

УДК 681.142.353

МЕТОД ОПТИМИЗАЦИИ СИНТЕЗА КОНЕЧНЫХ АВТОМАТОВ
В ВЫЧИСЛИТЕЛЬНОЙ СРЕДЕ

Н.Г.Топольский, А.Я.Дровяников

1. В в е д е н и е

При реализации конечных автоматов в однородной универсальной вычислительной среде (ВС) [1] весьма важной является проблема уменьшения избыточности синтезированных в ВС конечных автоматов.

В настоящее время известны некоторые методы оптимизации синтеза автоматов в ВС. Например, в работе [2] рассматривается оптимизация синтеза логических сетей в ВС, основанная на использовании определенных преобразований графа логической сети.

В данной работе рассматривается возможный подход к решению этой задачи для одного из методов синтеза автоматов в ВС, формулируются условия, необходимые для осуществления такой оптимизации, и приводятся алгоритмы оптимизации, которые затем иллюстрируются на примерах. Методы могут оказаться полезными при синтезе автоматов в широком классе ВС [2,3,4,5], построенных на основе различных функционально полных логических базисов элементов ВС.

Пусть задана двумерная плоская ВС с индивидуальной наст -

ройкой элементов, обладающая функциональной и соединительной полнотой [1]. Автомат задан множеством входных $M = \{m_1, m_2, \dots, m_k\}$ внутренних $U = \{u_1, u_2, \dots, u_s\}$ и $L = \{l_1, l_2, \dots, l_p\}$ выходных двоичных переменных, а также дизъюнктивными нормальными формами (д.н.ф.) минимизированных функций возбуждения элементарных автоматов

$$u_{ij}(t) = f_{ij}(\bar{m}_1, \dots, \bar{m}_k, \bar{u}_1, \dots, \bar{u}_s), \quad (1)$$

которые подаются на j -й вход i -го элементарного автомата, и минимизированных функций выходов

$$l_n(t) = f_n(\bar{m}_1, \dots, \bar{m}_k, \bar{u}_1, \dots, \bar{u}_s), \quad (2)$$

$$n \in N = \{1, 2, \dots, p\}.$$

Будем реализовывать автомат в ВС по следующей блок-схеме [3]. Весь массив элементов ВС состоит из нескольких участков программы. Программа Π_1 служит для получения элементарных конъюнкций формул (1) и (2), программы Π_2 - для сборки элементарных конъюнкций, а программы Π_3 - для реализации определенного типа элементарных автоматов (э.а.). Для каждой входной и внутренней переменных в программе Π_1 отводится c столбцов элементов ВС ($c = 1, 2, \dots$) в зависимости от типа и особенностей функционального элемента ВС. Таким образом, число столбцов в программе Π_1 равно $c(k+s)$, а число строк в ней в общем случае равно $S + \omega$, где ω - число членов д.н.ф. в формулах (1) и (2). В такой программе в каждой строке из элементов ВС реализуется один член д.н.ф. Нетрудно показать, что по такой блок-схеме можно реализовать автоматы в целом классе ВС, в котором функциональный элемент может быть построен на основе большинства функционально полных логических базисов.

Если функции (1) и (2) получить с использованием декомпозиционного метода синтеза конечных автоматов [6, 7] или канонического метода синтеза с минимизацией функций u_i , то в правой части может отсутствовать некоторая группа входных и внутренних переменных. В этом случае в программе Π_1 u_i -го элементарного автомата соответствующие столбцы будут состоять только из соединительных элементов. То же самое относится к соответствующим участкам программы Π_1 функций выходов (2).

Будем искать такую реализацию автомата в ВС, при которой можно было бы сократить число соединительных участков столбцов и тем самым уменьшить избыточность реализации автомата в среде.

