

УДК 681.142.1.01.

СИНТЕЗ КОМБИНАЦИОННЫХ СХЕМ С МИНИМАЛЬНОЙ
ГЛУБИНОЙ ПРЕОБРАЗОВАНИЯ

В.П. Чистов

I. Постановка задачи

По мере развития технологии производства вычислительных машин и интегрализации логических элементов все большее значение приобретает задача синтеза схем, обладающих максимальным быстродействием, при одновременном выполнении тех или иных конструктивных ограничений. Максимальное быстродействие достигается минимизацией глубины преобразования схемы по всем или некоторым из её выходных полюсов.

Под глубиной преобразования схемы по данному выходному полюсу будем понимать максимальное число элементов (задержек), включенных последовательно между этим полюсом и любым из входных полюсов схемы. Будем считать, что комбинационная схема обладает минимальной глубиной, если глубина преобразования по каждому из её выходных полюсов минимальна.

Всякий логический элемент, входы которого связаны лишь с входными полюсами, будем относить к первому уровню преобразования. Логический элемент, входы которого связаны с элементами первого уровня и входными полюсами, будем относить ко второму уровню преобразования. В общем случае к K -му уровню преобразования будем относить всякий логический элемент, хотя бы один вход которого связан с элементами $(K-1)$ -го уровня. Связь между элементами K -го уровня и элементами $(K-1)$ -го уровня будем называть связью единичной глубины. Соответственно связь между элементами K -го уровня и входными полюсами комбинационной схемы будем называть связью глубины K .

Одним из конструктивных ограничений может служить требование реализации комбинационной схемы с глубиной связей, не превышающих некоторого числа λ .

При проектировании отдельных схем инженер не ограничивается рассмотрением одного варианта, а интересуется совокупностью возможных решений в различных базисах при различных ограничениях.

Пусть T - глубина преобразования, Q - число элементов в схеме, и λ - допустимая глубина связей между элементами суть параметры, характеризующие схемное решение. В пространстве этих параметров схемные решения будут представлены некоторой совокупностью точек, знание которой позволит инженеру выбрать наиболее подходящий вариант. Поэтому следует так строить алгоритмы синтеза, чтобы с их помощью можно было получить указанную характеристику, если не в любом базисе, то по крайней мере в некотором классе базисов.

В данной работе дается описание метода синтеза схем с минимальной глубиной преобразования, предназначенного для использования на универсальной ЦВМ.

Под правильной комбинационной схемой будем понимать [1] произвольную композицию логических элементов без петель, где каждый логический элемент может иметь несколько входов и выходов.

Пусть F - требуемое логическое преобразование n входных сигналов в m выходных - задано таблицей соответствия, состоящей из ℓ строк.

Т а б л и ц а I

| Строка | x_1 | x_2 | x_3 | ... | x_n | y_1 | y_2 | y_3 | ... | y_m |
|--------|-------|-------|-------|-----|-------|-------|-------|-------|-----|-------|
| 1 | 0 | 1 | 0 | ... | 1 | 1 | 1 | 0 | ... | 0 |
| 2 | 1 | 1 | 0 | | 0 | 0 | 1 | 0 | | 1 |
| 3 | 0 | 1 | 1 | | 1 | 1 | 1 | 1 | | 0 |
| ⋮ | | | | ⋮ | | | | | ⋮ | |
| ℓ | 1 | 1 | 0 | | 0 | 0 | 1 | 1 | | 1 |

Таблицу соответствия можно рассматривать как описание связей между словами, возникающими последовательно во времени на входах и выходах схемы. В этом смысле последовательности входных и выходных символов на каждом из полюсов схемы можно считать функциями времени или функциями последовательности натуральных чисел 1, 2, 3, ...

Функции длины ℓ , образуемые столбцами таблицы соответствия, будем называть входными и выходными, обозначая их через $X_1, X_2, X_3, \dots, X_n$ и $Y_1, Y_2, Y_3, \dots, Y_m$, соответственно.

Запись функций в виде последовательности символов будем называть канонической формой записи. Если в качестве символов используются 0 и 1, то каждой канонической форме можно поставить в соответствие некоторый двоичный или восьмеричный код.

Входные функции будем относить к нулевому уровню преобразования, а функции, формируемые на выходах элементов K -го уровня, будем называть функциями K -го уровня.

Комбинационную схему, осуществляющую преобразование F , будем считать избыточной, если на выходах любой пары логических элементов реализуются различные функции, не совпадающие с входными, и в ней отсутствуют такие связи между элементами, исключение которых не нарушает указанного преобразования.

Поставим следующую задачу.

Пусть задан набор логических элементов. Требуется построить избыточную комбинационную схему минимальной глубины, реализующую преобразование F при условии, что глубина связей между элементами не превышает некоторого значения λ .

Известно, что преобразование F осуществимо, если заданный набор логических элементов является функционально полным. Условия же функциональной полноты найдены в предположении реализации между элементами связей любой глубины. Однако ограничение глубины связей может привести к нереализуемости некоторых преобразований. Как будет видно из дальнейшего, этот факт легко обнаруживается на первом же этапе синтеза.

