

5743

В результате работы корректора исправленная часть программы будет выглядеть так: § 5  $\Delta y \Rightarrow fF, \Delta 1 \Rightarrow 11$ .

### ЛИТЕРАТУРА

1. П. Наур. Сообщение об алгоритмическом языке АЛГОЛ-60. Журн. вычисл. матем. и матем. физики, № 2, 1961.
2. А. Д. Закревский. Описание языка ЛЯПАС. Сб.: «Логический язык для представления алгоритмов синтеза релейных устройств», М., изд. «Наука», 7—38, 1966.
3. М. Г. Карповский. Конечные автоматы с обнаружением и исправлением ошибок. Сб.: «Вычислительная техника и вопросы кибернетики», вып. 5. Изд. ЛГУ, 1968.

Статья поступила в редакцию 2 марта 1967 г.

М. Г. КАРПОВСКИЙ, М. В. ЛЕВИТ, В. И. РУЖАНСКИЙ

## НЕКОТОРЫЕ МАШИННЫЕ АЛГОРИТМЫ КОДИРОВАНИЯ И МИНИМИЗАЦИИ АВТОМАТОВ С КОРРЕКЦИЕЙ ОШИБОК

### Введение

Проблема надежности является одной из наиболее важных проблем, встающих при синтезе сложных дискретных устройств. Учет соображений надежности уже на этапах абстрактного и структурного синтеза устройства [1], как правило, позволяет понизить требуемую для достижения заданной надежности избыточность. В [2] описан метод построения дискретных устройств с коррекцией ошибок, использующий кодирование состояний устройства помехоустойчивыми кодами. При использовании этого метода учет возможности коррекции ошибок в устройстве производится на этапах абстрактного и структурного синтеза, причем способ введения избыточности определяется отображением, реализуемым устройством. Так как использование помехоустойчивых кодов для повышения надежности становится эффективным при сравнительно больших длинах кодов [3] (т. е. при достаточно большом числе состояний заданного устройства), то задача отыскания оптимального по длине используемого кода кодирования становится, как правило, весьма громоздкой.

Отыскание оптимального кодирования для автомата с 10 входами требует перебора 750 вариантов (см. ниже), а для автомата с 11 входами — уже около 12 тысяч. Ручное кодирование автомата с 10 входами и 100 состояниями занимает уже около 20 часов рабочего времени. Авторами были разработаны и экспериментально проверены машинные алгоритмы кодирования состояний дискретных устройств с коррекцией ошибок по методу, описанному в [2].

### § 1. Основные определения и алгоритм кодирования

Пусть задан автомат  $A(x)$  (под словом автомат всюду далее понимается детерминированный, инициальный автомат Мура [1]):

$$\begin{cases} a(t+1) = f(x(t+1), a(t)), \\ Y(t) = g(a(t)), \end{cases} \quad (1)$$

где  $a(t)$ ,  $x(t)$ ,  $Y(t)$  — состояние, вход и выход  $A(x)$  в момент  $t$  соответственно.

Состояние  $a_i$  будем называть  $x_i$ -достижимым в  $A(x)$ , если существует вход  $x_i$  и состояние  $a_j$  такие, что  $a_i = f(x_i, a_j)$ . Число  $x_i$ -достижимых в  $A(x)$  состояний обозначим  $n_a(x_i)$ , а число входов  $n_x$ . В [2] показано, что для исправления в автомате  $A(x)$  ошибки кратности не выше  $l$  достаточно, чтобы число  $m$  элементов памяти автомата  $A(x)$  удовлетворяло условиям:

$$\max_i n_a(x_i) \leq B(m, 2l+1), \quad (2)$$

$$\lceil \log_2 n_x \rceil + \lceil \log_2 \max_i n_a(x_i) \rceil \leq m. \quad (3)$$

Здесь и далее: символ  $\lceil N \rceil$  означает ближайшее целое число большее  $N$ ;  $B(m, 2l+1)$  — максимальное число слов  $m$ -разрядного группового кода [3] с кодовым расстоянием  $2l+1$ ; понятия ошибки в автомате и автомата с исправлением (обнаружением) ошибок используются в смысле [2]. При этом автомат с исправлением ошибок строится следующим образом.

I. По заданному автомату  $A(x)$  синтезируется некоторый эквивалентный [1] ему автомат  $A_0(x)$ . Для этого каждое состояние  $a_i$  автомата  $A(x)$  заменяется в  $A_0(x)$   $n_{x_i}(a_i)$  эквивалентными состояниями  $(n_{x_i}(a_i) — \text{число различных входов } x_i, (i = 1, 2, \dots, n_x(a_i)))$  таких, что  $a_i$  для любого  $i$   $x_i$ -достижимо в  $A(x)$ , причем для каждого состояния  $a_i$  автомата  $A_0(x)$  существует один и только один вход  $x_i$  такой, что  $a_i$   $x_i$ -достижимо в  $A_0(x)$ .

II. Для автомата  $A_0(x)$  определяется  $x_m$  такое, что  $n_a(x_m) = \max_i n_a(x_i)$  и из условий (2), (3) выбирается  $m$  и соответствующий групповой код с исправлением  $l$  ошибок.

III. Каждому входу  $x_i$  ( $i = 1, 2, \dots, n_x$ ) в соответствии с условием (3) сопоставляется смежный класс по выбранному коду так, чтобы различным входам соответствовали различные смежные классы:  $x_i$ -достижимые в  $A_0(x)$  состояния кодируются элементами соответствующих смежных классов.

Так как каждый из смежных классов является кодом с исправлением  $l$  ошибок, то при наличии ошибки в автомате истинное состояние может быть восстановлено по состоянию с ошибкой и входу (входной сигнал считается безошибочным).

Обнаружение ошибок кратности не выше  $l$  может быть осуществлено аналогичным образом, но только для кодирования состояний  $A_0(x)$  используются смежные классы по групповому коду с обнаружением  $l$  ошибок, и в достаточном условии (2)  $2l + 1$  заменяется на  $l + 1$ .

