

АНАЛИЗ И СИНТЕЗ УПРАВЛЕНИЙ ОДНИМ АВТОМАТОМ С
ПЕРЕМЕННОЙ СТРУКТУРОЙ

Б.Г. Матиенко

В ряде важных для практики случаев требуется, чтобы в процессе работы измерительной информационной системы (ИИС) [1] изменялось априори принятое взаимодействие между различными частями системы, менялся алгоритм работы некоторых подсистем, наконец, менялось число и состав средств, осуществляющих сбор информации.

Ряд перечисленных выше задач может быть решен при наличии высоконадежных дискретных систем, обладающих большими возможностями в отношении автоматической перестройки своей структуры. Перспективным принципом проектирования таких систем является реализация их в виде однородных микроэлектронных структур (вычислительных сред) с настройкой [2,3]. В данной работе предпринята попытка исследования одной из возможных моделей дискретных систем, функционирование которых основано на принципе переменности структуры однородных вычислительных сред.

Рассматриваемый ниже автомат S^* (рис.1) считается состоящим из двух основных блоков S_v и T . Блок S_v

объявляется устройством с управляемой (переменной) структурой, а T - настроечным автоматом, в котором вырабатывается информация об организации структуры S_V . Последняя считается состоящей из конечного числа N однотипных управляемых логических элементов (УЛЭ), между которыми заданы жесткие связи.

Возможности S^* по переработке информации в данной работе ограничены функциями алгебры логики (ФАЛ) в дизъюнктивной (конъюнктивной) нормальной форме (д.н.ф., к.н.ф.) от двоичных переменных, являющихся входами S_V . Функции, конкретный вид которых априори может быть неизвестным, считаются каждый раз поступающими на входы T в виде закодированной определенным образом упорядоченной строчки (списка) буквенных символов. Считается, что функции реализуются в S_V последовательно, по мере поступления списка в T , и поэтому предполагается, что каждая конъюнкция (дизъюнкция) булевых функций списка реализуется при настройке структуры S_V отдельно. Перечисленные выше особенности продиктованы рядом соображений.

Известен целый ряд задач логического управления и автоматического контроля, в которых не требуется одновременная реализация всех имеющихся алгоритмов. Более того, иногда основная часть алгоритмов используется с очень большой дискретностью. Система, в которой применена однородная микроселектронная структура с автоматической настройкой (перенастройкой) в процессе эксплуатации, позволяет последовательно на одном и том же оборудовании реализовать все необходимые функции. Последние могут длительное время храниться на каком-либо носителе информации или вырабатываться в реальном времени как результат сложных процессов обработки информации и вводятся оператором или автоматически лишь по мере надобности.

Таким образом, по сравнению с рядом других принципов проектирования, использование однородных структур позволит получить выигрыш в сложности, надежности и, кроме того, обеспечит большую функциональную гибкость собственно управляющего устройства или системы автоконтроля. Платой за это является, в частности, весь комплекс устройств, позволяющих осуществить управление (настройку). В блок-схеме S^* это отражено введением специального настроечного автомата T , который, кстати говоря, может быть реализован на тех же принципах, что и S_V .

Рассмотрение сложности реализации настроечных устройств необходимо и для того, чтобы ответить на ряд довольно общих вопросов, связанных с оценкой эффективности применения однородных вычислительных сред и структур. Так, одностороннее рассмотрение только простоты конструкции управляемых микроселектронных ячеек в S_V может привести к тому, что удачная с технологической точки зрения структура потребует настолько сложной конструкции T , что окажется нецелесообразным создание автоматической системы из этих двух устройств для решения некоторой задачи по управлению, контролю или обработке информации.

Данную работу можно рассматривать как попытку выяснения некоторых частных вопросов вышеупомянутой проблемы.

I. В качестве условного описания системы S^* (рис. I) с

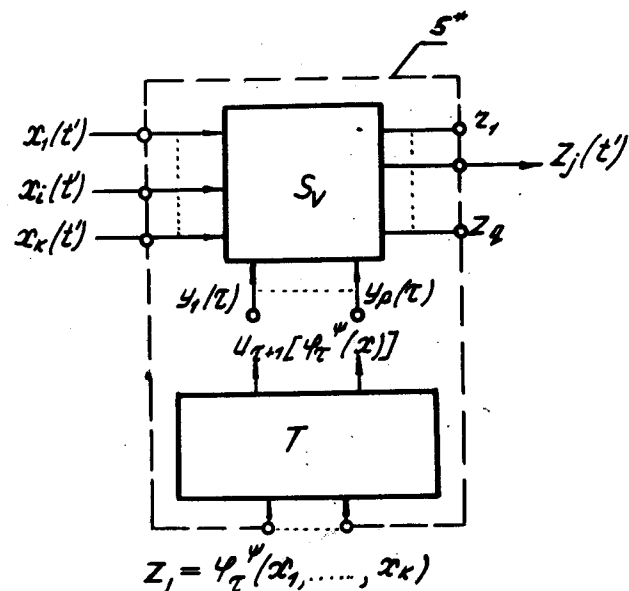


Рис. I.

информационной точки зрения примем следующие соотношения ($k > 2$, $k \neq q \neq 1$):

$$u_{q+1} [\varphi_q^\psi(X)] = \Phi [z_j = \varphi_q^\psi(X)], \quad (I)$$

$$z_j = f_{\tau+1}^{\psi}(X) \rightarrow E \left\{ \omega_{\tau+1} \left[\varphi_{\tau}^{\psi}(X) \right] \right\}, \quad (2)$$

$$z_j = f_{\tau+1}^{\psi}(X) \sim \varphi_{\tau}^{\psi}(X), \quad (3)$$

где $\varphi \in \{V, \Lambda\}$, а, через $X = \{x_1, \dots, x_k\}$, $Z = \{z_1, \dots, z_l\}$ обозначены соответственно множества входных и выходных двоичных переменных автомата S_v ; $\tau = 0, 1, 2, \dots$ - индексы дискретного времени, называемого тактами управления S_v ; $z_j = \varphi_{\tau}^{\psi}(X)$ - требования к преобразованию информации в S_v на такте с произвольным номером τ ; одновременно $\varphi_{\tau}^{\psi}(X)$ - оператор, заданный на входе T в виде упорядоченного списка буквенных символов. $\omega_{\tau+1} \left[\varphi_{\tau}^{\psi}(X) \right]$ - управляющее слово на выходе автомата T в момент $\tau+1$, являющееся входным для множества управляющих полюсов $Y = \{y_1, \dots, y_p\}$ автомата S_v , где $p \neq k$. Время между тактами τ и $\tau+1$ - задержка, связанная с переработкой автоматом T в соответствии с оператором $\varphi \left[z_j = \varphi_{\tau}^{\psi}(X) \right]$ входного слова в выходное, управляющее.