Ниже изложен алгоритм построения всего множества избыточных комбинационных схем, удовлетворяющих указанным требованиям. Последовательный или выборочный анализ этих схем позволяет отобрать схемы с минимальным индексом $S = \sum_{j=1}^r s_j n_j$, принятым за критерий качества схемы. Здесь s_j , n_j - индекс и число элементов типа j , а r - общее число типов элементов.

В общем случае индекс схемы (элемента) может отражать её

стоимость, вес, надежность и т.п.

Схемы с минимальным значением S будем называть оптимальными.

Пусть P - множество всех избыточных схем, реализующих любые логические преобразования над n функциями длины ℓ ; F - множество всех избыточных схем, реализующих заданное преобразование над теми же входными функциями; T - подмножество схем множества F , обладающих минимальной глубиной преобразования. Тогда под полным перебором будем понимать перебор всех схем из множества P . Соответственно, под F и T - переборами будем понимать переборы всех схем из множеств F и T . Отбор всех оптимальных схем с минимальной глубиной преобразования связан с реализацией T - перебора, который, хотя и значительно уже полного перебора, тем не менее резко возрастает по мере увеличения n и ℓ .

Алгоритм решения поставленной задачи состоит из двух этапов. На первом этапе составляются таблицы функций, реализуемых на различных уровнях, объединенные в общую "таблицу уровней". На втором этапе эта таблица используется для построения схем, из которых сортировкой выделяются оптимальные или приемлемые варианты.

2. Основной алгоритм синтеза.

Построение таблиц уровней

Функции, реализуемые на первом уровне, можно получить путем поочередного выполнения логических операций над всеми возможными комбинациями входных функций. Упорядочение полученного множества функций путем исключения повторяющихся и равных константам дает искомую таблицу функций первого уровня. Аналогично строятся таблицы функций для всех последующих уровней.

Отметим, что каждая из функций K -го уровня реализуется с использованием хотя бы одной функции $(K-1)$ -го уровня. Таблицы уровней представляют самостоятельный интерес, так как по ним можно составить представление о возможном быстродействии логических схем, реализующих заданное преобразование.

Резкое снижение времени построения таблиц уровней и синтеза схем по этим таблицам может быть получено за счет использования матричной (картинной) логики [2], где логические операции выполняются не над каноническими формами записи функций, а над их отображениями в матрицах. Это позволяет выполнять логичес-

кие операции одновременно над множествами логических переменных с последующим упорядочением результата. Матричные операции наиболее легко реализуются при синтезе схем из элементов типа "И-ИЛИ-НЕ" и "И-НЕ" с любым числом входов.

Число входов логического элемента "И" будем называть рангом конъюнкции r_c , а число входов элемента "ИЛИ" - рангом дизъюнкции - r_d . Для таких элементов таблицы уровней строятся с помощью матричных операций соединения (\cup), пересечения (\cap), вычитания (\setminus), прямого произведения конъюнктивного (Δ) и дизъюнктивного (∇) типа. Используя эти операции, можно получить следующие три основных типа таблиц уровней:

Расширенная таблица содержит все функции, реализуемые на каждом уровне. Алгоритм построения такой таблицы на базе элементов типа "И-ИЛИ-НЕ" состоит в следующем.

Пусть M - множество функций, образованных на λ предыдущих уровнях. Тогда множество функций, реализуемое конъюнктивной частью элементов на K -ом уровне, определяется следующим рекуррентным соотношением:

$$\begin{aligned} \text{при } r_c = 1 & \quad M_{K1} = M, \\ r_c = 2 & \quad M_{K2} = (M_{K1} \Delta M) \setminus M_{K1}, \\ r_c = t & \quad M_{Kt} = (M_{K(t-1)} \Delta M) \setminus M_{K(t-1)}, \end{aligned}$$

а после прохождения через дизъюнктивную часть:

$$\begin{aligned} \text{при } r_d = 1 & \quad M_{d1} = M_{K(t-1)}, \\ r_d = 2 & \quad M_{d2} = (M_{d1} \nabla M_{K(t-1)}) \setminus M_{d1}, \\ r_d = u & \quad M_{du} = (M_{d(u-1)} \nabla M_{K(t-1)}) \setminus M_{d(u-1)}. \end{aligned}$$

Множество $M_{d(u-1)}$ содержит все функции, реализуемые на искомом уровне. Если в приведенном алгоритме на каждом этапе вычислений исключить операцию вычитания, то множество $M_{d(u-1)}$ будет содержать все функции, реализуемые элементами с диапазоном рангов от 1 до r_c и от 1 до r_d .

Сокращенная таблица содержит на каждом уровне лишь те функции, которые не могут быть получены на более низких уровнях. Алгоритм её построения до получения $M_{d(u-1)}$ совпадает с алгоритмом построения расширенной таблицы, после чего операцией $M_{d(u-1)} \setminus M$ находятся все функции на искомом уровне сокращенной таблицы.