## § 2. Описание программы, реализующей алгоритм кодирования

Программа, моделирующая на ЭЦВМ рассмотренный выше алгоритм коррекции ошибок, в соответствии с пп. I, II, III, разбита на три части — п. I, п. II, п. III.

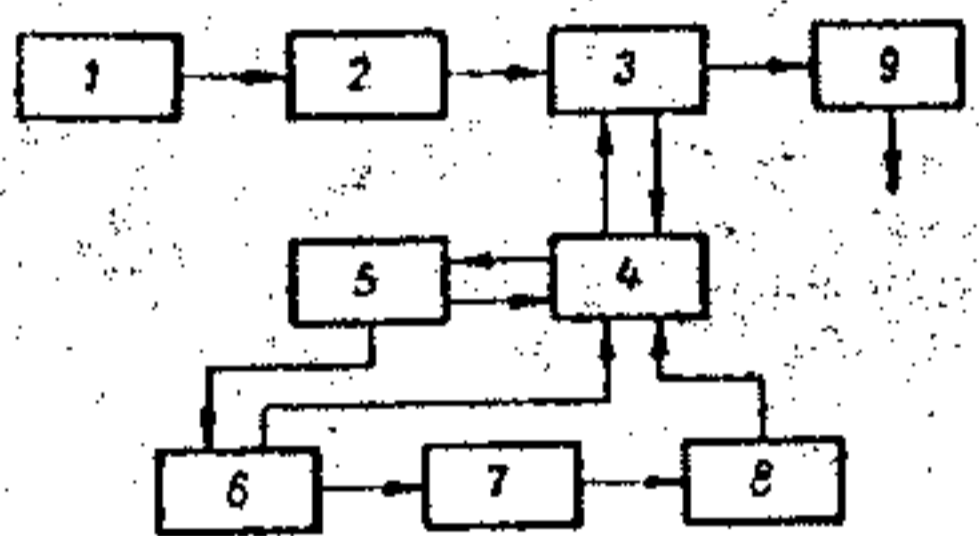


Рис. 1.

1. Блок-схема программы п. I (рис. 1) состоит из следующих блоков:

1—первая подпрограмма (подпрограмма настройки);

2—присвоение переменной  $l$  числа состояний автомата  $A_0(x)$  значения нуля;

3—фиксация входа  $x_i$  ( $i=1, 2, \dots, n_x$ );

4—последовательная фиксация  $x_i$ -достижимых состояний  $a_s$  ( $s=1, 2, \dots, n_a(x_i)$ );

5—проверка того, что  $a_s$  не встречалось ранее при выполнении алгоритма;

6—проверка того, что  $a_s$  не встречалось при зафиксированном входе  $x_i$ ;

7—увеличение переменной  $l$  числа состояний автомата  $A_0(x)$  на единицу;

8—замена состояния  $a_s$  на эквивалентное состояние  $a_{l+s}$  автомата  $A_0(x)$ ;

9—переход к программе п. II.

Программа п. I занимает в МОЗУ 178 ячеек и состоит из двух подпрограмм. Первая подпрограмма (87 ячеек, из них 53 для записи самой подпрограммы) подготавливает к работе вторую подпрограмму, изменяя адреса ее в соответствии с начальными данными (параметрами). Вторая подпрограмма (91 ячейка, из них 69 ячеек для записи самой подпрограммы) является основной и реализует п. I рассмотренного выше алгоритма. Программа п. I была проверена для  $\max n_a(x_i) < 64$ ;  $n_x < 10$ . Время

реализации на машине М-20 программы п. I для автомата с 10 входами и 100 состояниями не превышает 110 секунд (с учетом времени печати результата и входных данных).

2. Блок-схема программы п. II (рис. 2) состоит из следующих блоков: 1—первая подпрограмма (подпрограмма настройки); 2—фиксация входа  $x_i$ ; 3—вычисление  $n_a(x_i)$ —числа  $x_i$ -достижимых в  $A(x)$  состояний; 4—определение  $n_a(x_m) = \max n_a(x_i)$ ;

5—определение  $\lceil \log_2 n_x \rceil, \lceil \log_2 n_a(x_m) \rceil$ ; 6—определение порождающей матрицы [3] кода из неравенства (2); 7—проверка неравенства (5); 8—приписывание нулевых столбцов к порождающей матрице; 9—переход к п. III.

Программа п. II предназначена для выбора группового корректирующего кода. Корректирующие коды задаются своими порождающими матрицами.

Порождающие матрицы, которые могут потребоваться для нахождения кода, вводятся в память машины в виде массива начальных данных. В программе п. II из этого массива выделяется порождающая матрица, удовлетворяющая условию (2), и если при этом условии (3) не выполняется, то к найденной матрице приписываются справа нулевые столбцы так, чтобы условие (3) выполнялось (приписывание  $S$  столбцов увеличивает число смежных классов в  $2^S$  раза).

Порождающая матрица  $(m, k)$  кода имеет вид  $IP$ , где  $I$ —единичная матрица размерности  $k$ , а  $P$ —матрица размерности  $k(m-k)$ .

Таким образом, в память достаточно вводить только матрицу  $P$ , формируя порождающую матрицу в процессе работы программы. Программа п. II занимает в МОЗУ 237 ячеек и состоит из двух подпрограмм. Первая подпрограмма (92 ячейки, из них 69 для записи самой подпрограммы) подготавливает к работе вторую подпрограмму, изменяя адреса ее в соответствии с начальными данными. Вторая подпрограмма (145 ячеек, из них 101 ячейка для записи самой подпрограммы) является основной и реализует п. II рассмотренного выше алгоритма. Время реализации программы п. II для автомата с 10 входами и 100 состояниями не превышает 30 секунд.

