0	1	1
0	0	1	0	0	0	1	0	0
0	1	0	0	0	0	1	0	1
1	0	0	0	0	0	1	1	0
1	1	1	1	1	1	1	1	1

Функции табл. 5.35 могут быть реализованы на одной ПЛМ (6, 3, 8) второго уровня ($r=2$). На рис. 5.5 показана многоуровневая логическая схема, реализующая БФ, заданные табл. 5.31. Программирование ПЛМ второго уровня выполнено в табл. 5.36, а ПЛМ первого уровня — в табл. 5.37 и 5.38. Как и ранее, если бы в некоторых ПЛМ не хва-

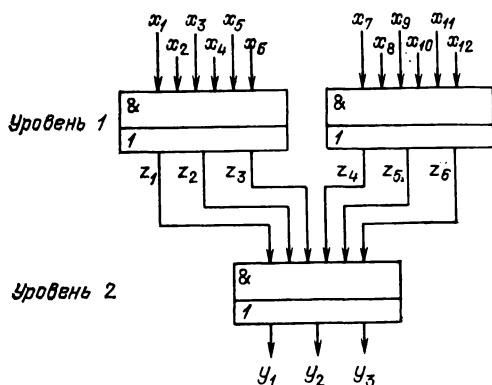


Рис. 5.5

Таблица 5.34

z_1	z_2	z_3	z_4	z_5	z_6	y_1	y_2	y_3
0	0	0	0	0	0	1	1	0
0	0	0	0	0	1	0	1	1
0	0	0	0	1	0	0	1	1
0	0	0	0	1	1	0	1	1
0	0	0	1	0	0	0	1	1
0	0	0	1	0	1	0	1	1
0	0	0	1	1	0	0	1	1
0	0	1	0	0	0	0	1	1
0	1	0	0	0	0	0	1	1
0	1	1	0	0	0	0	1	1
1	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	1	1
1	1	0	0	0	0	0	1	1
1	1	1	1	1	1	0	0	1

Таблица 5.53

z_1	z_2	z_3	z_4	z_5	z_6	y_1	y_2	y_3
0	0	0	0	0	0	1	1	0
0	0	0	1	0	—	0	1	1
0	0	0	0	—	1	0	1	1
0	0	0	—	1	0	0	1	1
1	0	—	0	0	0	0	1	1
0	—	1	0	0	0	0	1	1
—	1	0	0	0	0	0	1	1
1	1	1	1	1	1	0	0	1

тало выходных или промежуточных шин, то могли бы быть использованы тривиальные методы расширения, описанные в § 5.1.

Таблица 5.36

z_1	z_2	z_3	z_4	z_5	z_6	y_1	y_2	y_3
0	0	0	0	0	0	1	1	.
0	0	0	1	0	—	.	1	1
0	0	0	0	—	1	.	1	1
0	0	0	—	1	0	.	1	1
1	0	—	0	0	0	.	1	1
0	—	1	0	0	0	.	1	1
—	1	0	0	0	0	.	1	1
1	1	1	1	1	1	.	.	1

Допустим теперь, что в некоторых строках столбцов x_1, \dots, x_L таблицы, задающей систему θ , содержатся прочерки (—), т. е. соответствующие функции представлены в произвольной ДНФ. Пусть любая пара рассматриваемых строк ортогональна, более того, предположим, что в выделенных полосах любая пара строк также ортогональна (исключение могут составлять лишь те пары строк, которые оказываются равными). Если перечисленные

условия будут выполнены, то изложенный метод декомпозиции полностью остается в силе [1]. В тех случаях, когда сформулированные условия не выполняются, можно расщеплять строки в каждой полосе так, чтобы после расщепления все строки были взаимно ортогональны, а общее число различных строк было по возможности минимально. Описанный метод декомпозиции носит название метода ортогонализации интервалов и подробно рассмотрен в [1].

Таблица 5.37

x_1	x_2	x_3	x_4	x_5	x_6	z_1	z_2	z_3
0	0	0	0	0	0	.	.	.
0	0	0	0	0	1	.	.	1
0	0	0	0	1	0	.	1	.
0	0	0	1	0	0	.	1	1
0	0	1	0	0	0	1	.	.
0	1	0	0	0	0	1	.	1
1	0	0	0	0	0	1	1	.
1	1	1	1	1	1	1	1	1

Таблица 5.38

x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	z_4	z_5	z_6
0	0	0	0	0	0	.	.	.
0	0	0	0	0	1	.	.	1
0	0	0	0	1	0	.	1	.
0	0	0	1	0	0	.	1	1
0	0	1	0	0	0	1	.	.
0	1	0	0	0	0	1	.	1
1	0	0	0	0	0	1	1	.
1	1	1	1	1	1	1	1	1

5.4. Краткий обзор методов синтеза комбинационных схем из ПЛМ

Среди ПЛУ различных типов в качестве базовых элементов комбинационных схем наиболее часто используются ПЛМ, что объясняется их большой распространенностью, серийным промышленным выпуском и широкой номенклатурой. Известные методы синтеза комбинационных схем из ПЛМ можно разбить на две группы. В первую включаются те, в которых выполняется ограничение $L^{\max} \leq s$, а во вторую — в которых оно не выполняется. Первые [1, 17, 19, 23, 25] близки по своей сущности к методам решения задач компоновки на этапе технического (конструкторского) проектирования дискретных устройств [28].

Пусть E , X и Y — множества элементарных конъюнкций, входных и выходных переменных реализуемой системы θ булевых функций. Тогда задача синтеза может быть сведена к задаче компоновки элементов этих множеств в блоки $E^1, \dots, E^U, X^1, \dots, X^U, Y^1, \dots, Y^U$, такие что $\pi_E = \{E^1, \dots, E^U\}$, $\pi_X = \{X^1, \dots, X^U\}$ и $\pi_Y = \{Y^1, \dots, Y^U\}$ — разбиения или покрытия множеств E , X и Y соответственно. Заметим, что если π_Y — покрытие множества Y , то некоторые выходы различных ПЛМ будут объединяться по схеме ИЛИ. ДНФ функций из множества Y^u ($u=1, \dots, U$) включает лишь элементарные конъюнкции из множества E^u переменных из множества X^u , причем $|Y^u| \leq t$, $|E^u| \leq q$, $|X^u| \leq s$. Тогда число ПЛМ в комбинационной схеме равно U , а любая функция из множества Y^u реализуется (возможно частично) на u -й ПЛМ.

Методы второй группы [1, 21, 22, 24, 26, 27] основаны на использовании общего декомпозиционного подхода и некоторых его частных модификаций, таких как дизъюнктивное разложение булевых функций [21], факторизация [27] и др. Заметим, что представленное деление на две группы сделано для того, чтобы подчеркнуть специфику реализации различных методов, в то время как все они в конечном счете являются декомпозиционными.

Изложим основные принципы, лежащие в основе каждого из указанных методов [1, 17, 19, 20, 22—27]. В [17, 19, 20] решается задача построения разбиений либо покрытий π_E , π_x и π_y с минимальным числом блоков. Для этого предложены последовательные и параллельные алгоритмы. В первых блоки разбиений формируются последовательно. Для выбора очередного элемента, включаемого в блок, вводятся специальные оценки, аналогичные тем, которые рассматривались в § 5.2; в параллельных алгоритмах все U блоков заполняются параллельно.

В [22] предлагается двухуровневая реализация комбинационных схем (при $L > s$, $N > t$, $B > q$) на ПЛМ (s, t, q) , основанная на факторизационном подходе. Метод [22] наиболее эффективен для систем слабо определенных БФ. В [27] аналогичная задача решается с помощью предлагаемого метода дизъюнктивно-конъюнктивного разложения БФ, являющегося частным случаем задачи декомпозиции. Этот метод основан на многократном применении трех процедур: минимизации числа переменных реализуемых БФ (последние могут быть заданы в произвольной ДНФ), функционального разложения БФ — и их минимизации в классе ДНФ. В [25] рассматривается задача реализации системы θ на ПЛМ (s, t, q) при выполнении условий $L \leq s$, $B \leq q$, $N > t$. Предложены три приближенных конкурирующих алгоритма, позволяющих разбить систему θ на подсистемы, каждая из которых реализуется на одной ПЛМ, и сумма различных элементарных конъюнкций в каждой подсистеме после минимизации минимальна. В [24] рассматривается задача реализации на ПЛМ (s, t, q) системы θ булевых функций, заданных в совершенной ДНФ, при выполнении условия $L > s$. Решение этой задачи сводится к построению декомпозиции вида $\theta = \Phi[\Psi(A), B]$, где θ — система функций y_1, \dots, y_N множества переменных $X = \{x_1, \dots, x_L\}$, $L > s$, $N > 1$, $|A| = s$, $A \cup B = X$; $\Phi = \{\varphi_1, \dots, \varphi_N\}$ — система функций, реализующих исходную систему θ ; $\Psi(A) = \{\psi_1, \dots, \psi_p\}$ — система подфункций такая, что $p \leq t$. По результатам декомпозиции строится много-

уровневая схема, общая структура которой изображена на рис. 5.6. На каждом уровне схемы реализуется декомпозиция рассмотренного вида. Для ПЛМ (s, t, q) r -го уровня выполняется ограничение

$$|A_r| + |B_r| \leq s. \quad (5.14)$$

На выходах этих ПЛМ формируются значения функций y_1, \dots, y_N . Число уровней r определяется необходимым числом итераций для выполнения ограничения (5.14). В от-

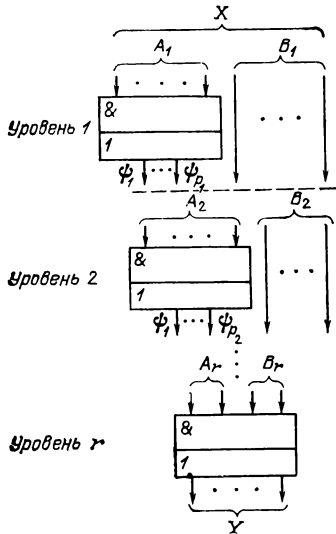


Рис. 5.6

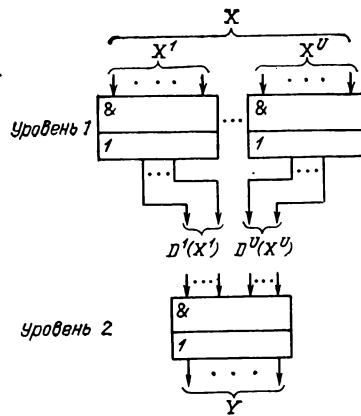


Рис. 5.7

личие от ранее перечисленных работ, в [24] детально исследуются вопросы выбора «хорошего» разбиения $\{A_i, B_i\}$ для декомпозиции и предлагается соответствующая формализованная процедура. В [26] предложен метод, аналогичный рассмотренному в [1] (см. § 5.3), который основан на декомпозиции векторной БФ следующего вида:

$$\theta = G [D^1(X^1), \dots, D^u(X^u)], \quad (5.15)$$

причем число функций в любом множестве D^u меньше, чем число переменных в множестве X^u . Другими словами, в (5.15) реализуется замена переменных из множества X^u меньшим числом переменных из множества D^u ($u = \overline{1, U}$) и исходные функции системы θ заменяются функциями су-

щественно меньшего числа переменных. По результатам декомпозиции строится двухуровневая схема, общая структура которой изображена на рис. 5.7. Качество декомпозиции оценивается двумя критериями, главный из которых заключается в минимизации числа выходов из элементов первого уровня, а второстепенный — в минимизации числа элементарных конъюнкций, реализуемых каждой ПЛМ.

В [23] предлагается реализовать систему θ на ПЛМ (s, t, q) при $L > s, N > t, B > q$ схемой, структура которой приведена на рис. 5.8. Подсхема предварительной обработ-

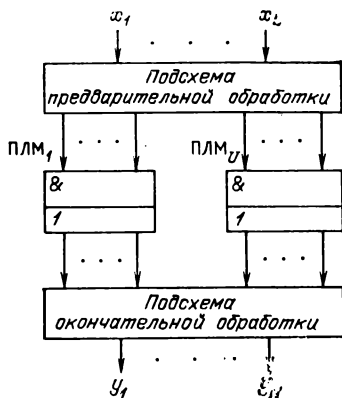


Рис. 5.8

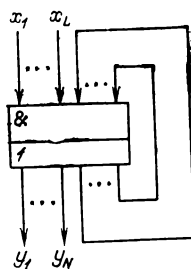


Рис. 5.9

ки служит для перекодирования значений входных переменных и распределения последних между ПЛМ₁, ..., ПЛМ_U. Подсхема окончательной обработки предназначена для преобразования сигналов на выходах ПЛМ₁, ..., ПЛМ_U в требуемые сигналы, определяющие значения выходных переменных y_1, \dots, y_N . Для построения первой и второй подсхем могут быть использованы любые логические элементы, в том числе малой и средней степеней интеграции. Заметим, что на целесообразность использования таких элементов указывается и в других работах, например в [27]. Далее в [23] довольно подробно рассматривается общий алгоритм синтеза комбинационной схемы θ , позволяющий минимизировать число U ПЛМ (s, t, q) . Процедура синтеза сводится к решению различных задач покрытия множеств. Указывается, что предлагаемый эвристический подход обладает высокой эффективностью и гарантирует получение приближенного решения с разумными вычислительными затратами. Описаны две методики формирования

максимальных классов совместимости. Первая сводится к постепенному увеличению числа элементов в каждом классе с проверкой на каждой итерации заданных ограничений, вторая — к формированию максимальных классов и удалению из них элементов до удовлетворения заданных ограничений. Большое число методов синтеза комбинационных схем из ПЛМ (s, t, q) , сводящихся к операциям над булевыми и троичными матрицами, описано в [1]; некоторые из них были описаны ранее в § 5.3.

Кроме рассмотренных методов при нетривиальной реализации заданной системы θ на стандартных ПЛМ (s, t, q) могут быть использованы следующие частные приемы. Так, если для заданной системы θ и ПЛМ (s, t, q) выполняются условия $L \leq s, N \leq t, B > q$, то следует произвести совместную минимизацию ДНФ заданных БФ, например, с помощью методов, изложенных в [1]. После этого может оказаться, что $L \leq s, N \leq t, B \leq q$ и система θ тривиально реализуется на одной ПЛМ (s, t, q) . Если в результате минимизации по-прежнему $B > q$, но $L < s, N < t$, то можно попытаться представить схему так, как это показано на рис. 5.9. В этом случае на одной ПЛМ (s, t, q) реализуется многоуровневая комбинационная схема и число задействованных промежуточных шин меньше, чем у одноуровневой (не превышает q). Примеры таких схем приведены в гл. 4 (см. рис. 4.13, 4.15—4.17). Аналогичный прием может быть использован и при построении комбинационной схемы на нескольких ПЛМ (s, t, q) , например при $L < s, N > t, B > q$. Для этих целей используются различные методы декомпозиции БФ.

Если для заданной системы θ и ПЛМ (s, t, q) выполняются условия $L > s, N \leq t, B \leq q$, то предварительно может быть выполнена минимизация числа переменных¹ во входящих в θ булевых функциях [1]. После этого может оказаться, что $L \leq s, N \leq t, B \leq q$ и система θ тривиально реализуется на одной ПЛМ (s, t, q) .

Если $L \leq s, N > t, B \leq q$, то предварительно может быть выполнена минимизация числа функций в системе θ . Для этих целей используется тот же метод, что и для минимизации числа микроопераций в микропрограмме, рассмотренный в работе [29]. После этого может оказаться, что $L \leq s, N \leq t, B \leq q$ и система θ реализуется на одной ПЛМ (s, t, q) .

¹ Эта задача обычно рассматривается применительно к неполностью определенным булевым функциям.

Таблица 5.39

Соотношение параметров	Используемый метод синтеза
$L \leq s, N \leq t, B \leq q$	Тривиальная реализация (§ 5.1)
$L \leq s, N \leq t, B > q$	Минимизация системы булевых функций [1] с последующей тривиальной реализацией. Расширение ПЛМ по промежуточным шинам (§ 5.1). Декомпозиция булевых функций и построение комбинационной схемы на ПЛМ с обратными связями от выходов к входам. Использование методов из [1, 17, 19] и § 5.2
$L \leq s, N > t, B \leq q$	Использование метода, изложенного в [29], с последующей тривиальной реализацией. Расширение ПЛМ по выходам (§ 5.1). Использование методов из [1, 17, 19, 20, 25] и § 5.2
$L \leq s, N > t, B > q$	Указанная задача сводится к двум предыдущим
$L > s$	Минимизация числа переменных заданной системы булевых функций [1] и сведение задачи к предыдущей
$L > s, L^{\max} \leq s, N > t, B > q$	Использование методов, изложенных в § 5.2 и в [1, 17, 19, 20]
$L > s, L^{\max} > s, N > t, B > q$	Использование методов, изложенных в § 5.3 и в [1, 21—24, 26, 27]

В заключение отметим, что для решения каждой конкретной задачи обычно может использоваться несколько различных методов, обеспечивая приблизительно одинаковое качество решения. Правила выбора конкретных методов в зависимости от соотношения параметров реализуемой системы БФ и используемых ПЛМ (s, t, q) сведены в табл. 5.39.

**СИНТЕЗ АВТОМАТОВ
НА ПРОГРАММИРУЕМЫХ
ЛОГИЧЕСКИХ МАТРИЦАХ
КОМБИНАЦИОННОГО ТИПА**

6.1. Тривиальная реализация

Микропрограммный автомат (МПА) может быть тривиально реализован на одной ПЛМ (s, t, q), если выполняются следующие условия: $(L+R) \leq s$, $(N+R) \leq t$, $B \leq q$, где L, N и R — соответственно число входов, выходов и элементов памяти МПА; B — число строк в таблице переходов или в структурной таблице МПА. При этом в матрице M_1 ПЛМ (s, t, q) формируется множество элементарных конъюнкций переменных $T_1, \dots, T_R, x_1, \dots, x_L$, соответствующих строкам структурной таблицы МПА, а матрицей M_2 реализуется множество функций выходов и функций возбуждения памяти автомата. Так, если в строке h структурной таблицы МПА записан переход из состояния a_m в состояние a_s под действием входного сигнала $X(a_m, a_s)$, то этому переходу соответствует промежуточная шина ПЛМ (s, t, q), на которой реализована конъюнкция $e_h = a_m X(a_m, a_s)$, где a_m — конъюнкция, соответствующая коду $K(a_m)$ состояния a_m . Пусть в структурной таблице микрооперация y_n записана в строках с номерами n_1, \dots, n_U , а функция возбуждения D_r — в строках с номерами r_1, \dots, r_Q , тогда

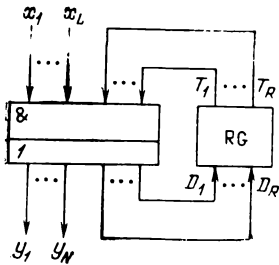


Рис. 6.1

соответствует промежуточная шина ПЛМ (s, t, q), на которой реализована конъюнкция $e_h = a_m X(a_m, a_s)$, где a_m — конъюнкция, соответствующая коду $K(a_m)$ состояния a_m . Пусть в структурной таблице микрооперация y_n записана в строках с номерами n_1, \dots, n_U , а функция возбуждения D_r — в строках с номерами r_1, \dots, r_Q , тогда

возбуждения D_r — в строках с номерами r_1, \dots, r_Q , тогда

$$y_n = e_{n_1} \vee \dots \vee e_{n_U}; \quad D_r = e_{r_1} \vee \dots \vee e_{r_Q}.$$

Схема МПА при его тривиальной реализации на ПЛМ (s, t, q) показана на рис. 6.1. Память автомата построена на R -разрядном регистре RG из D -триггеров, комбинационная часть — на ПЛМ (s, t, q). Для примера построим автомат Мили, заданный табл. 2.10, на ПЛМ (7, 10, 15). Таблица настройки ПЛМ показана в табл. 6.1. Каждая строка табл. 6.1 соответствует одной строке структурной таблицы МПА (см. табл. 2.10). Однако терм, соответству-

ющий строке 7 табл. 2.10, в ПЛМ отсутствует, так как все компоненты выходных функций и функций возбуждения на переходе в строке 7 равны нулю.

Укажем некоторые отличия в программировании ПЛМ для МПА Мура. В МПА Мура выходные шины y_1, \dots, y_N матрицы M_2 можно программировать двумя способами. Обозначим через $E(a_m)$ множество строк структурной таблицы, для которых исходным является состояние a_m . На-

Таблица 6.1

T_1	T_2	T_3	x_1	x_2	x_3	x_4	y_1	y_2	y_3	y_4	y_5	y_6	D_1	D_2	D_3
0	0	0	—	—	—	—	1	.	1	1
0	0	1	1	0	0	—	1
0	0	1	1	1	—	—	.	1	1	1	.
0	0	1	1	0	1	—	.	.	.	1	.	.	.	1	1
0	0	1	0	—	—	—	.	1	1	.	.
0	1	0	—	—	—	—	1	.	.	1	.	.	.	1	1
0	1	1	—	1	—	—	1	.	1	.	1
1	0	0	0	—	—	1	1	.	.	.
1	0	0	—	—	—	0	1	.	.	1	.	.	.	1	1
1	0	0	1	—	—	1	1	.	.
1	0	1	—	—	—	—	1	.	.	.

пример, $E(a_2)$ содержит 2-ю—5-ю строки табл. 2.11. Первый способ программирования заключается в следующем. Если в структурной таблице МПА Мура переменная y_k записана в столбце a_m рядом с некоторым состоянием a_i , то выходная шина y_k соединяется посредством ЭОП со всеми промежуточными шинами, на которых реализованы конъюнкции для строк из множества $E(a_i)$. Первый способ применим лишь тогда, когда для каждой строки из множества $E(a_i)$ выделена промежуточная шина, на которой реализована элементарная конъюнкция, соответствующая этой строке. Второй способ программирования заключается в том, что на дополнительной промежуточной шине реализуется элементарная конъюнкция переменных T_1, \dots, T_R , равная единице тогда и только тогда, когда МПА находится в состоянии a_i . Это промежуточная шина соединяется посредством ЭОП с выходной шиной y_k .

Первый способ программирования эффективнее второго в тех случаях, когда в каждой строке из множества $E(a_i)$ в столбце $F(a_m, a_s)$ записана хотя бы одна переменная из

множества $\{D_1, \dots, D_R\}$. Второй способ программирования эффективнее первого в тех случаях, когда в множестве $E(a_i)$ имеются две строки или более, для которых в столбце $F(a_m, a_s)$ не записаны никакие переменные. В этом случае при программировании ПЛМ по первому способу необходимо использовать дополнительно две или более промежуточных шин, а по второму — лишь одну промежуточную шину. Если в множестве $E(a_i)$ имеется лишь одна строка, в которой в столбце $F(a_m, a_s)$ не записаны никакие переменные, то могут быть применены как первый, так и второй способы программирования ПЛМ; в обоих случаях по отношению к числу промежуточных шин, используемых в ПЛМ для формирования функций возбуждения, добавляется одна шина. Для примера построим автомат Мура, заданный табл. 2.11, на ПЛМ (7, 10, 15). Таблица настройки ПЛМ показана в табл. 6.2.

Таблица 6.2

T_1	T_2	T_3	x_1	x_2	x_3	x_4	D_1	D_2	D_3	y_1	y_2	y_3	y_4	y_5	y_6
0	0	0	—	—	—	—	.	.	1
0	0	1	1	0	0	—	.	.	1	1	.	1	.	.	.
0	0	1	1	1	—	—	.	1	.	1	.	1	.	.	.
0	0	1	1	0	1	—	.	1	1	1	.	1	.	.	.
0	0	1	0	—	—	—	1	.	1	1	.	1	.	.	.
0	1	0	—	—	—	—	1	.	.	.	1	1	.	.	.
0	1	1	—	1	—	—	1	1
1	0	0	—	1	—	—	1	1
1	0	1	—	—	—	0	1	.	.	.	1
1	0	1	1	—	—	1	1	.	1	.	1
1	0	1	0	—	—	1	1	1	1	.	1
1	1	0	—	—	—	—	1	1	1	1	.
1	1	1	—	—	—	—	.	1	1
0	1	1	—	—	—	—	1	.	.
1	0	0	—	—	—	—	.	.	.	1	.	.	1	.	.

При синтезе МПА на ПЛМ (z, q) его тривиальная реализация на одной ПЛМ возможна при выполнении следующих условий: $B \leq q$, $(N+L+2R) \leq z$. После распределения переменных $x_1, \dots, x_L, T_1, \dots, T_R, y_1, \dots, y_N, D_1, \dots, D_R$ между z -шинами ПЛМ (z, q) она программируется так же, как и ПЛМ (s, t, q) .

Если выполняются условия $(L+R) \leq s, B \leq q, (N+R) > t$, т. е. выходов ПЛМ (s, t, q) не хватает для распределения

на них всех переменных $y_1, \dots, y_N, D_1, \dots, D_R$, то автомат может быть реализован на нескольких ПЛМ (s, t, q) за счет их расширения по выходам (см. § 5.1). Когда выполняются условия $(L+R) \leq s, (N+R) \leq t, B > q$, т. е. промежуточных шин в ПЛМ (s, t, q) не хватает, то автомат также может быть реализован на нескольких ПЛМ за счет их расширения по промежуточным шинам (см. § 5.1). Если $(L+R) \leq s, (N+R) > t, B > q$, т. е. в ПЛМ (s, t, q) не хватает как выходных, так и промежуточных шин, то может быть выполнено одновременное расширение ПЛМ по выходам и промежуточным шинам (см. § 5.1). Заметим, что такая тривиальная реализация обычно является в значительной степени избыточной. Применяя специальные приемы, рассмотренные в гл. 5, можно сократить число элементов в схеме МПА. Если для некоторого автомата и ПЛМ (s, t, q) выполняется условие $(L+R) > s$, то реализовать его тривиальным способом не удастся и должны использоваться специальные декомпозиционные методы.

Тривиальные методы построения схемы МПА на ПЛМ (s, t, q) , как правило, приводят к получению схемы автомата, далекой от оптимальной. В последующих параграфах настоящей главы рассматриваются автоматы, которые не могут быть построены на одной ПЛМ, и описываются различные методы их нетривиальной реализации.

6.2. Одноуровневая реализация автомата на ПЛМ

Синтез автомата на ПЛМ (s, t, q) при нетривиальной реализации сводится к нахождению некоторой декомпозиции его комбинационной схемы (КС) и к построению сети из ПЛМ. В зависимости от метода декомпозиции сеть может иметь различную конфигурацию. Практические соображения по унификации конструкции и топологии типовых элементов замены (ТЭЗ) проектируемого МПА накладывают определенные ограничения на структуру сети и предъявляют к ней требования регулярности. Последнее понимается в смысле повторяемости основных групп связей в логических схемах регулярных МПА. Повышение степени регулярности схемы автомата способствует упрощению разработки конструкции ТЭЗ микропрограммного автомата, поскольку топология ТЭЗ изменяется несущественно при переходе от одной схемы к другой. При размещении схемы автомата, состоящей из нескольких ПЛМ на одном кристалле БИС, требование регулярности также существенно [30], поскольку позволяет значительно упростить разра-

ботку топологии БИС и свести к минимуму необходимые изменения в фотошаблонах при настройке ПЛМ.

Один из подходов к построению регулярных сетей из ПЛМ состоит в следующем. При синтезе автомата накладываются определенные ограничения на структуру его будущей схемы и на способ размещения основных связей между ее элементами, причем введение указанных ограничений приводит к повышению регулярности схемы МПА. Снижение оптимальности схемы за счет роста числа со-

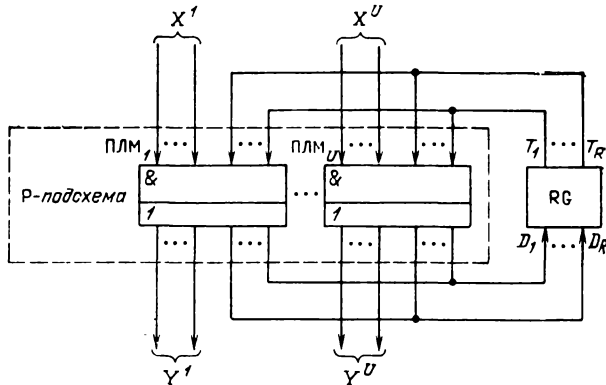


Рис. 6.2

ставляющих ее элементов, что, кстати, не всегда имеет место, компенсируется снижением трудозатрат на этапе конструкторского проектирования и сокращением сроков проектирования.

Один из классов схем, структура которых близка к регулярной, составляют одноуровневые схемы МПА на ПЛМ (рис. 6.2), рассмотренные в [17]. В них сигналы от входов x_1, \dots, x_L или T_1, \dots, T_R комбинационной схемы МПА проходят на ее выходы y_1, \dots, y_N или D_1, \dots, D_R не более, чем через один уровень ПЛМ (не более, чем через одну ПЛМ). Схему, обведенную штриховой линией на рис. 6.2, в дальнейшем будем называть Р-подсхемой. Она построена на ПЛМ₁, ..., ПЛМ_U, причем на входы каждой u -й ПЛМ поступают значения всех переменных обратной связи T_1, \dots, T_R и переменных из множества $X^u \subseteq X = \{x_1, \dots, x_L\}$, $X^1 \cup \dots \cup X^U = X$, а на выходах каждой u -й ПЛМ формируются значения переменных из множества $D^u \subseteq D = \{D_1, \dots, D_R\}$, $D^1 \cup \dots \cup D^U = D$ и из множества $Y^u \subseteq Y = \{y_1, \dots, y_N\}$, $Y^1 \cup \dots \cup Y^U = Y$, $u = \overline{1, U}$.

Если входная переменная или переменная обратной связи связана со входами различных ПЛМ, то будем говорить, что происходит дублирование входной переменной или переменной обратной связи. Когда все значения некоторой выходной переменной или функции возбуждения не могут быть сформированы на выходе одной ПЛМ, будем говорить, что по отношению к ПЛМ происходит дублирование выходной переменной или функции возбуждения. Если на промежуточных шинах различных ПЛМ реализована одна и та же элементарная конъюнкция (терм), то будем говорить о дублировании термов в ПЛМ. В частности, в схеме, изображенной на рис. 6.2, дублируются переменные обратной связи и могут дублироваться входные и выходные переменные, функции возбуждения и термы.

6.3. Синтез МПА по покрытиям структурных таблиц

Для одноуровневой реализации логической схемы МПА могут использоваться методы, описанные в гл. 5. Пусть задана ГСА Γ и необходимо синтезировать логическую схему МПА S , реализующего эту ГСА Γ , на ПЛМ (s, t, q) . Отметим ГСА Γ метками a_1, \dots, a_M по алгоритму Φ_1 или Φ_2 и построим таблицу переходов автомата. Закодируем как-нибудь способом внутренние состояния МПА и рассмотрим его структурную таблицу, которую будем использовать в качестве исходной для синтеза логической схемы. Учитывая то, что структурная таблица описывает булевы функции $y_1, \dots, y_N, D_1, \dots, D_R$, реализуемые логической схемой, для построения последней на ПЛМ (s, t, q) практически без всяких изменений используются методы гл. 5. Применение этих методов покажем на примере.

Таблица 6.3 представляет собой структурную таблицу МПА, которая уже построена по некоторой заданной ГСА Γ . Пусть по этой таблице необходимо синтезировать логическую схему МПА S на ПЛМ $(10, 10, 20)$. Для любого термина e_b табл. 6.3 выполняются условия $|X(e_b)| \leq (s-R) = 10-4=6$, $|F(e_b)| \leq t=10$ [здесь $F(e_b)$ — множество функций выходов и возбуждения памяти, в которые входит терм e_b], в силу чего может быть использован метод, описанный в § 5.2. По табл. 6.3 построим табл. 6.4, в ней каждому терму e_b поставлены в соответствие множество $X(e_b)$ его входных переменных и множества $Y(e_b)$ и $D(e_b)$ функций выходов и возбуждения элементов памяти, в которые терм входит. Для рассматриваемого примера микроопера-

Таблица 6.3

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$	b
a_1	0000	a_1	0000	$\overline{x_1 x_2}$	$y_1 y_2$ Y_1	—	1
		a_5	0001	$\overline{x_1}$	$y_1 y_2 y_{12}$ Y_2	D_4	2
		a_6	1010	$x_1 x_2 \overline{x_3}$	$y_2 y_{12}$ Y_7	$D_1 D_3$	3
		a_3	1110	$x_1 x_2 x_3$	$y_1 y_2$ Y_1	$D_1 D_2 D_3$	4
a_2	0111	a_2	0111	x_6	— Y_0	$D_2 D_3 D_4$	5
		a_3	1110	$\overline{x_6}$	$y_3 y_5$ Y_4	$D_1 D_2 D_3$	6
a_3	1110	a_3	1110	x_{10}	$y_3 y_5$ Y_4	$D_1 D_2 D_3$	7
		a_8	0101	$\overline{x_4 x_{10}}$	$y_6 y_8 y_9$ Y_6	$D_2 D_4$	8
		a_9	0110	$x_4 \overline{x_{10}}$	$y_{10} y_{15}$ Y_{10}	$D_2 D_3$	9
a_4	1101	a_4	1101	$\overline{x_7 x_9}$	$y_{13} y_{18}$ Y_5	$D_1 D_2 D_4$	10
		a_6	1010	x_7	y_{13} Y_3	$D_1 D_3$	11
		a_8	0101	$\overline{x_7 x_9}$	$y_{13} y_{14}$ Y_{13}	$D_2 D_4$	12
a_5	0001	a_5	0001	$\overline{x_1}$	$y_7 y_{11}$ Y_8	D_4	13
		a_6	1010	x_1	$y_{16} y_{17}$ Y_{12}	$D_1 D_3$	14
a_6	1010	a_1	0000	x_5	$y_1 y_2$ Y_1	—	15
		a_1	0000	$\overline{x_5}$	$y_{16} y_{17}$ Y_{12}	—	16
a_7	0010	a_2	0111	x_8	$y_{14} y_{18}$ Y_{15}	$D_2 D_3 D_4$	17
		a_4	1101	$\overline{x_8}$	$y_{13} y_{18}$ Y_5	$D_1 D_2 D_4$	18
a_8	0101	a_4	1101	$\overline{x_9}$	y_6 Y_9	$D_1 D_2 D_4$	19
		a_7	0010	x_9	$y_4 y_6$ Y_{11}	D_3	20
a_9	0110	a_2	0111	$\overline{x_6 x_{11}}$	$y_5 y_8 y_9$ Y_6	$D_2 D_3 D_4$	21
		a_8	1110	$\overline{x_{11}}$	$y_3 y_8 y_9$ Y_{14}	$D_1 D_2 D_3$	22
		a_9	0110	$x_6 x_{11}$	$y_{10} y_{15}$ Y_{10}	$D_2 D_3$	23

Таблица 6.4

e_b	$X(e_b)$	$Y(e_b)$	$D(e_b)$
e_1	x_1, x_2	y_1, y_2	—
e_2	x_1	y_1, y_2, y_{12}	D_4
e_3	x_1, x_2, x_3	y_2, y_{12}	D_1, D_3
e_4	x_1, x_2, x_3	y_1, y_2	D_1, D_2, D_3
e_5	x_6	—	D_2, D_3, D_4
e_6	x_6	y_3, y_5	D_1, D_2, D_3
e_7	x_{10}	y_3, y_5	D_1, D_2, D_3
e_8	x_4, x_{10}	y_5, y_8	D_2, D_4
e_9	x_4, x_{10}	y_{10}	D_2, D_3
e_{10}	x_7, x_9	y_{13}, y_{18}	D_1, D_2, D_4
e_{11}	x_7	y_{13}	D_1, D_3
e_{12}	x_7, x_9	y_{13}, y_{14}	D_2, D_4
e_{13}	x_1	y_7	D_4
e_{14}	x_1	y_{16}	D_1, D_3
e_{15}	x_5	y_1, y_2	—
e_{16}	x_5	y_{16}	—
e_{17}	x_8	y_{14}, y_{18}	D_2, D_3, D_4
e_{18}	x_8	y_{13}, y_{18}	D_1, D_2, D_4
e_{19}	x_9	y_6	D_1, D_2, D_4
e_{20}	x_9	y_4, y_6	D_3
e_{21}	x_6, x_{11}	y_5, y_8	D_2, D_3, D_4
e_{22}	x_{11}	y_3, y_8	D_1, D_2, D_3
e_{23}	x_6, x_{11}	y_{10}	D_2, D_3

Таблица 6.5

y_n	$Y(y_n)$	y_n	$Y(y_n)$
y_1	Y_1, Y_2	y_{10}	Y_{10}
y_2	Y_1, Y_2, Y_7	y_{11}	Y_8
y_3	Y_4, Y_{14}	y_{12}	Y_2, Y_7
y_4	Y_{11}	y_{13}	Y_3, Y_5, Y_{13}
y_5	Y_4, Y_6	y_{14}	Y_{13}, Y_{15}
y_6	Y_9, Y_{11}	y_{15}	Y_{10}
y_7	Y_8	y_{16}	Y_{12}
y_8	Y_6, Y_{14}	y_{17}	Y_{12}
y_9	Y_6, Y_{14}	y_{18}	Y_5, Y_{15}

ции в каждом из множеств $\{y_7, y_{11}\}$, $\{y_8, y_9\}$, $\{y_{10}, y_{15}\}$, $\{y_{16}, y_{17}\}$ встречаются всегда вместе; в табл. 6.4 из этих множеств оставлено по одному элементу y_7 , y_8 , y_{10} и y_{16} .

Для нахождения этих эквивалентных микроопераций построим табл. 6.5, в которой указано, в какие микрокоманды входит каждая микрооперация. Ясно, что если $Y(y_i) = Y(y_j)$, то микрооперации y_i и y_j входят в одни и те же микрокоманды, т. е. встречаются всегда вместе.

В последовательном алгоритме будем использовать те же оценки $c(e_i, e_j)$, $c(e_i, E^u)$, $W(e_i, E^u)$, что и в § 5.2. Близость $c(e_i, e_j)$ двух термов e_i и e_j определяется выражением

$$c(e_i, e_j) = k_1 |X(e_i) \cap X(e_j)| + k_2 |Y(e_i) \cap Y(e_j)| + k_3 |D(e_i) \cap D(e_j)|. \quad (6.1)$$

Как и ранее, k_1 , k_2 и k_3 — весовые коэффициенты, соотношение которых зависит от соотношений параметров автомата L , N и R и параметров s и t ПЛМ. Положим $k_1 = k_2 = k_3 = 1$. Величины $c(e_i, E^u)$ и $W(e_i, E^u)$ определяются выражениями (5.8) и (5.9). Ограничения (5.4) — (5.6) в данном случае переписутся в виде:

$$|X(E^u)| \leq (s - R); \quad (6.2)$$

$$|F(E^u)| \leq t; \quad (6.3)$$

$$|E^u| \leq q. \quad (6.4)$$

Исходные данные для вычисления оценок $c(e_i, e_j)$, $c(e_i, E^u)$ и $W(e_i, E^u)$ приведены в табл. 6.4. Матрица близости термов с учетом $k_1 = k_2 = k_3 = 1$ помещена в табл. 6.6, строки и столбцы которой отмечены номерами термов множества E (номер i соответствует терму e_i).

Последовательность формирования блоков разбиения π_E представлена в табл. 6.7. В столбце E таблицы перечислены все термы $e_b \in E$. В столбцах $C(e_i, E')$ записаны величины близости каждого терма со всеми остальными, еще не распределенными. Согласно алгоритму эти величины используются для выбора опорных элементов при формировании очередного блока разбиения. Величины $W(e_i, E^u)$ используются для добавления очередного терма в блок. Для облегчения контроля за выполнением ограничений (6.2) — (6.4) в нижних трех строках табл. 6.7 записаны входные переменные термов формируемого блока, а также функции выходов и функции возбуждения, в которые входят эти термы.

Таблица 6.6

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
1	—	3	3	3	1																		
2	3	—	3	6	1	2	2	1	1	2	2	1	2	1	2								
3	3	3	—	2	1	3	3	1	2	1	2	1	1	3	1						1	1	2
4	4	3	6	—	2	3	2	2	2	2	1	2	1	3	2						1	2	3
5		1	1	2	—	3	2	2	2	2	2	1	1	2	2						4	4	3
6			2	3	2	—	5	2	2	2	2	1	1	1	2						3	4	2
7			2	3	2	5	—	3	3	2	2	1	1	1	2						4	4	2
8		1	1	1	2	2	3	—	3	2	1	2	1	1	2						4	2	1
9		1	1	2	2	2	3	3	—	1	1	1	1	1	2						2	2	3
10		1	1	2	2	2	2	2	1	—	3	5	1	1	1						2	2	1
11		1	2	2	1	2	2	2	1	3	—	2	1	2	1						1	2	1
12		1	1	1	2	1	1	2	1	5	2	—	1	1	1						1	2	1
13		1	2	1	1	1	1	1	1	1	1	1	—	1	1						1	1	1
14		1	1	3	3	1	2	1	1	1	2		1	—	1						1	1	2
15		2	2	1												1					1	1	1
16																1					1	1	1
17		1	1	2	3	2	2	2	2	3	1	2	1	1	—						3	2	2
18		1	1	2	2	2	2	2	1	5	2	3	1	1	1						2	2	1
19		1	1	2	2	2	2	2	1	4	1	3	1	1	1						2	2	1
20			1	1	1	1	1	1	1	1	1	1	1	1	1						—	1	1
21		1	1	2	4	4	3	4	2	2	1	2	1	1	2						—	4	4
22		2	2	3	2	4	4	2	2	2	2	1	1	2	1						1	4	3
23				2	3	3	2	1	3	1	1	1	1	1	1						1	4	—

E	c(e _i , E)	W(e _i , E ¹)										c(e _i , E)		
		1	2	3	4	5	6	7	8	9	10		11	
e ₁	14	-6	-6	-2	⊕	+	+	+	+	+	+	+		
e ₂	22	-16	-16	-12	-6	⊕	+	+	+	+	+	+		
e ₃	33	-21	-21	-19	-13	-7	-5	⊕	+	+	+	+		
e ₄	49	⊕	+	+	+	+	+	+	+	+	+	+		
e ₅	37	-33	-33	-33	-33	-31	-29	-27	-25	-23	⊕	+		
e ₆	42	-35	-36	-36	-36	-36	-36	-32	-28	-24	-18	-	30	
e ₇	40	-30	-30	-30	-30	-30	-30	-26	-24	-20	-	-	30	
e ₈	31	-29	-29	-29	-29	-27	-25	-25	-25	-23	-19	-	25	
e ₉	29	-25	-25	-25	-25	-25	-25	-23	-21	-19	-	-	24	
e ₁₀	41	-37	-37	-37	-37	-35	-33	-31	-29	-23	-19	-	30	
e ₁₁	27	-23	-23	-23	-23	-23	-23	-19	-15	⊕	+	+		
e ₁₂	30	-28	-28	-28	-28	-26	-24	-24	-24	-20	-	-	25	
e ₁₃	14	-12	-10	-10	-8	-4	⊕	+	+	+	+	+		
e ₁₄	26	-20	-18	-18	-16	-14	-12	-6	⊕	+	+	+		
e ₁₅	8	-4	-4	⊕	+	+	+	+	+	+	+	+		
e ₁₆	2	-2	⊕	+	+	+	+	+	+	+	+	+		
e ₁₇	37	-33	-33	-33	-33	-31	-29	-27	-25	-23	-17	-	27	
e ₁₈	37	-33	-33	-33	-33	-31	-29	-27	-25	-21	-	-	29	
e ₁₉	35	-31	-31	-31	-31	-29	-27	-25	-23	-21	-	-	28	
e ₂₀	16	-14	-14	-14	-14	-14	-14	-12	-10	-	-	-	13	
e ₂₁	44	-40	-40	-40	-40	-38	-38	-36	-34	-32	-24	-	34	
e ₂₂	40	-36	-36	-36	-36	-36	-36	-32	-28	-14	-20	-	30	
e ₂₃	30	-26	-26	-26	-26	-26	-26	-26	-24	-22	-	-	26	
X		x ₁ , x ₂ , x ₃ , x ₅ , x ₇ , x ₈												
Y		y ₁ , y ₂ , y ₁₆ , y ₁₂ , y ₇ , y ₁₃												
D		D ₁ , D ₂ , D ₃ , D ₄												

В рассматриваемом примере разбиение π_E содержит три блока:

$$E^1 = \{e_1, e_2, e_3, e_4, e_5, e_{11}, e_{13}, e_{14}, e_{15}, e_{16}\};$$

$$X(E^1) = \{x_1, x_2, x_3, x_5, x_6, x_7\}; Y(E^1) = \\ = \{y_1, y_2, y_7, y_{12}, y_{13}, y_{16}\};$$

$$D(E^1) = \{D_1, D_2, D_3, D_4\};$$

$$E^2 = \{e_6, e_7, e_8, e_9, e_{19}, e_{20}, e_{21}, e_{22}, e_{23}\};$$

Таблица 6.7

$W(e_i, E^2)$										$c(e_i, E^2)$	$W(e_i, E^3)$			
1	2	3	4	5	6	7	8	9	10		1	2	3	4
-22	-20	-14	-10	-6	⊕	+	+	+	+					
-24	-22	-18	-12	-6	4	⊕	+	+	+					
-17	-18	-15	-9	⊕	+	+	+	+	+					
-20	-18	-12	⊕	+	+	+	+	+	+					
-26	-24	-22	-20	-16	-12	-8	-4	-4	-	13	-3	⊕	+	+
-21	-19	-17	-15	-11	-9	-7	-5	1	-	12	-6	4	⊕	+
-21	-19	-10	-11	-7	-3	1	5	-	-	11	-3	3	7	⊕
-25	-25	-23	-21	-17	-13	-9	-5	1	-	14	⊕	+	+	+
-24	-20	-18	-16	-12	-8	-4	0	⊕	+					
-11	⊕	+	+	+	+	+	+	+	+					
⊕	+	+	+	+	+	+	+	+	+					
-22	-20	-14	-10	-6	2	10	⊕	+	+					
-18	-16	⊕	+	+	+	+	+	+	+					
$x_8, x_{11}, x_9, x_4, x_{10}$ $y_5, y_8, y_4, y_6, y_{10}, y_3$ D_2, D_8, D_4, D_1										x_8, x_7, x_9 y_{13}, y_{18}, y_{14} D_1, D_2, D_3, D_4				

$$\begin{aligned}
 X(E^2) &= \{x_4, x_6, x_9, x_{10}, x_{11}\}; Y(E^2) = \\
 &= \{y_3, y_4, y_5, y_6, y_8, y_{10}\}; \\
 D(E^2) &= \{D_1, D_2, D_3, D_4\}; \\
 E^3 &= \{e_{10}, e_{12}, e_{17}, e_{18}\}; X(E^3) = \{x_7, x_8, x_9\}; \\
 Y(E^3) &= \{y_{13}, y_{14}, y_{18}\}; D(E^3) = \{D_1, D_2, D_3, D_4\}.
 \end{aligned}$$

Построенная по этим выражениям логическая схема приведена на рис. 6.3. Выходы ПЛМ на этом рисунке

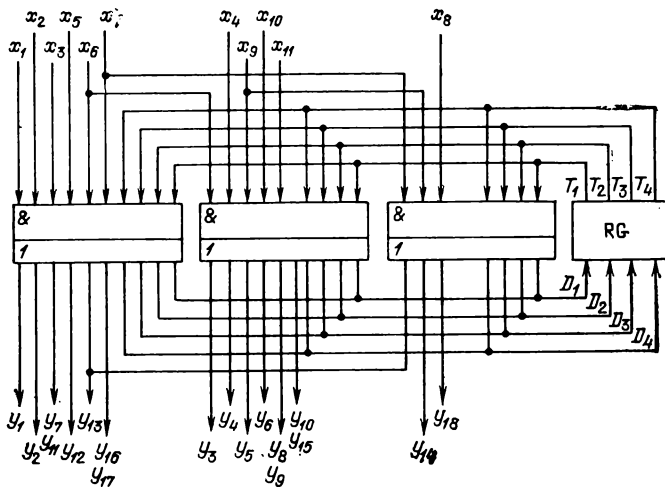


Рис. 6.3

дополнены микрооперациями y_9 , y_{11} , y_{15} и y_{17} , которые встречаются всегда вместе с y_8 , y_7 , y_{10} и y_{16} соответственно. Таблицы программирования каждой из ПЛМ приведены в табл. 6.8—6.10, в которых указаны термы структурной таблицы, реализуемые на промежуточных шинах.

Аналогичным образом для синтеза логической схемы МПА по его структурной таблице могут использоваться и другие методы, описанные в гл. 5, а также методы, рассмотренные в [1, 19—27].

Упростить решение задачи синтеза логической схемы МПА можно получением более простых функций в его

Таблица 6.8

T_1	T_2	T_3	T_4	x_1	x_2	x_3	x_5	x_6	x_7	e_h	D_1	D_2	D_3	D_4	y_1	y_2	y_7	y_{12}	y_{13}	y_{16}
0	0	0	0	1	0	—	—	—	—	e_1	1	1
0	0	0	0	0	—	—	—	—	—	e_2	.	.	.	1	1	1	.	1	.	.
0	0	0	0	1	1	0	—	—	—	e_3	1	.	1	.	1	.	.	1	.	.
0	0	0	0	1	1	1	—	—	—	e_4	1	1	1	.	1	1
0	1	1	1	—	—	—	—	—	1	e_5	.	1	1	1
1	1	0	1	—	—	—	—	—	1	e_{11}	1	.	1	1	.
0	0	0	1	0	—	—	—	—	—	e_{13}	.	.	.	1	.	1
0	0	0	1	1	—	—	—	—	—	e_{14}	1	.	1	1
1	0	1	0	—	—	—	1	—	—	e_{15}	1	1
1	0	1	0	—	—	—	0	—	—	e_{16}	1

Таблица 6.9

T_1	T_2	T_3	T_4	x_4	x_6	x_9	x_{10}	x_{11}	e_h	D_1	D_2	D_3	D_4	y_3	y_4	y_5	y_6	y_8	y_{10}
0	1	1	1	—	0	—	—	—	e_6	1	1	1	.	1	.	1	.	.	.
1	1	1	0	—	—	—	—	1	e_7	1	1	1	.	1	.	1	.	.	.
1	1	1	0	0	—	—	0	—	e_8	.	1	1	.	.	1	.	1	.	.
1	1	1	0	1	—	—	0	—	e_9	.	1	1	1
0	1	0	1	—	—	0	—	—	e_{19}	1	1	.	1	.	.	.	1	.	.
0	1	0	1	—	—	1	—	—	e_{20}	.	.	1	.	.	1	.	1	.	.
0	1	1	0	—	0	—	—	1	e_{21}	.	1	1	1	.	.	1	.	1	.
0	1	1	0	—	—	—	—	0	e_{22}	1	1	1	.	1	.	.	.	1	.
0	1	1	0	—	1	—	—	1	e_{23}	.	1	1	1

Таблица 6.10

T_1	T_2	T_3	T_4	x_7	x_8	x_9	e_h	D_1	D_2	D_3	D_4	y_{13}	y_{14}	y_{18}
1	1	0	1	0	—	1	e_{10}	1	1	.	1	1	.	1
1	1	0	1	0	—	0	e_{12}	.	1	.	1	1	1	.
0	0	1	0	—	1	—	e_{17}	.	1	1	1	.	1	1
0	0	1	0	—	0	—	e_{18}	1	1	.	1	1	.	1

структурной таблице. Функции y_1, \dots, y_N этой таблицы заданы при синтезе МПА и не могут быть далее изменены, а функции возбуждения D_1, \dots, D_R строятся по результатам кодирования внутренних состояний автомата, и формировать их можно по-разному. Если, например, состояния удастся закодировать так, что значения некоторой функции возбуждения D_r не будут зависеть от значений входных переменных $X = \{x_1, \dots, x_L\}$ на всех переходах из некоторого состояния a_m , то это значение D_r на указанных переходах можно сформировать на выходе любой ПЛМ независимо от того, какие переменные из множества X поступают на ее входы. Если состояния МПА из табл. 6.3 закодировать так, как в карте Карно на рис. 6.4, то не будут зависеть от значений входных переменных значения функций D_3, D_4 , формируемые на переходах из состояний a_1, a_3 , функций D_1, D_3, D_4 — на переходах из состояний a_2, a_5, a_8 , функций D_1, D_2 — на переходах из состояния a_4 и функций D_1, D_2, D_3, D_4 — на переходах из состояния a_6 .

Не останавливаясь подробно на таком способе кодирования (см. [31]), замечаем, что на рис. 6.4 состояния из

одного множества $A(a_m)$ размещены либо в одном столбце, либо в одной строке карты Карно. Здесь $A(a_m)$ — множество всех состояний, в которые есть переходы из a_m ; эти состояния записаны в столбце a_s в подмассиве переходов из a_m таблицы переходов МПА. Такое кодирование будем

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	a_1	a_2	a_2	
	01	a_3	a_3		
	11	a_6	a_3	a_4	
	10	a_5	a_9	a_7	

Рис. 6.4

называть кодированием с ослабленной зависимостью от входных переменных (ОЗ-кодированием).

Заметим, что значения функций возбуждения, которые не зависят от значений входных переменных, могут быть сформированы на выходах специально введенного ПЗУ, на входы которого поступают значения переменных T_1, \dots, T_R . С выходов этого ПЗУ могут сниматься и некоторые выходные сигналы, значения кото-

рых определяются лишь внутренним состоянием автомата. Выходы ПЛМ и ПЗУ, соответствующие одним и тем же переменным, объединяются по схеме «монтажное ИЛИ».

Синтез автомата Мура на ПЛМ осуществляется рассмотренным в этом параграфе методом без учета функций выходов — из табл. 6.4 исключается столбец $Y(e_b)$, после чего выходные переменные распределяются по выходам ПЛМ. Решение этой последней задачи подробно рассмотрено в конце § 6.4.