Полная таблица содержит все функции, реализуемые на каждом из уровней лишь избыточными логическими схемами.

дый уровень полной таблицы строится так же, как и для сокращенной таблицы с той лишь разницей, что исключаются не все функции, реализуемые на более низких уровнях, а лишь те, которые образуются через собственные же функции. Проверка проводится отдельно для каждой из повторяющихся функций.

Пусть исследуется какая-либо из функций, реализуемых на K -м уровне. С этой целью строится расширенная таблица, из которой на каждом уровне исключается исследуемая функция и все те, которые отсутствуют на данном же уровне в построенной части полной таблицы. Если при этом на K -м уровне будет получена исследуемая функция, то она реализуема избыточной схемой на K -м уровне. Детальное изложение алгоритма построения таблиц уровней приведено в [3]. Анализ таблицы уровней позволяет установить реализуемость заданного преобразования F .

П о с т р о е н и е к о м б и н а ц и о н н о й с х е м ы

С целью упрощения дальнейших рассуждений будем излагать алгоритм построения схем по таблице уровней для случая, когда заданный набор состоит из двухвыходных элементов типа "ИЛИ-НЕ", допускающих подключение к их выходам любого числа входов таких же элементов. Последнее допущение необходимо, так как излагаемый метод синтеза не учитывает ограничений по нагрузке выходов элементов.

Пусть на K -м уровне таблицы уровней содержится m_k выходовных функций. Одну из функций $(K-I)$ -го уровня, участвующую в формировании какой-либо функции K -го уровня, будем называть образующей. Вторую же функцию, участвующую в её формировании, будем называть парной независимо от того, с какого уровня она берется. Образующая функция и соответствующая ей парная составляют образующую пару.

Пусть совокупность всех входных и выходных функций составляет исходный список функций. Тогда под образующим списком на $(K-I)$ -м уровне будем понимать объединение исходного списка и всех функций, входящих в m_k образующих пар. А так как любая из выходных функций может формироваться различными образующими парами, то, перебирая все возможные комбинации m_k образующих пар, можно получить целый набор образующих списков на $(K-I)$ -м уровне. Графически это можно представить как разветвление исходного списка выходных функций

в дерево образующих списков - дерево синтеза.

Аналогично формируются образующие списки функций на $(K-I)$ -м уровне и при многовыходных элементах.

Для дальнейшего построения логической схемы можно принять любой из найденных образующих списков за исходный. Если повторять этот процесс на каждом уровне, то к моменту достижения нулевого уровня будет получен набор образующих списков, однозначно определяющих последовательность формирования выходных функций из функций нулевого уровня. Это и будет, по существу, структура искомой комбинационной схемы.

Изложенный метод синтеза логических схем с минимальной глубиной преобразования приводит к точному решению с точки зрения минимизации индекса схемы лишь при выполнении всего T -перебора*. При этом будут получены все возможные схемы с минимальной глубиной, в том числе - и схемы с минимальным индексом.

T -перебор можно организовать различными способами.

Один из таких способов состоит в последовательном формировании схем со все возрастающим функциональным номером. Функциональный номер схем определяется следующим образом.

Список функции будем считать упорядоченным, если входящие в него функции расположены в порядке возрастания их двоичных кодов. Последние будем называть функциональными номерами. Как указывалось выше, каждой функции K -го уровня соответствует набор образующих функций на $(K-I)$ -м уровне. Первую функцию из этого упорядоченного набора будем называть первой образующей функцией.

Каждой функции на K -м уровне и её первой образующей соответствует набор парных функций, из которого первую функцию аналогично будем называть первой парной. Будем считать, что первая образующая функция и первая парная составляют первую образующую пару. Вторая образующая пара будет составлена из первой образующей и второй парной и так далее до окончания перебора всех парных функций. Следующая образующая пара составляется из второй образующей функции и первой к ней парной.

Для элементов с равноценными входами из рассмотрения исключаются те образующие пары, которые получают перестановкой местами встречавшихся ранее образующей и парной функции.

* T -перебор реализуется при использовании полной таблицы уровней.

С этой целью к рассмотрению принимаются лишь те образующие пары, у которых парная функция, если она относится к тому же уровню, что и образующая, имеет функциональный номер, больший, чем образующая. Такие образующие пары будем называть правильными. Образующие пары нумеруются двумя функциональными номерами, из которых на первое место ставится функциональный номер образующей, а на второе — функциональный номер парной функции.

Совокупность всех образующих пар функций K -го уровня будем называть списком образующих пар. Функциональный номер списка образующих пар определяется совокупностью номеров самих образующих пар. Первый список состоит из первых образующих пар всех функций K -го уровня. Второй — из первых образующих пар всех функций K -го уровня, кроме последней; для последней принимается вторая образующая пара. И так далее.