По функциям (1) и (2) построим ориентированный граф G следующим образом. Каждой переменной (входной, внутренней и выходной) поставим в соответствие вершину графа. Таким образом, в графе G будет три типа вершин: $M = \{m_1, m_2, \dots, m_k\}$, $U = \{u_1, u_2, \dots, u_s\}$ и $L = \{l_1, l_2, \dots, l_p\}$. Назовем их входными, внутренними и выходными вершинами соответственно. В случае, если не будет необходимости уточнять наименование вершин графа G , будем обозначать ее символом $q_i \in Q$, $Q = MUUL$.

Если в функцию возбуждения u_i входит прямое или инверсное значение переменной u_j (либо m_j), то вершину u_i соединим с вершиной u_j (или m_j) дугой, выходящей из вершины u_i и входящей в u_j . Аналогично соединимся дугами выходные вершины l_j с вершинами u_i и m_j . Если каждой вершине графа G , за исключением входных вершин, поставить в соответствие число $\gamma(q_i)$ членов д.н.ф. в соответствующей функции из (1) или (2), то получим граф со взвешенными вершинами. Назовем построенный таким образом граф G графом зависимостей функций возбуждения элементарных автоматов и функций выходов, или просто графом зависимостей G .

Обозначим полу степень захода (число заходящих дуг) вершины q_i через $\rho(q_i)$, а полу степень исхода вершины q_i - $\varphi(q_i)$. Тогда для любой вершины $m_i \in M$, $u_j \in U$ и $l_n \in L$ графа G произвольного автомата A верны соотношения:

$$\rho(m_i) = 0, \quad \varphi(m_i) > 0; \quad (3)$$

$$\rho(u_j) > 0, \quad \varphi(u_j) \geq 0; \quad (4)$$

$$\rho(l_n) > 0, \quad \varphi(l_n) = 0. \quad (5)$$

Построение графа G рассмотрим на примере автомата B , функции возбуждения и функции выходов которого имеют следующий вид:

$$u_{11}(t) = \bar{m}_1 m_6 \bar{m}_8 \vee m_4 \bar{m}_6 m_8, \quad u_{10}(t) = m_1 \bar{m}_6 \bar{m}_8;$$

$$u_{21}(t) = \bar{m}_4 u_2 \vee m_4 m_7 u_5, \quad u_{20}(t) = m_4 \bar{m}_7 u_2;$$

$$\begin{aligned} u_{31}(t) &= \bar{m}_2 m_7 \vee m_2 \bar{m}_3 \bar{u}_2, & u_{30}(t) &= m_2 u_2 \vee m_3 \bar{m}_7 \bar{u}_2; \\ u_{41}(t) &= \bar{m}_1 m_6 \bar{u}_4, & u_{40}(t) &= m_1 \bar{m}_5 \vee m_5 \bar{m}_6 u_4; \\ u_{51}(t) &= m_2 \bar{m}_7 \bar{u}_5 \vee m_4 m_7, & u_{50}(t) &= m_2 \bar{m}_4 u_5; \\ u_{61}(t) &= m_1 \bar{m}_8 \vee m_8 \bar{u}_1, & u_{60}(t) &= \bar{m}_1 \bar{m}_8 u_1; \end{aligned}$$

$$e_1(t) = m_5 \bar{m}_7 u_2 \bar{u}_3;$$

$$e_2(t) = \bar{m}_3 m_4 \bar{u}_5 \vee m_2 \bar{m}_4 u_2;$$

$$e_3(t) = m_1 u_6 \vee m_1 \bar{m}_5 \bar{u}_1 \vee m_1 \bar{m}_4 \bar{u}_1 u_4.$$

Граф зависимостей G_B автомата B показан на рис. I.