$E \left\{ \omega_{\tau+1} \left[\varphi_{\tau}^{\psi}(X) \right] \right\}$ - оператор в S_v , мгновенно перерабатывающий управляющее слово автомата T в команды, организующие (строющие) некоторый фиксированный на время между тактами $\tau+1$ и $\tau+2$ автомат S'_v с оператором

$z_j = f_{\tau+1}^{\psi}(X)$ (условно отмечено знаком " \rightarrow " в (2)); " \sim " - знак равносильности между двумя операторами в (3):

оператором $z_j = \varphi_{\tau}^{\psi}(X)$, заданном на входе T в момент

τ и оператором $f_{\tau+1}^{\psi}(X)$, реализованном в виде S'_v на такте $\tau+1$ под воздействием управления от T .

Таким образом, поведение S_v во времени $\tau = 1, 2, \dots$ описывается последовательностью формул $\varphi_{\tau}^{\psi}(X) \sim f_{\tau}^{\psi}(X)$,

$\varphi_1^{\psi}(X) \sim f_2^{\psi}(X), \dots$, и на каждом такте управления реализуется один автомат S'_v с оператором $z_j = f_{\tau+1}^{\psi}(X)$, который будем предполагать существенно зависящим от своих аргу-

ментов.

Естественно считать, что длительность моментов времени между тактами τ в общем случае неравномерная, а κ невелика. Кроме того, будем предполагать, что некоторый $x_{\delta} \in X$ может быть аргументом-константой ($x_{\delta} \equiv 0$ или $x_{\delta} \equiv 1$).

Для любого $\tau > 0$ примем, что $\varphi_{\tau}^{\psi}(X)$, задающий требуемое отображение X на Z , отличается от $\varphi_{\tau+1}^{\psi}(X)$,

по крайней мере, одним слагаемым (множителем). Таким образом,

специфика задания автоматов S'_v в S_v состоит еще и в том, что формульное слово на входе T является переменоместным, начиная с $\tau = 0$, и поэтому от такта к такту можно характе-

ризовать формулы $\varphi_{\tau}^{\psi}(X)$ тремя числами:

- 1) числом слагаемых (множителей) - ω , где $1 \leq \omega \leq 2^k$;
- 2) числом переменных в каждом из ω произведений (сумм) κ_{ℓ} ($\ell = 1, 2, \dots, \omega$), где $1 \leq \kappa_{\ell} \leq k$;
- 3) числом переменных, входящих в произведения (суммы) под знаком отрицания θ_{ℓ} ($\ell = 1, 2, \dots, \omega$), где $0 \leq \theta_{\ell} \leq \kappa_{\ell}$.

По предположению, управляемая структура S_v автомата S^* состоит из некоторого числа N однотипных УЛЭ. Под УЛЭ будем понимать (γ, η, ρ) - полюсное устройство, где γ - число входных полюсов, η - число управляющих полюсов, ρ - число выходных полюсов.

Будем считать, что реализуемая УЛЭ функция f^{\oplus} и его связи с другими элементами каждый раз во времени τ однозначно задаются управляющим словом вида (4), хранимым в памяти элемента.

$$\begin{array}{|c|c|c|c|c|} \hline \sigma_1 & \dots & \sigma_g & \dots & \sigma_p \\ \hline \end{array} \quad (4)$$

В (4) $\sigma_g \in \{0, 1\}$ ($g = 1, 2, \dots, \eta$); $]A[$ - разряды, определяющие связи УЛЭ с предшествующими источниками информации (входные полюса $S_v - X$ и выходные полюса предшествующих УЛЭ); $\oplus[$ - разряды, задающие настройку УЛЭ на реализацию одной функции f^{\oplus} из некоторого заранее зафиксированного множества $f^{\oplus} \in \{f_1, \dots, f_p\}$ ($p \geq 2$); $]B[$ - разряды, определяющие связи УЛЭ с последующими приемниками информации (выходные полюса $S_v - Z$ и выходные полюса после -

дующих УЛЭ).

Можно показать, что слово (4) представляет из себя свернутую в строку матрицу соединений входов и выходов УЛЭ с остальными источниками и приемниками информации в S_V .

В таком случае, если n_{SE}^ψ - суммарное (общее) количество УЛЭ, требующихся для реализации некоторого оператора $\varphi_\tau^\psi(X)$

в момент τ , то заданием n_{SE}^ψ матриц соединений в памяти элементов процесс организации структуры автомата S_V полностью определен для момента τ .

Аналогично, если во времени $\tau = 0, 1, 2, \dots$ автоматом S_V требуется последовательно реализовать операторы $\varphi_0^\psi(X), \varphi_1^\psi(X)$

$\varphi_2^\psi(X), \dots$, то, задавая с помощью автомата \mathcal{T} в такты $\tau + 1 = 1, 2, 3, \dots$ соответствующие матрицы соединений, процесс управления становится полностью определенным во всем времени τ .

Основными условиями при реализации любого $x_j = \varphi_\tau^\psi(X)$

в S_V будет неравенство $n_{SE}^\psi \leq N$ и формула (3). Выше через n_{SE}^ψ обозначено общее (суммарное) число УЛЭ при ре-

ализации (κ, τ) - полюсной сетью формулы $x_j = f_{\tau+1}^\psi(X) \sim \varphi_\tau^\psi(X)$, заданной в д.н.ф. ($\psi = \vee$) или к.н.ф. ($\psi = \wedge$), где ε - номер базиса, по которому осуществляется управление.

Возникает задача по выбору такого логического базиса ε из множества возможных, при котором функции $x_j = \varphi_\tau^\psi(X)$

реализуются с помощью минимального количества УЛЭ.

