

# ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ

---

УДК 681.326; 681.32

*Н. П. Вашкевич*

## ПАРАЛЛЕЛЬНАЯ ДЕКОМПОЗИЦИЯ УПРАВЛЯЮЩИХ АЛГОРИТМОВ С УЧЕТОМ ИХ НЕОПРЕДЕЛЕННОСТИ<sup>1</sup>

*Аннотация.* Рассматривается методика эквивалентных преобразований управляющих алгоритмов с целью их разбиения на параллельные ветви. Особенностью данных алгоритмов является их неполная определенность.

*Ключевые слова:* управляющий алгоритм, параллельная декомпозиция, неполностью определенный алгоритм, матрица совместимости и включения.

*Abstract.* Technique of equivalent transformations of control algorithms with a view to divide them into parallel branches is considered. The peculiarity of the algorithms is their incomplete definition.

*Keywords:* control algorithm, parallel decomposition, incompletely defined algorithm, compatibility and inclusion matrix.

### Введение

Для многих практических управляющих алгоритмов функции переходов не всегда полностью определены, т.к. некоторые комбинации входных сигналов на таких переходах никогда не возникают. В связи с этим для таких входных сигналов состояние перехода автомата может быть определено произвольно. Это обстоятельство может быть использовано для разбиения управляющего алгоритма на параллельные ветви при условии, что не надо учитывать функции выходов автомата как не влияющих на функциональную связность элементов памяти управляющего устройства, реализующего исходный алгоритм.

Вопросам параллельной декомпозиции управляющих алгоритмов посвящен ряд работ, которые базируются в основном на использовании методов теории графов [1–3] и отличаются довольно значительной сложностью. Данная работа базируется на использовании методики, рассмотренной в работах автора [4, 5], в которых вопросы разбиения событий управляющего алгоритма на группы несовместимых событий решаются путем использования алгоритмического пошагового построения специальной матрицы включения.

Разбиение событий (состояний) на подмножества в рассматриваемой работе, как будет показано в дальнейшем, приводит фактически к кодированию событий (состояний) исходного алгоритма управления. Причем такие коды будут состоять из нескольких частей, каждая из которых представляет

---

<sup>1</sup> Работа выполнена в рамках аналитической целевой программы «Развитие научного потенциала высшей школы (2009–2010), номер гос. регистрации НИР 2.1.2/4257.

код состояния одного из подавтоматов. Задача заключается в том, чтобы обеспечить однозначность кодирования, в соответствии с которым в каждый момент времени работы алгоритма управления имеет место единственное событие (состояние) этого алгоритма, образуемое совокупностью (конъюнкцией) состояний, параллельно работающих подавтоматов, истинных в этот же момент времени.

**1 Методика разбиения событий (состояний) управляющего алгоритма на группы совместимых событий для каждого из параллельно-работающих подавтоматов**

Если исходный управляющий алгоритм представлен, например, в виде неполностью определенной таблицы переходов, то, выполняя попарное сравнение столбцов таблицы, можно определить события (состояния) автомата, которые могут быть объединены. Это возможно тогда, когда для каждого входного сигнала события (состояния) перехода либо одинаковые, либо одно из них или оба вместе неопределены. По результатам такого сравнения строится вспомогательная матрица совместимости таких событий (состояний), которая будет иметь вид (рис. 1), где  $\tau_{\alpha,\beta} = 0$ , если события (состояния)  $S_\alpha$  и  $S_\beta$  несовместимы;  $\tau_{\alpha,\beta} = 1$ , если  $S_\alpha$  и  $S_\beta$  совместимы. Такая матрица используется для построения основной матрицы включения событий (состояний) в группы совместимых событий, каждое из которых образует состояние одного из подавтоматов. Число таких групп, равное числу состояний подавтомата, должно быть не менее максимального числа событий (состояний), которые могут входить в отдельные группы совместимых событий (состояний) исходного алгоритма.