$q_1 - m_1 (\gamma_1 = 0); q_4 - m_4 (0); q_7 - m_7 (0); q_{10} - u_2 (3); q_{13} - u_5 (3);$
 $q_2 - m_2 (\gamma_2 = 0); q_5 - m_5 (0); q_8 - m_8 (0); q_{11} - u_3 (4); q_{14} - u_6 (3);$
 $q_3 - m_3 (\gamma_3 = 0); q_6 - m_6 (0); q_9 - u_1 (3); q_{12} - u_4 (3); q_{15} - e_1 (1);$
 $q_{16} - e_2 (2); q_{17} - e_3 (3).$

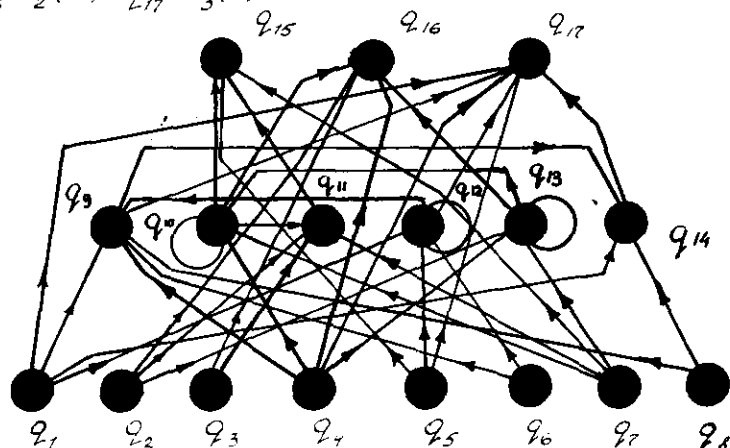


Рис. I.

Отметим, что при $U = \emptyset$ граф G будет соответствовать автомату без памяти (комбинационной схеме), при $L = \emptyset$ - автомату без выходов и т.д.

По графу зависимостей нетрудно оценить сложность реализации автомата $R(A)$ в ВС

$$R(A) = \left(\sum_{i=1}^{\beta} \gamma(q_i) + C_1 \right) (C(S+K) + C_2), \quad (6)$$

где $\beta = S + K + P$;

C - число столбцов элементов ВС программы Π_g , приходящихся на одну переменную m_i или u_j ;

C_1 - величина, равная числу строк элементов ВС, необходимых для подачи внутренних переменных на программу Π_g (подразумевается, что $C_1 \ll \sum_{i=1}^{\beta} \gamma(q_i)$);

C_2 - величина, равная числу столбцов в программах Π_v и Π_u ;

C, C_1 и C_2 зависят от типа элементов ВС.

2. Метод упорядочивания множества вершин графа G

Первый метод оптимизации программы автомата в ВС заключается в нахождении такого кортежа вершин графа G , при котором в программе Π_g содержится минимальное или близкое к нему число столбцов при неизменном числе строк. Сокращение числа столбцов в программе Π_g получим за счет использования пустующих участков столбцов для подачи других входных или внутренних переменных. Обозначим минимально и максимально возможное число столбцов в программе Π_g - r_{min} и r_{max} соответственно. Вершину, имеющую максимальную подустепень захода, обозначим q_m . Тогда из метода реализации конъюнктивных членов вытекает, что справедливы следующие соотношения;

$$r_{min} \geq C \cdot P(q_m), \quad (7)$$

$$r_{max} = C(K+S). \quad (8)$$

Нетрудно убедиться, что если для какой-либо вершины q_i графа оказывается верным соотношение

$$\rho(q_i) = s + k, \quad (9)$$

то число столбцов в программе Π_k уменьшить невозможно.

Таким образом, необходимым условием возможности оптимизации синтеза автомата A в ВС методом упорядочивания множества вершин графа зависимостей G является

$$\rho(q_m) < s + k \quad (10)$$

Условие (10) позволяет, не прибегая к алгоритму оптимизации, заранее определить целесообразность его применения к графу G . Однако выполнение (10) не является достаточным для осуществления оптимизации этим методом. Убедиться в этом можно лишь после выполнения алгоритма оптимизации.

Сформулируем алгоритм, конечной целью которого является нахождение такого кортежа вершин UUL графа G , который позволил бы получить минимальное или близкое к нему число столбцов в программе Π_k за счет подачи на один столбец нескольких переменных одновременно.

