БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ   ACADÉMIE BULGARE DES SCIENCES
ИЗВЕСТИЯ НА МАТЕМАТИЧЕСКИЯ ИНСТИТУТ
BULLETIN DE L'INSTITUT DE MATHÉMATIQUES
Том (Tome) XII

ИНЖЕНЕРНЫЕ МЕТОДЫ СИНТЕЗА КОМБИНАТОРНЫХ СТРУКТУР НА ПРОИЗВОЛЬНЫХ ЭЛЕМЕНТАХ

М. А. Гаврилов и В. М. Копыленко

1. ВВЕДЕНИЕ

Алгоритмические методы синтеза логических устройств и систем получают в настоящее время широкое развитие и применение в ряде областей современной техники. Существенные особенности и требования к этим методам с точки зрения прикладных задач, для которых они используются, могут быть сформулированы следующим образом:

большая размерность решаемых задач (число входных переменных и выходов до нескольких сотен);

пригодность для широкого класса логических элементов (в том числе избыточных наборов логических элементов);

учет реальных ограничений: по числу входов элементов, по числу разветвлений выходов, по характеру соединений между элементами, по топологии размещения элементов и т. п.;

наличие процедур, предусматривающих в случае необходимости выравнивание глубины структуры и минимизацию числа инверторов на входах;

наличие процедур, предусматривающих в случае необходимости устранения состязаний и получения структур с заданной степенью безотказности (введение избыточности);

получение неизбыточных структур (для заданных требований), близких к оптимальным с выбором оптимального набора элементов из числа заданных.

Существующие методы синтеза логических систем, как правило приспособлены к реализации структур на простейших логических элементах, давая в результате процедур минимизации описание структур в нормальной форме булевых функций, не учитывают реальных ограничений элементов, слабо приспособлены к реализации на элементах с усложненными структурными свойствами и характеризуются резким возрастанием числа необходимых операций с ростом размерности решаемых задач.

Поэтому для удовлетворения поставленных выше требований необходима разработка новых, более эффективных методов, приспособленных для решения задач большой размерности с учётом реальных ограничений элементов и достаточно универсальных в отношении структурных свойств последних.

Настоящий труд посвящен рассмотрению одного из таких методов, предназначенных для минимизации структур логических систем на этапах структурного синтеза.

2. ПОСТАНОВКА ЗАДАЧИ

Заданы условия работы многовыходной недоопределенной логической системы в виде набора k функций от n переменных $F_{(n,\ k)}=\{F_1, F_2, \ldots, F_k\}$. Каждая из функций F_i, $1\leq i\leq k$, задана наборами конституентов (или интервалов) $M_1^i=\{\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i\}$ и $M_0^i=\{\beta_1^i, \beta_2^i, \ldots, \beta_p^i\}$, где $\alpha_1^i, \alpha_2^i, \ldots, \alpha_m^i$ — конституенты (или интервалы), дающие на i-м выходе единицу, и $\beta_1^i, \beta_2^i, \ldots, \beta_p^i$ — конституенты (или интервалы), дающие на i-м выходе нуль.

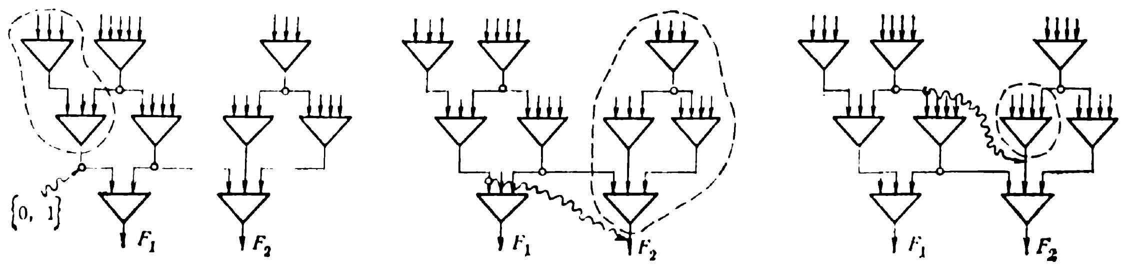
Задан функционально полный избыточный набор элементов, на которых должна быть реализована структура логической системы: $\Phi=\{\varphi_1, \varphi_2, \ldots \varphi_t\}$. Каждый элемент φ_i, $1\leq j\leq t$, задан наборами состояний его входов $\pi_1^j=\{\xi_1^j, \xi_2^j, \ldots, \xi_r^j\}$ и $\pi_0^j=\{\eta_1^j, \eta_2^j, \ldots, \eta_s^j\}$, где $\xi_1^j, \xi_2^j, \ldots, \xi_r^j$ — состояния входов, дающие на выходе элемента единицу, и $\eta_1^j, \eta_2^j, \ldots, \eta_r^j$ состояния входов, дающие на выходе элемента нуль.

Требуется построить неизбыточную структуру логической сети, реализующую заданную функцию $F(n, k)$ на заданном наборе элементов Φ, близкую к оптимальной с точки зрения числа необходимых элементов.

3. НЕИЗБЫТОЧНОСТЬ СТРУКТУР

Определим понятие неизбыточности структур. Будем считать, что структура **неизбыточна**, если в ней недопустимы (т. е. приводят к неправильной реализации заданных логических условий работы системы) следующие преобразования:

а) замена переменной на каком-либо входе какого-либо элемента структуры на константу (рис. 1, *а*), в результате чего некоторая часть структуры оказывается излишней (обведена на рис. 1, *а*, пунктиром);



б

Рис. 1

б) непосредственное соединение выхода какого-либо из элементов внутри структуры с каким-либо выходом всей структуры в целом (рис. 1, *б*), в результате чего некоторая часть структуры оказывается излишней;

в) непосредственное соединение выхода какого-либо из внутренних элементов структуры с входом какого-либо другого внутреннего или выходного элемента (рис. 1, *в*), в результате чего некоторая часть структуры оказывается излишней.

Недопустимость первого преобразования гарантирует неизбыточность структур типа „несвязного“ дерева. Два последних преобразования относятся к определению неизбыточности структур типа „связного“ дерева.

Понятие неизбыточности требует некоторых пояснений.

Прежде всего следует указать, что оно не совпадает с понятием „минимальности“ или „оптимальности“ структуры. Минимальной структурой (или точнее „абсолютно минимальной“) называется такая структура, для которой во всём множестве структур, соответствующих заданным условиям работы логической системы, не существует структуры, имеющей меньшее число логических элементов, входов и т. п., чем в данной, или, если речь идёт об оптимальности систем, для которой не существует реализация, обладающая значением функционала оптимизации. меньшим, чем для данной.

Неизбыточная структура может не быть минимальной, но обладать неизбыточностью в указанном выше смысле. Иначе говоря, неизбыточная реализация определяется по отношению к каждой данной реализации, какой бы она ни была.

Сформулированное выше определение относится к т. н. „ненужной“ избыточности. Однако, может быть и необходимая избыточность. Такая избыточность нужна, например, для устранения существенных состязаний между элементами памяти или воздействиями, подаваемыми на входы структуры, для обеспечения заданной безотказности работы логической системы и т. п. Требования устранения состязаний и обеспечения заданной безотказности следует, однако, относить к условиям работы логической системы и тогда указанное определение будет полностью определять „ненужную“ избыточность.

Как показывает опыт, в методах синтеза многовыходных структур со связностью между структурами отдельных выходов, не предусматривающих получение неизбыточных структур, избыточность в числе элементов может доходить до 30—40 %, дополнительные же операции по устранению избыточности являются весьма громоздкими. Поэтому при разработке методов синтеза следует ориентировать их на получение неизбыточных структур.