Возможность экономной реализации ФАЛ в S_V исследовалась при следующих ограничениях:

1) Считалось, что в каждый момент времени τ любой из N УЛЭ (под воздействием соответствующего внешнего управления) может реализовать только одну (любую) функцию f^ψ , где

$$\psi \in \{\supset(2), \neg(2), \&(\kappa), \vee(\kappa), \dagger(\kappa), \downarrow(\kappa), + (2), \neg(1), * (1)\}.$$

(Выше введены следующие обозначения: функции двух переменных - импликация \supset , запрет \neg , сумма по $\text{mod } 2$, эквиваленция \sim ; одноместные функции - отрицание \neg , функция повторения аргумента (и функция-константа, т.к. $x_j \equiv 0$ или $x_j \equiv 1$) - $*$; функции κ переменных ($\kappa \geq 2$) - $\&$, \vee , функция Шеффера (И - НЕ) - \dagger , стрелка Пирса (ИЛИ - НЕ) - \downarrow).

2) Предполагалось, что организация связей в позволяет произвести соединение входа каждого из N УЛЭ с любым полюсом $x_i \in X$ ($i = 1, 2, \dots, \kappa$) и выхода УЛЭ с любым $x_j \in Z$ ($j = 1, 2, \dots, \varphi$), и, кроме того, предполагалось, что обеспечено возможное соединение N УЛЭ по способу полного ориентированного графа.

Очевидно, что задавшись вышеприведенным универсальным запасом функций для УЛЭ, можно реализовать в S_V формулу

$$x_j = \varphi_\tau^\psi(X) \quad \text{в любом одном функционально полном ба-} \\ \text{зисе } \varepsilon, \text{ удовлетворяющем ослабленным критериям полноты (т.к.} \\ x_j \equiv 0 \text{ или } x_j \equiv 1) \text{ [4].}$$

Рассматривалось следующее множество полных базисов ($\varepsilon = 1, 2, \dots, 10$):

$$1 - \{\supset(2), * (1)\}, 2 - \{\neg(2), * (1)\}, 3 - \{\downarrow(\kappa), * (1)\};$$

$$4 - \{\dagger(\kappa), * (1)\}, 5 - \{\&(\kappa), \neg(1), * (1)\}, 6 - \{\&(\kappa), \sim(2),$$

$$* (1)\}, 7 - \{\&(\kappa) + (2), * (1)\}, 8 - \{\vee(\kappa), \neg(1), * (1)\},$$

$$9 - \{\vee(\kappa), \sim(2), * (1)\}, 10 - \{\vee(\kappa), + (2), * (1)\}.$$

Так как с самого начала предполагалось, что каждая конъюнкция (дизъюнкция) функций $\varphi_\tau^\psi(X)$ реализуется в S_V отдельно, то сложность сети над любым базисом ε можно подсчитать в виде:

$$n_{SE}^\vee = n_{SE}^{\vee(w)} + \sum_{\kappa=1}^w n_{SE}^{\&(\kappa\varepsilon)}, \quad (5)$$

$$n_{SE}^\wedge = n_{SE}^{\&(w)} + \sum_{\kappa=1}^w n_{SE}^{\vee(\kappa\varepsilon)}, \quad (6)$$

где через $n_{SE}^{\&(\kappa\varepsilon)}$, $n_{SE}^{\vee(\kappa\varepsilon)}$ обозначено соответственно число УЛЭ при реализации одной конъюнкции и одной дизъюнкции при настройке элементов только на функции выбранного базиса ε .

Ограничение 1 позволяет осуществить однозначный переход от числа функций, необходимых в базисе ε для реализации

$\varphi_\tau^\psi(X)$, к числу УЛЭ, реализующих эти функции. Ограничение 2 позволяет при этом считать, что УЛЭ можно соединять между собой самым произвольным образом.

Путем представления конъюнкций (дизъюнкций) функций $\varphi^\psi(X)$

во всех вышеперечисленных базисах ε , был получен ряд зависимостей $r_{s\varepsilon}^{\&(k_e)} = \mu(k_e, \theta_e)$, $r_{s\varepsilon}^{\vee(k_e)} = \mu(k_e, \theta_e)$.

Анализ характера этих зависимостей привел к выводу об эффективности переменнобазисного управления структурой блока S_v [5]. Суть такого управления в том, что УЛЭ настраиваются на тот или другой базис в зависимости от численных значений параметров k_e, θ_e реализуемой конъюнкции (или дизъюнкции). Естественно, что при этом список используемых функций расширяется, например:

$$\{\&(k), \vee(k), \neg(k), \uparrow(k), \downarrow(k), *(k), \uparrow(k), \downarrow(k), *(k), \uparrow(k), \downarrow(k), *(k)\}, \{\uparrow(k), \downarrow(k), \neg(k), *(k)\}, \{\&(k), \vee(k), \neg(k), \supset(k), \leftarrow(k), *(k)\}.$$

Приведенные выше наборы функций естественно назвать расширенными базисами. (Занумеруем их, соответственно, как I3, I4, I5).

В расширенных базисах можно указать ряд подстановок [6], реализация которых при настройке S_v приводит к тому, что

$$r_{sI3}^{\psi} \leq r_{sI5}^{\psi} \leq r_{sI4}^{\psi} \leq \dots \leq r_{sI(4, \dots, 10)}^{\psi} \leq \dots \leq r_{sI(2)}^{\psi}. \quad (7)$$

Таким образом, было выяснено, что нет необходимости проектировать УЛЭ, реализующие все перечисленные вначале функции Φ . В классе ФАЛ в д.н.ф. (к.н.ф.) существуют гораздо более короткие списки функций (см. базисы I3-I5), которые позволяют реализовать $\varphi_c^{\psi}(X)$ наилучшим образом в смысле минимума

используемых УЛЭ.

II. Основное внимание ниже уделено рассмотрению алгоритма, моделирующего работу автомата \mathcal{T} при синтезе в S_v логической сети, описываемой ФАЛ в д.н.ф.

