ИЗВЕСТИЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Том II, № 2 РАДИОФИЗИКА 1959

ОПЕРАТОРНЫЙ МЕТОД СИНТЕЗА АЛГОРИТМИЧЕСКИХ СИСТЕМ

А. Д. Закревский

Рассматриваются особенности последовательного преобразования информации и определяются требования к автоматам, преобразующим информацию. Предлагается метод синтеза цифровых автоматов, представляющий формальный переход от условий работы, задаваемых в виде алгоритмического разложения оператора преобразования, к структуре автомата.

1. Задачу автоматизации процессов преобразования информации, являющуюся одной из центральных задач кибернетики, можно сформулировать следующим образом.

Даны два множества X и Y; первое представляет набор подлежащих решению задач с исходными данными, а второе является множеством, среди элементов которого находятся ответы на эти задачи. Требуется синтезировать устройство, автоматически производящее отображение X в Y так, чтобы каждой задаче был поставлен в соответствие правильный ответ.

Ограничимся рассмотрением цифровых автоматов, оперирующих с двоичными переменными, допустив, что они могут давать удовлетворительные решения, и будем поэтому рассматривать лишь конечные множества.

На входе автомата должна быть представлена информация о выборе элемента x из X, на выходе — y из Y.

Для представления любого из N_x элементов множества X требуется

$$n_x = \check{E} \log_2 N_x \tag{1}$$

двоичных символов, где $\check{E}$ является символом ближайшего сверху натурального числа.

Аналогично

$$n_y = \check{E} \log_2 N_y. \tag{2}$$

Простейшим с теоретической точки зрения является однотактный автомат, воспринимающий сразу всю исходную информацию, т. е. n_x двоичных символов, представляемых пространственно разделенными физическими переменными. Ответ выдается автоматом значениями n_y двоичных физических переменных на выходе сразу же после прекращения переходных процессов, вызываемых сменой значений входных переменных.

Введем понятие входного сечения автомата $S_{вх}$, определив его через максимальное количество информации $J_{вх}$, которое можно зафиксировать на входе автомата; аналогично определим выходное сечение $S_{вых}$.

Преобразование информации, как правило, связано с ее потерями, поэтому обычно $S_{вых} < S_{вх}$; минимальные потери информации при однотактном преобразовании равны разности входного и выходного сечений.

Очевидно, что для рассматриваемого автомата

$$S_{вх} = J_{вх} = n_x; \qquad S_{вых} = J_{вых} = n_y. \tag{3}$$

Энтропия H_x и H_y множеств X и Y, определяемая через двоичный логарифм числа элементов множества, количественно совпадает с $S_{вх}$ и $S_{вых}$ при

$$N_x = 2^{n_x}; \qquad N_y = 2^{n_y}. \tag{4}$$

Введем символ a^0—символ действия, приводящего к нахождению y для задаваемого x, и назовем его оператором отображения X в Y; этим же символом будем обозначать производящий данное отображение автомат.

Существует множество A^0 из

$$N_{a^0} = N_y^{N_x} \tag{5}$$

операторов отображения X в Y; энтропия этого множества

$$H_{a^0} = \log_2(N_y^{N_x}), \tag{6}$$

а в случае выполнения равенств (4)

$$H_{a^0} = n_y 2^{n_x}. \tag{7}$$

Следовательно, информация об операторе a^0 из A^0 может быть представлена

$$n_{a^0} = n_y 2^{n_x} \tag{8}$$

двоичными символами. Эта величина будет служить количественной оценкой сложности оператора и соответствующего автомата, отображающего X в Y. Будем называть ее энтропией оператора (автомата), если об операторе известно лишь, что он отображает X в Y, но не известно каким способом.

Как видно из (8), сложность автомата является экспоненциальной функцией от входного сечения. Этот факт приводит к тому же выводу, что и результаты исследований Шеннона [1] и Лупанова [2, 3]: синтез автоматов с достаточно большими сечениями практически невозможен. Разумеется, имеются в виду такие автоматы, каждая из выходных переменных которых связана существенной функциональной зависимостью со всеми (или почти со всеми) входными переменными.

Поэтому при большом диапазоне задач, процесс решения которых должен быть автоматизирован, т. е. при большом числе элементов в множестве X, однотактный способ переработки информации трудно, если не невозможно, осуществить практически.

