

**UNDETERMINED EVENT AUTOMATES MODELS  
USED IN ORDER TO FORMALLY DESCRIBE  
PARALLEL ALGORITHMS  
OF LOGICAL MANAGEMENT**

**N. P. Vashkevich<sup>1</sup>, R. A. Biktashev<sup>1</sup>, D. V. Pashenko<sup>1</sup>, V. V. Kutuzov<sup>1</sup>, K. T. Sauanova<sup>2</sup>**

<sup>1</sup>Penza State University, Russia,

<sup>2</sup>Almaty University of Power Engineering & Telecommunications, Kazakhstan.

E-mail: klartag@mail.ru

**Key words:** parallel algorithms, systems of logical management, control algorithms, undetermined event automates.

**Abstract.** Considered the mathematical apparatus for an analytical description of parallel algorithms for logic control, based on undetermined event automates usage, represented as a standard system of canonical recurrent equations, which describe all of realizable partial events in the algorithm.

Showing expressive possibilities and effectiveness of UEA (Undetermined Event Automates) language, taking into account the relationships between partial events such as management and data taking into account the behavior of the control system in time.

In this resume considered relations between UEA language with other algorithms' describing languages, also there is an opportunity to transform the hardware describing languages for structural management algorithms implementation.

The effectiveness of the use of language UEA demonstrated on examples of formalization control algorithm synchronization interacting parallel processes with reference to the general critical resources with the use of critical sections.

In the article the mathematical apparatus for analytical description of parallel logic control algorithms based on the use patterns of event nondeterministic (event-NDA or SNDA) is revised, presented in the form of a standard system of canonical recurrence equations (MCS), which describe the algorithm implemented in all private events. When choosing a method of describing algorithms for management the main advantage of event nondeterministic models (event-NDA or SNDA) must be taken into consideration, it results in hiding them parallelism, which is manifested in the fact that under the influence of the same input signal may be a transition from one event to the private several at the same time there are events.

The paper demonstrates the expressive possibilities of SNDA language and efficiency, taking into account the connection between private events such as management, and on information, taking into account the behavior of the control system over time, which allows you to successfully use this language to verify the control algorithms.

The relationship of SNDA language with other languages describing algorithms, as well as the possibility of transformation to the hardware description languages such as VHDL and Verilog, for the implementation of the structural control algorithms is demonstrated.

The effectiveness of the use of SNDA language is demonstrated by examples of formalization control algorithm timing interacting parallel processes by reference to general critical resources with the use of critical sections.

It is shown that MCS can be made all kinds of equivalent conversion in accordance with the laws of Boolean algebra and discrete mathematics, including the use of well-known mathematical apparatus of the theory of finite automata.

The advantages of using the model and its analytical and graphical interpretation highlighted in the article SNDA, allow to not that such an approach could allow for a comprehensive solution to the issues of the specification, design, implementation, verification and analysis of complex control systems, including the management of processes and resources in parallel and distributed computing systems.

## ИСПОЛЬЗОВАНИЕ МОДЕЛЕЙ СОБЫТИЙНЫХ НЕДЕТЕРМИНИРОВАННЫХ АВТОМАТОВ ДЛЯ ФОРМАЛЬНОГО ОПИСАНИЯ ПАРАЛЛЕЛЬНЫХ АЛГОРИТМОВ ЛОГИЧЕСКОГО УПРАВЛЕНИЯ

Н. П. Вашкевич<sup>1</sup>, Р. А. Бикташев<sup>1</sup>, Д. В. Пашенко<sup>1</sup>, В. В. Кутузов<sup>1</sup>, К. Т. Сауанова<sup>2</sup>

<sup>1</sup>Россия, Пензенский государственный университет, Россия,

<sup>2</sup>Алматинский университет энергетики и связи, Казахстан

**Ключевые слова:** параллельные алгоритмы, системы логического управления, управляющие алгоритмы, событийные недетерминированные автоматы.

**Аннотация.** В статье рассматривается математический аппарат для аналитического описания параллельных алгоритмов логического управления основанный на использовании моделей событийных недетерминированных автоматов (событийных НДА, или СНДА), представленных в виде стандартной системы канонических рекуррентных уравнений (СКУ), описывающих все реализуемые в алгоритме частные события. При выборе методов описания алгоритмов управления следует учитывать основное достоинство моделей событийных недетерминированных автоматов (событийных НДА, или СНДА), заключающееся в скрытом в них параллелизме, который проявляется в том, что под действием одного и того же входного сигнала допускается переход от одного частного события к нескольким одновременно существующим событиям.

В работе показаны выразительные возможности и эффективность языка СНДА, учитывающего связи между частными событиями, как по управлению, так и по данным, с учётом поведения системы управления во времени, что позволяет успешно применять данный язык для верификации управляющих алгоритмов.

Продемонстрирована связь языка СНДА с другими языками описания алгоритмов, а также возможность трансформации на языки описания аппаратуры, такие как VHDL и Verilog, для структурной реализации алгоритмов управления.

Эффективность использования языка СНДА демонстрируется на примерах формализации алгоритма управления синхронизацией взаимодействующими параллельными процессами при обращении к общему критическому ресурсу с использованием механизма критических секций.

Показывается, что для СКУ могут быть выполнены всевозможные эквивалентные преобразования в соответствии с законами булевой алгебры и дискретной математики, в том числе с использованием широко известного математического аппарата теории конечных автоматов.

Освещённые в статье достоинства использования модели СНДА и её аналитической и графической интерпретации, позволяют отметить, что такой подход может позволить обеспечить комплексное решение вопросов спецификации, разработки, реализации, верификации и анализа сложных систем управления, в том числе управления процессами и ресурсами в параллельных и распределённых вычислительных системах.