4. ХАРАКТЕРИСТИКИ ЛОГИЧЕСКИХ ЭЛЕМЕНТОВ

Для построения метода синтеза, пригодного для широкого класса логических элементов, необходимо определение некоторых общих характеристик их, введение которых в алгоритм могло бы автоматически определять структурные свойства элемента. В особенности это необходимо при синтезе структуры на расширенном по сравнению с минимальным функционально полным наборе элементов, когда в этот набор могут входить элементы с весьма различными функциональными свойствами. В этом случае при синтезе структуры, в частности, при выборе состава элемента, дающего оптимальную реализацию структуры, в алгоритм вводятся структурные характеристики различных элементов.

Будем различать, во-первых, логические элементы с „симметричными“ и „несимметричными“ входами. Для элементов с симметричными входами перестановка переменных на входах не изменяет функции, реализуемой логическим элементом, т. е.

$$f(x_1, x_2) = f(x_2, x_1).$$

Для элементов с несимметричными входами функция, реализуемая элементом, при перестановке переменных на входах изменяется.

Среди несимметричных элементов следует выделить элементы с неупорядоченными входами, у которых вес входов неодинаков (в частном случае может иметь различный знак). В числе элементов с неупорядоченными входами следует выделить в связи с их особыми структурными свойствами элементы типа „запрет“, в которых выполнение функции, реализуемой элементом, запрещается, если на один или несколько специально выделенных входов подается обусловленное воздействие.

Как уже говорилось, для учета различных структурных свойств элементов необходимо выбрать некоторые их характеристики, которые можно было бы вводить в алгоритм. В рассматриваемом алгоритме в качестве таких характеристик выбраны следующие:

„Характеристическое число“ (обозначается $[\tau_\varphi]$) — минимальное число входов элемента (минимальная сумма весов входов), подача на которые определенные воздействий дает на выходе элемента также определенное воздействие независимо от воздействий, подаваемых на другие входы. Так, например, для элементов „И“ и „ИЛИ“ $[\tau_\varphi]=1$, для мажоритарных элементов $[\tau_\varphi]=\frac{q(\varphi)+1}{2}$, где $q(\varphi)$ — число входов элемента.

„Индекс вхождения“ (обозначается γ) — значение воздействий, подача которых на число входов элемента, равное $[\tau_\varphi]$, дает определенное значение воздействия на выходе его. Например, для элемента „И“ $\gamma=0$, для элементов „ИЛИ“ $\gamma=1$, для мажоритарных элементов $\gamma=1$.

„Индекс реализации“ (обозначается ε) — значение воздействия на выходе элемента, которое имеет место при подаче на его $[\tau_\varphi]$ входов воздействия, равных γ. Например, для элемента „И“ $\varepsilon=0$, для элемента „ИЛИ“ $\varepsilon=1$, для мажоритарного элемента $\varepsilon=1$.

В соответствии со значением индекса реализации ε реализация структур может производиться или по множеству состояний M_1 (будем называть их в дальнейшем „рабочими“), или по множеству состояний M_0 (будем называть их в дальнейшем „нерабочими“). В соответствии с этим будем обозначать реализуемую часть таблицы состояний (или таблицы интервалов) через M_ε и нереализуемую — через $M_{\bar\varepsilon}$. Подстановка в эти символы значения ε будет, очевидно, определять сторону реализации.

Понятие $[\tau_\varphi]$ целиком применимо лишь к элементам с симметричными входами. Однако, некоторое расширение этого понятия позволяет использовать его для более широкого класса элементов. Например, включение в это понятие не одного, а некоторого множества значений числа входов позволяет расширить его использование на элементы „сумма по модулю два“ ($[\tau_\varphi]$ есть нечетное число), на элементы равнозначность ($[\tau_\varphi]=\{0, q(\varphi)\}$), на пороговые и многопороговые элементы и т. п. Допущение, что $[\tau_\varphi]$ может иметь свое значение для каждого из состояний M_ε и $M_{\bar\varepsilon}$, расширяет понятие $[\tau_\varphi]$ до т. н. „произвольных“ элементов, под которыми будем понимать элементы, структурные свойства которых выражаются некоторой произвольной булевой функцией, выраженной в нормальной форме (см., например, табл. 7).

Сводка соответствующих характеристик структурных свойств элементов дана в табл. 1.

Таблица 1

Типы логических элементов	Порог τ			Симметричн. входов		Упорядочен. весов входов		Наличие запрета	
	const								
	$\tau=1$	$\tau>1$		да	нет	const			
„И“, „ИЛИ“, „НЕ“ „ИЛИ-НЕ“, „И-НЕ“	+			+		+			+
Сумма по модулю 2	Нечетно			+		+			+
Равнозначнсcть	$\tau=\{0, q\}$			+		+			+
Запрет	+				+	+		+	
Мажоритарные	$\tau=\frac{q+1}{2}$			+		+			+
Пороговые		+	+		+		+		+
Абстрактн. нейрон			+		+		+	+	
Произвольные	+	+	+	+	+	+	+	+	+

q — число входов элемента

5. НАПРАВЛЕННЫЙ ПОИСК ОПТИМАЛЬНОЙ РЕАЛИЗАЦИИ СТРУКТУР

Как известно, получение абсолютно минимальной реализации структур релейных устройств требует полного перебора. Поэтому применение классических методов, основанных на таком переборе, ограничивается, даже при использовании универсальных вычислительных машин, число переменных не превышающем 10—12. В связи с этим в последние годы стали развиваться процедуры, основанные на различных эвристических построениях. Такие процедуры основаны на поэтапном построении структуры, начиная от входов или выходов, с выбором на каждом этапе из всех возможных дальнейших шагов какого-либо одного с помощью различных критериев, оценивающих близость некоторого набора переменных к реализуемой функции, сложность остающейся для реализации части структуры и т. п.

Будем называть такой процесс построения структуры логической системы „направленным“ поиском оптимальной ее реализации. Не будем останавливаться на обзоре предложенных за последние годы различных критериев направленного поиска (краткий обзор их приведен в [1]). Отметим лишь, что доказательные построения были проведены только для критериев оценки сложности одновыходных структур.

Большинство предложенных в настоящее время критериев для конечного числа переменных носит эвристический характер и требует для своей оценки статистического обследования.

В основу описываемого алгоритма положено построение структуры логической системы от ее выходов. Для осуществления направленного поиска структуры, близкой к оптимальной, принят следующий порядок выбора отдельных ее элементов:

а) выбор оптимальной последовательности реализации выходов структуры;

б) для каждого i-го очередного выхода — выбор оптимального выходного элемента;

в) для выбранного выходного элемента — выбор функций, подаваемых на его входы, реализующих оптимально, непротиворечиво и неизбыточно заданную для него функцию. Для элементов с несимметричными входами вы бор, кроме того, оптимальной последовательности заполнения входов выходного элемента:

г) после реализации структуры каждого из i-го выходов — ввод в качестве дополнительных переменных при реализации следующего очередного выхода функций выходов всех входящих в структуру i-го выхода логических элементов;

д) осуществление тех же операций направленного выбора по отношению к каждой из функций, подаваемых на входы выходного элемента до тех пор, пока структура очередного i-го выхода не будет построена полностью;

е) переход к реализации следующего по порядку выхода в последовательности тех же операций.

Рассмотрим подробнее цели и критерии выбора на каждом этапе. Отметим, что в соответствии с назначением алгоритма ниже будут рассматриваться критерии, ориентированные на получение структур с наименьшим числом логических элементов, хотя сам алгоритм при соответствующей замене критериев может использоваться и для получения оптимальных структур с точки зрения других функционалов оптимальности.