6.4. Синтез МПА с разделением входных переменных

Пусть задана ГСА Γ , по которой необходимо синтезировать логическую схему реализующего ее МПА S на ПЛМ (s, t, q) . Отметим ГСА Γ метками a_1, \dots, a_M по алгоритму Φ_2 (синтез МПА Мура) и определим значения $N, L, B, R = \text{int} \log_2 M$. Предположим, что $(L+R) > s$, $(N+R) > t$, $B > q$; ясно, что в этом случае тривиальная реализация автомата S невозможна. Рассмотрим метод [31, 32] построения по ГСА Γ одноуровневой логической схемы МПА Мура S , который позволяет учитывать особенности используемого элементного базиса уже при синтезе МПА по ГСА и далее при кодировании внутренних состояний автомата и построении его структурной таблицы. Согласно этому методу процесс синтеза автомата делится на следующие этапы:

1. Специальная отметка ГСА Γ и построение подтаблиц переходов для каждой u -й, из которых выполняются условия: $(L_u + R) \leq s$, $B_u \leq q$, где $L_u = |X^u|$, X^u — множество

логических условий (входных переменных), записанных в столбце $X(a_m, a_s)$ u -й подтаблицы. Указанные подтаблицы в совокупности задают автомат S , синтезированный по ГСА Γ , а их число U определяется из выражения

$$U = \max (\] L / (s - R) [, \] B / q [). \quad (6.5)$$

2. Кодирование внутренних состояний автомата S , позволяющее минимизировать число выходов в каждой ПЛМ одноуровневой схемы, на которых формируются значения функций возбуждения D_1, \dots, D_R .

3. Распределение выходных переменных между ПЛМ₁, ..., ПЛМ_U. Основным критерием здесь является число U ПЛМ, а дополнительным — суммарное число использованных промежуточных шин.

4. Построение структурной таблицы автомата S и таблиц программирования ПЛМ₁, ..., ПЛМ_U.

Ниже подробно рассматривается каждый из перечисленных этапов синтеза. Изложение сопровождается примером синтеза одноуровневой схемы автомата Мура S по ГСА Γ , изображенной на рис. 6.5, на ПЛМ (7, 6, 12). Отметим ГСА Γ метками a_1, \dots, a_M по алгоритму Φ_2 , поставив в соответствие каждой операторной вершине одну из меток a_2, \dots, a_M , начальной и конечной вершинам — метку a_1 . ГСА на рис. 6.5 отмечена метками a_1, \dots, a_{10} и $M=10, R=4$. На дополнительные метки a_{11} и a_{12} не обращать внимания.

На множестве $X = \{x_1, \dots, x_L\}$ логических условийведем отношение связанности такое, что x_i и x_j ($i \neq j$) связаны, если выход условной вершины ГСА, в которой записано одно из них, соединен дугой с неотмеченным входом другой условной вершины ГСА, в которой записано второе из этих условий. Забегая вперед, отмечаем, что если логические условия x_i и x_j связаны и x_i встречается в столбце $X(a_m, a_s)$ в u -й подтаблице ($x_i \in X^u$), то и x_j должно быть в той же подтаблице, т. е. $x_j \in X^u$.

Построим взвешенный граф Γ_c отношения связанности. В этом графе L вершин, соответствующих логическим условиям x_1, \dots, x_L . Две вершины x_i и x_j соединяются ребром, если и только если логические условия x_i и x_j связаны. Над каждым ребром графа указывается его вес d , который будем определять следующим образом. Если в ГСА два логических условия x_i и x_j связаны неоднократно, например d раз, то над ребром (x_i, x_j) записывается число d . Если неотмеченный вход условной вершины, содержащей x_j , соединен дугами с выходами f условных вершин, содержащих логические условия x_{i1}, \dots, x_{if} , то меру связанности

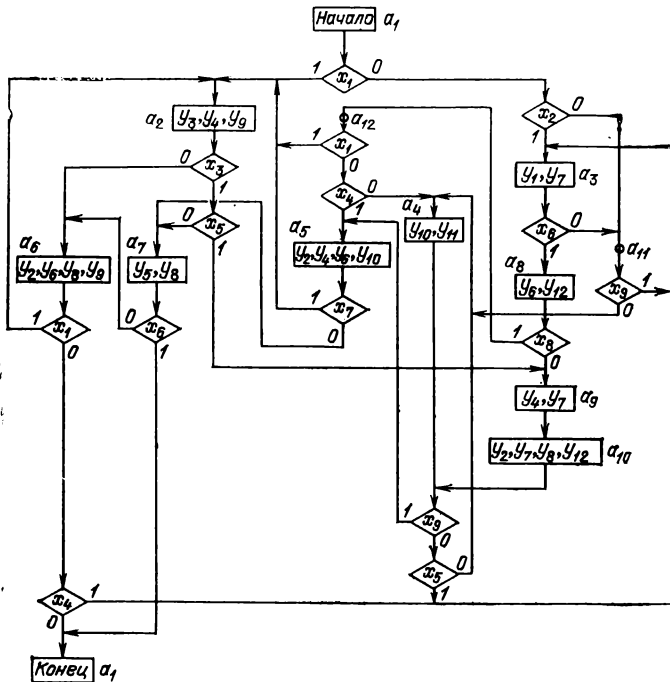


Рис. 6.5

пар $(x_j, x_{i1}), \dots, (x_j, x_{if})$ увеличим на $1/f$. Граф Γ_c , построенный по ГСА на рис. 6.5, показан на рис. 6.6.

Для реализации этапа 1 граф Γ_c необходимо разрезать на минимально возможное число U подграфов $\Gamma_1, \dots, \Gamma_U$ таких, что $L_u = |X^u| \leq (s-R)$, $B_u \leq q$ ($u = \overline{1, U}$). Здесь B_u — число строк в u -й подтаблице, т. е. число путей перехода в ГСА Γ , проходящих через условные вершины с логическими условиями из множества X^u .

Для решения указанной задачи могут использоваться алгоритмы, рассмотренные в [33]. При разрезании графа Γ_c удаляются некоторые ребра. Если удаляется ребро, соединяющее вершины x_i и x_j , то в ГСА неотмеченный вход условной вершины, содержащей x_i (x_j), соединенный дугой с выходом условной вершины, содержащей x_j (x_i), отмечается дополнительной меткой. Если вес удаленного ребра $d > 1$, то в ГСА вводится несколько дополнительных меток. Таким образом, минимизация суммарного веса удаленных ребер способствует минимизации числа вновь вводимых дополнительных меток.

Разрезание графа Γ_c на подграфы необходимо производить следующим образом. На первом шаге Γ_c разрезается на два подграфа, первый из которых содержит $s-R$, а второй $L-(s-R)$ вершин. Далее в ГСА вводятся дополнительные метки и корректируются веса ребер графа, так как введение дополнительной метки при удалении некоторого ребра графа Γ_c может привести к сокращению весов других ребер или к удалению некоторых из них. На втором шаге из подграфа, содержащего $L-(s-R)$ вершин, выделяется подграф, содержащий $s-R$ вершин, после чего снова корректируются веса ребер. Рассмотренный процесс продолжается до тех пор, пока граф Γ_c не будет разрезан на подграфы, каждый из которых содержит не более чем $s-R$ вершин.

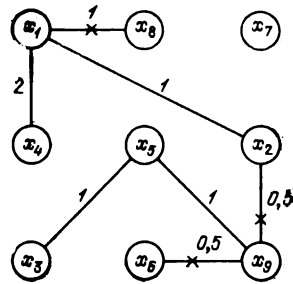


Рис. 6.6

Теперь должен быть ясен принцип выбора весов ребер графа Γ_c . Так, если неотмеченный вход условной вершины, содержащей x_j , соединен с выходами f условных вершин, содержащих логические условия x_{i1}, \dots, x_{if} (при $f > 1$), то связность любой пары $(x_j, x_{i1}), \dots, (x_j, x_{if})$ равна не 1, а $1/f$. Это объясняется тем, что достаточно отметить вход условной вершины x_j единственной меткой и все f указанных пар окажутся несвязанными.

В нашем примере $R=4$, $s=7$, поэтому граф Γ_c , изображенный на рис. 6.6, должен быть разрезан на подграфы, каждый из которых содержит не более $L_u=7-4=3$ логических условий. Граф на рис. 6.6 разрезается на три подграфа $\Gamma_1, \Gamma_2, \Gamma_3$ путем удаления ребер, отмеченных крестиками. В результате в ГСА Γ (см. рис. 6.5) вводятся две дополнительные метки a_{11} и a_{12} и $X^1=\{x_1, x_2, x_4\}$. $X^2=\{x_3, x_5, x_9\}$, $X^3=\{x_6, x_7, x_8\}$, $L_1=L_2=L_3=3$; так как $R=\text{int log}_2 M=\text{int log}_2 12=4$, $s=7$, то для всех $u=1, 2, 3$ выполняется условие $L_u=s-R$.

После отметки ГСА дополнительными метками сформируем разбиение $\pi_x=\{X^1, \dots, X^v\}$, каждый u -й блок которого включает логические условия, соответствующие вершинам u -го подграфа графа Γ_c .

Построим по ГСА Γ такие подтаблицы переходов МПА, в каждой u -й из которых записаны лишь логиче-

ские условия из одного блока X^u , где $u = \overline{1, U}$. Принцип отметки ГСА Γ позволяет это сделать. Действительно, пусть $x_i \in X^u$, $x_j \in X^k$, $u \neq k$; логические условия x_i и x_j записаны в условных вершинах, выход одной из которых соединен дугой со входом другой. Тогда вход условной вершины, в которую ведет дуга, всегда будет отмечен меткой, что следует из принципа разрезания графа Γ_c и дополнительной отметки ГСА Γ . В силу этого в ГСА Γ всегда можно сформировать путь перехода вида $a_m X(a_m, a_s) a_s$ или $a_m X(a_m, a_s) Y(a_m, a_s) a_s$ (см. гл. 2), для которого все буквы в конъюнкции $X(a_m, a_s)$ являются элементами одного множества X^u .

Будем говорить, что все пути перехода в ГСА, идущие из одной метки a_m , образуют подмассив путей переходов из метки a_m ; соответственно все строки таблицы переходов (структурной таблицы), описывающие переходы из одного состояния a_m , образуют подмассив переходов из состояния a_m . Формирование по ГСА Γ подтаблиц переходов выполняется так же, как и ранее (см. гл. 2). Другими словами, в ГСА формируются все пути перехода вида $a_m X(a_m, a_s) a_s$ или $a_m X(a_m, a_s) Y(a_m, a_s) a_s$. Если для некоторого пути все буквы в конъюнкции $X(a_m, a_s)$ являются элементами множества X^u , то ему ставится в соответствие одна из строк u -й подтаблицы переходов. Аналогичные действия продолжают до тех пор, пока все пути перехода не будут распределены между U подтаблицами. В правильно построенных подтаблицах дизъюнкция элементарных конъюнкций $X(a_m, a_s)$, записанных в строках одного подмассива переходов, тождественно равна 1.

Продолжим рассмотрение нашего примера. В табл. 6.11 представлены подтаблицы переходов W_0, W_1, W_2, W_3 построенные по ГСА на рис. 6.5 (различные подтаблицы отделены друг от друга двойными горизонтальными линиями). В табл. 6.11 выделена нулевая подтаблица, которую будем обозначать W_0 . В W_0 сведены все переходы в автомате, для которых $X(a_m, a_s) = 1$, что соответствует путям между связанными операторными вершинами в ГСА. Основными подтаблицами табл. 6.11 будем считать подтаблицы W_u , где $u \geq 1$.

Если в результате введения дополнительных меток общее их число в ГСА возрастет до M^* и $R = \text{int} \log_2 M < \text{int} \log_2 M^*$, то R увеличивается настолько, чтобы выполнялось условие $R = \text{int} \log_2 M^*$. В этом случае для построения МПА на ПЛМ (s, t, q) по схеме, изображенной

на рис. 6.2, все ранее рассмотренные действия повторяются с новым значением R .

При синтезе одноуровневой схемы МПА Мили выполняются аналогичные действия, но исходная ГСА отмечается предварительно по алгоритму Φ_1 .

Пусть состояния МПА в нашем примере закодированы так, как это представлено в карте Карно на рис. 6.7. Здесь также использовано ОЗ-кодирование (см. конец § 6.3) такое, что в пределах одной подтаблицы W_u ($u = \overline{1, U}$) от значений входных переменных не зависят значения одних и тех же функций возбуждения [31]. Коды состояний занесены в столбцы $K(a_m)$ и $K(a_s)$ структурной таблицы МПА (табл. 6.12), ее различные подтаблицы отделены друг от друга двойными горизонтальными линиями. В столбце $K(a_s)$ жирным выделены разряды кодов, которые не меняются на всех переходах из состояния a_m в пределах одной подтаблицы.

Пусть $T = \{T_1, \dots, T_R\}$ — множество D -триггеров — память автомата. Через G^u и H^u обозначим подмножества T , элементы которых принимают соответственно различные и одинаковые значения на переходах из одного состояния в подтаблице W_u : $G^u \subseteq T$, $H^u \subseteq T$, $G^u \cup H^u = T$, $G^u \cap H^u = \emptyset$. В примере:

$$\begin{aligned} G^1 &= \{T_1, T_2\}, & H^1 &= \{T_3, T_4\}; \\ G^2 &= \{T_2, T_4\}, & H^2 &= \{T_1, T_3\}; \\ G^3 &= \{T_3\}, & H^3 &= \{T_1, T_2, T_4\}. \end{aligned}$$

Обозначим через $D(a_m)$ множество функций возбуждения, которые принимают значения 1 на переходах из состояния a_m : $D(a_m) = D_G(a_m) \cup D_H(a_m)$.

Здесь $D_G(a_m)$ и $D_H(a_m)$ соответствуют разрядам кодов из множеств G^u и H^u — меняющимся и неизменным на переходах из a_m в подтаблице W_u . Например,

$$D(a_4) = \{D_1, D_2, D_4\}; \quad D_G(a_4) = \{D_2, D_4\}; \quad D_H(a_4) = \{D_1\}.$$

Функции возбуждения из множества $D_H(a_m)$ заключены в круглые скобки в столбце $F(a_m, a_s)$ подмассива переходов из a_m . Ясно, что в подтаблице W_0 для всех $a_m \in A_0$ $D_G(a_m) = \emptyset$, $D(a_m) = D_H(a_m)$ и все функции возбуждения — в круглых скобках.

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	a_1	a_{12}	a_9	a_5
	01	a_2	a_2		a_7
	11	a_{11}	a_5		a_8
	10	a_3	a_4	a_{10}	

Рис. 6.7

Таблица 6.11

$a_m, Y(a_m)$	a_s	$X(a_m, a_s)$
a_0, y_4, y_7	a_{10}	1
$a_1, -$	a_2 a_3 a_{11}	$\overline{x_1}$ $\overline{x_1 x_2}$ $\overline{x_1 x_3}$
a_6, y_2, y_6, y_8, y_9	a_1 a_2 a_3	$\overline{x_1 x_4}$ $\overline{x_1}$ $\overline{x_1 x_4}$
$a_{12}, -$	a_2 a_4 a_6	$\overline{x_1}$ $\overline{x_1 x_4}$ $\overline{x_1 x_4}$
a_3, y_3, y_4, y_9	a_6 a_7 a_9	$\overline{x_3}$ $\overline{x_3 x_5}$ $\overline{x_3 x_5}$
a_4, y_{10}, y_{11}	a_3 a_4 a_5	$\overline{x_5 x_9}$ $\overline{x_5 x_9}$ $\overline{x_9}$
$a_{10}, y_2, y_7, y_8, y_{12}$	a_3 a_4 a_5	$\overline{x_5 x_9}$ $\overline{x_5 x_9}$ $\overline{x_9}$
$a_{11}, -$	a_3 a_4	$\overline{x_9}$ $\overline{x_9}$
a_8, y_1, y_7	a_8 a_{11}	$\overline{x_6}$ $\overline{x_6}$
$a_6, y_2, y_6, y_4, y_{10}$	a_2 a_7	$\overline{x_7}$ $\overline{x_7}$
a_7, y_5, y_8	a_1 a_8	$\overline{x_8}$ $\overline{x_8}$
a_8, y_6, y_{12}	a_9 a_{12}	$\overline{x_8}$ $\overline{x_8}$

Таблица 6.12

$a_m, Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$F(a_m, a_s)$
a_9, y_4, y_7	0011	a_{10}	1011	1	$(D_1 D_3 D_4)$
$a_1, -$	0000	a_2 a_3 a_{11}	010- 1000 1100	x_1 $\overline{x_1} x_2$ $\overline{x_1} \overline{x_2}$	D_2 D_1 $D_1 D_2$
a_6, y_2, y_6, y_8, y_9	0010	a_1 a_2 a_3	0000 010- 1000	$\overline{x_1} \overline{x_4}$ x_1 $\overline{x_1} x_4$	- D_2 D_1
$a_{12}, -$	0001	a_2 a_4 a_5	010- 1001 1101	x_1 $\overline{x_1} \overline{x_4}$ $\overline{x_1} x_4$	$D_2 (D_4)$ $D_1 (D_4)$ $D_1 D_2 (D_4)$
a_2, y_3, y_4, y_9	010-	a_6 a_7 a_9	0010 0110 0011	$\overline{x_3}$ $x_3 \overline{x_5}$ $x_3 x_5$	(D_3) $D_2 (D_3)$ $D_4 (D_3)$
a_4, y_{10}, y_{11}	1001	a_3 a_4 a_5	1000 1001 1101	$x_5 \overline{x_9}$ $\overline{x_5} \overline{x_9}$ x_9	(D_1) $D_4 (D_1)$ $D_2 D_4 (D_1)$
$a_{10}, y_2, y_7, y_8, y_{12}$	1011	a_3 a_4 a_5	1000 1001 1101	$x_5 \overline{x_9}$ $\overline{x_5} \overline{x_9}$ x_9	(D_1) $D_4 (D_1)$ $D_2 D_4 (D_1)$
$a_{11}, -$	1100	a_3 a_4	1000 1001	x_9 $\overline{x_9}$	(D_1) $D_4 (D_1)$
a_3, y_1, y_7	1000	a_8 a_{11}	1110 1100	x_6 $\overline{x_6}$	$D_3 (D_1 D_2)$ $(D_1 D_2)$
$a_5, y_2, y_6, y_4, y_{10}$	1101	a_2 a_7	010- 0110	x_7 $\overline{x_7}$	(D_2) $D_3 (D_2)$
a_7, y_5, y_8	0110	a_1 a_6	0000 0010	x_6 $\overline{x_6}$	- D_3
a_8, y_6, y_{12}	1110	a_9 a_{12}	0011 0001	$\overline{x_8}$ x_8	$D_3 (D_4)$ (D_4)

Если каждая подтаблица W_u ($u = \overline{1, U}$) будет реализована одной u -й ПЛМ, то значения функции возбуждения $D_r \in D_G(a_m)$ должны формироваться в той ПЛМ, в которой реализованы переходы из a_m . Значения же $D_q \in D_H(a_m)$, так как они не зависят от входных переменных, могут быть сформированы в любой r -й ПЛМ, для которой $T_q \in G^r$. При этом в r -й ПЛМ используется новая промежуточная шина, на которой реализуется конъюнкция переменных T_1, \dots, T_R , равная 1 тогда и только тогда, когда МПА находится в состоянии a_m . Если $|D_H(a_m)| \geq 2$ и значения функций из множества $D_H(a_m)$ формируются на выходах нескольких ПЛМ, то в каждой из них произойдет увеличение числа использованных промежуточных шин на 1.

При распределении функций из каждого множества $D_H(a_m)$ ($a_m \in A_u$, $u = \overline{1, U}$) между структурными подтаблицами будем стремиться к тому, чтобы суммарное увеличение числа использованных промежуточных шин во всех ПЛМ было минимально. Если $|D_H(a_m)| \geq 2$, то для состояния a_m построим матрицу $M(a_m)$, столбцы которой поставим в соответствие элементам множества $D_H(a_m)$, а строки — в соответствие ПЛМ₁, ..., ПЛМ_U. На

пересечении j -го столбца и u -й строки $M(a_m)$ поставим 1, если значения функции возбуждения D_q , соответствующей столбцу j , могут быть сформированы на выходе ПЛМ_u, т. е. $T_q \in G^u$, и число промежуточных шин ПЛМ_u может быть увеличено на 1. Из матрицы $M(a_m)$ вычеркнем строки, не содержащие единиц, и найдем ее минималь-

Т а б л и ц а 6.13

D_{H_i}		
ПЛМ	D_1	D_2
ПЛМ ₁	1	1
ПЛМ ₂	0	1
ПЛМ ₃	0	0

Т а б л и ц а 6.14

ПЛМ ₁	ПЛМ ₂	ПЛМ ₃
$D_1 \in D_H(a_4)$ $D_1 \in D_H(a_{10})$ $D_1 \in D_H(a_{11})$ $D_1 \in D_H(a_3)$ $D_2 \in D_H(a_3)$ $D_1 \in D_H(a_9)$	$D_4 \in D_H(a_{12})$ $D_4 \in D_H(a_8)$ $D_3 \in D_H(a_5)$ $D_4 \in D_H(a_9)$	$D_3 \in D_H(a_9)$ $D_3 \in D_H(a_2)$

ное строчное покрытие, причем в первую очередь в минимальное строчное покрытие включаются те строки, которые соответствуют ПЛМ с большим числом неиспользуемых промежуточных шин. На выходах ПЛМ, соответствующих строкам минимального покрытия, формируются значения функций из множества $D_H(a_m)$. Если $a_m \in A_0$ (a_m — в подтаблице W_0), то $D_H(a_m) = D(a_m)$, $D_G(a_m) = \emptyset$ и функции возбуждения распределяются между ПЛМ аналогичным образом.

Последовательное выполнение рассмотренных действий для всех $a_m \in A_u$ и $u = 0, \overline{U}$ позволяет распределить между ПЛМ все функции $D_H(a_m)$, $a_m \in A_u$, $u = 0, \overline{U}$. В тех же случаях, когда покрытие некоторой матрицы $M(a_m)$ не может быть найдено, требуемые значения функций из соответствующего множества $D_H(a_m)$ могут быть сформированы на выходах дополнительной $(U+1)$ -й ПЛМ, на входы которой подаются значения переменных T_1, \dots, T_R (вместо ПЛМ здесь часто рациональнее использовать ПЗУ).

Распределим функции возбуждения из множеств $D_H(a_m)$ для табл. 6.12 между ПЛМ₁—ПЛМ₃ с параметрами $s=7$, $t=6$, $q=12$. Будем считать, что выходы всех ПЛМ свободны, т. е. выходные переменные еще не распределены. Так как $D_H(a_1) = D_H(a_6) = D_H(a_7) = \emptyset$, то эти множества не рассматриваются

Мощность множеств $D_H(a_{12}) = \{D_4\}$, $D_H(a_2) = \{D_3\}$, $D_H(a_4) = \{D_1\}$, $D_H(a_{10}) = \{D_1\}$, $D_H(a_{11}) = \{D_1\}$, $D_H(a_5) = \{D_2\}$, $D_H(a_8) = \{D_4\}$ равна единице, поэтому функции $D_4 \in D_H(a_{12})$ и $D_4 \in D_H(a_8)$ сразу же распределяем на выход ПЛМ₂, функции $D_1 \in D_H(a_4)$, $D_1 \in D_H(a_{10})$ и $D_1 \in D_H(a_{11})$ — на выход ПЛМ₁, функции $D_3 \in D_H(a_9)$ и $D_3 \in D_H(a_2)$ — на выход ПЛМ₃, функцию $D_2 \in D_H(a_5)$ — на выход ПЛМ₂.

Элементы множества $D_H(a_3) = \{D_1, D_2\}$ распределяются между выходами ПЛМ₁, ..., ПЛМ₃ после построения матрицы $M(a_3)$ (табл. 6.13) и нахождения ее минимального строчного покрытия. В результате распределения функции $D_1 \in D_H(a_3)$ и $D_2 \in D_H(a_3)$ назначаются на выходы ПЛМ₁. Окончательное распределение функций из множеств $D_H(a_m)$ по ПЛМ₁, ПЛМ₂ и ПЛМ₃ представлено в табл. 6.14. С учетом кодов $K(a_m)$ состояний a_m из табл. 6.12 эти функции можно описать тремя парами матриц¹ E^u, F^u :

¹ При минимизации функций, заданных матрицами E^1, F^1 , ниже допущена ошибка. Попробуйте найти ее и устранить неточность в полученной схеме при том же числе ПЛМ.

$$\begin{array}{c}
 T_1 \ T_2 \ T_3 \ T_4 \\
 E^1 = \begin{vmatrix} 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{vmatrix}; \quad F^1 = \begin{vmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \end{vmatrix}; \\
 D_1 \ D_2
 \end{array}$$

$$\begin{array}{c}
 T_1 \ T_2 \ T_3 \ T_4 \\
 E^2 = \begin{vmatrix} 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{vmatrix}; \quad F^2 = \begin{vmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{vmatrix}; \\
 D_2 \ D_4
 \end{array}$$

$$\begin{array}{c}
 T_1 \ T_2 \ T_3 \ T_4 \\
 E^3 = \begin{vmatrix} 0 & 1 & 0 & - \\ 0 & 0 & 1 & 1 \end{vmatrix}; \quad F^3 = \begin{vmatrix} 1 \\ 1 \end{vmatrix}. \\
 D_3
 \end{array}$$

Теперь в столбце $F(a_m, a_s)$ структурной таблицы можно уже не рассматривать элементов, заключенных в скобки (см. табл. 6.12), а любая пара матриц E^u, F^u будет давать дополнительную к структурной таблице информацию о программировании u -й ПЛМ для формирования значений функций из множеств $D_H(a_m)$. Например, первая строка матриц E^1 и F^1 указывает на то, что если на входы T_1, \dots, T_4 первой ПЛМ будет подан двоичный набор 1001 (код a_4), то на ее выходе D_1 должен быть сформирован сигнал с единичным, а на выходе D_2 — с нулевым значениями.

Булевы функции, заданные матрицами E^u и F^u , могут быть доопределены на двоичных наборах значений переменных, соответствующих пустым клеткам карты Карно на рис. 6.7, и минимизированы. Так, для нашего примера рассматриваемые матрицы после минимизации представлены в виде:

$$\begin{array}{c}
 T_1 \ T_2 \ T_3 \ T_4 \\
 E^1 = \begin{vmatrix} - & - & 1 & 1 \\ 1 & 0 & - & - \\ 1 & 0 & 0 & 0 \end{vmatrix}; \quad F^1 = \begin{vmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{vmatrix}; \\
 D_1 \ D_2
 \end{array}$$

$$E^2 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 0 & 0 & - & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{array} \right|; \quad F^2 = \begin{array}{c} D_3 \ D_4 \\ \left| \begin{array}{cc} 0 & 1 \\ 1 & 0 \\ 0 & 1 \end{array} \right|; \\ T_1 \ T_2 \ T_3 \ T_4 \quad D_3 \\ E^3 = \left| \begin{array}{cccc} 0 & 1 & 0 & - \\ 0 & 0 & 1 & 1 \end{array} \right|; \quad F^3 = \left| \begin{array}{c} 1 \\ 1 \end{array} \right|.$$

Число строк в матрицах E^u , F^u определяет дополнительное число промежуточных шин в u -й ПЛМ.

Вернемся к рис. 6.5, на котором изображена исходная ГСА Г. В этой ГСА выходы операторных вершин, отмеченные символами a_4 и a_{10} соединены со входом одной и той же вершины ГСА. Из этого следует, что все пути перехода из a_4 и a_{10} совпадают, что видно и из табл. 6.12. Тогда, если состояния a_4 и a_{10} закодированы соседними кодами, что и сделано в нашем примере [$K(a_4) = 1001$, $K(a_{10}) = 1011$], число конъюнкций, которые необходимо реализовать на промежуточных шинах ПЛМ₂, сократится вдвое. Действительно, конъюнкции $T_1 T_2 T_3 T_4 \bar{x}_5 \bar{x}_9$ и $T_1 T_2 T_3 T_4 x_5 \bar{x}_9$, $T_1 T_2 T_3 T_4 x_9$ и $T_1 T_2 T_3 T_4 \bar{x}_9$ склеятся по переменной T_3 , после чего останутся только две конъюнкции: $T_1 T_2 T_4 \bar{x}_5 \bar{x}_9$ и $T_1 T_2 T_4 x_9$.

После кодирования внутренних состояний МПА осуществляется распределение выходных переменных между ПЛМ₁, ..., ПЛМ_U. Заметим, что эта задача может решаться так же, как и известная задача реализации на ПЛМ систем БФ с учетом заданных импликантов [34].

Пусть p_u — число свободных выходов u -й ПЛМ (s , t , q) после распределения функций возбуждения по выходам ПЛМ, тогда общее число свободных выходов во всех ПЛМ $p = p_1 + \dots + p_U$. В примере $p_1 = p_2 = 4$, $p_3 = 5$. Если $p > 0$, то p выходных переменных из множества Y могут быть распределены между U ПЛМ. Нераспределенные выходные переменные назначаются на выходы дополнительных ПЛМ, входы которых подключаются к выходам элементов памяти автомата.

Рассмотрим структурные подтаблицы W_1, \dots, W_U , в столбец $F(a_m, a_s)$ которых занесены лишь функции возбуждения из множеств $D_G(a_m)$; символы D_r , заключенные в скобки, далее не принимаются во внимание, считается, что их в таблице нет. Для каждой u -й подтаблицы W_u сформируем множество Q^u таких элементарных конъюнкций переменных T_1, \dots, T_R , которые равны 1 на следующих

наборах их значений: а) заданных матрицей E^u ; б) записанных в столбце $K(a_m)$ подмассивов переходов таблицы W_u , не содержащих прочерков в столбце $F(a_m, a_s)$.

После настройки ПЛМ_u на реализацию функций возбуждения для каждой k -й конъюнкции множества Q^u на свободной выходной шине u -й ПЛМ может быть сформирован сигнал, равный единице, когда равна единице k -я конъюнкция, и нулю в противном случае.

Если u -й ПЛМ поставлено в соответствие множество Y^u выходных переменных (функций) и любая функция из множества Y^u может быть представлена дизъюнкцией элементарных конъюнкций множества Q_u , то число использованных промежуточных шин u -й ПЛМ увеличиваться не будет. Если в множестве Y^u имеется хотя бы одна функция, для которой указанное требование не выполняется, число использованных промежуточных шин u -й ПЛМ увеличится. Для табл. 6.12:

$$Q^1 = \{\bar{T}_1, \bar{T}_2, \bar{T}_3, \bar{T}_4, \bar{T}_1, \bar{T}_2, \bar{T}_3, T_4, T_3, T_4, T_1, \bar{T}_2, \bar{T}_3, \bar{T}_4, T_1, \bar{T}_2\};$$

$$Q^2 = \{\bar{T}_1, \bar{T}_2, T_4, T_1, T_2, \bar{T}_3, T_4, T_1, T_3, \bar{T}_4\};$$

$$Q^3 = \{\bar{T}_1, T_2, \bar{T}_3, \bar{T}_1, \bar{T}_2, T_3, T_4\}.$$

Будем стремиться к тому, чтобы распределить как можно большее число выходных переменных (функций) из множества Y между p выходами ПЛМ и при этом для всех $u = \overline{1, U}$ не нарушить условия $B_u \leq q$ (B_u — число задействованных промежуточных шин в u -й ПЛМ). Представим функции y_1, \dots, y_N из множества Y в ДНФ, и пусть y_i^g есть g -я элементарная конъюнкция переменных T_1, \dots, T_R в ДНФ функции y_i . Обозначим через $n_u(y_i)$ число дополнительных промежуточных шин, которые будут задействованы в u -й ПЛМ при назначении на ее свободный выход функции $y_i \in Y$. Для определения значения $n_u(y_i)$ вычеркнем из ДНФ функции y_i все элементарные конъюнкции y_i^g , которые покрываются хотя бы одной k -й конъюнкцией $T_1^{e_{k1}}, \dots, T_R^{e_{kR}}$ ($e_{kr} \in \{0, 1, -\}$, $r = \overline{1, R}$) множества Q^u и такой, что трюичный вектор (e_{k1}, \dots, e_{kR}) ортогонален любому трюичному вектору $K(a_m)$, $a_m \in A$, для которого $y_i \notin Y(a_m)$.

При выполнении этого условия конъюнкция y_i^g в ДНФ y_i может быть заменена конъюнкцией $T_1^{e_{k1}}, \dots, T_R^{e_{kR}}$ из множества Q^u , т. е. для реализации конъюнкции y_i^g в ПЛМ_u не требуется увеличивать число использованных

промежуточных шин. Тогда значение $n_u(y_i)$ равно числу невычеркнутых элементарных конъюнкций в ДНФ функции y_i . Если проверка рассмотренного условия является слишком трудоемкой задачей, ее можно упростить, вычеркивая из ДНФ функции y_i лишь те элементарные конъюнкции, которые совпадают с элементарными конъюнкциями множества Q^u . Если u -я ПЛМ не имеет свободных выходов, то положим $n_u(y_i) = \infty$ для всех $i = \overline{1, N}$.

Для распределения переменных y_1, \dots, y_N между ПЛМ может быть предложен следующий простой приближенный алгоритм. Среди чисел $n_u(y_i)$, $u = \overline{1, U}$, $i = \overline{1, N}$, выбирается минимальное. Если таких чисел несколько, то среди них выбирается такое, для которого $[B_u + n_u(y_i)] \leq q$. Если таких чисел также несколько, то выбирается любое из них. Когда число $n_u(y_i)$ выбрано, переменная y_i назначается на один из свободных выходов u -й ПЛМ. Если теперь $B_u > q$, то на выходной шине, которой соответствует переменная y_i , реализуется дизъюнкция лишь части элементарных конъюнкций функции y_i с тем, чтобы $B_u = q$; при этом полагается $n_u(y_i) = \infty$, а оставшиеся элементарные конъюнкции функции y_i распределяются, как и ранее. В этом случае переменной y_i будет соответствовать несколько выходов различных ПЛМ, которые в дальнейшем объединяются по схеме «монтажное ИЛИ».

После распределения каждой очередной переменной y_i на выход u -й ПЛМ корректируется множество Q_u . В это множество добавляются новые элементарные конъюнкции переменных T_1, \dots, T_R , которые реализуются на промежуточных шинах u -й ПЛМ при назначении на ее выход переменной y_i . Далее соответствующим образом корректируются значения $n_u(y_j)$, $j = \overline{1, N}$. Аналогично должны быть скорректированы значения $n_u(y_i)$ при распределении переменной y_i на выход u -й ПЛМ и при реализации u -й ПЛМ лишь части элементарных конъюнкций функции y_i . В этом случае $n_u(y_i) = \infty$ и в значениях $n_f(y_i)$, $f = \overline{1, U}$, $f \neq u$, учитывается лишь число еще нереализованных элементарных конъюнкций функции y_i .

Когда все элементарные конъюнкции в ДНФ функции y_i реализованы, положим для всех $u = \overline{1, U}$ $n_u(y_i) = \infty$. Распределение выходных переменных заканчивается, когда для всех $u = \overline{1, U}$, $i = \overline{1, N}$, $n_u(y_i) = \infty$; при этом будут сформированы все множества Y^1, \dots, Y^U . Если $Y \setminus (Y^1 \cup \dots \cup Y^U) \neq \emptyset$, то значения оставшихся выходных переменных формируются на выходах дополнительных ПЛМ (часто для

этих целей выгоднее использовать ПЗУ). Заметим, что если $y_i \in Y^u$, но в u -й ПЛМ реализуется дизъюнкция лишь части элементарных конъюнкций функции y_i , то эта функция также должна быть назначена на один из выходов дополнительной ПЛМ, в которой реализуется оставшаяся часть элементарных конъюнкций функции y_i . Соответствующие выходы основной и дополнительной ПЛМ объединяются по схеме «монтажное ИЛИ».

Процедура распределения выходных переменных по ПЛМ в нашем примере представлена в табл. 6.15. Строки этой таблицы соответствуют выходным переменным $y_1, \dots, \dots, y_{12}$, множествам Q^1, Q^2 и Q^3 , числам p_1, p_2 и p_3 незанятых выходов в каждой ПЛМ после распределения функций возбуждения и числам B_1, B_2 и B_3 промежуточных шин, использованных для реализации конъюнкций функций возбуждения. Во втором столбце таблицы для каждого y_i записаны E_{y_i} — множества кодов состояний автомата, в которых выходной сигнал y_i равен 1, а для каждого Q^u — множества интервалов ($u=1, 2, 3$), соответствующих конъюнкциям из множества Q^u (для краткости будем это множество интервалов также называть Q^u).

Последующие столбцы таблицы соответствуют шагам алгоритма, причем в столбце «0» записаны значения $n_u(y_i)$, p_u и B_u до первого шага. На первых трех шагах алгоритма переменные y_1, y_3 и y_7 распределяются на выходы ПЛМ₁, ПЛМ₃ и вновь ПЛМ₁, так как $n_1(y_1) = n_3(y_3) = n_1(y_7) = 0$. Числа p_1 и p_3 свободных выходов в ПЛМ₁ и ПЛМ₃ уменьшились на два и один соответственно, а B_1 и B_3 не изменились, так как для реализации y_1, y_3 и y_7 не требуются дополнительные промежуточные шины.

На четвертом шаге распределяется переменная y_4 на выход ПЛМ₃, так как равных нулю $n_u(y_i)$ нет, а $n_3(y_4) = 1$. В множество Q^3 добавляется набор $K(a_5) = 1101$, который входит в E_{y_4} , но не покрывается ни одним интервалом из Q^3 , т. е. для соответствующей ему конъюнкции и нужна дополнительная промежуточная шина в ПЛМ₃. Значение p_3 уменьшается на 1 — занимает выход, а B_3 увеличивается на 1 — занимает шина. Значения $n_3(y_i)$ пересчитываются в строках y_2, y_6 и y_{10} , так как именно они записаны в табл. 6.12 рядом с a_5 и y_4 . После введения дополнительной промежуточной шины для y_4 реализация y_2, y_6 и y_{10} потребует на одну [$n_3(y_4) = 1$] шину меньше.

Остальные шаги алгоритма аналогичны. Процедура заканчивается на 12-м шаге распределением последней выходной переменной y_{11} .

Таблица 6.15

y_i	E_{y_i}	$n_{y_i}(y_i)$													
		0	1	2	3	4	5	6	7	8	9	10	11	12	
y_1	1000	0 1 1	ПЛМ ₁	—	—	—	—	—	—	—	—	—	—	—	—
y_2	0010 1001 1101	3 2 3				3 2 2						3 1 2	ПЛМ ₂		
y_3	010—	1 1 0		ПЛМ ₂	—	—	—	—	—	—	—	—	—	—	—
y_4	0011 010— 1101	3 2 1				ПЛМ ₂									
y_5	0110	1 1 1					ПЛМ ₁								
y_6	0010 1101 1110	3 1 3				3 1 2						ПЛМ ₂			
y_7	0011 1011 1000	0 3 2						ПЛМ ₁							
y_8	0010 1011 0110	3 3 3								2 3 3	2 2 3	2 1 3	ПЛМ ₂		
y_9	0010 010—	2 2 1									2 1 1		2 1	ПЛМ ₂	

Продолжение табл. 6.15

y_i	$F y_i$	$n_c(y_i)$																
		0	1	2	3	4	5	6	7	8	9	10	11	12				
y_{10}	1001 1101	2 1 2												2 1	2 1	ПЛМ ₃	—	
y_{11}	1001	1 1 1												1 1	1 1	1 0	ПЛМ ₃	
y_{12}	1011 1110	2 1 2											2 0 2	ПЛМ ₃	—	—	—	
Q^1	0000 0001 —11 10— 1000						0110											
Q^2	00—1 1101 1110											0010	1011					
Q^3	010— 0011									1101					0010	1001		
P_1		4	3	3	2	2	1	1	1	2	1	1	1	1	1	1	1	1
P_2		4	4	4	4	4	4	4	3	2	1	3	2	0	0	0	0	0
P_3		5	5	4	4	4	3	3	3	4	3	3	3	3	2	1	0	0
B_1		11	11	11	11	11	12	12	12	12	12	12	12	12	12	12	12	12
B_2		8	8	8	8	8	8	9	10	10	10	10	10	11	11	11	11	11
B_3		6	6	6	6	7	7	7	7	7	7	7	7	8	8	9	9	9

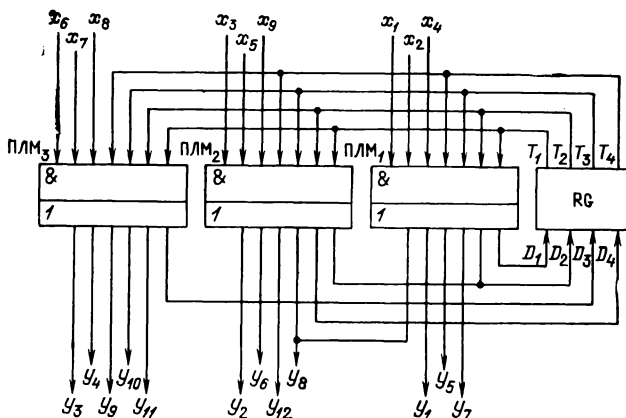


Рис. 6.8

Заметим, что в ПЛМ₁ один выход не используется (после 12-го шага $p_1=1$). Если этот выход соединить с выходом y_8 ПЛМ₂ («монтажное ИЛИ») и реализовать в ПЛМ₁ конъюнкцию $T_1T_2T_3T_4$, входящую в ДНФ функции y_8 , а в ПЛМ₂ — оставшиеся конъюнкции $T_1T_2T_3T_4$ и $T_1T_2T_3T_4$ ДНФ этой функции, то число использованных промежуточных шин в ПЛМ₁ не изменится, а в ПЛМ₂ сократится на одну.

Окончательная одноуровневая схема МПА Мура, построенная по ГСА (см. рис. 6.5), изображена на рис. 6.8. В схеме — три ПЛМ (7, 6, 12), информация об их настрой-

Таблица 6.16

T_1	T_2	T_3	T_4	x_3	x_5	x_9	D_1	D_2	y_1	y_6	y_7
0	0	0	0	1	—	—	.	1	.	.	.
0	0	0	0	0	1	—	1
0	0	0	0	0	0	—	1	1	.	.	.
0	0	1	0	1	—	—	.	1	.	.	.
0	0	1	0	0	—	1	1
0	0	0	1	1	—	—	.	1	.	.	.
0	0	0	1	0	—	0	1
0	0	0	1	0	—	1	1	1	.	.	.
—	—	1	1	—	—	—	1	.	.	.	1
1	0	—	—	—	—	—	1
1	0	0	0	—	—	—	1	1	1	.	1
0	1	1	0	—	—	—	.	.	.	1	.

Таблица 6.17

T_1	T_2	T_3	T_4	x_3	x_5	x_6	D_2	D_4	y_2	y_5	y_6	y_{11}
0	1	0	—	1	0	—	1
0	1	0	—	1	1	—	.	1
1	0	—	1	—	0	0	.	1
1	0	—	1	—	—	1	1	1
1	1	0	0	—	—	0	.	1
0	0	—	1	—	—	—	.	1
1	1	0	1	—	—	—	1	.	1	1	.	.
1	1	1	0	—	—	—	.	1	.	1	.	1
0	0	1	0	—	—	—	.	.	1	1	1	.
1	0	1	1	—	—	—	.	.	1	.	1	1
0	1	1	0	—	—	—	1	.

Таблица 6.18

T_1	T_2	T_3	T_4	x_6	x_7	x_8	D_3	y_3	y_4	y_9	y_{10}	y_{11}
1	0	0	0	1	—	—	1
1	1	0	1	—	0	—	1
0	1	1	0	0	—	—	1
1	1	1	0	—	—	0	1
0	1	0	—	—	—	—	1	1	1	1	.	.
0	0	1	1	—	—	—	1	.	1	.	.	.
1	1	0	1	—	—	—	.	.	1	.	1	.
0	0	1	0	—	—	—	.	.	.	1	.	.
1	0	0	1	—	—	—	1	1

ке сведена в табл. 6.16—6.18 соответственно. Заметим, что синтезированная схема является минимальной в том смысле, что по ГСА на рис. 6.5 невозможно построить схему реализующего ее автомата с меньшим числом ПЛМ (7, 6, 12). В целом можно отметить, что рассмотренный метод позволяет минимизировать дублирование переменных из множеств X , Y и D и обеспечивает хорошее использование входных и выходных шин ПЛМ. Следствием этого является высокое качество синтезированных схем по числу используемых элементов. Аналогичный метод может быть использован и для синтеза по ГСА одноуровневой схемы МПА Мили. Процедура синтеза МПА Мили подробно рассмотрена в [32].

6.5. Синтез многоуровневой схемы автомата

Рассмотренные выше методы синтеза одноуровневых схем достаточно эффективны и имеют широкую область применения. Однако им присущи и некоторые недостатки, которые могут проявиться при синтезе автоматов определенного класса. Отметим некоторые из них.

1. При синтезе МПА Мили по ГСА часто возникает необходимость введения большого числа дополнительных меток для распределения выходных переменных между ПЛМ [32] либо происходит значительное дублирование переменных из множества Y на выходах. В последнем случае коэффициент $k_y = (N+R)/Ut$ использования выходов ПЛМ Р-подсхемы достаточно низок. При синтезе МПА Мура этот недостаток отсутствует, но в некоторых случаях решение ухудшается из-за того, что число внутренних состояний автомата Мура существенно больше числа внутренних состояний автомата Мили, реализующего ту же ГСА.

2. При синтезе МПА с максимально достижимым быстродействием нежелательно применять метод, описанный в § 6.4. При использовании метода из § 6.2 может быть значительное дублирование переменных из множества X на входах ПЛМ, многие входные шины ПЛМ остаются неиспользованными. Другими словами, достаточно низок коэффициент $k_x = L/U(s-R)$ использования входов ПЛМ.

Перечисленные недостатки в ряде случаев приводят к увеличению числа ПЛМ и усложнению общей схемы автомата, чего в значительной степени можно избежать при построении многоуровневых схем МПА. Многоуровневой схема названа потому, что в комбинационной части автомата сигналы от входов x_1, \dots, x_L или T_1, \dots, T_R проходят на выходы y_1, \dots, y_N или D_1, \dots, D_R более чем через одну ПЛМ.

На рис. 6.9 изображена одна из многоуровневых схем, в которой реализуется принцип кодирования микрокоманд [17, 31]. Комбинационная часть этой схемы состоит из двух подсхем — Р и Y. Р-подсхема строится так же, как и ранее, и позволяет сформировать функции возбуждения элементов памяти D_1, \dots, D_R , коды $K(Y_0), K(Y_1), \dots, K(Y_{J-1})$ микрокоманд Y_0, Y_1, \dots, Y_{J-1} , представленные наборами значений переменных из множества $Z = \{z_1, \dots, z_V\}$, и некоторые выходные переменные из множества Y . Микрокоманда Y_j ($j=0, J-1$) — это подмножество микроопераций, записанных в одной операторной вершине Y_j ГСА или в одной строке таблицы переходов МПА в столбце $Y(a_m, a_s)$.

Y-подсхема обеспечивает распознавание некоторых (возможно, всех) микроопераций (выходных переменных МПА) из множеств Y_j на основании кодов $K(Y_j) = (e_{j1}, \dots, e_{jV})$ микрокоманд Y_j ($j=0, J-1$), формируемых P-подсхемой.

Пусть задана ГСА Γ и необходимо синтезировать на ПЛМ (s, t, q) схему МПА Мили с PУ-структурой, показанной на рис. 6.9. Для этого выполняются следующие процедуры.

1. С помощью метода, рассмотренного в § 6.4, по ГСА Γ строятся подтаблицы переходов W_0, W_1, \dots, W_U , для каждой u -й из которых выполняются условия $(L_u + R) \leq s, B_u \leq q, u = \overline{1, U}$. Если в ГСА Γ введены дополнительные метки, то будем считать, что на переходах в соответствующие им состояния формируется пустая микрокоманда $Y_0 = \emptyset$. В подтаблицы переходов W_0, W_1, \dots, W_U введем дополнительный столбец Y_j , куда занесем микрокоманды Y_0, Y_1, \dots, Y_{J-1} , формируемые на каждом переходе в МПА.

2. Состояния МПА кодируются R -разрядными двоичными кодами ($R = \text{int} \log_2 M$) так, как это делалось в § 6.4. В результате определяется число R_u функций возбуждения, значения которых формируются на выходах каждой u -й ПЛМ P-подсхемы ($u = \overline{1, U}$).

3. Аналогично кодированию состояний выполняется кодирование микрокоманд Y_0, Y_1, \dots, Y_{J-1} двоичными кодами разрядности $V = \text{int} \log_2 J$. Целью кодирования является минимизация суммарного числа выходов ПЛМ P-подсхемы, на которых формируются значения переменных z_1, \dots, z_V . В результате определяется число V_u переменных из множества Z , значения которых формируются на выходах каждой u -й ПЛМ P-подсхемы ($u = \overline{1, U}$).

4. Если для некоторой u -й ПЛМ P-подсхемы выполняется условие $(R_u + V_u) > t$, то производится расширение ПЛМ $_u$ по выходам; если $p_u = t - (R_u + V_u) > 0$, то на выходы

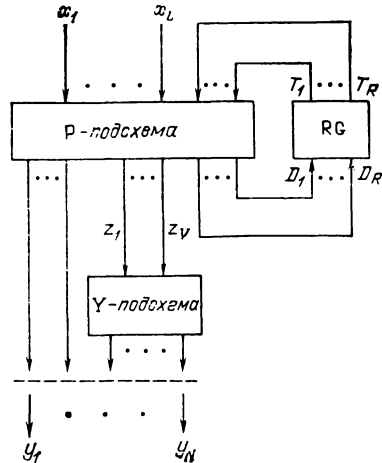


Рис. 6.9

ПЛМ_u распределяются выходные переменные из множества Y .

5. Образуется подмножество $Y(PY)$ множества Y выходных переменных, которые не распределены в п. 4. Значения этих переменных формируются на выходах Y -подсхемы. Заметим, что если $|Y(PY)| \leq V_1 + \dots + V_u$, то сле-

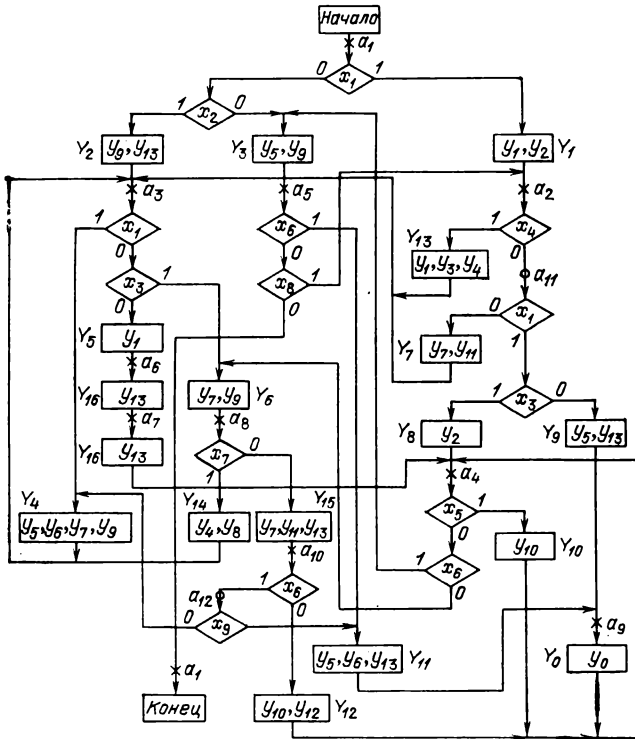


Рис. 6.10

дует попытаться использовать выходы ПЛМ P -подсхемы, где формируются значения переменных z_1, \dots, z_v , для распределения на них всех переменных из множества $Y(PY)$. Если это возможно, то Y -подсхема отсутствует.

На рис. 6.10 задана граф-схема алгоритма* Γ и необходимо синтезировать схему микропрограммного автомата

* Один из путей перехода записан в табл. 6.19 неверно; исправьте ГСА на рис. 6.10 так, чтобы устранить ошибку.

