УДК 681.325.6

КОДИРОВАНИЕ И ОПТИМИЗАЦИЯ СИНТЕЗА АБСТРАКТНЫХ АВТОМАТОВ В УНИВЕРСАЛЬНОЙ ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Н.Г. Топольский

I. Введение

Рассмотрим метод синтеза и оптимизации в универсальной вычислительной среде (ВС) с индивидуальной настройкой элементов [1] абстрактных автоматов, заданных матрицами соединений [2] или каким-либо другим эквивалентным способом. В данной работе используется терминология из [2,3,4]. Приведем вкратце основные понятия и определения, необходимые для изложения метода.

Пусть автомат задан множествами внутренних состояний $A = \{a_1, a_2, \dots, a_n\}$, входных $X = \{x_1, x_2, \dots, x_\nu\}$ и выходных $Y = \{y_1, y_2, \dots, y_Q\}$ сигналов. Функции переходов и выходов задаются матрицей соединений. В качестве примера взята матрица соединений R_B – одноразрядного комбинационного сумматора последовательного действия [5]:

$$R_B = \begin{array}{c} \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{array} \left\| \begin{array}{cccc} a_1(y_1) & a_2(y_1) & a_3(y_2) & a_4(y_2) \\ x_1 & x_4 & x_2 \vee x_3 & 0 \\ 0 & x_2 \vee x_3 & x_1 & x_4 \\ x_1 & x_4 & x_2 \vee x_3 & 0 \\ 0 & x_2 \vee x_3 & x_1 & x_4 \end{array} \right\|$$

В общем случае для автомата Мили $\hat{A}$ введем следующие определения. Функцией переходов автомата в состоянии a_i, или просто функцией переходов (ФП), назовем двоичную функцию

$$(F_i)^t = \bigvee_{\lambda, j} (x_\lambda a_j)^t; \quad \lambda=1,2,\dots,v; \; j=1,2,\dots,n; \; t=0,1,2,\dots, \quad (I)$$

определенную на множестве пар $x_\lambda a_j \in \{X\times A\}$ и принимающую единичное значение только для тех пар $x_\lambda a_j$, которые вызывают переход автомата $\hat{A}$ в состояние a_i и которые соответственно собраны под знаком дизъюнкции в (I).

Пусть каждое состояние a_j автомата $\hat{A}$ кодируется набором внутренних двоичных переменных $\tilde{u}_1, \tilde{u}_2, \dots, \tilde{u}_s$, где $s =]\log_2 n[$, а $\tilde{u}_i$ означает либо прямое, либо инверсное значение переменной u_i. Каждой переменной u_i сопоставим элементарный автомат u_i. Прямой обобщенной функцией переходов (ОФП) элементарного автомата u_i назовем двоичную функцию

$$u_i(t+1) = \bigvee_j (F_j)^t, \quad (2)$$

где дизъюнкция берется по всем F_j^t таким, что в кодах соответствующих состояний $(a_j)\,\tilde{u}_i = 1$.

Обратной ОФП элементарного автомата u_i назовем двоичную функцию

$$\bar{u}_i(t+1) = \bigvee_j (F_j)^t, \quad (3)$$

где дизъюнкция берется по всем F_j, таким, что в кодах соответствующих состояний a_j, $\tilde{u}_i = 0$. Прямая и обратная ОФП также определены на множестве пар $x_\lambda a_j \in \{X\times A\}$ и принимают единичное значение только для тех пар $x_\lambda a_j$, которые содержатся под знаком дизъюнкции.

Функцией выходов (ФВ) $y_r(t)$ назовем двоичную функцию

$$y_r(t) = \bigvee_{\lambda, j} (x_\lambda a_j)^t, \quad r = 1,2,\dots,q, \quad (4)$$

значение которой определяется по аналогии с ФП (I) и в которой под знаком дизъюнкции собраны все пары (полные состояния) $x_\lambda a_j$, отмеченные выходом y_r. Далее, пусть каждый выход y_r кодируется набором выходных двоичных переменных $\tilde{\ell}_1, \tilde{\ell}_2, \dots, \tilde{\ell}_p$, где $p =]\log_2 q[$.

Обобщенной функцией выходов (ОФВ) назовем двоичную функцию

$$\ell_h(t) = \bigvee_r y_r(t), \quad h = 1,2,\dots,p, \quad (5)$$

значение которой определяется по аналогии с ОФП (2) и в которой под знаком дизъюнкции собраны все выходы y_r такие, что в кодах этих выходов $\tilde{\ell}_h = 1$.