Выбранная ниже форма записи списка (входной язык \mathcal{T}) близка к общепринятой записи формул алгебры логики в д.н.ф. и наличие некоторой специфики связано лишь с определенными удобствами переработки списка в \mathcal{T} . Например, формула

$$x_2 = \varphi^{\vee}(x_1, \dots, x_4) = \bar{x}_1 x_2 \bar{x}_4 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \quad (8)$$

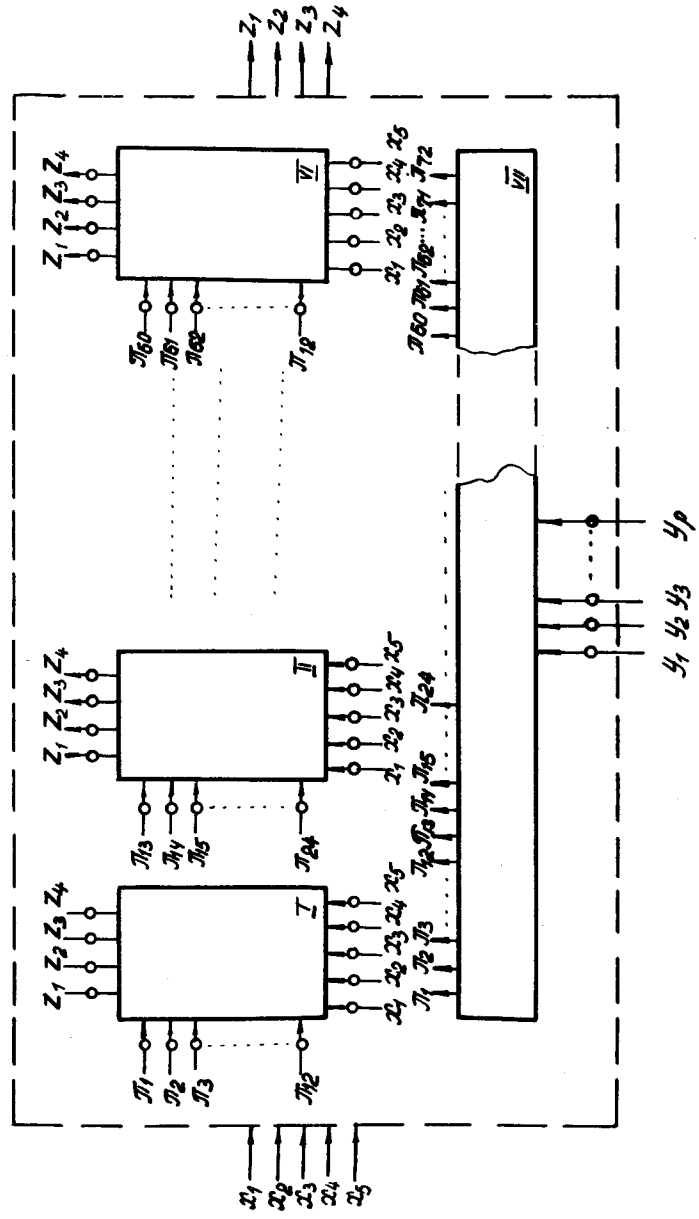


Рис. 2.

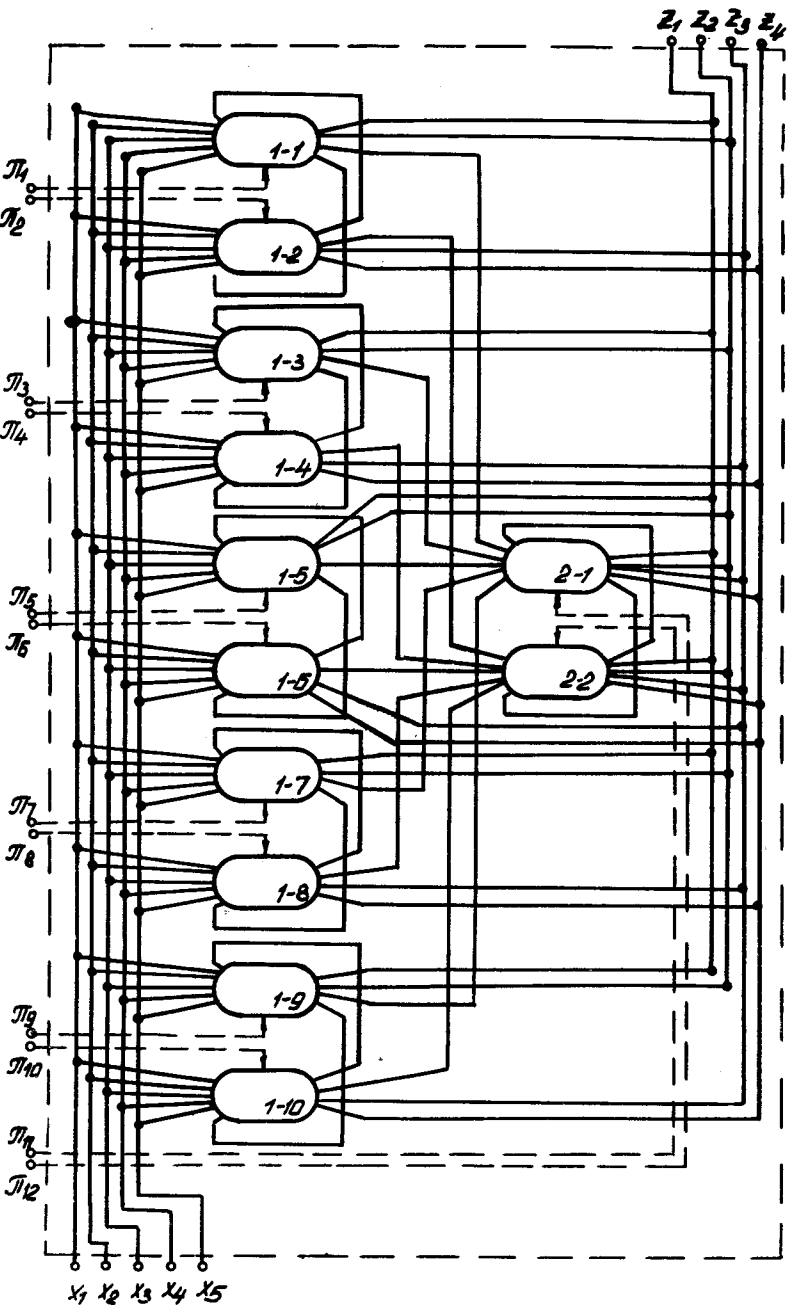


Рис. 3.

записывается здесь в виде следующего списка:

$H, j2, 7 \times 1, X2, 7 \times 4, \vee, 7 \times 1, X2, X3, \vee,$
 (1) (2) (3) (4) (5) (6) (7) (8) (9) (10)
 $X1, 7 \times 2, X3, \vee, X1, 7 \times 2, 7 \times 4, \vee, X1, X2,$
 (11) (12) (13) (14) (15) (16) (17) (18) (19) (20) (9)
 $7 \times 3, X4, \vee, 7 \times 1, 7 \times 2, 7 \times 3, X4, \square$
 (21) (22) (23) (24) (25) (26) (27) (28)

где „H“ и „□“ — символы начала и конца списка; „ \vee “ — символ логической связки „ИЛИ“, а „ $X1, 7 \times 2$...“ — символы переменных с номерами „1“ и „2“, ..., входящих в произведения соответственно без знака отрицания и под знаком отрицания (в (9) нумерация в скобках под символами указывает порядковый номер символа в списке и нужна для дальнейших целей).