Выбор оптимальной последовательности реализации выходов логической системы

Основной целью этого выбора является установление такого порядка реализации выходов, при котором функции на выходах логических элементов и отдельных частей структуры уже реализованного выхода могли бы быть в наибольшей степени использованы для реализации последующих выходов с целью уменьшения общей сложности структуры (введение связности между структурами отдельных выходов). Принимается последовательность реализации выходов в порядке возрастания сложности их структуры на основании предположения, что в более простой структуре большее число входов ее логических элементов сможет быть использовано в качестве дополнительных переменных при реализации последующих выходов логической системы.

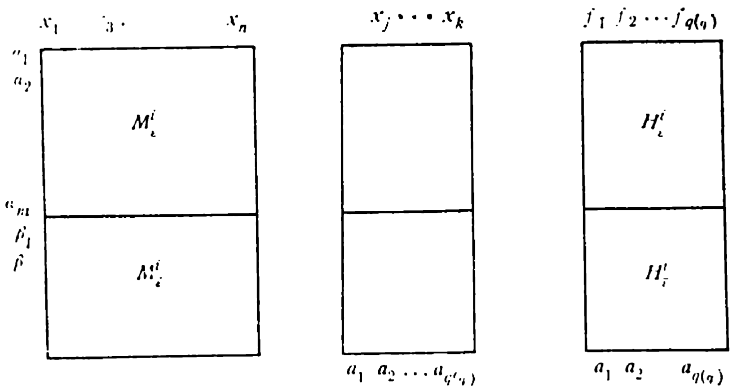
Для оценки сложности реализации выходов и определения последовательности реализации выходов используется предложенный Л. Шоломовым [2] критерий

$$I^i \quad M^i\log_2 M^i - M^i_\varepsilon \log_2 M^i_\varepsilon - M^i_{\bar\varepsilon} \log_2 M^i_{\bar\varepsilon},$$

где I^i — сложность реализации i-го выхода логической структуры, M^i_ε число состояний реализуемой стороны таблицы состояний, $M^i_{\bar\varepsilon}$ — число состояний нереализуемой стороны таблицы, $M^i = M^i_\varepsilon + M^i_{\bar\varepsilon}$. Справедливость критерия доказана в асимптотике. Статистическая проверка его на УВМ для конечного числа переменных показала, что он с некоторым постоянным коэффициентом характеризует сложность реализации структуры, выраженную числом букв нормальной формы булевой функции.

Выбор оптимальных переменных или функций, подаваемых на входы логических элементов

Задача синтеза выходного элемента всей структуры в целом или какой-либо ее части будет очевидно полностью решена, если на входы его будет подан такой набор переменных или функций, что для каждого из



Фиг. 2

состояний, принадлежащего к подмножеству M_ε, на выходе этого элемента будет появляться воздействие, равное ε, а для каждого из состояний, принадлежащего к подмножеству $M_{\bar{\varepsilon}}$, воздействие, равное $\bar{\varepsilon}$.

Обозначим совокупность значений переменных или функций, поданных на входы элемента φ_i, соответствующую какому-либо состоянию из M^i_ε, через h^i_ε, и соответствующую какому-либо состоянию из $M^i_{\bar{\varepsilon}}$ — через $h^i_{\bar{\varepsilon}}$.

Назовем „приведенной" таблицей состояний функций F_i по элементу φ_i таблицу, содержащую множество $H^i_\varepsilon = \{h^i_{\varepsilon,1}, h^i_{\varepsilon,2}, \ldots, h^i_{\varepsilon,r}\}$ строк, соответствующих всем состояниям из M_ε, и множество $H^i_{\bar{\varepsilon}} = \{h^i_{\bar{\varepsilon},1}, h^i_{\bar{\varepsilon},2}, \ldots, h^i_{\bar{\varepsilon},p}\}$ строк, соответствующих всем состояниям из $M_{\bar{\varepsilon}}$. Эта таблица будет содержать очевидно M строк и $q(\varphi)$ столбцов.

Легко доказать следующую теорему:

Теорема 1. Для того, чтобы функция F_i непротиворечиво реализовалась структурой с выходным элементом φ_i, необходимо и достаточно, чтобы для каждого состояния $\alpha^i_j \in M^i_\varepsilon$, $1 \leq j \leq m$, и $\beta^i_h \in M^i_{\bar{\varepsilon}}$, $1 \leq h \leq p$, выполнялось условие

$$h^i_\varepsilon(\alpha^i_j) \in \pi^i_1, \quad h^i_{\bar{\varepsilon}}(\beta^i_h) \in \pi_0,$$

где $h_\varepsilon^i(\alpha_j^i)$ — состояние приведенной таблицы, соответствующее состоянию α_j^i, и $h_{\bar\varepsilon}^{i}(\beta_h^i)$ — состояние приведенной таблицы, соответствующее состоянию β_h^i.

Назовём рангом r состояния из множества H $(H = H_\varepsilon \bigcup H_{\bar\varepsilon})$ — число переменных, имеющих в этом состоянии значение γ.

Очевидно, что в приведенной таблице ранг состояний для всех состояний из множества H_ε должен быть равен или больше $[\tau_\varphi]$, а для всех состояний из множества $H_{\bar\varepsilon}$ — равен или меньше $[\tau_\varphi] - 1$.

Реализация заданной структуры выходного элемента будет очевидно наиболее простой, если эти условия выполнены при подаче на входы элемента только одиночных переменных. Заданная функция при этом будет реализована только на одном элементе.

Если такого набора не существует, то реализация заданной функции F_i на элементе с данными логическими свойствами будет тем сложнее, чем более функция, реализуемая при данном наборе переменных на его выходах, будет отличаться от заданной.

Таким образом, целью выбора оптимальных переменных, подаваемых на входы логического элемента, является подбор такого подмножества переменных из заданного набора их, который обеспечивал бы реализацию на данном логическом элементе функции, наиболее приближающейся к заданной.

Будем называть „переходной" таблицу с числом строк, равным M, и числом столбцов, равным $q(\varphi)$, в которой каждый из столбцов сопоставлен какой-либо переменной. Задача выбора указанного оптимального набора переменных сводится, очевидно, к выбору такого набора, который бы в наибольшей степени приближал переходную таблицу к приведенной.

Для этого в предлагаемом алгоритме используются следующие критерии:

$$s_\gamma = \frac{n_\varepsilon^\gamma\, n_{\bar\varepsilon}^{\bar\gamma}}{n_\varepsilon^\gamma + n_{\bar\varepsilon}^{\bar\gamma}} \qquad s_{\bar\gamma} = \frac{n_\varepsilon^{\bar\gamma}\, n_{\bar\varepsilon}^{\gamma}}{n_\varepsilon^{\bar\gamma} + n_{\bar\varepsilon}^{\gamma}} \qquad s = \max\{s_\gamma,\ s_{\bar\gamma}\}.$$

Здесь n_ε^γ $(n_\varepsilon^{\bar\gamma})$ — число значений $\gamma\,(\bar\gamma)$ переменной в состояниях M_ε, соответствующих состояниям переходной таблицы H_ε, для которых ранг r меньше $[\tau_\varphi]$, а $n_{\bar\varepsilon}^{\gamma}$ $(n_{\bar\varepsilon}^{\bar\gamma})$ — число значений γ $(\bar\gamma)$ переменной в состояниях $M_{\bar\varepsilon}$, соответствующих состояниям переходной таблицы $H_{\bar\varepsilon}$, для которых $r + l \geq [\tau_\varphi]$, где l — число незаполненных переменными столбцов переходной таблицы. Подсчет значений s_γ и $s_{\bar\gamma}$ с учетом значений рангов в заполненных столбцах переходной таблицы обеспечивает более точное соответствие выбираемой переменной к изменяющейся таблице состояний по мере ее реализации.