**Введение.** При создании систем логического управления (СЛУ) применяют формальные методы описания управляющих алгоритмов (УА), которые должны обеспечивать комплексное решение задач спецификации, разработки, верификации, анализа и реализации сложных систем управления, в том числе систем управления взаимодействующими параллельными процессами и ресурсами в параллельных и распределённых системах. Математический аппарат должен обладать высокой эффективностью, позволяющей решать задачи, связанные с повышением производительности, надёжности и безопасности систем управления. Под высокой эффективностью понимается выполнение следующих основных требований [1]:

- простота, наглядность, компактность и удобство использования для заказчика и разработчика;

- высокая степень выразительности при описании сложных алгоритмов, включающих как синхронные, так и асинхронные взаимодействующие процессы, с отсутствием тупиковых ситуаций и отражением временного фактора;

- наличие методов оптимизационных равносильных преобразований, связанных с минимизацией, композицией, декомпозицией и верификацией УА на моделях [2] и возможностью простой трансформации описания с одного языка на другой;

- простой переход от формального аналитического представления к структурной реализации УА;
- математический аппарат должен быть ориентирован на описание параллельно-последовательных алгоритмов;
- при простоте математического аппарата языка должно быть обеспечено описание широкого круга применений по управлению процессами, ресурсами, начиная с торговых автоматов и кончая распределенными операционными системами [1].

При выборе методов описания алгоритмов управления следует учитывать основное достоинство моделей событийных недетерминированных автоматов (событийных НДА, или СНДА), заключающееся в скрытом в них параллелизме, который проявляется в том, что под действием одного и того же входного сигнала допускается переход от одного частного события к нескольким одновременно существующим событиям. Это обстоятельство является отличительной особенностью СНДА, где описание переходов выполняется не в терминах состояний детерминированных автоматов (ДА), а в терминах частных событий, что и обеспечивает такую же простоту, компактность и удобство описания параллельных алгоритмов [3,4], какое обеспечивают ДА для описания последовательных алгоритмов. В дальнейшем аналитическую форму представления управляющего алгоритма в виде систем канонических уравнений (СКУ) будем для краткости называть языком НД СКУ.

Другие полезные приложения, касающиеся теории недетерминированных автоматов, обсуждаются в статьях [5-13].

В статье [5] большое внимание уделяется теории диспетчерского управления для недетерминированных дискретно-событийных систем, представляющей спецификации как языка, так и траекторных моделей. Показано, что хорошо известные алгоритмы для супервизорного управления, построенные на основе детерминированных систем, могут быть в некоторых случаях адаптированы для синтеза супервизоров недетерминированных систем, основанного на спецификациях языка и траекторных моделей.

Статья [6] касается расширения определений недетерминированных процессов. Другими словами, рассматривается случай, когда как система, так и контроллер могут быть представлены моделью недетерминированного автомата.

В статьях [7, 8] основное внимание уделяется установке режима оптимального управления для недетерминированных систем переходов на основе спецификаций временной логики. Авторы используют элементы временной логики для спецификации таких задач, как обеспечение безопасной навигации, реакции на окружающую среду, живучести и наблюдаемости.

В статье [9] показана реализация регулярных выражений для схем согласования, основанных на модели МНДА (модульных недетерминированных конечных автоматов). Кроме того, в этой работе показано, что сложность параллельных аппаратных средств, реализованных на базе НДА, ниже, чем у тех, которые построены на основе модели ДА.

В статьях [10, 11] дан подробный анализ абстрактной модели НДА при реализации супервизорного управления.

В статье [12] большое внимание уделяется применению НДА для детальной проверки сетевых пакетов. Представлены новые алгоритмы для эффективного построения НДА для пакетной проверки и дано пояснение, каким образом эти новые методы могут превзойти существующие алгоритмы с точки зрения необходимого времени и памяти. Абстрактная модель НДА для супервизорного управления анализируется в статье [13].

Этот краткий обзор показывает, что до сих пор малое внимание уделялось приложениям НДА, связанным с синтезом параллельных и конкурирующих устройств управления. Поэтому в настоящей статье уделяется большое внимание модели НДА и ее использованию для формального описания параллельных систем логического управления.

**Простая каноническая форма языка НД СКУ.** Для формального аналитического представления УА может быть использована простейшая модель конечного цифрового НДА Мура, представленного в виде стандартной системы канонических рекуррентных бескванторных предикатных уравнений вида (1), описывающих все реализуемые в алгоритме частные события  $S_j$ :

$$S_j^{Y_j}(t+1) = \bigvee_{i,j} X_{i,j}(t) \& S_i(t) \vee \bigvee_j X_{j,j}(t) \& S_j(t), \quad j = \overline{0, m}, \quad S_0(0) = 1, \quad (1)$$

где  $X_{i,j}$  – частный входной сигнал, представляющий собой сочетание (конъюнкцию) элементарных двоичных входных сигналов из структурного алфавита [  $X$  ];  $S_i(t)$  – сокращенное обозначение выражения, описывающего частное событие  $S_j$ , непосредственно предшествующее событию  $S_j$  (в дальнейшем для простоты записи знак конъюнкции и время  $(t)$  в правой части уравнений типа СКУ (1) будем опускать);  $Y_j$  – частный выходной сигнал, представляющий собой сочетание элементарных двоичных выходных сигналов из структурного алфавита [  $Y$  ], отмечающих событие  $S_j$ .

В системе канонических уравнений (1) первая часть каждого уравнения формализует описание условий первоначального появления или зарождения события  $S_j$ , а вторая – формализует описание условий возобновления или сохранения события  $S_j$ .

Таким образом, простая каноническая форма языка НД СКУ определяет только причинно-следственные связи между событиями  $S_i$  и  $S_j$ , т.е. связь по управлению и по данным. Для такой формы представления событий предполагается, что событие  $S_j$  может выполняться сразу же после события  $S_i$ , когда выработаны необходимые сигналы и данные для реализации события  $S_j$ .