Отмеченной ФП назовем двоичную функцию

$$(F_{i,r})^t = \bigvee_{\lambda, j} (x_\lambda a_j)^t, \quad (6)$$

в которой под знаком дизъюнкции собраны такие полные состояния $x_\lambda a_j$, которые входят в F_i и отмечены выходом y_r. Таким образом, в отмеченной $F_{i,r}$ индекс i указывает номер состояния a_i, в которое вызывается переход автомата каждым полным состоянием $x_\lambda a_j$ из $F_{i,r}$, а индекс r указывает, каким выходом (y_r) отмечены все полные состояния из $F_{i,r}$.

Заметим, что с учетом (I) и (4) выражения (2), (3) и (5) могут быть записаны в виде дизъюнкции различных полных состояний. Поэтому нетрудно получить выражения ФП, ОФП, ФВ, ОФВ и отмеченных ФП в виде совершенных дизъюнктивных нормальных форм этих функций, если под знаком дизъюнкции заменить каждое состояние a_j и вход x_λ соответствующим двоичным набором.

Для автоматов Мура в выражениях (4) и (5) под знаком дизъюнкции будут стоять только внутренние состояния a_j (вместо полных состояний) и, кроме того, отмеченная ФП будет совпадать с ФП, то есть

$$(F_i)^t = (F_{i,r})^t. \quad (7)$$

С учетом сделанных определений для автомата Мура B, заданного матрицей соединений R_B, запишем (здесь и далее для простоты записи индексы автоматного времени t при ФП опустим)

$$F_1 = x_1 a_1 \vee x_1 a_3; \; F_2 = x_4 a_1 \vee x_2 a_2 \vee x_3 a_2 \vee x_4 a_3 \vee x_2 a_4 \vee x_3 a_4, \dots$$

Пусть заданы элементы ВС (рис. I), обладающие соединительной и функциональной полнотой, причем состояния настройки эле-

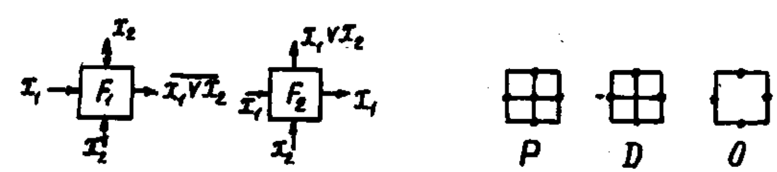

Рис. I.

мента ВС позволяют получать конъюнкцию нескольких переменных в строке, а дизъюнкцию – в столбце из элементов ВС. Элементарный автомат в такой ВС реализуется соответствующей настройкой нескольких элементов ВС [1,4]. На рис. 1 приведены конкретные элементы ВС, однако излагаемый здесь метод можно применить к широкому классу ВС, удовлетворяющих отмеченным выше требованиям к состояниям настройки элемента ВС [4].

2. Получение программы настройки ВС

2.1. Поставим задачу получения программы настройки элементов указанной ВС на реализацию синхронного абстрактного автомата по матрице соединений. Для решения этой задачи будем использовать подход, который оказался весьма удобным для реализации автоматов в широком классе ВС [4,6].

Вся программа настройки автомата Мили в ВС состоит из нескольких участков [4]. В строках программы $П_\wedge$ (рис. 2) реализуются все полные состояния $x_n a_j$ (точнее – двоичные наборы, соответствующие полным состояниям $x_n a_j$). На столбцы этой программы подаются входные $m_1, m_2, \dots, m_k$ и внутренние $u_1, u_2, \dots, u_s$ переменные, соответственно. Программа $П_\vee$ имеет $s+\rho$ столбцов, в каждом из которых реализуются либо ОФП $u_i(t+1)$, либо функция $\ell_h(t)$. Если полное состояние $x_n a_j$ входит в функцию $u_i(t+1)$ (или в $\ell_h(t)$), то на пересечении соответствующей строки программы $П_\wedge$ и соответствующего столбца программы $П_\vee$ ставится элемент F_2, в противном случае – элемент D. Наконец, программа $П_u$ служит для реализации элементарных автоматов $u_1, u_2, \dots, u_s$, в качестве которых будем использовать триггеры с раздельными входами. На их единичные входы подаются ОФП $u_i(t+1)$, а на нулевые – обратные ОФП, полученные инвертированием функций $u_i(t+1)$ (свойство дополнительности функций возбуждения этих входов триггера [7]).