Миля S с PV-структурой на ПЛМ (7, 6, 12). С помощью алгоритма Ф₁ ГСА Г отмечена метками a_1, \dots, a_{10} . Использование метода, изложенного в § 6.4, приводит к введению в ГСА двух дополнительных меток a_{11} и a_{12} , построению четырех подтаблиц переходов W_0, W_1, W_2, W_3 (табл. 6.19)

Таблица 6.19

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_j	a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_j
a_6	a_7	1	y_{13}	Y_{16}	a_4	a_4	x_5	y_{10}	Y_{10}
a_7	a_4	1	y_{13}	Y_{16}		a_5	$\overline{x_5 x_6}$	y_5, y_9	Y_3
						a_8	$\overline{x_5 x_8}$	y_7, y_9	Y_6
a_9	a_4	1	y_0	Y_0	a_5	a_1	$\overline{x_6 x_8}$	y_0	Y_0
a_1	a_2	x_1	y_1, y_2	Y_1		a_9	x_6	y_5, y_6, y_{13}	Y_{11}
						a_{11}	$\overline{x_6 x_8}$	y_0	Y_0
a_3	a_3	$\overline{x_1 x_2}$	y_9, y_{13}	Y_2	a_{10}	a_4	$\overline{x_6}$	y_{10}, y_{12}	Y_{12}
	a_5	$\overline{x_1 x_2}$	y_5, y_9	Y_3		a_{12}	x_6	y_0	Y_0
	a_8	$\overline{x_1 x_3}$	y_5, y_6, y_7, y_9	Y_4		a_2	a_3	x_4	y_1, y_3, y_4
a_3	a_6	$\overline{x_1 x_3}$	y_1	Y_5	a_{11}		$\overline{x_4}$	y_0	Y_0
	a_8	$\overline{x_1 x_3}$	y_7, y_9	Y_6	a_8		a_3	x_7	y_4, y_8
a_{11}	a_3	$\overline{x_1}$	y_7, y_{11}	Y_7		a_{10}	$\overline{x_7}$	y_7, y_{11}, y_{13}	Y_{15}
	a_4	$\overline{x_1 x_3}$	y_2	Y_8	a_9	a_3	$\overline{x_9}$	y_5, y_6, y_7, y_9	Y_4
	a_9	$\overline{x_1 x_3}$	y_5, y_{13}	Y_9		a_9	x_9	y_5, y_6, y_{13}	Y_{11}

и структурных подтаблиц (табл. 6.20). Правила заполнения столбца $F(a_m, a_s)$ табл. 6.20 были описаны в § 6.4. Матрицы E^1, F^1, E^2, F^2 представляются в следующем виде:

$$E^1 = \begin{matrix} & T_1 & T_2 & T_3 & T_4 \\ \begin{pmatrix} 0 & 1 & 1 & 1 \\ 0 & - & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} & ; & F^1 = \begin{matrix} & D_1 & D_2 \\ \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} & ; & \end{matrix} \end{matrix} \quad (6.6)$$

Таблица 6.20

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_j	$F(a_m, a_s)$	$K(Y_j)$	$z(a_m, a_s)$
a_6	0101	a_7	0111	1	y_{13}	Y_{16}	$(D_3 D_3 D_4)$	11001	$(z_1 z_2 z_3)$
a_7	0111	a_4	1110	1	y_{13}	Y_{16}	$(D_1 D_2 D_3)$	11001	$(z_1 z_2 z_3)$
a_9	0—10	a_4	1110	1	y_0	Y_0	$(D_1 D_3 D_3)$	—100	(z_3)
a_1	0000	a_2 a_3 a_6	0100 10— 1100	x_1 $\overline{x_1 x_2}$ $x_1 \overline{x_2}$	y_1, y_2 y_9, y_{13} y_6, y_9	Y_1 Y_3 Y_3	D_2 D_1 $D_1 D_2$	00000 01000 10000	— z_2 z_1
a_3	10—	a_3 a_6 a_8	10— 0101 1101	x_1 $\overline{x_1 x_3}$ $\overline{x_1 x_3}$	y_5, y_6, y_7, y_9 y_1 y_7, y_9	Y_4 Y_6 Y_6	$D_1(D_4)$ $D_2(D_4)$ $D_1 D_2(D_4)$	00001 01001 10001	(z_6) $z_3(z_6)$ $z_1(z_6)$
a_{11}	0001	a_3 a_4 a_9	10— 1110 0—10	$\overline{x_1}$ $x_1 \overline{x_3}$ $x_1 \overline{x_3}$	y_7, y_{11} y_2 y_5, y_{13}	Y_7 Y_8 Y_9	$D_1(D_3)$ $D_1 D_2(D_3)$ (D_3)	10010 01010 00010	$z_1(z_4)$ $z_3(z_4)$ (z_4)

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_j	$F(a_m, a_s)$	$K(Y_j)$	$z(a_m, a_s)$
a_4	1110	a_4	1110	x_5 $\overline{x_5 x_6}$	y_{10}	Y_{10}	$(D_1 D_2) D_3$	10011	$(z_1) z_4 z_5$
		a_5	1100	$\overline{x_5 x_6}$	y_6, y_9	Y_3	$(D_1 D_2)$	10000	(z_1)
		a_8	1101	$\overline{x_5 x_6}$	y_7, y_9	Y_6	$(D_1 D_2) D_4$	10001	$(z_1) z_5$
a_6	1100	a_1	0000	$\overline{x_6 x_8}$	y_0	Y_0	—	—100	(z_3)
		a_9	0—10	x_6	y_6, y_8, y_{13}	Y_{11}	D_3	00101	$(z_3) z_5$
		a_{11}	0001	$\overline{x_6 x_8}$	y_0	Y_0	D_4	—100	(z_3)
a_{10}	0011	a_4	1110	$\overline{x_6}$	y_{10}, y_{13}	Y_{12}	$(D_1 D_3) D_4$	00110	$(z_3) z_4$
		a_{12}	1111	x_6	y_0	Y_0	$(D_1 D_3) D_3 D_4$	—100	(z_3)
a_2	0100	a_3	10—	x_4	y_1, y_3, y_4	Y_{13}	$D_1 (D_4)$	11000	$(z_1 z_2)$
		a_{11}	0001	$\overline{x_4}$	y_0	Y_0	(D_4)	—100	$(z_1 z_2) z_3$
a_8	1101	a_3	10—	x_7	y_4, y_6	Y_{14}	$D_1 (D_3 D_4)$	00011	$(z_4 z_6)$
		a_{10}	0011	$\overline{x_7}$	y_7, y_{11}, y_{13}	Y_{15}	$(D_3 D_4)$	00111	$z_3 (z_4 z_5)$
a_{12}	1111	a_3	10—	$\overline{x_9}$	y_6, y_8, y_7, y_9	Y_4	$D_1 (D_3)$	00001	(z_6)
		a_9	0—10	x_9	y_6, y_8, y_{13}	Y_{11}	(D_3)	00101	$z_3 (z_5)$

$$E^2 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & - & 1 & 0 \\ 1 & 0 & - & - \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right| \end{array}; \quad F^2 = \begin{array}{c} D_3 \ D_4 \\ \left| \begin{array}{cc} 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \end{array} \right| \end{array}. \quad (6.7)$$

Для рассматриваемого примера матрицы E^3 и F^3 отсутствуют, так как все значения функций возбуждения D_1 , записанные в скобках в подтаблицах W_6 и W_2 , представлены в матрицах E^1 , F^1 . После минимизации соответствующих функций матрицы (6.6) и (6.7) запишутся в виде:

$$E^1 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 0 & - & 1 & - \\ 0 & 1 & 0 & 1 \end{array} \right| \end{array}; \quad F^1 = \begin{array}{c} D_1 \ D_2 \\ \left| \begin{array}{cc} 1 & 1 \\ 1 & 1 \\ 0 & 1 \end{array} \right| \end{array};$$

$$E^2 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} - & 1 & - & 1 \\ 0 & 0 & 0 & 1 \\ 0 & - & 1 & 0 \\ 0 & 1 & 0 & - \\ 1 & 1 & 0 & 1 \\ 1 & 0 & - & - \end{array} \right| \end{array}; \quad F^2 = \begin{array}{c} D_3 \ D_4 \\ \left| \begin{array}{cc} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 0 & 1 \end{array} \right| \end{array}.$$

Остановимся более подробно на кодировании микрокоманд. Очевидно, что если удастся, как и выше при кодировании состояний, закодировать микрокоманды так, что значения некоторого разряда кода z_k не будут зависеть от значений входных переменных $X = \{x_1, \dots, x_L\}$ на всех переходах из некоторого состояния a_m , то это значение z_k на указанных переходах можно сформировать на выходе любой ПЛМ независимо от того, какие переменные из множества X поступают на ее входы. Например, если микрокоманды закодированы так, как это показано в карте Карно на рис. 6.11, то не будут зависеть от значений входных переменных значения функций z_3, z_4, z_5 , формируемые на переходах из состояний a_1, a_3 и a_{11} , функций z_1, z_2, z_3 на переходе из a_4 , функций z_1, z_2, z_3, z_4 на

переходе из a_5 и т. д. Заметим, что на рис. 6.11 микрокоманды из одного множества $Y_M(a_m)$ размещены либо в одном столбце, либо в одной строке карты Карно. Здесь $Y_M(a_m)$ — множество микрокоманд, вырабатываемых на всех переходах из состояния a_m ; эти микрокоманды записаны в столбце Y_j в подмассиве переходов из a_m структурной таблицы МПА. Таким образом, здесь также использовано ОЗ-кодирование, причем такое, что в пределах одной подтаблицы W_u , $u=1, \dots, U$ (см. табл. 6.20), от значений входных переменных не зависят значения одних и тех же кодирующих переменных z_K ($K \in \{1, \dots, V\}$).

		$z_3 z_4 z_5$							
		000	001	011	010	110	111	101	100
$z_1 z_2$	00	Y_1	Y_4	Y_{14}	Y_9	Y_{12}	Y_{15}	Y_{11}	Y_0
	01	Y_2	Y_5		Y_8				
	11	Y_{13}	Y_{16}						Y_0
	10	Y_3	Y_6	Y_{10}	Y_7				

Рис. 6.11

Как и при кодировании состояний для каждой подтаблицы W_u , разобьем множество $Z = \{z_1, \dots, z_V\}$ на два подмножества: $Z = G^u \cup H^u$; $G^u \cap H^u = \emptyset$. Через G^u и H^u обозначим подмножества Z , элементы которых принимают соответственно различные и одинаковые значения на переходах из одного состояния в подтаблице W_u . В примере:

$$G^1 = \{z_1, z_2\}, \quad H^1 = \{z_3, z_4, z_5\};$$

$$G^2 = \{z_4, z_5\}, \quad H^2 = \{z_1, z_2, z_3\};$$

$$G^3 = \{z_3\}, \quad H^3 = \{z_1, z_2, z_4, z_5\}.$$

Обозначим через $Z(a_m)$ подмножество множества переменных Z , которые принимают значения 1 на переходах из состояния a_m , и пусть $Z(a_m) = Z_G(a_m) \cup Z_H(a_m)$. Здесь $Z_G(a_m)$ и $Z_H(a_m)$ соответствуют разрядам кода микрокоманды из множеств G_u и H_u , меняющимся и неизменным на переходах из a_m в подтаблице W_u . Например,

$$Z(a_4) = \{z_1, z_4, z_5\}; \quad Z_G(a_4) = \{z_4, z_5\}; \quad Z_H(a_4) = \{z_1\}.$$

Элементы множества $Z_H(a_m)$ заключены в круглые скобки в столбце $Z(a_m, a_s)$ подмассива переходов из a_m . Ясно, что в подтаблице W_0 для всех $a_m \in A_0$ можно записать $Z_G(a_m) = \emptyset$, $Z(a_m) = Z_H(a_m)$ и все элементы в столбце $Z(a_m, a_s)$ — в круглых скобках, поскольку в столбце $X(a_m, a_s)$ нет входных переменных.

Если каждая подтаблица W_u ($u = \overline{1, U}$) будет реализована одной u -й ПЛМ, то значения $z_q \in Z_G(a_m)$ должны формироваться в той ПЛМ, в которой реализованы переходы из a_m . Значения же $z_q \in Z_H(a_m)$, так как они не зависят от значений входных переменных, могут быть сформированы в любой p -й ПЛМ, для которой $z_q \in G^p$.

Функции $z_k = Z_k(T_1, \dots, T_R)$ из множеств $Z_H(a_m)$ можно описать матрицами E^u, F^u ($u = \overline{1, U}$), как это делалось при построении функций возбуждения. Для нашего примера это будут матрицы:

$$E^1 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right|; \quad F^1 = \begin{array}{c} z_1 \ z_2 \\ \left| \begin{array}{cc} 1 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{array} \right|; \end{array} \quad (6.8)$$

$$E^2 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & - & - \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{array} \right|; \quad F^2 = \begin{array}{c} z_4 \ z_5 \\ \left| \begin{array}{cc} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{array} \right|; \end{array} \quad (6.9)$$

$$E^3 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 0 & - & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{array} \right|; \quad F^3 = \begin{array}{c} z_3 \\ \left| \begin{array}{c} 1 \\ 1 \\ 1 \end{array} \right|. \end{array} \quad (6.10)$$

Матрицы (6.8) и (6.9) после минимизации соответствующих булевых функций запишутся в виде:

$$E^1 = \begin{array}{c} T_1 \ T_2 \ T_3 \ T_4 \\ \left| \begin{array}{cccc} 1 & 1 & 1 & 0 \\ 0 & 1 & - & - \end{array} \right|; \quad F^1 = \begin{array}{c} z_1 \ z_2 \\ \left| \begin{array}{cc} 1 & 0 \\ 1 & 1 \end{array} \right|;$$

$$E^2 = \begin{matrix} & T_1 & T_2 & T_3 & T_4 \\ \begin{matrix} 1 & 0 & - & - \\ - & 1 & - & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \end{matrix} & ; & F^2 = \begin{matrix} & z_4 & z_5 \\ \begin{matrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{matrix} \end{matrix}.$$

Заметим, что функции, заданные матрицами (6.6), (6.8), а также (6.7) и (6.9), следует минимизировать совместно, поскольку они реализуются одной ПЛМ. Другими словами, необходимо найти кратчайшую ДНФ функций D_1 , D_2 , z_1 , z_2 и D_3 , D_4 , z_4 , z_5 .

Поставим в соответствие каждой микрокоманде Y_j конъюнкцию Z_j переменных z_1, \dots, z_V , равную 1 тогда и только тогда, когда формируется микрокоманда Y_j . Если (e_{j1}, \dots, e_{jV}) — код микрокоманды Y_j , $e_{jk} \in \{0, 1, -\}$, $k = \overline{1, V}$, то $Z_j = z_1^{e_{j1}} \dots z_V^{e_{jV}}$, где $z^0 = \bar{z}$, $z^1 = z$, $z^- = 1$. Тогда выходные переменные (функции) МПА могут быть представлены в ДНФ следующего вида:

$$y_i = \bigvee_{\substack{j=0 \\ y_i \in Y_j}}^{j-1} z_1^{e_{j1}} \dots z_V^{e_{jV}}. \quad (6.11)$$

Запишем ДНФ некоторых функций из табл. 6.20:

$$\begin{aligned} y_1 &= \overline{z_1 z_2 z_3 z_4 z_5} \vee \overline{z_1 z_2 z_3 z_4 z_5} \vee z_1 z_2 \overline{z_3 z_4 z_5}; \\ y_5 &= z_1 \overline{z_2 z_3 z_4 z_5} \vee \overline{z_1 z_2 z_3 z_4 z_5} \vee \overline{z_1 z_2 z_3 z_4 z_5} \vee \\ &\vee z_1 \overline{z_2 z_3 z_4 z_5} \vee \overline{z_1 z_2 z_3 z_4 z_5} \vee \overline{z_1 z_2 z_3 z_4 z_5}; \\ y_8 &= \overline{z_1 z_2 z_3 z_4 z_5}. \end{aligned}$$

Функции (6.11) могут быть использованы для построения Y -подсхемы. Целесообразно предварительно выполнить их доопределение на наборах значений переменных z_1, \dots, z_V , соответствующих незаполненным клеткам карты Карно на рис. 6.11, и совместную минимизацию с целью получения кратчайшей ДНФ.

Для построения Y -подсхемы на одной ПЛМ (s, t, q) должны быть выполнены условия: $V \leq s$, $N_Y \leq t$, $B(Y) \leq q$, где $B(Y)$ — число различных элементарных конъюнкций в ДНФ функций y_1, \dots, y_N вида (6.11), $N_Y = |Y(PY)|$. Усло-

вие $V \leq s$ практически всегда выполняется. Если $N_Y > t$ и (или) $B(Y) > q$, то Y -подсхема строится на нескольких ПЛМ и для ее синтеза используются методы гл. 5. Заметим, что функции вида (6.11) представлены в совершенной ДНФ либо близкой к ней форме. Близость понимается в том смысле, что если эти функции и представлены в произвольной ДНФ, то ее расширение до совершенной ДНФ приводит к несущественному увеличению длины исходной ДНФ. Поэтому в качестве элементов Y -подсхемы выгодно использовать ПЗУ (s, t) , что исключает необходимость минимизации системы функций (6.11). Условие $V \leq s$ для ПЗУ (s, t) , так же как и для ПЛМ (s, t, q) , почти всегда выполняется. Если $N_Y > t$, то производится расширение ПЗУ (s, t) по выходам.

Рассмотрим теперь способ распределения выходных переменных из множества Y по свободным выходам ПЛМ P -подсхемы. Число p таких выходов равно $p = p_1 + \dots + p_U$, где p_u — число свободных выходов u -й ПЛМ. Обозначим через Q^u множество элементарных конъюнкций переменных из множества $\{x_1, \dots, x_L, T_1, \dots, T_R\}$, соответствующих всем строкам u -й структурной подтаблицы W_u и дополнительно реализованных на промежуточных шинах u -й ПЛМ P -подсхемы по результатам кодирования внутренних состояний и микрокоманд ($u = \overline{1, U}$). Представим все функции $y_i(x_1, \dots, x_L, T_1, \dots, T_R)$, $i = \overline{1, N}$, в ДНФ, и пусть $E(y_i)$ — множество элементарных конъюнкций в ДНФ функции y_i , $E^u(y_i) = E(y_i) \cap Q^u$; $J(y_i)$ — число непустых множеств среди $E^1(y_i), \dots, E^U(y_i)$.

Формирование множеств Y^1, \dots, Y^U выходных переменных, реализуемых на выходах ПЛМ₁, ..., ПЛМ_U P -подсхемы, осуществляется следующим образом:

1. Среди всех чисел $J(y_i)$, $i = \overline{1, N}$, выбирается минимальное $J(y_i)$ и проверяется возможность распределения функции y_i на выход P -подсхемы. Это допустимо лишь в том случае, если все конъюнкции из множества $E(y_i)$ могут быть реализованы на промежуточных шинах ПЛМ₁, ..., ПЛМ_U и на некотором выходе y_l P -подсхемы может быть реализована дизъюнкция всех элементарных конъюнкций из множества $E(y_i)$ по схеме «монтажное ИЛИ». Если функция y_l может быть распределена на выход P -подсхемы, то для всех $u = \overline{1, U}$ при $E^u(y_l) \neq \emptyset$ переменная y_l включается в множество Y^u . Если функция y_l не может быть распределена на выход P -подсхемы, то полагается $J(y_l) = \infty$.

2. Если на некотором шаге имеется несколько одинаковых чисел: $J(y_g), J(y_k), \dots, J(y_d)$, — то из множества $\{y_g, y_k, \dots, y_d\}$ выбирается переменная y_i , которая может быть распределена на выход Р-подсхемы (т. е. не нарушаются требования п. 1), и после этого $\sum_{u=1}^U B_u = \min$. Здесь B_u —

число использованных промежуточных шин в u -й ПЛМ Р-подсхемы.

3. Если выполнение п. 2 дает одинаковые результаты для переменных y_g, y_k, \dots, y_d , то выбирается та из них y_i , для которой $|E(y_i)| = \max$.

4. Если выполнение п. 3 дает одинаковые результаты для переменных y_g, y_k, \dots, y_d , то выбирается любая из них.

Пусть на некотором шаге для всех $i = \overline{1, N}$ числа $J(y_i) = \infty$. Если при этом все выходы ПЛМ Р-подсхемы заняты, то формирование множеств Y^1, \dots, Y^U заканчивается и $Y(PY) = Y \setminus (Y^1 \cup \dots \cup Y^U)$. Если при $J(y_i) = \infty$ для всех $i = \overline{1, N}$ некоторые выходы ПЛМ Р-подсхемы свободны, то на них могут быть частично реализованы некоторые функции из множества $Y(PY)$ с целью сокращения числа $B(Y)$. Значения этих функций будут формироваться на соответствующих выходах ПЛМ Р- и Y-подсхем, которые объединяются по схеме «монтажное ИЛИ». Для сокращения числа $B(Y)$ последовательно просматриваются все непустые подмножества переменных $Y_j \cap Y(PY)$, $j = \overline{0, J-1}$. Пусть микрокоманда Y_j записана в i, l, \dots, k -й (и только в этих) строках структурных подтаблиц. Тогда если значения переменных из множества $Y_j \cap Y(PY)$ для i, l, \dots, k -й строк структурных подтаблиц будут сформированы на выходах ПЛМ Р-подсхемы, то конъюнкцию $Z_j = z_1^{e_j^1} \dots z_V^{e_j^V}$ можно не реализовать на промежуточных шинах ПЛМ Y-подсхемы и число $B(Y)$ сокращается на 1. Очевидно, не имеет смысла анализировать множества $Y_j \cap Y(PY)$ с мощностью, большей, чем число свободных выходных шин у ПЛМ Р-подсхемы. Среди оставшихся множеств в первую очередь просматриваются те, для которых значение $N_j = |Y_j \cap Y(PY)|$ минимально. Процедура заканчивается, когда либо все выходы ПЛМ Р-подсхемы будут заняты, либо число $B(Y)$ не может быть уменьшено.

Продолжим решение задачи синтеза МПА Мили с РY-структурой по ГСА, изображенной на рис. 6.10. Для

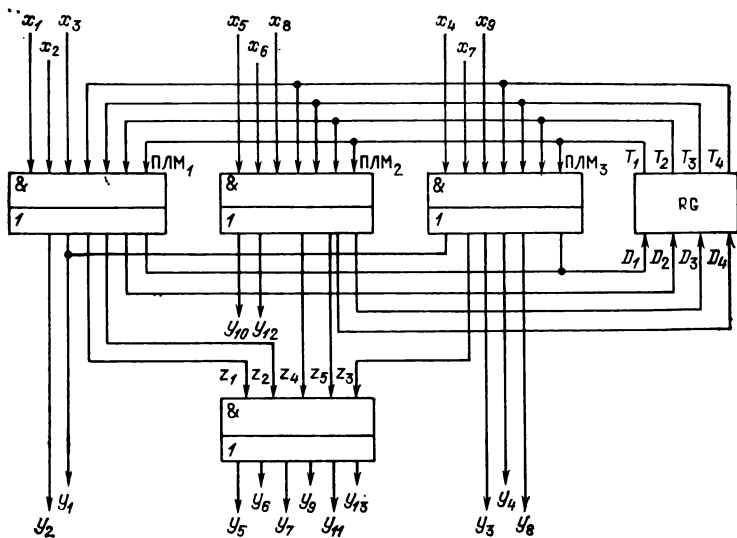


Рис. 6.12

нашего примера $J(y_2) = J(y_3) = J(y_4) = J(y_8) = J(y_{10}) = J(y_{12}) = 1$; $J(y_1) = J(y_{11}) = 2$; $J(y_5) = J(y_6) = J(y_7) = J(y_9) = J(y_{13}) = 3$. Тогда переменные y_1, y_2 распределяются на свободные выходы ПЛМ₁, y_{10}, y_{12} — на ПЛМ₂ и y_1, y_3, y_4, y_8 — на ПЛМ₃. Выходы y_1 ПЛМ₁ и ПЛМ₃ объ-

Таблица 6.21

T_1	T_2	T_3	T_4	x_1	x_2	x_3	D_1	D_2	z_1	z_2	y_1	y_2
0	0	0	0	1	—	—	.	1	.	.	1	1
0	0	0	0	0	1	—	1	.	.	1	.	.
0	0	0	0	0	0	—	1	1	1	.	.	.
1	0	—	—	1	—	—	1
1	0	—	—	0	—	0	.	1	.	1	1	.
1	0	—	—	0	—	1	1	1	1	.	.	.
0	0	0	1	0	—	—	1	.	1	.	.	.
0	0	0	1	1	—	1	1	1	.	1	.	1
1	1	1	0	—	—	—	1	1	1	.	.	.
0	—	1	—	—	—	—	1	1
0	1	0	1	—	—	—	.	1
0	1	—	—	—	—	—	.	.	1	1	.	.

Таблица 6.22

T_1	T_2	T_3	T_4	x_5	x_6	x_8	D_3	D_4	z_4	z_5	y_{10}	y_{12}
1	1	1	0	1	—	—	1	.	1	1	1	.
1	1	1	0	0	0	—	.	1	.	1	.	.
1	1	0	0	—	1	—	1	.	.	1	.	.
1	1	0	0	—	0	1	.	1
0	0	1	1	—	0	—	.	1	1	.	1	1
0	0	1	1	—	1	—	1	1
1	0	—	—	—	—	—	.	1	.	1	.	.
—	1	—	1	—	—	—	1	.	.	1	.	.
0	0	0	1	—	—	—	1	.	1	.	.	.
1	1	0	1	—	—	—	1	1	1	1	.	.
0	—	1	0	—	—	—	1
0	1	0	—	—	—	—	.	1

Таблица 6.23

T_1	T_2	T_3	T_4	x_4	x_7	x_9	D_1	z_3	y_1	y_2	y_4	y_5
0	1	0	0	1	—	—	1	.	1	1	1	.
0	1	0	0	0	—	—	.	1
1	1	0	1	—	1	—	1	.	.	.	1	1
1	1	0	1	—	0	—	.	1
1	1	1	1	—	—	0	1
1	1	1	1	—	—	1	.	1
0	—	1	0	—	—	—	.	1
1	1	0	0	—	—	—	.	1
0	0	1	1	—	—	—	.	1

Таблица 6.24

z_1	z_2	z_3	z_4	z_5	y_5	y_6	y_7	y_9	y_{11}	y_{13}
1	1	0	0	1	1
0	1	0	0	0	.	.	.	1	.	1
1	0	0	0	0	1	.	.	1	.	.
0	0	0	0	1	1	1	1	1	.	.
1	0	0	0	1	.	.	1	1	.	.
1	0	0	1	0	.	.	1	.	1	.
0	0	0	1	0	1	1
0	0	1	0	1	1	1	.	.	.	1
0	0	1	1	1	.	.	1	.	1	1

единяются по схеме «монтажное ИЛИ». После этого ПЛМ₁—ПЛМ₃ не имеют свободных выходов и $Y(PY) = \{y_5, y_6, y_7, y_9, y_{11}, y_{13}\}$ [функции $y_1, y_2, y_3, y_4, y_8, y_{10}, y_{12}$ исключаются из системы (6.11)], $B(Y) = 9 < q, N_Y = 6 = t$. Информация для настройки ПЛМ₁—ПЛМ₃ Р-подсхемы и ПЛМ₄ Y-подсхемы приведена соответственно в табл. 6.21—6.24. Минимизация БФ, реализуемых Y-подсхемой не выполнялась.

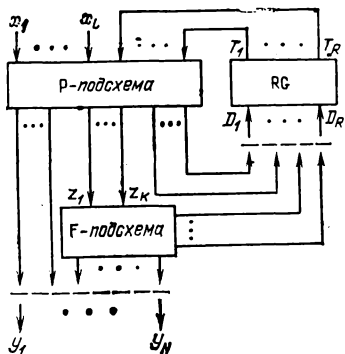


Рис. 6.13

Структура полученной логической схемы автомата показана на рис. 6.12.

Сравним теперь полученные результаты с результатами, приведенными в § 6.4. Пусть заданная ГСА Г (см. рис. 6.10) предварительно отмечена метками по алгоритму Φ_1 и $R=4$. Тогда согласно выражению (6.5) $U=3$. Однако применение

любого из известных методов приводит к тому, что $U > 3$. Причины этого были пояснены в начале этого параграфа. Отметим теперь ГСА, изображенную на рис. 6.10, по алгоритму Φ_2 , тогда $R=5$ и согласно выражению (6.5) $U=5$.

В целом можно сказать, что автоматы Мили с PY-структурой целесообразно синтезировать по ГСА, для которых применение метода, изложенного в § 6.4, после предварительной отметки ГСА по алгоритмам Φ_1 и Φ_2 приводит к тому, что $T(\Phi_1) < T(\Phi_2)$. Здесь $T(\Phi_k)$ — число подтаблиц переходов, полученных с помощью метода, описанного в § 6.4 после предварительной отметки ГСА по алгоритму $\Phi_k, k \in \{1, 2\}$. Без подробных пояснений укажем, что схема МПА Мура с PY-структурой не имеет каких-либо преимуществ по сравнению с одноуровневой схемой. Поэтому в настоящем параграфе она не рассматривается.

При синтезе МПА с PY-структурой часто дублируется большое число переменных из множества DUZ на выходах Р-подсхемы, что приводит к увеличению числа ПЛМ. Ниже рассматриваются многоуровневые схемы автоматов с PY-структурой [31, 35], в которых реализуется принцип кодирования строк структурной таблицы. При этом число

дублирование переменных на выходах ПЛМ существенно сокращается, что в ряде случаев приводит к упрощению логической схемы МПА. Комбинационная часть таких автоматов (рис. 6.13) состоит из двух подсхем: Р и F. Р-подсхема строится как и ранее, и позволяет сформировать коды $K(c_1), \dots, K(c_B)$ строк c_1, \dots, c_B всех структурных подтаблиц, представленные наборами значений переменных из множества $Z = \{z_1, \dots, z_K\}$, а также значения некоторых выходных переменных и функций возбуждения из множества $Y \cup D$; F-подсхема обеспечивает распознавание некоторых (возможно всех) переменных из множества $Y \cup D$ на основании кодов $K(c_i) = (e_{i1}, \dots, e_{iK})$ строк c_i , в которых записаны соответствующие переменные ($i = \overline{1, B}$).

Пусть задана ГСА Γ и необходимо синтезировать на ПЛМ (s, t, q) схему МПА Мили с РФ-структурой. Для этого выполняются следующие действия:

1. С помощью метода, рассмотренного в § 6.4, по ГСА Γ строятся подтаблицы переходов W_0, W_1, \dots, W_U , для каждой u -й из которых выполняются условия $(L_u + R) \leq s, B_u \leq q, u = \overline{1, U}$.

2. Строки подтаблиц W_0, W_1, \dots, W_U переходов МПА кодируются K -разрядными двоичными кодами ($K = \lceil \log_2 B \rceil, B = \sum_{u=0}^U B_u$) так, как это делалось в § 6.4 при

кодировании состояний. Цель кодирования — минимизация суммарного числа выходов ПЛМ Р-подсхемы, на которых формируются значения переменных z_1, \dots, z_K . Коды любых двух строк должны быть различны независимо от того, находятся указанные строки в одной или в разных подтаблицах. В результате определяется число K_u переменных из множества Z , значения которых формируются на выходах каждой u -й ПЛМ Р-подсхемы ($u = \overline{1, U}$).

3. Кодирование внутренних состояний автомата.

4. Если для некоторой u -й ПЛМ Р-подсхемы выполняется условие $K_u > t$, то производится расширение ПЛМ $_u$ по выходам (указанная ситуация практически не встречается); если же $p_u = (t - K_u) > 0$, то на выходы ПЛМ $_u$ распределяются выходные переменные и функции возбуждения из множества $Y \cup D$.

5. На выходах F-подсхемы формируются значения выходных переменных и функций возбуждения из множества $Y(PF)$, которые не распределены при выполнении п. 4; $Y(PF) \subseteq Y \cup D$.

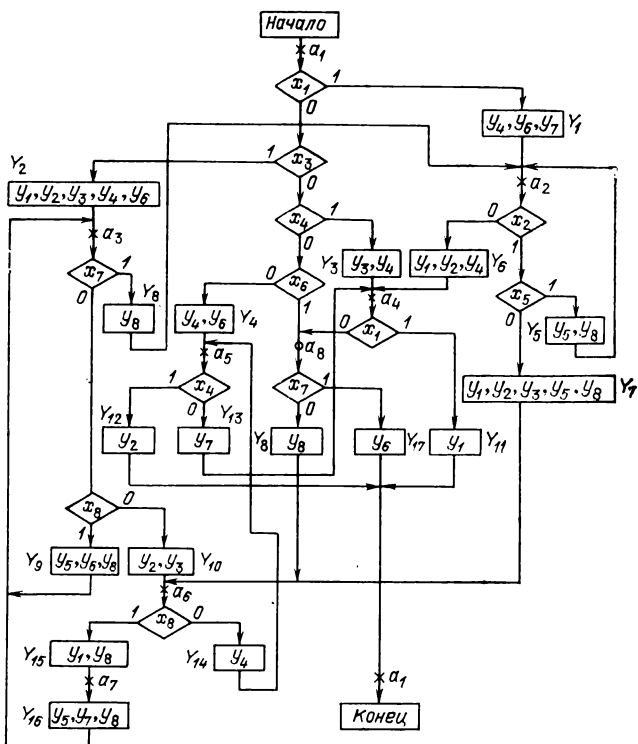


Рис. 6.14

Решение задач, перечисленных в пп. 1—5, производится с помощью методов, рассмотренных ранее. Применение этих методов проиллюстрируем примером. Рассмотрим ГСА Г, показанную на рис. 6.14, и синтезируем МПА S на ПЛМ (7, 6, 12), реализующий эту ГСА. Отметим ГСА Г метками a_1, \dots, a_7 по алгоритму Φ_1 . Использование метода, изложенного в § 6.4, приводит к построению трех подтаблиц переходов W_0, W_1, W_2 и введению в ГСА дополнительной метки a_8 .

В табл. 6.25 представлены подтаблицы W_0, W_1, W_2 , причем подтаблица W_0 — верхняя, а W_2 — нижняя. Поставим в соответствие каждой строке табл. 6.25 символ c_i ($i = \overline{1, B}$; $B = B_0 + B_1 + B_2 = 20$), который запишем в специально выделенный столбец c_i . Пусть строки c_1, \dots, c_{20} МПА закодированы так, как это представлено в карте

Таблица 6.25

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	Y_j	c_l
a_7	a_3	1	y_6, y_7, y_8	Y_{16}	c_1
a_1	a_2	x_1	y_4, y_6, y_7	Y_1	c_2
	a_3	$\overline{x_1 x_3}$	y_1, y_2, y_3, y_4, y_6	Y_2	c_3
	a_4	$\overline{x_1 x_3 x_4}$	y_3, y_4	Y_3	c_4
	a_5	$\overline{x_1 x_3 x_4 x_6}$	y_4, y_6	Y_4	c_5
	a_8	$\overline{x_1 x_3 x_4 x_6}$	y_0	Y_0	c_6
a_4	a_1	x_1	y_1	Y_{11}	c_7
	a_8	$\overline{x_1}$	y_0	Y_0	c_8
a_5	a_1	x_4	y_2	Y_{12}	c_9
	a_4	$\overline{x_4}$	y_7	Y_{13}	c_{10}
a_2	a_2	$x_2 x_5$	y_5, y_8	Y_5	c_{11}
	a_4	$\overline{x_2}$	y_1, y_2, y_4	Y_6	c_{12}
	a_8	$\overline{x_2 x_5}$	y_1, y_2, y_3, y_5, y_8	Y_7	c_{13}
a_3	a_2	x_7	y_8	Y_8	c_{14}
	a_3	$\overline{x_7 x_8}$	y_5, y_6, y_8	Y_9	c_{15}
	a_6	$\overline{x_7 x_8}$	y_2, y_3	Y_{10}	c_{16}
a_6	a_5	$\overline{x_8}$	y_4	Y_{14}	c_{17}
	a_7	x_8	y_1, y_8	Y_{15}	c_{18}
a_8	a_1	x_7	y_6	Y_{17}	c_{19}
	a_8	$\overline{x_7}$	y_8	Y_8	c_{20}

Карно на рис. 6.15. Эти коды занесены в специальный столбец $K(c_i)$ структурной таблицы (табл. 6.26), построенной по табл. 6.25.

Обозначим через G^u и H^u подмножества $Z = \{z_1, \dots, z_K\}$, элементы которых принимают соответственно различные и одинаковые значения на переходах в подтаблице W_u ; $G^u \subseteq Z$, $H^u \subseteq Z$, $G^u \cup H^u = Z$, $G^u \cap H^u = \emptyset$. В примере:

$$G^1 = \{z_1, z_2, z_3\}; \quad H^1 = \{z_4, z_5\};$$

$$G^2 = \{z_4, z_5\}; \quad H^2 = \{z_1, z_2, z_3\}.$$

Пусть $Z(a_m)$ — подмножество множества $Z = \{z_1, \dots, z_K\}$ переменных, единичные значения которых формируются на всех переходах из состояния a_m : $Z(a_m) = Z_G(a_m) \cup \cup Z_H(a_m)$, где переменные из множества $Z_G(a_m)$ соответствуют разрядам кода строки из множества G^u , а $Z_H(a_m)$ — из множества H^u ($u = \overline{1, U}$). Функции из множеств $Z(a_m)$, $a_m \in A$, занесем в специально выделенный столбец $Z(a_m, a_s)$ табл. 6.26. Правила формирования значений этих функций в каждой строке таблицы полностью

					T_3	T_4	T_5			
		000	001	011	010	110	111	101	100	
T_1	T_2	00	c_2	c_7	c_1	c_9	c_{10}		c_8	c_3
		01	c_4	c_{19}		c_{20}				c_5
		11	c_{14}	c_{15}		c_{16}			c_{18}	c_{17}
		10	c_6				c_{13}		c_{12}	c_{11}

Рис. 6.15

совпадают с аналогичными правилами, изложенными в § 6.4. Разница состоит лишь в том, что рассматриваются не коды внутренних состояний, записанные в столбце $K(a_s)$, а коды строк, записанные в столбце $K(c_i)$. Так, в i -й строке u -й подтаблицы переходов W_u записываются в скобках те функции Z_j , у которых нижний индекс совпадает с номером позиции единицы в разрядах кода строки из множества H^u , записанного в этой же строке, а без скобок — те функции Z_j , у которых нижний индекс совпадает с номером позиции единицы в разрядах кода строки из множества G^u , записанного в этой же i -й строке. Функции из множеств $Z_H(a_m)$ описываются матрицами E^u, F^u ($u = \overline{1, U}$), которые строятся точно так же, как и в § 6.4. Поскольку внутренние состояния на данном этапе еще не закодированы, то договоримся записывать в строках матриц E^u ($u = \overline{1, U}$) вместо кодов $K(a_m)$, представленных наборами значений переменных T_1, \dots, T_R , абстрактные символы a_m , т. е. каждому набору значений переменных из множества $Z_H(a_m)$ будет поставлено в соответствие состояние a_m , в котором формируется данный набор. После кодирования внутренних состояний автомата символы a_m в матрицах E^u ($u = \overline{1, U}$) будут замены соот-

Таблица 6.26

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_{III}, a_s)$	$Y(a_{II}, a_s)$	Y_j	$F(a_{II}, a_s)$	c_f	$K(c_f)$	$Z(a_{II}, a_s)$
a_7	110	a_3	010	1	y_5, y_7, y_8	Y_{16}	D_2	c_1	00011	$(z_4 z_5)$
a_1	000	a_2	001	$\frac{x_1}{x_1 x_3}$	y_4, y_6, y_7	Y_1	D_3	c_2	00000	—
		a_3	010	$\frac{x_1 x_3}{x_1 x_3 x_4}$	y_1, y_2, y_3, y_4, y_6	Y_2	D_2	c_3	00100	z_3
		a_4	011	$\frac{x_1 x_3 x_4}{x_1 x_3 x_4 x_6}$	y_3, y_4	Y_3	$D_2 D_3$	c_4	01000	z_2
		a_5	100	$\frac{x_1 x_3 x_4 x_6}{x_1 x_3 x_4 x_6}$	y_4, y_6	Y_4	D_1	c_5	01100	$z_2 z_3$
a_4	011	a_8	111	$\frac{x_1 x_3 x_4 x_6}{x_1 x_3 x_4 x_6}$	y_0	Y_0	$D_1 D_2 D_3$	c_6	10000	z_1
		a_1	000	$\frac{x_1}{x_1}$	y_1	Y_{11}	—	c_7	00001	(z_6)
		a_8	111	$\frac{x_1}{x_1}$	y_0	Y_0	$D_1 D_2 D_3$	c_8	00101	$z_3(z_5)$
a_5	100	a_1	000	$\frac{x_4}{x_4}$	y_2	Y_{12}	—	c_9	00010	(z_4)
		a_4	011	$\frac{x_4}{x_4}$	y_7	Y_{13}	$D_3 D_3$	c_{10}	00110	$z_3(z_4)$
a_2	001	a_2	001	$\frac{x_2 x_6}{x_2 x_6}$	y_5, y_8	Y_5	D_3	c_{11}	10100	$(z_1 z_3)$
		a_4	011	$\frac{x_2}{x_2}$	y_1, y_2, y_4	Y_6	$D_2 D_3$	c_{12}	10101	$(z_1 z_3) z_5$
		a_6	101	$\frac{x_2 x_6}{x_2 x_6}$	y_1, y_2, y_3, y_5, y_8	Y_7	$D_1 D_3$	c_{13}	10110	$(z_1 z_3) z_4$
a_3	010	a_2	001	x_7	y_8	Y_8	D_3	c_{14}	11000	$(z_1 z_2)$
		a_3	010	$\frac{x_7 x_8}{x_7 x_8}$	y_5, y_6, y_8	Y_9	D_2	c_{15}	11011	$(z_1 z_2) z_5$
		a_8	101	$\frac{x_7 x_8}{x_7 x_8}$	y_2, y_3	Y_{10}	$D_1 D_3$	c_{16}	11010	$(z_1 z_2) z_4$
a_6	101	a_5	100	$\frac{x_8}{x_8}$	y_4	Y_{14}	D_1	c_{17}	11100	$(z_1 z_2 z_3)$
		a_7	110	x_8	y_1, y_6	Y_{15}	$D_1 D_2$	c_{18}	11101	$(z_1 z_2 z_3) z_5$
a_8	111	a_1	000	$\frac{x_7}{x_7}$	y_6	Y_{17}	—	c_{19}	01001	$(z_2) z_5$
		a_6	101	$\frac{x_7}{x_7}$	y_8	Y_8	$D_1 D_3$	c_{20}	01010	$(z_2) z_4$

ветствующими кодами $K(a_m)$. Для нашего примера матрицы E^1, F^1, E^2, F^2 запишутся в виде:

$$E^1 = \begin{vmatrix} a_m \\ a_2 \\ a_3 \\ a_6 \\ a_8 \end{vmatrix}; \quad F^1 = \begin{vmatrix} z_1 & z_2 & z_3 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix}; \quad E^2 = \begin{vmatrix} a_7 \\ a_4 \\ a_5 \end{vmatrix}; \quad F^2 = \begin{vmatrix} z_4 & z_5 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{vmatrix}. \quad (6.12)$$

На следующем этапе решается задача кодирования внутренних состояний автомата. Пусть заданы матрицы E^u и F^u , $u=1, \overline{U}$. Состояниям автомата необходимо назначить такие коды, чтобы эти матрицы после минимизации соответствующих функций z_1, \dots, z_K с учетом их допустимого доопределения имели бы минимально возможное число строк. Это способствует сокращению числа использованных промежуточных шин в ПЛМ₁, ..., ПЛМ_U. Указанная задача широко известна и существуют различные подходы к ее решению (см., например, [36]). Здесь мы не будем останавливаться на рассмотрении способов выбора кодов внутренних состояний и закодируем их произвольным образом, например так, как это сделано в табл. 6.26. После этого матрицы (6.12) запишутся в виде:

$$E^1 = \begin{vmatrix} T_1 & T_2 & T_3 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{vmatrix}; \quad F^1 = \begin{vmatrix} z_1 & z_2 & z_3 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{vmatrix};$$

$$E^2 = \begin{vmatrix} T_1 & T_2 & T_3 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{vmatrix}; \quad F^2 = \begin{vmatrix} z_4 & z_5 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{vmatrix}.$$

Функции z_1, z_2, z_3 могут быть минимизированы и представлены в кратчайшей ДНФ вида:

$$z_1 = \overline{T_2}T_3 \vee \overline{T_1}T_2\overline{T_3}; \quad z_2 = T_1T_3 \vee \overline{T_1}T_2\overline{T_3}; \quad z_3 = \overline{T_2}T_3.$$

Поставим в соответствие каждой строке c_i ($i=1, \overline{B}$) конъюнкцию Z_i переменных z_1, \dots, z_K , равную единице, тогда и только тогда, когда в МПА выполняется переход, описываемый строкой c_i . Если (e_{i1}, \dots, e_{iK}) — код строки c_i , $e_{ik} \in \{0, 1, -\}$, $k=1, \overline{K}$, то $Z_i = z_1^{e_{i1}} \dots z_K^{e_{iK}}$, где $z_k^0 =$

$= \bar{z}_k, z_k^1 = z_k, z_k^- = 1$. Если переменная $y_j (D_r)$ записана в строке c_i , то будем использовать обозначение $y_j \in c_i (D_r \in c_i)$. Тогда выходные переменные и функции возбуждения МПА могут быть представлены в ДНФ следующего вида:

$$y_j = \bigvee_{\substack{i=1 \\ y_j \in c_i}}^B z_1^{e_{i1}} \dots z_K^{e_{iK}}, \quad D_r = \bigvee_{\substack{i=1 \\ D_r \in c_i}}^B z_1^{e_{i1}} \dots z_K^{e_{iK}}. \quad (6.13)$$

Запишем ДНФ некоторых функций из табл. 6.26:

$$y_1 = \bar{z}_1 \bar{z}_2 z_3 \bar{z}_4 z_5 \vee \bar{z}_1 \bar{z}_2 \bar{z}_3 \bar{z}_4 z_5 \vee z_1 \bar{z}_2 z_3 \bar{z}_4 z_5 \vee z_1 \bar{z}_2 z_3 z_4 \bar{z}_5 \vee z_1 z_2 z_3 \bar{z}_4 z_5;$$

$$D_2 = \bar{z}_1 \bar{z}_2 \bar{z}_3 z_4 z_5 \vee \bar{z}_1 \bar{z}_2 z_3 \bar{z}_4 z_5 \vee \bar{z}_1 z_2 \bar{z}_3 \bar{z}_4 \bar{z}_5 \vee z_1 \bar{z}_2 z_3 \bar{z}_4 \bar{z}_5 \vee$$

$$\vee \bar{z}_1 \bar{z}_2 z_3 \bar{z}_4 z_5 \vee \bar{z}_1 \bar{z}_2 z_3 z_4 \bar{z}_5 \vee z_1 \bar{z}_2 z_3 \bar{z}_4 z_5 \vee z_1 z_2 \bar{z}_3 \bar{z}_4 z_5 \vee z_1 z_2 z_3 \bar{z}_4 z_5.$$

Функции (6.13) могут быть использованы для построения F-подсхемы. Целесообразно предварительно выполнить их доопределение на наборах значений переменных z_1, \dots, z_K , соответствующих незаполненным клеткам карты Карно на рис. 6.15, и совместную минимизацию с целью получения кратчайшей ДНФ.

Для построения F-подсхемы на одной ПЛМ (s, t, q) должны быть выполнены условия $K \leq s, N_F \leq t, B(F) \leq q$, где $B(F)$ — число различных элементарных конъюнкций в ДНФ функций $y_1, \dots, y_N, D_1, \dots, D_R$ вида (6.13); $N_F = |Y(PF)|$. Среди приведенных условий наиболее часто не выполняются два последних, т. е. $N_F \leq t$ и $B(F) \leq q$. Поскольку функции вида (6.13) представлены в совершенной ДНФ или близкой к ней форме, для построения F-подсхемы выгодно использовать ПЗУ (s, t), что исключает необходимость учета условия $B(F) \leq q$. Если $N_F > t$, то производится расширение ПЗУ (s, t), по выходам. Условие же $K \leq s$ как для ПЗУ (s, t), так и для ПЛМ (s, t, q) почти всегда выполняется.

Формирование множеств Y^1, \dots, Y^v переменных, реализуемых на выходах ПЛМ₁, ..., ПЛМ_v P-подсхемы, осуществляется точно так же, как и ранее. Единственное отличие заключается в том, что в эти множества могут быть включены не только выходные переменные y_1, \dots, y_N , но и функции возбуждения D_1, \dots, D_R . Для рассматриваемого примера: $J(y_5) = J(y_7) = J(y_8) = 1; J(y_1) = J(y_2) = J(y_3) = J(y_4) = J(y_6) = J(D_1) = J(D_2) = J(D_3) = 2$. Тогда

переменные y_3, y_6 и y_7 распределяются на свободные выходы ПЛМ₁, а y_3, y_5, y_6, y_8 — на свободные выходы ПЛМ₂ Р-подсхемы. Выходы y_3 , а также y_6 ПЛМ₁ и ПЛМ₂ объединяются по схеме «монтажное ИЛИ». После этого ПЛМ₁ и ПЛМ₂ Р-подсхемы не имеют свободных выходов и $Y(PF) = \{y_1, y_2, y_4, D_1, D_2, D_3\}$, $Y^1 = \{y_3, y_6, y_7\}$, $Y^2 = \{y_3, y_5, y_6, y_8\}$. Функции y_3, y_5, y_6, y_7, y_8 должны быть исключены из системы (6.13). Поскольку $B(F) = 19 > q$, $N_F = 6 = t$, $K = 5 < s$, то F-подсхема строится на двух ПЛМ (7, 6, 12) с расширением по промежуточным шинам.

Таблица 6.27

T_1	T_2	T_3	x_1	x_3	x_4	x_6	z_1	z_2	z_3	y_3	y_6	y_7
0	0	0	1	—	—	—	1	1
0	0	0	0	1	—	—	.	.	1	1	1	.
0	0	0	0	0	1	—	.	1	.	1	.	.
0	0	0	0	0	0	0	.	1	1	.	1	.
0	0	0	0	0	0	1	1
0	1	1	0	—	—	—	.	.	1	.	.	.
1	0	0	—	—	0	—	.	.	1	.	.	1
—	0	1	—	—	—	—	1	.	1	.	.	.
0	1	0	—	—	—	—	1	1
1	—	1	—	—	—	—	.	1
1	1	0	—	—	—	—	1

Таблица 6.29

Адрес	D_1	D_2	D_3	y_1	y_2	y_4
0	.	.	1	.	.	1
1	.	.	.	1	.	.
2	1	.
3	1
4	.	1	.	1	1	1
5	1	1	1	.	.	.
6	.	1	1	.	.	.
8	.	1	1	.	.	1
10	1	.	1	.	.	.
12	1	1
16	1	1	1	.	.	.
20	.	.	1	.	.	.
21	.	1	1	1	1	1
22	1	.	1	1	1	.
24	.	.	1	.	.	.
25	.	1
26	1	.	1	.	1	.
28	1	1
29	1	1	.	1	.	.

Таблица 6.28

T_1	T_2	T_3	x_2	x_5	x_7	x_8	z_4	z_5	y_3	y_5	y_6	y_8
0	0	1	1	1	—	—	.	.	.	1	.	1
0	0	1	0	—	—	—	.	1
0	0	1	1	0	—	—	1	.	1	1	.	1
0	1	0	—	—	1	—	1
0	1	0	—	—	0	1	.	1	.	1	1	1
0	1	0	—	—	0	0	1	.	1	.	.	.
1	0	1	—	—	—	1	.	1	.	.	.	1
1	1	1	—	—	1	—	.	1	.	.	1	.
1	1	1	—	—	0	—	1	1
1	1	0	—	—	—	—	1	1	.	1	.	1
0	1	1	—	—	—	—	.	1
1	0	0	—	—	—	—	1

Пусть для построения F-подсхемы использованы ПЗУ (5, 6). Тогда с учетом того, что $N_F=6=t$, $K=5=s$, F-подсхема строится на одном таком ПЗУ. Информация для настройки ПЛМ₁ и ПЛМ₂ R-подсхемы и ПЗУ F-подсхемы представлена соответственно в табл. 6.27—6.29. В табл. 6.29 использовано упрощенное изображение ПЗУ (5, 6), у которого матрица M_2 описана так же, как и матрица M_2 ПЛМ, а номера выходов дешифратора, связан-

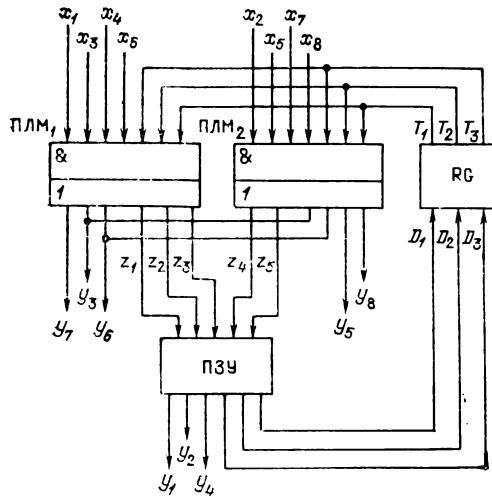


Рис. 6.16

ные со входами матрицы M_2 , указаны в столбце «Адрес». Для простоты из табл. 6.29 исключены строки, содержащие в столбцах $D_1, D_2, D_3, y_1, y_2, y_4$ только точки. В табл. 6.27 дописана строка, соответствующая элементарной конъюнкции $T_1 T_2 \bar{T}_3$, для формирования значения функции y_7 в подтаблице W_0 . Структура полученной логической схемы автомата показана на рис. 6.16.

По аналогии с ранее изложенным, если при $J(y_i) = \infty$ и $J(D_r) = \infty, i=1, N, r=1, R$, некоторые выходы ПЛМ R-подсхемы свободны, то на них могут быть частично реализованы отдельные функции из множества $Y(PF) = (Y \cup D) \setminus (Y \cup \dots \cup Y_V)$ с целью уменьшения числа $B(F)$.

Критерии оценки целесообразности построения МПА Мили с PF-структурой совпадают с аналогичными критериями для автоматов с PY-структурой. При выборе спо-

соста реализации автоматов с РУ- или с РF-структурой последние оказываются предпочтительнее,

$$\left(R + \sum_{u=1}^U K_u \right) < \sum_{u=1}^U (V_u + R_u).$$

Здесь предполагается, что F-подсхема может быть построена на ПЗУ (s, t) , для которых $s \geq K$. Без подробных пояснений заметим, что схема МПА Мура с РF-структурой не имеет каких-либо преимуществ по сравнению с аналогичной одноуровневой схемой, и в связи с этим здесь она не рассматривается. В заключение отметим, что многоуровневые схемы других типов будут рассмотрены в гл. 8.

ГЛАВА СЕДЬМАЯ.

СИНТЕЗ АВТОМАТОВ НА ПРОГРАММИРУЕМЫХ ЛОГИЧЕСКИХ МАТРИЦАХ С ПАМЯТЬЮ

7.1. Тривиальная реализация

Напомним, что ПЛМ (s, t, q, r) с памятью, в отличие от ПЛМ (s, t, q) , содержит r -разрядный внутренний регистр RG в обратной связи между матрицами M_2 и M_1 . Любой МПА с числами L, N, B и R входных переменных, микроопераций, строк в таблице переходов и элементов памяти, удовлетворяющий выражениям

$$L \leq s, N \leq t, B \leq q, R \leq r, \quad (7.1)$$

реализуется тривиальным образом на одной ПЛМ (s, t, q, r) . При такой реализации все s входов и t выходов ПЛМ (s, t, q, r) могут быть использованы для входных и выходных переменных МПА, тогда как при реализации на ПЛМ (s, t, q) R входов и R выходов отводилось для связи с элементами памяти МПА, для которых требовались дополнительные схемы. Очевидно, что применение ПЛМ (s, t, q, r) при реализации МПА в ряде случаев может оказаться эффективнее, чем применение ПЛМ (s, t, q) .

Если не выполняется хотя бы одно из условий в выражении (7.1), то для построения МПА потребуется несколько соединенных между собой ПЛМ (s, t, q, r) . Когда в МПА только $N > t$, а $L \leq s, B \leq q$ и $R \leq r$, его реализация возможна на основе расширения ПЛМ (s, t, q, r) по выходам в $k = \lceil N/t \rceil$ раз, которое, как и при расширении

по выходам ПЛМ (s, t, q) , сводится к объединению одноименных входов всех k ПЛМ (s, t, q, r) . Матрицы M_1 у всех ПЛМ (s, t, q, r) реализуют одни и те же термы (число их $B \leq q$), а на kt выходах матриц M_2 реализуются микрооперации из множества $Y = \{y_1, \dots, y_N\}$. Это множество должно быть произвольным образом разбито на $k = \lfloor N/t \rfloor$ блоков, содержащих не более t элементов. Микрооперации каждого блока реализуются в одной из k ПЛМ (s, t, q, r) .

