

УДК 004.75

ФОРМАЛЬНЫЕ АВТОМАТНЫЕ МОДЕЛИ АЛГОРИТМОВ ОБРАБОТКИ КЭШИРУЕМЫХ ДАННЫХ

¹Вашкевич Н.П., ²Сибиряков М.А.

¹Пензенский государственный университет, Пенза, e-mail: vt@alice.pnzgu.ru;

²Поволжский государственный технологический университет, Йошкар-Ола, e-mail: maxover777@bk.ru

Правильная организация алгоритмического управления в любой вычислительной системе значительно влияет на ее производительность, эффективность управления процессами и ресурсами. При разработке таких систем актуальна задача формализации алгоритмов обработки и управления. Особое внимание в данной статье уделяется математическому аппарату, связанному с формальным представлением алгоритмов в виде модели конечных автоматов. В качестве основной модели конечного автомата принята система рекуррентных канонических уравнений (СКУ), описывающих все реализуемые в системе частные события, принят язык граф-схем алгоритмов. Основной целью статьи является формализация рассматриваемых алгоритмов обработки кэшируемых данных в системах хранения данных с помощью СКУ. В статье представлены граф-схемы алгоритмов, на основе которых составлены системы канонических уравнений. Полученное математическое представление является, по существу, исполнимой формализованной спецификацией, которая позволяет осуществить непосредственный переход от СКУ к программной или аппаратной реализации данных алгоритмов.

Ключевые слова: системы хранения данных, обработка и хранение данных, кэширование, формализация алгоритмов, конечные автоматы, система канонических уравнений

THE FORMAL AUTOMATIC MODELS OF ALGORITHMS OF PROCESSING OF THE CACHED DATA

¹Vashkevich N.P., ²Sibirjakov M.A.

¹Penza State University, Penza, e-mail: vt@alice.pnzgu.ru;

²Volga State University of Technology, Yoshkar-Ola, e-mail: maxover777@bk.ru

The correct organization of algorithmic control in any computing system considerably influences its productivity, effective management of processes and resources. By development of such systems the task of formalization of algorithms of processing and control is actual. Special attention in this article is paid to the mathematical apparatus connected to the formal representation of algorithms in the form of model of finite state machines. As the main model of the finite state machine the system of the recurrent canonical equations (SKU) describing all private events realized in system and language of graphs diagrams of algorithms is accepted. Main objective of article is formalization of the considered algorithms of processing of the cached data in storage systems of data with the help of SKU. In article are provided the graph diagram of algorithms on the basis of which systems of the canonical equations are made. The gained mathematical impression is, in essence, the executable formalized specification which allows to realize direct transition from SKU to program or hardware implementation of these algorithms.

Keywords: storage systems, processing and data storage, caching, formalization of algorithms, finite state machines, system of the canonical equations

Для обеспечения обработки и хранения большого объема информации в современных сетевых инфраструктурах применяются системы хранения данных (СХД). Одним из способов повышения производительности СХД является использование кэширования. Производительность применяемой в таких системах кэш-памяти во многом определяется эффективностью реализуемых алгоритмов обработки кэшируемых данных. При разработке таких алгоритмов актуальной задачей является их формальное представление. Одним из способов формального описания алгоритмов является применение математического аппарата, связанного с их представлением на основе модели конечных автоматов с использованием рекуррентных бескванторных предикатных уравнений или систем канонических уравнений, предложенных Н.П. Вашкевичем для последовательных [1, 2] и параллельных [3, 4] алгоритмов.

Системы канонических уравнений для алгоритмов обработки кэшируемых данных

Формализуем граф-схемы алгоритмов обработки кэшируемых данных (обработки команды чтения и вытеснения данных в кэш-памяти) с помощью систем канонических уравнений. СКУ позволяет формировать компактные представления алгоритмов с описанием функций переходов в терминах частных событий. Данные события реализуются в частичных детерминированных автоматах Мура, в которых соблюдаются условия однозначности переходов, то есть под воздействием одного частичного входного сигнала или комбинации таких сигналов возможен переход от некоторого исходного события только к одному событию.

На рис. 1 и 2 представлены блок-схемы алгоритмов обработки кэшируемых данных,

разработанных в рамках модификации метода обработки кэшируемых данных с использованием индексных таблиц [5]. Для них разработаны укрупненные граф-схемы алгоритмов (ГСА, рис. 3 и 4), на основе которых построены SKU. Результаты укрупнения представлены в табл. 1 и 2. В данных ГСА следующие подряд условные вершины разделены опера-

торными вершинами – *разделительными событиями*, которые подразумевают ожидание. Они введены для упрощения и корректной реализации SKU. Операторы ГСА требуют различных времен выполнения, поэтому каждое уравнение SKU построено не только при учете условий зарождения, но и при учете условия сохранения события.