2. Однако хорошо известно, что большинство встречающихся на практике задач допускает последовательную, многотактную переработку информации по частям. При этом действие оператора a^0 оказывается возможным заменить эквивалентной по результатам последовательностью действий операторов a_i^1 из некоторого множества A^1, причем каждый из этих операторов перерабатывает гораздо меньшее количество информации; следовательно, синтез соответствующих автоматов a_i^1 значительно проще синтеза автомата a^0.

Порядок действия операторов из A^1 определяется как функция некоторых двоичных переменных ξ_j; этой функцией выражается существующая зависимость от исходной информации и от результатов ее переработки:

$$\tilde{a}^0(\xi_1, \xi_2, \ldots, \xi_j); \tag{9}$$

значениями функции являются фиксированные последовательности действий операторов a_i^1.

Будем говорить, что эта функция представляет алгоритмическое разложение оператора a^0 на систему операторов a_i^1.

Операторы a_i^1, в свою очередь, могут быть разложены на систему операторов a_j^2 из множества A^2 и т. д. В конечном счете мы получим многоступенчатую систему операторов, у которой операторы ступени r управляют операторами ступени $r+1$ $(r=0,1,\ldots, q-1)$—эти операторы назовем управляющими и присвоим им индекс ~. Непосредственно перерабатывают исходную информацию* лишь операторы последней ступени q.

Операторы непосредственной переработки информации, образующие множество A^q, могут, в частности, представлять: а) набор команд некоторой вычислительной машины, б) элементарные логические операции, в) функциональные свойства блоков вычислительных и управляющих устройств, г) функциональные свойства сложных систем накопления и преобразования информации: универсальных и специализированных вычислительных машин и управляющих автоматов и их совокупностей. Задача разложения исходного оператора a^0 на многоступенчатую систему операторов управления $\tilde{a}_s^r$ в случае а) практически решается методом операторного программирования, предложенным Ляпуновым [5]. Представляемая в коде вычислительной машины система операторов $\tilde{a}_s^r$ называется программой решения задачи (соответствующей оператору a^0) и заносится в запоминающее устройство машины.

Очевидно, что в случаях б), в), г), отличающихся от случая а) лишь уровнем, до которого происходит разложение оператора a^0, задача получения системы $\tilde{a}_s^r$ может быть решена также в рамках теории операторного программирования. Получаемая при этом система операторов $\tilde{a}_s^r$, являющаяся формой алгоритмического представления оператора a^0, принимается нами за исходную форму представления функциональных свойств системы переработки информации, метод синтеза которой рассматривается в следующих разделах статьи.

Следует подчеркнуть, что набор автоматов, реализующих операторы множества A^q, считается заданным, поэтому синтезируется собственно лишь схема управления этими автоматами. В случае б) метод дает, однако, полное решение задачи синтеза автомата из простейших логических элементов, в случае в) и г) проводится синтез системы организации взаимодействия отдельных блоков или целых вычислительных и управляющих устройств.

3. Известны различные формы представления алгоритмических разложений [5—10]. Мы будем пользоваться наиболее удобной для

---

* Последовательность совместных действий операторов всех ступеней может быть представлена как многократное действие оператора Ω, производящего последовательное преобразование информации J:

$$J_1 = \Omega(J_0);$$
$$J_2 = \Omega(J_1);$$
$$\cdots$$
$$J_z = \Omega(J_{z-1}),$$

где J_0 есть исходная информация о поставленной задаче, а в J_z содержится ответ на эту задачу. При каждом шаге перерабатывается лишь часть преобразуемой информации; существенной особенностью оператора Ω является ограниченность количества перерабатываемой за один шаг информации. С учетом этой важной для практики преобразования информации особенности подошли к теории алгоритмов Колмогоров и Успенский [4].

дальнейшего изложения матричной формой, отличающейся от формы, употреблявшейся Яновым и Цетлиным [8, 9, 11, 12], большей компактностью, удобством перехода к системе булевых функций при синтезе реализующей алгоритм схемы.

Способ получения этой формы рассмотрим на следующем примере. Пусть алгоритмическое разложение некоторого оператора $\tilde{a}_s^r$ исходной многоступенчатой системы операторов определяется как функция двоичных переменных ξ_1, ξ_2, ξ_3 (таблица 1)*.