Пусть, например, на ПЛМ $(10, 10, 20, 5)$ необходимо реализовать некоторый МПА с параметрами $L=10$, $N=18$, $B=20$, заданный некоторой структурной таблицей A . Так как необходимо расширение ПЛМ $(10, 10, 20, 5)$ по выходам, разобьем множество микроопераций $Y = \{y_1, \dots, y_{18}\}$ на два блока: $Y^1 = \{y_1, \dots, y_{10}\}$ и $Y^2 = \{y_{11}, \dots, y_{18}\}$. Представим себе две новые структурные таблицы (B и C), которые отличаются от таблицы A и друг от друга только микрооперациями в столбце $Y(a_m, a_s)$. В таблице B в столбце $Y(a_m, a_s)$ запишем микрооперации из множества Y^1 , а в таблице C — из множества Y^2 . Микрооперации в столбцах $Y(a_m, a_s)$ таблиц B и C должны быть записаны в тех же строках, в которых они встречались в таблице A . Например, если в таблице A на некотором переходе вырабатываются микрооперации y_1, y_3, y_{12} , то в соответствующей строке таблицы B в столбце $Y(a_m, a_s)$ будут записаны y_1, y_3 , а в точно такой же строке таблицы C записано y_{12} . В результате получаются два почти идентичных МПА, заданных таблицами B и C и отличающихся только столбцами выходных сигналов. Если один из них переходит из состояния a_m в состояние a_s под действием входного сигнала $X(a_m, a_s)$ с выдачей микрокоманды $Y_1(a_m, a_s) \subseteq Y^1$, то в другом также будет переход (a_m, a_s) под действием $X(a_m, a_s)$, но с выдачей микрокоманды $Y_2(a_m, a_s) \subseteq Y^2$. Ясно, что каждый из этих МПА может быть реализован тривиальным образом на одной ПЛМ $(10, 10, 20, 5)$ непосредственно по структурным таблицам B и C .

Если в МПА число входных переменных больше числа входов в ПЛМ ($L > s$), число строк структурной таблицы больше числа промежуточных шин в ПЛМ ($B > q$) или разрядность памяти больше числа элементов памяти ПЛМ ($R > r$), то реализация такого МПА на ПЛМ (s, t, q, r) возможна лишь после его декомпозиции на более простые автоматы, каждый из которых может быть реализован на одной ПЛМ.

7.2. Декомпозиция микропрограммного автомата

Под декомпозицией автомата принято понимать построение для заданного автомата S сети C , состоящей из связанных между собой компонентных автоматов. Каждой сети C можно сопоставить ее результирующий автомат S_C . Сеть C будет результатом декомпозиции автомата S , если у S_C есть подавтомат, изоморфный S .

В [37] предложен простой метод декомпозиции МПА, основанный на следующих соображениях:

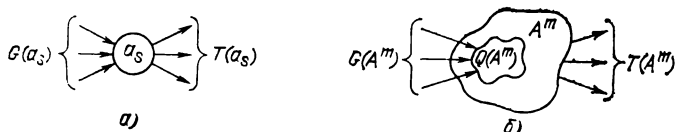


Рис. 7.1

1) в результате декомпозиции должна получиться сеть из компонентных автоматов такая, что длина таблицы переходов исходного автомата соизмерима с суммарной длиной таблиц переходов компонентных автоматов;

2) в результате декомпозиции должна получиться сеть из последовательно работающих компонентных автоматов.

Декомпозицию МПА Мили проиллюстрируем примером автомата, заданного табл. 6.3.

Пусть на множестве A состояний декомпозируемого автомата $S = (A, X, Y, \delta, \lambda, a_1)$ определено разбиение $\pi = \{A^1, \dots, A^n\}$. На вопросе выбора такого разбиения остановимся подробнее после рассмотрения процедуры декомпозиции. В нашем примере положим $\pi = a_1, a_5, a_6; a_2, a_3, a_9; a_4, a_7, a_8$, т. е. $A^1 = \{a_1, a_5, a_6\}$, $A^2 = \{a_2, a_3, a_9\}$, $A^3 = \{a_4, a_7, a_8\}$. Как и ранее, через $X(a_s)$ будем обозначать множество входных переменных, встречающихся в столбце $X(a_m, a_s)$ таблицы переходов МПА на всех переходах из состояния a_s . Точно так же, $Y(a_s)$ — множество выходных переменных (микроопераций), записанных в столбце $Y(a_m, a_s)$ на всех переходах из a_s . Обозначим через $G(a_s)$ множество других состояний, $a_s \notin G(a_s)$, из которых есть переходы в состояние a_s , а через $T(a_s)$ — множество состояний, в которые есть переходы из состояния a_s , $a_s \notin T(a_s)$. Смысл множеств $G(a_s)$ и $T(a_s)$ иллюстрируется рис. 7.1, а:

$$G(a_s) = \{a_i | \delta(a_i, X_h) = a_s, i \neq s\};$$

$$T(a_s) = \{a_j | \delta(a_s, X_f) = a_j, j \neq s\},$$

где $X_h = X(a_i, a_s)$; $X_f = X(a_s, a_j)$. Например, для МПА из табл. 6.3:

$$X(a_3) = \{x_4, x_{10}\}; Y(a_3) = \{y_3, y_5, y_8, y_9, y_{10}, y_{15}\};$$

$$G(a_3) = \{a_1, a_2, a_9\}; T(a_3) = \{a_8, a_9\}.$$

Для блока A^m разбиения π введем аналогичные обозначения:

$$X(A^m) = \bigcup_{a_s \in A^m} X(a_s);$$

$$Y(A^m) = \bigcup_{a_s \in A^m} Y(a_s).$$

Таким образом, $X(A^m)$ и $Y(A^m)$ — множества входных и выходных переменных, встречающихся на всех переходах из всех состояний блока A^m в таблице переходов МПА.

Таблица 7.1

A^m	$Q(A^m)$	$G(A^m)$	$T(A^m)$
a_1, a_5, a_6 a_2, a_3, a_9 a_4, a_7, a_8	a_6 a_2, a_3 a_8	a_4 a_1, a_7 a_3	a_3 a_8 a_2, a_6

Пусть $G(A^m)$ — множество состояний, не вошедших в A^m , из которых есть переходы в состояния блока A^m ; $T(A^m)$ — множество состояний, не вошедших в A^m , в которые есть переходы из состояний блока A^m ; $Q(A^m)$ — множество состояний блока A^m , в которые есть переходы из состояний других блоков [состояний множества $G(A^m)$] (рис. 7.1,б):

$$G(A^m) = \{a_i \notin A^m \mid \delta(a_i, X_h) = a_s, a_s \in A^m\};$$

$$T(A^m) = \{a_j \notin A^m \mid \delta(a_s, X_f) = a_j, a_s \in A^m\};$$

$$Q(A^m) = \{a_s \in A^m \mid \delta(a_i, X_h) = a_s, a_i \notin A^m\},$$

где $X_h = X(a_i, a_s)$; $X_f = X(a_s, a_j)$. Эти множества для каждого блока рассматриваемого примера сведены в табл. 7.1. Микропрограммному автомату S и разбиению π поставим в соответствие сеть из связанных между собой компонентных МПА S^1, \dots, S^n , число которых равно числу блоков в разбиении π . В примере сеть будет включать

три МПА: S^1 , S^2 и S^3 . Компонентный автомат $S^m = (B^m, X^m, Y^m, \delta^m, \lambda^m, a_{m1})$, $m = \overline{1, n}$, определим следующим образом:

1. Образует множество B^m состояний МПА S^m , добавив к m -му блоку A^m разбиения π одно дополнительное состояние b_m , т. е. $B^m = A^m \cup \{b_m\}$. Таким образом, в каждом компонентном автомате число состояний будет на единицу больше числа состояний в соответствующем блоке π . В примере: $B^1 = \{b_1, a_1, a_5, a_6\}$; $B^2 = \{b_2, a_2, a_3, a_9\}$; $B^3 = \{b_3, a_4, a_7, a_8\}$.

2. Определим входные переменные компонентного автомата S^m

$$X^m = X(A^m) \cup Z_x^m. \quad (7.2)$$

Здесь Z_x^m — дополнительные переменные, связывающие по входу автомат S^m с другими компонентными автоматами. Для нахождения Z_x^m поставим в соответствие каждому состоянию $a_s \in Q(A^m)$ входную переменную z_s компонентного МПА S^m с тем же индексом s . Тогда, как будет показано ниже, если в исходном МПА был переход из $a_i \notin A^m$ в $a_s \in A^m$, то в компонентном автомате, в котором есть состояние a_i , должен быть выработан выходной сигнал (микроперация) z_s , который заставит автомат S^m перейти в состояние a_s :

$$Z_x^m = \{z_s \mid \delta(a_i, X_h) = a_s; a_s \in A^m; a_i \notin A^m\},$$

где $X_h = X(a_i, a_s)$. Непосредственно по столбцу $Q(a_m)$ табл. 7.1 получим: $Z_x^1 = \{z_6\}$; $Z_x^2 = \{z_2, z_3\}$; $Z_x^3 = \{z_8\}$. В соответствии с (7.2):

$$X^1 = \{x_1, x_2, x_3, x_5, x_6\};$$

$$X^2 = \{x_4, x_6, x_{10}, x_{11}, z_2, z_3\};$$

$$X^3 = \{x_7, x_8, x_9, z_8\}.$$

3. Определим выходные переменные компонентного автомата

$$Y^m = Y(A^m) \cup Z_y^m. \quad (7.3)$$

Здесь Z_y^m — дополнительные переменные, связывающие по выходу автомат S^m с другими компонентными автоматами. Для нахождения Z_y^m поставим в соответствие каждому состоянию $a_j \in T(A^m)$ выходную переменную z_j компонентного МПА S^m с тем же индексом j :

$$Z_y^m = \{z_j \mid \delta(a_s, X_f) = a_j; a_s \in A^m; a_j \notin A^m\},$$

где $X_j = X(a_s, a_j)$. Непосредственно по столбцу $T(A^m)$ табл. 7.1 получим: $Z_{y^1} = \{z_3\}$; $Z_{y^2} = \{z_8\}$; $Z_{y^3} = \{z_2, z_6\}$. В соответствии с (7.3)

$$Y^1 = \{y_1, y_2, y_7, y_{11}, y_{12}, y_{16}, y_{17}, z_3\};$$

$$Y^2 = \{y_3, y_5, y_8, y_9, y_{10}, y_{15}, z_8\};$$

$$Y^3 = \{y_4, y_6, y_{13}, y_{14}, y_{18}, z_2, z_6\}.$$

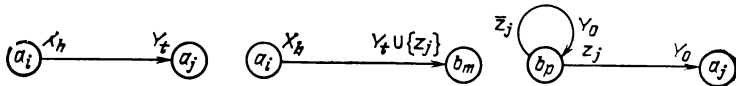


Рис. 7.2

4. Определим функции переходов и выходов компонентного автомата S^m . Пусть в исходном автомате S есть переход из a_i в a_j под действием входного сигнала X_h с выдачей выходного сигнала Y_t (левый подграф на рис. 7.2):

$$\delta(a_i, X_h) = a_j; \quad \lambda(a_i, X_h) = Y_t$$

и пусть $a_i \in A^m$, т. е. a_i — состояние компонентного автомата S^m . Возможны два случая:

а) $a_j \in A^m$, т. е. a_j — то же состояние компонентного автомата S^m . Положим:

$$\delta^m(a_i, X_h) = \delta(a_i, X_h) = a_j; \quad \lambda^m(a_i, X_h) = \lambda(a_i, X_h) = Y_t,$$

тогда соответствующий этому переходу подграф в автомате S^m совпадает с левым подграфом автомата S на рис. 7.2.

В нашем примере в третьей строке табл. 6.3 записан переход из a_1 в a_6 под действием сигнала $x_1 x_2 \bar{x}_3$ с выдачей микрокоманды $Y_7 = \{y_2, y_{12}\}$. Так как a_1 и a_6 — элементы блока A^1 разбиения π , то в компонентном автомате S^1 будет точно такой же переход из a_1 в a_6 :

$$\delta^1(a_1, x_1 x_2 \bar{x}_3) = a_6;$$

$$\lambda^1(a_1, x_1 x_2 \bar{x}_3) = \{y_2, y_{12}\};$$

б) $a_j \in A^p$ ($p \neq m$), т. е. a_j — состояние компонентного автомата S^p . Таким образом, a_i и a_j находятся в разных компонентных автоматах. Будем говорить, что в этом случае компонентный автомат S^m возбуждает компонентный автомат S^p . Определим в автомате S^m функции переходов и выходов (средний подграф на рис. 7.2):

$$\delta^m(a_i, X_h) = b_m; \quad \lambda^m(a_i, X_h) = Y_t \cup \{z_j\}, \quad (7.4)$$

где z_j — дополнительная микрооперация, вырабатываемая в S^m и являющаяся входным сигналом для S^p .

В автомате S^p (правый подграф на рис. 7.2):

$$\delta^p(b_p, z_j) = \delta(a_i, X_h) = a_j; \lambda^p(b_p, z_j) = Y_0. \quad (7.5)$$

Здесь Y_0 — пустая микрокоманда; ей соответствует векторный выходной сигнал, все компоненты которого равны 0. Индекс j микрооперации z_j равен номеру состояния a_j компонентного автомата S^p .

В нашем примере в четвертой строке табл. 6.3 записан переход из a_1 в a_3 под действием сигнала $x_1x_2x_3$ с выдачей микрокоманды $Y_1 = \{y_1, y_2\}$. Так как $a_1 \in A^1$, а $a_3 \in A^2$, то в компонентном автомате S^1 будет переход в дополнительное состояние b_1 под действием того же выходного сигнала $x_1x_2x_3$. На выходе МПА S^1 выдаст ту же микрокоманду $\{y_1, y_2\}$ и дополнительную микрооперацию z_3 , номер которой совпадает с номером состояния перехода a_3 :

$$\delta^1(a_1, x_1x_2x_3) = b_1; \lambda^1(a_1, x_1x_2x_3) = \{y_1, y_2, z_3\}.$$

В соответствии с выражением (7.5) в компонентном автомате S^2 будет переход из дополнительного состояния b_2 в состояние a_3 под действием входного сигнала z_3 с выдачей пустой микрокоманды Y_0 .

Таким образом, число элементов в множестве дополнительных входных переменных Z_x^m в компонентном автомате S^m равно числу состояний в множестве $Q(A^m)$, в которые в таблице переходов декомпозируемого МПА S есть переходы из состояний, не вошедших в A^m . Число элементов в множестве дополнительных выходных переменных Z_y^m в компонентном автомате S^m равно числу состояний в множестве $T(A^m)$, в которые в таблице переходов декомпозируемого МПА S есть переходы из состояний множества A^m . Очевидно, что

$$\bigcup_{i=1}^n Z_x^i = \bigcup_{i=1}^n Z_y^i.$$

5. Начальное состояние a_{m1} компонентного автомата S^m определим следующим образом:

$$a_{m1} = \begin{cases} a_1, & \text{если } a_1 \in A^m; \\ b_m & \text{в противном случае.} \end{cases}$$

Таким образом, в начальный момент времени все компонентные автоматы находятся в состояниях типа b_m , за исключением того автомата, у которого начальное состоя-

Таблица 7.2

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	h
a_1	a_1	$\overline{x_1 x_2}$	$y_1 y_2$	1
	a_5	$\overline{x_1}$	$y_1 y_2 y_{12}$	2
	a_6	$\overline{x_1 x_2 x_3}$	$y_2 y_{12}$	3
	b_1	$\overline{x_1 x_2 x_3}$	$y_1 y_2 z_3$	4
a_5	a_5	$\overline{x_1}$	$y_7 y_{11}$	5
	a_6	$\overline{x_1}$	$y_{16} y_{17}$	6
a_6	a_1	$\overline{x_5}$	$y_1 y_2$	7
	a_1	$\overline{x_5}$	$y_{16} y_{17}$	8
b_1	a_6	$\overline{z_6}$	—	9
	b_1	$\overline{z_8}$	—	10

Таблица 7.3

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	h
a_2	a_2	$\overline{x_6}$	—	1
	a_3	$\overline{x_6}$	$y_3 y_5$	2
a_3	a_3	$\overline{x_{10}}$	$y_3 y_5$	3
	b_2	$\overline{x_4 x_{10}}$	$y_5 y_8 y_9 z_8$	4
	a_9	$\overline{x_4 x_{10}}$	$y_{10} y_{15}$	5
a_9	a_2	$\overline{x_6 x_{11}}$	$y_5 y_8 y_9$	6
	a_3	$\overline{x_{11}}$	$y_3 y_5 y_9$	7
	a_9	$\overline{x_6 x_{11}}$	$y_{10} y_{15}$	8
b_2	a_3	$\overline{z_3}$	—	9
	a_2	$\overline{z_2}$	—	10
	b_2	$\overline{z_3 z_2}$	—	11

ние совпадает с начальным состоянием декомпозируемого автомата S .

Таблицы переходов компонентных автоматов S^1 , S^2 и S^3 приведены в табл. 7.2, 7.3, 7.4 соответственно. Сеть C из трех компонентных автоматов изображена на рис. 7.3. Построение таблиц переходов компонентных автоматов очевидно. Строки таблицы переходов декомпозируемого МПА S расписываются последовательно, одна за другой, по таблицам переходов компонентных автоматов. Пусть на некотором t -м шаге анализируется t -я строка таблицы переходов автомата S , в которой записан переход

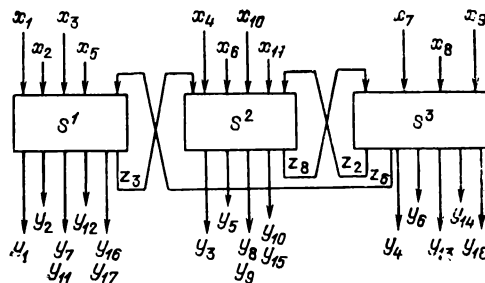


Рис. 7.3

$a_i a_j X_h Y_t$. Если состояние a_i принадлежит множеству A^m компонентного МПА S^m , то эта строка переписывается в таблицу переходов автомата S^m . Если при этом a_i и a_j в разных компонентных автоматах ($a_i \in A^m, a_j \in A^p$), то в строке таблицы переходов компонентного автомата S^m в столбец $Y(a_m, a_s)$ добавляется микрооперация z_j , а в таблице переходов компонентного автомата S^p записывается переход из b_p в a_j под действием сигнала z_j с выдачей

пустой микрокоманды $Y_0 = \emptyset$. Очевидно, что число

строк B_m в таблице переходов компонентного автомата S^m равно

$$B_m = B(A^m) + |Q(A^m)| + 1, \quad (7.6)$$

где $B(A^m)$ — суммарное число строк в переходах из состояний множества A^m в таблице переходов декомпозируемого автомата.

Из самого способа декомпозиции ясно, что каждый раз не может возбуждаться более одного автомата, т. е. компонентные автоматы работают последовательно, за исключением

моментов возбуждения. Тогда, если автомат S^m возбужден (находится в состоянии типа a_i), все другие компонентные автоматы находятся в состояниях типа b_p ($p=1, \dots, m-1, m+1, \dots, n$). Таким образом, всегда справедливо равенство $a_i a_j = 0$ при $a_i \in A^m$ и $a_j \in A^p$. Здесь, как и ранее, a_i и a_j — конъюнкции переменных T_1, \dots, T_R , соответствующие состояниям a_i и a_j .

Если число состояний $Q(A^m)$ блока A^m , в которые в декомпозируемом автомате есть переходы из состояний других блоков, более двух, т. е. $|Q(A^m)| > 2$, то целесообразно кодировать входные сигналы, под действием которых осуществляются переходы из состояния b_m в состояния из множества $Q(A^m)$. Пусть, например, $Q(A^m) = \{a_1, \dots, a_R\}$. Введем множество двоичных входных переменных $U^m = \{u_1, \dots, u_F\}$, где $F = \text{intlog}_2(R+1)$. Закодируем каждое состояние a_r множества $Q(A^m)$ двоичным ко-

Т а б л и ц а 7.4

a_m	a_s	$X(a_m, a_s)$	$Y(a_m, a_s)$	h
a_4	a_4	$\overline{x_7 x_9}$	$y_{13} y_{18}$	1
	b_3	x_7	$y_{13} y_6$	2
	a_8	$\overline{x_7 x_9}$	$y_{13} y_{14}$	3
a_7	b_3	x_8	$y_{14} y_{18} z_2$	4
	a_4	$\overline{x_8}$	$y_{13} y_{18}$	5
a_8	a_4	$\overline{x_9}$	y_6	6
	a_7	x_9	$y_4 y_6$	7
b_3	a_8	$\overline{z_8}$	—	8
	b_3	$\overline{z_8}$	—	9

дом длины F :

$$K(a_r) = (e_{r1}, \dots, e_{rF}), e_{rf} \in \{0, 1\}, f = \overline{1, F}.$$

Тогда если в декомпозируемом автомате S есть переход из $a_i \in A^p$ в $a_r \in A^m$ ($p \neq m$) под действием входного сигнала X_h , то в компонентном автомате S^m будет переход из b_m в a_r под действием входного сигнала (e_{r1}, \dots, e_{rF}) , а в его структурной таблице вместо входного сигнала z_r будет записана конъюнкция

$$I_r = u_1^{e_{r1}} \dots u_F^{e_{rF}} = \bigwedge_{f=1}^F u_f^{e_{rf}}, u_f^0 = \overline{u_f}, u_f^1 = u_f.$$

Такой же, но уже выходной сигнал выдаст компонентный автомат S^p при переходе из состояния a_i в состояние b_p под действием входного сигнала X_h . Один из входных наборов значений переменных u_1, \dots, u_F , например нулевой, резервируется для входного сигнала, который оставляет автомат S^m в состоянии b_m , когда возбужден другой компонентный автомат.

7.3. Синтез микропрограммного автомата

Если в микропрограммном автомате $S = (A, X, Y, \delta, \lambda, a_1)$ число L входных переменных, число R элементов памяти или число B строк в структурной таблице превышает соответственно число s, r или q в ПЛМ (s, t, q, r) , то реализация МПА S возможна с использованием описанных выше методов декомпозиции, в результате которой для заданного автомата S по выбранному разбиению $\pi = \{A^1, \dots, A^n\}$ на множестве его состояний строится сеть из компонентных автоматов $S^m = (B^m, X^m, Y^m, \delta^m, \lambda^m, a_{m1})$, $m = \overline{1, n}$, каждый из которых затем реализуется на ПЛМ (s, t, q, r) . Это возможно, если выполняются следующие условия:

1) число входных переменных в каждом компонентном МПА S^m , включая дополнительные входные переменные от других компонентных автоматов, не превосходит числа s входов в ПЛМ (s, t, q, r) ;

2) число выходных переменных (микроопераций) в каждом МПА S^m , включая дополнительные выходные переменные для возбуждения других компонентных автоматов, не превосходит числа t выходов в ПЛМ (s, t, q, r) ;

3) число B_m строк структурной таблицы каждого компонентного МПА S^m не превосходит числа q промежуточных шин в ПЛМ (s, t, q, r) ;

4) число R_m элементов памяти в каждом компонентном МПА S^m не превосходит числа r элементов памяти в ПЛМ (s, t, q, r) .

Таким образом, задача синтеза МПА S на ПЛМ (s, t, q, r) сводится к нахождению минимального по числу блоков разбиения $\pi = \{A^1, \dots, A^n\}$, каждый блок A^m которого определяет компонентный автомат S^m ($m = \overline{1, n}$) такой, что

$$|X^m| \leq s; \quad (7.7)$$

$$|Y^m| \leq t; \quad (7.8)$$

$$B_m \leq q; \quad (7.9)$$

$$R_m \leq r. \quad (7.10)$$

Рассмотрим приближенный последовательный алгоритм поиска разбиения π . Изложение будем сопровождать примером синтеза МПА, заданного табл. 6.3, на ПЛМ (10, 10, 20, 5) с десятью входами, десятью выходами, двадцатью промежуточными шинами и пятью D -триггерами. По табл. 6.3 построим табл. 7.5, в которой каждому состоянию a_i поставим в соответствие: множества $X(a_i)$ и $Y(a_i)$ входных и выходных переменных, записанных в табл. 6.3 на всех

Таблица 7.5

a_i	$X(a_i)$	$Y(a_i)$	$G(a_i)$	$T(a_i)$	$B(a_i)$
a_1	$x_1 x_2 x_3$	y_1, y_2, y_{12}	a_8	a_3, a_5, a_6	4
a_2	x_8	y_3, y_5	a_7, a_9	a_3	2
a_3	x_4, x_{10}	y_3, y_5, y_8, y_{10}	a_1, a_2, a_9	a_8, a_9	3
a_4	x_7, x_9	y_{13}, y_{14}, y_{18}	a_7, a_8	a_6, a_8	3
a_5	x_1	y_7, y_{16}	a_1	a_6	2
a_6	x_5	y_1, y_2, y_{16}	a_1, a_4, a_5	a_1	2
a_7	x_8	y_{13}, y_{14}, y_{18}	a_8	a_2, a_4	2
a_8	x_9	y_4, y_6	a_3, a_4	a_4, a_7	2
a_9	x_6, x_{11}	y_3, y_5, y_8, y_{10}	a_3	a_2, a_3	3

переходах из a_i ; множество $G(a_i)$ состояний, из которых есть переходы в a_i [$a_i \notin G(a_i)$]; множество $T(a_i)$ состояний, в которые есть переходы из a_i , а также число $B(a_i)$ строк в переходах из состояния a_i . Как и выше, в табл. 7.5 опущены микрооперации $y_{11}, y_9, y_{15}, y_{17}$, встречающиеся всегда вместе в табл. 6.3 с микрооперациями y_8, y_7, y_{10} и y_{16} соответственно.

Из табл. 7.5 сразу видно, что не любые сочетания состояний могут войти в один блок разбиения π , так как соответствующий компонентный автомат не удастся реализовать на одной ПЛМ (10, 10, 20, 5). Например, если $A^m = \{a_1, a_2, a_3, a_4\}$, то одновременно нарушаются ограничения (7.7) и (7.8), так как в автомате S^m : $X(A^m) = \{x_1, x_2, x_3, x_4, x_6, x_7, x_9, x_{10}\}$; $Z_x^m = \{z_1, z_2, z_3, z_4\}$ — в каждое из состояний a_1, a_2, a_3, a_4 есть переходы из состояний, не вошедших в A^m [см. строки 5—9 в столбце $T(a_i)$ табл. 7.5]; $X^m = X(A^m) \cup Z_x^m$, $|X^m| = 12$; $Y(A^m) = \{y_1, y_2, y_3, y_5, y_8, y_{10}, y_{12}, y_{13}, y_{14}, y_{18}\}$; $Z_y^m = \{z_5, z_6, z_8, z_9\}$, так как $a_5, a_6, a_8, a_9 \in T(A^m)$ [см. строки 1—4 в столбце $T(a_i)$ табл. 7.5]; $Y^m = Y(A^m) \cup Z_y^m$, $|Y^m| = 14$.

Как и в § 6.3, в последовательном алгоритме будем использовать три основные оценки:

$c(a_i, a_j)$ — близость двух состояний a_i и a_j ;

$c(a_i, A^m)$ — близость состояния a_i к блоку A^m ;

$W(a_i, A^m)$ — весовая функция состояния a_i по отношению к блоку $A^m \subset A$. Здесь a_i — состояние, не вошедшее в блок $A^m (a_i \in A \setminus A^m)$, а $W(a_i, A^m)$ является критерием выбора очередного состояния для включения в блок A^m разбиения π . Близость двух состояний будем определять выражением

$$\begin{aligned} c(a_i, a_j) = & k_1 |X(a_i) \cap X(a_j)| + k_2 |Y(a_i) \cap Y(a_j)| + \\ & + k_3 [|G(a_i) \cap G(a_j)| + |T(a_i) \cap T(a_j)| + \\ & + |\{a_i\} \cap T(a_j)| + |\{a_j\} \cap T(a_i)|]. \end{aligned} \quad (7.11)$$

Член с коэффициентом k_3 в (7.11) учитывает суммарное число состояний, из которых имеются переходы в a_i и a_j [$G(a_i) \cap G(a_j)$], в которые есть переходы из a_i и a_j [$T(a_i) \cap T(a_j)$], а также входят ли сами a_i и a_j в множества состояний $T(a_j)$ и $T(a_i)$ соответственно. В примере положим $k_1 = k_2 = k_3 = 1$. Так,

$$\begin{aligned} c(a_2, a_3) = & |\{x_8\} \cap \{x_4, x_{10}\}| + |\{y_3, y_5\} \cap \{y_5, y_8, y_{10}\}| + \\ & + |\{a_7, a_9\} \cap \{a_1, a_2, a_9\}| + |\{a_3\} \cap \{a_8, a_9\}| + \\ & + |\{a_2\} \cap \{a_8, a_9\}| + |\{a_3\} \cap \{a_3\}| = 4. \end{aligned}$$

Близость состояния a_i к блоку A^m будем определять по формуле

$$c(a_i, A^m) = \sum_{a_j \in A^m} c(a_i, a_j). \quad (7.12)$$

Для весовой функции $W(a_i, A^m)$ может быть использовано соотношение

$$W(a_i, A^m) = c(a_i, A^m) - c(a_i, A \setminus A^m), \quad (7.13)$$

определяемое как разность между близостями состояния a_i к блоку A^m и к множеству $A \setminus A^m$, не вошедших в A^m состояний.

Алгоритм поиска разбиения π почти полностью совпадает с рассмотренным выше поиском разбиения π_E при синтезе МПА на ПЛМ комбинационного типа. Формирование блока A^m ($m = \overline{1, n}$) осуществляется в два этапа. На первом из них в блок A^m включается опорный элемент — состояние a_i , для которого

$$c(a_i, A' \setminus \{a_{ij}\}) = \max_j c(a_j, A' \setminus \{a_j\}),$$

где $A' = A \setminus \left(\bigcup_{k=1}^{m-1} A^k \right)$ — множество еще не распределенных

состояний; $a_i, a_j \in A'$, а $c(a_j, A' \setminus \{a_j\})$ определяется по формуле (7.12). Таким образом, в качестве опорного элемента блока A^m выбирается состояние, имеющее наибольшую близость ко всем еще не распределенным состояниям, не вошедшим в блоки A^1, \dots, A^{m-1} .

На каждом шаге второго многошагового этапа по формуле (7.13) определяется весовая функция $W(a_i, A^m)$ для каждого из еще не распределенных состояний, у которых $c(a_i, A^m) \neq 0$. Из всех таких состояний выбирается состояние a_j , для которого $W(a_j, A^m) = \max_i W(a_i, A^m)$, если его

включение в блок A^m не нарушает какого-либо ограничения из (7.7) — (7.10). Если таких состояний несколько, из них выбирается состояние с $\max c(a_i, A^m)$; если и эти оценки одинаковы для нескольких состояний, в блок A^m включается любое из них.

Если состояние a_j с наибольшей весовой функцией $W(a_j, A^m)$ нельзя включить в блок из-за нарушения ограничений (7.7) — (7.10), то в порядке убывания весовой функции просматриваются еще не включенные в блоки состояния с ненулевой близостью к блоку A^m . Среди них выбирается первое, не нарушающее ограничений (7.7) — (7.10). В последнюю очередь в том же порядке просматриваются состояния с нулевой близостью к блоку A^m , т. е. $c(a_i, A^m) = 0$. Процесс формирования блока A^m заканчивается, если все состояния уже распределены или если в блок A^m нельзя добавить ни одного состояния без нарушения какого-либо из ограничений (7.7) — (7.10).

Продолжим рассмотрение примера синтеза МПА из табл. 6.3 на ПЛМ (10, 10, 20, 5). Исходные данные для вычисления оценок $c(a_i, a_j)$, $c(a_i, A^m)$ и $W(a_i, A^m)$ приведены в табл. 7.5. Матрица близости состояний с учетом $k_1=k_2=k_3=1$ представлена в табл. 7.6, а последовательность формирования блоков разбиения π — в табл. 7.7, структура которой аналогична структуре табл. 6.4. В примере разбиение π содержит два блока:

$$\pi = \{A^1, A^2\};$$

$$A^1 = \{a_2, a_3, a_4, a_7, a_8, a_9\}; \quad A^2 = \{a_1, a_5, a_6\}.$$

Таблица 7.6

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9
a_1	—	1	1	1	3	4	0	0	1
a_2	1	—	4	1	0	0	1	0	5
a_3	1	4	—	1	1	1	0	1	6
a_4	1	1	1	—	1	1	5	3	0
a_5	3	0	1	1	—	3	0	0	0
a_6	4	0	1	1	3	—	0	1	0
a_7	0	1	0	5	0	0	—	2	1
a_8	0	0	1	3	0	1	2	—	1
a_9	1	5	6	0	0	0	1	1	—

Таблица 7.7

A	$c(a_i, A^l)$	$W(a_i, A^1)$							$c(a_i, A^l)$	$W(a_i, A^2)$		
		1	2	3	4	5	6	7		8	9	10
a_1	11	-9	-7	-5	-5	-3	-3	—	7	⊕	+	+
a_2	12	-4	6	⊕	+	+	+	+				
a_3	15	⊕	+	+	+	+	+	+				
a_4	13	-11	-13	-11	-1	⊕	+	+				
a_5	8	-6	-6	-6	-6	-4	-4	—	6	0	6	⊕
a_6	10	-8	-8	-8	-6	-2	-2	—	7	1	⊕	+
a_7	9	-9	-7	-5	-1	11	⊕	+				
a_8	8	-6	-4	-4	⊕	+	+	+				
a_9	14	-2	⊕	+	+	+	+	+				
X		$x_4, x_{10}, x_6, x_{11}, x_9, x_7, x_8$								x_1, x_2, x_3, x_5		
Y		$y_3, y_5, y_8, y_{10}, y_4, y_6, y_{13}, y_{14}, y_{18}$								$y_1, y_2, y_{12}, y_{16}, y_7$		

Таблица 7.8

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$	h
a_2	000	a_2	000	$\overline{x_6}$	—	—	1
		a_3	001	$\overline{x_6}$	$y_3 y_5$	D_3	2
a_3	001	a_3	001	$\overline{x_{10}}$	$y_3 y_5$	D_3	3
		a_8	100	$\overline{x_4 x_{10}}$	$y_5 y_8$	D_1	4
		a_9	101	$\overline{x_4 x_{10}}$	y_{10}	$D_1 D_3$	5
a_4	010	a_4	010	$\overline{x_7 x_9}$	$y_{13} y_{18}$	D_2	6
		b_1	110	$\overline{x_7}$	$y_{13} z_6$	$D_1 D_2$	7
		a_8	100	$\overline{x_7 x_9}$	$y_{13} y_{14}$	D_1	8
a_7	011	a_2	000	$\overline{x_8}$	$y_{14} y_{18}$	—	9
		a_4	010	$\overline{x_8}$	$y_{13} y_{18}$	D_2	10
a_8	100	a_4	010	$\overline{x_9}$	y_6	D_2	11
		a_7	011	x_9	$y_4 y_6$	$D_2 D_3$	12
a_9	101	a_2	000	$\overline{x_8 x_{11}}$	$y_5 y_8$	—	13
		a_3	001	$\overline{x_{11}}$	$y_3 y_8$	D_3	14
		a_9	101	$x_8 x_{11}$	y_{10}	$D_1 D_3$	15
b_1	110	a_3	001	z_3	—	D_3	16
		b_1	110	$\overline{z_3}$	—	$D_1 D_2$	17

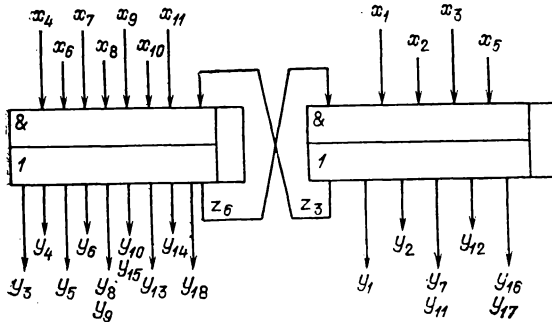


Рис. 7.4

Таблица 7.9

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$	h
a_1	00	a_1	00	$\overline{x_1 x_2}$	$y_1 y_2$	—	1
		a_5	01	$\overline{x_1}$	$y_1 y_2 y_{12}$	D_2	2
		a_6	10	$x_1 x_2 \overline{x_3}$	$y_2 y_{12}$	D_1	3
		b_2	11	$x_1 x_2 x_3$	$y_1 y_2 z_3$	$D_1 D_2$	4
a_5	01	a_5	01	$\overline{x_1}$	y_7	D_2	5
		a_6	10	x_1	y_{16}	D_1	6
a_6	10	a_1	00	x_5	$y_1 y_2$	—	7
		a_1	00	$\overline{x_5}$	y_{16}	—	8
b_2	11	a_6	10	z_6	—	D_1	9
		b_2	11	$\overline{z_6}$	—	$D_1 D_2$	10

Таблица 7.10

x_4	x_6	x_7	x_8	x_9	x_{10}	x_{11}	z_3	T_1	T_2	T_3	e_h	D_1	D_2	D_3	y_2	y_4	y_5	y_6	y_8	y_{10}	y_{12}	y_{14}	y_{15}	z_6
—	0	—	—	—	—	—	—	0	0	0	e_2	.	.	1	1	.	1
—	—	—	—	—	1	—	—	0	0	1	e_3	:	.	1	1	.	1
0	—	—	—	—	0	—	—	0	0	1	e_4	1	1	.	1	:
1	—	—	—	—	0	—	—	0	0	1	e_5	1	.	1	1
—	—	0	—	1	—	—	—	0	1	0	e_6	.	1	1	.	.	.
—	—	1	—	—	—	—	—	0	1	0	e_7	1	1	1	.	.	1
—	—	0	—	0	—	—	—	0	1	0	e_8	1	1	1	.	.
—	—	—	1	—	—	—	—	0	1	1	e_9	1	1	.
—	—	—	0	—	—	—	—	0	1	1	e_{10}	.	1	1	.	1	.
—	—	—	—	0	—	—	—	1	0	0	e_{11}	.	1	1
—	—	—	—	1	—	—	—	1	0	0	e_{12}	.	1	1	.	1	.	1
—	0	—	—	—	—	—	1	—	1	0	e_{13}	1	.	1
—	—	—	—	—	—	—	0	—	1	0	e_{14}	.	.	1	1	.	.	.	1
—	1	—	—	—	—	—	1	—	1	0	e_{15}	1	.	1	1
—	—	—	—	—	—	—	—	1	1	0	e_{16}	.	.	1
—	—	—	—	—	—	—	—	0	1	0	e_{17}	1	1

Структурные таблицы компонентных автоматов, построенных по этому разбиению, представлены в табл. 7.8 и 7.9, а соответствующая логическая схема — на рис. 7.4. В табл. 7.10 и 7.11 осуществлено программирование ПЛМ в соответствии с табл. 7.8 и 7.9. В табл. 7.10 терм e_1 , соответствующий первой строке табл. 7.8, отсутствует, так как он не входит ни в одну функцию возбуждения памяти или функцию выходов. Сравнение рис. 7.4 и 6.3 показывает, что ис-

Таблица 7.11

x_1 x_2 x_3 x_5 x_6 T_1 T_2	e_h	D_1 D_2 y_1 y_2 y_7 y_{12} y_{16} z_3
1 0 — — — 0 0	e_1	. . 1 1 . . .
0 — — — — 0 0	e_2	. 1 1 1 . 1 . .
1 1 0 — — 0 0	e_3	1 . . 1 . 1 . .
1 1 1 — — 0 0	e_4	1 1 1 1 . . . 1
0 — — — — 0 1	e_5	. 1 . . 1 . . .
1 — — — — 0 1	e_6	1 1 .
— — — 1 — 1 0	e_7	. . 1 1 . . .
— — — 0 — 1 0	e_8 1 .
— — — — 1 1 1	e_9	1
— — — — 0 1 1	e_{10}	1 1

пользование ПЛМ с памятью позволило уменьшить число ПЛМ при одинаковых параметрах s , t и q в ПЛМ с памятью и в ПЛМ комбинационного типа.

При нахождении разбиения π на множестве A состояний МПА S в процессе его реализации на ПЛМ (s , t , q , r) наиболее серьезными являются ограничения (7.7) и (7.8), поскольку число L входных и N выходных переменных даже в автоматах средней сложности превосходит число входов s и выходов t в ПЛМ (s , t , q , r). В связи с этим оказывается желательным, чтобы при декомпозиции у любых двух компонентных автоматов S^m и S^p множества входных $X(A^m)$ и $X(A^p)$ или выходных $Y(A^m)$ и $Y(A^p)$ переменных, а может быть и тех и других, не пересекались либо это пересечение было минимальным. Добиться разбиения этих переменных, если это возможно, удастся введением ряда отношений на множестве A состояний МПА, используя свойства его таблицы переходов.

Введем на множестве A внутренних состояний МПА отношение ω . Два состояния a_i и a_j ($i \neq j$) автомата S находятся в отношении ω ($a_i \omega a_j$), если и только если в множе-

ствах $X(a_i)$ и $X(a_j)$ есть общие входные переменные: $X(a_i) \cap X(a_j) \neq \emptyset$. Построим граф Γ_ω этого отношения (рис. 7.5,а) для автомата S из табл. 6.3. Графу Γ_ω можно поставить в соответствие некоторое разбиение π_ω на множестве A такое, что в один блок π_ω попадут все состояния из одного компонента связности Γ_ω . Число блоков π_ω , таким образом, равно числу компонентов связности Γ_ω . В примере (рис. 7.5,а)

$$\pi_\omega = \overline{a_1, a_5}; \overline{a_2, a_9}; \overline{a_3}; \overline{a_4, a_8}; \overline{a_6}; \overline{a_7}.$$

Пусть A^m и A^p — множества вершин в m -м и p -м компонентах связности Γ_ω . Если $X(A^m) = \bigcup_{a_i \in A^m} X(a_i)$ и $X(A^p) = \bigcup_{a_j \in A^p} X(a_j)$, то

$$X(A^m) \cap X(A^p) = \emptyset \quad (7.14)$$

при $m \neq p$ для любых $m, p = \overline{1, n}$, где n — число компонентов связности графа Γ_ω . Тогда если в качестве исходного разбиения выбрать π_ω , то пересечение множеств входных переменных для любой пары компонентных автоматов будет пусто.

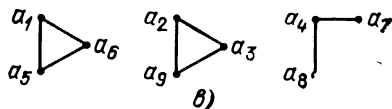
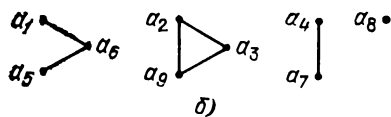
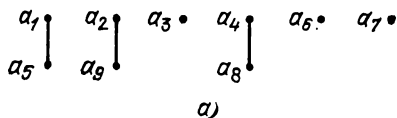


Рис. 7.5

Часто оказывается необходимым, чтобы сеть удовлетворяла еще каким-либо свойствам, кроме определенных выражением (7.14). Например, может быть задано число компонентных автоматов, меньшее числа блоков в разбиении π_ω , или число состояний в каждом компонентном автомате, большее числа элементов в каждом блоке π_ω . Тогда по π_ω можно построить некоторое разбиение $\pi'_\omega > \pi_\omega$, объединяя несколько блоков π_ω в один. Если π'_ω используется в качестве исходного разбиения при декомпозиции, то и для π'_ω соотношения вида (7.14) тоже будут справедливы.

Точно так же на множестве A состояний автомата S можно задать отношение φ такое, что $a_i \varphi a_j$ ($i \neq j$), если и только если пересечение множеств микроопераций на всех

переходах из состояний a_i и a_j не пусто: $Y(a_i) \cap Y(a_j) \neq \emptyset$. Тогда если в качестве исходного разбиения использовать π_Φ или $\pi'_\Phi > \pi_\Phi$, то для любой пары компонентных автоматов S^m и S^p множества выходных переменных также разделяются: $Y(A^m) \cap Y(A^p) = \emptyset$. Из графа Γ_Φ (рис. 7.5,б) для автомата S получаем

$$\pi_\Phi = \overline{a_1, a_5, a_6}; \overline{a_2, a_3, a_9}; \overline{a_4, a_7}; \overline{a_8}.$$

Если на множестве A определить отношение α , в котором будут находиться два состояния a_i и a_j ($i \neq j$), если и только если пересечение множеств входных переменных или микроопераций на всех переходах из состояний a_i и a_j не пусто:

$$[(X(a_i) \cap X(a_j) \neq \emptyset) \vee (Y(a_i) \cap Y(a_j) \neq \emptyset)] = 1,$$

то использование разбиения π_α или $\pi'_\alpha > \pi_\alpha$ при декомпозиции позволит разделить входные и выходные переменные у компонентных автоматов. Непосредственно из рис. 7.5,б для автомата S получаем разбиение

$$\pi_\alpha = \overline{a_1, a_5, a_6}; \overline{a_2, a_3, a_9}; \overline{a_4, a_7, a_8},$$

которое и было использовано в § 7.2 при реализации МПА S сетью из трех компонентных автоматов.

На практике наибольший интерес представляет такое решение задачи декомпозиции, при котором разделяются входные или выходные переменные (или и те, и другие) в компонентных автоматах и минимизируется число связей между ними. Эта задача сводится к задаче нахождения разбиения $\pi'_\omega \geq \pi_\omega$ ($\pi'_\Phi \geq \pi_\Phi$ или $\pi'_\alpha \geq \pi_\alpha$), в которой ищется объединение блоков π_ω в блоки π'_ω (блоков π_Φ в блоки π'_Φ или блоков π_α в блоки π'_α) с минимизацией числа связей между состояниями, попавшими в различные блоки. Так, использованное при синтезе МПА S на ПЛМ (10, 10, 20, 5) разбиение $\pi = \overline{a_1, a_5, a_6}; \overline{a_2, a_3, a_4, a_7, a_8, a_9}$ получается объединением второго и третьего блоков разбиения π_α ($\pi > \pi_\alpha$). Необходимо, однако, еще раз подчеркнуть, что разбиение входных и выходных переменных возможно не в любом автомате, тогда как алгоритмы, аналогичные приведенному выше последовательному алгоритму, как правило, позволяют минимизировать число элементов в пересечении множеств входных и выходных переменных.

Рассмотренный метод синтеза одноуровневой логической схемы МПА на ПЛМ с памятью подобен описанному

в § 6.3. Если его использование не позволяет построить разбиение множества A с желаемыми свойствами, то может быть использован метод, изложенный в § 6.4. Например, если МПА S синтезируется по ГСА Γ , последняя отмечена по алгоритму Φ_1 или Φ_2 и выполняются условия $L > s$, $N \leq t$, $B \leq q$, $R \leq r$, то применение метода, описанного в § 6.4, позволяет отметить ГСА Γ таким образом, что требуемое разбиение π_ω будет формироваться тривиально. Заметим, что если выполняется условие $L^{\max} > s$ [L^{\max} — наибольшее число букв в одной конъюнкции X_h , записанной в столбце $X(a_m, a_s)$ таблицы переходов МПА], то синтезировать логическую схему МПА на ПЛМ (s, t, q, r) можно только при использовании метода, рассмотренного в § 6.4. Если в этом случае такой метод по каким-либо причинам использовать нежелательно, то по ГСА Γ может быть построена многоуровневая схема МПА на ПЛМ (s, t, q, r), порядок синтеза которой описан в [31, 38].

ГЛАВА ВОСЬМАЯ.

СИНТЕЗ АВТОМАТОВ НА ЭЛЕМЕНТАХ КОМБИНИРОВАННОГО БАЗИСА

8.1. Принципы использования комбинированного базиса

Вернемся к рассмотрению многоуровневых схем автоматов. В общем случае такие схемы могут быть представлены в виде структуры, показанной на рис. 8.1. Подсхема A имеет большое число входов и существенно меньшее число выходов и позволяет преобразовать множество $A_{\text{вх}} = \{x_1, \dots, x_L, T_1, \dots, T_R\}$ переменных большой мощности в множество $A_{\text{вых}}$ переменных ограниченной мощности ($|A_{\text{вх}}| \gg |A_{\text{вых}}|$). Число элементов множества $A_{\text{вых}}$ достаточно для представления всех входных наборов значений переменных из множества $A_{\text{вх}}$, существенно влияющих на переходы в МПА. Подсхема C имеет малое число входов и большое число выходов и позволяет преобразовать множество $C_{\text{вх}}$ входных переменных ограниченной мощности в множество $C_{\text{вых}} = \{y_1, \dots, y_N, D_1, \dots, D_R\}$ выходных переменных существенно большей мощности ($|C_{\text{вых}}| \gg |C_{\text{вх}}|$). Число элементов множества $C_{\text{вх}}$ достаточно для представления в закодированном виде всех наборов значений пере-

менных из множества $C_{\text{вых}}$, которые формируются на любом переходе в МПА. Множества входных $V_{\text{вх}}$ ($V_{\text{вх}}=A_{\text{вых}}$) и выходных $V_{\text{вых}}$ ($V_{\text{вых}}=C_{\text{вх}}$) переменных подсхемы В имеют ограниченную мощность. Эта подсхема имеет достаточные внутренние ресурсы для заданного законом функционирования МПА преобразования наборов значений переменных из множества $V_{\text{вх}}$ в наборы значений переменных из множества $V_{\text{вых}}$. Многоуровневая структура автомата, показанная на рис. 8.1, наиболее эффективна при использовании в схеме МПА разнотипных элементов по реализуемым функциям и числу входов и выходов.

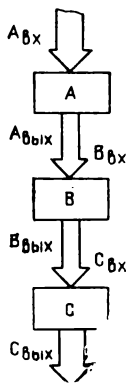


Рис. 8.1

Заметим, что подсхемы А и С в схеме на рис. 8.1 фактически выполняют функции мультиплексирования и демультимплексирования, подсхема В — функции преобразования информации, определяемые законом функционирования МПА. Будем рассматривать подсхему В как ядро схемы автомата такое, что настройка последнего на заданный алгоритм функционирования осуществляется путем перестройки только ядра, в то время как подсхемы А и С выполняют одни и те же функции для автоматов, настроенных по-разному. Использование ядра как основного настраиваемого блока схемы позволяет изменять программируемые матричные БИС лишь для построения подсхемы В. Подсхемы же А и С могут состоять из стандартных логических элементов, не допускающих возможности настройки или перестройки. К числу последних относятся, например, мультиплексоры (для построения подсхемы А) и дешифраторы (для построения подсхемы С).

Использование дешифраторов и мультиплексоров в схеме автомата оправдывается экономическими соображениями, поскольку суммарное число элементов в этом случае не увеличивается, а стоимость стандартных логических элементов обычно ниже, чем настраиваемых матричных схем. Заметим, что при программировании последних часто появляются бракованные элементы, т. е. число реально использованных матричных БИС для построения логической схемы МПА часто превышает число элементов такой схемы. В силу этого довольно часто схему МПА целесообразно строить не в однородном, а в комбинированном базисе, который объединяет различные матричные схемы и стандарт-

ные логические элементы типа дешифраторов, мультиплекторов и т. д. Каждый элемент комбинированного базиса применяется лишь тогда, когда наиболее полно используются его функциональные возможности; соответственно для построения каждой подсхемы МПА выбирается такой элемент, использование которого наиболее эффективно.

8.2. Построение автоматов на ПЛМ и ПЗУ

Постоянные запоминающие устройства целесообразно применять для построения таких подсхем автомата, которые реализуют БФ, представленные в совершенной ДНФ либо в близкой к ней форме. Рекомендации по использованию ПЛМ или ПЗУ для построения одних и тех же схем даны в [39]. Следуя этим рекомендациям и подводя итог рассмотренным в гл. 6 методам синтеза МПА, перечисляем структуры автоматов, в составе которых вместе с ПЛМ целесообразно использовать ПЗУ.

1. В автоматах с одноуровневой структурой ПЗУ рационально использовать в качестве элементов схемы МПА Мура, реализующих только функции выходов, тогда как ПЛМ в этом случае можно использовать для реализации как функций выходов, так и функций возбуждения памяти автомата. Значения функций возбуждения, не зависящие от значений входных переменных, также могут быть реализованы ПЗУ; это будет показано в гл. 9. Аналогично если в одноуровневой схеме МПА Мили некоторые значения выходных переменных или функций возбуждения не зависят от значений входных переменных, то часто их также выгодно реализовать на ПЗУ.

2. В автоматах с РУ-структурой ПЗУ целесообразно применять для построения Y-подсхемы; Р-подсхема, как и ранее (см. § 6.5), строится на ПЛМ.

3. В автоматах с РФ-структурой ПЗУ рационально использовать для построения F-подсхемы; Р-подсхема, как и ранее (см. § 6.5), строится на ПЛМ.

Заметим, что если для построения схемы МПА могут быть использованы лишь ПЗУ (s, t), то принципы тривиальной и нетривиальной реализации автоматов на их основе будут совпадать с аналогичными принципами тривиальной реализации автоматов на ПЛМ (s, t, q) с тем отличием, что отсутствует необходимость учета ограничения типа $B \leq q$. Например, при формировании в гл. 6 структурных подтаблиц для каждой подтаблицы W_u должны быть вы-

полнены лишь два условия: $(L_u + R) \leq s$, $(N_u + R_u) \leq t$. В этом смысле задача синтеза МПА на ПЗУ (s, t) не сложнее, чем задача синтеза автоматов на ПЛМ (s, t, q) .