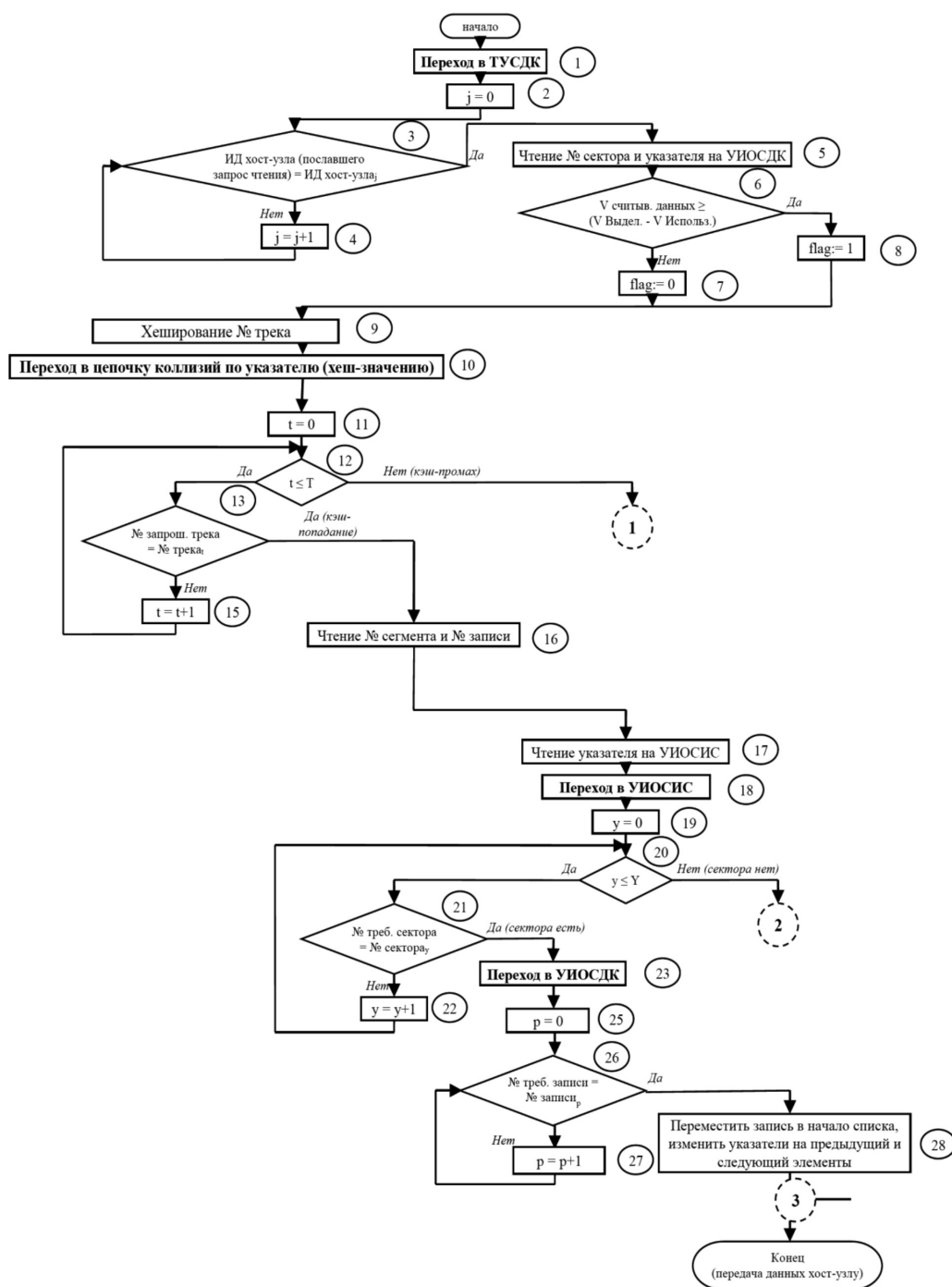
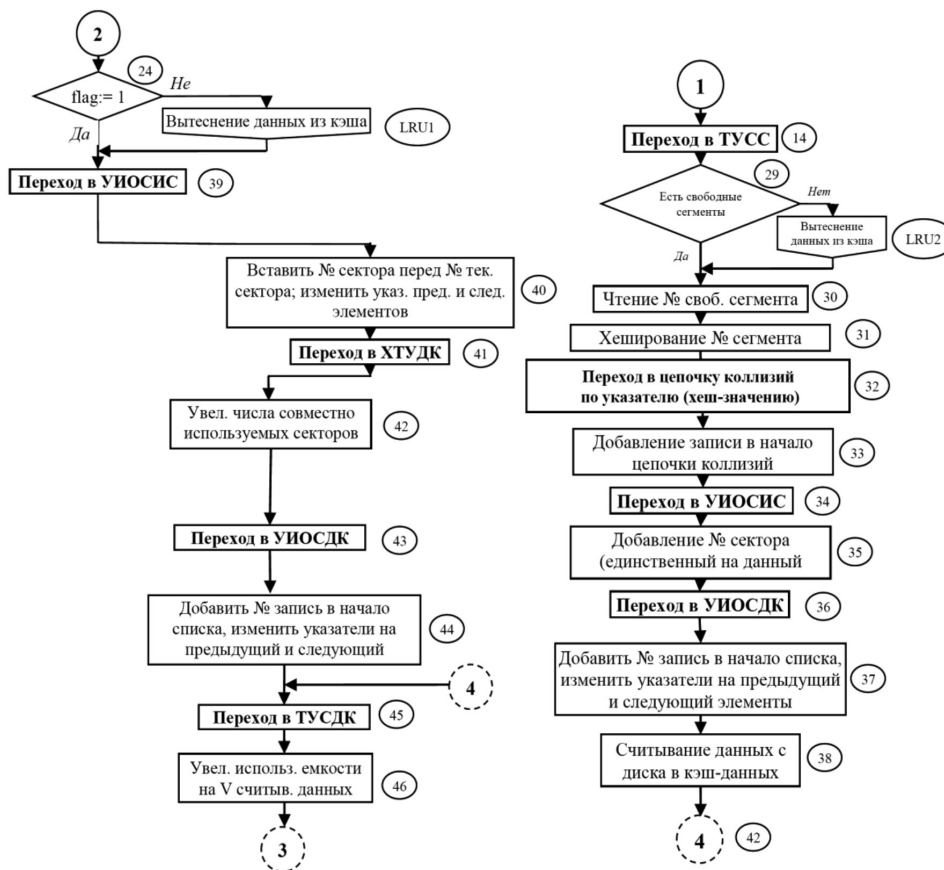