	$S_1$	$S_2$	...	$S_\beta$	...	$S_n$
$S_1$	0	$\tau_{1,2}$	...	$\tau_{1,\beta}$	...	$\tau_{1,n}$
$S_2$	$\tau_{2,1}$	0		$\tau_{2,\beta}$		$\tau_{2,n}$
$S_\alpha$	$\tau_{\alpha,1}$	$\tau_{\alpha,2}$	...	$\tau_{\alpha,\beta}$	...	$\tau_{\alpha,n}$
$S_n$	$\tau_{n,1}$	$\tau_{n,2}$	...	$\tau_{n,\beta}$	...	0

Рис. 1 Вспомогательная матрица совместимости событий (состояний) исходного управляющего алгоритма

При решении задачи распределения событий (состояний) исходного алгоритма по группам совместимых событий для каждого из подавтоматов необходимо руководствоваться следующими условиями:

$$m_i \cap m_j = \emptyset; i \neq j; i, j = \overline{1, H}; \tag{1}$$

$$m_1 \cup m_2 \cup \dots \cup m_n = n, \tag{2}$$

где  $m_i$  – подмножество событий (состояний) исходного алгоритма, входящих в  $i$ -ю группу совместимых событий, формирующих состояние одного из подавтоматов.

Первое условие обеспечивает требование несовместимости событий (состояний), входящих в любые пары из  $H$  групп. Второе условие обеспечивает обязательное включение каждого из всех  $n$  событий (состояний) в какую-либо группу из  $H$  групп, определяющих общее число состояний подавтоматов.

В матрице включения (рис. 2) после завершения ее построения каждая  $i$ -я строка будет содержать подмножество  $m_i$  событий (состояний), размещаемых в одной из групп, а каждый столбец матрицы должен включать только одно событие (состояние). Это следует из условий (1), где  $p_{i,\beta} = 1$ , если  $S_\beta \subset m_i$ ;  $p_{i,\beta} = 0$ , если  $S_\beta \not\subset m_i$ .

	$S_1$	$S_2$	...	$S_\beta$	...	$S_n$
$m_1$	$p_{1,1}$	$p_{1,2}$	...	$p_{1,\beta}$	...	$p_{1,n}$
$m_2$	$p_{2,1}$	$p_{2,2}$	...	$p_{2,\beta}$	...	$p_{2,n}$
$m_i$	$p_{i,1}$	$p_{i,2}$	...	$p_{i,\beta}$	...	$p_{i,n}$
$m_H$	$p_{H,1}$	$p_{H,2}$	...	$p_{H,\beta}$	...	$p_{H,n}$

Рис. 2 Матрица включения событий (состояний) в группы совместимых событий (состояний) для одного из параллельно работающих подавтоматов

Матрица включения строится постепенно, по шагам, число которых равно числу всех событий (состояний) исходного алгоритма. При этом для облегчения решения задачи на каждом шаге используется вспомогательная матрица совместимости событий (состояний) (рис. 1).

На каждом шаге построения матрицы включения для очередного события (состояния)  $S_\alpha$  отыскивается такое подмножество  $m_i$ , для которого рассматриваемое событие (состояние)  $S_\alpha$  с ним совместимо. Это условие может быть представлено таким образом:

$$S_\alpha \cap m_i \neq \emptyset, \text{ когда } m_i \subset S_\alpha, \tag{3}$$

где под  $S_\alpha$  понимается строка вспомогательной матрицы совместимости, отмеченная буквой  $S_\alpha$ . Если условие (3) выполняется, то  $S_\alpha$  может войти в  $m_i$ . В противном случае, когда условие (3) не выполняется, образуется новое подмножество  $m_{i+1}$ , в которое  $S_\alpha$  включается в качестве первого элемента.

Для первого шага построения матрицы включения для  $S_1$  имеем пустое подмножество  $m_1$ , поэтому для  $S_1$  назначают подмножество

$$m_1 = [S_1, 0, \dots, 0].$$

Для второго шага для  $S_2$  определяют результат операции пересечения  $m_1$  со второй строкой вспомогательной матрицы совместимости событий

(состояний), обозначенной буквой  $S_2$ . Если  $m_1 \cap S_2 \neq \emptyset$ , то  $S_2$  войдет в  $m_1$ , в противном случае образуется новое подмножество

$$m_2 = [0, S_2, 0, \dots, 0].$$

Для третьего шага для  $S_3$  определяет результат операции пересечения  $m_1$  и  $m_2$  с третьей строкой вспомогательной матрицы совместимости. Если  $m_1 \cap S_3 \neq \emptyset$  или  $m_2 \cap S_3 \neq \emptyset$ , то  $S_3$  войдет в одно из подмножеств:  $m_1$  или в  $m_2$  соответственно. Если результаты операции пересечения в обоих случаях равны  $\emptyset$ , то образуется новое подмножество  $m_3$ .

Последним шагом для  $S_n$  завершится построение матрицы включения для одного из подавтоматов, и может быть выполнена коррекция (перераспределение) состава подмножеств событий (состояний), если в процессе построения матрицы включения выявилась возможность отнести очередное событие (состояние)  $S_\alpha$  в одно из нескольких подмножеств  $m_i$ .