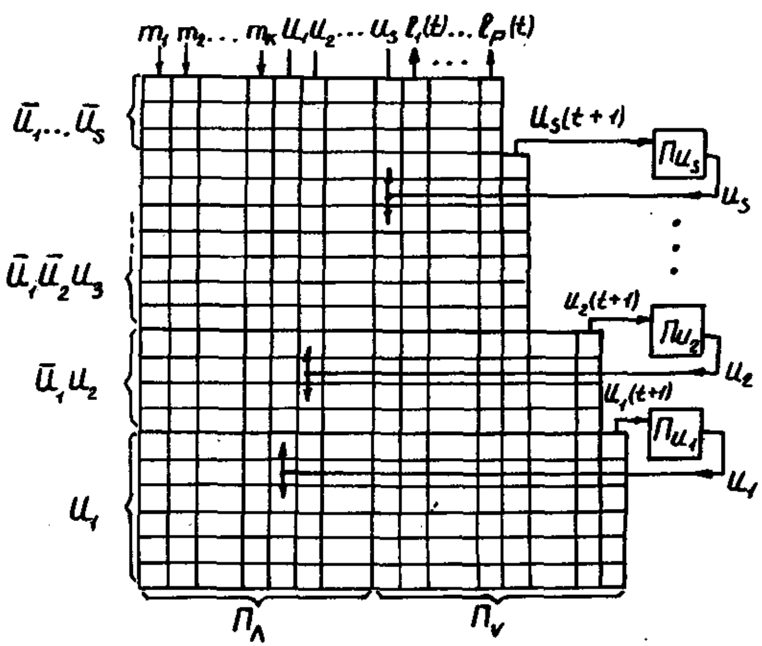

Рис. 2.

Аналогично реализуются в ВС автоматы Мура, если считать, что все полные состояния из F_i отмечены тем же выходом, что и внутреннее состояние a_i. В этом случае, выход автомата Мура в момент времени t определяется внутренним состоянием автомата в момент времени $(t+1)$.

С целью компактного размещения программ $П_{u_i}$ в ВС реализуем в программе $П_\wedge$, начиная снизу, вначале полные состояния из всех ФП F_j таких, для которых в кодах состояний a_j (с этими же индексами j) $\tilde{u}_1 = 1$. Затем реализуем полные состояния из всех ФП $F_{j'}$ таких, для которых в кодах состояний $a_{j'}$ $\tilde{u}_1 = 0$ и $u_2 = 1$, затем полные состояния из ФП $F_{j''}$, для которых в кодах состояний $a_{j''}$ $\tilde{u}_1 = 0$, $\tilde{u}_2 = 0$, $\tilde{u}_3 = 1$ и т.д. Последними реализуем полные состояния из F_n таких, что в коде состояния a_n, $\tilde{u}_1 = \tilde{u}_2 = \dots = \tilde{u}_s = 0$. Тогда в первых ρ столбцах программы $П_\vee$ реализуются ρ функций $\ell_j(t)$, в $\rho+1$ столбце реализуется ОФП $u_s(t+1)$, в $(\rho+2)$ столбце – ОФП $u_{s-1}(t+1)_s \dots$, в $(\rho+s)$ столбце – ОФП $u_1(t+1)$. При таком "ступенчатом" построении программы $П_\vee$ верхние участки последних s столбцов будут свободными (то есть не занятыми для реализации прямых ОФП), и их можно использовать для получения обратных ОФП $u_1(t+1)$ и размещения программ $П_{u_i}$. На рис. 2 показана в общем виде схема размещения программ $П_{u_i}$ и реализации ОФП.

Программа реализации в ВС автомата B, заданного матрицей R_B, дана на рис. 3 (незаполненные клетки - элементы D). Здесь m_1 и m_2 - разряды суммируемых чисел, l_1 -

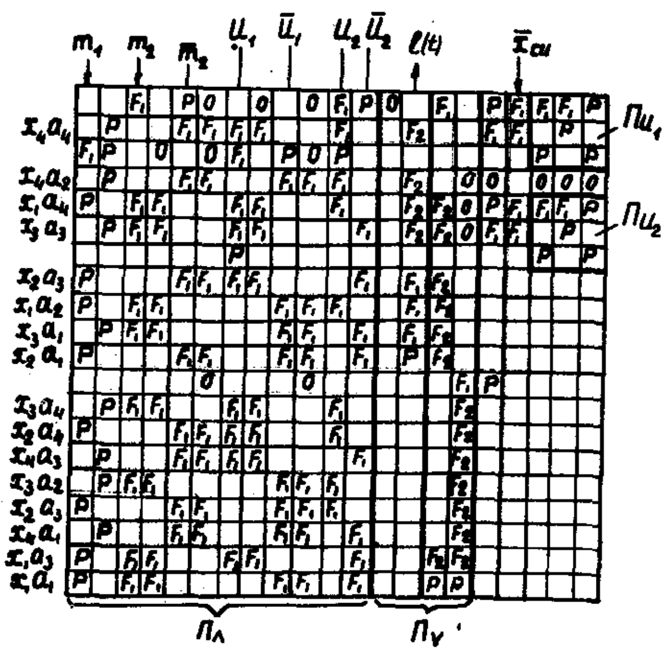

Рис. 3.