Рис. 1. Блок-схема модифицированного алгоритма обработки команды чтения данных



Продолжение рис. 1

Таблица 1

Вершины модифицированного алгоритма вытеснения данных, составляющие операторы укрупненных ГСА

Вершина укрупненной ГСА, S_i	Номера составляющих вершин
S_0	Начало
S_1	51–55
X_1	56
S_2	57
X_2	58, 59
X_3	60
S_6	61, 62
S_5	63
S_7	64–67
S_8	68–70
X_4	71
S_9	72
S_{10}	73–75
S_{11}	76, 77
S_k	Конец

Примечание. События S_3, S_4 – разделительные.

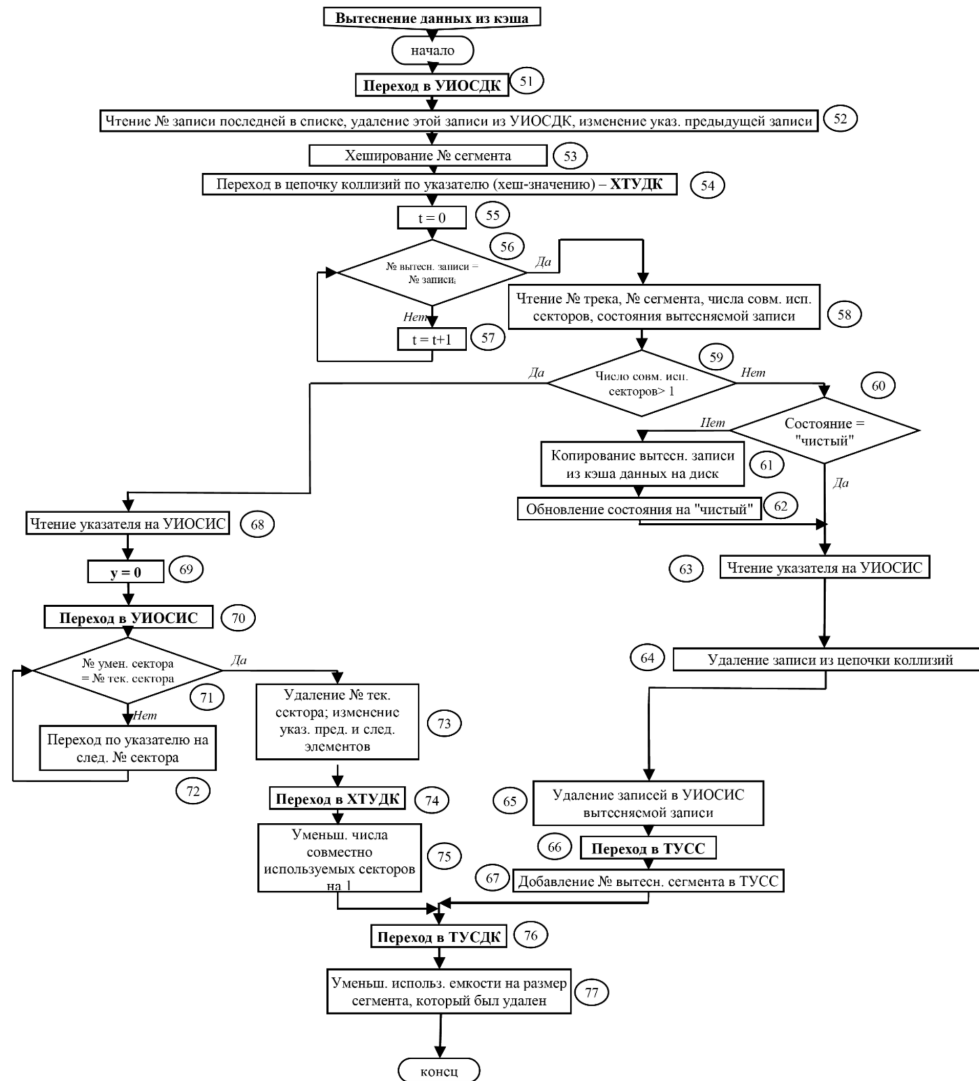


Рис. 2. Блок-схема модифицированного алгоритма вытеснения чтения

СКУ₁ для ГСА₁:

$$\begin{aligned}
 S_0(t+1) &= S_0(t) \& \neg x_0(t) \vee x_{begin}(t); \\
 S_1(t+1) &= S_0(t) \& x_0(t) \vee S_1(t) \& \neg z_1(t); \\
 S_2(t+1) &= S_1(t) \& z_1(t) \& x_1(t) \vee S_2(t) \& z_2(t) \& x_2(t) \vee S_2(t) \& \neg z_2(t); \\
 S_3(t+1) &= S_1(t) \& z_1(t) \& \neg x_1(t) \vee S_2(t) \& z_2(t) \& \neg x_1(t) \vee S_3(t) \& \neg z_3(t); \\
 S_4(t+1) &= S_3(t) \& z_3(t) \& x_2(t) \vee S_4(t) \& \neg z_4(t); \\
 S_5(t+1) &= S_4(t) \& z_4(t) \& x_3(t) \vee S_5(t) \& \neg z_5(t); \\
 S_6(t+1) &= S_4(t) \& z_4(t) \& \neg x_3(t) \vee S_6(t) \& \neg z_6(t); \\
 S_7(t+1) &= S_5(t) \& z_5(t) \vee S_6(t) \& z_6(t) \vee S_7(t) \& \neg z_7(t); \\
 S_8(t+1) &= S_5(t) \& z_5(t) \& \neg x_2(t) \vee S_8(t) \& \neg z_8(t); \\
 S_9(t+1) &= S_8(t) \& z_8(t) \& x_4(t) \vee S_9(t) \& z_9(t) \& x_4(t) \vee S_9(t) \& \neg z_9(t); \\
 S_{10}(t+1) &= S_8(t) \& z_8(t) \& \neg x_4(t) \vee S_9(t) \& z_9(t) \& \neg x_4(t) \vee S_{10}(t) \& \neg z_{10}(t); \\
 S_{11}(t+1) &= S_{10}(t) \& z_{10}(t) \vee S_7(t) \& z_7(t) \vee S_{11}(t) \& \neg z_{11}(t); \\
 S_k(t+1) &= S_{11}(t) \& z_{11}(t) \vee S_k(t) \& \neg z_k(t),
 \end{aligned}$$

где $S_i(t)$ – унарный предикат, определенный на множестве значений дискретного времени t . Истинность $S_i(t)$ означает, что

автомат находится в состоянии S_i в момент времени t , или в автомате реализуется частное событие S_i в момент времени t .

Входные сигналы $x_j(t)$ – частичные, т.е. на переход влияет сам сигнал, а не комбинация всех сигналов, как в случае полного автомата; $x_{begin}(t)$ – сигнал установки автомата в начальное состояние. Входные сигналы $x_j(t)$ представлены унарными предикатами.

Исходя из того, что времена работы операторов различные, необходимо учесть условия окончания работы операторов, то есть каждый оператор по окончании своей работы должен выдать сигнал окончания. Тогда примем, что $z_i(t)$ – условие сохранения события S_i при $z_i(t) = 0$ («false»); при $z_i(t) = 1$ («true») событие S_i не сохраняется и происходит переход к следующему событию. Сигнал, или условие $z_i(t)$, формиру-

ется при выполнении события S_i (условие $z_i(t)$ может быть истинным только после выполнения события S_i). Использование условия сохранения позволяет событию S_i выполняться за необходимое количество тактов.

Используемый в работе алгоритмов индекс включает в себя шесть базовых индексных таблиц [4]: *ХТУДК* – хеш-таблица управления; *ТУСС* – таблица управления свободными сегментами; *УИОСДК* – таблица, хранящая управляющую информацию о секторах дискового кэша; *УИОСИС* – таблица, содержащая управляющую информацию о совместно используемых сегментах; *ТУСДК* – таблица управления секторами дискового кэша.

Таблица 2

Вершины модифицированного алгоритма обработки команды чтения данных, составляющие операторы укрупненных ГСА

Вершина укрупненной ГСА, S_i	Номера составляющих вершин
1	2
S_0	Начало
S_1	1, 2
X_1	3
S_2	4
S_3	5–11
X_2	12
X_5	13
S_6	15
S_7	16–18
X_6	20
X_4	24
S_{17}	<i>LRU2</i>
S_{18}	39–44
1	2
S_{19}	45, 46
X_7	21
S_{10}	22
S_{11}	23, 25
X_8	26
S_{12}	27
S_{13}	28
X_3	14, 29
S_{14}	<i>LRU1</i>
S_{15}	30
S_{16}	31–38
S_k	Конец

Примечание. События S_4, S_5, S_8, S_9 – разделительные.