4. Блок-схема программы п. III (рис. 3) состоит из следующих блоков: 1—первая подпрограмма (подпрограмма настройки); 2—последовательная фиксация состояний  $a_s$  ( $s=1, 2, \dots, n_a(x_m)$ ); 3—проверка того, встречалось ли ранее при кодировании  $x_m$ -достижимых состояний  $a_s$ ; 4—определение порядкового номера  $s$  состояния  $a_s$ ; 5—исхождение кодового слова, соответствующего  $a_s$ ; 6—последовательная фиксация входов

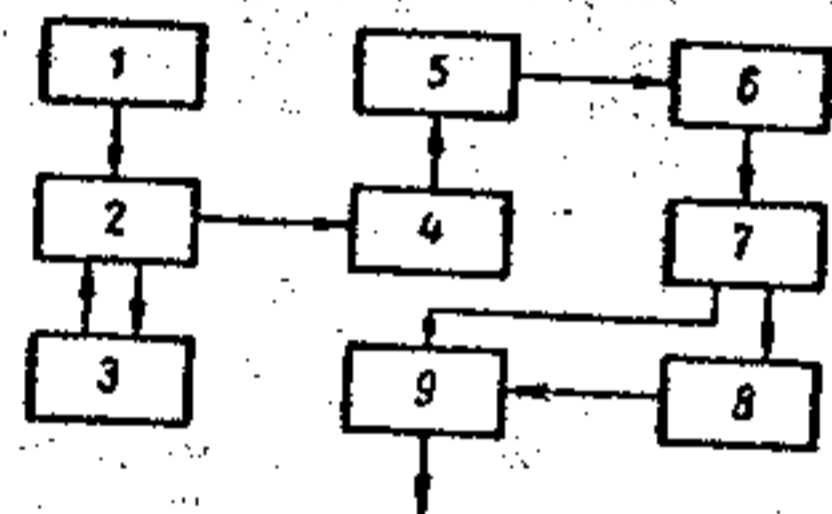


Рис. 2.



$x_i$  ( $i=1, 2, \dots, n_x$ ); 7— проверка условия  $x_i = x_m$ ; 8— определение образующей смежного класса, соответствующего  $x_i$ ; 9— последовательная фиксация состояния  $a_s$  ( $s=1, 2, \dots, n_a(x)$ ); 10— проверка того, встречалось ли ранее при кодировании  $x_i$ -достижимых состояний  $a_s$ ; 11— нахождение кодового слова, соответствующего  $a_s$ ; 12— проверка того, закодированы ли все состояния; 13— конец программы.

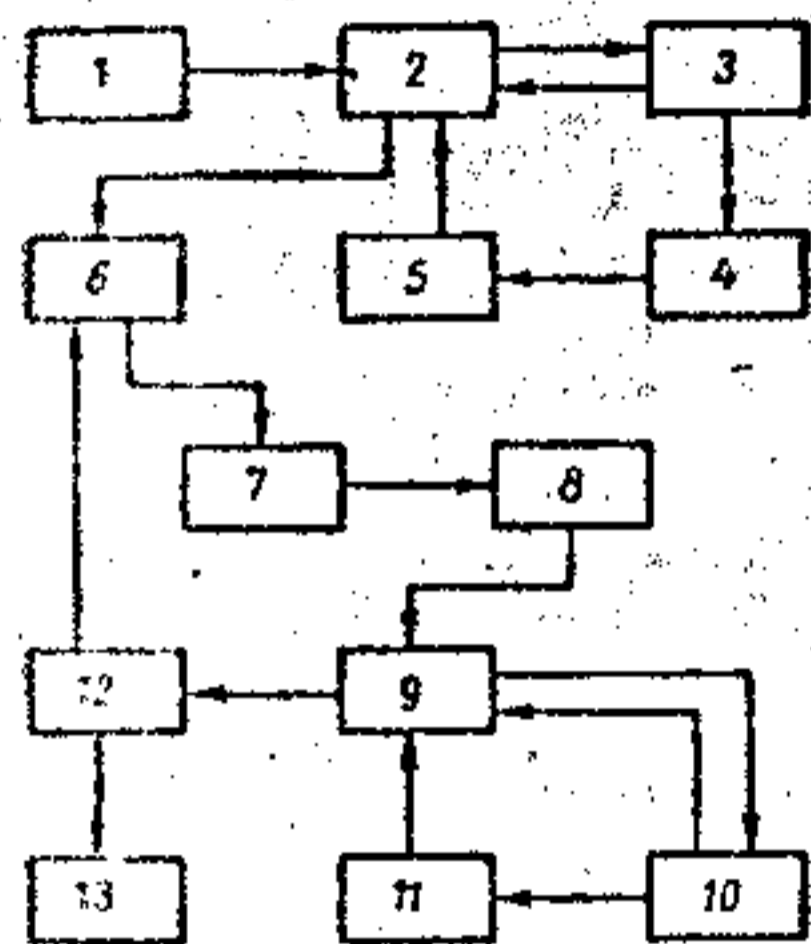


Рис. 3.

С помощью программы п. II находится вход  $x_m$  из условия  $n_a(x_m) = \max_i n_a(x_i)$ . В программе п. III  $x_m$ -достижимые состояния кодируются групповым кодом. Каждому слову этого кода можно сопоставить двоичное слово длины  $k$  следующим образом: произведение строчки (слова) длины  $k$  на порождающую матрицу равно выбранному слову группового кода. Слово длины  $k$  можно рассматривать как натуральное число, меньшее  $2^k$ , записанное в двоичной системе счисления.

Занумеруем  $x_m$ -достижимые состояния в порядке их расположения в матрице переходов исходного автомата  $A(x)$  [1], выкидывая повторяющиеся элементы, и закодируем эти состояния кодовыми словами, соответствующими их порядковым номерам.

Число смежных классов  $(m, k)$  кода равно  $2^{m-k}$ , причем за образующие смежных классов можно взять множество  $F$  слов, соответствующих натуральным числам, меньшим  $2^{m-k}$ .

Таким образом, любой смежный класс представляет собой множество слов, полученное поразрядным сложением по mod 2 слов группового кода и некоторого слова  $z$ , где  $z \in F$ . Кодирование  $x_i$ -достижимых состояний ( $i=1, 2, \dots, n_x$ ;  $x_i \neq x_m$ ) производим следующим образом: каждому входу сопоставляем некоторое слово  $z \in F$  в соответствии с порядком нумерации входов и каждому состоянию из кодируемых  $x_i$ -достижимых состояний сопоставим слово, равное поразрядной сумме по mod 2  $z$  и некоторого слова  $V$  группового кода, где  $V$  соответствует порядковому номеру кодируемого состояния.