- сигнал выдачи суммы в текущем разряде, а кодирование входов и выходов автомата [5] (кодирование состояний дано на матрице R_B) следующее:

$$x_1 - \bar m_1 \bar m_2,\quad x_3 - m_1 \bar m_2,\quad y_1 - l_1,$$

$$x_2 - \bar m_1 m_2,\quad x_4 - m_1 m_2,\quad y_2 - l_1.$$

2.2. Оценим сложность программы реализации автомата в ВС. Под сложностью $L(R)$ будем понимать произведение числа занятых столбцов элементов ВС. Для указанного класса ВС и изложенного метода получения программы реализации автомата $\hat A$ в ВС легко получить непосредственно по матрице соединений R автомата $\hat A$ следующую общую оценку сложности:

$$L(R)=\left(\sum_{j=1}^{n} N_j + s + C_1\right)\left[(c+1)s+ck+p+C_2\right],$$

где N_j - число полных состояний в F_j; c - число столбцов элементов ВС в программе $П_\wedge$, приходящихся на одну входную или внутреннюю переменные; C_1 и C_2 - константы, зависящие от типа ВС и отражающие затрату элементов ВС на реализацию $П_{u_i}$, на синхронизацию и т.д.

Для заданного типа ВС (рис. I) и для автомата B получим

$$c=4,\ C_1=2,\ C_2=2,\ N_1=N_2=N_3=N_4=4,\ s=2,\ k=2;\quad L(R_B)=420.$$

3. Оптимизация программ абстрактных автоматов в ВС

Для изложенного метода получения программы реализации абстрактного автомата в ВС поставим задачу сокращения числа строк в программе $П_\wedge$ без непосредственной минимизации функций (I) и (4). Подобная задача была поставлена и решена в [8] для случаев реализации асинхронных и синхронных автоматов в криотронной ВС. Здесь рассматриваются некоторые алгоритмы оптимизации кодирования входов, выходов и состояний, позволяющие приближенно решить поставленную задачу при синтезе синхронных автоматов в универсальной ВС.

3.I. Из изложенного метода получения функций (I) и (4) и метода их реализации в ВС вытекает

ПРЕДЛОЖЕНИЕ I. Функционирование автомата Мили в ВС не изменится, если из программы $П_\wedge$ исключить все строки, соответствующие полным состояниям $x_\lambda a_j$ отмеченной ФП $F_{i,r}$ такой, что состояние a_i и выход y_r кодируются нулевыми наборами соответствующих переменных.

Отсюда следует, что в зависимости от выбора внутреннего состояния a_i и выхода y_r для кодирования нулевыми кодами из программы Π_λ можно исключить различное число строк. Сформулируем простой алгоритм, позволяющий найти внутреннее состояние a_i и выход y_r, кодируемые нулевыми наборами, для которых из программы Π_λ можно исключить максимальное число строк.

I°. Для каждой отмеченной ФП $F_{i,r}$ подсчитываем число полных состояний, входящих в $F_{i,r}$.

2°. Выбираем и отмечаем максимальное число $t^m_{i,r}$.

3°. Состояние a_i и выход y_r кодируем нулевыми наборами соответствующих переменных.

В том случае, если реализуемый в ВС автомат является автоматом Мура, алгоритм еще более упрощается. Необходимо лишь найти F_i с максимальным числом N^m_i и закодировать состояние a_i и выход y_r, отмечающий состояние a_i, нулевыми кодами. Очевидно, что приведенный метод исключения строк из программы Π_λ с некоторыми изменениями можно использовать и тогда, когда кодирование либо состояний, либо выходов уже задано.

Для заданного автомата B нетрудно найти $N^m_2 = N^m_3 = 6$. Поскольку выход y_1 уже закодирован нулевым кодом и y_1 отмечает состояние a_2, то максимального исключения числа строк в программе можно достичь, лишь закодировав состояние a_2 нулевым кодом, причем $t^m_{i,r} = t^m_{21} = N^m_{21} = 6$.