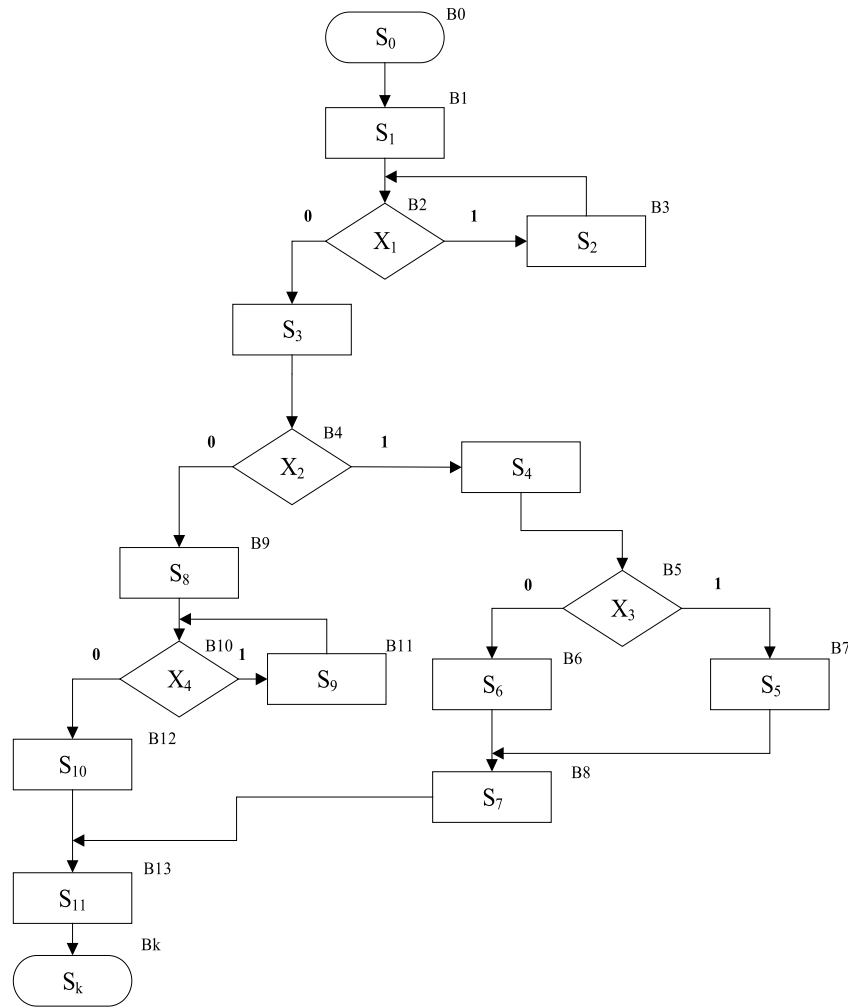


Рис. 3. Граф-схема модифицированного алгоритма вытеснения данных (ГСА₂)

СКУ₂ для ГСА₂:

$$\begin{aligned}
 S_0(t+1) &= S_0(t) \& \neg x_0(t) \vee x_{begin}(t); \\
 S_1(t+1) &= S_0(t) \& x_0(t) \vee S_1(t) \& \neg z_1(t); \\
 S_2(t+1) &= S_1(t) \& z_1(t) \& x_1(t) \vee S_2(t) \& z_7(t) \& x_1(t) \vee S_2(t) \& \neg z_2(t); \\
 S_3(t+1) &= S_1(t) \& z_1(t) \& \neg x_1(t) \vee S_2(t) \& z_7(t) \& \neg x_1(t) \vee S_3(t) \& \neg z_3(t); \\
 S_4(t+1) &= S_3(t) \& z_3(t) \& x_2(t) \vee S_6(t) \& z_6(t) \& x_2(t) \vee S_4(t) \& \neg z_4(t); \\
 S_5(t+1) &= S_3(t) \& z_3(t) \& \neg x_2(t) \vee S_6(t) \& z_6(t) \& \neg x_2(t) \vee S_5(t) \& \neg z_5(t); \\
 S_6(t+1) &= S_5(t) \& z_5(t) \& x_5(t) \vee S_6(t) \& \neg z_6(t); \\
 S_7(t+1) &= S_5(t) \& z_5(t) \& \neg x_5(t) \vee S_7(t) \& \neg z_7(t); \\
 S_8(t+1) &= S_7(t) \& z_7(t) \& x_6(t) \vee S_{10}(t) \& z_{10}(t) \& x_6(t) \vee S_8(t) \& \neg z_8(t); \\
 S_9(t+1) &= S_7(t) \& z_7(t) \& \neg x_6(t) \vee S_{10}(t) \& z_{10}(t) \& \neg x_6(t) \vee S_9(t) \& \neg z_9(t); \\
 S_{10}(t+1) &= S_9(t) \& z_9(t) \& x_7(t) \vee S_{10}(t) \& \neg z_{10}(t); \\
 S_{11}(t+1) &= S_9(t) \& z_9(t) \& \neg x_7(t) \vee S_{11}(t) \& \neg z_{11}(t); \\
 S_{12}(t+1) &= S_{11}(t) \& z_{11}(t) \& x_8(t) \vee S_{12}(t) \& z_{12}(t) \& x_8(t) \vee S_{12}(t) \& \neg z_{12}(t); \\
 S_{13}(t+1) &= S_{11}(t) \& z_{11}(t) \& \neg x_8(t) \vee S_{12}(t) \& z_{12}(t) \& \neg x_8(t) \vee S_{13}(t) \& \neg z_{13}(t); \\
 S_{14}(t+1) &= S_{12}(t) \& z_{12}(t) \& x_3(t) \vee S_{14}(t) \& \neg z_{14}(t); \\
 S_{15}(t+1) &= S_{12}(t) \& z_{12}(t) \& \neg x_3(t) \vee S_{14}(t) \& z_{14}(t) \vee S_{15}(t) \& \neg z_{15}(t); \\
 S_{16}(t+1) &= S_{15}(t) \& z_{15}(t) \vee S_{16}(t) \& \neg z_{16}(t); \\
 S_{17}(t+1) &= S_{15}(t) \& z_{15}(t) \& x_4(t) \vee S_{17}(t) \& \neg z_{17}(t); \\
 S_{18}(t+1) &= S_{17}(t) \& z_{17}(t) \vee S_{18}(t) \& z_8(t) \& \neg x_4(t) \vee S_{18}(t) \& \neg z_{18}(t); \\
 S_{19}(t+1) &= S_{16}(t) \& z_{16}(t) \vee S_{18}(t) \& z_{18}(t) \vee S_{19}(t) \& \neg z_{19}(t); \\
 S_k(t+1) &= S_{13}(t) \& z_{13}(t) \vee S_{19}(t) \& z_{19}(t) \vee S_k(t) \& \neg z_k(t).
 \end{aligned}$$