**Обобщенная каноническая форма языка НД СКУ, позволяющая расширить его выразительные возможности.** Расширение выразительных возможностей формального языка НД СКУ достигается путем учёта временных факторов для асинхронного режима работы системы управления. Реализация таких возможностей языка НД СКУ осуществляется варьированием описаний условий зарождения и сохранения события  $S_j$ , которые могут зависеть не только от значений внешних частных входных сигналов  $X_{i,j}$ , но и от значений совокупности некоторых частных событий, имеющих место в моменты времени, предшествующие событию  $S_j$ . В связи с этим математическую модель СНДА Мура можно представить в виде следующей обобщённой канонической формы (2) [3]:

$$S_j^{Y_j}(t+1) = \bigvee_{i,j} S_{j,z} S_i \vee \bigvee_j S_{j,c} S_j, \quad (2)$$

где  $S_{j,z}$  и  $S_{j,c}$  – комбинационные события, определяющие условия зарождения и сохранения события  $S_j$ , можно представить так:  $S_{j,z} = X_{i,j} R_i$ ,  $S_{j,c} = X_{j,j} R_j$ , где  $R_i$  и  $R_j$  являются конъюнкциями переменных, представляющих реализуемые в параллельных ветвях некоторые частные события, влияющие на условия зарождения и сохранения события  $S_j$  соответственно, т.е. такие события можно представить так:  $R_i = \tilde{S}_{i,1} \tilde{S}_{i,2} \cdots \tilde{S}_{i,k}$ ,  $R_j = \tilde{S}_{j,1} \tilde{S}_{j,2} \cdots \tilde{S}_{j,r}$ , где  $\sim$  – знак, означающий, что переменные (частные события), входящие в  $R_i$  и  $R_j$ , могут быть взяты с отрицанием или без него.

События, входящие в  $R_i$  и  $R_j$ , могут иметь место как внутри рассматриваемого управляющего устройства, так и вне его (в последнем случае события представляются как внешние входные сигналы). К их числу могут относиться также такие события, которые при параллельной обработке информации определяют взаимоисключение событий типа  $S_j$  и их приоритетность, а также события, отражающие временной фактор, определяющий продолжительность, начало и конец обработки информации, характерной для асинхронного режима работы.

После построения системы НД СКУ ее проверяют на отсутствие в ней недостижимых событий, т.е. событий, которые не имеют событий предшественников. Чтобы убедиться в этом необходимо по исходной СКУ построить прямую таблицу переходов. Она должна строиться таким образом, чтобы очередное событие можно было бы занести в таблицу в качестве исходного только в том случае, если оно имело место в одном из предшествующих шагов построения такой таблицы.

**Отражение временного фактора в управляющих алгоритмах, определяющих поведение системы управления во времени.** Для асинхронного режима работы характерно использование ждущих и временных логических условий [3, 4]. Для его реализации вводится дополнительное пустое событие  $S_r$ , которое имитирует ожидание появления истинности некоторого события  $S_p$ , обеспечивающего переход от события  $S_i$  к событию  $S_j$ . В данном случае под логическим условием

$S_p$  понимается выполнение некоторого временного события, например,  $t_n \leq T \leq t_b$ , или окончание каких-либо действий, которые вызывают переход от события  $S_i$  к событию  $S_j$ . Тогда уравнения, описывающие события будут иметь вид:

$$S_j(t+1) = S_r S_p, \quad S_r(t+1) = (S_i \vee S_r) \bar{S}_p. \quad (3)$$

Анализируя выразительные возможности структуры модели НД СКУ (2) для описания УА, определяющих поведение системы во времени, когда порядок выполнения событий описывается без привлечения времени в явном виде, можно утверждать, что язык НД СКУ является по существу одной из разновидностей языка временной темпоральной логики типа CTL [2]. В связи с этим этот язык может быть успешно использован и для верификации управляющих алгоритмов. Для этого исследуемые основные свойства управляющего алгоритма описываются в виде формальной модели на языке логики СНДА, преобразуют её в программный код на языке VHDL и определяют истинность таких свойств.

**Связь языка НД СКУ с другими языками описания управляющих алгоритмов.** Предлагаемая обобщённая каноническая форма представления УА в виде НД СКУ (2) позволяющая естественно и просто переводить описание алгоритма на другие языки, например, описания аппаратуры (Verilog, VHDL) [4]. Действительно, при представлении исходного УА в виде программы на языке VHDL, правые части всех уравнений исходной СКУ НДА преобразуются в коды, которые имеют такой же вид, что и исходные уравнения, но только с другими обозначениями переменных и операций над ними. Например, для события типа  $S_k^i$  входа процесса в критический интервал имеем [4]:

$$S_k^i(t+1) = S_{vp}^i S_{vz}^i S_{pr}^i \vee \overline{S_k^i S_p^i}, \quad sk < = (svp \text{ and } svz \text{ and } spr) \text{ or } (sk \text{ and } (not \ sp)).$$

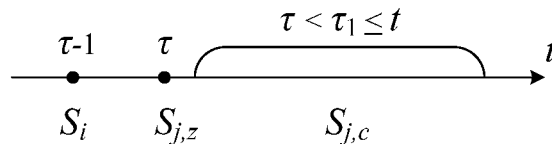
Кроме того, модель СНДА может быть использована для преобразования описания управляющих алгоритмов, представленных на широко известных начальных языках, в виде стандартной системы НДСКУ. К числу таких языков относятся, например, язык регулярных выражений алгебры событий (РВАС), язык исчисления предикатов первого порядка, язык операторных схем алгоритмов с параллельными ветвями (ГСАП) и др. [3].

Покажем такое преобразование на примере описания события  $S_j^{Yj}$ , включающее одну ветвь на языке НД СКУ (2). Представление этого описания на языке РВАС для свёрнутой и развёрнутой формы имеет вид:

$$S_j^{Yj} = S_i S_{j,z} \{ S_{j,c} \} = S_i S_{j,z} \vee S_j S_{j,c}. \quad (4)$$

На языке исчисления предикатов первого порядка с ограниченными кванторами и временной схемой его формирования событие  $S_j^{Yj}$  будет иметь вид:

$$S_j^{Yj}(t) = (\exists \tau) S_{j,z}(\tau) \& \bigvee_{\tau < \tau_1 \leq t} S_{j,c}(\tau_1) \& S_i(\tau-1), \quad (5)$$



Как видно из (4) описание события  $S_j$  представлено на основе использования трёх основных операций: конкатенации, дизъюнкции и итерации, а описание (5) – с использованием кванторов существования ( $\exists$ ), всеобщности ( $\forall$ ) и операции конъюнкции ( $\&$ ).