Поскольку основной интерес проявляется здесь к правилам, с помощью которых можно проследить, как кодированная последовательность буквенных символов преобразуется автоматом \mathcal{T} в последовательность кодов, задающих настройку УЛЭ, выбор параметров структуры S_V подчинен в основном иллюстративным целям. Кроме того, целый ряд вопросов, в частности все вопросы, связанные с обсуждением способов засылки управляющей информации в память УЛЭ, здесь не рассматриваются, так как последние подробно освещены в работе [3].

Будем считать, что УЛЭ реализует под воздействием управления одну и только одну функцию f^Φ , где $\Phi \in \{\&, \vee, \wedge, \neg, \times\}$. Число входных переменных K выберем равным пяти, число выходных переменных $q = 4$. Таким образом, $X = \{x_1, \dots, x_5\}$, а $Z = \{z_1, \dots, z_4\}$. Кроме того, считается, что $y \leq 6$, а $\rho = 1$, хотя в принципе допускается, что $\rho \leq 5$. Число УЛЭ/выбрано равным 72, но управляемая структура S_V разбита на шесть однотипных блоков (модулей) ω , содержащих по 12 УЛЭ каждый *).

Вариант конструкции S_V приведен на рис. 2, где 1, ..., 11 — однотипные по своей конструкции модули, а УИ — устройство, осуществляющее засылку управляющей информации по каналам настройки $\pi_1, \dots, \pi_{12}, \dots, \pi_{71}, \pi_{72}$. На рис. 3 пока-

* Модульный принцип естественно принять из технологических соображений и удобств эксплуатации. В реальной конструкции УЛЭ может представлять из себя целый участок однородной структуры, состоящий из простейших по своей конструкции элементов.

зана конструкция модуля I.

Автоматом S^* рассматриваемого типа при соответствующих управлениях структурой S_V можно реализовать любую $z_j = \varphi_z^V(X)$,

$z_j \in Z$, принадлежащую к множеству всех формул алгебры логики в д.н.ф. от переменных X , где $2 \leq k \leq 5$. (Для случаев $k=5$ число слагаемых в формулах - ω при принятой организации структуры ограничено усл. обозн. $\omega \leq 30$).

В каждом модуле ω , где $\omega = \{I, \dots, \overline{VI}\}$, будем различать два яруса УЛЭ (первый, содержащий 10 УЛЭ, и второй, содержащий 2 УЛЭ), связанных каналами передачи информации таким образом, что форматы управляющих элементами слов запишутся в виде:

$$\begin{array}{l}
 \omega-1-1 \\
 \omega-1-2 \\
 \omega-1-3 \\
 \omega-1-10
 \end{array}
 \left. \begin{array}{l}
 \begin{array}{c}
 \overbrace{\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12}}^{] \oplus [\quad z_1 \quad z_2 \quad \tilde{z}_{1-1,2-1}} \\
 \overbrace{\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12}}^{] \oplus [\quad z_3 \quad z_4 \quad \tilde{z}_{1-1,1-2} \quad \tilde{z}_{1-2,2-2}} \\
 \overbrace{\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12}}^{] \oplus [\quad z_1 \quad z_2 \quad \tilde{z}_{1-2,1-1} \quad \tilde{z}_{1-3,2-1}} \\
 \overbrace{\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12}}^{] \oplus [\quad z_3 \quad z_4 \quad \tilde{z}_{1-3,1-4} \quad \tilde{z}_{1-10,2-2}} \\
 \overbrace{\lambda_1 \lambda_2 \lambda_3 \lambda_4 \lambda_5 \lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12}}^{] \oplus [\quad \tilde{z}_{1-10,1-9}}
 \end{array}
 \end{array} \right\} \text{(I0)}$$

$$\begin{array}{l}
 \omega-2-1 \\
 \omega-2-2
 \end{array}
 \left. \begin{array}{l}
 \begin{array}{c}
 \overbrace{\lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12} \lambda_{13}}^{] \oplus [\quad z_1 \quad z_2 \quad z_3 \quad z_4 \quad \tilde{z}_{2-1,2-2}} \\
 \overbrace{\lambda_6 \lambda_7 \lambda_8 \lambda_9 \lambda_{10} \lambda_{11} \lambda_{12} \lambda_{13}}^{] \oplus [\quad z_1 \quad z_2 \quad z_3 \quad z_4 \quad \tilde{z}_{2-2,2-1}}
 \end{array}
 \end{array} \right\} \text{(II)}$$

где в (I0), (II) везде $\lambda \in \{0,1\}$, а с помощью $\omega-1-1, \omega-1-2, \dots, \omega-1-10, \omega-2-1, \omega-2-2$ занумерованы УЛЭ подструктур $\omega \in \{I, \dots, \overline{VI}\}$ и где вторая цифра означает номер яруса, а третья - номер УЛЭ в ярусе; $\oplus \in \{\&, \vee, \wedge, \uparrow, \downarrow, *, *\}$

- код логической связки, реализуемой УЛЭ; $\tilde{z}_{1-1,1-2}, \dots; \tilde{z}_{2-1,2-2}$ обозначают связи между УЛЭ в первом и втором ярусах, позволяющие осуществить соединение (каскадирование) элементов из разных ярусов при управлении в выбранной расширенной ф.п.с.е.

Для управления структурой S_V с помощью автомата \mathcal{T} требуется (при задании $\varphi_z^V(X)$ списком) выполнить следующее:

1) Провести анализ формулы, поступающей на входы автомата \mathcal{T} , поскольку по условию в рассматриваемой модели от такта к такту во времени τ формулы переменностны. Здесь анализ сведется к выработке значений $\kappa_1, \theta_1; \kappa_2, \theta_2; \dots; \kappa_w, \theta_w$ последовательно для каждой конъюнкции по мере её поступления.