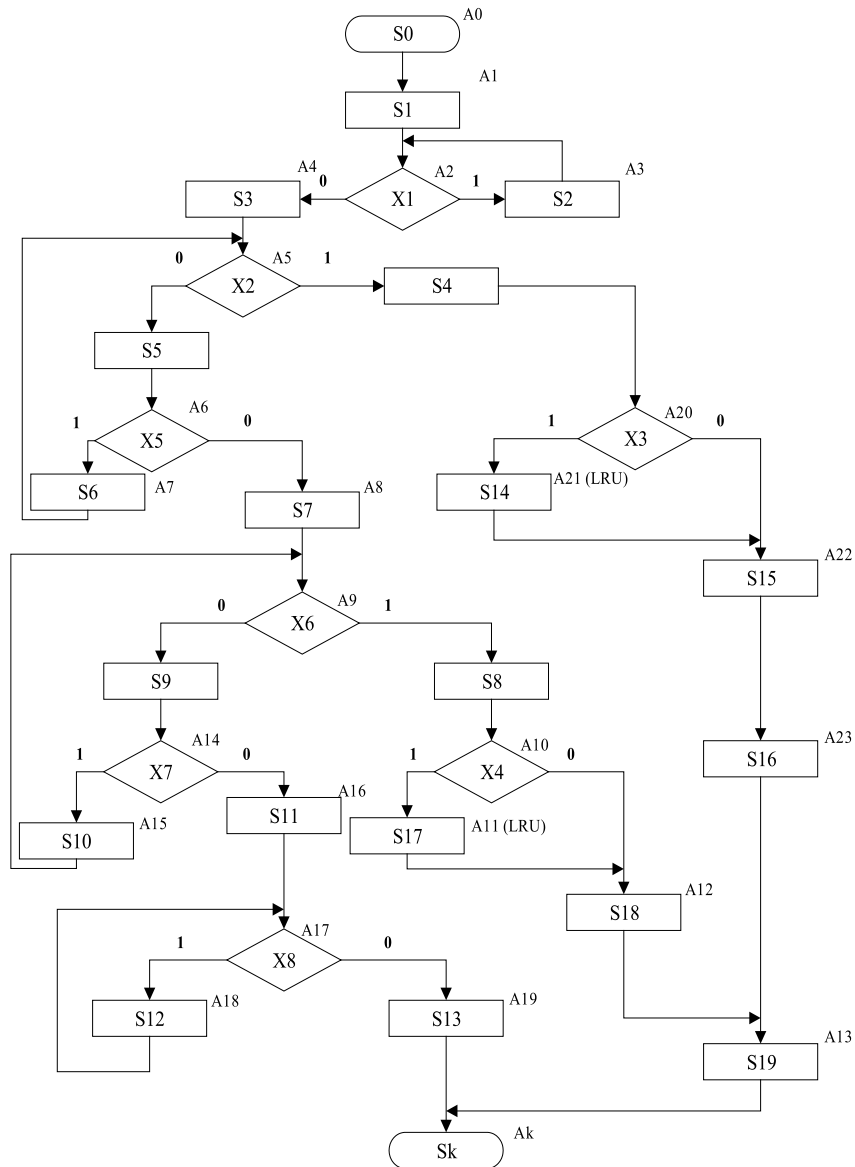


Рис. 4. Граф-схема алгоритма обработки команды чтения данных (ГСА₃)

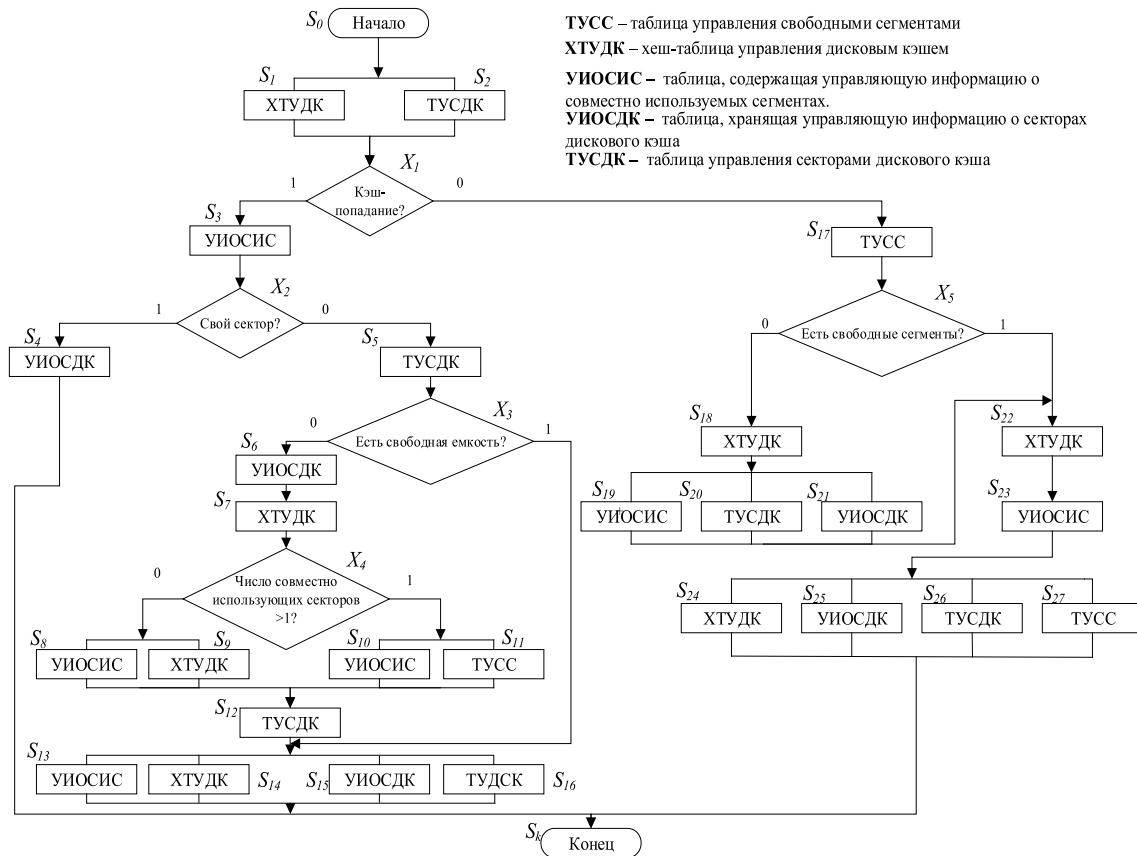
Система канонических уравнений для параллельного алгоритма обращения к индексным таблицам

Аналогичным образом представлена граф-схема разработанного параллельного алгоритма обращения к индексным таблицам (рис. 5), применяемого в рамках выполнения алгоритмов обработки кэшируемых данных. Приведена ее формализация с помощью СКУ.