После введения дискретного времени и замены знака конкатенации на знак конъюнкции уравнение (4) примет вид, соответствующий языку НД СКУ:

$$S_j^{Yj}(t+1) = S_i(t) \& S_{j,z}(t) \vee S_j(t) \& S_{j,c}(t).$$

Аналогичный результат получим и для уравнения (5) после спуска кванторов [3].

**Пример формализации алгоритма управления ресурсами с использованием механизма критических участков программ.** Основной задачи «обращение к общему критическому ресурсу», является формализация функций взаимного исключения критических участков, т.е. таких участков программы, которые содержат группу операторов, обеспечивающих доступ к разделяемым данным или устройствам. На критические участки налагаются три основных требования [14, 15]:

1) В любой момент времени только один процесс может находиться внутри критического участка.

2) Ни один процесс не может оставаться внутри критического участка бесконечно долго.

3) Ни один процесс не должен бесконечно долго ждать входа в критический участок.

*Первое требование*, предъявляемое к критическим участкам, связано с взаимным исключением (несовместимостью) событий, обеспечивающих как входы процессов в свои критические участки, так и нахождение процессов в своих критических участках с учетом принятой дисциплины приоритетности процессов.

*Второе требование*, предъявляемое к критическим участкам, связано с таким описанием условий сохранения событий, обеспечивающих входы процессов в свои критические участки, для которых эти события будут существовать, пока не закончится процедура однократного обращения к разделяемым данным.

Учитывая эти замечания, система уравнений для событий, определяющих входы для любого  $i$ -го процесса в свой критический участок, будет иметь вид:

$S_k^i(t+1) = S_{vp}^i(t) \& S_{vz}^i(t) \& S_{pr}^i(t) \vee S_k^i(t) \bar{S}_p^i$ ,  $i = \overline{1, n^*}$ , где  $S_k^i$  – событие, определяющее

вход и нахождение  $i$ -го процесса в своём критическом участке;  $S_{vp}^i$  – событие, определяющее приём заявки  $i$ -го процесса на обслуживание для обращения к разделяемым данным; это событие является непосредственно предшествующим событием для события  $S_k^i$ , обеспечивая его

достижимость;  $S_p^i$  – событие, обеспечивающее выход  $i$ -го процесса из критического участка после окончания процедуры обращения к разделяемым данным;  $S_{vz}^i$  – комбинационное событие, обеспечивающее взаимное исключение критических интервалов на основе несовместимости событий

$S_k^i$  с другими событиями из их общего числа, равного  $n$ :  $S_{vz}^i = \bigwedge_{(\forall \alpha)[\alpha \neq i]} \bar{S}_k^\alpha$ ,  $i = \overline{1, n}$ , где

$S_{pr}^i$  – событие, обеспечивающее заданное приоритетное обслуживание  $i$ -го процесса.

Событие  $S_{pr}^i$ , например, для циклической дисциплины обслуживания с относительным приоритетом, может быть представлено следующим выражением:

$$S_{pr}^i(t+1) = S_{pr}^i(0) \vee S_{vp}^i S_T^i, \quad (6)$$

где  $S_{pr}^i(0)$  – первая составляющая выражения (6) представляет собой обозначение комбинационного события, определяющего начальный приоритет обслуживания  $i$ -го процесса

$S_{pr}^i(0) = S_0 x_n S_{vp}^i \bigwedge_{(\forall \alpha)[\alpha < i]} \bar{S}_{vp}^\alpha$ , где  $S_0$  – начальное событие системы управления перед обращением

к разделяемому ресурсу  $S_0(t+1) = x_0 \vee S_0 \bar{x}_n$ , где  $x_n$  и  $x_0$  – сигналы инициализации системы

управления и приведения ее в начальное состояние соответственно;  $S_T^i = \bigvee_{k=1}^n (S_{pk}^i \bigwedge_{(\forall \alpha)[\alpha > i]} \bar{S}_{vp}^\alpha)$ , где

$S_T^i$  – комбинационное событие, определяющее приоритет  $i$ -го процесса при наличии воспринятой

заявки в повторных циклах циклической дисциплины обслуживания;  $S_{pk}^i$  – событие, определяемое как событие  $S_p^i$ , задержанное на один такт; это событие используется для принятого варианта организации циклической дисциплины обслуживания.

Введение события  $S_{pr}^i(0)$  в условие зарождения события  $S_k^i$  объясняется необходимостью обеспечения функции взаимоисключения на начальном этапе работы алгоритма управления процессами, чтобы в один и тот же момент времени не могли бы зародиться сразу несколько событий типа  $S_k^i$ .

Третье требование, предъявляемое к критическим участкам, связано с такой формализацией событий, определяющих приём заявок на обслуживание при обращении к разделяемым данным, когда заявка какого-либо процесса на обслуживание воспринимается только в том случае, когда данный процесс не находится в своем критическом участке. Тем самым исключается повторное обращение к разделяемым данным для процесса, имеющего высокий приоритет, по сравнению обслуживаемой заявки другого процесса, имеющего низкий приоритет. Учитывая эти замечания, событие, определяющее приём заявки для  $i$ -процесса, будет иметь следующий вид:

$S_{vp}^i(t+1) = (S_z^i \vee S_{vp}^i) \overline{S}_k^i$ , где  $S_z^i$  – событие, определяющее наличие заявки  $i$ -го процесса на обслуживание.