При значении критерия s_γ или $s_{\bar\gamma}$, равным s_1, он определяет близость реализации заданной функции одной неинверсной переменной и применяется тогда, когда при реализации функции ставится задача уменьшения числа инверторов на ее входах, а критерий s определяет близость к реализации заданной функции одной инверсной или неинверсной переменной и применяется, когда задача уменьшения числа инверторов на входах элемента не ставится.

Переменные, подаваемые на входы элемента, выбираются по максимальному значению критериев s_γ и s. Эти критерии дают точную оценку близости к заданной функции в предельных случаях. Действительно, если в столбце данной переменной в заданной таблице состояний имеются в M_ε и M_ε^- только нули или единицы (переменная несущественная), то $s_\gamma = s_\gamma^- = 0$. Если в столбце в данной переменной в M_ε имеются только единицы, а в M_ε^- — только нули или наоборот, то это означает, что при $[r_\varphi]=1$ функция реализуется только одной буквой, а при $[r_\varphi]>1$ на соответствующий вход логического элемента должна быть подана только одна переменная. Однако в этой ситуации в первом случае (единицы в M_ε) $s_1 = \max$, $s_0 = 0$, а во втором случае (единицы в M_ε^-) $s_0 = \max$, $s_1 = 0$, что и определяет выбор соответствующей переменной, как оптимальной.

В промежуточных случаях критерии s_γ, s дают приближенный результат, приводящий к некоторому набору переменных, дающему переходную таблицу, отличную от приведенной, т. е. содержащую для некоторых состояний из H_ε ранг, меньший $[r_\varphi]$, и для некоторых состояний из H_ε^- — ранг, больший $[r_\varphi]-1$. Для устранения этой противоречивости переменные на входах элемента должны быть заменены доопределяющими функциями. Состав единиц и нулей в столбцах переходной таблицы даёт возможность определить таблицы состояний для этих функций.

Переходная таблица представляет собой некоторый план реализации заданной функции на выходном элементе. План, полученный с помощью выбора переменных, подаваемых на входы элемента, по критериям s_γ или s будем называть „опорным“. Так как в опорном плане переменные выбирались без учета их совместного влияния на ранги состояний переходной таблицы, то среди остальных переменных реализуемой функции могут быть такие, которые позволяют улучшить план реализации. Соответствующая процедура состоит в том, что в каждом столбце переходной таблицы, соответствующая ему переменная x_i поочередно заменяется другими переменными x_j (в том числе, и переменными, уже использованными при построении переходной таблицы) и определяется, улучшает ли эта замена план реализации или нет. Для определения этого факта для переменной x_i, выбранной ранее, подсчитывается величина

$$t_i = n_\varepsilon^i - n_{\varepsilon^-}^i,$$

где n_ε^i — число состояний из M_ε, в которых ранг с учетом значений данной переменной меньше $[r_\varphi]$, и $n_{\varepsilon^-}^i$ — число состояний из M_ε^-, в которых ранг с учетом значений данной переменной больше $[r_\varphi]-1$. Для переменных, заменяющих x_i, подсчитывается величина

$$t_j = \max\,\{t_j^\gamma,\ t_j^{\bar\gamma}\},$$

где $t_j^\gamma = n_{\varepsilon,j}^\gamma - n_{\varepsilon^-,j}^\gamma$ и $t_j^{\bar\gamma} = n_{\varepsilon,j}^{\bar\gamma} - n_{\varepsilon^-,j}^{\bar\gamma}$.

В этих выражениях t_j^γ и $t_j^{\bar\gamma}$ характеризуют степень улучшения плана соответственно для неинверсного и инверсного значений переменной x_j, а величины $n_{\varepsilon,j}^\gamma$, $n_{\varepsilon,j}^{\bar\gamma}$, $n_{\varepsilon^-,j}^\gamma$, $n_{\varepsilon^-,j}^{\bar\gamma}$ имеют тот же смысл, что и величины n_ε^i, $n_{\varepsilon^-}^i$

Затем для всех x_j определяется наибольшая разность $t_{ji}=t_j-t_i$ и, если она больше нуля, то переменная x_i заменяется на x_j, имеющую $t_{ji}=\max$.

Эта процедура проводится для переменных, соответствующих всем столбцам переходной таблицы, включая и новые переменные, заменившие первоначальные, до тех пор, пока все переменные в соответствии со значениями критериев t не останутся на своих местах.

Выбор оптимального выходного элемента

Простота реализации структуры какого-либо выхода, в том случае если он не реализуется одним выходным элементом, зависит от сложности реализации функций, подаваемых на его входы. Эта реализация будет, очевидно, тем сложнее, чем в большей степени переходная таблица будет отличаться от приведенной к данному выходному элементу. Поэтому процедура выбора оптимального выходного элемента состоит в том, что для каждого из элементов φ_i, входящих в набор Φ, строится переходная таблица и производится улучшение плана. Близость к заданной для данного выхода функции определяется по критерию

$$N_i = n^i_{\varepsilon} + n^i_{\bar{\varepsilon}},$$

где n^i_{ε} и $n^i_{\bar{\varepsilon}}$ имеют то же значение, что и выше.

6. ОПЕРАЦИИ НАПРАВЛЕННОГО ПОИСКА ОПТИМАЛЬНОЙ РЕАЛИЗАЦИИ СТРУКТУР

Последовательность операций направленного поиска рассмотрим на примере реализации одновыходного релейного устройства с 22 входными переменными, условия работы которого записаны в интервальной форме в табл. 2, *а*. Требуется синтезировать структуру устройства на наборе элементов „И“, „ИЛИ“, „НЕ“. Число входов элементов „И“ и „ИЛИ“ ограничено и равно $q(\varphi) = 6$. На переменные $x_{16}, x_{17}, \ldots, x_{22}$ наложены ограничения в том смысле, что они могут подаваться на входы элементов только в неинверсной форме.

В качестве выходного элемента примем вначале элемент „ИЛИ“. Для него $|r_\varphi| = 1$, $\gamma = 1$ и $\varepsilon = 1$. Следовательно, реализуемой стороной таблицы состояний является подмножество M_ε и нереализуемой стороной — подмножество $M_{\bar{\varepsilon}}$.

Построим переходную таблицу. Для этого вычислим значения критериев s_γ и $s_{\bar{\gamma}}$ (табл. 2, *б*). В связи с указанными ограничениями для переменных $x_{16}, x_{17}, \ldots, x_{22}$ при выборе оптимальной переменной для них учитывается только значение критерия s_1. При подсчете критериев прочерк в таблице состояний не учитывается.

В соответствии со значением критерия s выбираем переменную x_4 и заполняем её значениями первый столбец переходной таблицы (табл. 3, *а*). Как видно из этой таблицы, переменная x_4 непротиворечиво реализует состояния 3 и 7 из M_ε. Для определения следующей наиболее оптимальной переменной производим снова подсчет критериев s_1 и s_0, но уже

Таблица 2

				x_4		x_6		x_8		x_{10}			x_{13}	x_{14}	x_{15}	x_{16}	x_{17}		x_{19}		x_{21}	
1	—	1	0	—	1	—	—	1	—	—	1	0	—	1	1	0	—	1	0	1	0	0
2	0	0	—	0	1	0	0	—	0	—	—	—	—	1	0	1	0	—	—	1	1	1
3	1	—	—	1	1	1	—	0	—	0	0	—	0	1	—	—	—	—	1	1	1	1
4	—	—	0	—	0	—	0	1	0	1	—	—	1	—	—	0	0	0	0	0	0	0
5	0	—	—	0	—	0	1	—	—	—	0	0	0	0	0		—	—	1	1	—	0
6	1	—	0	—	—	—	0	0	—	0	0	—	—	0	1	—	—	—	—	—	1	0
7	—	0	1	1	0	—	—	1	—	0	—	1	—	—	—	0	—	0	—	—	0	0
8	—	1	—	—	1	0	0	1	0	—	0	—	—	0	0	0	0	—	—	—	0	0
9	0	0	—	0	0	—	—	1	1	—	1	1	1	—	0	0	0	0	0	0	0	0
10	1	0	—	0	1	—	—	—	—	—	—	0	—	—	0	1	—	—		1	1	—
11	0	—	0	0	—	0	0	1	1	—	—	1	0	1	—	—	1	0	—	1	—	—