2) Вычислить $\pi_{se}^{\&(k_2)}$ по всем $e=1,2,\dots,w$ и $\pi_{se}^{V(w)}$

в выбранной для управления ф.п.с. $\varepsilon = \{\&, \vee, \wedge, \uparrow, \downarrow, *\}$ с целью проверки условия реализуемости формулы $z_j = \varphi_z^V(X)$;

при этом должно быть, что $\pi_{se}^V \leq N$.

3) Сформировать в точности π_{se}^V команд управления (настройки), образующих управляющее слово $\mathcal{U}_{\tau+1} [z_j = \varphi_z^V(X)]$ для структуры S_V .

4) Заслать π_{se}^V команд управления (настройки) в ячейки памяти π_{se}^V УЛЭ, выбранных для управления в S_V .

5) С точки зрения общего быстродействия требуется, по возможности, совместить п. I-4, так как всегда желательно вести синтез сети с минимальной задержкой на обработку списка и, по возможности, одновременно с поступлением формулы на входы автомата \mathcal{T} , не дожидаясь появления символа "□".

В этой связи примем за такты пульсаций структуры S_V все нечетные моменты времени, начиная с $\tau=1$. Будем считать, что оператор $z_j = \varphi_z^V(X)$ строится управлением до момента, ко-

торый обозначим как $\tau=1$. В этот момент в память π_{se}^V -го УЛЭ автоматом \mathcal{T} засылается управление (информация о настройке). Момент $\tau=2$ будем отождествлять с моментом поступления первого символа следующей формулы $\varphi_z^V(X)$ на входы \mathcal{T} .

Во времени между тактами $\tau=1$ и $\tau=2$ автомат S^* функционирует как "жесткий" автомат S_V' с оператором

$$z_j = f_j^{\vee}(X) \sim \varphi_0^{\vee}(X)$$

и т.д.

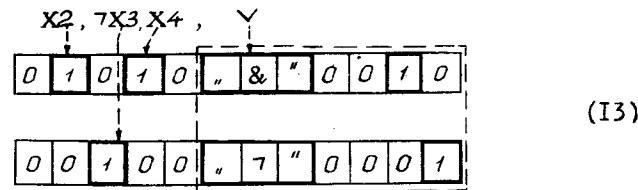
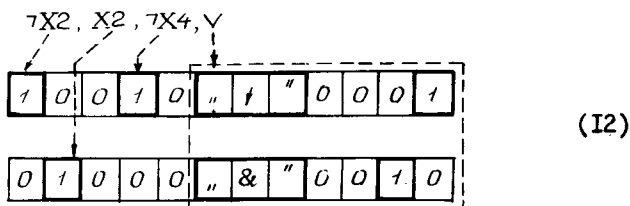
В соответствии с определением управляющего слова (см. (4)), последнее содержит информацию о требуемом типе функционирования УЛЭ и о точках, в которых элемент соединяется с предшествующими и последующими полюсами.

Для автомата \mathcal{T} , перерабатывающего список символ за символом, микротакты управления выбраны равными времени анализа одной конъюнкции: вначале, по мере поступления символов переменных, формируются разряды $\lambda_1, \dots, \lambda_5$; одновременно ведется подсчет k_e и θ_e в конъюнкции; затем, когда на вход \mathcal{T} поступает символ " \vee ", в зависимости от того, какие значения приняли числа k_e, θ_e в поступившей конъюнкции, формируются значения остальных разрядов $\lambda_6, \dots, \lambda_{12}$ (см. таблицу I и (12), (13)).

Вначале настраиваются последовательно УЛЭ первого яруса $\omega = I$. Затем, если $w > k$, настраивается II ярус $\omega = I$ и происходит переход на настройку элементов первого яруса в модуле с номером II. Процесс повторяется до тех пор, пока вместо символа " \vee " на вход \mathcal{T} не поступит символ конца формулы " \square ". Таким образом, список (II) реализуется по частям, и процесс синтеза может остановиться на любом шаге ($1 \leq w < k, w = k, k < w < 2k, \dots$). Для рассматриваемого автомата S^* в каждом модуле $\omega \in \{I, \dots, IV\}$ можно реализовать до пяти слагаемых от пяти переменных.

Для элементов 2 яруса (в случае перехода от ω' к ω'' , где $\omega'' = \omega' + 1$ при $w > k$; а также в случае поступления символа " \square ") автоматом \mathcal{T} реализуется правило "равномерной" нагрузки входов (таблица 2).

Сказанное выше можно проиллюстрировать следующими примерами для конъюнкций " $\neg X_1, X_2, \neg X_4$ ", " $X_2, \neg X_3, X_4$ " в формуле (9), у которых k_e, θ_e равны соответственно 3, 2 и 5, 1:



Ниже (14) приводится логическая схема (7) алгоритма работы автомата \mathcal{T} . Алгоритм не содержит какую-либо процедуру минимизации формулы, а организует структуру автомата S_V , моделируя в ф.п.с. $\{\&, \vee, \neg, /, \cdot, \times\}$ формулу $z_j = \varphi_j^{\vee}(X)$, поступающую на вход \mathcal{T} . Граф формулы (8), построенный алгоритмом управления, приведен на рис. 4. Цифрами 3, 4, ..., 28 обозначены управления, выработанные в \mathcal{T} для символов списка (9) с теми же номерами. \square , $\&$, ..., \times - обозначения кодов, задающих настройку УЛЭ на реализацию логической функции с отмеченной в квадрате связкой.

$$\begin{aligned}
 & H_0 \xrightarrow{12, 2, 9, 8, 3} H_1 P_2 L H_3 \xrightarrow{1} P_4 \xrightarrow{1} A_5 \\
 & P_6 \xrightarrow{1} A_7 \Phi_8 \xrightarrow{1} P_9 \xrightarrow{1} P_{10} \xrightarrow{1} A_{11} \Phi_{12} \xrightarrow{1} \\
 & \xrightarrow{1} P_{13} \xrightarrow{1} P_{16} \Phi_{14} A_{15}; \xrightarrow{1} P_{16} \xrightarrow{1} P_{18} \xrightarrow{1} P_{17} \xrightarrow{1} A_{18}
 \end{aligned} \quad (14)$$

Выше через H_0 обозначен оператор начала: A_{15}, A_{18} - операторы останова. H_1 - обобщенный нестандартный оператор, осуществляющий, в частности, проверку условий $n_{SE}^{\vee} \leq N, w < k$,