Учитывая основные требования к организации межпроцессного взаимодействия при обращении к разделяемым данным, граф СНДА, представляющий одну ветвь  $i$  алгоритма такого взаимодействия, будет иметь вид:

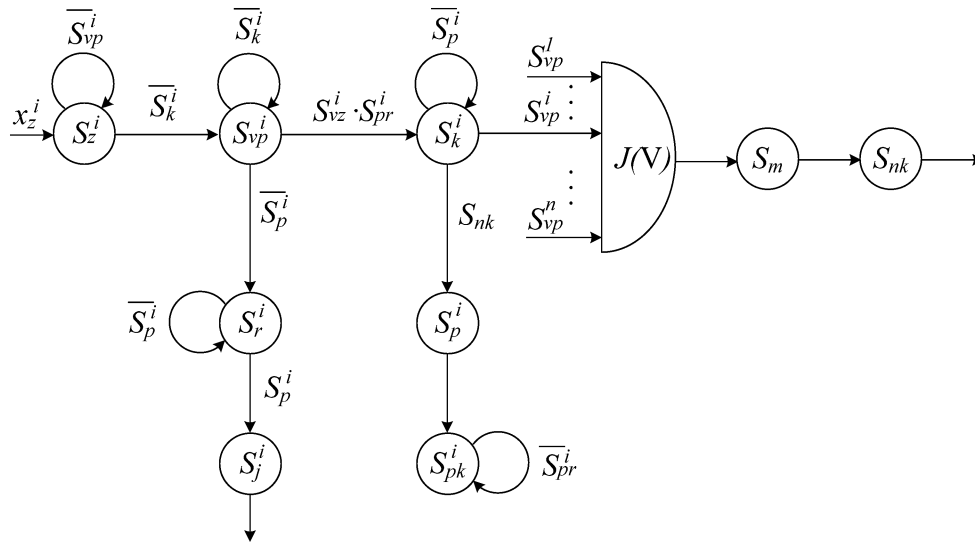


Рисунок 1 – Граф СНДА алгоритма управления ресурсом

Дополнительные события, представленные на графе, имеют следующий вид:

$$\begin{aligned}
 S_z^i(t+1) &= x_z^i \vee S_z^i \overline{S}_{vp}^i, & S_p^i(t+1) &= S_{nk} S_k^i, \\
 S_m(t+1) &= S_k^1 S_{vp}^1 \vee S_k^2 S_{vp}^2 \vee \dots \vee S_k^n S_{vp}^n, & S_{pk}^i(t+1) &= S_p^i \vee S_{pk}^i \overline{S}_{pr}^i, \\
 S_{nk}(t+1) &= S_m, & S_j^i(t+1) &= S_r^i S_p^i, \\
 S_r^i(t+1) &= (S_{vp}^1 \vee S_r^i) \overline{S}_p^i,
 \end{aligned}$$

где  $x_z^i$  – сигнал обращения к общему ресурсу  $i$ -го процесса;  $S_m$  и  $S_{nk}$  – события, обеспечивающие начало и окончание процедуры реализации обращения к разделяемым данным соответственно;  $S_r^i$  и  $S_j^i$  – события, символизирующие ожидание условий выхода  $i$ -го процесса из критического участка и инициирующие продолжение работы  $i$ -го процесса после выхода из критического интервала соответственно.

**Основные методы и подходы для структурной реализации алгоритмов управления, представляемых моделями СНДА.** Возможны несколько вариантов построения таких структур УА, которые базируются на использовании результатов детерминизации исходного УА, когда будет определён состав всех состояний детерминированного автомата, эквивалентного исходному СНДА. В результате детерминизации состояние ДА будет представлено совокупностью частных событий, одновременное существование которых в соответствии с исходным УА возможно. Рассмотрим два возможных варианта построения структур УА.

Первый вариант базируется на результатах детерминизации, при которых все частные события можно разбить на группы несовместимых событий. Тогда структуру в самом общем виде можно представить в виде графа, включающего последовательную и параллельные компоненты. При этом последовательную компоненту реализует главный подавтомат, а параллельные – рабочие подавтоматы. Рассмотренный подход возможен только в том случае, если частные события исходного УА не зависят по данным и ресурсам. Если существуют пары событий, зависимые по данным и ресурсам, которые размещаются в разных параллельных ветвях, то возможны тупиковые ситуации. Для их избегания необходимо выполнить (если это возможно) коррекцию исходного УА путем его эквивалентного преобразования таким образом, чтобы частные события, связанные по данным и ресурсам, ни при каких условиях не могли бы выполняться совместно.

Второй вариант базируется на использовании унитарного кодирования частных событий, когда такие события не могут быть разложены на группы несовместимых событий, что следует из результатов детерминизации исходного УА. Для этого подхода структурная схема системы управления будет соответствовать классической структурной схеме цифрового автомата с памятью, для которой функции возбуждения элементов памяти для любого  $i$ -го элемента будет полностью соответствовать описанию правой части для любого  $i$ -го частного события системы НД СКУ, представляющих исходный УА. Такой подход к построению структурной схемы УА характеризуется её значительной простотой и компактностью, так как функции переходов в УА представляются не в терминах состояний ДА, а в терминах частных событий, число которых значительно меньше числа состояний ДА.

**Верификация алгоритмов управления, представленных на языке СНДА.** Как было написано выше, после построения НД СКУ необходимо проверить ее на отсутствие в ней недостижимых событий и произвести верификацию управляющего автомата. Но процесс ручного составления прямой таблицы переходов для достаточно большой СКУ является длительной задачей, так же всегда есть вероятность допустить ошибки при занесении очередного события в таблицу. Аналогично и для перевода НД СКУ в описание на языках Verilog или VHDL. Поэтому для ускорения выполнения этих задач и снижения вероятности ошибки была разработана Система верификации алгоритмов управления, представленных на языке СНДА[16]. Система позволяет выполнять следующие действия:

- преобразовать исходную СКУ в модель, представленную в виде графа;
- проводить пошаговое моделирование с возможностью ручного задания входных сигналов;
- проводить автоматическое моделирование с возможностью задания входных сигналов в виде таблицы и генерацией отчета с пошаговым описанием состояния системы;
- генерация описания управляющего автомата на языках SMV и VHDL для верификации в других системах;
- генерация описания управляющего автомата на языках C++ и C# подходящего для использования в прикладных программах.