Таблица 1

i	ξ_3, ξ_2, ξ_1	$\tilde{a}_s^r$
0	0 0 0	$\tilde{a}_5^{r+1}\left(\tilde{a}_2^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_3^{r+1}\right)$
1	0 0 1	$\tilde{a}_1^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_5^{r+1}$
2	0 1 1	$\tilde{a}_2^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_1^{r+1}\,\tilde{a}_5^{r+1}$
3	0 1 0	$\tilde{a}_5^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_5^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_2^{r+1}$
4	1 1 0	$\tilde{a}_1^{r+1}\,\tilde{a}_5^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_5^{r+1}$
5	1 1 1	$\tilde{a}_1^{r+1}\,\tilde{a}_2^{r+1}\left(\tilde{a}_3^{r+1}\right)$
6	1 0 1	$\tilde{a}_1^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_5^{r+1}$
7	1 0 0	$\tilde{a}_1^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_5^{r+1}$

Таблица 2

i \ k		$\tilde{a}_1^{r+1}$	$\tilde{a}_2^{r+1}$	$\tilde{a}_3^{r+1}$	$\tilde{a}_4^{r+1}$	$\tilde{a}_5^{r+1}$	$\tilde{a}_5^{r+1}$	
	0	1	2	3	4	5	6	7
0	5		4	2	3	2		7
1	1	2	3	5		7		7
2	2	5	3	1		7		7
3	5		7	2	6	4	3	7
4	1	5	3	6		2	7	7
5	1	2	3	3				7
6	1	4		5	3	7		7
7	1	2	3	4	5	7		7

Скобки являются символом циклически повторяющейся группы операторов, например,

$$\tilde{a}_5^{r+1}\left(\tilde{a}_2^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_3^{r+1}\right) \sim \tilde{a}_5^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_4^{r+1}\,\tilde{a}_3^{r+1}\,\tilde{a}_2^{r+1}\,\tilde{a}_4^{r+1}\ \ldots, \tag{10}$$

где $\sim$ является символом эквивалентности. Выход из цикла возможен лишь при смене комбинаций значений ξ_1, ξ_2, ξ_3.

Преобразуя таблицу 1 в матричную форму, получаем таблицу 2. Строки i матрицы соответствуют строкам i таблицы 1, каждый столбец k, за исключением нулевого и последнего, поставлен в соответствие с определенным оператором из множества $\tilde{A}^{r+1}$. Элемент матрицы a_{ik} представляет номер столбца j, соответствующего тому оператору, который должен действовать вслед за оператором, соответствующим столбцу k, если значения ξ_1, ξ_2, ξ_3 соответствуют строке i. В рассматриваемом примере столбцы $k = 1, 2, 3, 4, 5$ соответствуют операторам $\tilde{a}_k^{r+1}$. В строке 3 таблицы 1 представлен случай, когда после оператора

* Все двоичные коды, с помощью которых устанавливается 1 — 1 соответствие между комбинациями значений двоичных переменных и натуральными числами, для описываемого метода формально эквивалентны. Мы будем пользоваться симметричным кодом Грея исключительно из-за удобств визуальной минимизации булевой функции по ее матричной форме [13].

$\tilde{a}_5^{r+1}$ один раз следует $\tilde{a}_4^{r+1}$, а другой—$\tilde{a}_2^{r+1}$; чтобы представить на матрице оба эти чередования, мы поставили в соответствие с $\tilde{a}_5^{r+1}$ также столбец 6 (то же сделано в строке 4). Номер оператора, который должен действовать первым, проставляется в столбце 0; в столбце, соответствующем последнему действующему оператору, ставится 7—номер соответствующего окончанию разложения столбца, в который также ставится 7.

Так как элементы a_{ik} указывают переходы между операторами $\tilde{a}_j^{r+1}$, назовем (a_{ik}) матрицей переходов. Как видно из способа построения матрицы (a_{ik}), определяются не все ее элементы (в рассматриваемом примере не определены a_{01}, a_{06}, a_{14} и т. д.). При фиксированном i алгоритмическое разложение, представляемое матрицей, инвариантно относительно этих элементов. Будем иметь это в виду, говоря, что матрицы, имеющие различия в указанных неопределенных элементах, эквивалентны. Число эквивалентных матриц равно u^v, где v — число неопределенных элементов, а u — число столбцов матрицы. Матрица (a_{ik}), с которой мы будем оперировать в дальнейшем, может быть любой из эквивалентных матриц.