Образующему списку приписывается тот же функциональный номер, что и списку образующих пар. Совокупность номеров списков образующих пар на различных уровнях, располагаемых в порядке убывания уровня, будем называть функциональным номером схемы.

Следует отметить, что введенный таким образом функциональный номер схемы не позволяет распознать изоморфизм синтезируемых схем, в связи с чем при выполнении T -перебора возможно построение эквивалентных схем, отличающихся функциональным номером из-за того, что входные функции подаются на различные их входы.

3. Модификация алгоритма построения схем

Сокращение T -перебора

Описанный выше T -перебор всех схем будем называть алгоритмом А.

Сокращенным перебором будем называть перебор логических схем из множества T , при котором построение образующих списков на каждом из уровней ограничено каким-либо правилом, а к дальнейшему синтезу принимаются лишь те, которые удовлетворяют некоторому критерию качества. Причем из числа последних может также производиться отбор по какому-либо из правил.

Ниже приводится краткое описание некоторых алгоритмов сокращенного перебора. Подробное рассмотрение этих алгоритмов выполнено в [4].

1. Метод пробных построений (алгоритм В).

Строятся несколько схем с различными образующими списками, причем выбор образующего списка на каждом из уровней производится по какому-либо правилу или случайно. Оцениваются полученные схемы и отбирается одна из них с наилучшим показателем качества. Для выбранной схемы фиксируется образующий список на $(K-1)$ -м уровне и повторяется аналогичная процедура вплоть до нулевого уровня. В результате первоначально принятая схема будет непрерывно либо улучшаться, либо оставаться неизменной. Рассмотренный алгоритм соответствует спуску к некоторому локальному минимуму критерия качества.

2. Синтез схем с минимальным расширением (алгоритм С).
Строятся все или несколько образующих списков на $(K-1)$ -м уровне. Построение списков может быть и случайным. Для дальнейшего построения принимается то направление (образующий список), у которого на $(K-1)$ -м уровне окажется минимальное число функций $(K-1)$ -го уровня. Аналогичная процедура повторяется и на более низких уровнях.

3. Метод случайных построений (алгоритм D) состоит в том, что на каждом уровне случайно строится лишь один образующий список, а задача синтеза повторяется заданное число раз.

4. В приведенных выше алгоритмах дополнительное сокращение T -перебора может производиться за счет построения образующих списков лишь на основе тех образующих пар, у которых парная функция находится на минимально возможном уровне. В этой модификации будут строиться так называемые схемы с максимальными связями. Опыт показывает, что такое сокращение, как правило, наиболее быстро приводит к схемам с минимальным числом элементов.

Синтез наложением

Так как рассматриваемый метод не критичен к числу входных функций, то представляется возможность вести синтез многовыходных схем путем последовательного синтеза одновыходных схем или схем с малым числом выходов на базе ранее полученных результатов. С этой целью все входные, выходные и промежуточные функции уже синтезированной части схемы условно принимают за входные, относят их к нулевому уровню и строят соответствующую таблицу уровней. Приняв за выходные функции часть тех, которые требуется синтезировать, проводят дальнейшее построение недостающей части схемы. Этот процесс заканчивается после построения схемы, реализующей все необходимые выходные функции.

Естественно, нельзя утверждать, что полученная таким путем схема будет обладать минимальной глубиной преобразования, так как каждая часть схемы будет обладать минимальной глубиной по отношению к ранее синтезированной части, а не к входным полюсам. При этом можно ожидать значительно большего пересечения элементов схемы, чем при синтезе отдельных схем для каждого выхода с последующим объединением полученных схем, как это принято в большинстве алгоритмов синтеза многовыходных схем. В ряде случаев можно получить даже уменьшение числа элементов по сравнению с результатом синтеза многовыходной схемы за один этап. Так, например, минимальное число элементов в схеме одноразрядного вычитателя равно 14, причем функция разности реализуется на пятом уровне, а функция переноса — на четвертом. При синтезе этой схемы наложением число элементов сократилось до 12, но обе функции были получены на пятом уровне. Сокращение числа элементов здесь произошло за счет снижения быстродействия схемы по выходу функции переноса.

Синтез схем с фиксированной глубиной преобразования

В ряде случаев необходимо по всем или части выходов получить заданную глубину преобразования. Такая задача разрешима лишь на том уровне полной или расширенной таблицы уровней, где присутствуют требуемые выходные функции. Для ликвидации возможной избыточности в процессе построения схемы следует исключать из таблицы уровней те функции, которые входят в образующие списки на более высоких уровнях. Следует отметить, что исключение функций из расширенной таблицы уровней может привести к неразрешимости задачи. В таких случаях не остается иного выхода, как строить избыточную схему или отказаться от требования заданной глубины преобразования.

4. Достоинства и недостатки метода

1. Метод обеспечивает синтез схем с минимальной или заданной глубиной преобразования по любому из выходов.
2. Метод допускает учет ограничений на глубину связей между элементами.
3. Многовыходные схемы синтезируются как единое целое.
4. Метод допускает синтез схем как при полностью, так и

неполностью определенных таблиц соответствия.