1°. Проверяем условие (9) для вершины q_m графа G . Если это условие выполняется — переход к 9°, в противном случае — к 2°.

2°. Для каждой вершины $q_j \in (UUL)$ графа G формируем два множества: множество A_j всех вершин $q_i \in Q$, из которых выходят дуги в q_j , и множество вершин $C_j = Q \setminus (A_j \cup B_j \cup \{q_j\})$, где B_j — множество таких вершин $q_i \in Q$, в которые заходят дуги из вершины q_j . Все вершины графа G считаем неотмеченными. Переход к 3°.

3°. Из множества UUL неотмеченных вершин выбираем вершину q_i с $\rho(q_i)_{\min}$ и отмечаем ее индексом $k - 1$, где k — номер цикла алгоритма. Переход к 4°.

4°. Каждой неотмеченной вершине q_i множества UUL ставим в соответствие число $\delta_i^k = \delta_i^{k-1} + \delta_i^{k'}$, где $\delta_i^0 = 0$, $\delta_i^{k'}$ — число множеств C_j и A_j , содержащих q_i . Переход к 5°.

5°. Из множества неотмеченных вершин выбираем вершину q_i с δ_i^k и отмечаем её индексом k . Если $|UUL| - k > 1$ — переход к 6°, если $|UUL| - k = 1$ — переход к 7°.

6°. Если множество $\bigcup_{i \in I} (A_i \cup B_i)$, $I = \{1, 2, \dots, k\}$, содержит неотмеченные вершины, — переход к 4°, в противном случае среди неотмеченных вершин выбираем вершину q_n с $\rho(q_n)_{\min}$,

присваиваем ей индекс $k + 1$ и переходим к 4°.

7°. Неотмеченную вершину отмечаем индексом $k + 1$. Переход к 8°.

8°. Конец работы алгоритма.

9°. Алгоритм не применим к данному графу.

Расположив вершины $q_j \in UUL$ графа G по возрастанию их отметок k , получим искомый кортеж

$$\langle q_{j_1}^0, q_{j_2}^1, \dots, q_{j_\sigma}^{\sigma-1} \rangle,$$

где $\sigma = |UUL|$.

Действие алгоритма покажем на примере графа зависимостей G_3 . Выполняя для множества вершин UUL этого графа все пункты алгоритма, получим кортеж

$$\langle u_6^0, u_3^1, u_4^2, u_1^3, u_5^4, u_2^5, u_7^6, u_8^7 \rangle.$$

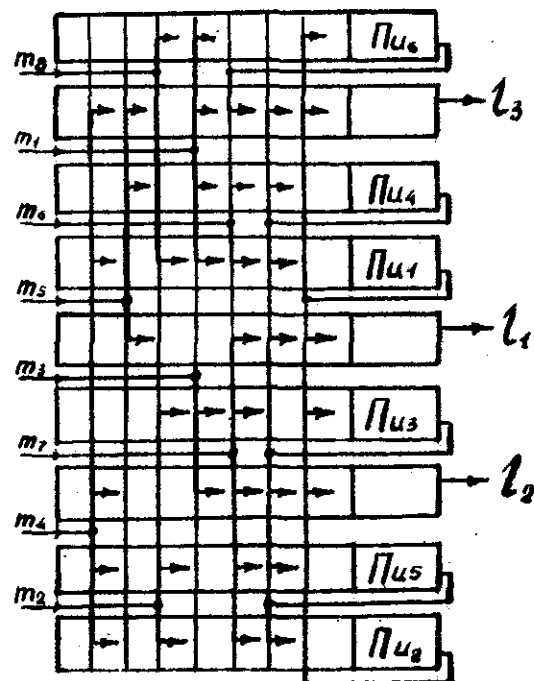


Рис. 2.