4. Будем считать, что алгоритмическое разложение оператора $\tilde{a}_s^r$ на систему операторов из $\tilde{A}^{r+1}$, представляемое матрицей (a_{ik}), реализуется автоматом $\tilde{a}_s^r$, если его состояния поставлены во взаимнооднозначное соответствие со столбцами матрицы k и порядок их смены целиком определяется матрицей переходов (a_{ik}).

Назовем такой автомат блоком алгоритмического разложения $\tilde{a}_s^r$ и рассмотрим метод его синтеза. Этот метод делает возможной замену сложного автомата a_s^r системой более простых автоматов a_j^{r+1} и блоком алгоритмического разложения $\tilde{a}_s^r$, функции которого сводятся к управлению автоматами a_j^{r+1}.

С целью облегчить дальнейшее изложение введем следующие функции. Функция

$$\alpha_j(a) \tag{11}$$

представляет значение j-го разряда двоичного кода натурального числа a. Функция

$$\beta' \tag{12}$$

представляет значение переменной β, задержанное на промежуток времени Δt. Введение подобной функции необходимо для представления поведения автоматов, содержащих линии задержки (с временем задержки Δt); нетрудно доказать линейность оператора задержки [14]:

$$\varphi'(\gamma_1, \gamma_2, \ldots) = \varphi(\gamma_1', \gamma_2', \ldots), \tag{13}$$

где φ — произвольная булева функция. Функция

$$\beta^\gamma \tag{14}$$

представляет β при $\gamma=1$ и $\bar{\beta}$ при $\gamma=0$.

Здесь, как и в дальнейшем, буквами греческого алфавита обозначаются двоичные переменные (и двоичные функции).

Из правила построения матрицы (a_{ik}) (см. таблицу 1) следует:

$$\xi_j = \alpha_j(i). \tag{15}$$

Аналогично введем переменные

$$\eta_j = \alpha_j(k). \tag{16}$$

Выходные переменные схемы ν_j являются функциями состояния

$$\nu_j = (k = j), \tag{17}$$

т. е. принимают значение 1, если схема находится в состоянии j, принимая в противном случае значение 0.

Из (14) и (17) следует выражение

$$\nu_j = (k = j) = \bigwedge_i \eta_i^{\alpha_i(j)}, \tag{18}$$

которое может быть принято за структурную формулу выхода.

Элемент матрицы a_{ik} представляет номер состояния, следующего за состоянием k при фиксированном i, поэтому состояние автомата определяется элементом матрицы $a_{ik'}$ для предыдущего состояния. Так как i фиксировано,

$$k = a'_{ik}, \tag{19}$$

это уравнение эквивалентно системе двоичных функций, получаемых из (16) и (19):

$$\eta_j = \alpha_j(a'_{ik}), \tag{20}$$

эквивалентных, в свою очередь, системе булевых функций

$$\eta_j = \varphi'_j(\xi_1, \xi_2, \ldots; \eta_1, \eta_2, \ldots), \tag{21}$$

аргументы которых связаны с индексами i и k элемента матрицы a_{ik} соотношениями (15) и (16).

Матричные изображения функций η_j получаются, таким образом, разложением матрицы (a_{ik}) на матрицы, элементами которых являются двоичные переменные — значения j-го разряда двоичного кода Грея чисел a_{ik}. Это разложение легко производится таблично, в чем можно убедиться на рассматриваемом примере.

Таблица 3

$i \backslash k$	0	1	2	3	4	5	6	7
0	111		110	011	010	011		100
1	001	011	010	111		100		100
2	011	111	010	001		100		100
3	111		100	011	101	110	010	100
4	001	111	010	101		011	100	100
5	001	011	010	010				100
6	001	110		111	010	100		100
7	001	011	010	110	111	100		100

η_3 η_2 η_1

ξ_3 ξ_2 ξ_1

В таблице 3 представлена та же матрица (a_{ik}), что и в таблице 2, но элементы матрицы записаны в двоичном коде Грея. Кодирование строк и столбцов значениями переменных ξ_1, ξ_2, ξ_3 и η_1, η_2, η_3 отражено расположенными сверху и слева матрицы линиями, на которых жирной чертой указываются те строки (или столбцы), в кодах которых соответствующая переменная принимает значение 1.