5. Разработанные матричные операции позволяют синтезировать схемы из конъюнктивно-дизъюнктивных элементов любой сложности. Разработка эффективных алгоритмов выполнения матричных операций для других элементов позволит расширить границы применимости метода.

6. Недостатком метода является необходимость хранить в памяти машины таблицу уровней. При этом по мере увеличения длины таблицы соответствия резко возрастает требуемый объем памяти и время синтеза. Для современных машин можно считать приемлемым $l \leq 14$.

7. Метод не позволяет синтезировать схемы с учетом ограничений на нагрузку элементов и входных полюсов.

5. Распространение метода на синтез автоматов с памятью

Состояние автомата Мура будем считать устойчивым по отношению к каждому сигналу X_i , если его действие изменяет состояние автомата не более одного раза.

Автомат логически устойчив, если он устойчив ко всем входным сигналам.

Ниже приведен пример таблицы переходов устойчивого автомата.

| | | | |
|-------|-------|-------|-------|
| | Y_1 | Y_2 | Y_3 |
| X_1 | Y_1 | Y_2 | Y_2 |
| X_2 | Y_3 | Y_3 | Y_3 |

Если входные и выходные сигналы устойчивого автомата зако-

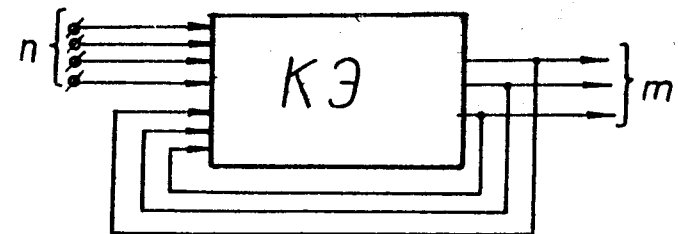


Рис. 1.

дированы соответственно кодами длины m и m , то его можно представить в виде $(m+m, m)$ -полусинхронизатора (рис.1) с m цепями обратных связей без задержек, логическую часть которого называют одноктактным или комбинационным эквивалентом [5]. Для комбинационного эквивалента по таблице переходов автомата однозначно строится таблица соответствия в принятых кодах. Такая таблица для приведенного выше автомата имеет следующий вид:

| | |
|-----------|-------|
| $X_1 Y_1$ | Y_1 |
| $X_1 Y_2$ | Y_2 |
| $X_1 Y_3$ | Y_2 |
| $X_2 Y_1$ | Y_3 |
| $X_2 Y_2$ | Y_3 |
| $X_2 Y_3$ | Y_3 |

По таблице соответствия описанном в настоящей работе методом может быть синтезирована комбинационная схема с минимальной глубиной преобразования. Соответствующий автомат также будет обладать минимальной глубиной, и под действием входных сигналов будет переходить лишь в одно из устойчивых состояний. Важно то, что полученный автомат будет формально синтезирован лишь на базе потенциальных логических элементов. В полученной схеме число обратных связей может оказаться меньше m .

Логически неустойчивые автоматы могут быть синтезированы на элементах импульсного или потенциально-импульсного типа. Для реализации неустойчивого автомата на базе потенциальных элементов его сводят к устойчивому путем тактирования. Формально тактирование автомата сводится к следующему. Если автомат неустойчив по отношению к входному сигналу X_i , то под его действием может образоваться цепочка или кольцо переходов типа $Y_k \rightarrow Y_i \rightarrow Y_j$, $Y_k \rightarrow Y_j \rightarrow Y_e \rightarrow Y_k$. Например, автомат, заданный следующей таблицей:

| | | | | | |
|-------|-------|-------|-------|-------|-------|
| | Y_1 | Y_2 | Y_3 | Y_4 | Y_5 |
| X_1 | Y_3 | Y_2 | Y_2 | Y_2 | — |
| X_2 | Y_2 | Y_4 | — | Y_1 | Y_1 |
| X_3 | Y_2 | Y_2 | — | — | — |

неустойчив по отношению к входным сигналам X_1 и X_2 . Под действием этих сигналов образуются цепочки переходов $Y_1 \rightarrow Y_3 \rightarrow Y_2$ и кольцо $Y_1 \rightarrow Y_2 \rightarrow Y_4 \rightarrow Y_1$, в которое автомат может войти и из состояния Y_5 . Пропуски в таблице переходов означают, что указанные ситуации либо невозможны, либо в этих ситуациях допустим переход в любое состояние.

Здесь неустойчивыми состояниями являются: по первой строке — Y_1 и Y_3 , а по второй — Y_1, Y_2, Y_4 . Для превращения автомата в устойчивый достаточно ликвидировать переходы указанного типа.