8.3. Синтез автоматов на ПЛМ и дешифраторах

Структура логической схемы автомата при реализации его на ПЛМ и дешифраторах показана на рис. 8.2 (назовем ее PD-структурой). Комбинационная часть этой схемы состоит из двух подсхем — Р и D. Р-подсхема строится так же, как и ранее (см. гл. 6), и позволяет сформировать значения функций возбуждения элементов памяти D_1, \dots, D_R , значения выходных переменных из множества $Y^1 \cup \dots \cup Y^U$ и двоичные коды элементов множества $Y_D = Y \setminus (Y^1 \cup \dots \cup Y^U)$, представленные наборами значений переменных из множества $Z = \{z_1^1, \dots, z_{V_1}^1, \dots, z_1^j, \dots, z_{V_j}^j\}$

(смысл параметра $V_i, i = \overline{1, j}$, будет пояснен ниже). D-подсхема построена из j стандартных дешифраторов; она позволяет формировать значения выходных переменных из множества Y_D на основе их кодов. Каждый дешифратор

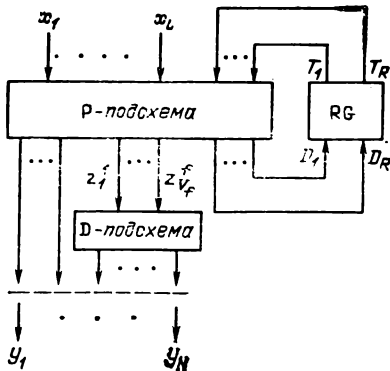


Рис. 8.2

D-подсхемы будем характеризовать числами входов r и выходов m (для полного дешифратора $m = 2^r$); такие дешифраторы будем называть DC (r, m) . Напомним, что при подаче на входы дешифратора двоичного кода с десятичным эквивалентом j единичный сигнал появится только на его j -м выходе. В [40] показано, что использование дешифраторов в схеме МПА часто позволяет сократить число элементов схемы. Элементы схемы здесь — ПЛМ или дешиф-

ратор; предполагается, что их сложность (цена) одинакова. Если цены дешифратора и ПЛМ различны, можно ввести соответствующие масштабные коэффициенты.

Пусть задана ГСА Γ и необходимо синтезировать схему МПА Мура с PD-структурой на ПЛМ (s, t, q) и DC (r, m) . Для этого выполняются следующие процедуры:

1) с помощью метода из § 6.4 по ГСА Γ строятся подтаблицы переходов W_0, W_1, \dots, W_U , для каждой из кото-

рых выполняются условия: $(L_u + R) \leq s$, $B_u \leq q$, $u = 1, U$;

2) с помощью метода из § 6.4 выполняется кодирование внутренних состояний МПА R -разрядными двоичными кодами, в результате определяется число R_u функций возбуждения, значения которых формируются на выходах u -й ПЛМ P -подсхемы ($u = \overline{1, U}$);

3) выполняется синтез D -подсхемы, в результате чего формируются множество Y_D и двоичные коды переменных из множества Y_D ;

4) с помощью метода из § 6.4 переменные из множества $Z \cup (Y \setminus Y_D)$ распределяются на выходы P -подсхемы.

Методы, описанные в § 6.4, используются для решения задач в пп. 1, 2 и 4 практически без всяких изменений.

Для синтеза D -подсхемы построим граф Γ_y отношения совместимости выходных переменных МПА из множества $Y = \{y_1, \dots, y_N\}$. Вершинам графа Γ_y поставим в соответствие выходные переменные (микрооперации) y_1, \dots, y_N , две вершины y_i и y_j соединим ребром, если и только если выходные переменные y_i и y_j совместимы между собой, т. е. встречаются вместе хотя бы в одной строке таблицы переходов (структурной таблицы) МПА. Выделим из графа Γ_y максимальный подграф Γ_{y1} , ни одна пара вершин которого не соединена ребром. Эта задача может быть решена раскраской вершин графа Γ_y , при которой любые две смежные вершины окрашиваются в разные цвета и число ψ_1 вершин, окрашенных в первый цвет, максимально возможно. Из найденных ψ_1 вершин формируется подграф Γ_{y1} . Выделение подграфа Γ_{y1} может выполняться с помощью метода нахождения максимальных совместимых подмножеств [1]. Поскольку любые две выходные переменные y_i и y_j , соответствующие вершинам подграфа Γ_{y1} , никогда одновременно не равны единице, значения каждой из них могут быть сформированы на выходах одного дешифратора.

Оценим целесообразность введения в D -подсхему одного, если $\psi_1 \leq (m-1)$, или нескольких, если $\psi_1 > (m-1)$, дешифраторов; правила построения схемы дешифратора из нескольких модулей $DC(r, m)$ при $\psi_1 > (m-1)$ очевидны. Предположим, что все выходные переменные, соответствующие вершинам подграфа Γ_{y1} , закодированы V_1 -разрядными двоичными кодами ($V_1 = \text{int} \log_2[\psi_1 + 1]$) и включены в множество Y_D . Введение дешифраторов приводит к сокращению числа элементов в схеме МПА, если $(U_1 + K_1) < U$,

где U — число ПЛМ в одноуровневой схеме МПА при его реализации только на ПЛМ; K_1 — число введенных дешифраторов на первом этапе, $K_1 = \lceil (\psi_1 + 1)/m \rceil$; U_1 — число ПЛМ в Р-подсхеме после введения на первом этапе K_1 дешифраторов. Число выходных переменных в Р-подсхеме после введения K_1 дешифраторов сокращается на $\psi_1 - V_1$. Если $K_1 > 1$ и в последнем из K_1 дешифраторов используется менее чем m выходов, то необходимо проверить, не меньше ли будет элементов, если ввести $(K_1 - 1)$ дешифратор, в каждом из которых используются все выходы. Если общее число элементов в схеме МПА при введении K_1 и $(K_1 - 1)$ дешифраторов не меняется, выбирается тот вариант, в котором используется меньше выходов во всех элементах схемы. Если оказалось, что целесообразно ввести $(K_1 - 1)$ дешифратор, то $(\psi_1 + 1) - (K_1 - 1)m$ вершин подграфа Γ_{y^1} возвращаются из подграфа Γ_{y^1} в остаточный подграф $\Gamma_{y^1} = \Gamma_y \setminus \Gamma_{y^1}$ графа Γ_y , причем для упрощения последующих этапов в первую очередь возвращаются вершины, которые максимизируют суммарное число изолированных вершин в графе Γ_{y^1} . После выполнения указанных действий под ψ_1 будем понимать число оставшихся вершин в графе Γ_{y^1} .

После введения первой группы дешифраторов для остаточного подграфа Γ_{y^1} аналогичным образом (путем подсчета числа элементов в схеме МПА до и после введения дешифраторов на втором этапе) определяется целесообразность введения второй группы дешифраторов. Если на некотором $(f+1)$ -м этапе окажется, что дешифраторы вводить нецелесообразно (так как число элементов в схеме МПА увеличивается по сравнению с числом элементов, полученным на предыдущем этапе) и $Y \setminus Y_D \neq \emptyset$, то выходные переменные МПА из множества $Y \setminus Y_D$ распределяются между выходами ПЛМ Р-подсхемы (см. § 6.4). Здесь $Y \setminus Y_D$ — множество переменных, соответствующих вершинам подграфа Γ_{y^f} .

Обозначим множество выходных переменных МПА, которым соответствуют вершины подграфа Γ_{y_i} , через Y (Γ_{y_i}). Переменные из каждого множества $Y(\Gamma_{y_i})$, $i = I, f$, кодируются двоичными V_i -разрядными кодами ($V_i = \text{intlog}_2 [\psi_i + 1]$), причем код $00 \dots 0$ (содержащий все нули) для кодирования не используется и, если МПА должен выдать единичное значение переменной $y_j \in Y(\Gamma_{y_i})$, на выходах $z_1^i, \dots, z_{V_i}^i$ Р-подсхемы должен быть сформирован код $K(y_j)$, которым закодирована переменная y_j . Таким

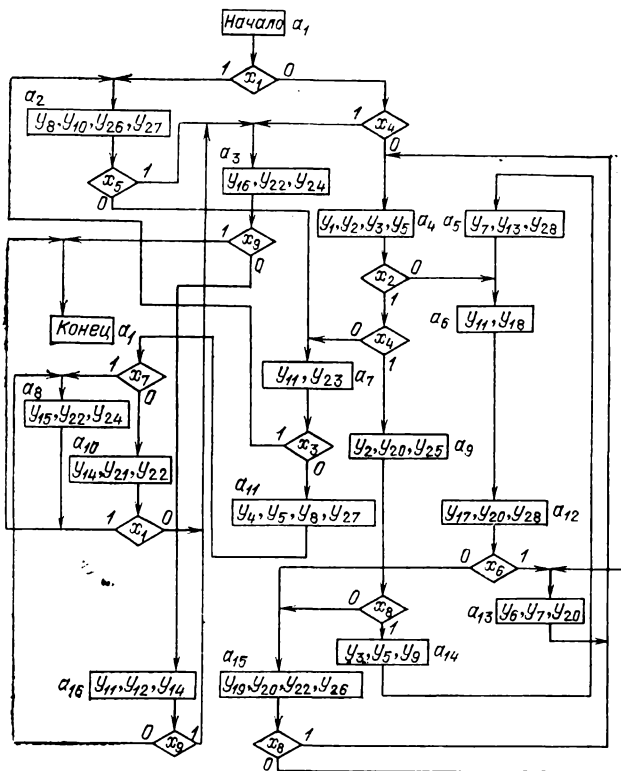


Рис. 8.3

образом, в любом состоянии a_m МПА формируются единичные значения переменных из множества $Z(a_m) \cup Y(a_m)$, где $Y(a_m) = Y(a_m) \setminus Y_D$, $Z(a_m) \subseteq Z$. Переменные из множества $Z(a_m) \cup Y(a_m)$ для каждого состояния $a_m \in A$ МПА договоримся заносить в специально выделенный для этих целей столбец $Z(a_m)$, $Y(a_m)$ таблицы переходов (структурной таблицы) МПА.

Рассмотрим пример синтеза МПА S с PD-структурой на ПЛМ (7, 6, 12) и DC (4, 16) по ГСА Г, изображенной на рис. 8.3. Отметим ГСА Г метками a_1, \dots, a_{16} (в a_{14} значение y_{27} любое) по ал-

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	a_4	a_6	a_5	a_{16}
	01	a_2	a_9	a_{12}	a_{11}
	11	a_4	a_{13}	a_{15}	a_{14}
	10	a_3	a_7	a_{10}	a_8

Рис. 8.4

горитму Φ_2 (автомат Мура) и построим граф Γ_c . Этот граф может быть разбит на три подграфа с числом вершин $s - R = 7 - 4 = 3$ без разрезания ребер. Пусть $X^1 = \{x_1, x_2, x_4\}$, $X^2 = \{x_5, x_7, x_8\}$, $X^3 = \{x_3, x_6, x_9\}$. Построим подтаблицы переходов W_0, W_1, W_2, W_3 МПА и выполним кодирование его внутренних состояний так, как это представлено в карте Карно на рис. 8.4. Коды состояний занесены в столбцы $K(a_m)$ и $K(a_s)$ структурной таблицы (табл. 8.1). Различные подтаблицы табл. 8.1 отделены друг от друга двойными горизонтальными линиями. Матрицы $E^1, F^1, E^2, F^2, E^3, F^3$, сформированные по результатам кодирования, представляются в следующем виде:

$$E^1 = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array} \\ \hline \end{array}; \quad F^1 = \begin{array}{c} \begin{array}{cc} D_1 & D_2 \\ \hline 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{array} \end{array};$$

$$E^2 = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \\ \hline 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{array} \\ \hline \end{array}; \quad F^2 = \begin{array}{c} \begin{array}{cc} D_1 & D_2 \\ \hline 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{array} \end{array};$$

$$E^3 = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \\ \hline 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{array} \\ \hline \end{array}; \quad F^3 = \begin{array}{c} \begin{array}{cc} D_1 & D_2 \\ \hline 1 & 1 \\ 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{array} \end{array}.$$

После минимизации функций матрицы примут вид:

$$E^1 = \begin{array}{c} \begin{array}{cccc} T_1 & T_2 & T_3 & T_4 \\ \hline - & 1 & - & 1 \\ 0 & 1 & - & 0 \\ 0 & 0 & 1 & 0 \\ - & 0 & 0 & 1 \end{array} \\ \hline \end{array}; \quad F^1 = \begin{array}{c} \begin{array}{cc} D_1 & D_2 \\ \hline 1 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{array} \end{array};$$

Таблица 8.1

$a_m, Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$F(a_m, a_s)$	$Z(a_m), \tilde{Y}(a_m)$
a_6, y_{11}, y_{18}	0001	a_{12}	0111	1	$(D_2 D_3 D_4)$	z_4, y_{11}
a_5, y_7, y_{13}, y_{28}	0011	a_6	0001	1	(D_4)	z_3, z_4, y_7, y_{28}
$a_8, y_{16}, y_{22}, y_{24}$	1010	a_1	0000	1	—	z_1, z_3, y_{22}, y_{24}
a_{13}, y_6, y_7, y_{20}	1101	a_4	1100	1	$(D_1 D_2)$	$z_1, z_3, z_4, y_7, y_{20}$
a_{14}, y_3, y_5, y_9	1110	a_5	0011	1	$(D_3 D_4)$	z_1, z_2, z_3, y_3, y_5
a_1 —	0000	a_2 a_3 a_4	0100 1000 1100	$\overline{x_1}$ $\overline{x_1 x_4}$ $\overline{x_1 x_4}$	D_2 D_1 $D_1 D_2$	—
a_4, y_1, y_2, y_3, y_5	1100	a_6 a_7 a_9	0001 1001 0101	$\overline{x_2}$ $x_2 \overline{x_4}$ $x_2 x_4$	(D_4) $D_1 (D_4)$ $D_2 (D_4)$	z_1, z_2, y_2, y_3, y_5
$a_{10}, y_{14}, y_{21}, y_{23}$	1011	a_1 a_2	0000 1000	x_1 $\overline{x_1}$	— D_1	$z_1, z_3, z_4,$ y_{14}, y_{23}
$a_2, y_8, y_{10},$ y_{26}, y_{27}	0100	a_2 a_7	1000 1001	x_5 $\overline{x_5}$	(D_1) $(D_1) D_4$	z_2, y_8, y_{26}, y_{27}
a_9, y_2, y_{20}, y_{25}	0101	a_{14} a_{15}	1110 1111	x_8 $\overline{x_8}$	$(D_1 D_2 D_3)$ $(D_1 D_2 D_3) D_4$	z_2, z_4, y_2, y_{20}
$a_{11}, y_4, y_5,$ y_6, y_{27}	0110	a_8 a_{10}	1010 1011	x_7 $\overline{x_7}$	$(D_1 D_3)$ $(D_1 D_3) D_4$	$z_2, z_3, y_5, y_6, y_{27}$
$a_{15}, y_{19}, y_{20},$ y_{22}, y_{26}	1111	a_4 a_{13}	1100 1101	x_8 $\overline{x_8}$	$(D_1 D_2)$ $(D_1 D_2) D_4$	$z_1, z_2, z_3, z_4,$ y_{20}, y_{22}, y_{26}
$a_3, y_{16}, y_{23}, y_{24}$	1000	a_1 a_{16}	0000 0010	x_9 $\overline{x_9}$	— D_3	z_1, y_{22}, y_{24}
a_7, y_{11}, y_{23}	1001	a_2 a_{11}	0100 0110	x_3 $\overline{x_3}$	(D_2) $(D_2) D_3$	z_1, z_6, y_{11}
$a_{12}, y_{17}, y_{20}, y_{28}$	0111	a_{13} a_{15}	1101 1111	x_6 $\overline{x_6}$	$(D_1 D_2 D_4)$ $(D_1 D_2 D_4) D_3$	$z_2, z_3, z_4,$ y_{20}, y_{28}
$a_{16}, y_{11}, y_{13}, y_{14}$	0010	a_3 a_8	1000 1010	x_9 $\overline{x_9}$	(D_1) $(D_1) D_3$	z_3, y_{11}, y_{14}

$$E^2 = \begin{vmatrix} T_1 & T_2 & T_3 & T_4 \\ 0 & 0 & - & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & - & 0 \end{vmatrix}; \quad F^2 = \begin{vmatrix} D_4 \\ 1 \\ 1 \\ 1 \end{vmatrix};$$

$$E^3 = \begin{vmatrix} T_1 & T_2 & T_3 & T_4 \\ 0 & - & 0 & 1 \\ - & 1 & 1 & 0 \end{vmatrix}; \quad F^3 = \begin{vmatrix} D_3 \\ 1 \\ 1 \end{vmatrix}.$$

Граф Γ_y , построенный по табл. 8.1, показан на рис. 8.5, а. Окрашенные в первый цвет вершины графа Γ_y зачернены на рис. 8.5, а; из этих вершин фомируется первый подграф Γ_{y1} . В результате $\psi_1=15$, $V_1=4$, $K_1=1$. Подсчитаем число элементов в схеме МПА до и после введения дешифратора. Поскольку $N=28$, $R=4$, то до введения дешифратора в схеме МПА будет не менее чем $\lfloor (N+R)/t \rfloor = \lfloor (28+4)/6 \rfloor = 6$ ПЛМ (7, 6, 12). После введения дешифратора $U_1 = \lfloor (N+R+V_1-\psi_1)/t \rfloor = \lfloor (28+4+4-$

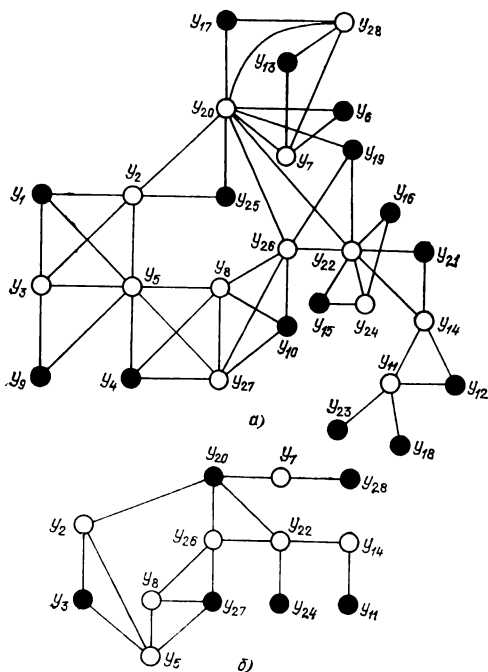


Рис. 8.5

—15)/6[=4, $K_1=1$, в силу чего данный вариант схемы более предпочтителен (в ней содержится $U_1+K_1=5$ элементов). Остаточный подграф Γ_{y^1} графа Γ_y показан на рис. 8.5, б. Окрашенные в первый цвет вершины графа Γ_{y^1} зачернены на рис. 8.5, б; из этих вершин формируется подграф Γ_{y2} . В результате $\psi_2=6$, $V_2=3$, $K_2=1$. Число элементов в схеме МПА до введения дешифратора на втором этапе равно 5 (так как $U_1=4$, $K_1=1$), а после введения также 5 ($U_2=$] $(N+R+V_1+V_2-\psi_1-\psi_2)/t[=]$ $(28+4+4+3-15-6)/6[=3$, $K_1=K_2=1$). Поскольку число элементов не уменьшилось, выберем окончательно схему, полученную на первом этапе. Тогда $Y_D=Y(\Gamma_{y1})=\{y_1, y_4, y_6, y_9, y_{10}, y_{12}, y_{13}, y_{15}, y_{16}, y_{17}, y_{18}, y_{19}, y_{21}, y_{23}, y_{25}\}$, $Y \setminus Y_D=\{y_2, y_3, y_5, y_7, y_8, y_{11}, y_{14}, y_{20}, y_{22}, y_{24}, y_{26}, y_{27}, y_{28}\}$.

Назначим переменным из множества Y_D такие четырехразрядные коды ($V_1=4$), которыми закодированы состояния, соответствующие этим переменным. Это можно сделать, поскольку в любом состоянии МПА единичное

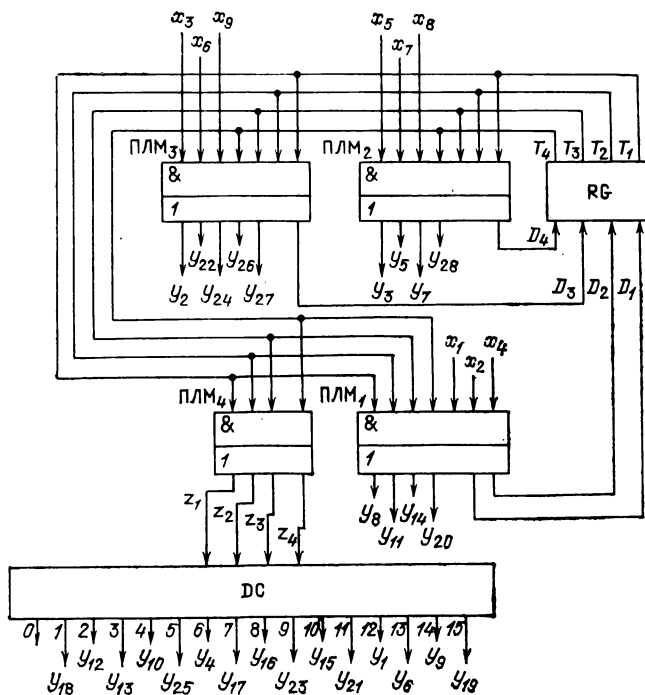


Рис. 8.6

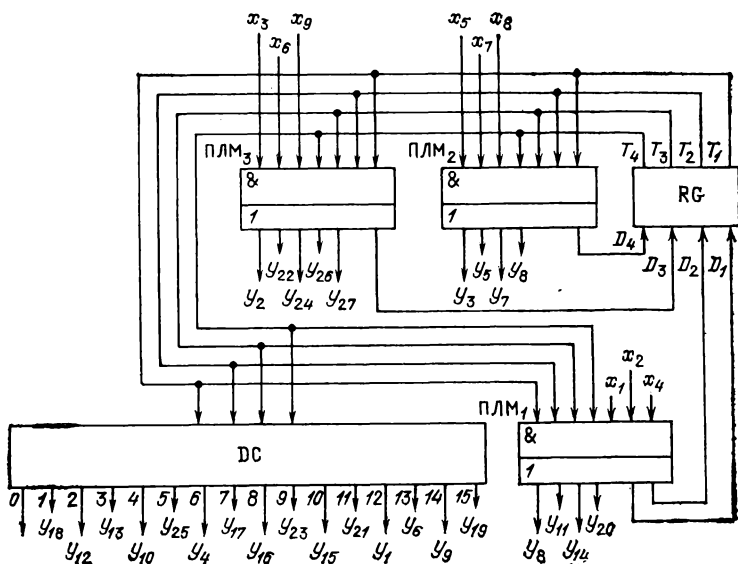


Рис. 8.7

значение может принимать лишь одна переменная из множества Y_D . В результате:

$$\begin{aligned}
 K(y_1) &= 1100; & K(y_{12}) &= 0010; & K(y_{18}) &= 0001; \\
 K(y_4) &= 0110; & K(y_{13}) &= 0011; & K(y_{19}) &= 1111; \\
 K(y_6) &= 1101; & K(y_{16}) &= 1010; & K(y_{21}) &= 1011; \\
 K(y_9) &= 1110; & K(y_{18}) &= 1000; & K(y_{28}) &= 1001; \\
 K(y_{10}) &= 0100; & K(y_{17}) &= 0111; & K(y_{25}) &= 0101
 \end{aligned}$$

и $Z = \{z_1, z_2, z_3, z_4\}$. Переменные из множеств $Z(a_m) \cup Y(a_m)$ для каждого состояния $a_m \in A$ занесены в столбец $Z(a_m)$, $Y(a_m)$ табл. 8.1. Логическая схема синтезированного автомата с PD-структурой показана на рис. 8.6, в табл. 8.2—8.5 представлена информация для программирования ПЛМ₁—ПЛМ₄ Р-подсхемы МПА. Для распределения переменных из множества $Z \cup (Y \setminus Y_D)$ между выходами ПЛМ Р-подсхемы использован метод, описанный в § 6.4. Заметим, что в рассматриваемом примере число элементов в схеме на рис. 8.6 может быть дополнительно сокращено, если подключить входы дешифратора DC (4, 16) непосредственно к выходам T_1, \dots, T_R элементов памяти МПА

Таблица 8.2

T_1	T_2	T_3	T_4	x_1	x_2	x_4	D_1	D_2	y_8	y_{11}	y_{14}	y_{20}
0	0	0	0	1	—	—	.	1
0	0	0	0	0	—	1	1
0	0	0	0	0	—	0	1	1
1	1	0	0	—	1	0	1
1	1	0	0	—	1	1	.	1
1	0	1	1	0	—	—	1
—	1	—	1	—	—	—	1	1	.	.	.	1
0	1	—	0	—	—	—	1	.	1	.	.	.
0	0	1	0	—	—	—	1	.	.	1	1	.
—	0	0	1	—	—	—	.	1	.	.	1	.
1	0	1	1	—	—	—	1	.

Таблица 8.3

T_1	T_2	T_3	T_4	x_5	x_7	x_8	D_4	y_3	y_5	y_7	y_{28}	—
0	1	0	0	0	—	—	1
0	1	0	1	—	—	0	1
0	1	1	0	—	0	—	1
1	1	1	1	—	—	0	1
0	0	—	1	—	—	—	1
0	1	1	1	—	—	—	1	.	.	.	1	.
1	1	—	0	—	—	—	1	1	1	.	.	.
0	1	1	0	—	—	—	.	.	1	.	.	.
0	0	1	1	—	—	—	.	.	.	1	1	.
1	1	0	1	—	—	—	.	.	.	1	.	.

(рис. 8.7). Информация для программирования ПЛМ₁—ПЛМ₃ в этой схеме содержится в табл. 8.2—8.4. Аналогичный прием может быть использован и в других случаях, причем если $R > V_i$, то дешифратор соединяется лишь с V_i выходами элементов памяти, а на одном из выходов Р-подсхемы формируется возбуждающий сигнал, подключающий дешифратор (или группу дешифраторов) в нужных состояниях.

Рассмотренный метод может использоваться и для синтеза автоматов Мили; он наиболее эффективен при выполнении условия $(L+R) \leq s$. Если $(L+R) > s$, то для МПА Мили возникает задача специального кодирования выход-

Таблица 8.4

T_1	T_2	T_3	T_4	x_1	x_2	x_3	D_3	y_2	y_{22}	y_{24}	y_{26}	y_{27}
1	0	0	0	—	—	0	1
1	0	0	1	0	—	—	1
0	1	1	1	—	0	—	1
0	0	1	0	—	—	0	1
0	—	0	1	—	—	—	1
0	1	0	1	—	—	—	.	1
—	1	1	0	—	—	—	1	1
1	1	0	0	—	—	—	.	1
1	0	—	0	—	—	—	.	.	1	1	.	.
1	—	1	1	—	—	—	.	.	1	.	.	.
0	1	0	0	—	—	—	1	1
1	1	1	1	—	—	—	1	.

ных переменных в каждом множестве $Y(\Gamma_{yi})$, $i = \overline{1, f}$, которая аналогична задаче кодирования внутренних состояний, рассмотренной в § 6.4.

В заключение отметим, что D-подсхема может использоваться и в многоуровневых схемах, что, например, показано в [35]. Из самого метода синтеза логической схемы МПА как одноуровневой [40], так и многоуровневой [35] следует, что D-подсхема вводится лишь тогда, когда это приводит к сокращению суммарного числа элементов схемы.

Таблица 8.5

T_1	T_2	T_3	T_4	z_1	z_2	z_3	z_4
1	—	—	—	1	.	.	.
—	1	—	—	.	1	.	.
—	—	1	—	.	.	1	.
—	—	—	1	.	.	.	1

8.4. Синтез автоматов на ПЛМ и мультиплексорах

Структура логической схемы автомата при реализации его на ПЛМ и мультиплексорах показана на рис. 8.8 (назовем ее РМ-структурой). Комбинационная

часть этой схемы состоит из двух подсхем (Р и М). Р-подсхема строится так же, как и ранее; М-подсхема позволяет заменить множество $X = \{x_1, \dots, x_L\}$ входных переменных множеством $P = \{p_1, \dots, p_G\}$ переменных значительно меньшей мощности ($G \ll L$). Предварительно изложим общий принцип построения автоматов с РМ-структурой без учета ограничений на параметры элементного базиса.

Таблица 8.6

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$P(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$
a_1	000	a_2	001	$\overline{x_1}$	$\overline{p_1}$	y_1, y_4, y_{10}	D_3
		a_3	010	$\overline{x_1 x_3}$	$\overline{p_1 p_2}$	y_1, y_{10}	D_2
		a_5	100	$\overline{x_1 x_3 x_4}$	$\overline{p_1 p_2 p_3}$	y_4	D_1
		a_6	101	$\overline{x_1 x_3 x_4 x_8}$	$\overline{p_1 p_2 p_3 p_4}$	y_1, y_4	$D_1 D_3$
		a_7	110	$\overline{x_1 x_3 x_4 x_8}$	$\overline{p_1 p_2 p_3 p_4}$	y_4, y_{10}	$D_1 D_2$
a_2	001	a_1	000	$\overline{x_{13}}$	$\overline{p_3}$	—	—
		a_5	100	$\overline{x_6 x_{13}}$	$\overline{p_1 p_3}$	y_2, y_3, y_9	D_1
		a_6	101	$\overline{x_6 x_8 x_{13}}$	$\overline{p_1 p_2 p_3}$	y_3, y_5, y_9	$D_1 D_3$
		a_8	111	$\overline{x_6 x_8 x_{13}}$	$\overline{p_1 p_2 p_3}$	y_2	$D_1 D_2 D_3$
a_3	010	a_1	000	$\overline{x_7}$	$\overline{p_1}$	—	—
		a_2	001	$\overline{x_7 x_9}$	$\overline{p_1 p_2}$	y_6, y_{11}	D_3
		a_4	011	$\overline{x_7 x_9 x_{14}}$	$\overline{p_1 p_2 p_3}$	y_7	$D_2 D_3$
		a_6	101	$\overline{x_7 x_9 x_{14}}$	$\overline{p_1 p_2 p_3}$	y_8, y_{11}	$D_1 D_3$
a_4	011	a_1	000	$\overline{x_2}$	$\overline{p_1}$	—	—
		a_4	011	$\overline{x_2 x_9}$	$\overline{p_1 p_2}$	y_4	$D_2 D_3$
		a_5	100	$\overline{x_2 x_4 x_9}$	$\overline{p_1 p_2 p_3}$	y_1	D_1
		a_8	111	$\overline{x_2 x_4 x_9}$	$\overline{p_1 p_2 p_3}$	y_1, y_4	$D_1 D_2 D_3$
a_5	100	a_1	000	$\overline{x_5}$	$\overline{p_1}$	—	—
		a_4	011	$\overline{x_5 x_6}$	$\overline{p_1 p_2}$	y_1, y_4, y_{10}	$D_2 D_3$
		a_6	101	$\overline{x_4 x_5 x_6}$	$\overline{p_2 p_2 p_3}$	y_{10}	$D_1 D_3$
		a_7	110	$\overline{x_4 x_5 x_6}$	$\overline{p_1 p_2 p_3}$	y_6	$D_1 D_2$
a_6	101	a_2	001	$\overline{x_{13}}$	$\overline{p_3}$	y_2, y_9	D_3
		a_3	010	$\overline{x_3 x_{13}}$	$\overline{p_2 p_3}$	y_3	D_2
		a_4	011	$\overline{x_3 x_{12} x_{13}}$	$\overline{p_1 p_2 p_3}$	y_5	$D_2 D_3$
		a_7	110	$\overline{x_3 x_{12} x_{13}}$	$\overline{p_1 p_2 p_3}$	y_2, y_3, y_5	$D_1 D_2$
a_7	110	a_3	010	$\overline{x_{10}}$	$\overline{p_1}$	y_6, y_7	D_2
		a_5	100	$\overline{x_{10} x_{14}}$	$\overline{p_1 p_3}$	y_7, y_8	D_1
		a_7	110	$\overline{x_{10} x_{11} x_{14}}$	$\overline{p_1 p_2 p_3}$	y_6, y_7, y_8	$D_1 D_3$
		a_8	111	$\overline{x_{10} x_{11} x_{14}}$	$\overline{p_1 p_2 p_3}$	y_{11}	$D_1 D_2 D_3$
a_8	111	a_1	000	$\overline{x_{13}}$	$\overline{p_3}$	—	—
		a_6	101	$\overline{x_{10} x_{13}}$	$\overline{p_1 p_3}$	y_2	$D_1 D_3$
		a_8	111	$\overline{x_{10} x_{13}}$	$\overline{p_1 p_3}$	y_5	$D_1 D_2 D_3$

Обозначим через $X(a_m)$ подмножество множества X входных переменных, существенно влияющих на переходы из состояния a_m . Например, для табл. 8.6: $X(a_1) = \{x_1, x_3, x_4, x_8\}$; $X(a_2) = \{x_5, x_6, x_{13}\}$; $X(a_8) = \{x_{10}, x_{13}\}$. Характерной особенностью МПА является то, что почти всегда для всех $a_m \in A$ выполняется условие $|X(a_m)| \ll$

$\ll L$ [3]. На этой особенности основывается принцип замены входных переменных.

Пусть $G = \max_{a_m \in A} |X(a_m)|$,

тогда каждую переменную $x_i \in X(a_m)$, $m \in \{1, \dots, M\}$ можно заменить переменной $p_g \in P$, $g \in \{1, \dots, G\}$; если автомат находится в состоянии a_m , то $p_g = x_i$; если он находится в состоянии a_s и $s \neq m$, то переменная p_g может принимать любое значение и, в частности, заменять переменную $x_h \in X(a_s)$. Для замены переменных построим таблицу, столбцы

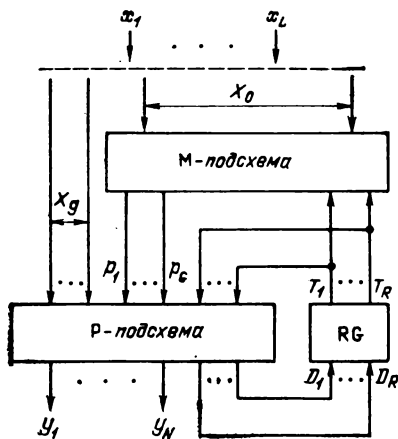


Рис. 8.8

p_1, \dots, p_G которой поставим в соответствие переменным из множества P , а строки a_1, \dots, a_M — состояниям из множества A . На пересечении строки a_m этой таблицы и столбца p_g будем записывать переменную x_i , если x_i заменяется переменной p_g в состоянии a_m . В строке a_m должны быть записаны все элементы множества $X(a_m)$, причем только по одному разу.

Замена переменных из табл. 8.6 представлена в табл. 8.7. Согласно этой таблице в состоянии a_1 переменная p_1 заменяет переменную x_1 , а в состоянии a_2 та же переменная p_1 заменяет переменную x_5 и т. д. Таблица 8.7 построена таким образом, чтобы одни и те же переменные x_i не встречались в различных

Таблица 8.7

a_m	p_g			
	p_1	p_2	p_3	p_4
a_1	x_1	x_3	x_4	x_8
a_2	x_5	x_6	x_{13}	—
a_3	x_7	x_9	x_{14}	—
a_4	x_2	x_9	x_4	—
a_5	x_5	x_6	x_4	—
a_6	x_{12}	x_3	x_{13}	—
a_7	x_{10}	x_{11}	x_{14}	—
a_8	x_{10}	—	x_{13}	—

столбцах, однако это не всегда возможно и допускается одну и ту же переменную x_i размещать в различных столбцах. В то же время выполнение этого требования способствует минимизации функций p_1, \dots, p_G , реализуемых М-подсхемой [3].

Представим функции p_1, p_2, p_3 и p_4 из табл. 8.7 в следующем виде:

$$\begin{aligned} p_1 &= a_1 x_1 \vee a_2 x_5 \vee a_3 x_7 \vee a_4 x_2 \vee a_5 x_6 \vee a_6 x_{12} \vee a_7 x_{10} \vee \\ &\quad \vee a_8 x_{10}; \\ p_2 &= a_1 x_3 \vee a_2 x_8 \vee a_3 x_9 \vee a_4 x_9 \vee a_5 x_6 \vee a_6 x_8 \vee a_7 x_{11}; \\ p_3 &= a_1 x_4 \vee a_2 x_{13} \vee a_3 x_{14} \vee a_4 x_4 \vee a_5 x_4 \vee a_6 x_{13} \vee a_7 x_{14} \vee a_8 x_{13}; \\ p_4 &= a_1 x_8, \end{aligned} \quad (8.1)$$

где, как и ранее, a_m — конъюнкция переменных T_1, \dots, T_R , равная 1, тогда и только тогда, когда МПА находится в состоянии a_m , т. е. $a_m = T_1^{e_{m1}} \dots T_R^{e_{mR}}$ (e_{m1}, \dots, e_{mR}) — значения разрядов кода $K(a_m)$ состояния a_m , $e_{mr} \in \{0, 1, -\}$, $r = 1, \bar{R}$, $T^0 = \bar{T}$, $T^1 = T$, $T^- = 1$.

Если МПА находится в состоянии a_m , то лишь одна переменная a_m из множества $\{a_1, \dots, a_M\}$ равна 1. Пусть конъюнкция $a_m x_i$ входит в ДНФ функции p_g и $a_m = 1$; в этом случае $p_g = 1$, если $x_i = 1$, и $p_g = 0$, если $x_i = 0$, поэтому, когда автомат находится в состоянии a_m , справедливо равенство $p_g = x_i$. Если на пересечении некоторого столбца p_g и строки a_m рассматриваемой таблицы записан прочерк, то это означает, что на переход из состояния a_m существенно не влияет ни одна переменная, записанная в столбце p_g , и в состоянии a_m функция p_g может принимать любое (0 или 1) значение. Таким образом, функции p_1, \dots, p_G в общем случае являются неполностью определенными, и это свойство можно использовать при их минимизации.

Выражения (8.1) после замены переменных a_m конъюнкциями $T_1^{e_{m1}} \dots T_R^{e_{mR}}$ (см. табл. 8.6) запишутся в следующем виде:

$$\begin{aligned} p_1 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_1 \vee \bar{T}_1 \bar{T}_2 T_3 x_5 \vee \bar{T}_1 T_2 \bar{T}_3 x_7 \vee \bar{T}_1 T_2 T_3 x_2 \vee T_1 \bar{T}_2 \bar{T}_3 x_6 \vee \\ &\quad \vee T_1 \bar{T}_2 T_3 x_{12} \vee T_1 T_2 \bar{T}_3 x_{10} \vee T_1 T_2 T_3 x_{10}; \\ p_2 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_3 \vee \bar{T}_1 \bar{T}_2 T_3 x_6 \vee \bar{T}_1 T_2 \bar{T}_3 x_9 \vee \bar{T}_1 T_2 T_3 x_9 \vee T_1 \bar{T}_2 \bar{T}_3 x_6 \vee \\ &\quad \vee T_1 \bar{T}_2 T_3 x_3 \vee T_1 T_2 \bar{T}_3 x_{11} \vee [T_1 T_2 T_3]; \end{aligned}$$

$$\begin{aligned}
p_3 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_4 \vee \bar{T}_1 \bar{T}_2 T_3 x_{13} \vee \bar{T}_1 T_2 \bar{T}_3 x_{14} \vee \bar{T}_1 T_2 T_3 x_4 \vee \\
&\quad \vee T_1 \bar{T}_2 \bar{T}_3 x_4 \vee T_1 \bar{T}_2 T_3 x_{13} \vee T_1 T_2 \bar{T}_3 x_{14} \vee T_1 T_2 T_3 x_{13}; \\
p_4 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_8 \vee [\bar{T}_1 \bar{T}_2 T_3 \vee \bar{T}_1 T_2 \bar{T}_3 \vee \bar{T}_1 T_2 T_3 \vee T_1 \bar{T}_2 \bar{T}_3 \vee \\
&\quad \vee T_1 \bar{T}_2 T_3 \vee T_1 T_2 \bar{T}_3 \vee T_1 T_2 T_3],
\end{aligned}$$

где в квадратных скобках записаны элементарные конъюнкции, которыми могут быть доопределены функции p_1, \dots, p_4 . После минимизации этих функций получим следующие выражения:

$$\begin{aligned}
p_1 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_1 \vee \bar{T}_1 \bar{T}_2 T_3 x_5 \vee \bar{T}_1 T_2 \bar{T}_3 x_7 \vee \bar{T}_1 T_2 T_3 x_2 \vee T_1 \bar{T}_2 \bar{T}_3 x_5 \vee \\
&\quad \vee T_1 \bar{T}_2 T_3 x_{12} \vee T_1 T_2 x_{10}; \\
p_2 &= \bar{T}_1 \bar{T}_2 \bar{T}_3 x_8 \vee \bar{T}_1 \bar{T}_2 T_3 x_6 \vee \bar{T}_1 T_2 x_9 \vee T_1 \bar{T}_2 \bar{T}_3 x_6 \vee T_1 T_3 x_5 \vee \\
&\quad \vee T_1 T_2 x_{11}; \\
p_3 &= \bar{T}_2 \bar{T}_3 x_4 \vee \bar{T}_2 T_3 x_{13} \vee T_2 \bar{T}_3 x_{14} \vee T_1 T_3 x_{13} \vee \bar{T}_1 T_2 T_3 x_4; \\
p_4 &= x_8.
\end{aligned}$$

По полученным функциям может быть построена М-подсхема на ПЛМ (s, t, q) с параметрами $s \geq 17$, $t \geq 3$, $q \geq 18$ (поскольку $p_4 = x_8$, то новую переменную p_4 можно не вводить, а значение x_8 будет подаваться непосредственно на вход Р-подсхемы). Отметим, что для построения М-подсхемы рациональнее использовать ПЛМ (z, q) , перераспределяя число входов и выходов ПЛМ таким образом, чтобы входов было много, а выходов мало.

Заменяем в структурной таблице МПА столбец $X(a_m, a_s)$ на столбец $P(a_m, a_s)$, в каждой i -й строке которого будем записывать конъюнкцию $P_i(a_m, a_s)$ переменных из множества P такую, что в состоянии a_m $P_i(a_m, a_s) = X_i(a_m, a_s)$; $X_i(a_m, a_s)$ — конъюнкция переменных из множества X , записанная в столбце $X(a_m, a_s)$ этой же i -й строки. После замены переменных синтез МПА по структурной таблице со столбцами $a_m, K(a_m), a_s, K(a_s), P(a_m, a_s), Y(a_m, a_s), F(a_m, a_s)$ выполняется, как и ранее. Например, по табл. 8.6 может быть построена Р-подсхема на одной ПЛМ (7, 14, 27).

Предположим, что для построения М-подсхемы должны быть использованы стандартные (СМ) или программируемые (ПМ) мультиплексоры. Поскольку СМ могут рассматриваться как ПМ $(s, 1, n)$, $s = 2^n$, то ограничимся рассмотрением принципов построения М-подсхемы только на

основе ПМ. Для синтеза автомата с РМ-структурой по заданной ГСА Γ на ПЛМ (s, t, q) и ПМ (s, t, n) выполняются следующие действия:

1. Производится отметка ГСА Γ по алгоритму Φ_1 или Φ_2 и определяется значение $G = \max_{a_m \in A} |X(a_m)|$. Если для

заданных ПЛМ (s, t, q) справедливо неравенство $(G + R) > s$, то выполняется дополнительная отметка ГСА Γ с целью удовлетворения условия $(G + R) \leq s$. Для дополнительной отметки может быть использован любой из ранее описанных методов.

2. По отмеченной ГСА Γ строятся структурные подтаблицы W_1, \dots, W_U , число U которых минимально, и для каждой u -й подтаблицы W_u и заданных ПЛМ (s, t, q) выполняются условия $(N_u + R_u) \leq t$, $B_u \leq q$, $u = 1, \dots, U$. Здесь $N_u = |Y^u|$; Y^u — подмножество переменных множества Y , записанных в столбце $Y(a_m, a_s)$ u -й подтаблицы переходов W_u .

3. Формируются множество X_0 входных переменных, значения которых подаются на входы М-подсхемы, и множество X_g входных переменных, значения которых подаются непосредственно на входы Р-подсхемы: $X_0 \subseteq X$; $X_g \subseteq X$; $X_0 \cup X_g = X$ (допустимо $X_0 \cap X_g \neq \emptyset$). Минимизируется число элементов в множестве X_0 и значение переменной G .

4. Выполняется синтез М-подсхемы, реализующей замену переменных из множества X_0 переменными из множества $P = \{p_1, \dots, p_G\}$, с учетом того, что значение G скорректировано в п. 3.

5. Синтезируется Р-подсхема по подтаблицам W_1, \dots, W_U , в которых часть, а возможно, и все переменные из множества X заменены переменными из множества P .

Ниже рассматриваются все перечисленные этапы, изложение сопровождается примером синтеза логической схемы МПА S с РМ-структурой по ГСА Γ , изображенной на рис. 8.9. В качестве элементов схемы используются ПЛМ $(7, 6, 12)$ и ПМ $(7, 2, 2)$. ГСА Γ на рис. 8.9 отмечена по алгоритму Φ_1 метками a_1, \dots, a_8 . Затем строится структурная таблица МПА [табл. 8.6, столбцы $a_m, K(a_m), a_s, K(a_s), X(a_m, a_s), Y(a_m, a_s), F(a_m, a_s)$], реализующего эту ГСА. Заметим, что табл. 8.6 уже использовалась выше, внутренние состояния автомата в столбцах $K(a_m)$ и $K(a_s)$ закодированы произвольным образом (этот вариант кодирования является далеко не лучшим и ниже будет изменен). Для табл. 8.6 $G=4$, $R=3$ и $G+R=s$.

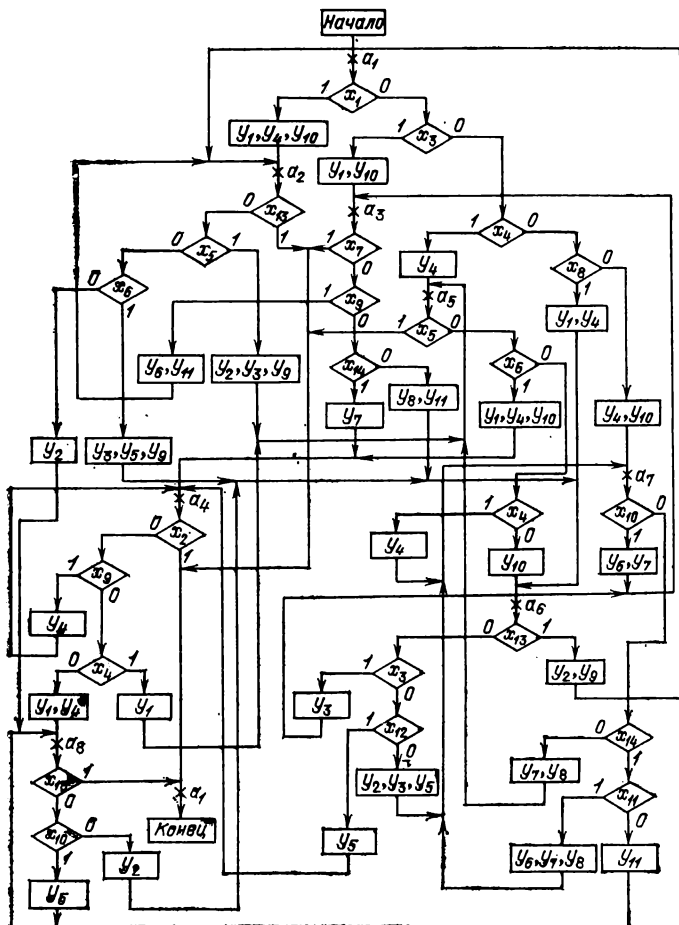


Рис. 8.9

Учитывая то, что $N=11$, $R=3$ и $(N+R) > t$, структурная таблица согласно п. 2 должна быть разделена на такие подтаблицы, в которых для каждой u -й из них выполняются условия $(N_u + R_u) \leq t$, $B_u \leq q$. Указанная задача может быть решена с помощью любого метода, описанного в гл. 6. Здесь мы воспользуемся следующим простым приемом. Введем на множестве A внутренних состояний автомата отношение φ , определенное в § 7.3: $(a_m \varphi a_s) \leftrightarrow Y(a_m) \cap Y(a_s) \neq \emptyset$. Граф Γ_φ отношения φ для нашего

примера показана на рис. 8.10. Сформируем разбиение π_Φ (см. § 7.3):

$$\pi_\Phi = \overline{a_1, a_4, a_5}; \overline{a_2, a_6, a_8}; \overline{a_3, a_7}$$

— и построим U подтаблиц переходов W_1, \dots, W_U . В каждую u -ю подтаблицу включим все переходы из состояний, принадлежащих u -му блоку разбиения π_Φ . Закодируем внутренние состояния автомата с помощью метода, рассмотренного в § 6.4. Матрицы E^2, F^2, E^3, F^3 представятся в виде:

$$E^2 = \begin{matrix} T_1 & T_2 & T_3 \\ | 1 & 1 & 1 | \end{matrix}; F^2 = \begin{matrix} D_2 \\ | 1 | \end{matrix}; E^3 = \begin{matrix} T_1 & T_2 & T_3 \\ | 1 & 0 & 0 | \end{matrix}; F^3 = \begin{matrix} D_3 \\ | 1 | \end{matrix}.$$

Полученные структурные подтаблицы W_1, W_2, W_3 приведены в табл. 8.8 [столбцы $a_m, K(a_m), a_s, K(a_s), X(a_m, a_s), Y(a_m, a_s), F(a_m, a_s)$], как и ранее, они отделены друг от друга двойными горизонтальными линиями. Для каждой u -й подтаблицы ($u=1, 3$) выполняются условия п. 2: $N_u + R_u = t, B_u < q$. Здесь мы не останавливаемся подроб-

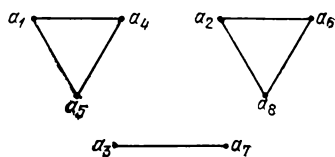


Рис. 8.10

но на том, как были построены подтаблицы, соответствующие способы были рассмотрены в предыдущих главах.

При построении M -подсхемы минимизируется число составляющих ее ПМ (s, t, n). Для этого будем решать ряд частных задач в соответствии с представленными ниже алгоритмами Φ_6, Φ_7 и Φ_8 , которые позволяют разделить входные переменные между входами M - и P -подсхем таким образом, чтобы уменьшить число ПМ и ПЛМ этих подсхем.

С помощью перемещения элементов из X_0 в X_g в алгоритме Φ_6 минимизируется число G выходов, а в алгоритме Φ_7 — число входов M -подсхемы. В обоих алгоритмах множество X входных переменных разбивается на два непересекающихся подмножества X_0 и X_g . В алгоритме Φ_8 некоторые входные переменные, оставаясь в X_0 , передаются в множество X_g , что также может привести к сокращению числа выходов M -подсхемы. Таким образом, в алгоритмах Φ_6 и Φ_7 множество $\{X_0, X_g\}$ есть разбиение, а для алгоритма Φ_8 — покрытие множества X . Необходимость выполнения того или иного алгоритма определяется ограничениями на число входов или выходов элементов

Таблица 8.8

a_m	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$XP(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$
a_1	000	a_2	001	$\overline{x_1}$	x_1	y_1, y_4, y_{10}	D_3
		a_3	011	$\overline{x_1 x_3}$	$\overline{x_1 p_1}$	y_1, y_{10}	$D_2 D_3$
		a_5	010	$\overline{x_1 x_3 x_4}$	$\overline{x_1 p_1 x_4}$	y_4	D_2
		a_6	100	$\overline{x_1 x_3 x_4 x_8}$	$\overline{x_1 p_1 x_4 p_2}$	y_1, y_4	D_1
		a_7	111	$\overline{x_1 x_3 x_4 x_8}$	$\overline{x_1 p_1 x_4 p_2}$	y_4, y_{10}	$D_1 D_2 D_3$
a_4	101	a_1	000	x_2	p_2	—	—
		a_4	101	$\overline{x_2 x_9}$	$\overline{p_1 p_2}$	y_4	$D_1 D_3$
		a_5	010	$\overline{x_2 x_4 x_9}$	$\overline{p_1 p_2 x_4}$	y_1	D_2
		a_8	110	$\overline{x_2 x_4 x_9}$	$\overline{p_1 p_2 x_4}$	y_1, y_4	$D_1 D_2$
a_5	010	a_1	000	x_5	p_2	—	—
		a_4	101	$\overline{x_5 x_6}$	$\overline{p_1 p_2}$	y_1, y_4, y_{10}	$D_1 D_3$
		a_6	100	$\overline{x_4 x_5 x_6}$	$\overline{x_4 p_1 p_2}$	y_{10}	D_1
		a_7	111	$\overline{x_4 x_5 x_6}$	$\overline{x_4 p_1 p_2}$	y_4	$D_1 D_2 D_3$
a_2	001	a_1	000	x_{13}	x_{13}	—	—
		a_5	010	$\overline{x_5 x_{13}}$	$\overline{x_5 x_{13}}$	y_2, y_3, y_9	D_2
		a_6	100	$\overline{x_5 x_6 x_{13}}$	$\overline{x_5 p_1 x_{13}}$	y_3, y_5, y_6	D_1
		a_8	110	$\overline{x_5 x_6 x_{13}}$	$\overline{x_5 p_1 x_{13}}$	y_2	$D_1 D_2$
a_6	100	a_2	001	x_{13}	x_{13}	y_2, y_9	(D_3)
		a_3	011	$\overline{x_3 x_{13}}$	$\overline{p_1 x_{13}}$	y_3	$D_2 (D_3)$
		a_4	101	$\overline{x_3 x_{12} x_{13}}$	$\overline{p_1 x_{12} x_{13}}$	y_5	$D_1 (D_3)$
		a_7	111	$\overline{x_3 x_{12} x_{13}}$	$\overline{p_1 x_{12} x_{13}}$	y_2, y_3, y_5	$D_1 D_2 (D_3)$
a_8	110	a_1	000	x_{13}	x_{13}	—	—
		a_6	100	$\overline{x_{10} x_{13}}$	$\overline{p_1 x_{13}}$	y_2	D_1
		a_8	110	$\overline{x_{10} x_{13}}$	$\overline{p_1 x_{13}}$	y_5	$D_1 D_2$
a_3	011	a_1	000	x_7	x_7	—	—
		a_2	001	$\overline{x_7 x_9}$	$\overline{x_7 p_1}$	y_6, y_{11}	D_3
		a_4	101	$\overline{x_7 x_9 x_{14}}$	$\overline{x_7 p_1 x_{14}}$	y_7	$D_1 D_3$
		a_6	100	$\overline{x_7 x_9 x_{14}}$	$\overline{x_7 p_1 x_{14}}$	y_6, y_{11}	D_1
a_7	111	a_3	011	$\overline{x_{10}}$	$\overline{p_1}$	y_6, y_7	$(D_2) D_3$
		a_5	010	$\overline{x_{10} x_{14}}$	$\overline{p_1 x_{14}}$	y_7, y_8	(D_2)
		a_7	111	$\overline{x_{10} x_{11} x_{14}}$	$\overline{p_1 x_{11} x_{14}}$	y_6, y_7, y_8	$D_1 D_2 (D_3)$
		a_8	110	$\overline{x_{10} x_{11} x_{14}}$	$\overline{p_1 x_{11} x_{14}}$	y_{11}	$D_1 (D_2)$

М-подсхемы. Если ограничения на число входов этих элементов более жесткие, чем на число выходов, то необходимо последовательно выполнять алгоритмы Φ_7 и Φ_8 ; при более жестких ограничениях на число выходов — алгоритмы Φ_6 и Φ_8 . В общем случае, когда приоритет ограничений на число входов и выходов не может быть установлен, то последовательно выполнять алгоритмы Φ_6 , Φ_7 и Φ_8 .