Выделяя первые разряды элементов матрицы, получаем матрицу, представляющую функцию η_1 (таблица 4), определенную с точностью до пустых элементов.

Таблица 4

$i \backslash k$	0	1	2	3	4	5	6	7
0	1		0	1	0	1		0
1	1	1	0	1		0		0
2	1	1	0	1		0		0
3	1		0	1	1	0	0	0
4	1	1	0	1		1	0	0
5	1	1	0	0				0
6	1	0		1	0	0		0
7	1	1	0	0	1	0		0

η_3, η_2, η_1; ξ_3, ξ_2, ξ_1

Совокупность единичных значений η_1 образует изображение, являющееся полным прообразом единичного значения η_1 на множестве комбинаций значений ее аргументов. Кодирование строк и столбцов матрицы в симметричном коде Грея облегчает визуальный поиск минимальной алгебраической формы, представляемой матрицей булевой функции [13]. Обозначая подмножество элементов матрицы, лежащих против жирной черты ξ_i или η_i, через ξ_i или η_i, а их дополнения — через $\overline{\xi_i}$ или $\overline{\eta_i}$ и используя соответствия операций в алгебре Буля и в теории множеств: сложения — объединению, умножения — пересечению, отрицания — дополнению, получим следующую алгебраическую форму*:

$$\overline{\eta_2}\,\overline{\eta_3}\,(\overline{\eta_1}+\overline{\xi_1}+\xi_2+\overline{\xi_3})+\eta_2(\overline{\eta_1}\,(\eta_3\,(\xi_1\overline{\xi_2}+\overline{\xi_3})+\overline{\xi_1}\,(\xi_2+\xi_3\eta_3))+$$
$$+\,\eta_3\overline{\xi_1}\,(\xi_2\,\xi_3+\overline{\xi_2}\,\overline{\xi_3}\,\eta_1)), \tag{22}$$

которую можно рассматривать, как структурную формулу реализующей функцию η_1 схемы, если имеются в наличии элементы, реализующие операции сложения, умножения и инверсии.

Определяемое по (19) очередное значение k используется затем в качестве аргумента при определении следующего значения k; однако, если не все η_j (21) будут принимать новые значения сразу, k может принять ложное значение, следствием чего явится непредусмотренная последовательность состояний схемы.

Надежность работы схемы обеспечим, введя переменную синхронизации Σ, принимающую значение 1 в промежутках времени длительностью τ, с периодом повторения T, причем должно выполняться условие

$$\tau < \Delta t < T. \tag{23}$$

Из (19), (20) получим соответственно:

$$k = a'_{ik}\,\Sigma + k\overline{\Sigma}; \tag{24}$$

$$\eta_j = \alpha_j(a'_{ik})\,\Sigma + \eta_j\,\overline{\Sigma}. \tag{25}$$

Описанный метод позволяет совершить переход от заданных таблично или матрично операторов $\tilde{a}^r_s$ к системам булевых функций, определяющих собственные переменные η_j (наборы значений η_j служат кодами состояний) автоматов, функциональные свойства которых представляются операторами $\tilde{a}^r_s$. Аналогичная задача рассмотрена в работах Трахтенброта [7], Цетлина [11, 12] и Хуффмана [15]; предлагаемый здесь метод решения задачи близок к матричному методу Цетлина, отличаясь от него а) видом исходной матрицы переходов (матрицы состояний у Цетлина), в частности, большей компактностью, б) простотой разложения матрицы переходов на систему булевых матриц, в) удобством представления неполностью определенных матриц переходов, дающих при разложении неполностью определенные булевы функции (перенос неопределенности в булевы функции обеспечи-

* Например, первый член (22) получен при учете значений, заданных в первых двух столбцах; неопределенные значения в данном случае оказалось удобным принять за единичные.

вает возможность их оптимального, в смысле получения простейшей структуры синтезируемой схемы, доопределения [16, 17]).