Программа п. III занимает в МОЗУ 185 ячеек и состоит из двух подпрограмм. Первая подпрограмма (47 ячеек, из них 40 ячеек сама подпрограмма) подготавливает к работе вторую

подпрограмму. Вторая подпрограмма является основной и реализует п. III рассмотренного алгоритма. Время реализации программы п. III для автомата с 10 входами и 100 состояниями не превышает 60 секунд.

Программы п. I, п. II, п. III были проверены в режимах обнаружения и исправления одиночных и двойных ошибок.

### § 3. Обобщенный метод кодирования и минимизация автомата

Изложенный выше метод коррекции ошибок в дискретных устройствах не является в общем случае оптимальным по числу  $m$  элементов памяти. Это связано с тем, что число состояний автомата  $A_0(x)$ , кодируемых элементами смежных классов по коду с коррекцией ошибок, в ряде случаев оказывается существенно большим числа состояний исходного автомата  $A(x)$ .

Рассмотрим теперь другой, в общем случае более экономичный, метод коррекции ошибок в дискретных устройствах, который будет обобщением метода, описанного выше.

Множество  $x_i$ -достижимых в  $A(x)$  состояний обозначим  $M_i$ . Произвольное разбиение  $\lambda$  множества  $\{M_1, M_2, \dots, M_{n_x}\}$  назовем блочным покрытием для автомата  $A(x)$ . Для блочного покрытия  $\lambda$  обозначим:  $\lambda_s$  —  $s$ -ый элемент (блок) покрытия  $\lambda$ ;  $n_\lambda$  — число блоков.

Для заданного блочного покрытия  $\lambda$  состояние  $a_i$  автомата будем называть  $\lambda_s$ -достижимым в  $A(x)$ , если существует  $x_i$  такое, что  $a_i$   $x_i$ -достижимо в  $A(x)$  и  $M_i \in \lambda_s$ . Число  $\lambda_s$ -достижимых в  $A(x)$  состояний обозначим  $n_a(\lambda_s)$ . Тогда [2] для исправления в автомате  $A(x)$  ошибки кратности не выше  $l$  достаточно, чтобы  $m$  удовлетворяло условиям:

$$\max_i n_a(\lambda_s) \leq B(m, 2l+1), \quad (4)$$

$$[\log_2 n_a(\lambda_s)] + [\log_2 \max_i n_a(\lambda_s)] \leq m. \quad (5)$$

Исправление ошибок в автомате  $A(x)$  производится следующим образом.

1. Обозначим через  $n_r(a_i)$  число различных блоков  $\lambda_r$  ( $r=1, 2, \dots, n_\lambda(a_i)$ ) таких, что  $a_i$  для любого  $r$   $\lambda_r$ -достижимо в  $A(x)$ .

По заданному автомату  $A(x)$  и блочному покрытию  $\lambda$  для него строим эквивалентный автомат  $A_\lambda(x)$ , для которого каждое состояние  $a_i$  автомата  $A(x)$  заменяется в  $A_\lambda(x)$   $n_r(a_i)$ -эквивалентными состояниями, для каждого состояния  $a_i$  автомата  $A(x)$  существует один и только один блок  $\lambda_r$  покрытия  $\lambda$  такой, что  $a_i$   $\lambda_r$ -достижимо в  $A_\lambda(x)$ .

2. Для автомата  $A_\lambda(x)$  определяется  $\lambda_m$  такое, что

$n_a(0, m) = \max n_a(i_s)$  и из условий (4), (5) выбирается  $m$  и соответствующий групповой код с исправлением  $l$  ошибок.

3. Каждому блоку  $\lambda_s$  ( $s=1, 2, \dots, n_s$ ) в соответствии с условием (5) сопоставляем смежный класс по выбранному коду так, чтобы различным блокам соответствовали различные смежные классы. Кодировем  $\lambda_s$ -достижимые в  $A_s(x)$  состояния элементами соответствующих смежных классов.

При наличии ошибки в автомате истинное состояние может быть восстановлено по состоянию с ошибкой и входу  $x(t)$  следующим образом. Сначала по  $x(t)$  восстанавливается блок  $\lambda_s$  такой, что  $M_{x(t)} \in \lambda_s$ , затем по состоянию с ошибкой и блоку производится декодирование и восстанавливается истинное состояние  $a(t)$ .

Описанный метод коррекции ошибок можно рассматривать как обобщение предыдущего метода, если положить  $\lambda=0$ , где  $0$  — блочное покрытие, для которого  $0_i = M_i$  ( $i=1, 2, \dots, n_s$ ).

Число  $m$  элементов памяти автомата с коррекцией ошибок зависит от выбора блочного покрытия  $\lambda$ . Множество  $L$  всех блочных покрытий для данного автомата  $A(x)$  образует структуру [4], для которой упорядочение понимается в теоретико-множественном смысле, а  $1$  есть блочное покрытие, для которого  $n_1=1$ . Задача выбора блочного покрытия  $\lambda_{\min} \in L$ , для которого величина  $m$ , определяемая из (4), (5), достигает минимума, является, таким образом, задачей отыскания минимума числового оператора, заданного на структуре. Производящая функция для числа  $N(n_s)$  элементов структуры  $L$  совпадает с производящей функцией числа разбиений множества из  $n_s$  элементов и имеет вид

$$\exp(\exp(z) - 1) = \sum_{n_s=1}^{\infty} \frac{N(n_s)}{n_s!} z^{n_s}. \quad (*)$$

При отыскании  $\lambda_{\min}$ , как видно из (5), достаточно лишь рассмотреть кроме покрытия  $0$  покрытия, для которых  $n_s = 2^i$  ( $i=0, 1, \dots, \lfloor \log_2 n_s \rfloor$ ).