Алгоритм Φ_6 .

1. Пусть X^u — подмножество множества X входных переменных, записанных в столбце $X(a_m, a_s)$ подтаблицы переходов W_u . Положим $X_0 = X$, $X_g = \emptyset$, $G = \max_{a_m \in A} |X(a_m)|$.

Напомним, что X_0 и X_g — это множества входных переменных, значения которых передаются соответственно на входы М- и Р-подсхем. Назначим каждой переменной $x_i \in X_0$ вес ρ_i , равный числу подмножеств X^u ($u = \overline{1, U}$), в которые входит x_i .

2. Входные переменные с минимальным суммарным весом переносим из множества X_0 в множество X_g , тем самым значение G уменьшается на 1^* ($G := G - 1$). Элементы множества X_g удаляем из всех множеств $X(a_m)$. Обозначим $X^u = X^u \cap X_g$, $g_u = |X^u|$.

3. Если для всех $u = \overline{1, U}$ выполняется условие $(g_u + G_u) \leq (s - R)$, где $G_u = \max_{a_m \in A_u} |X(a_m)|$, то переходим к

п. 2, в противном случае возвращаем предыдущее значение G , т. е. $G := G + 1$; элементы, переданные при последнем выполнении п. 2 (из X_0 в X_g), возвращаем из X_g в X_0 и переходим к п. 4. Напомним, что A_u — это множество состояний, записанных в столбце a_m подтаблицы W_u .

4. Фиксируем значение $G = \max_{a_m \in A} |X(a_m)|$.

Выполним алгоритм Φ_6 для рассматриваемого примера (см. табл. 8.8). Предварительно $X_0 = \{x_1, \dots, x_{14}\}$, $X_g = \emptyset$. Вычисляем веса ρ_1, \dots, ρ_{14} переменных x_1, \dots, x_{14} : $\rho_1 = \rho_2 = \rho_4 = \rho_7 = \rho_8 = \rho_{11} = \rho_{12} = \rho_{13} = \rho_{14} = 1$; $\rho_3 = \rho_5 = \rho_6 = \rho_9 = \rho_{10} = 2$. Например, $\rho_1 = 1$, поскольку переменная x_1 включена лишь в одно подмножество X^1 , а $\rho_3 = 2$, так как переменная x_3 включена в два подмножества: X^1 и X^2 . образуем множество переменных $X(\rho_{\min}) = \{x_1, x_2, x_4, x_7, x_8, x_{11}, x_{12}, x_{13}, x_{14}\}$, вес которых минимален и равен единице. Рас-

* Учитывая то, что в качестве элементов М-подсхемы используются ПМ (s, t, n), значение G правильнее уменьшить не на 1, а настолько, чтобы первоначальное значение $|G/t|$ сократилось бы на 1. Если при этом $|G/t| = 1$, то далее выполнять алгоритм Φ_6 нецелесообразно.

смотрим подтаблицу W_1 и выберем из множества $X(\rho_{\min})$ первую (например, слева) переменную, которая имеется в W_1 . Такой переменной будет x_1 . Если x_1 удалить из множества X_0 и включить в множество X_g , то $X(a_1) = \{x_3, x_4, x_8\}$, $G = \max_{a_m \in A} |X(a_m)| = 3$. В результате $X_0 = \{x_2, \dots, x_{14}\}$, $X^1 = X_g = \{x_1\}$, $X^2 = X^3 = \emptyset$. Удалим переменную x_1 из множества $X(\rho_{\min})$. Затем снова рассмотрим подтаблицу W_1 и выберем из множества $X(\rho_{\min})$ первую переменную (x_4), которая имеется в W_1 . Эта переменная может быть включена в множество X^1 [$G_1 = \max_{a_m \in A_1} |X(a_m)| = 2$, $g_1 + G_1 = s - R$].

Рассмотрим подтаблицу W_2 и выберем из множества $X(\rho_{\min})$ первую переменную (x_{12}), которая имеется в W_2 . Однако если включить x_{12} в X^2 , то значение переменной G_2 не уменьшается ($G_2 = 3$), поэтому временно оставим переменную x_{12} и перейдем к следующей переменной x_{13} . Включение x_{13} в X^2 приводит к сокращению значения G_2 на единицу ($G_2 = 2$), в результате $X^2 = \{x_{13}\}$. Выполняя аналогичные действия, получаем $X^3 = \{x_{14}\}$. После этого $X_0 = \{x_2, x_3, x_5, \dots, x_{12}\}$, $X_g = \{x_1, x_4, x_{13}, x_{14}\}$, $X^1 = \{x_1, x_4\}$, $X^2 = \{x_{13}\}$, $X^3 = \{x_{14}\}$, $G = G_1 = G_2 = G_3 = 2$. Далее значение $G = 2$ сократить не удается.

Последовательность выполнения алгоритма Φ_6 иллюстрируется табл. 8.9 (столбцы 1—3). В столбце 1 записаны элементарные конъюнкции из столбца $X(a_m, a_s)$ табл. 8.8. В столбце 2 из этих конъюнкций удалена переменная x_1 , а в столбце 3 — переменные x_4, x_{13}, x_{14} . После этого в любом подмассиве переходов (см. столбец 3 табл. 8.9) записано не более двух различных переменных множества X , т. е. $G = 2$. Формирование множеств X_g, X^1, X^2, X^3 отражено в нижней части табл. 8.9.

В целом выполнение алгоритма Φ_6 сводится к перебору различных вариантов с целью сокращения значения G_u для каждой подтаблицы W_u в отдельности до $G = 1$.

Алгоритм Φ_7 .

1. Для каждого элемента $x_i \in X_0$ проверяем возможность его переноса в множество X_g , причем в первую очередь переносим элементы x_i с минимальным весом ρ_i . Элемент x_i можно перенести в множество X_g , если при $x_i \in X_g$ для всех $u = \overline{1, U}$ выполняется условие $(g_u + G_u) \leq (s - R)$. Если ни один элемент множества X_0 нельзя перенести в множество X_g , переходим к п. 2.

2. Фиксируем значение $G = \max_{a_m \in A} |X(a_m)|$, $a_m \in A$.

$X(a_m, a_s)$					
1	2	3	4	5	6
$\frac{x_{10}}{x_{10}x_{14}}$	$\frac{x_{10}}{x_{10}x_{14}}$	$\frac{x_{10}}{x_{10}}$	$\frac{x_{10}}{x_{10}}$	$\frac{x_{10}}{x_{10}}$	$\frac{x_{10}}{x_{10}}$
$\frac{x_{10}x_{11}x_{14}}{x_{10}x_{11}x_{14}}$	$\frac{x_{10}x_{11}x_{14}}{x_{10}x_{11}x_{14}}$	$\frac{x_{10}x_{11}}{x_{10}x_{11}}$	$\frac{x_{10}x_{11}}{x_{10}x_{11}}$	$\frac{x_{10}}{x_{10}}$	$\frac{x_{10}}{x_{10}}$
$X_g = \emptyset,$ $\tilde{X}^1 = \tilde{X}^2 =$ $= \tilde{X}^3 = \emptyset$	$X_g = \tilde{X}^1 =$ $= \{x_1\},$ $\tilde{X}^2 = \tilde{X}^3 =$ $= \emptyset$	$X_g = \{x_1,$ $x_4, x_{13}, x_{14}\},$ $\tilde{X}^1 = \{x_1,$ $x_4\},$ $\tilde{X}^2 = \{x_{13}\},$ $\tilde{X}^3 = \{x_{14}\}$	$X_g = \{x_1,$ $x_4, x_7, x_{13},$ $x_{14}\},$ $\tilde{X}^1 = \{x_1,$ $x_4\},$ $\tilde{X}^2 = \{x_{13}\},$ $\tilde{X}^3 = \{x_7,$ $x_{14}\}$	$X_g = \{x_1,$ $x_4, x_7,$ $x_{11}, x_{12},$ $x_{13}, x_{14}\},$ $\tilde{X}^1 = \{x_1,$ $x_4\},$ $\tilde{X}^2 = \{x_{12},$ $x_{13}\},$ $\tilde{X}^3 = \{x_7,$ $x_{11}, x_{14}\}$	$X_g = \{x_1,$ $x_4, x_5, x_7,$ $x_{11}, x_{12},$ $x_{13}, x_{14}\},$ $\tilde{X}^1 = \{x_1,$ $x_4\},$ $\tilde{X}^2 = \{x_5,$ $x_{13}, x_{13}\},$ $\tilde{X}^3 = \{x_7,$ $x_{11}, x_{14}\}$

В нашем примере с учетом выполнения алгоритма Φ_6 $X(p_{\min}) = \{x_2, x_7, x_8, x_{11}, x_{12}\}$. Непосредственно из табл. 8.9 (столбец 3) видно, что переменные x_2 и x_8 не могут быть включены в множество X_g , поскольку это приведет к тому, что $g_1 + G_1 = 3 + 2 = 5 > s - R = 4$. Любая из оставшихся переменных (x_7, x_{11} или x_{12}) может быть включена в множество X_g , причем во всех случаях значения G_1, G_2, G_3 не изменяются. Включим в множество X_g первую из них — x_7 (столбец 4 табл. 8.9). Из этого столбца видно, что на последующих двух шагах в множество X_g могут быть включены обе оставшиеся переменные: x_{11} и x_{12} (столбец 5 табл. 8.9). После этого $X_0 = \{x_2, x_3, x_5, x_6, x_8, x_9, x_{10}\}$, $X_g = \{x_1, x_4, x_7, x_{11}, x_{12}, x_{13}, x_{14}\}$, и далее ни одна переменная из множества X_0 не может быть передана в множество X_g .

Алгоритм Φ_8 .

1. Полагаем $u=1$.
2. Каждой переменной $x_i \in X^u$ назначаем вес $\rho_i^u (G_u - K)$, равный числу таких подмножеств $X(a_m)$, что x_i включена в $X(a_m)$, $a_m \in A_u$ и $|X(a_m)| = G_u - K$; K — целое неотрицательное число, $K \leq G_u$.
3. Полагаем $K=0, X_a = X^u$.

4. Из множества X_a выделим подмножество X_p переменных с максимальным значением веса $\rho_i^u(G_u - K)$.

5. Если $|X_p| = 1$, то переменную из множества X_p заносим в множество \bar{X}^u . Если $|X_p| > 1$ и $K = G_u$, то любую переменную из множества X_p заносим в множество \bar{X}^u . Все элементы множества \bar{X}^u удаляем из множеств $X(a_m)$, для которых $a_m \in A_u$. Переходим к п. 6.

Если $|X_p| > 1$ и $K < G_u$, то полагаем $K := K + 1$, $X_a = X_p$, после чего переходим к п. 4.

6. Если $(g_u + G_u) \leq (s - R)$, то переходим к п. 2, в противном случае переменную, выбранную при последнем выполнении п. 2, удаляем из множества \bar{X}^u и переходим к п. 7.

7. Если $u = U$, то переходим к п. 8, в противном случае увеличиваем u ($u := u + 1$) и переходим к п. 2.

8. Фиксируем значения $G = \max_{a_m \in A} |X(a_m)|$ и $X_0 = \bigcup_{a_m \in A} X(a_m)$.

Продолжим рассмотрение нашего примера. Пусть $u = 1$, $K = 0$, $X_a = X^1 = \{x_2, x_3, x_5, x_6, x_8, x_9\}$. Так как $G_1 = 2$, то $G_1 - K = 2$. Тогда $\rho_2^1(2) = \rho_3^1(2) = \rho_5^1(2) = \rho_6^1(2) = \rho_8^1(2) = \rho_9^1(2) = 1$ и, если $K \in \{1, 2\}$, для всех $i \in \{2, 3, 5, 6, 8, 9\}$ вес $\rho_i^1(G_1 - K) = 0$. Включение любой одной переменной из X_a в \bar{X}^1 недопустимо, поскольку приводит к тому, что $g_1 + G_1 = 3 + 2 = 5 > s - R = 4$. Пусть $u = 2$, $K = 0$, $X_a = X^2 = \{x_3, x_5, x_6, x_{10}\}$. Тогда $X_p = \{x_5, x_6\}$, так как веса переменных x_5 и x_6 максимальны [$\rho_5^2(2) = \rho_6^2(2) = 1$, $\rho_3^2(2) = \rho_{10}^2(2) = 0$]. Легко убедиться в том, что любая переменная x_5 или x_6 может быть включена в множество \bar{X}^2 ; выберем первую из них, т. е. x_5 . После этого (см. столбец 6 табл. 8.9) $X_0 = \{x_2, x_3, x_5, x_6, x_8, x_9, x_{10}\}$, $X_g = \{x_1, x_4, x_5, x_7, x_{11}, x_{12}, x_{13}, x_{14}\}$, $\bar{X}^1 = \{x_1, x_4\}$, $\bar{X}^2 = \{x_5, x_{12}, x_{13}\}$, $\bar{X}^3 = \{x_7, x_{11}, x_{14}\}$, $G = G_1 = 2$, $G_2 = G_3 = 1$. Последующие действия согласно алгоритму Φ_8 не позволяют расширить множество X_g .

После формирования множеств X_0 и X_g значение $\lfloor G/t \rfloor$ определяет число групп ПМ в М-подсхеме. Синтез М-подсхемы сводится к минимизации числа ПМ в каждой группе, для чего заполняются специальные таблицы (табл. 8.10). В столбцах p_1, \dots, p_G табл. 8.10 записываются переменные из множества X_0 , которые заменяются переменными p_1, \dots, p_G . Все переменные, включенные в одно множество $X(a_m)$, должны быть записаны в одной строке, а все элементы множества X_0 должны быть записаны в таблице. Если $x_k \in X^u$, $x_k \in X_0$, $x_k \in X(a_m)$, $a_m \in A_u$ и на u -ю ПЛМ передаются лишь значения переменных $P^u \subset P$, то все пере-

менные из множества $X(a_m)$ записываются в одной строке и переменная x_k должна быть обязательно записана в столбце табл. 8.10, отмеченном символом из множества P^u . Разрешается любую переменную из множества X_0 записывать в табл. 8.10 несколько раз.

В столбце «Состояние» каждой j -й строки записываются состояния $a_m \in A_u$ ($u=1, \bar{U}$) МПА, для которых в столбцах из множества P^u этой j -й строки записаны все переменные из множества $X(a_m)$. Каждое состояние a_m , для которого $X(a_m) \neq \emptyset$, должно быть записано лишь в

Таблица 8.10

Код	Состояние	P_g		
		p_1	...	p_G
000			...	
001			...	
010			...	
011			...	
100			...	
101			...	
110			...	
111			...	

одной из строк таблицы. Состояния a_m , для которых $X(a_m) = \emptyset$, не записываются. Столбцы табл. 8.10 заполняются сверху вниз. Число ПМ в каждой группе¹ с номером $1, \dots, G/t$ [определяется цифрой, записанной справа от таблицы около группы строк, в которую попала последняя (самая нижняя) входная переменная соответственно в столбцах p_1, \dots, p_G . Число строк в группе для ПМ, имеющего n управляющих входов, равно 2^n (для табл. 8.10 $n=2$).

Обозначим через λ множество столбцов табл. 8.10, соответствующих выходам одного ПМ. Тогда в этих столбцах для каждой группы строк может быть записано не более чем s различных входных переменных из множества X_0 . Минимизация числа ПМ сводится к такому заполнению таблицы, при котором входные переменные попадали бы в группы строк, отмеченные как можно меньшей цифрой (рассматриваемая таблица уплотняется кверху). Затем строки кодируются χ -разрядными двоичными кодами, где χ — минимальное число двоичных разрядов, достаточное для кодирования попарно-неравными двоичными кодами всех заполненных строк. Полученные χ -разрядные коды используются для управления программируемыми мультимплексорами М-подсхемы и формируются специальной схемой, подключенной к выходам памяти автомата; структура этой схемы будет уточнена ниже. На информа-

¹ Обычно число групп G/t равно 1; по крайней мере, при необходимости любая ГСА Γ всегда может быть отмечена так, чтобы для любого a_m выполнялось условие $|X(a_m)| < t$.

ционные входы ПМ каждой i -й группы подаются значения входных переменных, записанных в столбцах табл. 8.10, описывающих выходы соответствующего ПМ. Номер информационного выхода определяется исходя из того, чтобы двоичный код соответствующей строки мог быть использован для передачи значения переменной, записанной на пересечении i -го столбца и этой строки, на выход p_i .

Продолжим рассмотрение примера и произведем замену переменных из множества X_0 переменными из множества $P = \{p_1, p_2\}$ ($G=2$) с помощью описанных таблиц. Такая замена сделана в табл. 8.11. В результате ее заполнения $\chi=2$. Заметим, что $P^2 = P^3 = \{p_1\}$, поэтому переменные из множества $X^2 = \{x_3, x_6, x_{10}\}$ и $X^3 = \{x_9, x_{10}\}$ встречаются лишь в столбце p_1 табл. 8.11.

Таблица 8.11

Код	Состояние	P_g	
		p_1	p_2
00	a_1, a_6	x_3	x_8
01	a_4, a_3	x_9	x_2
10	a_5, a_2	x_6	x_5
11	a_8, a_7	x_{10}	—

После выполнения рассмотренных действий М-подсхема может быть построена на одном ПМ (7, 2, 2), его выходы p_1 и p_2 описывают столбцы p_1 и p_2 табл. 8.11. Входными переменными Р-подсхемы будут переменные из множества $XP = \{x_1, x_4, x_5, x_7, x_{11}, x_{12}, x_{13}, x_{14}, p_1, p_2\}$. Конъюнкции этих переменных, равные единице на переходах МПА из состояния

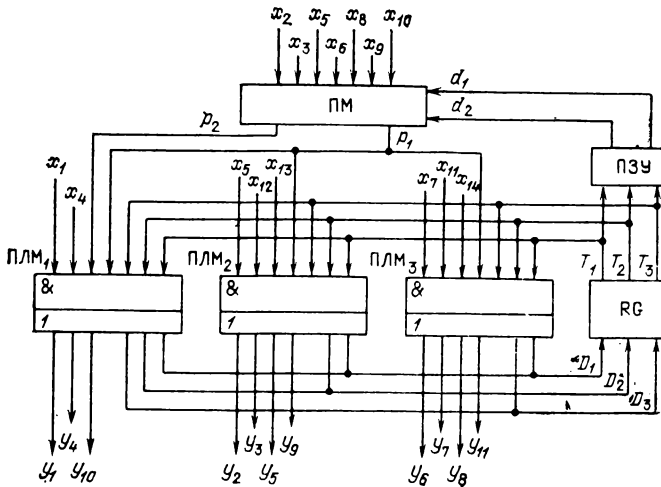


Рис. 8.11

Таблица 8.12

T_1	T_2	T_3	x_1	x_4	p_1	p_2	D_1	D_2	D_3	y_1	y_4	y_{10}
0	0	0	1	—	—	—	.	.	1	1	1	1
0	0	0	0	—	1	—	.	1	1	1	.	1
0	0	0	0	1	0	—	.	1	.	.	1	.
0	0	0	0	0	0	1	1	.	.	1	1	.
0	0	0	0	0	0	0	1	1	1	.	1	1
1	0	1	—	—	1	0	1	.	1	.	1	.
1	0	1	—	1	0	0	.	1	.	1	.	.
1	0	1	—	0	0	0	1	1	.	1	1	.
0	1	0	—	—	1	0	1	.	1	1	1	1
0	1	0	—	0	0	0	1	1
0	1	0	—	1	0	0	1	1	1	.	1	.

Таблица 8.13

T_1	T_2	T_3	x_5	x_{12}	x_{13}	p_1	D_1	D_2	y_2	y_3	y_6	y_9
0	0	1	1	—	0	—	.	1	1	1	.	1
0	0	1	0	—	0	1	1	.	.	1	1	1
0	0	1	0	—	0	0	1	1	1	.	.	.
1	0	0	—	—	1	—	.	.	1	.	.	1
1	0	0	—	—	0	1	.	1	.	1	.	.
1	0	0	—	1	0	0	1	.	.	.	1	.
1	0	0	—	0	0	0	1	1	1	1	1	.
1	1	0	—	—	0	0	1	.	1	.	.	.
1	1	0	—	—	0	1	1	1	.	.	1	.
1	1	1	—	—	—	—	.	1

Таблица 8.14

T_1	T_2	T_3	x_7	x_{11}	x_{14}	p_1	D_1	D_3	y_6	y_7	y_8	y_{11}
0	1	1	0	—	—	1	.	1	1	.	.	1
0	1	1	0	—	1	0	1	1	.	1	.	.
0	1	1	0	—	0	0	1	.	.	.	1	1
1	1	1	—	—	—	1	.	1	1	1	.	.
1	1	1	—	—	0	0	.	.	.	1	1	.
1	1	1	—	1	1	0	1	1	1	1	1	.
1	1	1	—	0	1	0	1	1
1	0	0	—	—	—	—	.	1

a_m в состоянии a_s , записаны в соответствующих строках столбца $XP(a_m, a_s)$ табл. 8.8.

Синтез Р-подсхемы по структурным подтаблицам W_1, W_2, W_3 со столбцами $a_m, K(a_m), a_s, K(a_s), XP(a_m, a_s), Y(a_m, a_s), F(a_m, a_s)$ выполняется так же, как и ранее. Логическая схема синтезированного автомата с РМ-структурой показана на рис. 8.11. Настроенный ПМ (7, 2, 2), на основе которого выполнена М-подсхема, изображен на рис. 8.12. Аналогичная М-подсхема может быть построена на двух стандартных мультиплексорах M_1 и M_2 с четырьмя информационными D_1, \dots, D_4 и двумя управляющими V_1, V_2 входами.

Таблица 8.15

Адрес	d_1	d_2
1	1	.
2	1	.
3	.	1
5	.	1
6	1	1
7	1	1

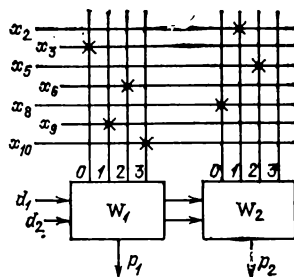


Рис. 8.12

В схему на рис. 8.11 введен дополнительный элемент — ПЗУ (3, 2), которое предназначено для преобразования значений переменных T_1, T_2, T_3 в двоичные коды управления М-подсхемой, представленные значениями переменных d_1 и d_2 . Правила такого преобразования определены табл. 8.11; для каждого состояния, записанного в i -й строке этой таблицы, ПЗУ формирует код i -й строки. Например, для состояния a_1 будет сформирован код 00 ($d_1=d_2=0$), а для a_2 — код 10 ($d_1=1, d_2=0$). Дополнительный элемент не требуется, если соответствующие преобразования могут быть выполнены в самой М-подсхеме, построенной, например, на ПЛМ (z, q). Ясно также, что в качестве такого элемента могут быть использованы не только ПЗУ, но и любые другие схемы, позволяющие реализовать подобное преобразование.

Информация для программирования ПЛМ₁ — ПЛМ₃ Р-подсхемы и дополнительного ПЗУ (3, 2) представлена в табл. 8.12—8.15 соответственно.

**БАЗОВЫЕ СХЕМЫ АВТОМАТОВ
НА ПРОГРАММИРУЕМЫХ
ЛОГИЧЕСКИХ УСТРОЙСТВАХ**

9.1. Принцип построения базовых схем

Общий процесс проектирования схемы МПА и изготовления соответствующего управляющего устройства включает следующие этапы:

1) разработку алгоритма функционирования проектируемого дискретного устройства и представление его на каком-либо формальном языке, например на языке ГСА;

2) проектирование логической схемы МПА, реализующего заданный алгоритм, с использованием одного из известных формальных методов (см., например, гл. 6—8) и преобразование ее в электрическую схему, для которой удовлетворены ограничения по нагрузочным способностям элементов, расставлены номера контактов, разведены шины питания и т. д.;

3) разработку конструкции одного или нескольких типовых элементов замены (ТЭЗ) проектируемого МПА, для чего решаются задачи компоновки элементов схемы в подсхемы заданной сложности, размещения элементов каждой подсхемы в монтажном пространстве ограниченного объема, трассировки проводящих покрытий для организации необходимых связей между соответствующими элементами; изготовление платы ТЭЗ и установку на этой плате модулей интегральных микросхем.

Операции первого этапа носят творческий характер, поскольку они требуют интеллектуальных усилий инженера-проектировщика. Сам процесс составления алгоритмов плохо автоматизируется. Операции второго этапа практически полностью формализованы и достаточно просто решаются на ЭВМ. Большинство задач конструкторского (третьего) этапа полностью автоматизировано, однако процесс разработки и изготовления печатных плат весьма трудоемок и продолжителен. Проектирование конструкции ТЭЗ может быть начато лишь после того, когда полностью разработана электрическая схема автомата. Другими словами, задачи логического и конструкторского этапов не могут быть решены параллельно и время, отведенное на разработку и изготовление ТЭЗ МПА, существенно удлинит общий процесс проектирования и изготовления

соответствующего устройства. Все это является причиной того, что разработчики цифровой аппаратуры стремятся применять типовые (базовые) ТЭЗ, т. е. такие, которые могут быть использованы многократно для устройств, реализующих различные функции. Особенно актуальным является применение базовых ТЭЗ при малой серийности разрабатываемых устройств. В настоящей главе излагаются методы построения базовых ТЭЗ для МПА, реализованных на программируемых БИС с матричной структурой.

Для разработки базового ТЭЗ выполняются следующие действия [31, 41]:

- 1) проектируется ТЭЗ, на котором фиксированы расположение (посадочные места) и типы элементов;
- 2) проектируется печатный монтаж ТЭЗ, конфигурация которого сохраняется неизменной для устройств, реализующих различные функции управления (поведение таких устройств задано различными ГСА);
- 3) синтез МПА по конкретной ГСА сводится лишь к получению информации о настройке элементов — программируемых БИС с матричной структурой, устанавливаемых на заранее заданных посадочных местах базового ТЭЗ без изменения печатного монтажа.

Оптимизация схемы МПА понимается как минимизация числа устанавливаемых на плате элементов. Для разработки структуры базового ТЭЗ находятся значения следующих параметров микропрограммных автоматов [31] (будем называть их определяющими):

1. Максимальное число выходных переменных МПА N_{\max} , совпадающее с максимальным числом различных микроопераций, среди которых нет тождественно равных, в одной ГСА. Напомним, что тождественно равными, или эквивалентными, считаются микрооперации, которые встречаются в любой микрокоманде всегда вместе. Любое множество эквивалентных микроопераций можно заменить одной микрооперацией, и такую замену необходимо выполнить перед определением значения параметра N_{\max} .

2. Максимальное число входных переменных МПА L_{\max} , совпадающее с максимальным числом различных логических условий в одной ГСА.

3. Максимальное число меток M_{\max} , которыми отмечена одна ГСА. Для определения значения параметра M_{\max} договоримся использовать при отметке алгоритмы Φ_3 и Φ_4 (см. § 2.5), применение которых влечет за собой введение в ГСА наибольшего числа меток; для синтеза же авто-

мата на сформированной базовой структуре могут быть использованы глубокие другие алгоритмы, рассмотренные в предыдущих главах.

4. Максимальное число элементов памяти МПА $R_{\max} = \text{intlog}_2 M_{\max}$.

5. Максимальное число строк B_{\max} в таблице переходов МПА. Значение B_{\max} совпадает с максимальным числом различных путей перехода между метками в одной ГСА. Формулы для нахождения числа путей перехода при использовании алгоритмов Φ_3 и Φ_4 отметки ГСА приведены в § 2.5.

Численные значения параметров N_{\max} , L_{\max} , M_{\max} , R_{\max} , B_{\max} можно найти двумя способами:

а) проанализировать все реализуемые ГСА из некоторого заданного множества $\Theta = \{\Gamma^1, \dots, \Gamma^j\}$ (это можно сделать лишь при условии, что все ГСА $\Gamma^1, \dots, \Gamma^j$ уже построены). Тогда $N_{\max} = \max[N(\Gamma^1), \dots, N(\Gamma^j)]$, $N(\Gamma^j)$ — число различных микроопераций ГСА Γ^j , среди которых нет тождественно равных; $L_{\max} = \max[L(\Gamma^1), \dots, L(\Gamma^j)]$, $L(\Gamma^j)$ — число различных логических условий ГСА Γ^j ; $M_{\max} = \max[M(\Gamma^1), \dots, M(\Gamma^j)]$, $M(\Gamma^j)$ — число меток, которыми отмечена ГСА Γ^j по алгоритму Φ_3 или Φ_4 ; $R_{\max} = \text{intlog}_2 M_{\max}$; $B_{\max} = \max[B(\Gamma^1), \dots, B(\Gamma^j)]$, $B(\Gamma^j)$ — число различных путей перехода между метками ГСА Γ^j ;

б) сформировать значения указанных параметров исходя из опыта предшествующих разработок, экспертных оценок и прогнозов на будущие разработки. В этом случае уточняется класс МПА, подлежащих разработке, и анализируется определенное множество МПА (или ГСА), реализованных ранее в соответствующей аппаратуре.

В основу построения базовых ТЭЗ могут быть положены одно- и многоуровневые структуры автоматов, рассмотренные в гл. 6, 8. Найденные значения параметров N_{\max} , L_{\max} , M_{\max} , R_{\max} , B_{\max} используются для определения состава и числа элементов базовой схемы МПА, числа контактов разъема платы базового ТЭЗ и т. д. Методы построения базовой схемы и проектирования схем МПА по конкретным ГСА будут рассмотрены в последующих параграфах настоящей главы.

9.2. Одноуровневые базовые схемы

В основу предлагаемой базовой схемы МПА Мура с одноуровневой структурой (рис. 9.1) положена схема, изображенная на рис. 6.2, методы синтеза которой рассмотрены в гл. 6. Комбинационная часть МПА построена из

ПЛМ (s, t, q) и ПЗУ (s', t'). Число ПЛМ (s, t, q) равно U_6 , число ПЗУ (s', t') равно Q_6 . В качестве памяти автомата использован R_6 -разрядный регистр RG, построенный, как и ранее, на D -триггерах. Все функции возбуждения D_1, \dots, D_{R_6} элементов памяти делятся на две группы: D_1, \dots, D_r и D_{r+1}, \dots, D_{R_6} . Первые r функций возбуждения соответствуют разрядам кодов состояний автомата из множества $G = G^1 = \dots = G^U$ (см. § 6.4); их значения форми-

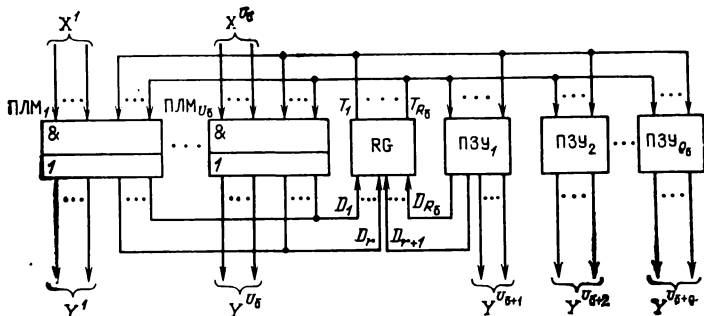


Рис. 9.1

руются на выходах ПЛМ₁, ..., ПЛМ_U, причем одноименные выходы D_1, \dots, D_r всех ПЛМ объединены по схеме «монтажное ИЛИ». Оставшиеся $R_6 - r$ функций возбуждения соответствуют разрядам кодов состояний автомата из множества $H = H^1 = \dots = H^U$; их значения формируются на выходах ПЗУ (значения всех функций D_{r+1}, \dots, D_{R_6} , как правило, могут быть сформированы на выходах единственного ПЗУ₁). Указанное разбиение может быть выполнено после того, как внутренние состояния в автомате будут закодированы с помощью метода, изложенного в § 6.4. В этом случае значения функций возбуждения D_1, \dots, D_r определяются внутренними состояниями МПА и значениями входных переменных из множества X , а значения функций возбуждения D_{r+1}, \dots, D_{R_6} — только внутренними состояниями МПА. Функции D_{r+1}, \dots, D_{R_6} представлены в совершенной ДНФ или близкой к ней форме, в силу чего они могут быть эффективно реализованы ПЗУ.

Пусть на основании анализа совокупности ГСА получены численные значения параметров $N_{\max}, L_{\max}, M_{\max}, R_{\max}, B_{\max}$ и заданы конкретные ПЛМ (s, t, q), ПЗУ (s', t') и регистры памяти, из которых должна быть построена схема МПА. Для определения количества элементов ба-

зовой схемы, показанной на рис 9.1, выполняется следующая процедура:

1. Определяется число R_6 элементов памяти автомата, которое должно быть равно R_{\max} . В ряде случаев для упрощения решения задач кодирования внутренних состояний автомата целесообразно принять $R_6 = R_{\max} + 1$.

2. Из выражения $U_6 = \max(\lfloor L_{\max}/(s - R_6) \rfloor, \lfloor B_{\max}/q \rfloor)$, полученного на основе формулы (6.5) (см. § 6.4), определяется число U_6 ПЛМ (s, t, q) в базовой схеме.

3. Значение r выбирается эмпирически. Например, если использовать только алгоритмы Φ_3 и Φ_4 отметки ГСА, рассмотренные в § 2.5, то можно принять $r = 1$. Однако в этом случае быстродействие синтезированного автомата будет очень низким. Для построения автоматов с наибольшим быстродействием необходимо, чтобы выполнялось условие $r \geq \text{int} \log_2 \sigma$ (σ — максимальная мощность множества состояний, записанных в столбце a_s одного подмассива переходов структурной таблицы синтезируемого автомата). На практике следует выбирать $r = 2, 3$ или 4. Уточнить значение этого параметра можно на основании изучения реализуемых ГСА.

Анализ более 150 различных ГСА показал, что после их отметки по алгоритму Φ_1 или Φ_2 (см. § 2.4) параметр r принимает следующие значения: $r_{\text{ср}} = 2,6$; $r_{\max} = 5,0$.

4. Из выражения $Q_6 = \lfloor [N_{\max} + (R_6 - r) - (t - r)U_6] / t' \rfloor$ определяется число Q_6 ПЗУ (s', t') в базовой схеме; при этом предполагается, что значения каждой выходной переменной будут формироваться лишь на одном выходе ПЛМ (s, t, q) или ПЗУ (s', t') и что $s' \geq R_6$ (последнее на практике почти всегда можно выполнить)¹.

Пусть полностью сформирована базовая схема, т. е. определены численные значения всех ее параметров: U_6 , Q_6 , R_6 и r . При синтезе автомата S по конкретной ГСА Γ необходимо «вложить» его схему в заданную базовую структуру и при этом минимизировать число $U + Q$ используемых ПЛМ (s, t, q) и ПЗУ (s', t') , выполнив условия $U \leq U_6$ и $Q \leq Q_6$. Если $(U + Q) < (U_6 + Q_6)$, то на плате базового ТЭЗ устанавливаются лишь необходимые элементы, число которых равно $U + Q$, а некоторые посадочные места ТЭЗ остаются незаполненными. Для синтеза МПА выполняются следующие этапы:

¹ В приведенном выражении $\lfloor C \rfloor = 1$ при $C \leq 1$ и $\lfloor C \rfloor$ — ближайшее целое число, не меньшее C , при $C > 1$.

а) с помощью метода, аналогичного рассмотренному в § 6.4, по заданной ГСА Γ строятся подтаблицы переходов W_0, W_1, \dots, W_U , число $U+1$ которых минимально возможно, и для каждой W_u выполняются условия $R \leq R_6$, $(L_u + R_6) \leq s$, $B_u \leq q$, $u = \overline{1, U}$;

б) согласно методу, аналогичному описанному в § 6.4, осуществляется кодирование внутренних состояний МПА; при этом для всех $u = \overline{1, U}$ обеспечивается выполнение условия $R_u \leq r$;

в) с помощью метода, рассмотренного в § 6.4, производится распределение выходных переменных из множества Y между выходами ПЛМ (s, t, q) и ПЗУ (s', t') базовой схемы.

Несмотря на то, что способы решения задач, перечисленных в этапах а)—в), в значительной степени основаны на методах, рассмотренных в § 6.4, они имеют и ряд отличий, на которых следует остановиться более подробно.

Предположим, что по некоторой заданной ГСА Γ необходимо синтезировать схему МПА S на сформированной базовой структуре. Предварительно проверяется справедливость неравенств $N \leq N_{\max}$, $L \leq L_{\max}$. При их нарушении автомат S нереализуем на заданной базовой структуре. Если эти условия выполняются, то с помощью метода, описанного в § 6.4, строится $U+1$ подтаблица переходов, для каждой u -й из которых W_u выполняются условия $(L_u + R_6) \leq s$, $B_u \leq q$ ($u = \overline{1, U}$). При $U \leq U_6$, $R \leq R_6$ осуществляется переход к этапу б) синтеза, в противном случае ($U > U_6$ или $R > R_6$)* выполняются следующие действия.

Граф-схема алгоритма Γ отмечается по алгоритму Φ_4 , и проверяется справедливость неравенств $R \leq R_6$, $B \leq B_{\max}$. При их невыполнении автомат S нереализуем на заданной базовой структуре. Если указанные неравенства справедливы, то переменные x_1, \dots, x_L ГСА Γ компонуется в блоки X^1, \dots, X^U , число U которых минимально возможно, и для каждого X^u ($u = \overline{1, U}$) выполняются условия $L_u = |X^u| \leq (s - R_6)$, $B_u \leq q$, где B_u — число путей перехода в ГСА Γ , проходящих через условные вершины с переменными из множества X^u (в этом случае при $L < L_{\max}$ будем допускать, чтобы $\pi_x = \{X^1, \dots, X^U\}$ было не разбиением, а покрытием множества X). Формирование подмножеств X^1, \dots, X^U необходимо производить таким образом, чтобы удалить как можно больше меток, введенных в результа-

* Если $R > R_6$, то автомат нереализуем на заданной базовой структуре.

те выполнения алгоритма Φ_4 , причем разрешается удалять только те метки, которыми отмечены входы условных вершин ГСА. Пусть меткой a_s отмечен вход условной вершины U_k , содержащей логическое условие $x_j \in X^u$. Тогда эта метка может быть удалена лишь в том случае, если ни одна дуга, ведущая ко входу U_k , не выходит из условной вершины U_g с логическим условием $x_i \notin X^u$.

Изложим простой последовательный приближенный алгоритм решения рассмотренной задачи компоновки. Обозначим через ρ_l число повторений в различных условных вершинах ГСА Γ логического условия x_l , $l = \overline{1, L}$. Упорядочим элементы множества $\rho = \{\rho_1, \dots, \rho_L\}$ в порядке убывания их значений слева направо (в этом случае значение любого левого элемента в множестве ρ будет не меньше, чем правого) и пусть $\bar{X} = X = \{x_1, \dots, x_L\}$. Процесс формирования подмножеств X^1, \dots, X^u делится на U этапов. На первом шаге каждого u -го этапа из множества ρ выбирается крайний левый элемент ρ_l , значение которого максимально; логическое условие x_l включается в подмножество X^u и удаляется из множества \bar{X} . Элементам множества \bar{X} назначаются веса $v_1(X^u), \dots, v_L(X^u)$. Вес $v_i(X^u)$ логического условия x_i указывает на то, сколько меток может быть потенциально удалено в ГСА, если переменную x_i включить в подмножество X^u . На каждом последующем шаге u -го этапа из множества \bar{X} выбирается элемент x_i , имеющий максимальный вес $v_i(X^u)$.

Обозначим через $B(X^u \cup \{x_i\})$ число путей перехода в ГСА, идущих через условные вершины с логическими условиями из множества $X^u \cup \{x_i\}$. Тогда если $B(X^u \cup \{x_i\}) \leq q$, $|X^u \cup \{x_i\}| \leq (s - R_0)$, то элемент x_i включается в подмножество X^u и удаляется из множества \bar{X} . Если $B(X^u \cup \{x_i\}) > q$ или $|X^u \cup \{x_i\}| > (s - R_0)$, то выбирается новый элемент x_j , который имеет максимальное значение веса $v_j(X^u)$ среди всех элементов множества $\bar{X} \setminus \{x_i\}$. Когда на некотором шаге несколько элементов множества \bar{X} имеют одинаковые веса, среди них выбирается тот x_h , для которого значение ρ_h максимально. Если последнее условие также равнозначно для нескольких элементов множества \bar{X} , выбирается любой из них. После того как очередное логическое условие x_i включается в подмножество X^u , в ГСА Γ со входа любой отмеченной условной вершины U_f , содержащей логическое условие $x_p \in X^u$, удаляется метка, если она может быть удалена, и $C(U_f) \leq 0$, где $C(U_f)$ — цена условной вершины U_f ГСА (см. § 2.6).

На очередном шаге алгоритма для всех $x_i \in \bar{X}$ могут

оказаться истинными условия $B(X^u \cup \{x_i\}) > q$, $|X^u \cup \{x_i\}| \leq (s - R_6)$. В этом случае переменная x_i заменяется в некоторых условных вершинах ГСА переменной x_c так, чтобы выполнялось условие $B(X^u \cup \{x_c\}) = q$ и цена $v_i(X^u)$ сократилась бы максимально возможно. В дальнейшем входные полюса схемы, изображенной на рис. 9.1, соответствующие переменным x_i и x_c , объединяются и на них подаются значения переменной x_i . При этом множество $\pi_x = \{X^1, \dots, X^v\}$ будет уже не разбиением, а покрытием множества X . Формирование каждого очередного подмножества X^u заканчивается, когда $|X^u| = s - R_6$ либо $|X^u| < (s - R_6)$, но в множестве X^u не может быть включен ни один элемент из множества X без нарушения неравенства $B_u \leq q$. Выполнение алгоритма заканчивается, когда $X = \emptyset$, т. е. сформированы все подмножества X^1, \dots, X^v .

Кроме рассмотренного последовательного алгоритма может быть предложен аналогичный параллельный алгоритм, когда на первом шаге из множества ρ выбираются U крайних левых элементов $(U =]L / (s - R_6) [)$, а из множества X — соответствующие им опорные элементы. На каждом очередном шаге во все множества X^1, \dots, X^v вводится по одному новому элементу, выбор которых осуществляется по тем же критериям, что и последовательного алгоритма. В результате выполнения последовательного или параллельного алгоритма число меток в ГСА, как правило, увеличивается по сравнению с тем числом, которое получается в результате реализации метода, изложенного в § 6.4.

Качество решения задачи компоновки с помощью рассмотренных алгоритмов определяется тем, как часто одни и те же логические условия встречаются в различных условных вершинах ГСА. Обозначим $\rho_{cp} = \left(\sum_{l=1}^L \rho_l \right) / L$. Тогда

качество решения задачи будет тем выше, чем меньше значение ρ_{cp} . Для ГСА, встречающихся на практике, значение ρ_{cp} обычно небольшое. Анализ более чем 150 граф-схем показал, что для них значение ρ_{cp} колеблется в пределах 1,0—3,5.

Пусть выполнен алгоритм компоновки и сформированы все подмножества X^1, \dots, X^v . Тогда если $U < U_6$, то как и в § 6.4, строятся подтаблицы W_0, W_1, \dots, W_U и осуществляется переход к этапу б); если $U > U_6$, то автомат нереализуем на заданной базовой структуре. Это последнее условие выполняется чрезвычайно редко для встречающихся

ся на практике граф-схем, у которых $N \leq N_{\max}$, $L \leq L_{\max}$, $R \leq R_6$, $B \leq B_{\max}$.

Последующее изложение будем иллюстрировать примером синтеза автомата Мура по ГСА, изображенной на рис. 9.2. Согласно параметрам базовой схемы $U_6=3$, $Q_6=2$, $R_6=4$, $r=2$, $s=7$, $t=6$, $q=12$, $s'=4$, $t'=6$, используемой в примере, в МПА может быть не более трех ПЛМ и двух ПЗУ.

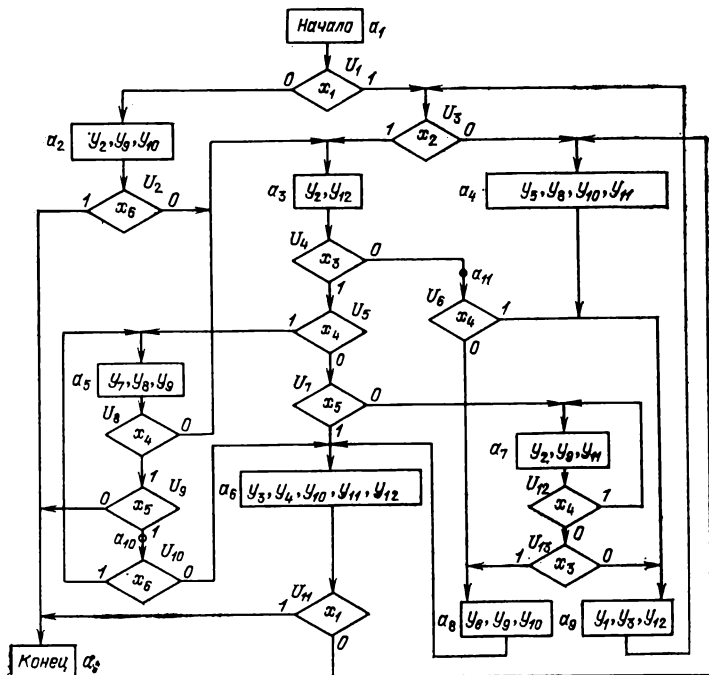


Рис. 9.2

С помощью метода, изложенного в § 6.4, ГСА на рис. 9.2 отмечается основными a_1, \dots, a_8 и дополнительной a_{10} метками. Поскольку в базовой схеме только r функций возбуждения снимается с выходов ПЛМ, в МПА из каждого состояния a_m не должно быть переходов более чем в 2^r состояний:

$$|A(a_m)| \leq 2^r, \quad (9.1)$$

где $A(a_m)$ — множество состояний, записанных в столбце a_s одного подмассива переходов из a_m . Более того, на всех переходах в каждой подтаблице W_u только r и притом

одних и тех же разрядов кодов состояний могут изменять свои значения на переходах из одного состояния, (R_6-r) разрядов, соответствующих выходам с ПЗУ, остаются неизменными.

Допустим, что для некоторых состояний a_m условие (9.1) не выполняется (в ГСА на рис. 9.2 $|A(a_3)| = 5 > 2^2$). В этом случае будем ограничивать мощность всех множеств $A(a_m)$ значением 2^r за счет дополнительной отметки заданной ГСА Γ . Построим граф Γ_m (рис. 9.3), отображающий фрагменты ГСА Γ , по которым получены подмассивы переходов с нарушением условия (9.1). Обозначим через \bar{A} множество состояний (и соответствующих меток в ГСА Γ), включенных в те множества $A(a_m)$, для которых $|A(a_m)| > 2^r$. Множество условных вершин, через которые проходят пути перехода в метки из множества \bar{A} , обозначим через \bar{O} . Элементы множеств \bar{A} и \bar{O} образуют вершины графа Γ_m . В графе Γ_m проведем дуги:

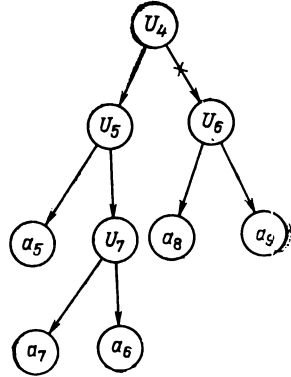


Рис. 9.3

- а) из вершины $U_i \in \bar{O}$ в вершину $U_j \in \bar{O}$, если в ГСА Γ есть дуга из U_i в U_j ;
- б) из вершины $U_i \in \bar{O}$ в вершину $a_s \in \bar{A}$, если в ГСА Γ есть дуга из U_i в метку a_s . Заметим, что в Γ_m отсутствуют дуги из вершин $a_s \in \bar{A}$ в вершины $U_i \in \bar{O}$.

Разрежем граф Γ_m на такие подграфы, что в каждом из них сумма числа вершин $a_s \in \bar{A}$ и разрезанных выходящих дуг не превышает значения 2^r . Формирование подграфов будем производить путем разрыва дуг Γ_m , число которых минимально возможно. Если некоторая дуга графа Γ_m , направленная в вершину U_i , разрывается, то вход условной вершины U_i ГСА Γ отмечается меткой. Граф Γ_m на рис. 9.3 разрезается на два подграфа путем удаления дуги, помеченной крестиком (существуют и другие варианты разрезания графа Γ_m). В результате в ГСА (рис. 9.2) вводится дополнительная метка a_{11} . После дополнительной отметки ГСА подтаблицы переходов W_1, \dots, W_U МПА перестраиваются таким образом, что для всех $a_m \in \bar{A}$ выполняется условие $|A(a_m)| \leq 2^r$. Таблица 9.1 [столбцы $a_m Y(a_m)$, a_s , $X(a_m, a_s)$] представляет построенные подтаблицы переходов.

Таблица 9.1

$a_m Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$F(a_m, a_s)$
a_4 y_6, y_8, y_{10}, y_{11}	1100	a_9	0011	1	$(D_3 D_4)$
a_8 y_6, y_9, y_{10}	0111	a_6	0001	1	(D_4)
a_1	00—0	a_2 a_3 a_4	0100 10—0 1100	$\overline{x_1}$ $x_1 x_2$ $x_1 \overline{x_2}$	D_2 D_1 $D_1 D_2$
a_2 y_2, y_9, y_{10}	0100	a_1 a_3	00—0 10—0	x_6 $\overline{x_6}$	— D_1
a_6 $y_3, y_4, y_{10},$ y_{11}, y_{12}	0001	a_1 a_4	00—0 1100	x_1 $\overline{x_1}$	— $D_1 D_2$
a_9 y_1, y_3, y_{12}	0011	a_3 a_4	10—0 1100	x_2 $\overline{x_2}$	D_1 $D_1 D_2$
a_{10}	0110	a_5 a_6	0101 0001	x_6 $\overline{x_6}$	$D_2 (D_4)$ (D_4)
a_3 y_2, y_{12}	10—0	a_5 a_6 a_7 a_{11}	0101 0001 10—1 1101	$x_3 x_4$ $\overline{x_3 x_4 x_5}$ $x_3 x_4 x_5$ $\overline{x_3}$	$D_2 (D_4)$ (D_4) $D_1 (D_4)$ $D_1 D_2 (D_4)$
a_5 y_7, y_8, y_9	0101	a_1 a_3 a_{10}	00—0 10—0 0110	$x_4 x_5$ $\overline{x_4}$ $x_4 x_5$	(D_3) $D_1 (D_3)$ $D_2 (D_3)$
a_7 y_2, y_9, y_{11}	10—1	a_7 a_8 a_9	10—1 0111 0011	x_4 $\overline{x_3 x_4}$ $\overline{x_3 x_4}$	$D_1 (D_3 D_4)$ $D_2 (D_3 D_4)$ $(D_3 D_4)$
a_{11}	1101	a_8 a_9	0111 0011	$\overline{x_4}$ x_4	$D_2 (D_3 D_4)$ $(D_3 D_4)$

При кодировании состояний МПА, как и выше в гл. 6, необходимо установить соответствие между столбцами карты Карно и различными множествами $A(a_m)$. При этом строки карты Карно должны быть помечены частью кода состояний, содержащей r разрядов (изменяющейся на переходах из a_m), а столбцы — $(R_6 - r)$ разрядов (неизменной на переходах из a_m).

Для решения задачи кодирования на множестве A внутренних состояний МПА введем отношение ξ , определяемое следующим образом:

$$a_m \xi a_s \leftrightarrow A(a_m) \cap A(a_s) \neq \emptyset.$$

Построим граф Γ'_ξ , M вершина которого поставим в соответствие внутренние состояния a_1, \dots, a_M МПА из множества A ; две вершины a_m и a_s соединим ребром, если и только если выполняется отношение $a_m \xi a_s$. Каждой вершине a_m графа Γ'_ξ дополнительно припишем элементы множества $A(a_m)$ и удалим те вершины, для которых $|A(a_m)| < 2$. Если вершины a_m и a_s соединены ребром, то припишем этому ребру элементы множества $A(a_m) \cap A(a_s)$ и вес $\rho(a_m, a_s)$, вычисляемый по формуле

$$\rho(a_m, a_s) = \sum_{a_k \in A(a_m) \cap A(a_s)} |A(a_k)|.$$

Граф Γ'_ξ для нашего примера приведен на рис. 9.4,а. Построим граф Γ_ξ , который получается из Γ'_ξ объединением его вершин. Вершины a_m, a_s, \dots, a_h графа Γ'_ξ могут быть объединены в одну тогда и только тогда, когда $|A(a_m) \cup A(a_s) \cup \dots \cup A(a_h)| \leq 2^r$. Если отмеченные вершины графа Γ'_ξ объединяются в одну вершину графа Γ_ξ , то ей приписываются элементы множества $A(a_m) \cup A(a_s) \cup \dots \cup A(a_h)$. Множества состояний, приписанных ребрам графа Γ_ξ , и веса ребер определяются так же, как и для графа Γ'_ξ . Граф Γ_ξ должен быть сформирован таким образом, чтобы число его вершин было не более чем $2^{R_6 - r}$ (желательно, чтобы это число было как можно меньшим); если это невозможно, то синтезируемый автомат не может быть реализован заданной базовой схемой (указанная ситуация на практике возникает исключительно редко). Граф Γ_ξ в примере изображен в верхней части рис. 9.4,б.