а

s_1^1 | 1 1 — 1,2 0,6 0,5 0,5 — — — — 0,5 0,5 —1 0,5 0 0,6 0,0 0,8 0,7 0,6 |
s_0^1 | 0,6 0 0,7 0 0 0 0 1 1,2 0 0,8 1 0,6 0,7 0 — — — — — — — | x_4

б

s_1^2 0,6 1 — 0 0,7 0 0,5 — — — — 0 0,5 — 1 0,5 0 0,6 0,5 0,7 0,6 0,5 |
s_0^2 0,6 0 0,7 0 0,5 0 0 0,1 1,2 0 0,7 1 0,5 0,7 0 — — — — — — — | $\bar{x}_9$

в

s_1^3 | 0,6 0,6 — 0 0,5 0 0,5 — — — — 0 0 — 1 0 0 0,5 0,5 0,5 0,5 0 |
s_0^3 | 0,5 — 0,6 0 — 0 0 0,6 — 0 0,6 1 0,5 0,6 0 — — — — — — — | $\bar{x}_{12}$

г

t_1 | — 1 0 — 3 — 0 — 2 — 2 — 1 — — — — 0 0 0 0 — — — 1 0 (1) |
t_0 | — 1 — 2 — 1 — 2 — 1 — 1 0 — 1 $\boxed{2}$ — 0 — — 0 — — — — — — — |

д

t_1 — 1 — — 0 — (1) — 1 — — 0 —2 — 0 1 — — 1 1 0 — —
t_0 — 1 — — 2 — 0 0 — — — 1 $\boxed{1}$ 0 1 — 1 — — — — — — —

е

без учета реализованных состояний 3 и 7 (табл. 2, *в*). Поочередное осуществление этой операции выделяет переменные $\bar{x}_9$, $\bar{x}_{12}$ и x_1 (табл. 2, *в* и *г*). Поскольку после заполнения четвертого столбца переходной таблицы значениями переменной x_1 ранги всех строк ее оказались равными или большими $[r_\varphi]$, заполняем остальные столбцы нулями и на этом построение переходной таблицы заканчиваем. Она не является приведенной для элемента „ИЛИ", так как ранг строки 10 из $H_{\bar{\varepsilon}}$ не равен нулю.

Перейдем к операции улучшения плана. Так как переменная x_4 не реализует ни одного состояния из $M_{\bar{\varepsilon}}$, то операции по улучшению плана начнем с переменной $\bar{x}_9$. Удалим из переходной таблицы соответствующий столбец и подсчитаем для оставшихся столбцов ранги строк (табл. 3, *б*, столбец $r_{\bar{x}_9}$). Подсчитаем значения t_j^1 и t_j^0 при замене переменной $\bar{x}_9$ на все другие переменные (табл. 2, *д*). Наибольшее значение величина $\bar{t}_j$ имеет для переменных x_{20} и x_{22} (t_j 1). Так как переменная $\bar{x}_9$ имеет t_i 2, то обе эти переменные не улучшают плана. Аналогичная процедура, проведенная для столбца с переменной $\bar{x}_{12}$ (табл. 2, *е*) показывает, что и для этой переменной улучшение плана не существует. Переменная x_1 выбиралась последней и, следовательно, для совокупности предыдущих переменных является наилучшей. Поэтому для нее нет необходимости проводить операции улучшения плана.

Таким образом переходная таблица 3, *а*, для элемента „ИЛИ" является наилучшим планом реализации функции, заданной таблицей 2, *а*.

Оценка этого элемента в качестве выходного имеет величину

$$N_{\text{„ИЛИ"}} = 0 + 1 = 1.$$

Рассмотрим теперь элемент „И" в качестве выходного для функции табл. 2, *а*. Для этого элемента $[r_\varphi] = 1$, $\gamma = 0$ и $\varepsilon = 0$. Реализуемым является подмножество $M_{\bar{\varepsilon}}$ и нереализуемым — подмножество M_ε Наиболее оптимальной является та же переменная x_4, которая полностью реализует все состояния из M_ε.

Заполняя все остальные столбцы единицами, получим переходную таблицу 3, *в*. Оценка плана реализации, даваемого этой таблицей, равна

$$N_{\text{„И"}} = 0 + 6 = 6.$$

Выберем поэтому в качестве выходного для реализации функции табл. 2, *а*, элемент „ИЛИ".

В связи с противоречивой реализацией функции табл. 2, *а*, переменными $\bar{x}_9$, $\bar{x}_{12}$ и x_1 их необходимо заменить функциями. Состав таблиц состояний для этих функций дает размещение нулей, единиц и прочерков в соответствующих столбцах переходной таблицы. Так, например, для функции, заменяющей переменную x_9, рабочими будут состояния 2, 4 и 8, нерабочими 9 и 11 и безразличными (имеющими прочерк) 1, 3, 5, 6, 7 в M_ε и 10 в $M_{\bar{\varepsilon}}$.

Сложность реализации этих функций может быть уменьшена с помощью замены констант на незаполненных входах элемента на функции. В качестве рабочих для таблиц состояний этих функций берутся состояния первоначальной таблицы состояний, для которых ранг в строках переходной таблицы равен $[r_\varphi]$ и меньше, а в качестве нерабочих — все состояния, для которых ранг равен $[r_\varphi]$—2 и больше. Реализация такой функции производится на одном элементе. При этом из всей совокупности элементов Φ выбирается такой, который реализует наибольшее число состояний из M_ε. После выбора такого элемента определяются значения реализуемой им функции и подставляются в переходную таблицу вместо константы. Одновременно к переменным первоначальной таблицы добавляется переменная, учитывающая выход этого элемента для использования его при реализации других функций с целью образования связных структур. Кроме того, для устранения избыточности все состояния, в которых после внесения значений очередной функции ранг строк в переходной таблице в M_ε в реализованных столбцах становится равным или большим $[r_\varphi]$, а в $M_{\bar{\varepsilon}}$ равным или меньшим $[r_\varphi]$ $1+f$ (где f — число нереализованных еще функций в столбцах переходной таблицы), удаляются соответственно из подмножеств $M_в$ и $M_{\bar{\varepsilon}}$ в таблицах состояний для функций, доопределяющих переменные в столбцах переходной таблицы.

Совокупность функции, доопределяющих переменные переходной таблицы, можно рассматривать по отношению к выходному элементу „ИЛИ“ как некоторую многовыходную функцию. Поэтому дальнейшие операции по реализации структуры выхода сводятся к определению последовательности реализации доопределяющих функций и поочередной реализации их в этой последовательности с учетом возможной связности. По отношению к каждой доопределяющей функции при этом повторяются все рассмотренные выше операции.

После реализации каждой из доопределяющих функций ее значения вносятся в переходную таблицу, а функции выходов всех элементов, содержащихся в реализующей ее структуре, вводятся в качестве дополнительных переменных для использования при синтезе структур последующих функций. При этом учитываются ограничения по числу разветвлений входов. Одновременно производятся указанные выше операции удаления из подмножеств $M_в$ и $M_{\bar{\varepsilon}}$ состояний, для которых ранг строк в переходной таблице свидетельствует об их реализации.