#### § 4. Некоторые оценки сложности метода минимизации

Приведем несколько соображений, которые позволяют еще более понизить число рассматриваемых покрытий. Обозначим  $m_1(i)$  и  $m_2(i)$  минимальные  $m$ , удовлетворяющие условиям (4) и (5) соответственно. Кроме того, положим  $m(i) = \max(m_1(i), m_2(i))$ . Обозначим  $k_s = \lfloor \log_2 \max n_s(i) \rfloor$ .

Так как  $m_1(i)$  и  $m_2(i) - k_s$  являются монотонно-возрастающими числовыми операторами на структуре  $L$ , то из (4), (5) следует, что если для некоторого  $i \in L$

$$m_1(i) - k_s \geq \lfloor \log_2 n_s \rfloor, \quad (6)$$

то для любого  $i'' \in L$  — такого, что  $i'' \geq i'$ , следует:  $m(i'') \geq m(i')$ , т. е. если для  $i'$  выполняется (6), то все покрытия  $i''$  такие, что  $i'' \geq i'$ , для отыскания  $\lambda_{\min}$  могут не рассматриваться.

Из условия (6) при  $i'=0$  имеем следующее необходимое условие для  $\lambda_{\min} \neq 0$  (т. е. необходимое условие целесообразности использования второго алгоритма):

$$m_1(0) - k_0 < \lfloor \log_2 n_s \rfloor. \quad (7)$$

На рис. 4 приведены зависимости от  $k_0$  минимального  $n_s$ , для которого  $\lambda_{\min} \neq 0$  для оптимальных кодов с различным кодовым расстоянием [3].

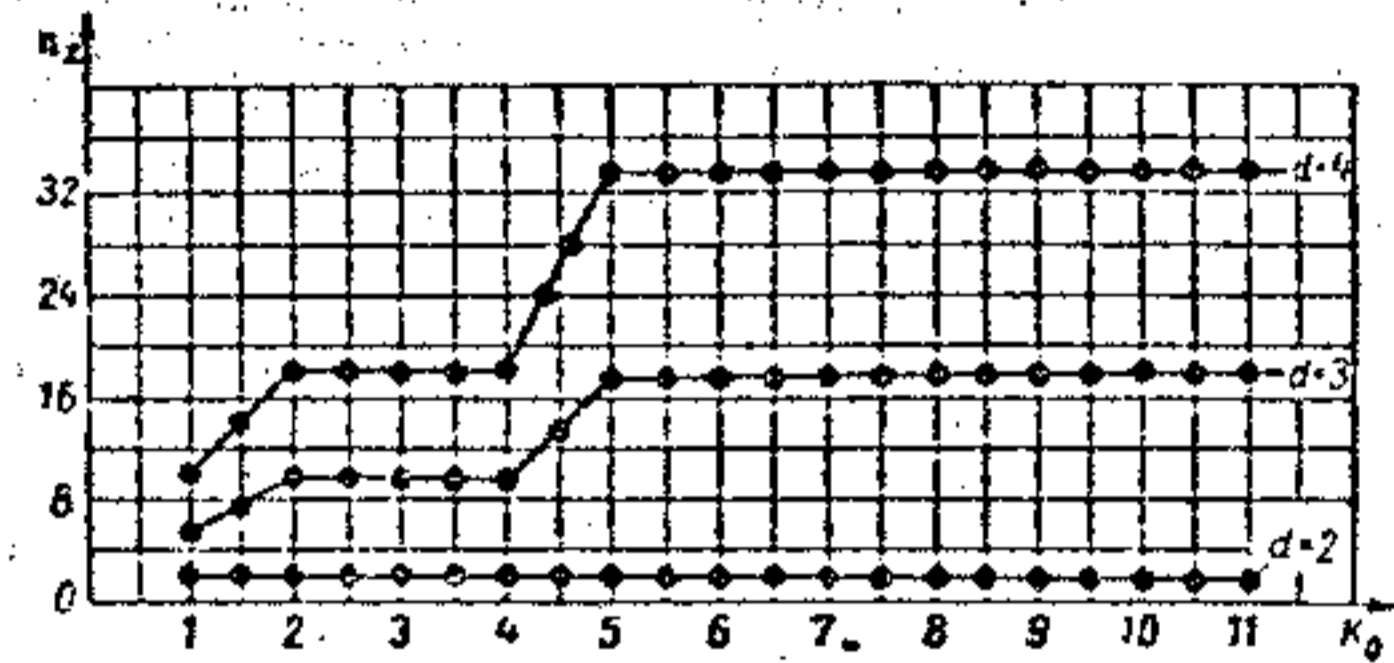


Рис. 4.

Таким образом, из (4), (5) и (6) следует, что для отыскания  $\lambda_{\min}$  достаточно рассмотреть, кроме покрытия  $0$ , лишь покрытия, для которых  $n_s = 2^{\lfloor \log_2 n_s \rfloor}, 2^{\lfloor \log_2 n_s \rfloor - 1}, \dots, 2^{m_1(0) - k_0}$ .

Верхние границы для числа  $N(n_s)$  рассматриваемых покрытий при различных  $k_0$  для исправления одиночной ошибки даны на графиках рис. 5.

В случае обнаружения одиночной ошибки  $m_1(i) - k_s = 1$  для любого  $k_s$  и, как видно из (4) и (5), для нахождения  $\lambda_{\min}$  достаточно рассмотреть покрытия  $\lambda$ , для которых  $n_s = 2$ . Число таких покрытий равно  $2^{n_s - 1} - 1$ .

#### § 5. Алгоритм минимизации

Отыскание покрытия  $\lambda_{\min}$  с учетом рассмотренных выше соображений будем осуществлять следующим образом.

Набор  $(C_1, C_2, \dots, C_{n_s})$  натуральных чисел назовем правильным, если  $C_1 = 1$  и для любого  $i$  ( $i=1, 2, \dots, n_s$ ) существует  $l < i$  такое, что  $C_l = C_i + 1$ .