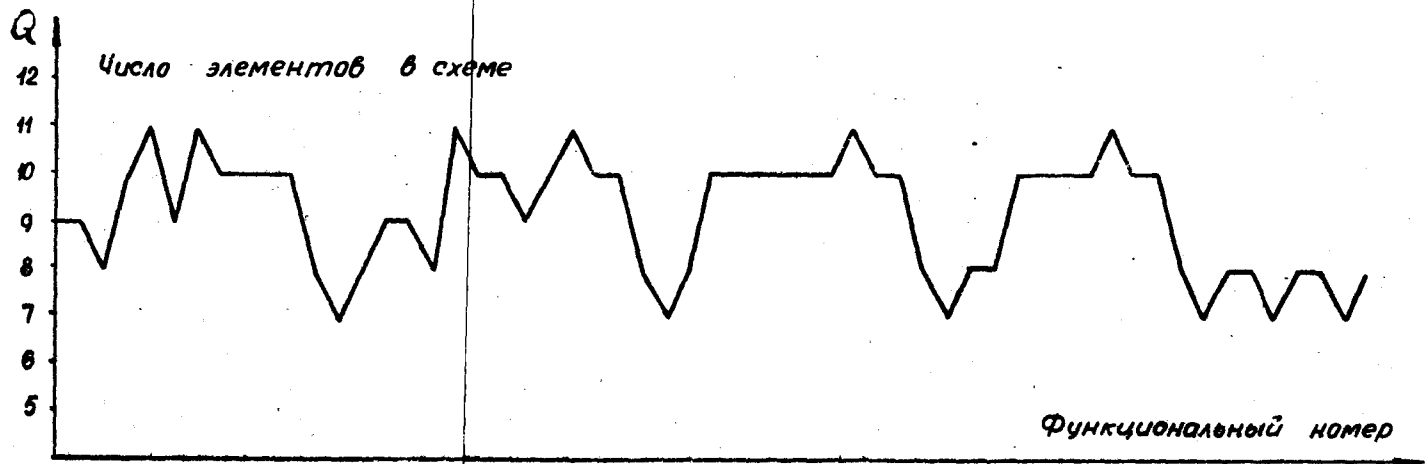
С этой целью заменим каждый из входных сигналов X_i , относительно которых автомат неустойчив, на последовательность двух сигналов X_i' и X_i'' , сопоставив им в таблице переходов по отдельной строке.

Пусть $Y_k \dots Y_e$ — все неустойчивые состояния. Дополним таблицу переходов столбцами, соответствующими состояниями $Y_k' \dots Y_e'$. После этого вновь полученную таблицу заполним следующим образом: там, где под действием сигнала X_i имел место переход в неустойчивое состояние Y_j , в строке X_i' запишем переход в состояние Y_j' . Там же, где под действием сигнала X_i имел место переход в устойчивое состояние, в строке X_i' сохраним прежний переход. В этой же строке состояния Y_j' должны быть неизменяемыми.

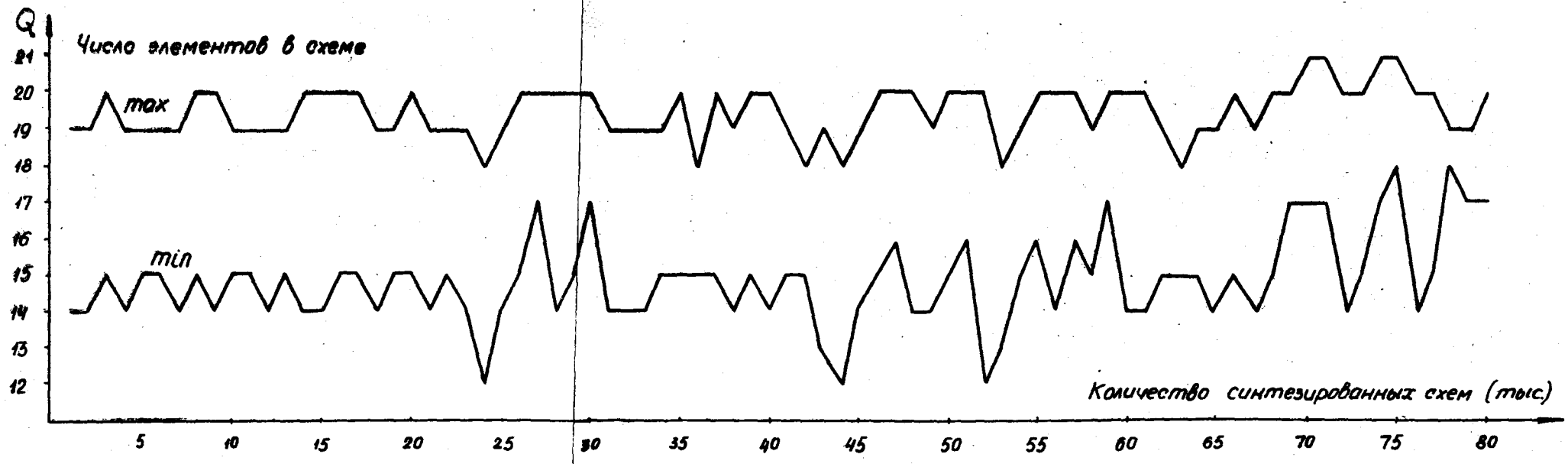
В строке X_i'' все состояния Y_j , за исключением неопределенных и тех, в которые ранее не мог переходить автомат под действием сигнала X_i , следует оставить неизменяемыми. Состояниям же Y_j' должны соответствовать переходы в Y_j . В рассматриваемом примере дополненная таблица будет иметь вид:

| | | | | | | | | | |
|---------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| | Y_1 | Y_2 | Y_3 | Y_4 | Y_5 | Y_1' | Y_2' | Y_3' | Y_4' |
| X_1' | Y_3' | Y_2 | Y_1' | Y_4 | — | Y_1' | — | Y_3' | — |
| X_1'' | Y_1 | Y_2 | Y_3 | Y_4 | — | Y_1 | — | Y_3 | — |
| X_2' | Y_2' | Y_4' | — | Y_1' | Y_1' | Y_1' | Y_2' | — | Y_4' |
| X_2'' | Y_1 | Y_2 | — | Y_4 | — | Y_1 | Y_2 | — | Y_4 |
| X_3 | Y_2 | Y_2 | — | — | — | — | — | — | — |

После кодирования полученного таким образом автомата по его таблице переходов можно составить однозначную таблицу соответствия и синтезировать соответствующую схему комбинационного эквивалента. Замена входных сигналов последовательностями X_i' , X_i'' не вызывает трудностей, если представляется возможным определение начала действия любого входного сигнала. По этому признаку легко формируется кратковременный тактирующий сигнал, используемый для образования X_i' . Тогда X_i'' может совпадать с X_i .



а)



б)

Рис. 2.

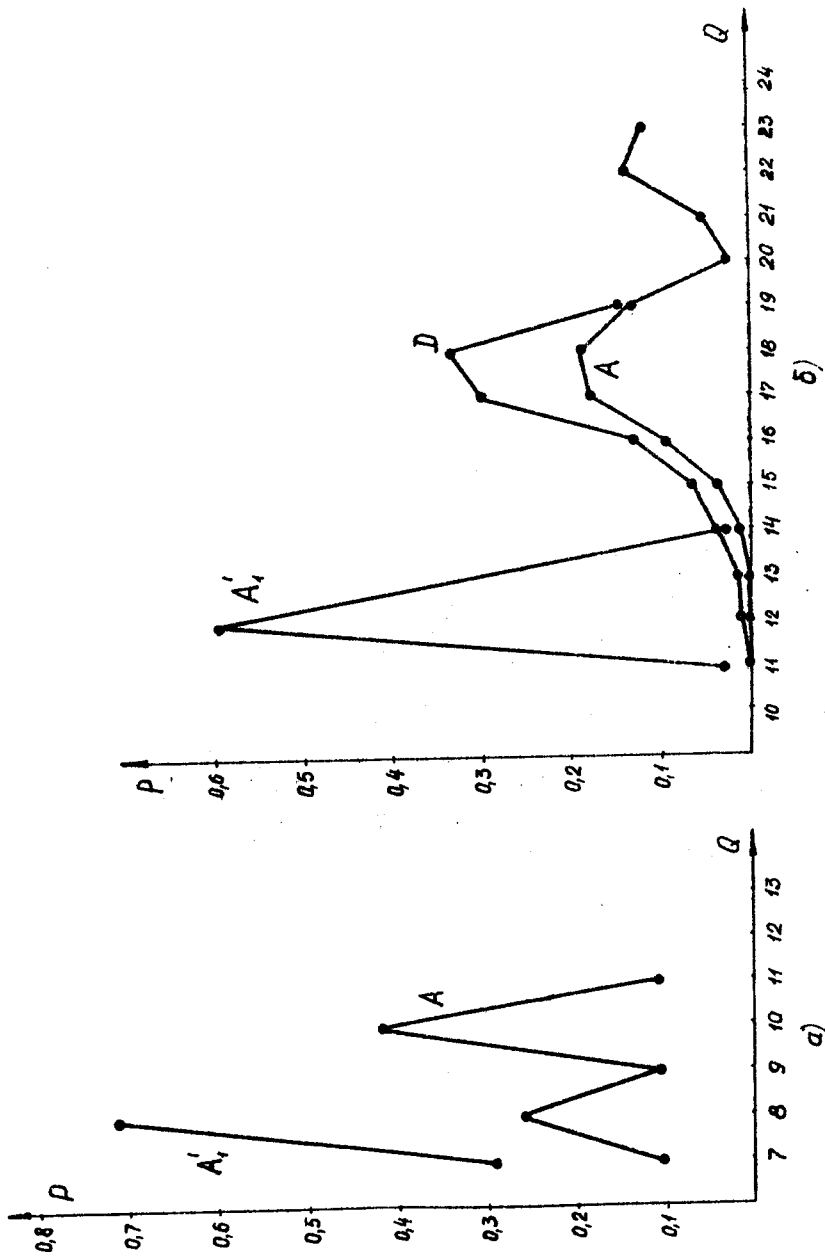


Рис. 3.