3.2. Рассмотрим метод дальнейшего сокращения числа строк в программе Π_λ. Для этого отмеченную ФП $F_{i,r}$ представим следующим образом:

$$F_{i,r} = (\bigvee_\lambda F^{x_\lambda}_{i,r}) \vee (\bigvee_j F^{a_j}_{i,r}),$$

где подфункция $F^{x_\lambda}_{i,r}$ – та часть отмеченной $F_{i,r}$, которая содержит только полные состояния с одинаковыми входными сигналами, равными x_λ, а подфункция $F^{a_j}_{i,r}$ – та часть отмеченной ФП, которая содержит только полные состояния с одинаковыми внутренними состояниями, равными a_j. Заметим, что одно и то же полное состояние может входить сразу в обе подфункции отмеченной ФП. Эти подфункции нетрудно получить непосредственно по матрице R следующим образом. Если в i-м столбце имеется несколько одинаковых элементов $x_\lambda (y_r)$ в строках $j_1, j_2, \ldots, j_k$, то можно записать

$$F^{x_\lambda}_{i,r} = x_\lambda a_{j_1} \vee x_\lambda a_{j_2} \vee \ldots \vee x_\lambda a_{j_k}.$$

Следовательно, в подфункцию $F^{x_\lambda}_{i,r}$ входят только те полные состояния, которые вызывают переход автомата $\hat{A}$ в одно и то же состояние a_i; отмечены одним и тем же выходом y_r; содержат одинаковые входы x_λ. Подфункция $F^{a_j}_{i,r}$ содержит только те полные состояния, которые вызывают переход в состояние a_i; отмечены выходом y_r; содержат одинаковые внутренние состояния a_j. Такая подфункция включает в себя более чем одно полное состояние, если на пересечении j-й строки матрицы R и i-го столбца имеется два или более элементов $x_{\lambda_1}(y_r), \ldots, x_{\lambda_k}(y_r)$, соединенных знаком дизъюнкции.

Из описанного выше метода реализации автомата в ВС следует, что если полное состояние $x_\lambda a_j$ из отмеченной ФП $F_{i,r}$ одновременно содержится в ОФП $u_i(t+1)$ и в ОФВ $\ell_h(t)$, то и все остальные полные состояния из $F_{i,r}$ входят в эти ОФП и ОФВ. Отсюда можно заключить, что любые эквивалентные преобразования функции $F_{i,r}$ не нарушают функционирование автомата в ВС, определяемое выражениями (I) и (4). В частности, если в полных состояниях $x_{\lambda_1} a_{j_1}$ и $x_{\lambda_2} a_{j_2}$ из $F^{x_\lambda}_{i,r}$ внутренние состояния a_{j_1} и a_{j_2} закодировать соседними кодами, то указанные полные состояния в $F^{x_\lambda}_{i,r}$ можно заменить одним – $x_\lambda a_{j_1,j_2}$. Внутреннее состояние a_{j_1,j_2}, которое назовем обобщенным состоянием, получено из a_{j_1} (либо из a_{j_2}) удалением переменной u_i, на которую отличаются коды внутренних состояний a_{j_1} и a_{j_2}, называемых образующими состояниями. Аналогично определяется понятие обобщенного входа x_{λ_1,λ_2}, полученного из образующих входов x_{λ_1} и x_{λ_2}, пары полных состояний $x_{\lambda_1}a_{j_1}$ и $x_{\lambda_2}a_{j_2}$ из $F_{i,r}^{a_j}$. Два различных обобщенных состояния a_{j_1,j_2} и a_{j_3,j_4} (назовем их обобщенными состояниями I ранга) могут служить образующими состояниями для обобщенного состояния a_{j_1}, a_{j_2,j_3,j_4} 2 ранга. Аналогично определяются обобщенные состояния и входы 3,4 и т.д. рангов. Нетрудно видеть, что образованию в подфункции $F_{i,r}^{x_\lambda}$ одного обобщенного состояния соответствует уменьшение на единицу числа строк в программе Π_λ.

Символом β_{j_1,j_2} обозначим число различных подфункций $F_{i,r}^{x_\lambda}$ таких, что каждая из них содержит одну и ту же пару полных состояний $x_\lambda a_{j_1}$ и $x_\lambda a_{j_2}$, $j_1 \neq j_2$. Таким образом, число β_{j_1,j_2} тем больше, чем большее число столбцов матрицы R будут иметь в j_1 и j_2 – строках одинаковые элементы $x_\lambda(y_r)$ $\lambda = 1,2,\ldots$ Потенциальное число β_{j_1,j_2} равно числу сокращенных строк в программе Π_λ, если состояния a_{j_1} и a_{j_2} закодировать соседними кодами и затем заменить их одним обобщенным состоянием в тех подфункциях $F_{i,r}^{x_\lambda}$, в которые они входят совместно. Аналогично для подфункции $F_{i,r}^{a_j}$ определяется число $\gamma_{\lambda_1,\lambda_2}$.