Таблица 3

			$\bar{x}_{12}$				
1	—	—	1	—	0	0	1
2	0	1	—	0	0	0	1
3	1	—	—	1	0	0	2
4	—	1	—	—	0	0	1
5	0	—	1	0	0	0	1
6	—	—	—	1	0	0	1
7	1	—	0	—	0	0	1
8	—	1	—	—	0	0	1
9	0	0	0	0	0	0	0
10	0	—	1	1	0	0	3
11	0	0	0	0	0	0	0

а

1	0
0	1
2	2
0	1
1	0
1	1
1	1
0	1
0	0
2	1
0	0

б

—	1	1	1	1	1
0	1	1	1	1	1
1	1	1	1	1	1
—	1	1	1	1	1
0	1	1	1	1	1
—	1	1	1	1	1
1	1	1	1	1	1
—	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1
0	1	1	1	1	1

в

В рассматриваемом примере функции, заменяющие константы, соответствующие столбцам 5 и 6 в табл. 3, *а*, оказались равными

$$f_5 = x_1\bar{x}_9 \text{ и } f_6 = x_2\bar{x}_3.$$

Для доопределяющих функций была определена следующая последовательность их реализации:

$$f_{x_{12}},\ f_{x_1},\ f_{\bar{x}_9}$$

Соответствующие функции получились

$$f_{\bar{x}_{12}} = \bar{x}_1 x_{12};\ f_{x_1} = x_1 x_{16};\ f_{\bar{x}_9} = x_9(x_2+x_5).$$

Связность между ними оказалась нерациональной. Функция табл. 2, *а* оказалась таким образом реализованной в следующем виде:

$$F = x_4 + x_9(x_1 + x_2 + x_3) + x_1 x_{12} + x_2 x_3.$$

7. ДРУГИЕ ВОЗМОЖНОСТИ АЛГОРИТМА

В предыдущем разделе алгоритм был описан применительно к задаче синтеза одновыходных структур на элементах „И“, „ИЛИ“, „НЕ“ с учетом ограничений по числу входов.

Ниже кратко рассматриваются возможности применения алгоритма для синтеза структур на различных элементах и с различными ограничениями, а так же для синтеза многовыходных структур.

Синтез многовыходного релейного устройства

Примем, что синтезируемое многовыходное релейное устройство задано таблицей состояний, в которой каждому состоянию на входе соответствуют значения на всех выходах (см. табл. 4).

Таблица 4

									x_9						x_{15}	F_1	F_2	F_3
1	0	0	—	0	0	—	—	1	1	—	1	1	1	—	0	0	0	—
2	—	1	0	—	1	—	—	1	—	—	1	0	—	1	1	1	1	0
3	0	0	—	0	1	0	0	—	0	—	—	—	—	1	0	1	1	1
4	1	—	—	—	1	1	—	0	—	0	0	—	0	1	—	1	—	1
5	—	—	0	—	0	—	0	1	0	1	—	—	1	—	—	1	0	—
6	0	—	—	0	—	0	1	—	—	—	0	0	0	0	0	1	1	0
7	1	—	—	—	0	—	—	0	0	0	—	—	—	1	—	—	—	1
8	1	0	—	0	1	—	—	—	—	—	—	0	—	—	0	0	1	—
9	1	—	0	—	—	—	0	1	—	0	0	—	—	0	1	1	—	0
10	—	1	1	—	—	—	—	1	0	1	1	0	—	1	—	—	0	—
11	0	—	1	0	—	0	0	1	1	—	—	1	0	1	—	0	1	—
12	—	0	1	1	0	—	—	1	—	0	—	1	—	—	—	1	—	0
13	—	1	—	—	1	0	0	1	0	—	0	—	—	0	0	1	—	0
14	—	—	—	—	0	1	1	1	1	—	1	0	0	1	—	—	1	0

Для применения к синтезу такого релейного устройства изложенного выше алгоритма необходимо:

1) определить предварительную сложность реализации каждого выхода по критерию I;

2) выбрать выход с наименьшим значением I;

3) провести синтез структуры выбранного в пункте 2) выхода;

4) после реализации выбранного выхода, перейти к пункту 2) для выбора следующего. При этом, в исходную таблицу состояний следует добавить новые переменные $x_{n+1}, x_{n+2}, \ldots$, соответствующие функциям, реализуемым выходами выходного и всех внутренних элементов построенной части структуры.

В табл. 4 приведена таблица обобщенных состояний для релейного устройства, содержащего 15 входов и 3 выхода. Оценка сложности реализации показала, что $I_1 = 9{,}35$, $I_2 = I_3$ 7,15.

Реализация устройства начата со второго выхода. После ее завершения был реализован третий выход, а затем первый.

Для реализации первого выхода к таблице состояний добавлены переменные $x_{16}, x_{17}, \ldots, x_{22}$, соответствующие функциям, реализованным на

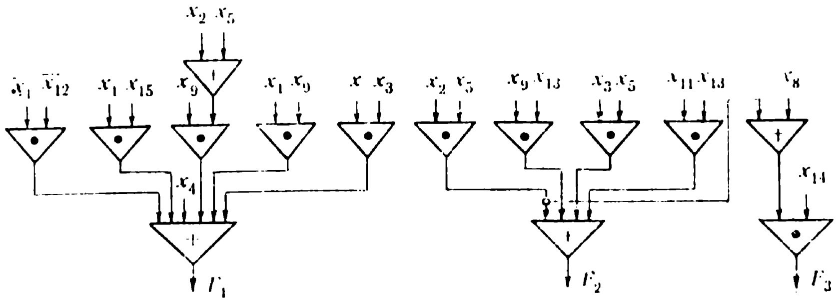


Рис. 3

выходах элементов готовой части структуры. Реализация первого выхода была рассмотрена при описании алгоритма.

Структура, реализующая функцию табл. 4, приведена на рис. 3.

Синтез релейных устройств с минимальным числом инверторов на входах

Если входные переменные релейного устройства не имеют инверсных значений, то необходимо для тех переменных, которые подаются с инверсиями, ставить на входах элементов инверторы, или же использовать в качестве таковых некоторые функциональные элементы. Так как в последнем случае функциональные свойства элементов используются не полностью, то это существенно ослабит эффект, получаемый от минимизации структуры.

Поэтому при синтезе структуры весьма важно иметь возможность определять такую последовательность выбора элементов и переменных на их входах, которые обеспечивали бы получение структур с наименьшим числом инверторов на входах. В рассмотренном выше методе это достигается путем выбора переменных по критерию s_γ. Очевидно, что в этом случае в первую очередь будут выбираться переменные, используемые с неинверсными значениями, что и дает минимизацию числа инверторов на входах структуры.

Рассмотрим реализацию некоторого одновыходного релейного устройства, заданного таблицей состояний 5 один

Таблица 5

					x_8		x_4			
1′	0	0	0	1	0	0	0	1	0	1
2′	0	0	0	0	1	1	0	1	1	0
3′	0	0	0	1	0	0	0	0	0	1
4′	0	0	0	1	1	1	1	0	1	1
5′	0	0	1	0	1	0	1	0	1	0
6′	0	1	0	0	1	1	0	0	1	0
7′	0	1	0	1	1	1	0	0	1	0
8′	0	1	1	0	1	1	0	1	0	1
9′	0	1	1	1	0	0	0	0	1	0
10′	1	0	0	1	1	0	0	1	0	1
11′	1	0	1	0	1	0	1	0	0	0
12′	1	0	1	1	0	0	0	0	0	0
13′	1	0	0	1	1	0	0	0	0	0
1°	0	1	0	1	1	0	0	1	1	0
2°	0	1	0	1	1	1	1	0	0	0
3°	0	1	1	0	0	0	0	1	1	1
4°	0	1	1	0	1	0	1	0	0	0
5°	0	1	1	1	1	1	1	1	1	1
6°	1	0	0	0	1	0	1	1	1	1
7°	1	0	0	1	1	1	0	1	0	1
8°	1	1	1	0	0	0	0	1	1	0
9°	1	1	1	0	0	1	0	1	1	0
10°	0	1	1	1	1	1	0	1	1	1

раз с выбором переменных по критерию s и другой раз по критерию s_1. В обоих случаях будем реализовать структуру на четырехвходовых элементах „ИЛИ-НЕ“.