Построим еще один граф Γ_μ отношения соседства кодов, приписанных столбцам карты Карно. Вершинам графа Γ_μ , число которых равно $2^{R_6 - r}$, соответствуют различ-

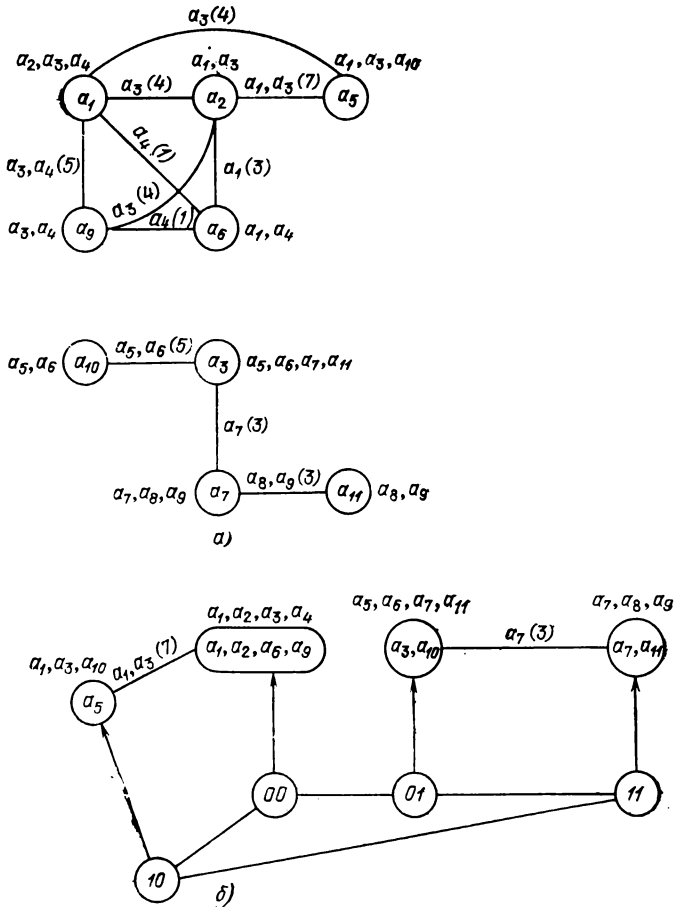


Рис. 9.4

ные (R_6-r) -разрядные двоичные коды. Две вершины Γ_μ соединяются ребром, если и только если соответствующие им двоичные коды соседние. Между вершинами графов Γ_ξ и Γ_μ необходимо установить такое соответствие, при котором суммарный вес ребер графа Γ_ξ , покрытых ребрами графа Γ_μ , максимален. Полученное соответствие фактически устанавливает аналогичное соответствие между столбцами карты Карно и состояниями, размещаемыми в этих столбцах. Действительно, если вершина e_i графа Γ_μ соответствует вершине графа Γ_ξ , которой при-

писаны состояния из множества Δ , то эти состояния должны быть размещены в столбце e_i карты Карно, и это всегда можно сделать, поскольку $|\Delta| \leq 2^r$.

При записи состояний МПА в столбцы карты Карно должны быть выполнены следующие требования. Если некоторая пара вершин графа Γ_ξ соединена ребром, которому приписаны состояния из множества Δ , и это ребро покрыто ребром графа Γ_μ , инцидентным вершинам e_i и e_j , то любое состояние $a_m \in \Delta$ записывается в одной строке столбцов e_i и e_j (ясно, что любое состояние $a_m \in \Delta$ дублируется в карте Карно). Состояния a_k , наиболее часто встречающиеся в столбце a_s подтаблиц переходов W_0, W_1, \dots, W_U , следует по возможности размещать в той строке карты Карно, которая закодирована кодом, содержащим все нули. Выполнение указанных требований обеспечивает минимизацию числа использованных про-

		$T_3 T_4$			
		00	01	11	10
$T_1 T_2$	00	a_1	a_5	a_9	a_4
	01	a_2	a_5	a_8	a_{10}
	11	a_4	a_{11}		,
	10	a_3	a_7	a_7	a_3

Рис. 9.5

межуточных шин в ПЛМ базовой схемы. Граф Γ_μ для рассматриваемого примера также приведен на рис. 9.4,б. Установленное соответствие между вершинами графов Γ_ξ и Γ_μ указано стрелками на этом рисунке. Карта Карно для нашего примера показана на рис. 9.5; в ней состояния a_1, a_3, a_7 продублированы, а коды $K(a_1), K(a_3), K(a_7)$ этих состояний представлены троичными векторами (см. § 6.4).

Таблица 9.2

T_1	T_2	T_3	T_4	x_1	x_2	x_6	D_1	D_2	y_1	y_3	y_4	y_{12}
0	0	—	0	0	—	—	.	1
0	0	—	0	1	1	—	1
0	0	—	0	1	0	—	1	1
0	1	0	0	—	—	0	1
0	0	0	1	0	—	—	1	1
0	0	1	1	—	1	—	1	.	1	1	.	1
0	0	1	1	—	0	—	1	1	1	1	.	1
0	1	1	0	—	—	1	.	1
0	0	0	1	—	—	—	.	.	.	1	1	1
1	0	—	0	—	—	—	1

После кодирования состояний формирование структурных подтаблиц W_1, \dots, W_U выполняется в полном соответствии с методом, рассмотренным в § 6.4. Если некоторые состояния из множества $A(a_m)$, для которых $|A(a_m)|=1$, отсутствуют в карте Карно, то их следует разместить в ней произвольным образом. Построенные структурные подтаблицы W_0, W_1 и W_2 приведены в табл. 9.1. Матрицы E^1 и F^1 для нашего примера отсутствуют, поскольку в столбце $F(a_m, a_s)$ подтаблицы W_0 для всех строк $D_1 = D_2 = 0$. Обозначим через $E^{\text{ПЗУ}}$ и $F^{\text{ПЗУ}}$ матрицы, которые описывают функции D_3 и D_4 , реализуемые ПЗУ. Тогда:

$$E^{\text{ПЗУ}} = \begin{pmatrix} T_1 & T_2 & T_3 & T_4 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & - & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & - & 1 \\ 1 & 1 & 0 & 1 \end{pmatrix}; \quad F^{\text{ПЗУ}} = \begin{pmatrix} D_3 & D_4 \\ 1 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

В табл. 9.2 и 9.3 представлена информация для программирования ПЛМ₁ и ПЛМ₂ схемы МПА, в табл. 9.4 — для программирования ПЗУ₁. Распределение выходных переменных между выходами ПЛМ и ПЗУ произведено с помощью метода, изложенного в § 6.4. По результатам син-

Таблица 9.3

T_1	T_2	T_3	T_4	x_3	x_4	x_5	D_1	D_2	y_6	y_7	y_8	y_{11}
1	0	—	0	1	1	—	.	1
1	0	—	0	1	0	0	1
1	0	—	0	0	—	—	1	1
0	1	0	1	—	0	—	1
0	1	0	1	—	1	1	.	1
1	0	—	1	—	1	—	1
1	0	—	1	1	0	—	.	1
1	1	0	1	—	0	—	.	1
1	1	0	0	—	—	—	.	.	1	.	1	1
0	0	0	1	—	—	—	1
0	1	0	1	—	—	—	.	.	1	1	.	.
1	0	—	1	—	—	—	1

Таблица 9.4

Адрес	D_3	D_4	y_2	y_6	y_9	y_{10}
1	1
4	.	.	1	.	1	1
5	1	.	.	.	1	.
6	.	1
7	.	1	.	1	1	1
8	.	1	1	.	.	.
9	1	1	1	.	1	.
10	.	1	1	.	.	.
11	1	1	1	.	1	.
12	1	1	.	.	.	1
13	1	1

теза в заданной базовой схеме МПА используются две ПЛМ и одно ПЗУ (одно посадочное место под ПЛМ и одно под ПЗУ остаются незаполненными). Полученная логическая схема изображена на рис. 9.6; неиспользуемые ПЛМ и ПЗУ, которые на базовый ТЭЗ не устанавливаются, обведены штриховой линией.

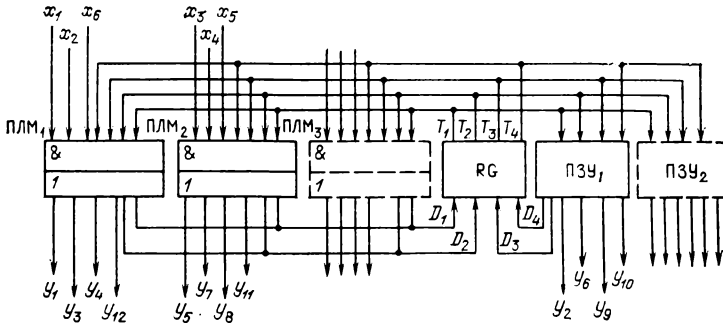


Рис. 9.6

В заключение этого параграфа заметим, что, если некоторая ГСА Γ не может быть реализована заданной базовой схемой, используются известные методы [42] ее декомпозиции на такие компонентные ГСА, число которых минимально возможно, и каждая из них в отдельности может быть реализована заданной базовой схемой. Исходная ГСА Γ реализуется сетью из базовых схем, правила соединения которых определяются на этапе декомпозиции.

9.3. Многоуровневые базовые схемы

В основу многоуровневых базовых схем могут быть положены схемы автоматов, рассмотренные в гл. 6 (§ 6.5). На рис. 9.7 изображена одна из них — базовая схема МПА Мили с РF-структурой. Комбинационная часть МПА состоит из Р-подсхемы (ПЛМ₁, ..., ПЛМ_{U₆}, ПЗУ₁) и F-подсхемы (ПЗУ₂, ..., ПЗУ_{Q₆}).

В качестве элементов памяти автомата использован R_6 -разрядный регистр RG. Все функции z_1, \dots, z_{K_6} (см. § 6.5), значения которых формируются Р-подсхемой, делятся на две группы: z_1, \dots, z_r и z_{r+1}, \dots, z_{K_6} . Первые r из них соответствуют разрядам кодов строк структурных подтаблиц из множества $G = G^1 = \dots = G^{U_6}$ (см. § 6.5); их значения формируются на выходах ПЛМ₁, ..., ПЛМ_{U₆}, причем одноимен-

ные выходы z_1, \dots, z_r всех ПЛМ объединены по схеме «монтажное ИЛИ». Оставшиеся $K_6 - r$ функций z_{r+1}, \dots, z_{K_6} соответствуют разрядам кодов строк структурных подтаблиц из множества $H = H^1 = \dots = H^U$; их значения формируются на выходах ПЗУ (значения всех функций, как правило, могут быть сформированы на выходах единственного ПЗУ₁).

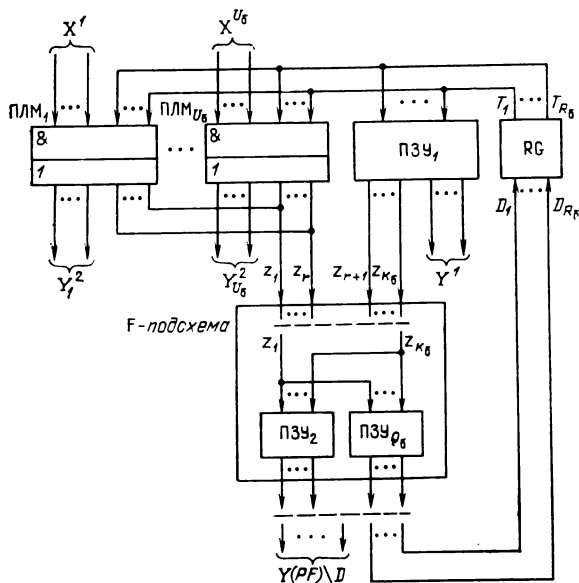


Рис. 9.7

Если для используемых ПЗУ (s', t') выполняется условие $(K_6 - r) < t'$, то на $t' - (K_6 - r)$ выходов ПЗУ₁ назначаются некоторые выходные переменные, значения которых не зависят от значений x_1, \dots, x_L . Эти переменные включаются в множество Y^1 . Когда для выбранных ПЛМ (s, t, q) выполняется условие $r < t$, на свободные выходы ПЛМ Р-подсхемы могут назначаться некоторые выходные переменные, включаемые в множества $Y^2_1, \dots, Y^2_{U_6}$. Из нераспределенных между элементами Р-подсхемы выходных переменных формируется множество $Y(PF)$ (см. § 6.5).

Пусть на основании анализа реализуемых ГСА получены численные значения параметров $N_{\max}, L_{\max}, M_{\max}, R_{\max}, B_{\max}$ и заданы конкретные ПЛМ (s, t, q), ПЗУ (s', t') и регистры памяти, из которых должна быть построена схема

МПА. Для определения числа элементов базовой схемы, изображенной на рис. 9.7, выполняются указанные ниже действия.

1. Определяется число R_6 элементов памяти автомата, которое должно быть равно R_{\max} , и $K_6 = \text{int} \log_2 B_{\max}$.

2. Из выражения $U_6 = \max [L_{\max}/(s-R_6), B_{\max}/q]$, полученного на основе (6.5) (см. § 6.4), определяется число ПЛМ (s, t, q) в базовой схеме.

3. Значение r выбирается эмпирически (см. § 9.2). На практике в этом случае следует выбирать $r=2, 3$ или 4 .

4. Из выражения

$$Q_6 = \lceil [N_{\max} + R_6 + (K_6 - r) - \frac{U_6}{k} (t - r)] / t' \rceil$$

определяется число Q_6 ПЗУ (s', t') в базовой схеме¹. Предполагается, что количество выходных переменных, значения которых не зависят от значений входных, не менее, чем $\lceil (K_6 - r) / t' \rceil \lceil t' - (K_6 - r) \rceil$, и что $s' \geq R_6$, $s' \geq K_6$. Здесь $k = U_6(t-r) / \left| \bigcup_{i=1}^{U_6} Y_2^i \right|$ — коэффициент дублирования переменных из множества Y на выходах ПЛМ Р-подсхемы, значение которого равно среднему числу выходов ПЛМ, приходящемуся на одну выходную переменную; в выражении для Q_6 следует принимать $k=1-3$.

Пусть полностью сформирована базовая схема, т. е. определены численные значения всех ее параметров: U_6 , Q_6 , R_6 , K_6 и r . При синтезе автомата S по конкретной ГСА Γ необходимо «вложить» его схему в заданную базовую структуру и при этом минимизировать число $U+Q$ используемых ПЛМ (s, t, q) и ПЗУ (s', t') ($U \leq U_6$, $Q \leq Q_6$). Если $(U+Q) < (U_6+Q_6)$, то на плате базового ТЭЗ устанавливаются лишь необходимые элементы, число которых равно $U+Q$, а некоторые посадочные места ТЭЗ остаются незаполненными. Для синтеза МПА выполняются следующие этапы:

а) с помощью методов, рассмотренных в § 6.4, 9.2, по заданной ГСА Γ строятся подтаблицы переходов W_0, W_1, \dots, W_U , число $U+1$ которых минимально возможно и для

¹ В приведенном выражении $\lceil C \rceil = 2$ при $C \leq 2$ (предполагается, что на выходах ПЗУ₁ формируются значения всех переменных z_{r+1}, \dots, z_{K_6} , а на выходах ПЗУ₂ — значения D_1, \dots, D_{R_6} и $\lceil C \rceil$ — ближайшее целое число, не меньшее C при $C > 2$).

каждой W_u выполняются условия $R \leq R_6$, $(L_u + R_6) \leq s$, $B_u \leq q$, $u = \overline{1, U}$ (если это сделать невозможно, то синтезируемый автомат нереализуем на заданной базовой структуре);

б) с помощью метода, изложенного в § 6.4, 6.5, 9.2, производится кодирование строк во всех подтаблицах переходов W_0, W_1, \dots, W_U ; при этом обеспечивается выполнение условий $K \leq K_6$, $K_u \leq r$, $u = \overline{1, U}$ (если приведенные условия нельзя выполнить, то синтезируемый автомат нереализуем на заданной базовой структуре);

в) согласно методу, рассмотренному, например, в [36], выполняется кодирование внутренних состояний автомата, целью которого является минимизация числа строк в матрицах типа G^u, H^u ($u = \overline{1, U}$); это позволяет сократить число использованных промежуточных шин в ПЛМ Р-подсхемы;

г) с помощью метода, изложенного в § 6.5, производится распределение выходных переменных и функций возбуждения из множества YUD между элементами базовой схемы¹; в результате формируются подмножества $Y^1, Y^2, \dots, Y(PF)$.

Все методы, перечисленные в пп. а)–г), подробно рассматривались ранее, причем отсутствуют какие-либо существенные отличия в их использовании для синтеза автоматов на предлагаемой базовой структуре. В силу этого в настоящем параграфе ограничимся изложением общей последовательности синтеза МПА и рассмотрением примера. Пусть задана ГСА Г (рис. 9.8) и необходимо синтезировать МПА Мили S , вложив его схему в базовую многоуровневую РФ-структуру (см. рис. 9.7) с параметрами: $U_6=3$, $Q_6=3$, $K_6=4$, $R_6=3$, $r=2$, $s=7$, $t=6$, $q=12$, $s'=4$, $t'=6$. С помощью алгоритма Φ_1 (см. § 2.4) ГСА на рис. 9.8 отмечается метками a_1, \dots, a_6 . Для сокращения числа строк в подмассиве переходов из a_5 введена дополнительная метка a_7 (см. § 9.2). Без a_7 число таких строк равно пяти, и они не могут быть закодированы различными r -разрядными двоичными кодами ($r=2$).

Далее с помощью метода, рассмотренного в § 6.4, строятся три подтаблицы переходов: W_0, W_1, W_2 (см. столбцы $a_m, a_s, X(a_m, a_s), Y(a_m, a_s)$ табл. 9.5, в которой различные подтаблицы отделены друг от друга двойными горизонталь-

¹ Несущественное отличие здесь состоит лишь в том, что функции D_1, \dots, D_{R_6} могут распределяться только на выходы F-подсхемы.

Таблица 9.5

a_{in}	$K(a_{in})$	a_s	$K(a_s)$	$X(a_m, a_s)$	$Y(a_m, a_s)$	$F(a_m, a_s)$	C_t	$K(C_t)$	$Z(a_m, a_s)$
a_3	001	a_4	011	1	y_1, y_3, y_6	D_2D_3	C_1	1110	$z_1z_2(z_3)$
a_4	011	a_5	111	1	y_3, y_7, y_{10}	$D_1D_2D_3$	C_2	1011	$z_1(z_2z_4)$
a_6	101	a_6	111	1	y_7, y_8, y_9	$D_1D_2D_3$	C_3	1111	$z_1z_2(z_3z_4)$
a_1	000	a_2	100	x_1x_2	y_9, y_{10}	D_1	C_4	0000	—
		a_3	001	x_1	y_2, y_4	D_3	C_5	0100	z_2
		a_4	011	$x_1x_2x_3$	y_9, y_{13}	D_2D_3	C_6	1000	z_1
		a_4	011	$x_1x_2x_3$	y_4, y_{10}	D_2D_3	C_7	1100	z_1z_2
a_2	100	a_1	000	x_3	y_{12}	—	C_8	0010	(z_3)
		a_4	011	x_3x_6	y_4, y_6, y_6	D_2D_3	C_9	0110	$z_2(z_3)$
		a_5	111	x_3x_5	y_9	$D_1D_2D_3$	C_{10}	1010	$z_1(z_3)$
a_7	111	a_1	000	x_4x_6	y_2, y_6, y_{11}	—	C_{11}	0001	(z_4)
		a_2	100	x_4x_6	y_{11}, y_{12}, y_{13}	D_1	C_{12}	0101	$z_2(z_4)$
		a_6	101	x_4x_7	y_2, y_6, y_{12}	D_1D_3	C_{13}	1001	$z_1(z_4)$
		a_7	010	x_4x_7	—	D_2	C_{14}	1101	$z_1z_2(z_4)$
			010	a_1	000	x_8	y_6, y_8, y_{13}	—	C_{16}
		a_1	000	x_8	—	—	C_{10}	0111	$z_2(z_3z_4)$

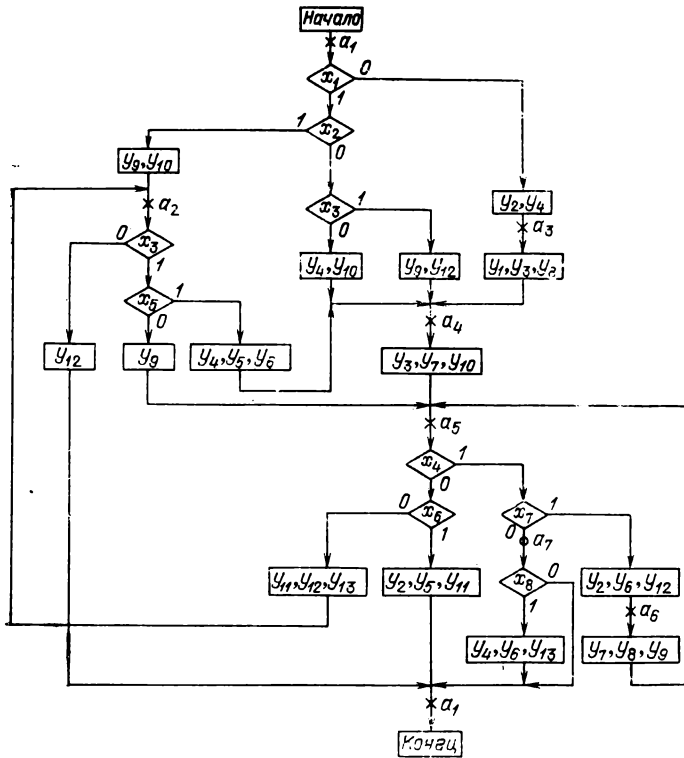


Рис. 9.8

ными линиями). Результаты кодирования строк табл. 9.5, помеченных в столбце C_i символами C_1, \dots, C_{16} , отображены в столбце $K(C_i)$, а кодирования состояний—в столбцах $K(a_m)$ и $K(a_s)$. После распределения выходных переменных формируются следующие множества: $Y^1 = \{y_1, y_3, y_7, y_8\}$; $Y^2_1 = \{y_9, y_{10}, y_4, y_{12}\}$; $Y^2_2 = \{y_4, y_{12}, y_{11}, y_{13}\}$, $k=1, 3$ (т. е. на одну выходную переменную в среднем приходится 1, 3 выхода ПЛМ); $Y(PF) = \{y_2, y_5, y_6, D_1, D_2, D_3\}$. Матрицы E^1, F^1, E^2, F^2 и $E^{ПЗУ}, F^{ПЗУ}$ представляются в следующем виде:

$$E^1 = \begin{vmatrix} T_1 & T_2 & T_3 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}; \quad F^1 = \begin{vmatrix} z_1 & z_2 \\ 1 & 0 \\ 1 & 1 \end{vmatrix};$$

$$E^2 = \begin{vmatrix} T_1 & T_2 & T_3 \\ 0 & 0 & 1 \end{vmatrix}; \quad F^2 = \begin{vmatrix} z_1 & z_2 \\ 1 & 1 \end{vmatrix};$$

$$E^{\text{ПЗУ}} = \begin{vmatrix} T_1 & T_2 & T_3 \\ 1 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix}; \quad F^{\text{ПЗУ}} = \begin{vmatrix} z_3 & z_4 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \end{vmatrix}.$$

Здесь $E^{\text{ПЗУ}}$, $F^{\text{ПЗУ}}$ — матрицы, которые описывают функции z_3 и z_4 , реализуемые ПЗУ₁.

Таблицы 9.6 и 9.7 представляют информацию для программирования ПЛМ₁ и ПЛМ₂ Р-подсхемы МПА, табл. 9.8 и 9.9 — для программирования ПЗУ₁ Р-подсхемы и ПЗУ₂ F-подсхемы. По результатам синтеза в заданной базовой

Таблица 9.6

T_1	T_2	T_3	x_1	x_2	x_3	x_5	z_1	z_2	y_9	y_{10}	y_4	y_{12}
0	0	0	1	1	—	—	.	.	1	1	.	.
0	0	0	0	—	—	—	.	1	.	.	1	.
0	0	0	1	0	1	—	1	.	1	.	.	1
0	0	0	1	0	0	—	1	1	.	1	1	.
1	0	0	—	—	0	—	1
1	0	0	—	—	1	1	.	1	.	.	1	.
1	0	0	—	—	1	0	1	.	1	.	.	.
0	1	1	—	—	—	—	1	.	.	1	.	.
1	0	1	—	—	—	—	1	1	1	.	.	.

Таблица 9.7

T_1	T_2	T_3	x_4	x_6	x_7	x_8	z_1	z_2	y_4	y_{12}	y_{11}	y_{13}
1	1	1	0	1	—	—	1	.
1	1	1	0	0	—	—	.	1	.	1	1	1
1	1	1	1	—	1	—	1	.	.	1	.	.
1	1	1	1	—	0	—	1	1
0	1	0	—	—	—	1	.	.	1	.	.	1
0	1	0	—	—	—	0	.	1
0	0	1	—	—	—	—	1	1

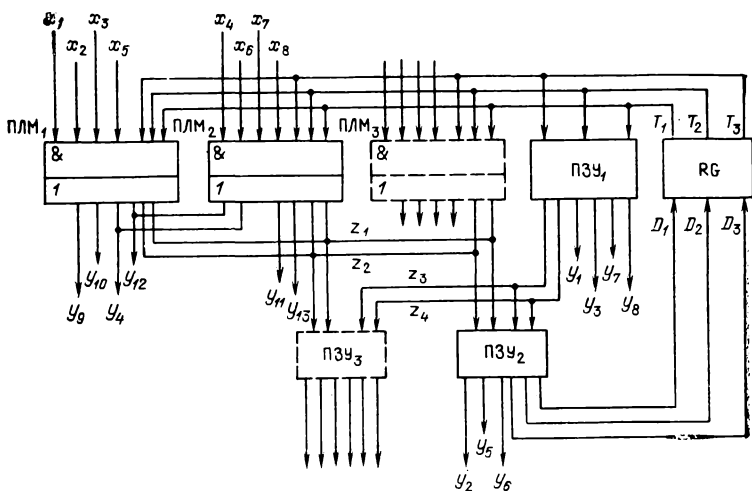


Рис. 9.9

Таблица 9.8

Адрес	z_3	z_4	y_1	y_3	y_7	y_8
1	1	.	1	1	.	1
2	1	1
3	1	1	.	1	1	.
4	1
5	1	1	.	.	1	1
7	.	1

Таблица 9.9

Адрес	D_1	D_2	D_3	y_2	y_6	y_8
0	1
1	.	.	1	1	.	.
3	1
4	.	.	1	1	.	.
5	1
6	.	1	1	.	1	1
8	.	1	1	.	.	.
9	1	.	1	1	.	1
10	1	1	1	.	.	.
11	1	1	1	.	.	.
12	.	1	1	.	.	.
13	.	1
14	.	1	1	.	.	.
15	1	1	1	.	.	.

схеме МПА используются две ПЛМ и два ПЗУ (одно посадочное место под ПЛМ и одно под ПЗУ остаются незаполненными). Полученная логическая схема изображена на рис. 9.9; неиспользуемые ПЛМ и ПЗУ, которые на базовый ТЭЗ не устанавливаются, обведены штриховой линией.

По аналогии с рассмотренной выше могут быть построены многоуровневые базовые схемы с РУ-, РМ- и РД-

структурами. Методы синтеза, необходимые для «вложения» схемы автомата в соответствующую многоуровневую базовую структуру, основаны на методах, изложенных в гл. 6 и 8.

9.4. Базовые схемы с перестраиваемой структурой

Процедура построения МПА, рассмотренная в § 9.2, 9.3, предполагает использование специальных структурных решений, в рамках которых выполняется оптимизация, причем в каждой базовой схеме реализовано лишь одно из таких решений. Например, формируется схема с одноуровневой (§ 9.2) либо многоуровневой (§ 9.3) структурой. В то же время каждое структурное решение будет приводить к наилучшим результатам синтеза, оцениваемым числом используемых элементов, для вполне определенного класса задач (реализуемых ГСА) [31]. В [43] предложен подход, связанный с возможностью построения быстродействующих

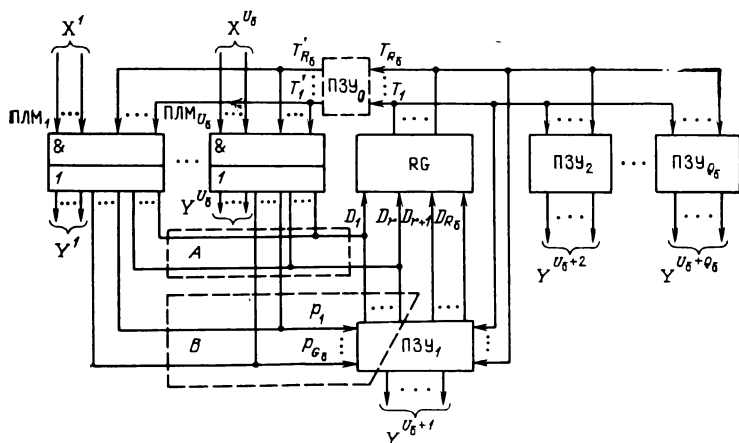


Рис. 9.10

базовых схем, допускающих настройку на различные структурные решения. Тем самым увеличивается размерность и расширяется класс решаемых задач (реализуемых ГСА). Использование этого подхода иллюстрируется ниже на примере организации одной из перестраиваемых базовых структур и рассмотрения метода реализации логических схем МПА на ее основе.

Предлагаемая базовая схема изображена на рис. 9.10. На ее основе может быть реализован МПА, у которого $X = \{x_1, \dots, x_L\}$ — множество входных переменных; $X \subseteq X^1 \cup \dots \cup X^{U_6}$, $Y = \{y_1, \dots, y_N\}$ — множество выходных переменных; $Y \subseteq Y^1 \cup \dots \cup Y^{U_6} \cup Y^{U_6+1} \cup \dots \cup Y^{U_6+Q_6}$, $D = \{D_1, \dots, D_R\}$ — множество функций возбуждения, $D \subseteq \{D_1, \dots, D_{R_6}\}$, $T = \{T_1, \dots, T_R\}$ — множество переменных обратной связи; $T \subseteq \{T_1, \dots, T_{R_6}\}$. Фрагмент A (см. рис. 9.10) включает r связей D_1, \dots, D_r с входов ПЛМ $_1, \dots, \dots, \text{ПЛМ}_{U_6}$ на входы регистра RG , а фрагмент B включает G_6 связей p_1, \dots, p_{G_6} с выходов ПЛМ $_1, \dots, \text{ПЛМ}_{U_6}$ на входы ПЗУ $_1$ и r связей D_1, \dots, D_r с выходов ПЗУ $_1$ на входы регистра RG . Удалим фрагмент B , тогда будет получена одноуровневая базовая схема, показанная на рис. 9.1. При удалении фрагмента A схема на рис. 9.10 будет двухуровневой с РМ-структурой (см. § 8.4), М-подсхема которой построена на ПЛМ $_1, \dots, \text{ПЛМ}_{U_6}$, а Р-подсхема — на ПЗУ $_1, \dots, \text{ПЗУ}_{Q_6}$. Выше в § 8.4 было показано, что М-подсхема позволяет заменить множество X входных переменных множеством P дополнительных переменных, как правило, меньшей мощности. В рассматриваемом случае $P = \{p_1, \dots, p_{G_6}\}$.

Схемы, в которых не используется либо фрагмент A , либо фрагмент B , будем называть предельными. Если в схеме используются оба эти фрагмента, то она будет частично одноуровневой и частично двухуровневой. Изменяя настройку ПЛМ и ПЗУ, можно реализовать как предельные, так и промежуточные структуры, и в этом смысле структура схемы на рис. 9.10 является перестраиваемой.

Рассмотрим предельную РМ-структуру (не используется фрагмент A). Очевидно, за один такт работы МПА из его любого состояния a_m можно перейти не более чем в 2^{G_6} различных состояний. В этом смысле целесообразно увеличивать значение G_6 . С другой стороны, увеличение G_6 на k единиц приводит к росту объема ПЗУ $_1$ в 2^k раз. Поэтому из соображений повышения эффективности использования площади ПЗУ $_1$ не следует выбирать значение G_6 больше чем 2—3.

Рассмотрим теперь предельную одноуровневую или Р-структуру (не используется фрагмент B). В этом случае за один такт работы МПА из его любого состояния a_m можно перейти не более чем в 2^r различных состояний, причем

увеличение значения r не приводит к существенному возрастанию площади матриц ПЛМ₁, ..., ПЛМ_{U₆} [31]. Недостатком Р-структуры является наличие довольно жестких ограничений на число переходов между состояниями МПА, обусловленных небольшим числом q промежуточных шин промышленных ПЛМ. В то же время для РМ-структуры при построении Р-подсхемы на ПЗУ указанные выше ограничения существенно смягчаются. Несмотря на то, что М-подсхема строится на ПЛМ₁, ..., ПЛМ_{U₆} при разумном кодировании внутренних состояний МПА и с учетом того, что $G_6 \leq 2-3$, при реализации автоматом даже очень сложных ГСА промежуточных шин ПЛМ оказывается вполне достаточно.

Таким образом, в РМ-структуре можно выполнить существенно большее, чем в Р-структуре, число переходов между состояниями МПА, но трудно реализовать многонаправленные переходы, т. е. переходы из состояний $a_m \in A$, для которых мощность множества $A(a_m)$ достаточно велика; в Р-структуре, наоборот, многонаправленные переходы реализуются довольно просто, но возникают жесткие ограничения на допустимое число переходов в автомате. Реализация перестройки позволяет соединить в одной базовой схеме преимущества обеих предельных структур.

Пусть из некоторого состояния МПА a_m имеются переходы в f_m различных состояний a_{s1}, \dots, a_{sf_m} . Тогда при $f_m \leq 2^{G_6}$, все переходы реализуются РМ-структурой; если же $f_m > 2^{G_6}$, то часть переходов реализуется РМ-структурой, а оставшаяся часть — Р-структурой. Результаты исследования ГСА, встречающихся на практике, показывают, что таких состояний a_m , для которых $f_m > 2^{G_6}$ ($G_6 \leq 2-3$), обычно немного, поэтому основная масса переходов будет реализована РМ-структурой, т. е. будет использована ее положительная особенность — простота реализации большого числа переходов. Многонаправленные переходы реализуются частично РМ- и частично Р-структурами. Использование положительных особенностей каждой из структур, определяемых спецификой внутренней организации ПЛМ и ПЗУ, применяемых для построения соответствующих подсхем, позволяет при малых затратах оборудования реализовать очень сложные ГСА при одновременном обеспечении высокого быстродействия.

При построении Р-структуры коды внутренних состояний автомата выбираются из условия ослабления функ-

циональной зависимости функций D_{r+1}, \dots, D_{R_6} от входных переменных. При синтезе же подсхемы из ПЛМ₁,, ПЛМ_{U₆} желательно минимизировать число различных членов в ДНФ системы БФ, реализуемых этой подсхемой. Решение последней задачи также тесно связано с задачей кодирования [36, 44, 45]. Кодирование состояний для ослабления функциональной зависимости функций $D_{r+1}, \dots, \dots, D_{R_6}$ и минимизации числа используемых промежуточных шин в ПЛМ₁,, ПЛМ_{U₆} можно было бы решать одновременно. Однако часто целесообразнее их решать раздельно, используя дополнительное ПЗУ₀ для соответствующего перекодирования (на рис. 9.10 выделено штриховой линией). Введение такого элемента, как правило, компенсируется резким снижением трудоемкости раздельного кодирования по сравнению с совместным. Если по каким-либо причинам желательно реализовать совместное кодирование, то ПЗУ₀ не используется, а на его место в базовом ТЭЗ устанавливается «заглушка» — элемент, обеспечивающий непосредственные проводные связи соответствующих входов T_1, \dots, T_{R_6} и выходов T'_1, \dots, T'_{R_6} (см. рис. 9.10).

Покажем теперь, как реализуются в перестраиваемой базовой схеме различные переходы между состояниями МПА. Будем описывать их строками таблицы переходов и положим, что в зависимости от назначения последней в ней либо будет присутствовать столбец $P(a_m, a_s)$ (табл. 9.10), либо нет (табл. 9.11) *. В столбец $P(a_m, a_s)$ будем записывать те переменные $p_g \in P$, которые на соответствующем переходе в МПА принимают единичные значения. Рассмотрим фрагменты ГСА (рис. 9.11), где кружками с символами a_{si}, a_{mj} обозначены позиции меток. Будем предполагать, что они введены в соответствующие фрагменты с помощью любого из ранее рассмотренных алгоритмов, например с помощью алгоритма Φ_1 или Φ_2 (см. § 2.4). Сами метки обозначены символами $a_{m1}, \dots, a_{m4}, a_{s1}, \dots, a_{s5}$. Покажем, как при $G_6=r=2$ реализовать все переходы из состояния a_{m1} в состояния из множества $A(a_{m1}) = \{a_{s1}, a_{s2}, a_{s3}, a_{s4}, a_{s5}\}$, число $f_{m1} = |A(a_{m1})| = 5$ которых больше, чем $2^{G_6} = 4$. Соответствующие пути перехода отмечены (рис. 9.11,а) ли-

* Из табл. 9.10 и 9.11 исключены выходные переменные; в столбце $P(a_m, a_s)$ табл. 9.10 показан один из возможных вариантов замены переменных.

Таблица 9.10

Номер строки	N_1	a_m	a_s	$X(a_m, a_s)$	$P(a_m, a_s)$	Адрес ПЗУ ₁
1	1	a_{m1}	a_{s1}	\bar{x}_{j1}	p_1	$K(a_{m1}), p_1=1, p_2=0$
2	2		a_{s2}	$x_{j1}x_{j2}$	p_2	$K(a_{m1}), p_1=0, p_2=1$
3	3		a_{s3}	$x_{j1}\bar{x}_{j2}x_{j3}$	p_1p_2	$K(a_{m1}), p_1=p_2=1$
4	4		b_i	$x_{j1}x_{j2}\bar{x}_{j3}$	—	$K(a_{m1}), p_1=p_2=0$
5	1	a_{m2}	a_{s1}	\bar{x}_{j1}	p_1	$K(a_{m2}), p_1=1$
6	2		b_i	x_{j1}	—	$K(a_{m2}), p_1=0$
7	1	a_{m3}	a_{s1}	$\bar{x}_{j1}x_{j2}$	p_1	$K(a_{m3}), p_1=1$
8	2		b_i	$\bar{x}_{j1}\bar{x}_{j2}$	—	$K(a_{m3}), p_1=0$
9	3		b_i	x_{j1}	—	$K(a_{m3}), p_1=0$
10	1	a_{m4}	b_{i1}	x_{j1}	p_1	$K(a_{m4}), p_1=1$
11	2		b_{i2}	\bar{x}_{j1}	—	$K(a_{m4}), p_1=0$

Таблица 9.11

Номер строки	N_1	a_m	a_s	$X(a_m, a_s)$
1	5	a_{m1}	a_{s4}	$x_{j1}\bar{x}_{j2}\bar{x}_{j3}x_{j4}$
2	6		a_{s6}	$x_{j1}\bar{x}_{j2}x_{j3}\bar{x}_{j4}$
3	3	a_{m2}	a_{s2}	$x_{j1}x_{j2}$
4	4		a_{s3}	$x_{j1}\bar{x}_{j2}$
5	4	a_{m3}	a_{s2}	$\bar{x}_{j1}\bar{x}_{j2}$
6	5		a_{s3}	$x_{j1}x_{j3}$
7	6		a_{s4}	$x_{j1}\bar{x}_{j3}$
8	3	a_{m4}	a_{s1}	$x_{j1}x_{j2}$
9	4		a_{s2}	$x_{j1}\bar{x}_{j2}$
10	5	a_{m4}	a_{s3}	$\bar{x}_{j1}x_{j3}$
11	6		a_{s4}	$\bar{x}_{j1}\bar{x}_{j3}$

Примечание. В табл. 9.10 и 9.11 N_1 — номер пути на рис. 9.11.

ниями и цифрами 1, ..., 6 (в кружках). Число состояний (меток), в которые ведут пути 1, 2, 3 (отнесем их к первому типу), выбрано равным $2^{G_6} - 1 = 3$. Путь 4 (отнесем его ко второму типу) имеет номер $2^{G_6} = 4$, заканчивается в точке b_i и является фиктивным (не ведет ни в какую метку). Оставшиеся два пути 5 и 6 отнесем к третьему типу.

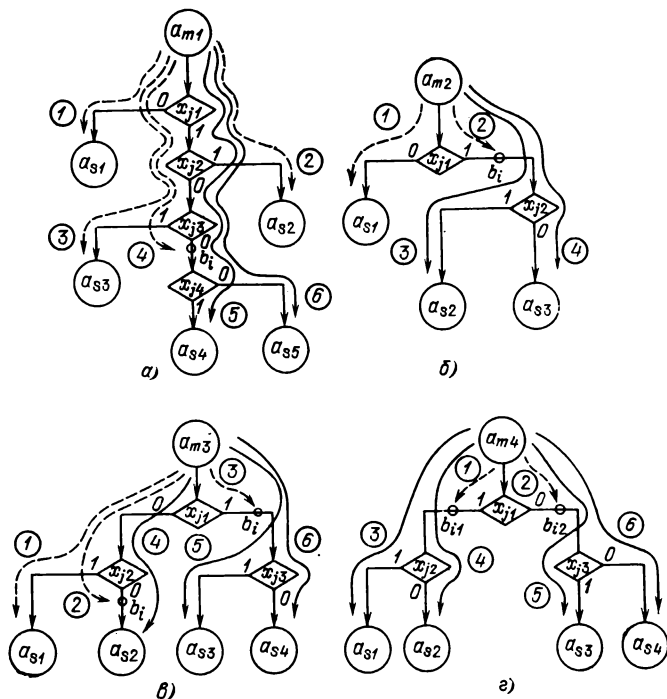


Рис. 9.11

Реализуем в МПА переходы из a_{m1} в a_{s1}, a_{s2}, a_{s3} РМ-структурой (см. табл. 9.10), а в a_{s4}, a_{s5} — Р-структурой (см. табл. 9.11). Рассмотрим некоторые из них. Пусть $x_{j1} = x_{j2} = x_{j3} = x_{j4} = 1$ (путь 2), тогда будет выполняться переход в соответствии со строкой 2 табл. 9.10, и он будет реализовываться РМ-структурой, т. е. по адресу ПЗУ₁ (см. табл. 9.10), который возбуждается кодом $K(a_{m1})$ при $p_1=0, p_2=1$, на его выходах D_1, \dots, D_R будет сформирован код $K(a_{s2})$ следующего состояния a_{s2} . В это же время на вы-

ходах D_1, \dots, D_r ПЛМ₁, ..., ПЛМ_Г сформированы нулевые сигналы. В точке объединения одноименных выходов реализуется функция «монтажное ИЛИ».

Пусть теперь $x_{j1}=x_{j4}=1$, $x_{j2}=x_{j3}=0$ (пути 4 и 5), тогда будет выполняться переход в соответствии со строкой 4 табл. 9.10 и строкой 1 табл. 9.11. По адресу ПЗУ₁, который возбуждается кодом $K(a_{m1})$ при $p_1=p_2=0$, на его выходах будут сформированы значения $(r+1)$ -го, ..., R -го разрядов кода $K(a_{s4})$ при нулевых сигналах на выходах 1, ..., r . В это же время на выходах D_1, \dots, D_r ПЛМ₁, ..., ПЛМ_Г будут сформированы значения 1-го, ..., r -го разрядов кода $K(a_{s4})$. Заметим, что из принципа кодирования состояний (см. § 9.2) значения $(r+1)$ -го, ..., R -го разрядов для кодов $K(a_{s4})$ и $K(a_{s5})$ попарно совпадают, а соответствующие функции D_{r+1}, \dots, D_R не зависят от входных переменных. В силу этого ясно, что эти функции D_{r+1}, \dots, D_R могут быть сформированы на выходах ПЗУ₁, на входы которого подаются лишь значения переменных T_1, \dots, T_{R_6} , p_1, \dots, p_{G_6} . Таким образом, из a_{m1} выполняется переход либо в одно из состояний a_{s1}, a_{s2}, a_{s3} , согласно табл. 9.10, либо в одно из состояний a_{s4}, a_{s5} согласно табл. 9.11, причем в последнем случае всегда возбуждается строка 4 табл. 9.10 (пути перехода, помеченные на рис. 9.11 штриховой линией, реализуются РМ-, а непрерывной линией — Р-структурами). Тогда из состояния a_{m1} за один такт работы автомата можно выполнить переходы в одно из $2^{G_6}-1+2r$ различных состояний.

Проанализируем другие фрагменты на рис. 9.11, для которых примем $G_6=1$, $r=2$. Принцип выполнения переходов из состояния a_{m2} (рис. 9.11,б), сохраняется таким же, как и из a_{m1} (см. строки 5, 6 табл. 9.10 и строки 3, 4 табл. 9.11). На рис. 9.11,в возникла необходимость введения двух фиктивных точек b_i . Действительно, если РМ-структурой реализуются переходы из a_{m3} в $2^{G_6}-1$ различных состояний, то все оставшиеся переходы из a_{m3} должны вести для РМ-структуры в одну и ту же фиктивную точку (см. строки 7—9 табл. 9.10). Следуя этому принципу, на переходах из некоторого состояния может вводиться и более двух одинаково помеченных фиктивных точек.

Поясним некоторые переходы из a_{m3} . Если $x_{j1}=0$, $x_{j2}=1$, то переход в a_{s1} реализуется РМ-структурой согласно строке 7 табл. 9.10. Если $x_{j1}=x_{j2}=0$, то РМ-структурой формируются значения функций возбуждения D_{r+1}, \dots, D_R при $D_1=\dots=D_r=0$ (строка 8 табл. 9.10), а Р-структурой —

значения функций D_1, \dots, D_r (строка 5 табл. 9.11) для состояния a_{s2} . При $x_{j1} = x_{j3} = 1$ (согласно строке 9 табл. 9.10 и строке 6 табл. 9.11) осуществляется переход в a_{s3} . На рис. 9.11, *г* иллюстрируется правило введения нескольких фиктивных точек, помеченных неодинаково (в рассматриваемом конкретном примере b_{i1} и b_{i2}). Это позволяет разделить подмассив переходов из состояния a_{m4} в табл. 9.11 на два подмассива (строки 8, 9 и 10, 11 табл. 9.11). Состояния из столбца a_s каждого подмассива при кодировании могут размещаться в различных столбцах карты Карно (см. § 9.2), что предоставляет возможность сокращать число состояний в столбце a_s некоторого подмассива переходов без введения дополнительных меток. Все переходы в фиктивные точки из a_{m4} реализуются РМ-структурой согласно строкам 10, 11 табл. 9.10, а переходы в $a_{s1}, a_{s2}, a_{s3}, a_{s4}$ — Р-структурой согласно строкам 8, 9, 10, 11 табл. 9.11. По аналогии при $G_6 > 1$ часть переходов может быть реализована РМ-структурой в состояния МПА, а оставшаяся часть — в фиктивные точки, некоторые из них помечены разными символами. При этом суммарное число переходов, реализуемых РМ-структурой, в различные состояния и неодинаково помеченные фиктивные точки не должно превышать 2^{G_6} .

Пусть на основании анализа совокупности ГСА получены численные значения параметров $N_{\max}, L_{\max}, M_{\max}, R_{\max}, B_{\max}$ (см. § 9.1) и заданы конкретные ПЛМ (s, t, q), ПЗУ (s', t') и регистры памяти, из которых должна быть построена схема МПА. Для определения числа элементов базовой схемы, показанной на рис. 9.10, выполняются следующие процедуры:

1. Определяется число R_6 элементов памяти автомата, которое должно быть равно R_{\max} .

2. Выбирается число G_6 переменных множества P . Увеличение значения G_6 приводит к росту числа переходов, реализуемых РМ-структурой, но в то же время к снижению эффективности использования площади ПЗУ₁, к сокращению числа выходов ПЛМ₁, ..., ПЛМ _{U_6} , на которые могут распределяться переменные из множества Y , и к усложнению процедуры синтеза логической схемы МПА. Поэтому не следует выбирать значение G_6 , большее 2—3. Для его уточнения можно использовать неравенство $G_6 \leq (s' - R_6)$, которое желательно выполнить.

3. Из выражения $U_6 = \lceil L_{\max} / (s - R_6) \rceil$, полученного на основе (6.5) (см. § 6.4), определяется число U_6 ПЛМ (s, t, q) в базовой схеме. Это выражение отличается от аналогичного выражения в § 9.2, 9.3, поскольку в нем

отсутствует член $\lceil B_{\max}/q \rceil$, что, как правило, допустимо в силу специфики перестраиваемой базовой схемы на рис. 9.10. Здесь предполагается, что значение q выбрано таким, как и у большинства промышленных ПЛМ, т. е., по крайней мере оно не меньше, чем $3s$. Смягчение ограничений на число промежуточных шин ПЛМ достигается использованием специальных алгоритмов замены входных переменных и кодирования внутренних состояний. Причем по отношению к перестраиваемым структурам указанные алгоритмы могут быть использованы более эффективно.

4. Значение r выбирается эмпирически (см. § 9.2). На практике в этом случае следует выбирать $r=2, 3$ или 4.

5. Из выражения $Q_6 = \lceil [N_{\max} + R_6 - U_6(t - (r + G_6))] / t' \rceil$ определяется число Q_6 ПЗУ (s', t') в базовой схеме¹. Для подсчета общего числа ПЗУ необходимо учесть число K_0 модулей ПЗУ₀. Обычно $K_0=1$, причем в некоторых случаях $K_0=0$ (на месте ПЗУ₀ установлена «заглушка»). Здесь предполагается, что значения каждой выходной переменной будут формироваться лишь на одном выходе ПЛМ (s, t, q) или ПЗУ (s', t') и что $s' \geq (R_6 + G_6)$.

6. Для базовой схемы на рис. 9.10 должно выполняться неравенство $B_{\max} \leq 2^{R_6 + G_6}$. Однако если для определения значений M_{\max} и B_{\max} использованы алгоритмы Φ_3 и Φ_4 отметки ГСА (см. § 2.5, 9.1), то всегда $B_{\max} \leq 2^{R_6 + 1}$, т. е. при $G_6 \geq 1$ указанное неравенство всегда справедливо.

Пусть полностью сформирована базовая схема, т. е. определены численные значения всех ее параметров: U_6, Q_6, R_6, G_6 и r . При синтезе автомата S по конкретной ГСА Γ необходимо «вложить» его схему в заданную базовую структуру и при этом минимизировать число используемых ПЛМ (s, t, q) и ПЗУ (s', t'). Рассмотрим алгоритм [43] синтеза логической схемы МПА Мура, а его выполнение будем иллюстрировать примером построения управляющего устройства по ГСА (рис. 9.12) при $U_6=3, Q_6=2, G_6=1, r=2, R_6=5, s=7, t=6, q=12, s'=t'=6$.

1. Исходная ГСА омечается символами a_1, \dots, a_m по алгоритму Φ_2 (см. § 2.4). Так, ГСА на рис. 9.12 отмечена символами a_1, \dots, a_{31} .

2. С помощью метода, рассмотренного в § 6.4, реализуется распределение входных переменных между ПЛМ₁, ... ПЛМ_U, $U \leq U_6$, в результате чего ГСА может быть

¹ В приведенном выражении $\lceil C \rceil = 1$ при $C \leq 1$ и $\lceil C \rceil$ — ближайшее целое число, не меньшее C при $C > 1$.

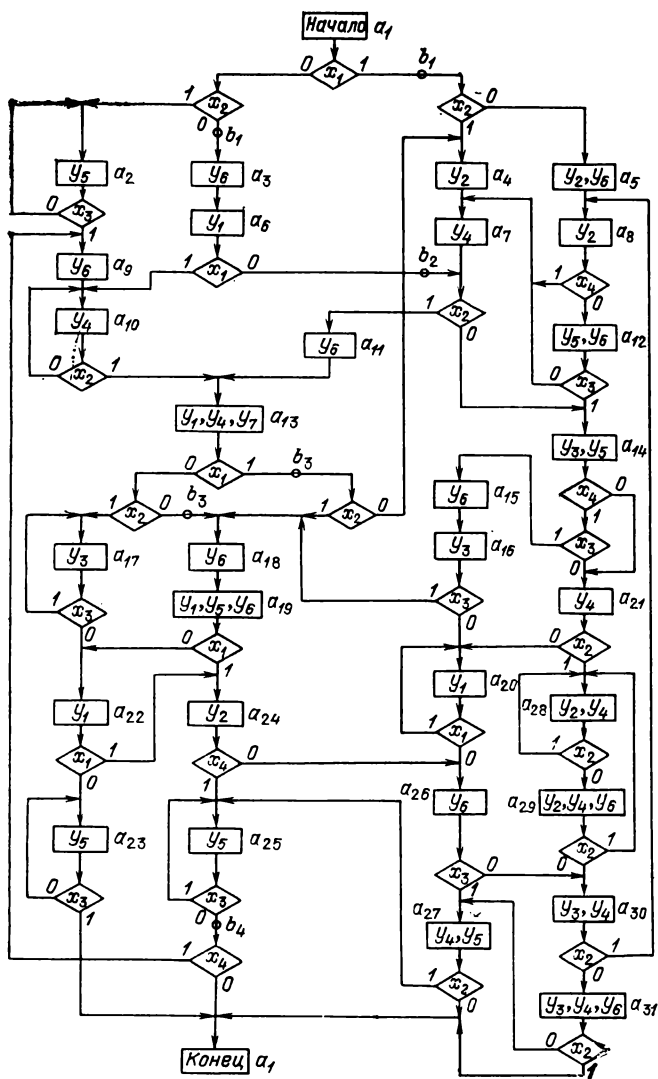


Рис. 9.12

отмечена дополнительными символами. Для заданной ГСА дополнительная отметка не требуется.

3. В [43] показано, что мощность f_m любого множества $A(a_m)$ не должна превышать величину $2^{G_6} - 1 + 2r$. На-

помним, что $A(a_m)$ — это множество состояний, записанных в столбце a_s таблицы переходов на переходах из состояния a_m . Если для некоторого $m \in \{1, \dots, M\}$ множество состояний $|A(a_m)| > (2^{G_6} - 1 + 2^r)$, то производится сокращение мощности $A(a_m)$ за счет дополнительной отметки ГСА. Для этих целей используется метод, описанный в § 9.2. В рассматриваемом примере дополнительная отметка не требуется.

4. Используя правила, рассмотренные в § 6.4, строятся такие подтаблицы переходов со столбцами $a_m Y(a_m)$, a_s , $X(a_m, a_s)$, что их число не больше чем $1 +]L/(s - R_6)[$, а количество входных переменных в каждой из них не превышает $s - R_6$ (число подтаблиц будет либо $1 +]L/(s - R_6)[$, либо $]L/(s - R_6)[$ и определяется наличием или отсутствием подтаблицы W_0). Ввиду тривиальности п. 4 его выполнение примером не иллюстрируется.