Между множеством всех покрытий и множеством правильных наборов можно установить взаимнооднозначное соответствие следующим образом:  $(M, \{i_s\}) \iff (C_i = S_i)$ . При этом  $n_s = \max C_i$  покрытию  $0$  соответствует набор  $(1, 2, \dots, n_s)$ , покрытию  $1$  соответствует набор  $(1, 1, 1, \dots, 1)$ .



Отыскание правильного набора  $C_{\min}$ , соответствующего покрытию  $\lambda_{\min}$ , производится следующим образом.

1. Проверяем условие (7). Если (7) не выполнено, то  $C_{\min} = (1, 2, \dots, n_x)$ , если (7) выполнено, то переходим к п. 2.

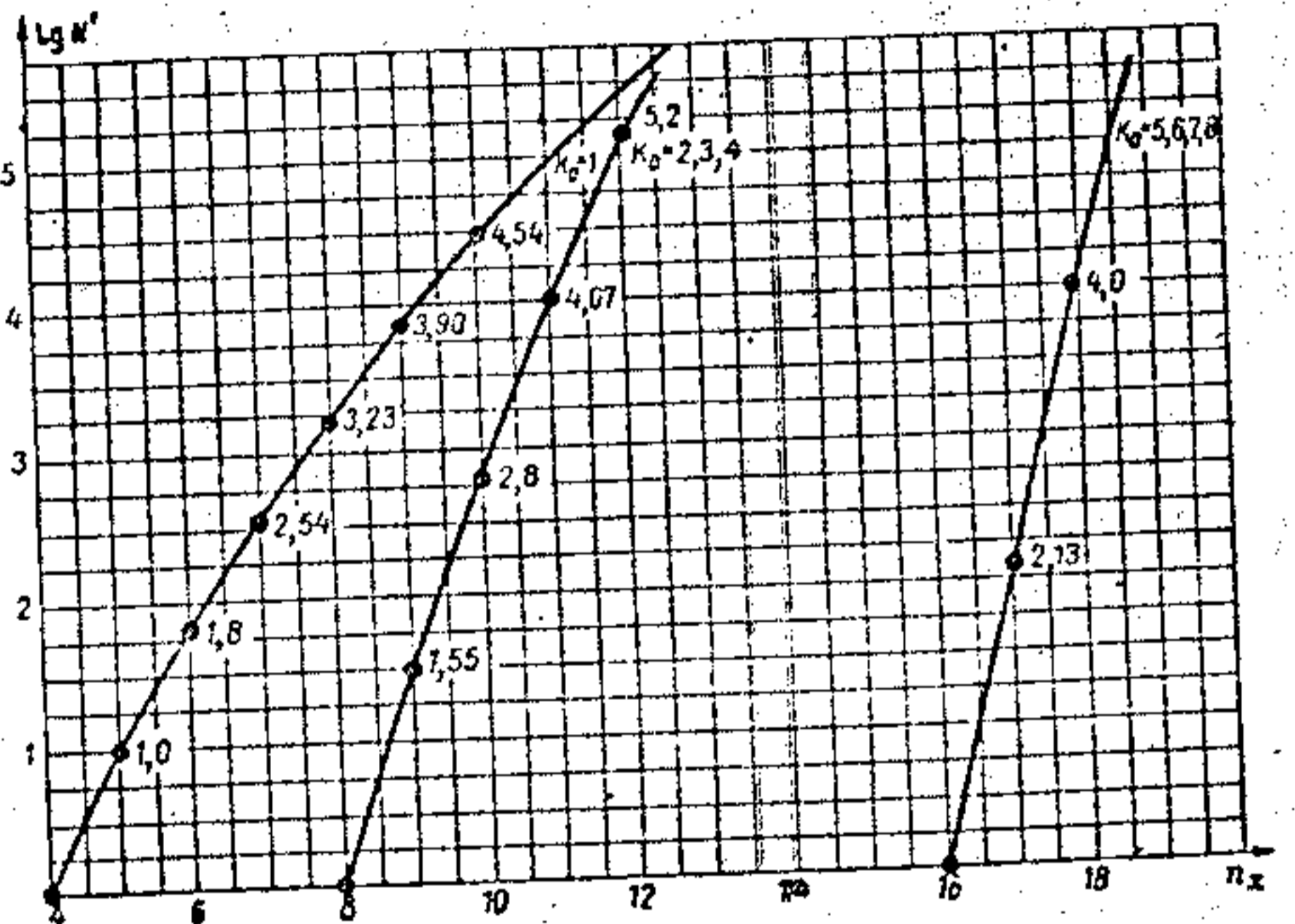


Рис. 5.

2. Рассмотрим набор  $C = (1, 2, \dots, n_1, n_2, \dots, n_x)_{n_x}$ , где  $n_i = 2^{\lfloor \log_2 n_x \rfloor}$ . Последовательным вычитанием единиц из числа  $\sum_{i=1}^{n_x} C_i (n_x + 1)^{n_x - i}$ , соответствующего этому набору в  $(n_x + 1)$ -ной системе счисления, рассмотрим все правильные наборы, для которых  $\max C_i = n_x$ . Для каждого из этих наборов найдем соответствующее  $\max n_s(\lambda)$ , затем среди них выберем  $C^{(1)}$ , которому соответствует минимальное  $\max n_s(\lambda)$ .

3. Для покрытия, соответствующего  $C^{(1)}$ , проверяем условие (6); если (6) выполнено, то переходим к п. 5, если (6) не выполнено, то запоминаем  $C^{(1)}$  и переходим к п. 4.

4. Повторяем пункты 2, 3, полагая последовательно

$$n_x = 2^{\lfloor \log_2 n_x \rfloor - 1}, 2^{\lfloor \log_2 n_x \rfloor - 2}, \dots, 2^{\lfloor \log_2 n_x \rfloor - n_0}$$

находим соответственно  $C^{(2)}, C^{(3)}, \dots, C^{(p)}$ , где

$$p = \lfloor \log_2 n_x \rfloor - m_1(0) + k_0 + 1.$$

5. Среди наборов  $C^{(1)}, C^{(2)}, \dots, C^{(p)}$  находим  $C^{(i)}$ , которому соответствует минимальное  $m$ ; при этом  $C^{(i)} = C_{\min}$ .