ПРЕДЛОЖЕНИЕ 2. Необходимым и достаточным условием сокращения числа строк в программе Π_λ полностью определенного абстрактного автомата Мили в ВС является выполнение одного из следующих условий:

1) существует β_{j_1,j_2} $(j_1 j_2 \in J = \{1,2,\ldots,n\})$ такое, что $\beta_{j_1,j_2} > 0$,

2) существует $\gamma_{\lambda_1,\lambda_2}$, $(\lambda_1 \lambda_2 \in \Lambda = \{1,2,\ldots,v\})$ такое, что $\gamma_{\lambda_1,\lambda_2} > 0$.

Доказательство в силу его простоты опускается.

Условимся считать, что с помощью обобщенного состояния a_{j_1,j_2} равномощные упорядоченные множества состояний A_1 и A_2 делаются частично соответствующими, если они упорядочиваются так, что образующие состояния $a_{j_1} \in A_1$ и $a_{j_2} \in A_2$ являются i-ми элементами множеств A_1 и A_2. Если же подобное соответствие между A_1 и A_2 устанавливается с помощью $|A_1|$ обобщенных состояний, назовем их полностью соответствующими. В общем случае будем говорить, что между A_1 и A_2 просто установлено соответствие, а образующие состояния a_{j_1} и a_{j_2} назовем соответственной парой состояний.

Исходя из предложения 2 и данных определений, сформулируем алгоритм кодирования внутренних состояний автомата Мили, для которого $n = 2^S$. Алгоритм основан на кодировании соседними кодами тех образующих состояний, которым соответствуют обобщенные состояния a_{j_1,j_2} с максимальными числами β_{j_1,j_2}, что, в свою очередь, приводит к максимальному или близкому к нему сокращению числа строк в программе Π_λ. На первом цикле этого алгоритма формируются пары состояний, на втором – четверки, на третьем – восьмерки состояний и т.д. На S – м цикле получается полное упорядоченное множество 2^S состояний. Этому множеству состояний ставится в соответствие упорядоченный набор двоичных кодов, соответствующий порядку следования этих кодов на матрице соединений. В результате такого кодирования каждое состояние a_j может служить образующим состоянием для S различных обобщенных состояний, так как для кода состояния a_j имеется всего S различных соседних кодов.

1°. Формируем подфункции $F_{i,r}^{x_\lambda}$ такие, что каждая из них содержит не менее двух различных полных состояний. Переход к 2°.

2°. Подсчитываем β_{i_1,j_1} для каждой пары внутренних состояний a_{i_1} и a_{j_1} из полных состояний $x_{\lambda_1}a_{i_1}$ и $x_{\lambda_2}a_{j_1}$, входящих в каждую сформированную в п.1° подфункцию $F_{i,r}^{x_\lambda}$, i, $j \in J = \{1,2,\ldots,n\}$, и формируем упорядоченное множество $B^{\delta-1}$ *) $= \{\beta_{i_1,j_1},\ldots,\beta_{i_h,j_h}\}$ по отношению порядка $\beta_{i_1j_1} \geqslant \beta_{i_2j_2} \geqslant \ldots \geqslant \beta_{i_h,j_h}$, $\beta_{i,j} \neq 0$. Формируем тривиальное разбиение $\rho^0 =$

*) δ – номер цикла алгоритма

$=\{\rho_1^0,\dots,\rho_n^0\}$, где $\rho_1^0=\{a_1\},\dots,\rho_n^0=\{a_n\}$. Если $B^0\neq\emptyset$, переход к 3^0, в противном случае – переход к 7^0.

3^0. Из блоков $\rho_1^{\delta-1},\dots,\rho_{t_{\delta-1}}^{\delta-1}$, $t_\delta=\frac{n}{2^\delta}$, разбиения $\rho^{\delta-1}$ формируем, начиная с ρ_1^δ , блоки разбиения $\rho^\delta=\{\rho_1^\delta,\dots,\rho_{t_\delta}^\delta\}$, где $\rho_i^\delta=\{\rho_{i_1}^{\delta-1},\rho_{i_2}^{\delta-1}\}$, следующим образом: ищем два блока $\rho_{i_1}^{\delta-1}$ и $\rho_{i_2}^{\delta-1}$, между множествами состояний которых можно установить такое частичное или полное соответствие, при котором для соответственных пар состояний можно записать

$$\sum_{\eta=1}^{\delta}\beta_{i_\eta j_\eta}=max,\quad \beta_{i_\eta,j_\eta}\in B^{\delta-1},$$