СКУ₃ для ГСА₃:

$$\begin{aligned}
 S_0(t+1) &= S_0(t) \& \neg x_0(t) \vee x_{begin}(t); \\
 S_1(t+1) &= S_0(t) \& x_0(t) \vee S_1(t) \& \neg z_1(t); \\
 S_2(t+1) &= S_0(t) \& x_0(t) \vee S_2(t) \& \neg z_2(t); \\
 S'_1(t+1) &= S_1(t) \& z_1(t) \vee S'_1(t) \& \neg (S_3(t) \vee S_{17}(t));
 \end{aligned}$$

$$\begin{aligned}
 S'_2(t+1) &= S_2(t) \& z_2(t) \vee S'_2(t) \& \neg (S_3(t) \vee S_{17}(t)); \\
 S_3(t+1) &= S'_1(t) \& S'_2(t) \& x_1(t) \vee S_3(t) \& \neg z_3(t); \\
 S_4(t+1) &= S_3(t) \& z_3(t) \& x_2(t) \vee S_4(t) \& \neg z_4(t); \\
 S_5(t+1) &= S_3(t) \& z_3(t) \& \neg x_2(t) \vee S_5(t) \& \neg z_5(t); \\
 S_6(t+1) &= S_5(t) \& z_5(t) \& \neg x_3(t) \vee S_6(t) \& \neg z_6(t); \\
 S_7(t+1) &= S_6(t) \& z_6(t) \vee S_7(t) \& \neg z_7(t); \\
 S_8(t+1) &= S_7(t) \& z_7(t) \& \neg x_4(t) \vee S_8(t) \& \neg z_8(t); \\
 S_9(t+1) &= S_7(t) \& z_7(t) \& \neg x_4(t) \vee S_9(t) \& \neg z_9(t); \\
 S_{10}(t+1) &= S_7(t) \& z_7(t) \& x_4(t) \vee S_{10}(t) \& \neg z_{10}(t); \\
 S_{11}(t+1) &= S_7(t) \& z_7(t) \& x_4(t) \vee S_{11}(t) \& \neg z_{11}(t); \\
 S'_8(t+1) &= S_8(t) \& z_8(t) \vee S'_8(t) \& \neg S_{12}(t); \\
 S'_9(t+1) &= S_9(t) \& z_9(t) \vee S'_9(t) \& \neg S_{12}(t); \\
 S'_{10}(t+1) &= S_{10}(t) \& z_{10}(t) \vee S'_{10}(t) \& \neg S_{12}(t); \\
 S'_{11}(t+1) &= S_{11}(t) \& z_{11}(t) \vee S'_{11}(t) \& \neg S_{12}(t);
 \end{aligned}$$



ТУСС – таблица управления свободными сегментами
 ХТУДК – хеш-таблица управления дисковым кэшем
 УИОСИС – таблица, содержащая управляющую информацию о совместно используемых сегментах.
 УИОСДК – таблица, хранящая управляющую информацию о секторах дискового кэша
 ТУСДК – таблица управления секторами дискового кэша

Рис. 5. Граф-схема параллельного алгоритма обращения к индексным таблицам (ГСА₃)

$$\begin{aligned}
 S_{12}(t+1) &= S'_8(t) \& S'_9(t) \& S'_{10}(t) \& S'_{11}(t) \vee S_{12}(t) \& \neg z_{12}(t); \\
 S_{13}(t+1) &= S_{12}(t) \& z_{12}(t) \vee S_5(t) \& x_3 \vee S_{13}(t) \& \neg z_{13}(t); \\
 S_{14}(t+1) &= S_{12}(t) \& z_{12}(t) \vee S_5(t) \& x_3 \vee S_{14}(t) \& \neg z_{14}(t); \\
 S_{15}(t+1) &= S_{12}(t) \& z_{12}(t) \vee S_5(t) \& x_3 \vee S_{15}(t) \& \neg z_{15}(t); \\
 S_{16}(t+1) &= S_{12}(t) \& z_{12}(t) \vee S_5(t) \& x_3 \vee S_{16}(t) \& \neg z_{16}(t); \\
 S'_{13}(t+1) &= S_{13}(t) \& z_{13}(t) \vee S'_{13}(t) \& \neg S_k(t); \\
 S'_{14}(t+1) &= S_{14}(t) \& z_{14}(t) \vee S'_{14}(t) \& \neg S_k(t); \\
 S'_{15}(t+1) &= S_{15}(t) \& z_{15}(t) \vee S'_{15}(t) \& \neg S_k(t); \\
 S'_{16}(t+1) &= S_{16}(t) \& z_{16}(t) \vee S'_{16}(t) \& \neg S_k(t); \\
 S_{17}(t+1) &= S'_1(t) \& S'_2(t) \& \neg x_1(t) \vee S_{17}(t) \& \neg z_{17}(t); \\
 S_{18}(t+1) &= S_{17}(t) \& z_{17}(t) \& \neg x_5(t) \vee S_{18}(t) \& \neg z_{18}(t); \\
 S_{19}(t+1) &= S_{18}(t) \& z_{18}(t) \& \vee S_{19}(t) \& \neg z_{19}(t); \\
 S_{20}(t+1) &= S_{18}(t) \& z_{18}(t) \& \vee S_{20}(t) \& \neg z_{20}(t); \\
 S_{21}(t+1) &= S_{18}(t) \& z_{18}(t) \& \vee S_{21}(t) \& \neg z_{21}(t); \\
 S'_{19}(t+1) &= S_{19}(t) \& z_{19}(t) \vee S'_{19}(t) \& \neg S_{22}(t); \\
 S'_{20}(t+1) &= S_{20}(t) \& z_{20}(t) \vee S'_{20}(t) \& \neg S_{22}(t); \\
 S'_{21}(t+1) &= S_{21}(t) \& z_{21}(t) \vee S'_{21}(t) \& \neg S_{22}(t); \\
 S_{22}(t+1) &= S'_{19}(t) \& S'_{20}(t) \& S'_{21}(t) \vee S_{17}(t) \& x_5 \vee S'_{22}(t) \& \neg z_{22}(t);
 \end{aligned}$$