Коррекция состава подмножеств  $m_i$  позволит в дальнейшем при структурном синтезе полнее использовать кодовые группы для событий (состояний), входящих в подмножество  $m_i$  с учетом того, что с точки зрения минимизации оборудования наиболее выгодно иметь равные числа событий в каждом из подмножеств  $m_i$ .

Коррекция состава построенных подмножеств  $m_i$  одного из подавтоматов может потребоваться и при построении матриц включения других параллельно работающих подавтоматов, реализующих исходный алгоритм управления. В частности, если исходный алгоритм управления реализуется двумя подавтоматами, то для выполнения однозначности и полноты переходов в реализуемом исходном алгоритме необходимо выполнение следующих условий:

$$m_i^1 \cap m_j^2 = S_\alpha, (\forall \alpha) (\alpha = \overline{1, n}), \quad (4)$$

где  $m_i^1$  –  $i$ -е условие подмножества, формирующее  $i$ -е состояние первого подавтомата;  $m_j^2$  –  $j$ -е подмножество, формирующее  $j$ -е состояние второго подавтомата;  $S_\alpha$  – событие (состояние) исходного управляющего алгоритма.

Из выражения (4) следует, что результат пересечения подмножеств событий (состояний), принадлежащих разным подавтоматам, должен быть равен только одному из событий (состояний) исходного управляющего алгоритма, а общее число таких пересечений должно быть равно числу событий (состояний) исходного управляющего алгоритма. Соблюдение этих условий гарантирует полноту и однозначность реализации исходного управляющего алгоритма.

## **2 Пример параллельной декомпозиции неполностью определенного управляющего алгоритма**

Пусть исходный неполностью определенный управляющий алгоритм задан следующей таблицей переходов (рис. 3). В данной работе в качестве

примера, иллюстрирующего предлагаемую методику, взята таблица переходов неполностью определенного автомата из [1].

Входные сигналы	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
$x_0$	$S_1$	–	$S_5$	$S_1$	–	$S_5$	–
$x_1$	–	$S_7$	$S_6$	$S_7$	–	–	–
$x_2$	–	–	–	$S_3$	$S_7$	$S_6$	$S_3$
$x_3$	$S_2$	$S_4$	–	–	–	$S_7$	–

Рис. 3 Таблица переходов неполностью определенного автомата

Вспомогательная матрица совместимости событий (состояний) в соответствии с таблицей переходов исходного автомата будет иметь вид (рис. 4).

	$S_1$	$S_2$	$S_3$	$S_4$	$S_5$	$S_6$	$S_7$
$S_1$	0	0	0	1	1	0	1
$S_2$	0	0	0	1	1	0	1
$S_3$	0	0	0	0	1	1	1
$S_4$	1	1	0	0	0	0	1
$S_5$	1	1	1	0	0	0	1
$S_6$	0	0	1	0	0	0	0
$S_7$	1	1	1	1	1	0	0

Рис. 4 Вспомогательная матрица совместимости событий (состояний)

Исходя из заданного управляющего алгоритма (рис. 3), максимально возможное число совместимых событий, входящих в одну группу, равно 3. К числу таких групп относятся:  $(S_1, S_4, S_7)$ ,  $(S_3, S_5, S_7)$ ,  $(S_2, S_5, S_7)$ , поэтому минимально возможное число состояний в подавтоматах должно быть не менее трех. Всего различных комбинаций подмножеств событий (состояний), входящих в разные подавтоматы, должно быть не менее  $n = 7$ . Для их реализации в соответствии с условием (4) необходимо два параллельно работающих подавтомата.

Рассмотрим вариант построения матрицы включения для первого подавтомата, которое будет содержать следующие шаги:

1. Для события (состояния)  $S_1$  подмножество совместимых событий пусто, поэтому для  $S_1$  назначается первое подмножество  $m_1^1$ , включающее пока один элемент:

$$m_1^1 = 1000000.$$

2. Поскольку  $S_2 \cap m_1^1 = \emptyset$ , то  $S_2$  не включается в  $m_1^1$ , а образуется новое подмножество  $m_2^1$ , в которое войдет  $S_2$  в качестве первого элемента:

$$m_1^1 = 1000000, m_2^1 = 0100000.$$

3. Поскольку  $m_1^1 \cap S_3 = \emptyset$  и  $m_2^1 \cap S_3 = \emptyset$ , то  $S_3$  не включается ни в  $m_1^1$ , ни в  $m_2^1$ , а образуется новое подмножество  $m_3^1$ , в которое войдет  $S_3$  в качестве первого элемента:

$$m_1^1 = 1000000, m_2^1 = 0100000, m_3^1 = 0010000.$$

4. Поскольку  $m_1^1 \cap S_4 \neq \emptyset$ ,  $m_2^1 \cap S_4 \neq \emptyset$ ,  $m_3^1 \cap S_4 = \emptyset$ , то  $S_4$  может войти в  $m_1^1$  или  $m_2^1$ , а в  $m_3^1$  не войдет. Поэтому поместим  $S_4$  в  $m_1^1$ , имея при этом ввиду вариант возможного перераспределения переменной  $S_4$  в этих подмножествах. Тогда состав подмножеств  $m_i$  на данном шаге будет следующим:

$$m_1^1 = 1001000, m_2^1 = 0100000, m_3^1 = 0010000.$$

5. Поскольку  $m_1^1 \cap S_5 \neq \emptyset$ ,  $m_2^1 \cap S_5 \neq \emptyset$ ,  $m_3^1 \cap S_5 \neq \emptyset$ , то  $S_5$  может войти в любое из трех подмножеств  $m_1^1$ ,  $m_2^1$  или  $m_3^1$ . Учитывая равномерность распределения переменных  $S_\alpha$  в подмножествах  $m_i^1$ , поместим  $S_5$  в подмножество  $m_2^1$ , имея ввиду вариант возможного перераспределения переменной  $S_5$  в этих подмножествах. Тогда состав подмножеств  $m_i$  на данном шаге будет иметь вид

$$m_1^1 = 1001000, m_2^1 = 0100100, m_3^1 = 0010000.$$

6. Поскольку  $m_1^1 \cap S_6 = \emptyset$ ,  $m_2^1 \cap S_6 = \emptyset$ ,  $m_3^1 \cap S_6 \neq \emptyset$ , то  $S_6$  может войти только в  $m_3^1$ . Тогда состав подмножеств  $m_i$  на данном шаге будет иметь вид

$$m_1^1 = 1001000, m_2^1 = 0100100, m_3^1 = 0010010.$$

7. Поскольку  $m_1^1 \cap S_7 \neq \emptyset$ ,  $m_2^1 \cap S_7 \neq \emptyset$ ,  $m_3^1 \cap S_7 \neq \emptyset$ , то  $S_7$  может войти в любое из подмножеств  $m_1^1$ ,  $m_2^1$ ,  $m_3^1$ . Поместим  $S_7$  в подмножество  $m_1^1$ , учитывая при этом возможность перераспределения переменной  $S_7$  в этих подмножествах. Тогда получим состав подмножеств  $m_i$ , соответствующих последнему шагу построения матрицы включения для первого подавтомата:

$$m_1^1 = 1001001, m_2^1 = 0100100, m_3^1 = 0010010. \quad (5)$$

Рассмотрим теперь вариант построения матрицы включения для второго подавтомата. При ее построении необходимо учитывать также и выполнение условий (4):

1. Очевидно, что первые три шага построения матрицы включения для второго подавтомата будут совпадать с тремя шагами построения матрицы включения для первого подавтомата. Тогда имеем

$$m_1^2 = 1000000, m_2^2 = 0100000, m_3^2 = 0010000.$$

2. Поскольку  $m_1^2 \cap S_4 \neq \emptyset$ ,  $m_2^2 \cap S_4 \neq \emptyset$ ,  $m_3^2 \cap S_4 = \emptyset$ , то  $S_4$  нельзя поместить в подмножество  $m_3^2$ . В то же время  $S_4$  нельзя также поместить и

в подмножество  $m_1^2$ , т.к. при помещении  $S_4$  в  $m_1^2$  условие (4) не будет соблюдаться, что следует из результата операции пересечения  $m_1^2 \cap m_1^1 = S_1 S_4$ . Тогда получим следующий состав подмножеств на данном шаге:

$$m_1^2 = 1000000, m_2^2 = 0101000, m_3^2 = 0010000.$$

3. Поскольку  $m_1^2 \cap S_5 \neq \emptyset, m_2^2 \cap S_5 = \emptyset, m_3^2 \cap S_5 \neq \emptyset$ , то  $S_5$  нельзя поместить в  $m_2^2$  из-за того, что не выполнялось бы условие (4), т.к. имело бы место соотношение  $m_2^1 \cap m_2^2 = S_2 S_5$ , что нарушает однозначность переходов. В то же время  $S_5$  можно поместить или в  $m_1^2$ , или в  $m_3^2$ . Выбираем произвольно первый вариант, учитывая при этом возможность перераспределения переменной  $S_5$  в эти подмножества. В результате получим следующий состав подмножеств на данном шаге:

$$m_1^2 = 1000100, m_2^2 = 0101000, m_3^2 = 0010000.$$

4. Поскольку  $m_1^2 \cap S_6 = \emptyset, m_2^2 \cap S_6 = \emptyset, m_3^2 \cap S_6 \neq \emptyset$ , то  $S_6$  нельзя поместить ни в одно из подмножеств, в том числе и в  $m_3^2$ . Это объясняется тем, что если бы  $S_6$  вошло в подмножество  $m_3^2$ , то не выполнялось бы условие (4), т.к.  $m_3^1 \cap m_3^2 = S_3 S_6$ , что вызвало бы неоднозначность переходов. В связи с этим образуется новое подмножество  $m_4^2$ , в которое вводится  $S_6$ . Учитывая это обстоятельство, получим следующий состав подмножеств:

$$m_1^2 = 1000100, m_2^2 = 0101000, m_3^2 = 0010000, m_4^2 = 0000010.$$

5. Для последнего шага для  $S_7$  имеем следующие соотношения:  $m_1^2 \cap S_7 \neq \emptyset, m_2^2 \cap S_7 \neq \emptyset, m_3^2 \cap S_7 \neq \emptyset, m_4^2 \cap S_7 = \emptyset$ . Откуда следует, что  $S_7$  можно поместить только в подмножество  $m_3^2$ , а в подмножества  $m_1^2$  или  $m_2^2$  – нельзя из-за невыполнения условия (4), т.к. имели бы место соотношения  $m_1^2 \cap m_1^1 = S_1 S_7$  и  $m_2^2 \cap m_1^1 = S_4 S_7$ , что нарушило бы однозначность переходов. Тогда окончательный вариант размещения событий (состояний) по подмножествам для второго подавтомата будет иметь следующий вид:

$$m_1^2 = 1000100, m_2^2 = 0101000, m_3^2 = 0010001, m_4^2 = 0000010. \quad (6)$$

### **3 Представление алгоритмов функционирования параллельно работающих подавтоматов по результатам декомпозиции исходного управляющего алгоритма**

По результатам построения матриц включения (5), (6), определяющих разбиение исходного управляющего алгоритма на параллельно работающие подавтоматы, получим коды всех событий (состояний) исходного управляющего алгоритма. Эти коды удовлетворяют требованию условий (4), обеспечивающих однозначность и полноту переходов алгоритма. Для нашего примера такие коды событий (состояний) имеют следующий вид:

$$S_1 = m_1^1 m_1^2, S_2 = m_2^1 m_2^2, S_3 = m_3^1 m_3^2, S_4 = m_1^1 m_2^2, \\ S_5 = m_2^1 m_1^2, S_6 = m_3^1 m_4^2, S_7 = m_1^1 m_3^2, \quad (7)$$

где  $m_i^1$  и  $m_j^2$  – коды внутренних состояний первого и второго подавтоматов соответственно,  $i = [1, 2, 3], j = [1, 2, 3, 4]$ .

Подставляя коды событий (состояний) исходного автомата (7) в таблицу переходов (рис. 3), можно построить кодированную таблицу переходов исходного управляющего автомата, которая будет иметь вид (рис. 5).

	$m_1^1 m_1^2$	$m_2^1 m_2^2$	$m_3^1 m_3^2$	$m_1^1 m_2^2$	$m_2^1 m_1^2$	$m_3^1 m_4^2$	$m_1^1 m_3^2$
$x_0$	$m_1^1 m_1^2$	–	$m_2^1 m_1^2$	$m_1^1 m_1^2$	–	$m_2^1 m_1^2$	–
$x_1$	–	$m_1^1 m_3^2$	$m_3^1 m_4^2$	$m_1^1 m_3^2$	–	–	–
$x_2$	–	–	–	$m_3^1 m_3^2$	$m_1^1 m_3^2$	$m_3^1 m_4^2$	$m_3^1 m_3^2$
$x_3$	$m_2^1 m_2^2$	$m_1^1 m_2^2$	–	–	–	$m_1^1 m_1^2$	–