На рис.2 показана схема совмещения подачи переменных в столбцах программы Π_g . Вместо 14 столбцов в ней будет использоваться 7 столбцов без нарушения функционирования автомата.

Из примера видно, что быстрое действие автомата при такой оптимизации (если она возможна) всегда будет увеличиваться, так как каждая элементарная конъюнкция будет реализовываться в более короткой строке.

3. Метод разбиения графа G на подграфы

Второй метод оптимизации состоит в разбиении графа зависимостей G на подграфы $G_1, G_2, \dots, G_v$ так, чтобы они имели минимальное число связей между собой и для каждого подграфа G_i выполнялось условие

$$\sum_{j=1}^{|J_i|} \gamma(q_j) = t_i, \quad q_j \in G_i, \quad (II)$$

где J_i — множество индексов вершин подграфа G_i .

Очевидно, что в общем случае в точности выполнить условие (II) невозможно, поэтому его можно заменить менее сильным условием

$$t_i - \xi \leq \sum_{j=1}^{|J_i|} \gamma(q_j) \leq t_i, \quad (I2)$$

где величины t_i и $\xi = 1, 2, \dots$ заданы заранее.

При этом методе оптимизации вся программа автомата A в ВС состоит из нескольких блоков, минимально связанных между собой. Высота i -го блока не превышает величины t'_i , причем $t'_i > t_i$, так как некоторое число горизонтальных строк каждого блока ВС используется для подачи переменных на столбце программы Π_g . В частном случае может быть $t'_1 = t'_2 = \dots = t'_v = t'$, т.е. все блоки ВС могут иметь одинаковую высоту t' . Условимся для простоты в дальнейшем рассматривать именно этот случай.

Для каждого подграфа G_i сделаем следующие обозначения:

$\lambda_1(G_i)$ — число внутренней зависимости подграфа G_i ,

равное сумме чисел таких вершин $q_j \in U$ подграфа G_i , у которых $\varphi(q_j) \neq 0$;

$\lambda_2(G_i)$ — число внешней зависимости подграфа G_i , равное числу таких вершин $q_j \in U$ подграфов $G_1, \dots, G_{i-1}, G_{i+1}, \dots, G_v$, из которых выходят дуги, заходящие в подграф G_i ;

$\lambda(G_i) = \lambda_1(G_i) + \lambda_2(G_i)$ — число общей зависимости подграфа G_i .

С учетом затраты строк ВС для подачи переменных на программы Π_{g_i} , условие (I2) изменится и будет выглядеть следующим образом:

$$t' - \xi \leq \lambda(G_i) + \sum_{j=1}^{|J_i|} \gamma(q_j) \leq t', \quad (I3)$$

Нетрудно убедиться в справедливости следующего предложения.

Для того, чтобы суммарное число элементов в программах Π_{g_i} всех блоков было меньше, чем в исходной неоптимизированной программе Π_g , в графе G необходимо существование хотя бы одного подграфа G_i , удовлетворяющего (I3) и неравенству

$$\lambda(G_i) < s + \kappa. \quad (I4)$$

Действительно, если нет подграфа, удовлетворяющего одновременно условиям (I3) и (I4), то для любого подграфа, в котором

$$t' - \xi \leq \sum_{j=1}^{|J_i|} \gamma(q_j) \leq t',$$

можно записать $\lambda(G_i) \geq s + \kappa$. Это значит, что в программе Π_{g_i} всех блоков, соответствующих подграфам G_i , необходимо иметь не менее $(s + \kappa)$ столбцов, т.е. нельзя сократить число столбцов ни в одном блоке. Поскольку общее число строк в блоках ВС больше или равно числу строк в неоптимизированной программе Π_g , то сокращение числа элементов в программах Π_{g_i} по сравнению с неоптимизированной программой Π_g получить невозможно.