и которое не нарушает соответствия сформированных блоков $\rho_{i_1}^{\Delta-1}$, $\rho_{i_2}^{\Delta-1}\in\rho_i^\Delta$, $\Delta=1,2,\dots,\delta-1$, установленного отмеченными обобщенными состояниями. Формируем блок $\rho_1^\delta=\{\rho_{i_1}^{\delta-1},\rho_{i_2}^{\delta-1}\}$,блоки $\rho_{i_1}^{\delta-1}$ и $\rho_{i_2}^{\delta-1}$ считаем соответствующими, а обобщенные состояния $a_{i_\eta j_\eta}$, которые устанавливают соответствие этих блоков, считаем сформированными и отмечаем их. Аналогично формируем остальные блоки разбиения ρ^δ . Отмечаем числа β_{i_η,j_η} . Если $\delta=s$, переход к 6^0, в противном случае – к 4^0.

4^0. Формируем B^δ , исключив из $B^{\delta-1}$ все отмеченные числа $\beta_{i,j}$. Если $B^\delta=\emptyset$, переход к 5^0, в противном случае – к 3^0.

5^0. Формируем разбиения $\rho^{\delta+1}$, $\rho^{\delta+2},\dots,\rho^s$, где

$$\rho^{\delta+w}=\{\rho_1^{\delta+w},\dots,\rho_{t_{\delta+w}}^{\delta+w}\},\ \rho_i^{\delta+w}=\{\rho_{i_1}^{\delta+w},\rho_{i_2}^{\delta+w-1}\},\quad w=1,2,\dots,(s-\delta),$$

и где соответствие блоков $\rho_i^{\delta+w}$ устанавливается произвольно. Переход к 6^0.

6^0. Множеству состояний $\rho_1^s=\{\rho_1^{s-1},\rho_2^{s-1}\}$ с соответствующими блоками ρ_1^{s-1} и ρ_2^{s-1} ставим во взаимно однозначное соответствие упорядоченное множество двоичных наборов, соответствующих десятичным числам 2^s -I, 2^s -2,... I,0.

7^0. Алгоритм не применим к данному автомату.

Для $n<2^s$ используется этот же алгоритм, если предварительно добавить к исходной матрице соединений 2^s-n нулевых строк и столбцов. Что касается неполностью определенных пере-

ходов, то вопрос об их использовании возникает только лишь при получении обобщенного состояния из 2-х, 4-х и более обобщенных состояний,то есть обобщенных состояний 2,3 и т.д.рангов (что соответствует замене четырех, восьми и более строк элементов ВС одной строкой),и подробно в этой статье не рассматривается. Отметим только, что данный алгоритм позволяет производить и такое сокращение числа строк.

Дальнейшее сокращение числа строк в программе Π_A можно получить применением алгоритма кодирования входов, который в целом совпадает с приведенным и поэтому отдельно не приводится.

После кодирования состояний и входов автомата изменяем функции (I) и (4) следующим образом: каждую пару полных состояний $x_{\lambda_1}a_{j_1}$ и $x_{\lambda_2}a_{j_2}$, содержащую образующие состояния a_{j_1} и a_{j_2} отмеченного обобщенного состояния a_{j_1,j_2} (либо образующие входы x_{λ_1} и x_{λ_2} отмеченного обобщенного входа x_{λ_1,λ_2}), заменим одним полным состоянием $x_\lambda a_{j_1,j_2}$ (либо $x_{\lambda_1,\lambda_2}a_{j_1}$). Далее обобщенные состояния и входы I ранга заменяем там, где возможно (то есть в классе отмеченных ФП),обобщенными состояниями и входами более высоких рангов и т.д. до тех пор,пока эти возможности,определяемые найденным вариантом кодирования,не будут исчерпаны полностью.

Отметим, что присвоение нулевых кодов необходимо выполнять после нахождения всех отмеченных обобщенных состояний и входов, так как измененным функциям (I) и (4) соответствуют новые измененные числа $t'_{j,r}$. Найдя $t'^{m}_{j,r}$, переупорядочиваем разбиения ρ_1^δ так, чтобы состояние a_j имело на матрице соединений нулевой код и чтобы соответствие блоков разбиений ρ_1^δ , установленное отмеченными обобщенными состояниями, не нарушалось.