5. Если для некоторого a_m значение $f_m > 2^{G_6}$ то необходимо разделить подмассив переходов из состояния a_m на два (или более) подмассива за счет введения в ГСА фиктивных точек типа b_i . Разделение выполняется таким образом, чтобы максимально увеличить число совпадающих элементарных конъюнкций в столбце $X(a_m, a_s)$ измененных подтаблиц переходов для строк, реализующих пути типа 1 и 2. Изменение подтаблиц переходов, полученных в п. 4, сводится к введению в них переходов, соответствующих фиктивным путям типа 2. Реализация п. 5 для примера приводит к введению на рис. 9.12 фиктивных точек $b_1 - b_4$.

В некоторых случаях позиции фиктивных точек b_i могут быть отмечены дополнительными символами типа a_j , что приведет к увеличению числа переходов, реализуемых РМ-структурой. Если установить критерий целесообразности замены лишь некоторых точек b_i символами a_j , то можно осуществить перераспределение реализации переходов в МПА между РМ- и Р-структурами.

6. Поочередно рассматривается каждая подтаблица переходов W_u ($u = \overline{0, U}$). Строки подтаблицы W_u , соответствующие путям перехода типа 1 и 2, выделяются в отдельную РМ-подтаблицу W_u^{PM} . Заметим, что все строки любого неразделенного подмассива переходов (их число не превышает 2^{G_6}) описывают пути перехода типа 1. Строки подтаблицы W_u , соответствующие путям перехода типа 3, выделяются в отдельную Р-подтаблицу W_u^P . В результате выполнения п. 6 будет сформировано столько же

РМ-подтаблиц, сколько было подтаблиц переходов, а число Р-подтаблиц обычно меньше, поскольку для некоторых подтаблиц W_u мощность любого множества $A(a_m)$ не превышает значения 2^{G_6} , в силу чего Р-подтаблица W_u^P не формируется. Переходы, описываемые РМ-подтаблицами, будем реализовать РМ-структурой, а Р-подтаблицами — Р-структурой. Три РМ-подтаблицы, построенные для нашего примера, показаны в табл. 9.12 — столбцы $a_m Y(a_m)$, a_s , $X(a_m, a_s)$. Выше верхней двойной горизонтальной черты выделены РМ-подтаблица W_0^{PM} со строками, описывающими переходы, на которых $X(a_m, a_s) = 1$. Другой двойной горизонтальной чертой разделены две РМ-подтаблицы: W_1^{PM} и W_2^{PM} . Переходы подтаблицы W_0^{PM} могут быть реализованы только на ПЗУ₁, а W_1^{PM} , W_2^{PM} — на ПЗУ₁ и ПЛМ. Построенные для нашего примера Р-подтаблицы W_1^P и W_2^P в табл. 9.13 разделены двойными горизонтальными линиями — столбцы $a_m Y(a_m)$, a_s , $X(a_m, a_s)$.

7. Состояния в столбце a_s Р-подтаблиц кодируются таким образом, чтобы ослабить функциональную зависимость от входных переменных функций возбуждения D_{r+1}, \dots, D_R ($R \leq R_6$), вырабатываемых на переходах, описанных этими подтаблицами. Соответствующий метод кодирования рассмотрен выше и более подробно в [31]. Результаты кодирования для нашего примера отображены в столбцах $K(a_m)$ и $K(a_s)$ табл. 9.13.

8. В РМ-подтаблицы добавляется столбец $P(a_m, a_s)$. Конъюнкции переменных из множества X , записанные в столбце $X(a_m, a_s)$, согласно правилам, описанным ниже, будут кодироваться G_6 -разрядными двоичными кодами. Пусть в i -й строке столбца $X(a_m, a_s)$, записана конъюнкция X_j и она закодирована кодом $P(X_j)$. Тогда в i -й строке столбца $P(a_m, a_s)$ будут записаны переменные p_k , для которых k — номер (слева направо) позиции единицы в коде $P(X_j)$.

9. Просматривается столбец a_s РМ-подтаблиц для каждого подмассива переходов из одного состояния. Если в некотором столбце a_s одного подмассива встречаются несколько раз одни и те же состояния (либо одни и те же символы b_i , соответствующие фиктивным точкам), то из них выбирается наиболее часто встречающееся и в соответствующих строках столбца $P(a_m, a_s)$ записывается прочерк.

Выполнение п. 9 позволяет записать в столбце $P(a_m, a_s)$ как можно больше прочерков, что упрощает дальней-

Таблица 9.12

Номер строки	$a_m Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$P(a_m, a_s)$	$F(a_m, a_s)$	Адрес ПЗУ ₁
1	a_3, y_6	01001	a_6	01010	1	1	$D_2 D_4$	18, 19
2	a_4, y_2	00001	a_7	00110	1	1	$D_3 D_4$	2, 3
3	a_5, y_2, y_6	10001	a_8	10010	1	1	$D_1 D_4$	34, 35
4	a_9, y_6	01000	a_{10}	00111	1	1	$D_3 D_4 D_5$	16, 17
5	a_{11}, y_6	10000	a_{13}	00010	1	1	D_4	32, 33
6	a_{15}, y_6	00011	a_{16}	01100	1	1	$D_2 D_3$	6, 7
7	a_{18}, y_6	11001	a_{19}	01011	1	1	$D_2 D_4 D_5$	50, 51
8	$a_1, —$	00000	a_2	00100	$\overline{x_1 x_2}$	p	D_3	1
9			b_1	00001	$\overline{x_1 x_2}$	—	D_5	0
10			b_1	00001	x_1	—	D_5	0
11	a_6, y_1	01010	a_{10}	00111	x_1	p	$D_3 D_4 D_5$	21
12			b_2	00000	$\overline{x_1}$	—	—	20
13	a_7, y_4	00110	a_{11}	10000	x_2	p	D_1	13
14			a_{14}	11000	$\overline{x_2}$	—	$D_1 D_2$	12
15	a_{10}, y_4	00111	a_{10}	00111	$\overline{x_2}$	—	$D_3 D_4 D_5$	14
16			a_{13}	00010	x_2	p	D_4	15
17	$a_{13}, y_1,$	00010	a_{17}	01101	$\overline{x_1 x_2}$	p	$D_2 D_3 D_5$	5
18	y_4, y_7		b_3	00001	$\overline{x_1 x_2}$	—	D_5	4
19			b_3	00001	x_1	—	D_5	4
20	$a_{19}, y_1,$	01011	a_{22}	11011	$\overline{x_1}$	—	$D_1 D_2 D_4 D_5$	22
21	y_5, y_6		a_{24}	10011	x_1	p	$D_1 D_4 D_5$	23
22	a_{20}, y_1	11010	a_{20}	11010	x_1	p	$D_1 D_2 D_4$	53
23			a_{26}	11100	$\overline{x_1}$	—	$D_1 D_2 D_3$	52
24	a_{21}, y_4	01110	a_{20}	11010	$\overline{x_2}$	—	$D_1 D_2 D_4$	28
25			a_{28}	10110	x_2	p	$D_1 D_3 D_4$	29
26	a_{22}, y_1	11011	a_{24}	10011	x_1	p	$D_1 D_4 D_5$	55
27			a_{23}	10100	$\overline{x_1}$	—	$D_1 D_3$	54
28	a_{27}, y_4, y_6	01111	a_{25}	10101	x_2	p	$D_1 D_3 D_5$	31
29			a_1	00000	$\overline{x_2}$	—	—	30

Продолжение табл. 9.12

Номер строки	$a_m Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$P(a_m, a_s)$	$F(J_m, a_s)$	Адрес ПЗУ ₁
30	a_{28}, y_2, y_4	10110	a_{28}	10110	x_2	p	$D_1 D_3 D_4$	45
31			a_{29}	10111	$\overline{x_2}$	—	$D_1 D_3 D_4 D_5$	44
32	a_{29}, y_2, y_4, y_6	10111	a_{28}	10110	x_2	p	$D_1 D_3 D_4$	47
33			a_{30}	11110	$\overline{x_2}$	—	$D_1 D_2 D_3 D_4$	46
34	a_{30}, y_3, y_4	11110	a_8	10010	x_2	p	$D_1 D_4$	61
35			a_{31}	11111	$\overline{x_2}$	—	$D_1 D_2 D_3 D_4 D_5$	60
36	a_{31}, y_3, y_4, y_6	11111	a_1	00000	x_2	p	—	63
37			a_{27}	01111	$\overline{x_2}$	—	$D_2 D_3 D_4 D_5$	62
38	a_2, y_5	00100	a_2	00100	$\overline{x_3}$	p	D_3	9
39			a_9	01000	x_3	—	D_2	8
40	a_8, y_2	10010	a_7	00110	x_4	p	$D_3 D_4$	37
41			a_{12}	00101	$\overline{x_4}$	—	$D_3 D_5$	36
42	a_{12}, y_5, y_6	00101	a_7	00110	$\overline{x_3}$	p	$D_3 D_4$	11
43			a_{14}	11000	x_3	—	$D_1 D_2$	10
44	a_{14}, y_3, y_5	11000	a_{21}	01110	$\overline{x_4}$	—	$D_2 D_3 D_4$	48
45			a_{21}	01110	$\overline{x_3 x_4}$	—	$D_2 D_3 D_4$	48
46			a_{15}	00011	$x_3 x_4$	p	$D_4 D_5$	49
47	a_{18}, y_3	01100	a_{18}	11001	x_3	—	$D_1 D_2 D_5$	24
48			a_{20}	11010	$\overline{x_3}$	p	$D_1 D_2 D_4$	25
49	a_{17}, y_3	01101	a_{17}	01101	x_3	—	$D_2 D_3 D_5$	26
50			a_{22}	11011	$\overline{x_3}$	p	$D_1 D_2 D_4 D_5$	27
51	a_{23}, y_5	10100	a_1	00000	x_3	—	—	40
52			a_{23}	10100	$\overline{x_3}$	p	$D_1 D_3$	41
53	a_{24}, y_2	10011	a_{25}	10101	x_4	p	$D_1 D_3 D_5$	39
54			a_{26}	11100	$\overline{x_4}$	—	$D_1 D_2 D_3$	38
55	a_{25}, y_5	10101	a_{25}	10101	x_3	—	$D_1 D_3 D_5$	42
56			b_4	00000	$\overline{x_3}$	p	—	43
57	a_{28}, y_6	11100	a_{27}	01111	x_3	—	$D_2 D_3 D_4 D_5$	56
58			a_{30}	11110	$\overline{x_3}$	p	$D_1 D_2 D_3 D_4$	57

шую реализацию функций p_1, \dots, p^{G_6} . Рассмотрим подмассив переходов из состояния a_1 (см. табл. 9.12), который включает строки 8—10. В столбце a_s этого подмассива символ b_1 повторяется 2 раза, и в строках 9, 10 столбца $P(a_m, a_s)$ записываются прочерки. По той же причине записываются прочерки в строках 18 и 19, 44 и 45 этого столбца.

Таблица 9.13

Номер строки	$a_m Y(a_m)$	$K(a_m)$	a_s	$K(a_s)$	$X(a_m, a_s)$	$F(a_m, a_s)$
1	$a_1, —$	00000	a_3	01001	$\overline{x_1 x_2}$	$D_2(D_5)$
2			a_4	00001	$x_1 x_2$	(D_5)
3			a_5	10001	$x_1 \overline{x_2}$	$D_1(D_5)$
4	a_6, y_1	01010	a_{11}	10000	$\overline{x_1 x_2}$	D_1
5			a_{14}	11000	$\overline{x_1 x_2}$	$D_1 D_2$
6	a_{13}, y_1, y_4, y_7	00010	a_{18}	11001	$\overline{x_1 x_2}$	$D_1 D_2(D_5)$
7			a_{18}	11001	$x_1 x_2$	$D_1 D_2(D_5)$
8			a_4	00001	$x_1 \overline{x_2}$	(D_5)
9	a_{25}, y_5	10101	a_1	00000	$\overline{x_3 x_4}$	—
10			a_9	01000	$\overline{x_3 x_4}$	D_2

10. Вычисляются значения чисел η_1, \dots, η_J , где $\eta_j = |A(X_j)|$, $j=1, J$, J — число различных конъюнкций, не равных тождественно единице записанных в столбце $X(a_m, a_s)$ РМ-подтаблиц. Здесь $A(X_j)$ — множество состояний, на переходах из которых в столбце $X(a_m, a_s)$ записана конъюнкция X_j , причем она еще не закодирована кодом $P(X_j)$ (это требование необходимо учитывать при вычислении значений η_1, \dots, η_J на последующих этапах). В табл. 9.14 записаны конъюнкции X_j из табл. 9.12, для которых $\eta_j \neq 0$, и соответствующие им множества $A(X_j)$ и числа η_j .

11. Рассматривается множество K_p различных G_6 -разрядных двоичных кодов, исключая код, содержащий все нули. Среди чисел η_1, \dots, η_J выберем наибольшее η_i и сопоставим ему и соответствующей конъюнкции X_j один из кодов множества K_p , т. е. $P(X_j)$. На выбор $P(X_j)$ оказывает влияние следующее ограничение. Двум любым конъюнкци-

ям X^a и X^b одного подмассива переходов нельзя назначать одинаковые коды, если в столбце a_s строк X^a и X^b указанного подмассива записаны различные состояния. Если это ограничение приведет к тому, что в множестве K_p будет отсутствовать допустимый код, то исключается из рассмотрения минимальное число строк РМ-подтаблиц с конъюнкциями X_j . Пусть $AP(X_j)$ — множество состояний, на переходах из которых в столбце $X(a_m, a_s)$ записана

Таблица 9.14

j	X_j	$A(X_j)$	η_j
1	$\overline{x_1 x_2}$	$\{a_1, a_{13}\}$	2
2	x_1	$\{a_8, a_{19}, a_{20}, a_{22}\}$	4
3	$\overline{x_1}$	$\{a_8, a_{19}, a_{20}, a_{22}\}$	4
4	x_2	$\{a_7, a_{10}, a_{21}, a_{27}, a_{28}, a_{29}, a_{30}, a_{31}\}$	8
5	$\overline{x_2}$	$\{a_7, a_{10}, a_{21}, a_{27}, a_{28}, a_{29}, a_{30}, a_{31}\}$	8
6	$\overline{x_3}$	$\{a_2, a_{12}, a_{16}, a_{17}, a_{23}, a_{25}, a_{26}\}$	7
7	x_3	$\{a_2, a_{12}, a_{16}, a_{17}, a_{23}, a_{25}, a_{26}\}$	7
8	x_4	$\{a_5, a_{24}\}$	2
9	$\overline{x_4}$	$\{a_5, a_{24}\}$	2
10	$x_3 x_4$	$\{a_{14}\}$	1

конъюнкция X_j и она закодирована кодом $P(X_j)$. Тогда если закодировать состояния МПА таким образом, что множеству $AP(X_j)$ будет сопоставлен некоторый интервал булева пространства $K[AP(X_j)]$, то каждую из функций p_1, \dots, P_{G_6} , записанных в строках подмассивов переходов из состояний множества $AP(X_j)$, где также записана X_j , можно представить одной конъюнкцией $A_j X_j$, где $A_j=1$ на интервале $K[AP(X_j)]$ и $A_j=0$ на любом интервале, ортогональном $K[AP(X_j)]$. Продолжим рассмотрение нашего примера. Согласно табл. 9.14 $\max_i \eta_j = \eta_4 = 8$, а $K_p = \{1\}$.

Закодируем конъюнкцию $X_4 = x_2$ кодом 1, тогда в строках 13, 16, 25, 28, 30, 32, 34, 36 столбца $P(a_m, a_s)$ табл. 9.12 будет записана переменная p (поскольку $G_6=1$ и в множестве P один элемент, вместо переменной p_1 будем записывать переменную p без нижнего индекса, т. е. положим $P = \{p\}$).

12. Если в столбце a_s РМ-подтаблиц для некоторого подмассива переходов из a_m осталось единственное состояние a_j (оно может встречаться более одного раза) либо единственный символ b_i , и конъюнкция X_j , записанная в одной строке с этим состоянием (символом b_i), еще не закодирована, а в столбце $P(a_m, a_s)$ этого подмассива отсутствуют прочерки, то назначается $P(X_j) = 00 \dots 0$ (все нули), а в соответствующей строке столбца $P(a_m, a_s)$ записывается прочерк. Так, в табл. 9.12 после первого выполнения п. 11 записаны прочерки в строках 14, 15, 24, 29, 31, 33, 35, 37 столбца $P(a_m, a_s)$.

13. Если $\eta_1 = \dots = \eta_l = 0$, то выполняется переход к п. 14, иначе — к п. 11. Так, после первого выполнения лп. 11, 12 в табл. 9.14 произошли следующие изменения: $\eta_4 = \eta_5 = 0$. Результаты кодирования конъюнкций X_j для нашего примера отображены в столбце $P(a_m, a_s)$ табл. 9.12.

14. Рассматриваются функции p_1, \dots, p_{G_6} и кодируются внутренние состояния таким образом, чтобы минимизировать число различных членов в их ДНФ. Указанная задача может быть решена с помощью методов, описанных в [36, 44, 45].

15. С помощью методов, аналогичных рассмотренным выше, по РМ- и Р-подтаблицам строятся структурные РМ- и Р-подтаблицы и выполняется распределение выходных переменных из множества $Y = \{y_1, \dots, y_N\}$ между ПЛМ и ПЗУ схемы на рис. 9.10. В общем случае после выполнения п. 14 некоторые состояния МПА могут быть закодированы достаточно произвольно¹ и это не влияет на сложность функции p_1, \dots, p_{G_6} . Поэтому можно сформулировать следующую задачу. Выделить из множества Y такие подмножества Y^1, \dots, Y^U , что реализация соответствующих выходных функций на ПЛМ₁, ..., ПЛМ_U после рационального декодирования состояний, которые в п. 14 могли быть закодированы произвольно, позволила бы минимизировать число используемых промежуточных шин ПЛМ₁, ..., ПЛМ_U. Метод нахождения решения указанной задачи в настоящей книге не рассматривается. Если состояния декодированы произвольно, распределение выходных переменных выполняется так же, как и в § 6.4.

16. Нераспределенные после реализации п. 15 выходные переменные сначала произвольным образом назнача-

¹ Произвольное кодирование понимается в том смысле, что из кодов некоторого оставшегося подмножества может быть выбран любой.

ются на свободные выходы ПЗУ₁, ..., ПЗУ_{Q₆} (см. рис. 9.10), а затем, если некоторые из них по прежнему не распределены, на выходы ПЛМ_{U+1}, ..., ПЛМ_{U₆} (при наличии соответствующих соединений в базовой схеме допустимо распределение выходных переменных и на свободные выходы ПЗУ₀). Выполнение пп. 14—16 для нашего примера приводит к построению структурных РМ-подтаблиц (см. табл. 9.12) и Р-подтаблиц (см. табл. 9.13), а также к формированию множеств $Y^1 = \{y_1, y_4, y_7\}$, $Y^2 = \{y_2, y_3, y_5\}$, $Y^3 = \emptyset$, $Y^4 = \{y_6\}$, $Y^5 = \emptyset$.



Рис. 9.13

17. Если коды одних и тех же состояний, записанных в структурных РМ- и Р-подтаблицах, различны, то реализуется соответствующее перекодирование на ПЗУ₀. Для нашего примера оно не требуется и на место ПЗУ₀ устанавливается «заглушка» (см. рис. 9.13).

Окончательно ГСА на рис. 9.12 реализуется схемой, содержащей две ПЛМ и одно ПЗУ (одно посадочное место под ПЛМ и одно под ПЗУ остаются свободными). Эта схема показана на рис. 9.13. Таблицы 9.15 и 9.16 содержат информацию для программирования ПЛМ₁ и ПЛМ₂, табл. 9.17 — для программирования ПЗУ₁. Подобный пример синтеза логической схемы МПА при $G_6=2$ рассмотрен в [43].

При построении логической схемы МПА Мили в базовой структуре (см. рис. 9.10) сигналы p_1, \dots, p_{G_6} следует

также подать на входы ПЗУ₂, ..., ПЗУ_{Q₆}. Процедура синтеза логической схемы будет дополнительно включать специальную отметку ГСА, позволяющую назначить переменные из множества Y на свободные выходы ПЛМ₁, ..., ПЛМ _{U} ($U \leq U_6$) [32], а также установить зависимость выходных переменных, распределяемых на выходы ПЗУ₁, ..., ПЗУ_{Q₆}, не более чем от G_6 входных переменных из множества X .

В заключение заметим, что принцип перестройки может быть также применен и в схемах с произвольной структурой. Основу для использования этого принципа дают ме-

Таблица 9.15

T_1	T_2	T_3	T_4	T_5	x_1	x_2	D_1	D_2	p	y_1	y_4	y_7
0	0	0	—	0	0	1	.	.	1	.	.	.
—	1	0	1	—	1	—	.	.	1	.	.	.
—	—	1	1	—	—	1	.	.	1	.	.	.
0	0	0	0	0	0	0	.	1
0	0	0	0	0	1	0	1
0	1	0	1	0	0	1	1
0	—	0	1	0	0	0	1	1
0	0	0	1	0	1	1	1	1
—	1	0	1	—	—	—	.	.	.	1	.	.
0	0	0	1	0	—	—	.	.	.	1	1	1
—	—	1	1	—	—	—	1	.

Таблица 9.16

T_1	T_2	T_3	T_4	T_5	x_3	x_4	D_1	D_2	p	y_2	y_3	y_5
—	—	1	0	—	0	—	.	.	1	.	.	.
1	0	0	1	—	—	1	.	.	1	.	.	.
1	1	0	0	0	1	1	.	.	1	.	.	.
1	0	1	0	1	0	1	.	1
1	0	—	1	—	—	—	.	.	.	1	.	.
—	0	0	0	1	—	—	.	.	.	1	.	.
0	1	1	0	—	—	—	1	.
1	1	1	1	—	—	—	1	.
1	1	0	0	0	—	—	1	1
—	0	1	0	—	—	—	1
0	1	—	1	1	—	—	1

Таблица 9.17

Адрес	D_1	D_2	D_3	D_4	D_5	y_6	Адрес	D_1	D_2	D_3	D_4	D_5	y_6
0	1	.	31	1	.	1	.	1	.
1	.	.	1	.	.	.	32	.	.	.	1	.	1
2	.	.	1	1	.	.	33	.	.	.	1	.	1
3	.	.	1	1	.	.	34	1	.	.	1	.	1
4	1	.	35	1	.	.	1	.	1
5	.	1	1	.	1	.	36	.	.	1	.	1	.
6	.	1	1	.	.	1	37	.	.	1	1	.	.
7	.	1	1	.	.	1	38	1	1	1	.	.	.
8	.	1	39	1	.	1	.	1	.
9	.	.	1	.	.	.	41	1	.	1	.	.	.
10	1	1	.	.	.	1	42	1	.	1	.	1	.
11	.	.	1	1	.	1	44	1	.	1	1	1	.
12	1	1	45	1	.	1	1	.	.
13	1	46	1	1	1	1	.	1
14	.	.	1	1	1	.	47	1	.	1	1	.	1
15	.	.	.	1	.	.	48	.	1	1	1	.	.
16	.	.	1	1	1	1	49	.	.	.	1	1	.
17	.	.	1	1	1	1	50	.	1	.	1	1	1
18	.	1	.	1	.	1	51	.	1	.	1	1	1
19	.	1	.	1	.	1	52	1	1	1	.	.	.
21	.	.	1	1	1	.	53	1	1	.	1	.	.
22	1	1	.	1	1	1	54	1	.	1	.	.	.
23	1	.	.	1	1	1	55	1	.	.	1	1	.
24	1	1	.	.	1	.	56	.	1	1	1	1	1
25	1	1	.	1	.	.	57	1	1	1	1	.	1
26	.	1	1	.	1	.	60	1	1	1	1	1	.
27	1	1	.	1	1	.	61	1	.	.	1	.	.
28	1	1	.	1	.	.	62	.	1	1	1	1	1
29	1	.	1	1	.	.	63	1

тоды синтеза и схемные реализации МПА, рассмотренные в предыдущих главах. Например, включение в схему на рис. 9.10 К-подсхемы позволяет организовать перестраиваемую РМК-структуру, причем в зависимости от выбранного подхода к проектированию ее можно рассматривать и как базовую, и как произвольную. Отметим также, что представленные в § 9.2—9.4 базовые схемы могут быть реализованы в виде БИС, ориентированной на построение различных устройств управления для систем автоматики

и вычислительной техники. Ее настройка будет сводиться лишь к удалению либо к введению элементов односторонней проводимости (см. гл. 3) в матрицах ПЛМ и ПЗУ, размещаемых на одном кристалле.

9.5. Оценка аппаратурной избыточности базовых схем

Отсутствие возможностей изменения связей между элементами базовой схемы приводит к тому, что при реализации одних и тех же ГСА число таких элементов часто оказывается большим, чем в схемах с произвольной структурой. Вернемся к рассмотрению одноуровневых, многоуровневых и перестраиваемых базовых схем и оценим их аппаратурную избыточность.

Предположим, что одноуровневая схема с произвольной структурой строится на основе ПЛМ (s, t, q) и ПЗУ (s', t') . Тогда число U ПЛМ (s, t, q) и Q ПЗУ (s', t') , необходимых для построения логической схемы МПА, реализующего заданную отмеченную ГСА Γ с параметрами L, N, R, B , можно приближенно определить из следующих выражений (см. § 6.4, а также [31, 32]):

$$U = \max(\lceil L/(s-R) \rceil, \lceil B/q \rceil); \quad (9.2)$$

$$Q = \lceil [N - U(t - R/\varphi)] / t' \rceil. \quad (9.3)$$

Здесь φ — коэффициент, учитывающий то, что не все функции возбуждения D_1, \dots, D_R формируются на выходах каждой ПЛМ (согласно [32] $1,0 \leq \varphi \leq 1,8$). С достаточной степенью точности можно принять: $\varphi = 1$ при $U = 1$ и $\varphi = 1,8$ при $U > 1$. Следует также учитывать, что Q не может быть меньше нуля.

Выражения, аналогичные (9.2) и (9.3), для одноуровневой базовой схемы имеют следующий вид (см. § 9.2):

$$U' = \max(\lceil L/(s - R_6) \rceil, \lceil B/q \rceil); \quad (9.4)$$

$$Q' = \lceil [N + (R_6 - r) - U'(t - r)] / t' \rceil. \quad (9.5)$$

Тогда необходимое число ПЛМ и ПЗУ для реализации ГСА Γ в произвольной и базовой одноуровневых схемах определяется соответственно следующим образом: $K = U + Q$, $K' = U' + Q'$. Анализ выражений (9.2) — (9.5) позволяет приближенно оценить аппаратурные затраты в рассматриваемых схемах при реализации различных ГСА. В табл. 9.18 приведены некоторые сравнительные данные для схем, которые строятся из ПЛМ (16, 8, 48), ПЗУ (10, 8) и при $r = 4$, $\lceil L/(s - R_6) \rceil \lceil B/q \rceil$ [для всех R] $\lceil L/(s - R) \rceil \lceil B/q \rceil$. В табл. 9.19 приведены аналогичные дан-

Таблица 9.18

Номер строки	N	L	R_6	U'	Q'	\tilde{K}'	R	U	Q	\tilde{K}	$\tilde{K}' - \tilde{K}$	\tilde{K}'/\tilde{K}
1	20	20	8	3	2	5	5	2	2	4	1	1,25
2												1,25
3												1,25
4												1,0
5	32	32	8	4	3	7	5	3	3	6	1	1,15
6												1,15
7												1,15
8												1,0
9	50	50	8	7	4	11	5	5	3	8	3	1,4
10												1,2
11												1,1
12												1,0
13			1,3									
14			1,2									
15			1,1									
16			1,0									
17	0,9											
18	100	100	8	13	7	20	5	10	6	16	4	1,25
19												1,2
20												1,05
21												1,0
22			1,3									
23			1,15									
24			1,1									
25			1,0									
26	0,9											

Таблица 9.19

Номер строки	N	B	R_6	U'	Q'	\tilde{K}'	R	U	Q	\tilde{K}	$\tilde{K}' - \tilde{K}$	\tilde{K}'/\tilde{K}
1	20	100	8	3	2	5	5	3	1	4	1	1,25
2							6	3	1	4	1	1,25
3							7	3	1	4	1	1,25
4							8	3	2	5	0	1,0
5			10	3	2	5	6	3	1	4	1	1,25
6							7	3	1	4	1	1,25
7							8	3	2	5	0	1,0
8							9	3	2	5	0	1,0
9							10	3	2	5	0	1,0
10	50	300	8	7	4	11	5	7	2	9	2	1,2
11							6	7	3	10	1	1,1
12							7	7	3	10	1	1,1
13							8	7	4	11	0	1,0
14			10	7	4	11	6	7	3	10	1	1,1
15							7	7	3	10	1	1,1
16							8	7	4	11	0	1,0
17							9	7	4	11	0	1,0
18							10	7	5	12	-1	0,9
19	100	500	8	11	8	19	5	11	6	17	2	1,1
20							6	11	7	18	1	1,05
21							7	11	7	18	1	1,05
22							8	11	8	19	0	1,0
23			10	11	8	19	6	11	7	18	1	1,05
24							7	11	7	18	1	1,05
25							8	11	8	19	0	1,0
26							9	11	9	20	-1	0,95
27							10	11	10	21	-2	0,9

ные с тем отличием, что $]B/q[>]L/(s-R_0)[$. Анализ табл. 9.18, 9.19 показывает, что по числу элементов базовые схемы сложнее произвольных не более чем в 1,4 раза, что на практике вполне приемлемо. Заметим, что данные этих таблиц хорошо согласуются с результатами проектирования дискретных управляющих устройств для различных систем автоматики и вычислительной техники. В некоторых случаях (строки 17, 26 табл. 9.18, 18, 26, 27 табл. 9.19) оказывается, что число элементов базовой схемы меньше, чем произвольной. Такая ситуация может возникнуть за счет введения в ГСА дополнительных меток (см. § 9.2) для сокращения мощности некоторых множеств $A(a_m)$. В конечном счете это приводит к уменьшению числа функций возбуждения, формируемых на выходах ПЛМ, хотя в то же время и к снижению быстродействия дискретного устройства, построенного на основе соответствующей базовой схемы.

Числа U ПЛМ (s, t, q) и Q ПЗУ (s', t') многоуровневой схемы с PF-структурой (см. § 6.5) могут быть приближенно определены из следующих выражений:

$$U = \max(]L/(s-R)[,]B/q[); \quad (9.6)$$

$$Q =] [N+R-U(t-K/\varphi)/k]/t' [. \quad (9.7)$$

Здесь k — коэффициент дублирования переменных из множества Y на выходах ПЛМ Р-подсхемы (см. § 9.3); φ — коэффициент, учитывающий то, что не все функции z_1, \dots, z_k (см. § 6.5) формируются на выходах каждой ПЛМ. С достаточной степенью точности можно принять: $\varphi=1$ при $U=1$, $K/\varphi=3,2$ при $U>1$.

Как и выше, значение Q не может быть меньше нуля. Выражения, аналогичные (9.6) и (9.7), для базовой схемы с PF-структурой имеют следующий вид (см. § 9.3):

$$U' = \max(]L/(s-R_0)[,]B/q[); \quad (9.8)$$

$$Q' =] [N+R_0+(K_0-r)-U'(t-r)/k]/t' [. \quad (9.9)$$

Анализ выражений (9.6)–(9.9) позволяет приближенно оценить аппаратные затраты в рассматриваемых схемах при реализации различных ГСА. В табл. 9.20 приведены некоторые сравнительные данные для схем, которые строятся из ПЛМ (16, 8, 48), ПЗУ (10, 8) и при $r=4$, $R_0=8$, $k=1,6$, $]L/(s-R_0)[>]B/q[$ для всех $R]L/(s-R)[>]B/q[$.

Перестраиваемую базовую схему на рис. 9.10 сравним с одноуровневой базовой схемой на рис. 9.1, поскольку

последняя уже сравнивалась с произвольными схемами. Числа U'' ПЛМ (s, t, q) и Q'' ПЗУ (s', t') в схеме на рис. 9.10 можно приближенно определить из следующих выражений (см. § 9.4):

$$U'' =]L/(s-R_6)[; \quad (9.10)$$

$$Q'' = K_0 +] [N+R_6-U''(t-(r+G_6))]/t' [. \quad (9.11)$$

Здесь значение Q'' не может быть меньше, чем $]R_6/t' [$.

Анализ выражений (9.4), (9.5), (9.10), (9.11) позволяет приближенно оценить аппаратные затраты в рассматриваемых схемах при реализации различных ГСА. В табл. 9.21 приведены некоторые сравнительные данные для схем, которые строятся из ПЛМ (16, 8, 48), ПЗУ (11, 8) при $K_0=1, r=4, B < 2^{R_6+G_6}$. Анализ табл. 9.21 показы-

Таблица 9.20

Номер строки	N	L	K_6	U'	Q'	\tilde{K}'	R	K	U	Q	\tilde{K}	$\tilde{K}'-\tilde{K}$	\tilde{K}'/\tilde{K}
1							6	7	5	6	11	2	1,2
2							6	8	5	6	11	2	1,2
3							6	9	5	6	11	2	1,2
4							6	10	5	6	11	2	1,2
5	50	50	10	7	6	13	8	9	7	5	12	1	1,1
6							8	10	7	5	12	1	1,1
7							7	8	6	5	11	2	1,2
8							8	8	7	5	12	1	1,1
9							7	10	6	5	11	2	1,2
10							6	7	8	11	19	2	1,1
11							6	8	8	11	19	2	1,1
12							6	9	8	11	19	2	1,1
13	100	80	9	10	11	21	8	9	10	10	20	1	1,05
14							7	8	9	10	19	2	1,1
15							8	8	10	10	20	1	1,05
16							7	9	9	10	19	2	1,1

вает, что преимущества схемы на рис. 9.10 возрастают при увеличении B . Заметим, что для многих строк табл. 9.21 можно принять $K_0=0$, что усиливает преимущества перестраиваемых базовых схем.

Рассмотрим теперь столбец \tilde{K}'' табл. 9.21. Легко заметить, что число элементов схемы (ПЛМ и ПЗУ) на рис. 9.10

Таблица 9.21

Номер строки	L	N	B	R ₆	U'	Q'	K'	G ₆	U''	Q''	\tilde{K}''	$\tilde{K}' - \tilde{K}''$	\tilde{K}' / \tilde{K}''
1	20	20	100	8	3	2	5	1	3	4	7	-2	0,7
2				2	3	4	7	-2	0,7				
3				3	3	5	8	-3	0,6				
4			10	4	2	6	1	4	4	8	-2	0,75	
5			700	8	15	1	16	2	3	4	7	9	2,3
6				3	3	5	8	8	2,0				
7				10	15	1	16	1	4	4	8	8	2,0
8			1000	8	21	1	22	3	3	5	8	14	2,75
9				10	21	1	22	1	4	4	8	14	2,75
10			1500	10	32	1	33	1	4	4	8	25	4,1
11	50	50	200	8	7	4	11	1	7	6	13	-2	0,85
12				2	7	7	14	-3	0,8				
13				3	7	8	15	-4	0,73				
14			10	9	3	12	1	9	6	15	-3	0,8	
15			700	8	15	1	16	2	7	7	14	2	1,15
16				3	7	8	15	1	1,1				
17				10	15	1	16	1	9	6	15	1	1,1
18			1000	8	21	1	22	3	7	8	15	7	1,5
19				10	21	1	22	1	9	6	15	7	1,5
20			1500	10	32	1	33	1	9	6	15	18	2,2

не превышает 15 и они могут быть размещены на базовом ТЭЗ для большинства применяемых типоразмеров печатных плат. Управляющие устройства, построенные на основе такого ТЭЗ, могут реализовать практически любые ГСА, встречающиеся при проектировании цифровых систем различного назначения.

Подводя итог выполненному анализу, можно сделать вывод, что применение базовых схем целесообразно для большинства практических задач, поскольку при этом обеспечивается существенное сокращение сроков проекти-

рования и изготовления дискретных устройств управления, а при небольшой серийности последних гарантируются и экономические выгоды. Исключение могут составлять лишь те устройства, для которых определяющим является требование минимальности числа используемых элементов.

9.6. Реализация цифровых устройств управления на основе базовых схем

Рассмотрим возможные варианты применения базовых структур для построения на их основе цифровых устройств управления различного назначения. Предположим, что по параметрам N_{\max} , L_{\max} , M_{\max} , R_{\max} , B_{\max} построена базовая схема, реализованная на базовом ТЭЭ. Ее внутренняя структура может быть любой (см. § 9.2—9.4). Пусть задана ГСА G , и на основе построенной базовой схемы необходимо синтезировать МПА S , реализующий ГСА G . Полученный ТЭЭ с установленными на нем запрограммированными ПЛМ и ПЗУ представляет собой управляющее устройство, функционирование которого описано граф-схемой алгоритма G . Отметим ГСА G метками с помощью любого из ранее рассмотренных методов и определим значения ее параметров N , L , M , R , B . Предположим, что $N \leq N_{\max}$, $L \leq L_{\max}$, $M \leq M_{\max}$, $R \leq R_{\max}$, $B \leq B_{\max}$, тогда синтез логической схемы МПА S выполняется согласно правилам, изложенным в § 9.2—9.4. Если $N > N_{\max}$, $L \leq L_{\max}$, $M \leq M_{\max}$, $R \leq R_{\max}$, $B \leq B_{\max}$, то расширение числа выходов может быть реализовано подключением к базовому ТЭЭ внешней Y -подсхемы или D -подсхемы (рис. 9.14). В этом случае множество Y делится на два подмножества Y^a и Y^b ($Y = Y^a \cup Y^b$) и число элементов множества Y^a не превышает числа свободных выходов базового ТЭЭ. Значения переменных z_1, \dots, z_V формируются либо на внешних выходах ПЛМ и (или) ПЗУ базовой схемы, либо на выходах T_1, \dots, T_R ($R \leq R_0$) регистра RG . Далее будем считать, что выходные шины T_1, \dots, T_{R_0} регистра базовой схемы всегда соединяются с внешним разъемом базового ТЭЭ. Y -подсхема (см. § 6.5) или D -подсхема (см. § 8.3) может быть реализована на отдельном ТЭЭ и будет содержать всего лишь несколько элементов типа ПЗУ (ПЛМ) или дешифраторов. С помощью рассмотренного подхода число выходов базового ТЭЭ может быть расширено практически неограниченно.

Если выполняются условия $L > L_{\max}$, $N \leq N_{\max}$, $M \leq M_{\max}$, $R \leq R_{\max}$, $B \leq B_{\max}$, то расширение числа входов может

быть реализовано подключением к базовому ТЭЗ внешней М-подсхемы (рис. 9.15). В этом случае множество X делится на два подмножества: X_g и X_0 ($X = X_g \cup X_0$). Переменные из множества X_0 заменяются переменными из множества $P = \{p_1, \dots, p_G\}$. Множества X_g , X_0 и P формируются таким образом, чтобы выполнялось условие $|X_g \cup P| \leq L_{\max}$ (см. § 8.4). Значения переменных d_1, \dots, d_x , управляющих работой М-подсхемы, формируются либо на внешних выходах ПЛМ и (или) ПЗУ базовой схемы, либо на

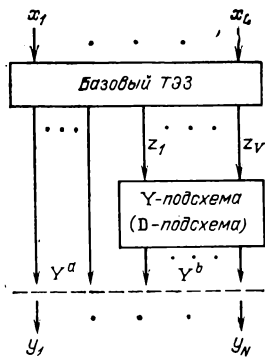


Рис. 9.14

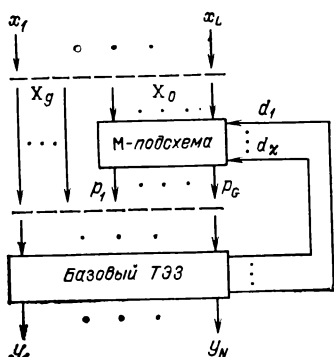


Рис. 9.15

выходах T_1, \dots, T_R ($R \leq R_0$) регистра. М-подсхема (см. § 8.4) может быть реализована на отдельном ТЭЗ и содержит всего лишь несколько элементов типа мультиплексоров или ПЛМ.

Объединение схем на рис. 9.14 и 9.15 позволяет синтезировать МПА S по ГСА Γ , для которой $N > N_{\max}$, $L > L_{\max}$, $M \leq M_{\max}$, $R \leq R_{\max}$, $B \leq B_{\max}$. Если для ГСА Γ не выполняются последние три неравенства, необходимо предварительно выполнить ее декомпозицию [42] на такие компонентные ГСА, для каждой из которых указанные неравенства справедливы (см. § 9.2).

Управляющее устройство, реализованное на базовом ТЭЗ, может применяться как самостоятельно, например для управления каким-либо электромеханическим объектом, так и в составе более сложных систем. В [31] рассматриваются базовые структуры, с помощью которых могут быть выполнены как собственно логическое управление, так и некоторые вычислительные операции. Указанные структуры предложено организовывать на основе микропроцессоров и матричных БИС и использовать в них принцип рас-

пределенного управления. Согласно этому принципу вычислительные операции выполняются в микропроцессорах по хранимой программе, а логика дискретных преобразований реализуется схемами вне микропроцессоров по специальным микропрограммам, причем построение таких схем осуществляется на основе подхода, рассмотренного выше (см. § 9.1—9.4). В [46] доказана целесообразность построения микропроцессорных устройств с распределенным управлением для решения ряда задач, например в системах промышленной автоматики показано, что они являются более быстродействующими, чем микро-ЭВМ с традиционной структурой, и в ряде случаев требуют управляющую память меньшего объема.

Список литературы

1. **Закревский А. Д.** Логический синтез каскадных схем. — М.: Наука, 1981.—414 с.
2. **Проектирование** цифровых вычислительных машин/ С. А. Майоров, Г. И. Новиков, О. Ф. Немолочнов и др.; Под ред. С. А. Майорова. — М.: Высшая школа, 1972.—344 с.
3. **Баранов С. И.** Синтез микропрограммных автоматов. — Л.: Энергия, 1979.—232 с.
4. **Sasao T.** Multiple-Valued Decomposition of Generalized Boolean Function and the Complexity of Programmable Logic Arrays. — IEEE, 1981, v. C-30, № 9, p. 633—645.
5. **Wood R. A.** A High Density Programmable Logic Array Chip. — IEEE Trans., 1979, v. C-28, № 9, p. 602—608.
6. **Hachel G. D., Newton A. R., Sangiovanni-Vincentelli A. Z.** An Algorithm for Optimal PLA Folding. — IEEE Trans., 1982, v. Cad-1, № 2, p. 63—67.
7. **Hnatek E. R.** Semiconductor Memory Update. Pt. I. ROMs — Computer Design, 1979, № 12, p. 67—77.
8. **Birkner J.** Reduce Random Logic Complexity. — Electronic Design, 1978, v. 26, № 17, p. 98—105.
9. **Larson D. V., Downey C.** Field Programmable Logic Devices. — Electronic Engineering, 1980, v. 52, № 1, p. 37—54.
10. **Исии О., Оомотэ И.** Аппаратура для микропрограмм. — Бит, 1979, т. 12, № 10, с. 1246—1263 (японск.).
11. **Баранов С. И., Баркалов А. А.** Применение программируемых логических матриц в цифровой технике. — Зарубежная радиоэлектроника, 1982, № 6, с. 67—79.
12. **Березенко А. И., Корягин Л. Н., Назарьян А. Р.** Микропроцессорные комплекты повышенного быстродействия. — М.: Радио и связь, 1981.—168 с.
13. **Адзума Ц.** Характеристики и методы использования программируемых логических матриц. — Дэнси Гидзюцу, 1980, т. 22, № 4, с. 102—106 (японск.).
14. **Такахире Т.** Логическая матрица программируемая пользователем. — NEC, 1982, т. 35, № 8, с. 62—64 (японск.).
15. **Такеда Т., Матчиро К., Судзуки М.** Биполярные программируемые пользователем матрицы с улучшенными характеристиками. — Кэнкю, 1982, т. 31, № 9, с. 1691—1705 (японск.).
16. **Минамидани Т.** Методы построения логики на ПЛМ.—Дэнси Кагаку, 1979, № 3, с. 18—30 (японск.).
17. **Баранов С. И., Синев В. Н.** Автоматы и программируемые матрицы. — Минск: Высшая школа, 1980.—136 с.

18. **Vaganov S.** Synthese des automates microprogrammes. — Moscow: Mir, 1983. — 336 с.
19. **Баранов С. И., Песчанский В. А., Синев В. Н.** Одноуровневая реализация микропрограммных автоматов на ПЛМ. — Изв. АН СССР. Сер. техническая кибернетика, 1983, № 5, с. 41—49.
20. **Скляров В. А., Лукьянова И. В.** Синтез автоматов по покрытиям структурной таблицы. — В кн.: Автоматика и вычислительная техника. — Минск: Вышэйшая школа, 1984, с. 81—86.
21. **Новиков С. В.** Синтез схем на программируемых логических матрицах. — Автоматика и вычислительная техника, 1977, № 5, с. 1—4.
22. **Новиков С. В.** Метод синтеза схем на ПЛМ. — Препринт 80—28. Киев: Институт кибернетики АН УССР, 1980.—23 с.
23. **Grass W.** Aufteilung von Schalfunktionen auf Programmierbare Logik Arrays (PLA). — Digital Processes, 1980, v. 6, № 1, p. 75—96.
24. **Буль Е. С.** Совместная реализация систем переключательных функций схемой из ПЛМ. — Автоматика и вычислительная техника, 1981, № 3, с. 36—42.
25. **Шварцман М. И.** Декомпозиция по выходам комбинационных ПЛМ структур. — Автоматика и вычислительная техника, 1981, № 6, с. 12—17.
26. **Сорокин Б. Л.** Декомпозиционный метод синтеза схем на программируемых логических матрицах. — Автоматика и вычислительная техника, 1982, № 4, с. 50—55.
27. **Супрун В. П.** Метод реализации на ПЛМ булевых функций, заданных в интервальной форме. — Автоматика и вычислительная техника, 1982, № 5, с. 30—36.
28. **Баранов С. И., Журавина Л. Н., Скляров В. А.** Автоматизация проектирования ЦВМ. — Минск: Вышэйшая школа, 1981.—206 с.
29. **Скляров В. А.** Минимизация числа микроопераций и логических условий в микропрограмме. — Автоматика и телемеханика, 1980, № 9, с. 157—164.
30. **Golden R. L. et al.** Design Automation and the Programmable Logic Array Macro. — IBM J. Res. Develop., 1980, v. 24, № 1, p. 23—31.
31. **Скляров В. А.** Синтез автоматов на матричных БИС/ Под ред. С. И. Баранова. — Минск: Наука и техника, 1984.—288 с.
32. **Скляров В. А.** Синтез микропрограммных автоматов на стандартных ПЛМ. — Автоматика и вычислительная техника, 1982, № 4, с. 28—35.
33. **Мелихов А. Н., Берштейн Л. С., Курейчик В. М.** Применение графов для проектирования дискретных устройств. — М.: Наука, 1974.—294 с.
34. **Франке Г.** Минимизация системы булевых функций с учетом заданных импликантов. — В кн.: Процессы и устройства управления в сетях связи. — М.: Наука, 1982, с. 118—121.
35. **Скляров В. А.** Построение управляющих автоматов на основе программируемых логических устройств. — Изв. АН СССР. Сер. техническая кибернетика, 1983, № 5, с. 59—69.
36. **Фрицнович Г. Ф.** Синтез дешифратора микрокоманд на ПЛМ. — Автоматика и вычислительная техника, 1981, № 2, с. 45—53.
37. **Баранов С. И., Килленберг Х.** Декомпозиция микропрограммных автоматов. — Автоматика и вычислительная техника, 1978, № 6, с. 5—11.
38. **Скляров В. А.** Синтез автоматов на ПЛМ с памятью. — Кибернетика, 1984, № 6, с. 57—64.
39. **Hemel A.** The PLA: a «Different Kind» of ROM. — Electronic Design, 1976, v. 24, № 1, p. 78—84.

40. **Скляров В. А.** Использование дешифраторов в микропрограммных автоматах с матричной структурой. — Изв. вузов. Сер. Приборостроение, 1982, № 12, с. 27—31.
41. **Скляров В. А.** Автоматы с регулярной структурой. — Управляющие системы и машины, 1984, № 2, с. 23—28.
42. **Баранов С. И., Журавина Л. Н., Песчанский В. А.** Обобщенный метод декомпозиции граф-схем алгоритмов. — Автоматика и вычислительная техника, 1982, № 5, с. 43—51.
43. **Скляров В. А.** Реализация управляющих устройств на основе базовых схем с перестраиваемой структурой. — Автоматика и вычислительная техника, 1985, № 2, с. 70—78.
44. **Каллиберзин А. Я., Чапенко В. П.** Способ кодирования входных состояний дискретных устройств при реализации на ПЛМ. — Автоматика и вычислительная техника, 1983, № 1, с. 41—47.
45. **Ланге Э. Э., Фрицкович Г. Ф.** Синтез логической структуры конечного автомата в классе ДНФ. — Автоматика и вычислительная техника, 1982, № 3, с. 43—50.
46. **Скляров В. А.** Микропроцессорное устройство управления технологическим оборудованием. — Автоматика и телемеханика, 1985, № 1, с. 118—121.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава первая. Элементы теории булевых функций	5
1.1. Булевы функции и их основные свойства	5
1.2. Минимизация булевых функций	8
1.3. Комбинационные схемы	13
Глава вторая. Автоматы и граф-схемы	15
2.1. Абстрактные автоматы	15
2.2. Структурные автоматы	18
2.3. Микропрограммы и граф-схемы	22
2.4. Микропрограммные автоматы и их синтез	25
2.5. Синтез М-автоматов	31
2.6. Минимизация числа строк таблицы переходов автомата	35
Глава третья. Программируемые логические устройства с матричной структурой	39
3.1. Матричная реализация булевых функций	39
3.2. Программируемые логические матрицы	42
3.3. Постоянные запоминающие устройства	49
3.4. Программируемые матрицы вентиляей	51
3.5. Программируемые матрицы логики	53
3.6. Программируемые мультиплексоры	55
Глава четвертая. Стандартные цифровые схемы на программируемых логических устройствах	56
4.1. Шифраторы и дешифраторы	56
4.2. Мультиплексоры и демультиплексоры	61
4.3. Схемы сравнения и контроля	63
4.4. Схемы для арифметико-логических устройств	68
4.5. Регистры и счетчики	72
Глава пятая. Синтез комбинационных схем на программируемых логических устройствах	77
5.1. Тривиальная реализация	77
5.2. Одноуровневые схемы	81
5.3. Многоуровневые схемы	89
5.4. Краткий обзор методов синтеза комбинационных схем из ПЛМ	102
Глава шестая. Синтез автоматов на программируемых логических матрицах комбинационного типа	108
6.1. Тривиальная реализация	108
6.2. Одноуровневая реализация автомата на ПЛМ	111
6.3. Синтез МПА по покрытиям структурных таблиц	113
6.4. Синтез МПА с разделением входных переменных	122
6.5. Синтез многоуровневой схемы автомата	140

Глава седьмая. Синтез автоматов на программируемых логических матрицах с памятью	164
7.1. Тривиальная реализация	164
7.2. Декомпозиция микропрограммного автомата	166
7.3. Синтез микропрограммного автомата	173
Глава восьмая. Синтез автоматов на элементах комбинированного базиса	183
8.1. Принципы использования комбинированного базиса	183
8.2. Построение автоматов на ПЛМ и ПЗУ	185
8.3. Синтез автоматов на ПЛМ и дешифраторах	186
8.4. Синтез автоматов на ПЛМ и мультиплексорах	196
Глава девятая. Базовые схемы автоматов на программируемых логических устройствах	214
9.1. Принцип построения базовых схем	214
9.2. Одноуровневые базовые схемы	216
9.3. Многоуровневые базовые схемы	229
9.4. Базовые схемы с перестраиваемой структурой	237
9.5. Оценка аппаратурной избыточности базовых схем	257
9.6. Реализация цифровых устройств управления на основе базовых схем	263
Список литературы	266

производственное издание

САМАРИИ ИОСИФОВИЧ БАРАНОВ
ВАЛЕРИИ АНАТОЛЬЕВИЧ СКЛЯРОВ

ЦИФРОВЫЕ УСТРОЙСТВА НА ПРОГРАММИРУЕМЫХ БИС С МАТРИЧНОЙ СТРУКТУРОЙ

Заведующий редакцией Ю. Н. Рысев
Редактор Л. В. Голованова
Художественный редактор Н. С. Шеин
Переплет художника Л. В. Брылева
Технический редактор Г. И. Колосова
Корректор З. Г. Галущкина

ИБ № 876

Сдано в набор 30.08.85	Подписано в печать 10.03.86	Т. (8701	
Формат 84×108 ^{1/32}	Бумага кн.-журн. № 2	Гарнитура литературная	
Печать высокая	Усл. печ. л. 14,2	Усл. кр.-отг. 14,515	Уч.-изд. л. 14,55
Тираж 28 000	Изд. № 20831	Заказ 421	Цена 90 к.
Издательство «Радио и связь». 101000 Москва, Почтамт, а/я 693			

Ордена Октябрьской Революции и ордена Трудового Красного Знамени МПО
«Первая Образцовая типография имени А. А. Жданова» Союзполиграфпрома
при Государственном комитете СССР по делам издательств, полиграфии и книж-
ной торговли. 113054, Москва, Валовая, 28.

УВАЖАЕМЫЕ ЧИТАТЕЛИ!

Издательство «Радио и связь» в 1987 г. готовит к изданию книгу **«Схемотехника БИС постоянных запоминающих устройств»** авторов О. А. Петросяна, И. Я. Козыря, Л. А. Коледова и Ю. И. Щетинина.

В этой книге изложены вопросы проектирования, производства и применения больших интегральных схем (БИС) постоянных запоминающих устройств (ПЗУ). Дан сравнительный анализ характеристик элементной и технологической базы микросхем ПЗУ. Рассмотрены вопросы синтеза электрических схем узлов, организации блоков, оптимизации схем и параметров запоминающих устройств по критериям максимальной информационной емкости и быстродействию, вопросы контроля параметров ПЗУ и проектирования функциональных узлов микроэлектронной аппаратуры с использованием больших интегральных схем для ПЗУ.

Предназначена для инженерно-технических работников, занимающихся проектированием и эксплуатацией электронной аппаратуры.