Рассмотрим работу системы на примере моделирования управляющего автомата, реализующего функцию обеспечения приоритетного взаимоисключающего доступа к критическому



ресурсу для четырех процессов. По представленным в [4] НД СКУ был сгенерирован граф (рисунок 2), на котором представлены все состояния управляющего автомата, входные и выходные сигналы, а так же связи между ними.

Далее была заполнена таблица входных сигналов (таблица 1), по которой будет производиться моделирование.

Таблица 1 – Таблица входных сигналов

Входной сигнал/Шаг моделирования	1	2	3	4	5	6	7	8	9	10	11
Xn	0	0	1	0	0	0	0	0	0	0	0
xz1	1	0	0	0	0	0	1	0	0	0	0
xz2	1	0	0	0	0	0	0	0	1	0	0
xz3	1	0	0	0	0	0	0	0	0	0	1
xz4	1	0	0	0	0	0	0	0	0	0	0
x0	1	0	0	0	0	0	0	0	0	0	0

При запуске моделирования значения из таблицы считываются и подставляются в модель, затем происходит пересчет модели и отображения текущего такта моделирования на графе. Когда моделирование доходит до конца таблицы моделирование продолжается с игнорированием ее значений, это позволяет не заполнять таблицу полностью.

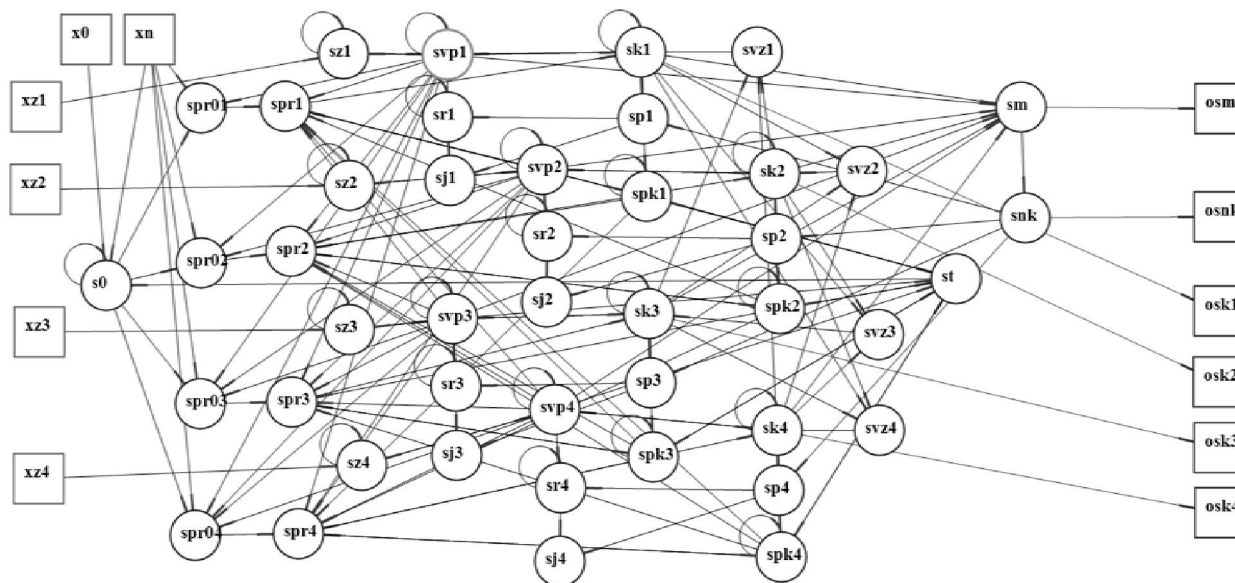


Рисунок 2 – Граф СНДА алгоритма управления 4-мя ресурсами в программе верификации

В результате моделирования получается отчет, в котором в табличной форме представлена информация о состоянии модели на каждом шаге моделирования. В таблице 2 представлен пример такой таблицы, по ней можно сделать вывод, что доступ к критическому ресурсу происходит в соответствии с требованиями, предъявленными к управляющему автомату (события  $S_k^i - S_k^d$  происходят в соответствии с заданным приоритетом и последовательно, о чем свидетельствуют соответствующие выходные сигналы  $O_{sk}^i - O_{sk}^d$ ).

Таблица 2 – Таблица с результатами моделирования

Шаг моделирования	xn	xz1	xz2	xz3	xz4	s0	x0	osm	Osnk	osk1	osk2	osk3	osk4
Шаг 0	0	1	1	1	1	0	1	0	0	0	0	0	0
Шаг 1	0	0	0	0	0	1	0	0	0	0	0	0	0
Шаг 2	0	0	0	0	0	1	0	0	0	0	0	0	0
Шаг 3	1	0	0	0	0	1	0	0	0	0	0	0	0
Шаг 4	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 5	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 6	0	0	0	0	0	0	0	0	0	1	0	0	0
Шаг 7	0	0	0	0	0	0	0	1	0	1	0	0	0
Шаг 8	0	0	0	0	0	0	0	0	1	1	0	0	0
Шаг 9	0	0	0	0	0	0	0	0	0	1	0	0	0
Шаг 10	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 11	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 12	0	0	0	0	0	0	0	0	0	0	1	0	0
Шаг 13	0	0	0	0	0	0	0	1	0	0	1	0	0
Шаг 14	0	0	0	0	0	0	0	0	1	0	1	0	0
Шаг 15	0	0	0	0	0	0	0	0	0	0	1	0	0
Шаг 16	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 17	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 18	0	0	0	0	0	0	0	0	0	0	0	1	0
Шаг 19	0	0	0	0	0	0	0	1	0	0	0	1	0
Шаг 20	0	0	0	0	0	0	0	0	1	0	0	1	0
Шаг 21	0	0	0	0	0	0	0	0	0	0	0	1	0
Шаг 22	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 23	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 24	0	0	0	0	0	0	0	0	0	0	0	0	1
Шаг 25	0	0	0	0	0	0	0	1	0	0	0	0	1
Шаг 26	0	0	0	0	0	0	0	0	1	0	0	0	1
Шаг 27	0	0	0	0	0	0	0	0	0	0	0	0	1
Шаг 28	0	0	0	0	0	0	0	0	0	0	0	0	0
Шаг 29	0	0	0	0	0	1	0	0	0	0	0	0	0

Таким образом, доказывается правильность составленного описания управляющего автомата и выполнения предъявленных к нему требований.