При оптимизации синтеза в ВС абстрактных автоматов Мура имеется одна особенность. Поскольку в таких автоматах все полные состояния из F_i отмечены одним и тем же выходом y_r ,то разложение функции F_i имеет вид

$$F_i=(\vee_\lambda F_i^{x_\lambda})\vee(\vee_j F_i^{a_j})=(\vee_\lambda F_{i,r}^{x_\lambda})\vee(\vee_j F_{i,r}^{a_j}).$$

В остальном алгоритм нахождения обобщенных состояний и входов автоматов Мили и Мура, по существу, совпадают.

Если приведенный алгоритм применить к автомату B, то с учетом нулевого кодирования получим

$$B^0=\{\beta_{1,3}\ \beta_{2,4}\},\quad B^1=\emptyset,\ \beta_{1,3}=\beta_{2,4}=4,$$

$$\rho_2^1=\{\rho_1^1,\rho_2^1\}=\{\{\rho_1^1\rho_3^1\}\{\rho_4^0\rho_2^0\}\}=\{\{a_1a_3\}\{a_4a_2\}\}.$$

Найденному порядку внутренних состояний на матрице R_1 соответствуют коды a_1 -II, a_3 -I0, a_4 -0I и a_2 -00. Нетрудно подсчитать, что в результате оптимизации число функциональных строк в программе Π_λ сократилось с I6 до 5. Кроме того, в измененных функциях (I), (2) и (4) автомата B отмеченными обобщенными состояниями $a_{1,3}$ и $a_{2,4}$ полностью устраняется переменная u_2, поэтому реализовать функцию $u_2(t+1)$ и программу Π_{u_2} в ВС нет необходимости (рис. 4).

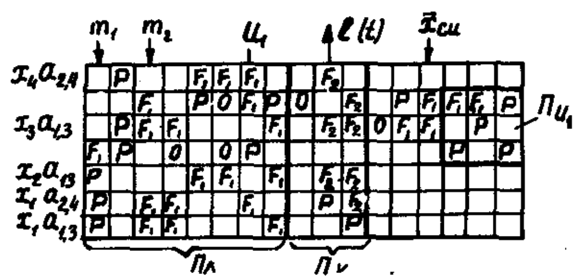

Рис. 4.

В заключение отметим, что следующие преимущества предложенного метода: сравнительная простота и эффективность алгоритмов кодирования, легкость получения программы настройки ВС, упрощение этапа получения и минимизации функций возбуждения элементарных автоматов и функций выходов – делают возможным применение его для синтеза и оптимизации абстрактных автоматов в широком классе вычислительных сред и однородных структур. В ряде случаев сложность получаемых при этом программ автоматов в ВС получается меньше сложности реализаций, полученных по минимизированным функциям возбуждения элементарных автоматов и функциям выходов [4]. Кроме того, поскольку критерием оптимизации в данном случае является минимальное число строк элементов ВС и, следовательно, минимальное число членов ДНФ в ОФП и ОФВ, то этот же метод можно использовать для оптимизации структурного синтеза синхронных автоматов на обычных структурных элементах [5].

Литература

I. ЕВРЕИНОВ Э.В., КОСАРЕВ Ю.Г. Однородные универсальные вычислительные системы высокой производительности. Новосибирск, "Наука", I967.

2. МЕЛИХОВ А.Н, ДВОРЯНЦЕВ Ю.А. Разложение графов и конечных автоматов относительно операции умножения. – "Кибернетика", I965, № 3.

3. ТОПОЛЬСКИЙ Н.Г. Оптимальное кодирование входных сигналов конечного автомата. – "Кибернетика и моделирование систем", Ростов-на-Дону, 1968.

4. МЕЛИХОВ А.Н., ТОПОЛЬСКИЙ Н.Г. Об одном методе синтеза автоматов в вычислительной среде. – "Цифровые модели и интегрирующие структуры", Таганрог, I970.

5. ВАВИЛОВ Е.Н., ПОРТНОЙ Г.П. Синтез схем электронных цифровых машин, М., "Сов.радио", I963.

6. БАНДМАН О.Л. Реализация автоматов в криотронной вычислительной среде по заданному графу состояний. – "Вычислительные системы", Новосибирск, "Наука" СО, I969, вып. 33.

7. CURTIS H.A. Systematic procedures for realizing synchronous sequential machines using flip-flop memory. Pt.I, IEEE Transactions on Computers, vol.C-18, December 1969, N 12.

8. БАНДМАН О.Л. Методы реализации автоматов в криотронной вычислительной среде. – "Цифровые модели и интегрирующие структуры", Таганрог, I970.

Поступила в редакцию
I6. IУ. I97I г.