Однако условия (I3) и (I4) не являются достаточными для уменьшения общего числа элементов в ВС. Это происходит по двум основным причинам: I) существование разницы в высотах блоков

ВС, соответствующих различным подграфам G_i ; 2) наличие между подграфами связей, которые требуют затраты строк из соединительных элементов для связей между блоками. Поэтому оценить сложность программы оптимизированной реализации автомата A в ВС можно лишь после разбиения графа A на подграфы следующим образом:

$$R'(A) = t' \left(c \sum_{j=1}^v \lambda(G_j) + v c_2 \right). \quad (15)$$

Сформируем приближенный алгоритм разбиения графа G на подграфы при заданной величине t' .

1°. Для q_j получаем A_j и C_j так, как показано в 2° алгоритма нахождения подстановки множества вершин графа. Переходим к 2°.

2°. Из неотмеченных вершин множества UUL выбираем вершину q_{j1} с $\gamma(q_{j1})_{max}$. Переходим к 3°.

3°. Из неотмеченных вершин формируем подмножество $Q^x \subseteq UUL$, удовлетворяющее одновременно условиям (13) и

$$\sum_j \delta_j^x = max, \quad q_j^x \in Q^x.$$

Это подмножество формируется путем последовательного выбора неотмеченных вершин, начиная с вершины q_{i1} с $\delta_{i1}^x_{max}$. Если выбор этой вершины нарушает условие (13), необходимо выбрать другую вершину q_{i2} с $\delta_{i2}^x_{max} \leq \delta_{i1}^x_{max}$ и т.д. до выполнения условия (13). После выбора очередной вершины числа δ_i^x подсчитываются вновь так, как показано в п.4° предшествующего алгоритма.

Если неотмеченных вершин для выполнения условия (13) недостаточно, формирование множества Q^x считаем законченным, отмечаем его элементы индексом x и переходим к 4°, в противном случае — к 3°.

4°. Конец работы алгоритма.

В результате выполнения алгоритма получаем разбиение графа G на подграфы $G_1, G_2, \dots, G_v$ с множествами вер-

шин $Q^1, Q^2, \dots, Q^v$ соответственно. Оценив сложность неоптимизированной и оптимизированной реализации автомата по формулам (6) и (15), можно сделать вывод об эффективности использованного алгоритма.

Действие алгоритма покажем на примере графа G_3 автомата B для $t' = 18, g = 3$. Граф G_3 разбивается на подграфы G_1 и G_2 с подмножествами вершин

$$Q^1 = \{u_2, u_3, u_5, e_1, e_2\},$$

$$Q^2 = \{u_1, u_4, u_6, e_3\}.$$

На рис. 3 показана схема подачи переменных на программы Π_{e_1} и Π_{e_2} . Суммарная сложность этих двух программ в 1,6 раза меньше, чем сложность программы Π_4 в неоптимизированном варианте.

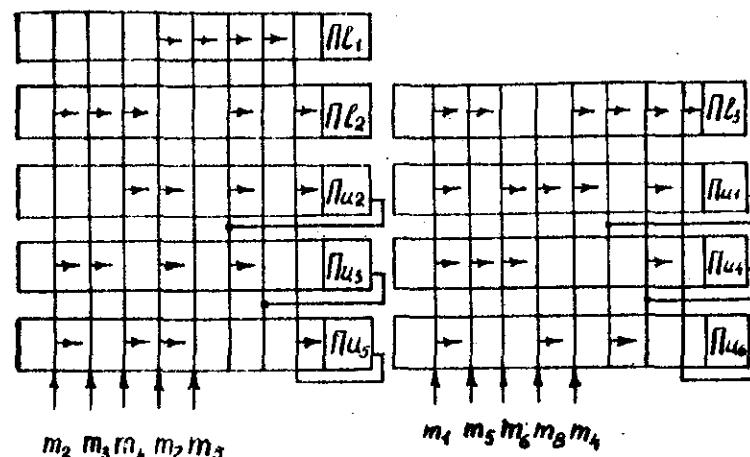


Рис. 3.