$$\begin{aligned}
 S_{23}(t+1) &= S_{22}(t) \& z_{22}(t) \vee S_{23}(t) \& \neg z_{23}(t); \\
 S_{24}(t+1) &= S_{23}(t) \& z_{23}(t) \vee S_{24}(t) \& \neg z_{24}(t); \\
 S_{25}(t+1) &= S_{23}(t) \& z_{23}(t) \vee S_{25}(t) \& \neg z_{25}(t); \\
 S_{26}(t+1) &= S_{23}(t) \& z_{23}(t) \vee S_{26}(t) \& \neg z_{26}(t); \\
 S_{27}(t+1) &= S_{23}(t) \& z_{23}(t) \vee S_{27}(t) \& \neg z_{27}(t); \\
 S'_{24}(t+1) &= S_{24}(t) \& z_{24}(t) \vee S'_{24}(t) \& \neg S_k(t); \\
 S'_{25}(t+1) &= S_{25}(t) \& z_{25}(t) \vee S'_{25}(t) \& \neg S_k(t); \\
 S'_{26}(t+1) &= S_{26}(t) \& z_{26}(t) \vee S'_{26}(t) \& \neg S_k(t); \\
 S'_{27}(t+1) &= S_{27}(t) \& z_{27}(t) \vee S'_{27}(t) \& \neg S_k(t); \\
 S'_k(t+1) &= S_k(t) \& z_k(t) \vee S'_{13}(t) \& S'_{14}(t) \& S'_{15}(t) \& S'_{16}(t) \vee S'_{24}(t) \& S'_{25}(t) \& S'_{26}(t) \& S'_{27}(t) \vee S_k(t) \& \neg z_k(t).
 \end{aligned}$$

Аналогично предыдущим случаям в СКУ₃ учтены условия сохранения событий. Для параллельных событий (S₈, S₉; S₁₀, S₁₁; S₁₃ – S₁₆; S₁₉ – S₂₁; S₂₄ – S₂₇) принято, что факт завершения каждого S_i-го события отмечается последующим появлением события S_i'. Например, для подтверждения фактов завершения событий S₁ и S₂ определены события S₁' и S₂'. С того момента t, как только высказывания S₁'(t) и S₂'(t) одновременно станут истинными («true»), в зависимости от значения входного символа x₁ появится одно из событий S₃ или S₁₇, которое в сле-

дующем такте «прекратит» выполнение событий S_1' и S_2' .

Выводы

Таким образом, построены системы канонических уравнений для алгоритмов обработки кэшируемых данных в системе хранения данных. Такая математическая модель является, по существу, исполнимой формализованной спецификацией. Она позволяет осуществить непосредственный переход от СКУ к программной или аппаратной реализации данных алгоритмов.

Список литературы

1. Вашкевич Н.П. Выразительные возможности и эффективность формального языка, построенного на основе использования моделей недетерминированных автоматов //

Известия высших учебных заведений. Поволжский регион. Технические науки. – 2006. – № 6. – С. 67–77.

2. Вашкевич Н.П., Зубков В.А. Алгоритм синтеза цифровых автоматов Мура с использованием языка исчисления предикатов // Вычислительная техника: Учебные записки. – Вып. 3. Пенза: Пенз. политехн. ин-т, 1969 – С. 25–36.

3. Вашкевич Н.П. Об одном способе синтеза цифровых автоматов по граф-схеме алгоритма с параллельными ветвями // Вычислительная техника: Межвуз. сб. науч. тр., Вып. 1, 2. – Пенза: Пенз. политехн. ин-т, 1973.

4. Вашкевич Н.П., Бикташев Р.А. Достоинство формального языка, основанного на концепции недетерминизма, при структурной реализации параллельных систем логического управления процессами и ресурсами // Известия высших учебных заведений. Поволжский регион. Технические науки. – 2011. – № 1 (7). – С. 3–11.

5. Sibiryakov M.A., Vasyaeva E.S., Koshpaev A.A. Analysis and comparison of cache memory control methods in storage systems // In the World Scientific. Ser.: Natural Technical Sciences. – Krasnoyarsk, 2014. – № 10 (58). – P. 276–280.