#### ЛИТЕРАТУРА

- [1] Hoare C. A. R. Communicating Sequential Processes. Prentice Hall International Series in Computer Science, 1985, 256 p.
- [2] Clarke E. M., Emerson E. A., Sifakis J. Model checking: algorithmic verification and debugging // Commun. ACM 52(11), 2009, pp. 74-84.
- [3] Вашкевич Н.П. Недетерминированные автоматы в проектировании систем параллельной обработки. – Пенза: изд-во Пенз. гос. ун-та. – 2004. – 280 с.
- [4] Вашкевич Н.П., Бикташев Р.А., Гурин Е.И. Аппаратная реализация функций синхронизации параллельных процессов при обращении к разделяемому ресурсу на основе ПЛИС // Известия вузов. Поволжский регион. Технические науки. – № 2. – 2007. – С.3-12.
- [5] Heymann M., Lin F. Discrete-Event Control of Nondeterministic Systems // IEEE Transaction on Automatic Control, 1998, vol. 43, no. 1, pp. 3-17.
- [6] Arnold A., Walukiewicz I. Nondeterministic controllers of nondeterministic processes // Logic and Automata: History and Perspectives. In Honor of Wolfgang Thomas. Series "Texts in Logic and Games", Amsterdam University Press, 2008, pp. 29-52.

[7] Wolff E. M., Topcu U., and Murray R. M. Optimal Control of Non-deterministic Systems for a Computationally Efficient Fragment of Temporal Logic // Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy, pp. 3197-3204.

[8] Baral C. Zhao J. Goal Specification, Non-determinism and Quantifying over Policies // In Proceedings of ECAI'04, pp. 273-277.

[9] Nakahara H., Sasao T., Matsuura M. A Regular Expression Matching Circuit Based on a Modular Non-deterministic Finite Automaton with Multi-Character Transition // SASIMI 2010 Proceedings (R4-1), pp. 359-364.

[10] Su R., Jan H., van Schuppen J. H., Rooda J. E. Maximally Permissive Distributed Supervisory Control of Non-deterministic Discrete-Event Systems // 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) Orlando, FL, USA, December 12-15, 2011, pp. 1155-1160.

[11] Jiang S., Kumar R. Supervisory Control of Nondeterministic Discrete Event Systems with Driven Events via Masked Prioritized Synchronization // IEEE Transactions on Automatic Control, 2002, vol. 47, no. 9, pp. 1438-1449.

[12] Avasse M., Rizzo F., Sisto R. Efficient Multistriding of Large Non-deterministic Finite State Automata for Deep Packet Inspection // IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 10-15, 2012, – pp. 1079-1084.

[13] Su R., van Schuppen J.H., Rooda J. E. Model Abstraction of Nondeterministic Finite State Automata in Supervisor Synthesis // SE Report: Nr. 2008-03Eindhoven, June 2008, 32 p. SE Reports are available via <http://se.wtb.tue.nl/sereports>.

[14] Tanenbaum A. S., Bos H. Modern Operating Systems (4th Edition), Prentice-Hall, 2014, 1136 p.

[15] Andrews G. R. Foundations of Multithreaded, Parallel, and Distributed Programming. Addison-Wesley, 1999, 664 p.

[16] В.В. Кутузов Реализация и сравнение производительности высокоскоростных способов передачи сообщений в многопроцессорных системах // В.В. Кутузов - Труды международного симпозиума «Надежность и качество», Издательство ПГУ, Пенза, 2013. – стр. 292-295

## REFERENCES

[1] Hoare C. A. R. Communicating Sequential Processes. Prentice Hall International Series in Computer Science, 1985, 256 (in Eng.).

[2] Clarke E. M., Emerson E. A., Sifakis J. Model checking: algorithmic verification and debugging // Commun. ACM 52(11), 2009, 74-84 (in Eng.).

[3] Vashkevich N.P. Nedeterminirovannyye avtomaty v proektirovanii sistem paralel'noj obrabotki. – Penza: izd-vo Penz. gos. un-ta. – 2004, 280 (in Rus.).

[4] Vashkevich N.P., Biktashev R.A., Gurin E.I.. Apparatnaya realizatsiya funktsij sinhronizatsii paralel'nykh processov pri obrashchenii k razdeljaemomu resursu na osnove PLIS // Izvestiya vuzov. Povolzhskij region. Tehnicheskie nauki. – № 2, 2007, 3-12 (in Rus.).

[5] Heymann M., Lin F. Discrete-Event Control of Nondeterministic Systems // IEEE Transaction on Automatic Control, 1998, vol. 43, no. 1, 3-17 (in Eng.).

[6] Arnold A., Walukiewicz I. Nondeterministic controllers of nondeterministic processes // Logic and Automata: History and Perspectives. In Honor of Wolfgang Thomas. Series “Texts in Logic and Games”, Amsterdam University Press, 2008, 29-52 (in Eng.).

[7] Wolff E. M., Topcu U., and Murray R. M. Optimal Control of Non-deterministic Systems for a Computationally Efficient Fragment of Temporal Logic // Proceedings of the 52nd IEEE Conference on Decision and Control, CDC 2013, December 10-13, 2013, Firenze, Italy, 3197-3204 (in Eng.).

[8] Baral C. Zhao J. Goal Specification, Non-determinism and Quantifying over Policies // In Proceedings of ECAI'04, 273-277. (in Eng.).