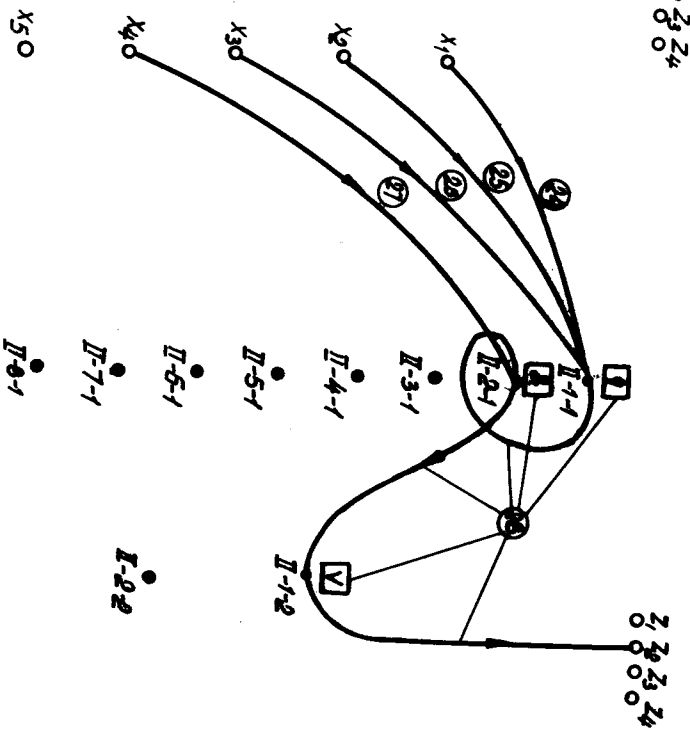
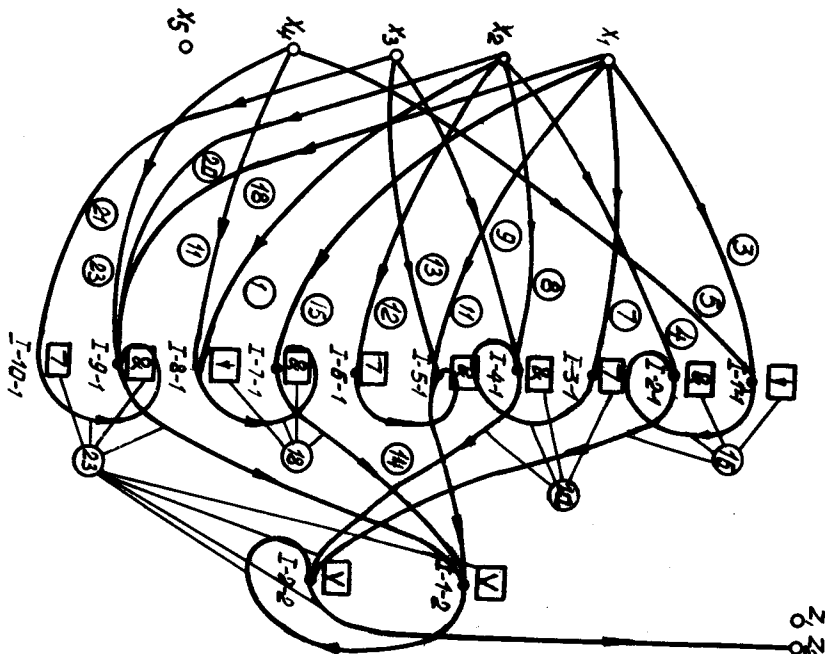
а также выделение кода символа из кода списка и формирование засылки следующего кода списка. $P_2 = 1$, если анализируемым символом является " H ". При этом условии H_3 очищает содержимое счетчиков (c_4) $k_e, \theta_e, w_1, w_2, w$. Если анализируемым символом является символ переменной " x_i ", где $i = 1, \dots, 5$, то $P_4 = 1$ и A_5 прибавляет + 1 к содержимому счетчика k_e ($c_4 k_e$). Если, кроме того, анализируемый " x_i " входит в конъюнкцию под знаком отрицания, то $P_6 = 1$ и A_7 прибавляет + 1 к содержимому $c_4 \theta_e$. Обобщенные операторы Φ_8, Φ_9 в соответствии с номером переменной i формируют $\lambda = 1$ в i -й позиции одного из двух управляющих слов, (см. (10), (11)). Если анализируемым символом является связка " \vee ", то $P_{10} = 1$ и A_{11} прибавляет + 1 к содержимому $c_4 w$;

Таблица I

Синтез управлений для логических элементов первого яруса

№	Значения			Типы управляющих слов $\alpha = \alpha(\theta_e, K_e)$	Обозначения
	θ_e	K_e	α_i		
1	$\theta_e = 0$	$K_e > \theta_e$	ΦA & 0010	α_1	
			000000000000	α_0	
2	$\theta_e = 1$	$K_e > \theta_e$	ΦA & 0010	α_1	
			ΦA 7 0001	α_2	
3	$\theta_e > 1$	$K_e > \theta_e$	ΦA & 0010	α_1	
			ΦA † 0001	α_3	
4	$\theta_e > 1$	$K_e = \theta_e$	ΦA † 0010	α_4	
			000000000000	α_0	
5	$\theta_e > 1$	$K_e > \theta_e$	ΦA † 0001	α_3	
			ΦA & 0010	α_1	
6	$\theta_e = 1$	$K_e > \theta_e$	ΦA 7 0001	α_2	
			ΦA & 0010	α_1	

- Примечания: 1. Содержимое разрядов $\lambda_1, \dots, \lambda_5$ формируется автоматом Γ в процессе анализа конъюнкции по мере её поступления (см. (I2), (I3)). Обозначено как "ФА".
2. Разряды λ_9, λ_{10} алгоритмом (I4) не используются.
3. В разрядах $\lambda_6, \lambda_7, \lambda_8$ вместо кода связи указана соответствующая связька.