Программа, реализующая на АЛГО-Те алгоритм минимизации, была проверена для автомата с 10 входами и 20 состояниями.

Программа занимает в МОЗУ 1160 ячеек.

Время перебора 100 покрытий составляет 90 секунд.

### § 6. Сведение кодирования в обобщенном методе к основному алгоритму кодирования

После того как для автомата  $A(x)$  найдено оптимальное покрытие  $\lambda = \lambda_{\min}$ , кодирование состояний, необходимое для коррекции ошибок, можно осуществить следующим образом. Построили произвольный автомат  $\bar{A}(x)$ , для которого число входов равно  $n_x$  и  $\bar{M}_s = \bigcup_{x \in M_s} M_s$  (где  $M_s$  — множество  $x_s$ -достижимых

$(s=1, 2, \dots, n_x)$  состояний  $\bar{A}(x)$ ). Тогда число состояний автомата  $A_0(x)$  равно числу состояний автомата  $A_{\lambda_{\min}}(x)$ , и кодирование состояний автомата  $\bar{A}_0(x)$ , найденное с помощью применения программ п. I, п. II, п. III к автомату  $\bar{A}(x)$ , будет совпадать с кодированием состояний автомата  $A_{\lambda_{\min}}(x)$ , достаточным для коррекции ошибки.

### ПРИЛОЖЕНИЕ

В приведенных ниже таблицах дано кодирование состояний автомата как по основному алгоритму кодирования (раздел 1, табл. 1), так и по обобщенному методу (табл. 2) в режимах обнаружения и исправления одиночных ошибок.

Таблица 1а

	1	2	3
1	1	2	3
2	2	4	5
3	3	6	7
4	1	1	
5	1	1	
6	1	1	
7	1	1	

Таблица 1б

	1	2	3
1	1	2	3
2	2	4	5
3	3	6	7
4	1	1	8
5	1	1	8
6	1	1	8
7	1	1	8

Таблица 1в

	1	1	1	0
1	1	1	1	0
2	0	0	0	0
3	0	0	0	1
4	0	1	1	1
5	0	1	0	0
6	1	0	1	1
7	1	0	0	0
8	1	1	1	1

Таблица 1г

	1	1	1	0	1	1
1	1	1	1	0	1	1
2	0	0	0	0	0	0
3	0	0	0	0	0	1
4	0	1	1	1	1	0
5	0	1	1	1	1	1
6	1	0	1	0	1	1
7	1	0	1	0	0	0
8	1	1	0	1	0	0

Таблица 2а

	1	2	3	4	5	6	7	8	9
1	15	1	8	8	18	13	1	4	5
2	10	2	11	8	5	13	1	2	5
3	3	2	14	3	4	6	3	4	5
4	7	1	11	10	2	18	3	2	1
5	5	13	14	10	17	6	3	1	7
6	5	2	8	18	4	13	5	14	16
7	9	10	16	8	17	14	16	7	10
8	7	4	16	14	5	6	9	7	10
9	3	2	14	3	4	18	1	2	9
10	15	7	11	3	5	18	3	10	12
11	10	2	17	12	18	6	16	7	1
12	7	7	11	16	13	13	15	4	1
13	9	6	8	12	13	14	5	4	1
14	5	13	11	3	17	13	12	7	7
15	10	6	16	8	5	14	3	1	7
16	7	7	8	12	2	6	16	7	5
17	1	13	8	3	4	18	15	7	10
18	1	6	8	12	17	6	1	7	16

Таблица 2б

	1	2	3	4	5	6	7	8	9
1	15	19	8	8	22	13	1	4	5
2	10	2	11	8	23	13	1	2	5
3	3	2	14	3	4	6	3	4	5
4	7	19	11	10	2	22	3	2	1
5	5	13	14	10	21	6	3	19	7
6	5	2	8	18	4	13	5	25	16
7	9	20	16	8	21	25	16	21	10
8	7	4	16	14	23	6	9	21	10
9	3	2	14	3	4	22	1	2	9
10	15	21	11	3	23	22	3	20	12
11	10	2	17	12	22	6	16	21	1
12	7	21	11	10	13	13	15	4	1
13	9	6	8	12	13	25	5	4	1
14	5	13	11	3	24	13	12	21	7
15	10	6	16	8	23	25	3	19	7
16	7	21	8	12	2	6	16	21	5
17	1	13	8	3	4	22	15	21	10
18	1	6	8	12	24	6	1	21	16

Таблица 2в

	1	2	3	4	5	6	7	8	9
1	15	19	8	23	28	34	38	4	45
2	10	2	11	23	29	34	38	2	45
3	3	2	22	24	30	35	39	4	45
4	7	19	11	25	31	36	39	2	46
5	5	13	22	25	32	35	39	19	47
6	5	2	8	18	30	34	40	14	48
7	9	20	16	23	32	37	41	21	49
8	7	4	16	26	29	35	42	21	49
9	3	2	22	24	30	36	38	2	50
10	15	21	11	24	29	36	39	20	51
11	10	2	17	12	28	35	41	21	46
12	7	21	11	27	33	34	43	4	46
13	9	6	8	12	33	37	40	4	46
14	5	13	11	24	32	34	44	21	47
15	10	6	16	23	29	37	39	19	47
16	7	21	8	12	31	35	41	21	45
17	1	13	8	24	30	36	43	21	49
18	1	6	8	12	32	35	38	21	46

Таблица 2г

	1	0	0	0	0	0
1	0	0	0	0	0	1
2	0	0	0	0	0	1
3	0	0	0	0	1	1
4	0	0	0	0	1	0
5	0	0	1	0	0	1
6	0	0	1	0	0	0
7	0	0	1	1	0	0
8	0	1	0	0	0	1
9	0	1	0	0	1	0
10	0	1	1	0	0	0
11	0	1	1	1	1	1
12	1	0	0	0	0	1
13	0	0	1	1	1	1
14	1	0	0	1	0	0
15	1	0	1	0	0	0
16	1	0	1	1	1	1
17	1	1	0	0	0	0
18	1	1	0	1	1	1
19	0	1	0	0	0	0
20	0	1	0	1	1	1
21	0	1	1	0	0	1
22	0	1	1	1	0	0
23	1	0	0	0	0	0
24	1	0	0	1	1	1
25	1	0	1	0	0	1

Табл. 1а и 2а представляют собой таблицы переходов [1] синтезируемых автоматов, причем столбцы их обозначаются входными сигналами соответствующего автомата, а строки — его состояниями.