Задача перехода от системы булевых функций к структурным формулам реализующей ее схемы (задача последнего этапа синтеза) здесь не рассматривается. Различные методы решения этой задачи излагаются в работах [1, 3, 15, 18–24].

5. Рассмотрим теперь метод объединения полученных автоматов (блоков-операторов) $\tilde{a}_s^r$ в единую управляющую систему, реализующую исходный оператор a^0. Структура связей автоматов $\tilde{a}_s^r$ должна соответствовать структуре алгоритмического разложения оператора a^0 на многоступенчатую систему операторов $\tilde{a}_s^r$: так, например, блок-оператор $\tilde{a}_s^k$ должен управлять (с помощью переменных $\nu_i|_s^k$) работой некоторых блоков следующей, $k+1$-ой ступени и, в свою очередь, управляться блоками предыдущей, $k-1$-ой ступени. Автоматы последней ступени должны непосредственно перерабатывать информацию, извлекаемую из внешнего по отношению к управляющей системе источника; запуск всей системы будет осуществляться подачей сигнала извне на блок $\tilde{a}^0$, стоящий во главе управления (например, посредством нажатия кнопки пуска).

Переходя к рассмотрению системы блоков-операторов $\tilde{a}_s^r$, присвоим логическим переменным выражения (24) индексы r и s:

$$k\,|_s^r = a'_{ik}|_s^r\,\Sigma|^r + k|_s^r\,\overline{\Sigma}|^r, \tag{26}$$

считая, что переменная синхронизация $\Sigma|^r$ является общей для всех операторов одной ступени. Выражение (26) можно записать более компактно, вынося индексы

$$|\,k = a'_{ik}\,\Sigma + k\,\overline{\Sigma}\,|_s^r. \tag{27}$$

Выходные переменные блока получим из (18)

$$|\,\nu_j = (k = j) = \bigwedge_i \eta_i^{\alpha_i(j)}|_s^r. \tag{28}$$

Рассмотрим связь блока с предыдущей ступенью. Пусть $\nu_{j\sim s}|_c^{r-1}$ — выходная переменная блока $\tilde{a}_c^{r-1}$, которая принимает значение 1, когда блок находится в состоянии j, поставленном в соответствие оператору $\tilde{a}_s^r$ при построении матрицы переходов $(a_{ik})|_c^{r-1}$. Управление блоком $\tilde{a}_s^r$ со стороны r—1-ой ступени операторов осуществляется переменной

$$\Theta\,|_s^r = V_j V_c(\nu_{j\sim s}|_c^{r-1}). \tag{29}$$

Чтобы запустить блок $\tilde{a}_s^r$, заставить его производить заданное разложение, нужно перевести его из „холостого“ последнего состояния m в нулевое, предоставив его затем самому себе. Блок $\tilde{a}_s^r$ должен запускаться в тот момент, когда $\Theta\,|_s^r$ принимает значение 1. С учетом этого получим из (27):

$$|\,k = (a'_{ik}\,\Sigma + k\overline{\Sigma})\,\overline{\Theta\nu_m\tau}\,|_s^r, \tag{30}$$

где

$$\nu_m = (k = m) \tag{31}$$

принимает значение 1 при „холостом“ состоянии блока, а

$$\tau = \Sigma\,|^z \tag{32}$$

является переменной синхронизации последней ступени z.

Признаком того, что работа блока разложения $\tilde{a}_s^r$ подходит к концу, т. е. действует последний в реализуемой последовательности оператор из $\tilde{A}^{r+1}$, является принятие значения 1 переменной

$$|\,\sigma = (a_{ik} = m)\,(k \neq m)\,|_s^r. \tag{33}$$

По цепи обратной связи это значение переменной должно быть подано на $r-1$-ую ступень и послужить сигналом запуска очередного блока r-ой ступени.