Отметим, что этот алгоритм можно использовать при реализации автомата на участках ВС, имеющих различную высоту t'_i . Кроме того, его можно перестроить так, чтобы граф G разбивался на заданное число подграфов с минимальным или близким к нему числом связей между подграфами.

В общем случае невозможно сказать, какой метод оптимиза -

ции из двух предложенных является лучшим. Это зависит от вида синтезируемых автоматов, типа ВС, размеров и формы участка среды и т.д. Однако некоторые выводы в пользу второго метода можно сделать. Во-первых, этот метод по своим возможностям охватывает более широкий класс автоматов, а именно: он может оказаться эффективным даже если для одной или нескольких вершин графа G окажется $\rho(q_i) = 3 + \kappa$, в то время как первым методом оптимизация в этом случае невозможна. Во-вторых, метод разбиения графа G на подграфы может оказаться более приемлемым при синтезе автоматов в "ширной" однородной структуре [5]. В этом случае горизонтальные шины можно использовать для подачи переменных на столбцы программы Π_i блоков, а строки элементов ВС — для получения членов д.н.ф. функций (1) и (2). В то же время тот факт, что на одну шину можно подавать только одну переменную, ограничивает применимость метода подстановки вершин графа G .

Для подграфов со слабой внутренней зависимостью может оказаться эффективным смешанный метод оптимизации, при котором вначале производится разбиение графа G на подграфы, а затем вершины каждого подграфа упорядочиваются с помощью первого метода. Очевидно, что при такой реализации автомата в ВС будет достигаться ещё большая компактность программы реализации автомата в среде.

Отметим, что по сравнению с методом оптимизации размещения графа логической сети в ВС предложенный подход может оказаться более приемлемым в том случае, если число переменных (входных, внутренних и выходных) автомата сравнительно велико, и соответствующий граф логической сети будет достаточно сложным. Если же такой автомат представить графом зависимостей, то он будет иметь во много раз меньшее число вершин. Поэтому предел применимости предложенных здесь методов будет выше. В то же время после оптимизации с помощью графа зависимостей можно производить дальнейшую оптимизацию с использованием графов логических сетей отдельных групп элементарных автоматов, полученных в результате разбиения графа зависимостей G на подграфы.

В заключение заметим, что предложенные методы оптимизации будут эффективны только для тех автоматов, графы зависимостей которых не являются полными или сильно связными.

Л и т е р а т у р а

1. Э.В. ЕВРЕЙНОВ, Ю.Г. КОСАРЕВ. Однородные вычислительные системы высокой производительности, Новосибирск, 1966.
2. В.А. СКОРОБОГАТОВ. Некоторые вопросы технического синтеза логических сетей в вычислительных средах. Диссертация на соискание ученой степени кандидата технических наук, 1969.
3. А.Н. МЕЛИХОВ, Н.Г. ТОПОЛЬСКИЙ. Об одном методе синтеза автоматов в вычислительной среде. — Сб. докладов I Межвузовской конференции по теории и принципам построения ЦМ и ЦИМ, Таганрог, 1970.
4. О.Л. БАНДМАН. Реализация автоматов в криотронной вычислительной среде. — Труды I Всесоюзной конференции по вычислительным системам, Новосибирск, 1968, вып. 3.
5. И.В. ПРАНГИШВИЛИ, М.А. УСКАЧ. Однородные структуры с минимальной задержкой в каналах передачи сигналов. — Материалы ко II Всесоюзной конференции по однородным вычислительным системам и средам, М., 1969.
6. А.Н. МЕЛИХОВ, Л.С. БЕРШТЕЙН, Н.Г. ТОПОЛЬСКИЙ. Декомпозиционный метод структурного синтеза конечных автоматов I, Кибернетика, 1969, № 5.
7. J. HARTMANIS, R.E. STEARNS. Algebraic structure theory of sequential machines, N.Y., 1966.
8. О. ОРБ. Теория графов, "Наука", М., 1968.

Поступила в редакцию
14.IX.70 года