Табл. 1б, 2б и 2в представляют собой результат применения п. 1 основного алгоритма и являются, таким образом, записанными в компактном виде таблицами переходов соответствующих А-автоматов. (Заметим, что любое состояние автомата А эквивалентно некоторому состоянию автомата А, поэтому строки, обозначенные этими состояниями, совпадают, и данные таблицы полностью определяют алгоритмы автоматов.)

В результате применения основного алгоритма в режимах обнаружения и исправления одиночных ошибок к автомату с таблицей переходов — табл. 1а получаем автомат с обнаружением одиночных ошибок (таблица переходов — табл. 1б, кодированные состояния — табл. 1в) и автомат с исправлением оди-



Таблица 2а

1	1	1	0	0	0	0
2	0	0	1	1	0	1
3	0	1	0	1	1	0
4	1	0	0	1	1	0
5	1	0	0	1	1	1
6	1	1	0	0	0	1
7	0	1	1	0	1	1
8	0	0	0	0	1	0
9	1	0	1	0	1	0
10	0	0	1	1	0	0
11	0	0	1	1	1	1
12	1	0	1	0	0	0
13	0	1	0	1	1	1
14	1	1	1	1	0	0
15	0	0	0	0	0	1
16	0	1	1	0	0	0
17	1	0	0	1	0	0
18	0	1	1	0	0	1
19	0	0	0	0	0	0
20	0	1	1	0	1	0
21	1	0	1	0	1	1
22	0	1	0	1	0	1
23	0	0	0	0	1	1
24	0	0	1	1	1	0
25	0	1	0	1	0	0
26	1	0	0	1	0	1

27	1	1	0	0	1	0
28	0	0	0	1	0	0
29	0	0	1	0	0	1
30	0	1	0	0	1	1
31	0	1	1	1	1	0
32	1	0	0	0	1	0
33	1	0	1	1	1	1
34	0	0	0	1	0	1
35	0	0	1	0	0	0
36	0	1	0	0	1	0
37	0	1	1	1	1	1
38	0	0	0	1	1	0
39	0	0	1	0	1	1
40	0	1	0	0	0	1
41	0	1	1	1	0	0
42	1	0	0	0	0	0
43	1	0	1	1	0	1
44	1	1	0	1	1	1
45	0	0	0	1	1	1
46	0	0	1	0	1	0
47	0	1	0	0	0	0
48	0	1	1	1	0	1
49	1	0	0	0	0	1
50	1	0	1	1	0	0
51	1	1	0	1	1	0

и других ошибок (таблица переходов — табл. 1б, кодированные состояния — табл. 1г).

В результате применения алгоритма минимизации к автомату с таблицей переходов — табл. 2а в режиме обнаружения одиночных ошибок получаем оптимальное покрытие, соответствующее правильному набору (1, 2, 1, 1, 2, 2, 1, 2, 1). Используя теперь основной алгоритм (как указано выше), получаем автомат с обнаружением одиночных ошибок (таблица переходов — табл. 1б, кодированные состояния — табл. 2г). В результате применения обобщенного метода в режиме исправления одиночных ошибок получаем оптимальное покрытие, соответствующее правильному набору (1, 2, 3, 4, 5, 6, 7, 2, 8), и автомат с исправлением одиночных ошибок (таблица переходов — табл. 2а, кодированные состояния — табл. 2д).

Отметим, что для рассмотренных примеров обнаружение одиночных ошибок методом, описанным в [5], потребовало бы в обоих случаях одного дополнительного элемента памяти, в то время как обнаружение ошибок описанными выше методами не требует избыточных элементов памяти. Исправление одиночных ошибок методом, описанным в [5], потребовало бы в первом примере на один элемент памяти больше, а во втором — на три элемента больше, чем исправление ошибок методами, описанными выше.

#### ЛИТЕРАТУРА

1. В. М. Глушков. Синтез цифровых автоматов. М., Физматгиз, 1962.
2. М. Г. Карповский. Конечные автоматы с обнаружением и исправлением ошибок. Сбл. «Вычислительная техника и вопросы кибернетики», вып. 5. Изд. ЛГУ, 1968.
3. У. Питерсон. Коды, исправляющие ошибки. М., изд. «Мир», 1964.
4. Г. Биркгоф. Теория структур. М., ИЛ, 1952.
5. М. А. Гаврилов. Структурная избыточность и надежность релейных устройств. Труды I международного конгресса ИФАК, т. 3. М.—Л., Изд. АН СССР, 1964.

Статья поступила в редакцию 27 февраля 1967 г.

М. К. ЧИРКОВ

### ВЫЧИСЛЕНИЕ ВЕРОЯТНОСТНЫХ ХАРАКТЕРИСТИК КОМБИНАЦИОННЫХ СХЕМ

Задача вероятностного анализа комбинационной схемы состоит в определении ее вероятностных характеристик, если заданы вероятностные свойства образующих ее комбинационных элементов и структура схемы. Наиболее общим методом решения подобной задачи является матричный метод [1, 2], позволяющий по заданным для элементов матрицам вероятностей входов и структуре схемы найти ее матрицу вероятностей выходов. Однако этот метод требует знания матриц вероятностей входов элементов и более удобен для анализа схем из многоразрядных (имеющих более одного элементарного входного канала) элементов. В приложениях же часто встречаются комбинационные схемы, построенные из одноканальных (с одним элементарным входным ка-