Рис. 5 Кодированная таблица переходов исходного управляющего автомата

По кодированной таблице переходов (рис. 5) нетрудно построить системы канонических уравнений (СКУ), формализующих алгоритмы функционирования отдельных подавтоматов. Введем обозначения для подмножеств внутренних состояний первого и второго подавтоматов, имеющих вид:

– для первого подавтомата:

$$\left[ a_1 = m_1^1, a_2 = m_2^1, a_3 = m_3^1 \right];$$

– для второго подавтомата:

$$\left[ b_1 = m_1^2, b_2 = m_2^2, b_3 = m_3^2, b_4 = m_4^2 \right];$$

искомые СКУ будут представлены следующим образом:

– для первого подавтомата:

$$a_1(t+1) = a_1(x_0 \vee x_1) \vee a_2(x_1 \vee x_2 \vee x_3) \vee a_3 x_3, \\ a_2(t+1) = a_1 x_3 \vee a_3 x_0, \quad a_3(t+1) = a_1 x_2 \vee a_3(x_1 \vee x_2); \quad (8)$$

– для второго подавтомата:

$$b_1(t+1) = (b_1 \vee b_2 \vee b_3 \vee b_4) x_0 \vee b_4 x_3, \quad b_2(t+1) = (b_1 \vee b_2) x_3, \\ b_3(t+1) = (b_1 \vee b_2 \vee b_3) x_2 \vee b_2 x_1, \quad b_4(t+1) = b_3 x_1 \vee b_4 x_2. \quad (9)$$

По полученным СКУ (8) и (9), имеющим стандартный вид, можно построить структурные схемы подавтоматов с использованием методов аппаратной или микропрограммной реализации.

### **Заключение**

В заключение отметим некоторые особенности рассмотренного метода реализации параллельной декомпозиции управляющего алгоритма и его автоматной интерпретации:



1) при построении матриц включения было обращено внимание на то, что для некоторых шагов построения таких матриц имелась возможность поместить очередное событие (состояние) в одно из нескольких подмножеств. Это обстоятельство может быть использовано при формировании состава внутренних состояний отдельных подавтоматов, когда необходимо учитывать оценку равномерной сложности структуры отдельных подавтоматов и зависимость по ресурсам отдельных событий, реализуемых системой;

2) автоматное представление параллельной композиции системы логического управления отличается большой компактностью и удобством моделирования системы особенно при весьма большом числе состояний системы управления, т.к. максимальное число состояний системы может быть представлено как произведение числа состояний параллельно работающих подавтоматов.

При этом любое свойство системы управления при ее верификации может быть проверено на основе моделирования СКУ отдельных подавтоматов, общее число состояний которых во много раз меньше числа состояний системы управления. При полной верификации системы управления выделяется начальное состояние системы в виде совокупности начальных состояний параллельно работающих подавтоматов и может быть использован алгоритм детерминизации, рассмотренный в работах [4, 5].

#### *Список литературы*

1. **Горбатов, В. А.** Регулярные структуры автоматного управления / В. А. Горбатов, Б. Л. Останков, С. А. Фролов. – М. : Машиностроение, 1980. – 216 с.
2. **Горбатов, В. А.** Логическое управление распределенными системами / В. А. Горбатов, М. И. Смирнов, И. С. Хлытчиев. – М. : Энергоатомиздат, 1991. – 284 с.
3. **Мелихов, А. Н.** Ориентированные графы и конечные автоматы / А. Н. Мелихов. – М. : Наука, 1971. – 416 с.
4. **Вашкевич, Н. П.** Недетерминированные автоматы и их использование для синтеза систем управления / Н. П. Вашкевич, С. Н. Вашкевич. – Пенза : Изд-во Пенз. гос. ун-та, 1996. – 88 с.
5. **Вашкевич, Н. П.** Недетерминированные автоматы в проектировании систем параллельной обработки / Н. П. Вашкевич. – Пенза : Изд-во Пенз. гос. ун-та, 2004. – 280 с.

---

**Вашкевич Николай Петрович**  
доктор технических наук, профессор,  
кафедра вычислительной техники,  
Пензенский государственный  
университет

E-mail: vt@alice.pnzgu.ru

**Vashkevich Nikolay Petrovich**  
Doctor of engineering sciences, professor,  
sub-department of computer science,  
Penza State University

---

УДК 681.326; 681.32

**Вашкевич, Н. П.**

**Параллельная декомпозиция управляющих алгоритмов с учетом их неопределенности / Н. П. Вашкевич // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2009. – № 3 (11). – С. 3–11.**