Так как в каждой ступени в любой момент времени действует не более одного блока, переменная обратной связи может быть представлена как

$$V_j(\sigma\,|_j^r). \tag{34}$$

Теперь уже нетрудно определить структуру схемы, выходная переменная которой представит $\Sigma\,|^{r-1}$ — переменную синхронизации блоков $r-1$-ой ступени. Эта переменная должна принимать единичное значение, когда

$$V_j(\sigma\,|_j^r) = 1 \tag{35}$$

и когда, кроме того,

$$\Sigma\,|^r = 1, \tag{36}$$

что свидетельствует об окончании работы очередного блока в r-ой ступени. Поэтому

$$\Sigma\,|^{r-1} = V_j\,(\sigma\,\Sigma\,|_j^r) \tag{37}$$

и, аналогично,

$$\Sigma\,|^r = V_j(\sigma\Sigma\,|_j^{r+1}). \tag{38}$$

Особенностью блоков крайних ступеней будет являться то, что переменная синхронизации последней ступени

$$\Sigma\,|^z = \tau \tag{39}$$

подается извне, от генератора синхронизующих импульсов, частота которого определяется техническими соображениями; а переменная запуска единственного блока первой ступени Θ^0 является вместе с тем переменной запуска всей алгоритмической системы и может быть задана кнопкой, нажатие которой запускает всю систему.

Формулы (27)—(39) при наличии элементов, реализующих употребляемые в формулах операции, могут быть приняты за структурные и вместе со всеми структурными формулами блоков-операторов представить детальное структурное описание синтезированной многоступенчатой системы управления.

В заключение подчеркнем, что изложенный метод, являясь по существу приложением операторного метода программирования к теории синтеза управляющих цифровых систем, обладает такими достоинствами операторного метода, как гибкость, обозримость функциональных свойств блоков и их совокупностей, возможность независимого синтеза блоков. Следует отметить, что эти качества часто влекут такой существенный недостаток, как неоправданная избыточность структуры.

В значительной своей части задача минимизации структуры управляющей системы может быть решена путем эквивалентных преобразований алгоритмического разложения (приводящих, в частности, к уменьшению числа ступеней) в рамках теории операторного программирования.

ЛИТЕРАТУРА

1. C. E. Shannon, Bell. Syst. Techn. J., **28**, 59 (1949).
2. О. Б. Лупанов, ДАН СССР, **103**, 561 (1955).
3. О. Б. Лупанов, Труды матем. ин-та им. В. А. Стеклова, **51**, 158 (1957).
4. А. Н. Колмогоров, В. А. Успенский, УМН, **13**, **3** (1958).
5. А. А. Ляпунов, Изв. высш. уч. зав.—Радиофизика, **1**, 1, 106 (1958).
6. А. А. Марков, Труды матем. ин-та им. В. А. Стеклова, **42**, 8 (1954).
7. Б. А. Трахтенброт, ДАН СССР, **118**, 646 (1958).
8. Ю. И. Янов, ДАН СССР, **113**, 28: (1957).
9. Ю. И. Янов, Проблемы кибернетики, вып. 1, 75, 1958.
10. А. П. Ершов, ДАН СССР, **118**, 427 (1958).
11. М. Л. Цетлин, ДАН СССР, **117**, 6 (1957).
12. М. Л. Цетлин, Проблемы кибернетики, вып. 1, 23, 1958.
13. А. Д. Закревский, Труды Сибирского физ.-техн ин-та, вып. 38 (1959).
14. Ю. Я. Базилевский, Вопросы теории математических машин, сб. 1, 1958.
15. D. A. Huffman, Information Theory, Third London Symposium, 1955.
16. В. Н. Рогинский, Автоматика и телемеханика, **15**, 206 (1954).
17. Ю. И. Журавлев, Труды матем. ин-та им. В. А. Стеклова, **51**, 143 (1958).
18. М. А. Гаврилов, Теория релейно-контактных схем, изд. АН СССР, М., 1950.
19. Синтез электронных вычислительных и управляющих схем (под ред. Айкена), ИЛ, М., 1954.
20. В. И. Шестаков, ЖТФ, **11**, 532 (1941).
21. Г. Н. Поваров, ДАН СССР, **100**, 909 (1955).
22. С. В. Яблонский, Труды III Всесоюзного матем. съезда, **3**, изд. АН СССР, М., 425, 1958
23. С. В. Яблонский, Труды матем. ин-та им. В. А. Стеклова, **51**, 5 (1958).
24. Б. А. Трахтенброт, ДАН СССР, **112**, 1005 (1957).

Томский государственный университет

Поступила в редакцию 23 октября 1958 г.