Синтез управлений для логических элементов второго яруса

№	Значения		Типы управляющих слов $\beta = \beta(w, w_1, w_2)$	Обозначения						
	W	w_1, w_2								
1,2	$w=1$	$w_1=1 (w_1=0)$	<table border="1"><tr><td>*</td><td>Φ</td><td>0</td></tr></table>	*	Φ	0	$\beta_1 (\beta_0)$			
		*	Φ	0						
$w_2=0 (w_2=1)$	<table border="1"><tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	0	0	0	0	0	0	0	0	$\beta_0 (\beta_1)$
0	0	0	0	0	0	0	0			
3	$w=2$	$w_1=1$	<table border="1"><tr><td>v</td><td>Φ</td><td>0</td></tr></table>	v	Φ	0	β_3			
		v	Φ	0						
$w_2=1$	<table border="1"><tr><td>*</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	*	0	0	0	0	1	β_2		
*	0	0	0	0	1					
4,5	$k > w > 2$	$w_1 > 1 (w_1=1)$	<table border="1"><tr><td>v</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	v	0	0	0	0	1	$\beta_4 (\beta_3)$
		v	0	0	0	0	1			
$w_2=1 (w_2 > 1)$	<table border="1"><tr><td>v</td><td>Φ</td><td>0</td></tr></table>	v	Φ	0	$\beta_3 (\beta_4)$					
v	Φ	0								
6	$k > w > 2$	$w_1 > 1, w_1 > w_2$	<table border="1"><tr><td>v</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	v	0	0	0	0	1	β_4
		v	0	0	0	0	1			
$w_2 > 1$	<table border="1"><tr><td>v</td><td>Φ</td><td>0</td></tr></table>	v	Φ	0	β_3					
v	Φ	0								
7	$k > w > 2$	$w_1 > 1, w_2 > w_1$	<table border="1"><tr><td>v</td><td>Φ</td><td>0</td></tr></table>	v	Φ	0	β_3			
		v	Φ	0						
$w_2 > 1$	<table border="1"><tr><td>v</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr></table>	v	0	0	0	0	1	β_4		
v	0	0	0	0	1					
8,9	$w=k$	$w_1 > 1, w_2 \neq w_1$	аналогичны №6 при $w_1 > w_2$	β_3, β_4						
		$w_2 > 1$	аналогичны №7 при $w_2 > w_1$	β_3, β_4						

- Примечания: 1. Содержимое разрядов $\lambda_9, \dots, \lambda_{12}$ формируется автоматом \mathcal{T} в зависимости от номера выходного полюса "j", относительно которого задана формула (список). Обозначено как " Φ ".
2. В разрядах $\lambda_6, \lambda_7, \lambda_8$ вместо кода связи указана соответствующая связь.
3. * - функция повторения аргумента.

а Φ_{12} формирует остальные разряды пары управляющих слов, задавая перекрестные связи между элементами I-го яруса и связь с одним элементом из 2-го яруса (см. (I2), (I3), таблицу I). $P_{13}=1$, если анализируемым символом является "0". При последнем условии Φ_{14} формирует перекрестные связи между элементами 2-го яруса и связь с j-м выходным полюсом (таблица 2). (Φ_{14} , кроме того, используется и при переходе от модуля ω' к ω'' при $w=k$; в (I4) передача управлений не указана). $P_{16}=1$, когда на вход \mathcal{T} поступает символ "j", где $j \in \{1, \dots, 4\}$. Последний засылается в оперативную память \mathcal{T} (Π_{17}) и используется каждый раз при применении оператора Φ_{14} .

Для простоты в (I4) отмечены два останова: при поступлении символа конца списка (\mathcal{Y}_{15}) и при сбое, когда поступивший на вход символ анализом не может быть отнесен ни к одному из типов символов списка (\mathcal{Y}_{18}). Кроме того, не отмечены подробно вопросы перехода от ω' к ω'' и не включены управления, связанные с "разрушением" работающей структуры перед началом реализации следующего по порядку списка.

Алгоритм обладает недостатками. В частности, в нем не использована возможность последовательной реализации двух и более формул от различных входных переменных X при условии, что сумма требующихся УЛЭ не превышает числа, имеющихся в S_v . Кроме того, в случаях, когда при $w=1$ после конъюнкции поступает символ конца списка, алгоритмом не используются выходы УЛЭ I-го яруса, а используется передача на j-й выходной полюс с помощью элемента второго яруса. Отмеченные недостатки легко устранимы расширением списка типов управляющих слов. Однако последнее связано, в свою очередь, с усложнением автомата \mathcal{T} . В целом же рассматривавшийся выше алгоритм позволяет представить характер переработки информации в \mathcal{T} и оценить в некоторой степени его конструктивную сложность.

Анализ операторов алгоритма настройки показывает, что конструктивно это устройство может быть сведено к дешифратору, нескольким двоичным счетчикам, небольшой оперативной памяти (для констант) и комбинационным схемам, управляющим выборкой из памяти в зависимости от содержимого счетчиков. Во всяком случае, объем памяти этого устройства оказывается на порядок меньше суммарного объема памяти в \mathcal{N} управляемых логических элементах, так как все управления могут быть выполнены с помощью сочетания анализа поступающей на вход \mathcal{T} формулы и восьми типов управляющих слов-констант ($\alpha_1, \dots, \alpha_4; \beta_1, \dots, \beta_4$)

различных попарных комбинациях (таблицы I и 2).

На практике управляемая структура S_v будет обладать высокой однородностью как в отношении элементов, так и связей между ними. В связи с этим можно высказать предположение о том, что реализация в однородной структуре (среде) сколь угодно сложных логических сетей может быть осуществлена с помощью сравнительно небольшого числа стандартных правил настройки, хранящихся в управляющем автомате. Иными словами, сложность конструкции этого устройства не будет являться доминирующей. К такому предположению приводят, в частности, результаты исследований ряда других алгоритмов настройки S_v при реализации ФАЛ в разных базисах.

Л И Т Е Р А Т У Р А

1. К.В. Карандеев. Измерительные информационные системы и автоматика. - Вестник АН СССР, 1961, № 10.
2. Э.В. Евреинов. О микроструктуре элементарных машин вычислительной системы. - Вычислительные системы. Новосибирск, ИМ СО АН СССР, 1962, вып. 4.
3. Э.В. Евреинов. Теоретические основы построения универсальных вычислительных сред. - Вычислительные системы, Новосибирск, ИМ СО АН СССР, 1965, вып. 17.
4. В.М. Глушков. Введение в кибернетику. Изд. АН УССР, Киев, 1964.
5. Б.Г. Матиенко. К возможности уменьшения числа логических элементов за счет комбинирования полных систем функций алгебры логики. - Известия СО АН СССР. Серия технических наук, 1964, Вып. 3, № 10.
6. Б.Г. Матиенко. К сравнительной оценке сложности одноктактных схем дискретных измерительных устройств. - Автометрия, 1965, № 5.
7. А.И. Китов, Н.А. Криницкий. Электронные цифровые машины и программирование. М, ФМ, 1959.