[9] Nakahara H., Sasao T., Matsuura M. A Regular Expression Matching Circuit Based on a Modular Non-deterministic Finite Automaton with Multi-Character Transition // SASIMI 2010 Proceedings (R4-1), 359-364. (in Eng.).

[10] Su R., Jan H., van Schuppen J. H., Rooda J. E. Maximally Permissive Distributed Supervisory Control of Non-deterministic Discrete-Event Systems // 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC) Orlando, FL, USA, December 12-15, 2011, pp. 1155-1160. (in Eng.).

[11] Jiang S., Kumar R. Supervisory Control of Nondeterministic Discrete Event Systems with Driven Events via Masked Prioritized Synchronization // IEEE Transactions on Automatic Control, 2002, vol. 47, no. 9, pp. 1438-1449. (in Eng.).

[12] Avasse M., Rizzo F., Sisto R. Efficient Multistriding of Large Non-deterministic Finite State Automata for Deep Packet Inspection // IEEE International Conference on Communications (ICC 2012), Ottawa, Canada, June 10-15, 2012, – pp. 1079-1084. (in Eng.).

[13] Su R., van Schuppen J.H., Rooda J. E. Model Abstraction of Nondeterministic Finite State Automata in Supervisor Synthesis // SE Report: Nr. 2008-03Eindhoven, June 2008, 32 p. SE Reports are available via <http://se.wtb.tue.nl/sereports>. (in Eng.).

[14] Tanenbaum A. S., Bos H. Modern Operating Systems (4th Edition), Prentice-Hall, 2014, 1136 p. (in Eng.).

[15] Andrews G. R. Foundations of Multithreaded, Parallel, and Distributed Programming. Addison-Wesley, 1999, 664 p. (in Eng.).

[16] V.V. Kutuzov Realizatsiya i sravnenie proizvoditel'nosti vysokoskorostnykh sposobov peredachi soobshhenij v mnogoprocessornykh sistemah // V.V. Kutuzov - Trudy mezhdunarodnogo simpoziuma «Nadezhnost' i kachestvo», Izdatel'stvo PGU, Penza, 2013. – str. 292-295. (in Rus.).

**ЛОГИКАЛЫҚ БАСҚАРУДЫҢ ПАРАЛЛЕЛЬДІ АЛГОРИТМДЕРІН  
ФОРМАЛЬДЫ СИПАТТАУ ҮШІН УАҚИҒАЛЫҚ ДЕТЕРМИНИРЛІ ЕМЕС  
АВТОМАТТАРДЫҢ МОДЕЛЬДЕРІН ҚОЛДАНУ**

**Н. П. Вашкевич<sup>1</sup>, Р. А. Бикташев<sup>1</sup>, Д. В. Пашенко<sup>1</sup>, В. В. Кутузов<sup>1</sup>, К. Т. Сауанова<sup>2</sup>**

<sup>1</sup>Пенза мемлекеттік университеті, Ресей,

<sup>2</sup>Алматы Энергетика және байланыс университеті, Қазақстан

**Тірек сөздер:** параллельді алгоритмдер, логикалық басқару жүйелері, басқару алгоритмдері уақиғалық детерминерлі емес автоматтар.

**Аннотация:** Алгоритмде жүзеге асырылған барлық дербес оқиғаларды сипаттайтын, канонды рекурренттік теңдеулер стандартты жүйелері түрінде көрсетілген, уақиғалық детерминерлі емес автоматтар (УДЕА) модельдері негізінде логикалық басқару параллельді алгоритмдерді аналитикалық сипаттау үшін математикалық құралдар қарастырылған.

УДЕА тілінің тиімділігі мен айқын мүмкіндіктері жеке оқиғалар арасындағы басқару мен мәліметтер бойынша байланыстарын, басқару жүйелерінің уақыт ағымындағы іс-әрекеттерін есепке алатын ақпаратпен көрсетілген.

УДЕА тілінің өзге алгоритмдерді сипаттайтын тілдермен байланысы көрсетілген және басқару алгоритмдерді құрылымдық іске асыру үшін, аппаратураларды сипаттайтын тілдерге өзгертіп келтіру мүмкіндіктері қарастырылған.

УДЕА тілін пайдалану тиімділігі ортақ кризистік қорға кризистік секциялар механизмнің пайдалану жолымен қол жеткізу өзара әрекеттес параллельді үрдістерді уақыт жағынан дәл, сәйкес келтіруді басқаратын алгоритмін формализациялау мысалдарында көрсетілген.

*Поступила 22.05.2015 г.*

**BULLETIN OF NATIONAL ACADEMY OF SCIENCES  
OF THE REPUBLIC OF KAZAKHSTAN**

ISSN 1991-3494

Volume 4, Number 356 (2015), 59 – 64

**THE PROJECT MANAGEMENT:  
PROBLEMS AND PERSPECTIVES OF DEVELOPMENT IN RK**

**A. S. Esengeldina**

Academy of Public Administration under the President of the Republic of Kazakhstan, Astana, Kazakhstan.

E-mail: Yanar77@inbox.ru

**Key words:** project, management, standard, manager, qualification.

**Abstract.** The purpose of work is studying of system of design management, identification of problems and development of the perspective directions of its development in the Republic of Kazakhstan. The methodology of work was made by methods of empirical research, supervision, a comparative method. Results of work was the conclusion that for improvement of system of design management vigorous actions on the organization of large-scale training of design managers in the country as through the leading higher education institutions, and specially created training centers of the working design managers are required. The author proves need of development and adoption of national standards on management of projects; developments of the academic and professional education in the field of design management; retrainings and professional development of heads of all levels for development of methodology and means of design management; creations of system of motivation and stimulation of use of design management, etc. And thus all this complex of works will demand the state support, and also efforts of a business community, the public and means the mass of information.

Scope of the results received by the author of article is improvement of system of design management. Conclusions of the author can be used in the course of teaching on economics departments of disciplines of a business cycle.