Функция выходов автомата останется неизменной, если состояниям y_j и y'_j будет соответствовать один и тот же выходной сигнал. Для этого комбинационный эквивалент автомата следует дополнить выходным комбинационным преобразователем $y_j(x_i', x_i'', y'_j)$.

5. Некоторые результаты

Функция качества. График зависимости качества (индекса) схемы от её функционального номера будем называть функцией качества.

На рис. 2 а приведена функция качества (Q - число элементов) для схемы равнозначности на три входа, синтезированной на базе двухвходовых элементов "ИЛИ-НЕ".

Во-первых, функция качества имеет характер ломаной линии со случайно разбросанными пиками вдоль всей последовательности синтезируемых схем, что резко затрудняет использование каких-либо методов спуска для поиска оптимальной схемы.

Во-вторых, схемы с глобальным минимумом числа элементов расположены на участках, где все близлежащие схемы имеют число элементов, близкое к минимальному. Последнее в какой-то мере указывает на возможную эффективность метода пробных построений.

Функция качества для более сложных схем имеет примерно такой же характер, как и в рассмотренном простом случае. На рис. 2б приведена зависимость числа элементов в схемах одноразрядного сумматора с тремя входами, синтезированных на базе двухвходовых элементов "ИЛИ-НЕ". Здесь каждая тысяча схем представлена верхней и нижней оценками.

Функции распределения. Эффективность сокращенного перебора весьма удобно оценивать по функции распределения (гистограммой), устанавливающей связь между критерием качества и относительным числом синтезируемых схем.

На рис. 3а приведен график распределения $P(q)$ для ранее упоминавшейся схемы равнозначности, полученной при выполнении всего T -перебора (кривая A). Кривая A' соответствует перебору всех схем с максимальными связями. Приведенные результаты указывают на целесообразность синтеза логических схем с максимальными связями.

Аналогичные результаты были получены и при исследовании более сложных схем. На рис. 3б приведена функция распределения для трехвходовой схемы суммирования по модулю два, синтезированной на двухвходовых элементах "ИЛИ-НЕ" по одной из ветвей дерева синтеза. Кривая A соответствует T -перебору, а кривая

A' - перебору схем с максимальными связями. Отметим, что на выполнение T - перебора было затрачено около 40 часов машинного времени (БЭСМ-2), а на перебор схем с максимальными связями лишь 3 минуты.

Как видно, T -перебор даже для простых схем требует достаточно много времени. В связи с этим была сделана попытка синтеза схем методами сокращенного перебора по алгоритмам В и С. Соответствующие функции распределения, полученные при синтезе схем одноразрядного двоичного сумматора, позволяют сделать следующие выводы:

1. Методы сокращенного перебора со случайным способом выбора образующих списков на каждом из уровней дают те же результаты, что и при выборе образующих списков упорядоченным способом.

2. Метод случайных построений (алгоритм D) приводит к той же функции распределения, что и алгоритм сокращенного перебора В и С.

3. Функция распределения, полученная при синтезе схем методом случайных построений, не совпадает с функцией распределения, соответствующей T -перебору (см. рис. 3б, кривые D и A). Это вызвано тем, что на каждом из уровней выбор образующих списков производился равновероятно. А так как каждый из списков имеет различное число продолжений, то в основном синтезировались схемы из тех ветвей дерева синтеза, которым соответствует меньшее число схем.

4. При синтезе схем с максимальными связями минимальное и максимальное число элементов в схеме мало (порядка 20%) отличается от среднего, получение которого наиболее вероятно. Таким образом, можно ожидать, что при синтезе арифметического и управляющего устройства вычислительной машины по лучшим из нескольких первых случайных построений общий расход оборудования будет отличаться от минимально возможного на величину порядка 10-20%. Этим можно пренебречь, если учесть, что в современных машинах стоимость процессора не превышает 30% от общей стоимости машины.

Л и т е р а т у р а

1. В.М. Глушков. Синтез цифровых автоматов, М., Физматгиз, 1962.
2. М.А. Гогина, Я.В. Ковалин, И.А. Кононенко, Ю.И. Кузьякин. Некоторые алгоритмы логических операций над оулевыми множествами. Настоящий сборник, стр. 72-81.
3. И.А. Кононенко, Ю.И. Кузьякин. Построение таблиц уровней логических структур в дизъюнктивно-конъюнктивном базисе. Настоящий сборник, стр. 90-101.
4. В.П. Битыцкий, Н.В. Закурдаев, Я.В. Ковалин. Синтез схем с минимальной глубиной преобразования методами сокращенного перебора. Настоящий сборник, стр. 82-89.
5. П.П. Пархоменко. Анализ релейных схем при помощи машин. Автоматика и телемеханика, 1959, XX, №4.

Поступила в редакцию
I/УИИ-1967г.