Для этих элементов $[r_\varphi]=1$, $\gamma=1$ и $\varepsilon=0$. В соответствии с этим $M_\varepsilon=M_0$, $M_{\bar{\varepsilon}}=M_1$. Алгоритм синтеза структуры ничем не отличается от

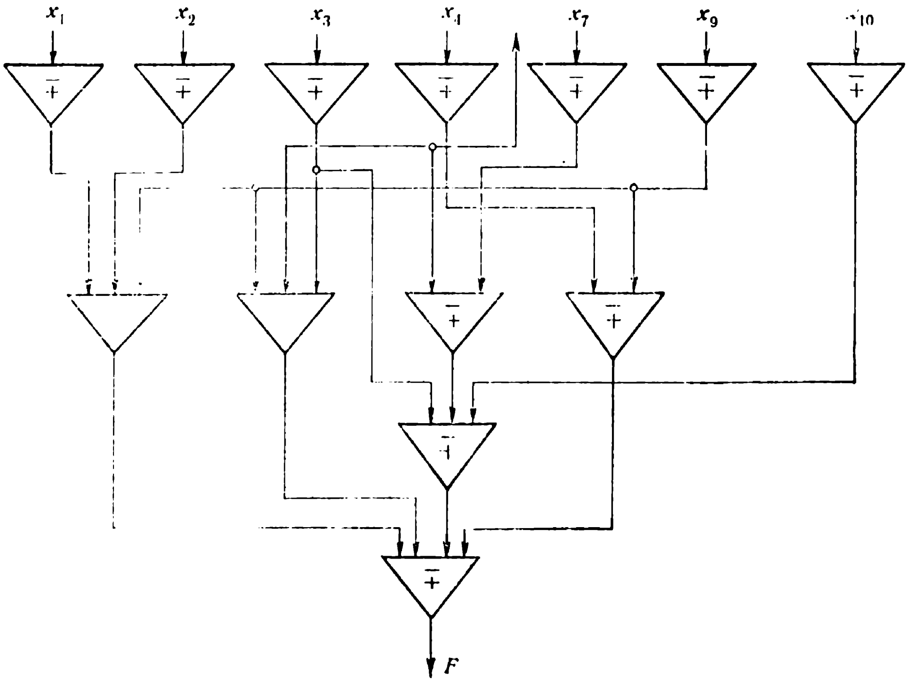


Рис. 4

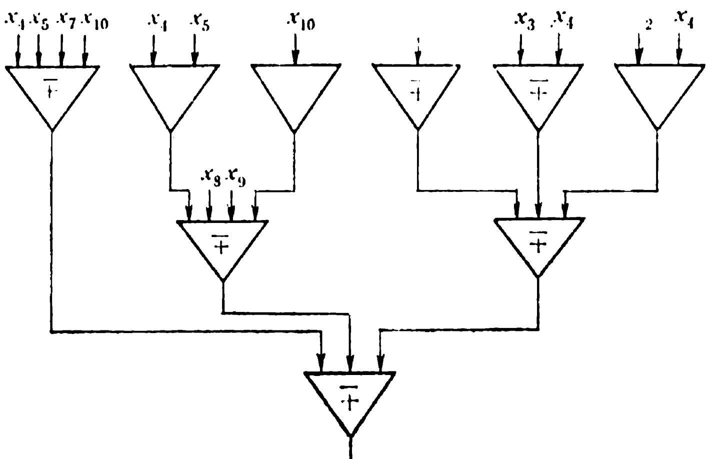


Рис. 5

изложенного выше. Полученная структура представлена на рис. 4. Она требует 13 элементов, из них 7 элементов (т. е. больше половины) используются в качестве инверторов.

Будем теперь выбирать переменные по критерию s_1. Исключение будем делать только для тех случаев, когда значения доопределяющей функции полностью совпадают с инверсным значением какой-либо из переменных. В этом случае ее можно заменить на эту переменную, используя инвертор, так как для построения доопределяющей функции все равно требуется не менее одного элемента.

Полученная в этом случае структура представлена на рис. 5. Она требует 10 элементов „ИЛИ-НЕ“ (сокращение по сравнению с приведенной ранее на 24 %).

Таблица 6

x_{10}	x_9	x_8	x_7	x_6	x_5	x_4	x_3	x_2	
0	0	0	1	0	0	0	0	0	1
0	0	0	1	1	1	1	0	1	1
0	0	1	0	1	0	1	0	1	0
0	1	0	0	1	1	0	0	1	0
0	1	0	1	1	1	0	0	1	0
0	1	1	0	1	1	0	1	0	1
0	1	1	1	0	0	0	1	1	0
1	0	0	1	1	0	0	1	0	1
1	0	1	0	1	0	1	0	0	0
1	0	1	1	0	0	0	0	0	0
1	0	0	1	1	0	0	0	0	0
0	1	0	1	1	0	0	1	1	0
0	1	0	1	1	1	1	0	0	0
0	1	1	0	0	0	0	1	1	1
0	1	1	0	1	0	1	0	0	0
0	1	1	1	1	1	1	1	1	1
1	0	0	0	1	0	1	1	1	1
1	0	0	1	1	1	0	1	0	1
1	1	1	0	0	0	0	1	1	0

Синтез релейных устройств на мажоритарных элементах

Мажоритарный элемент имеет $[\tau_q]$ $\frac{q(\varphi)+1}{2}$, $\gamma = 1$, $\varepsilon = 1$, $M_\varepsilon = M_1$, $M_{\bar{\varepsilon}} = M_0$. Алгоритм синтеза для них ничем не отличается от описанного в разделе 5 за исключением того, что в этом случае на входах выходного элемента нужно выбрать такой набор переменных или функций, который для каждого из рабочих состояний давал бы число единиц, равное или большее $\frac{q(\varphi)+1}{2}$, а для нерабочих состояний равное или меньшее $\frac{q(\varphi)+1}{2} - 1$.

Результат синтеза одновыходного релейного устройства, заданного таблицей состояний 6 на пяти-входовых мажоритарных элементах, приведен на рис. 6.

Синтез релейных устройств с учетом некоторых топологических ограничений

В ряде случаев при синтезе релейных устройств наряду с другими возникают ограничения топологического характера, связанные с взаиморасположением и взаимными связями элементов в среде. Это имеет место в частности при реализации структур в однородных средах.

На рис. 7 приведен пример реализации функции заданной таблицей состояний 6 в однородной среде, состоящей из чередующихся элементов „ИЛИ-НЕ“ и элементов, реализующих пересечение проводов („крест без точки“). Элемент „ИЛИ-НЕ“ среды может иметь максимум три входа и один выход. Ввод входных переменных и вывод выходных может осуществляться только по периметру среды.

Рассмотренный выше алгоритм по существу остается тем же, за тем исключением, что на каждом из этапов синтеза рассматриваются возможные размещения в среде элементов, реализующих доопределяющие функции.

В целом алгоритм состоит из следующих операций:

1. Построить переходную таблицу для трехвходового элемента „ИЛИ-НЕ“.

2. Разместить полученный элемент в однородной среде.

3. Из условий размещения выходного элемента определить наибольшее возможное число входов элемента доопределяющей функции. Перейти к п. 1 и т. д. до тех пор, пока не будет размещена вся структура.

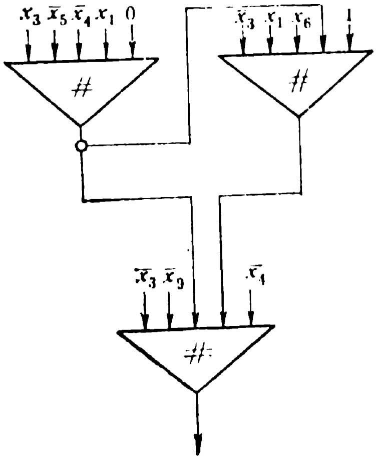


Рис. 6

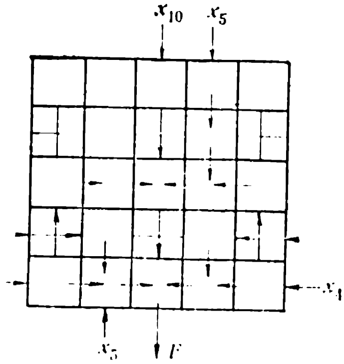


Рис. 7

Синтез релейных устройств, свободных от состязаний в исполнительных цепях

Приведенный ранее алгоритм не учитывает возможность появления состязаний в связи с наличием различных задержек при прохождении воздействий по структуре и в соответствии с этим разновременным появлением их на входах выходного элемента. Это может привести к неправильному функционированию релейного устройства в момент изменения

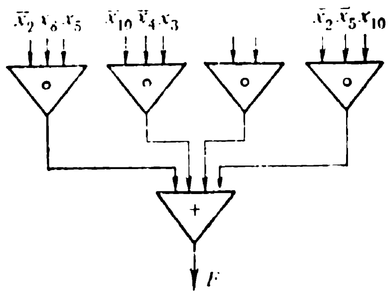


Рис. 8

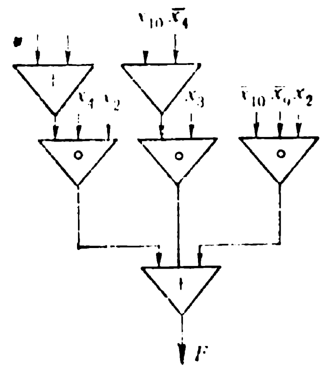


Рис. 9

входных переменных, т. е. к появлению кратковременных переходов типов $1 \to 0 \to 1$ или $0 \to 1 \to 0$.

Для устранения таких переходов обычно вводятся избыточные элементы, что может свести к нулю эффект, полученный при минимизации.

Рассмотренный выше алгоритм при некотором дополнении его позволяет проводить синтез структуры с учетом устранения состязаний. Существо дополнений заключается в том, что выбор переменных, подаваемых на входы выходного элемента, производится таким образом, что при реализации состояний для них в первую очередь выбираются переменные, для которых отсутствуют состязания, что учитывается при подсчете критериев s_γ и $s_{\bar{\gamma}}$.

На рис. 8 приведена реализация функций, заданной табл. 5, на четырехвходовых элементах в базисе „И“, „ИЛИ“, „НЕ“ без устранения состязаний. Анализ полученной структуры показывает, что в ней возможны состязания при изменении переменной x_{10}, связанном с переходом из состояний 3, 5 в состояния 10, 11, 12, 13.

При обычных методах устранения состязаний с помощью введения избыточности потребовалось бы ввести дополнительно два элемента „И“, реализующие функции $x_2x_3x_4x_5$ и $x_2x_5\bar{x}_9$. С учетом элемента „ИЛИ“, необходимого для размножения входов выходного элемента, необходимо ввести дополнительно три элемента. Сложность реализации при этом составит 8 элементов.

На рис. 9 приведена структура, реализующая функцию табл. 5, полученная с учетом при синтезе указанных выше особенностей выбора переменных. Устранение состязаний в процессе синтеза позволило реализовать структуру на шести элементах, т. е. на 2 элемента, или на 25% проще предыдущей реализации.

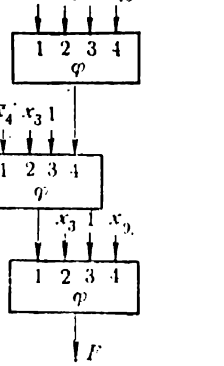


Рис. 10

Синтез релейных устройств на произвольных элементах, заданных таблицами истинности

В приведенном ранее алгоритме для построения переходной таблицы использовались понятия $[\tau_\varphi]$, γ и ε, которые применимы для всех элементов, реализующих монотонные функции, либо функции, которые можно свести к монотонным инвертированиям некоторых входных переменных.

Построение переходной таблицы для произвольных элементов становится возможным после введения понятий рабочей (h) и нерабочей (q) характеристических функций элемента.

В табл. 7 приведена таблица истинности некоторого элемента φ, на котором следует построить структуру, реализующую функцию, заданную таблицей состояний 6. Здесь же приведены аналитические выражения функции h и q элемента.

Таблица 7

1		3	4
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	1	0	1
1	1	1	0
1	0	0	0
1	0	0	1
1	0	1	0
1	0	1	1
1	1	0	0
1	1	1	1

$h = a + bcd + bcd$

$ab + acd + ccd$

Отличие алгоритма для синтеза структуры заключается в том, что для построения переходной таблицы выбираются такие переменные, чтобы для каждого состояния M_1 существовал импликант из h, который принимает в нем значение единицы и для каждого состояния M_0 существовал импликант из q, который в нем принимал значение единицы. Выбор переменных осуществляется по критериям s_γ и $s_{\bar{\gamma}}$, а порядок заполнения — из условия реализации, в первую очередь, наиболее короткими импликантами.

Структура, полученная в результате применения этих операций, представлена на рис. 10.

ЛИТЕРАТУРА

1. Гаврилов, М. А. Эвристические методы структурного синтеза релейных устройств. Finite Automata and Switching Systems. Сборник докладов на IV конгрессе ИФАК. Техническая секция 27. Изд. NOT, Варшава, 1969.
2. Шеломов, Л. А. Критерии сложности булевых функций. В сб.: Проблемы кибернетики, вып. 17.
3. Гаврилов, М. А., В. М. Копыленко. Метод минимизации структур бесконтактных релейных устройств на функционально полных элементах. В сб.: Теория дискретных автоматов. Изд. „Зинатне", 1967.
4. Гаврилов, М. А., В. М. Копыленко. Синтез структуры бесконтактных релейных устройств. Bulletin Mathématique, **10 (58)**, № 3, 1966, 227—259.

Поступила 23. II. 1970 г.

ИНЖЕНЕРНИ МЕТОДИ ЗА СИНТЕЗ НА КОМБИНАТОРНИ СТРУКТУРИ С ПРОИЗВОЛНИ ЕЛЕМЕНТИ

М. А. Гаврилов и В. М. Копиленко

(Резюме)

В работата се разглежда следната задача: Зададени са условията за работа на недоопределена логическа система и функционално пълен набор от елементи, чрез които трябва да се реализира структурата на логическата система. Иска се да се построи логическа структура, несъдържаща излишни елементи, която е близка до оптималната от гледна точка на необходимите елементи.

Като се прилагат някои евристични съображения за целенасочено търсене, изгражда се алгоритъм за решаване на поставената задача.

ENGINEERING METHODS FOR A SYNTHESIS OF COMBINED STRUCTURES OF ARBITRARY ELEMENTS

M. A. Gavrilov and V. M. Kopilenko

(Summary)

The paper treats the following problem: given are the operation conditions of an underdeterminate logical system and a functionally complete set of elements, by means of which the structure of the logical system must be realized. A logical structure must be constructed, which would not contain useless elements and would be near to the optimal one from the point of view of the necessary elements.

By applying certain heuristic considerations about a purposeful search an algorithm is built for solving of the set problem.