

Р.М. Бабаков, А.А. Баркалов

## Операционная реализация функции выходов микропрограммного автомата

Предложен новый принцип организации схемы формирования микроопераций микропрограммного автомата в виде множества функциональных узлов. Рассмотрен пример операционной реализации схемы при заданных кодах состояний автомата.

**Ключевые слова:** микропрограммный автомат, операционный автомат переходов, функция выходов, аппаратурные затраты.

Запропоновано новий принцип організації схеми формування мікрооперацій мікропрограмного автомата у вигляді множини функціональних вузлів. Розглянуто приклад операційної реалізації схеми при заданих кодах станів автомата.

**Ключові слова:** мікропрограмний автомат, операційний автомат переходів, функція виходів, апаратурні витрати.

**Введение.** В современных вычислительных системах функцию координирования работы всех узлов выполняет устройство управления (УУ). Один из структурных представлений УУ – микропрограммный автомат (МПА), схема которого характеризуется высоким быстродействием при значительных аппаратурных затратах [1, 2], а одним из путей решения задачи оптимизации затрат аппаратуры в логической схеме МПА есть разработка его новых структур и методов синтеза [3].

Представление функции переходов автомата в виде множества частичных функций приводит к структуре микропрограммного автомата с операционным автоматом переходов (МПА с ОАП) [4, 5]. Данный подход предполагает использовать для преобразования кодов состояний набор функциональных узлов, аппаратурные затраты в каждом из которых не зависят или зависят незначительно от количества автоматных переходов, реализуемых этим узлом. Это делает схему формирования переходов МПА структурно схожей с операционным автоматом, в котором выбор операции, выполняемой над кодом текущего состояния, зависит как от самого кода текущего состояния, так и от входных сигналов автомата.

Операционная реализация функции переходов автомата не накладывает ограничений на способ реализации функции выходов. Схема формирования микроопераций в МПА традиционно реализуется по системе булевых уравнений. В статье предлагается возможность реализации функции выходов операционным способом по аналогии с операционным автоматом переходов.

### Микропрограммный автомат с операционным автоматом переходов

Структурная схема МПА с ОАП [4] приведена на рис. 1.

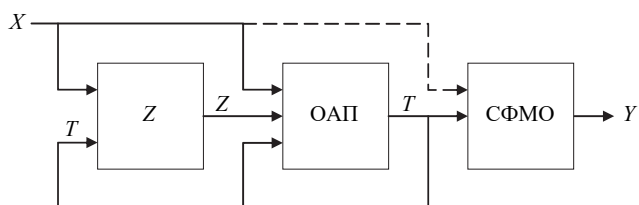


Рис. 1

В данной структуре блок ОАП представляет собой операционный автомат, выполняющий преобразование кода текущего состояния  $T$  и множества входных сигналов  $X$  под управлением кода операции переходов  $Z$ , формируемого специальной  $Z$ -подсхемой. Код  $T$  и сигналы  $X$  поступают также на вход  $Z$ -подсхемы, что позволяет сопоставлять операцию ОАП отдельному переходу автомата.

Схема формирования микроопераций (СФМО) реализует функцию выходов автомата, формируя множество микроопераций  $Y$  в соответствии с выражением (1) в случае автомата Мили и (2) в случае автомата Мура. Таким образом, наличие в структуре МПА с ОАП связи, показанной на рис. 1 пунктиром, позволяет считать данную структуру автоматом Мили, отсутствие связи – автоматом Мура.

$$Y = Y(T, X). \quad (1)$$

$$Y = Y(T). \quad (2)$$

Операционный автомат переходов включает в себя операционную часть (ОЧ) и регистр памяти (РП) (рис. 2). Операционная часть обра-

зована множеством комбинационных схем  $KC_i$ , каждая из которых реализует один из  $N$  законов преобразования кодов состояний, реализуя, таким образом, множество операций переходов (ОП) автомата. Выходы  $d_i$  блоков  $KC_i$  мультиплексируются под управлением сигналов  $Z$ . Получаемый в результате мультиплексирования код состояния перехода  $d$  поступает в регистр памяти, который есть единственной регистровой схемой в ОАП, и в каждой операции выступает одновременно в качестве регистра исходных данных и регистра результата [5].

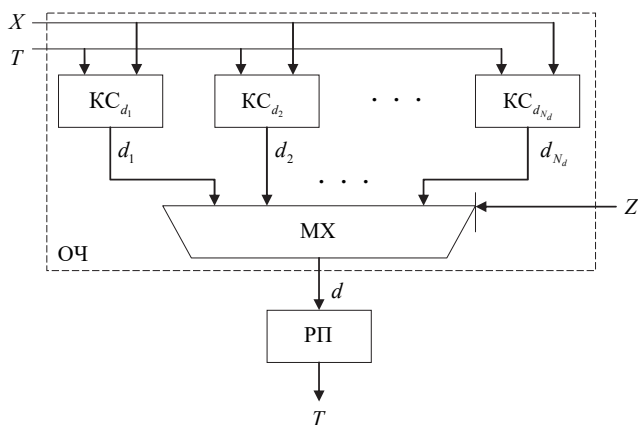


Рис. 2

### Операционная реализация функции выходов

Использование принципа операционного преобразования кодов состояний не накладывает каких-либо ограничений на синтез схемы СФМО. В процессе синтеза ОАП и  $Z$ -подсхемы осуществляется, в том числе и кодирование состояний, результаты которого должны учитываться при синтезе  $Z$ -подсхемы. Традиционно блок СФМО синтезируется по системе булевых уравнений (1) или (2) и реализуется в базе комбинационной логики или запоминающих устройств [2, 3].

Рассмотрим возможность распространения принципа операционного преобразования кодов состояний на функцию выходов автомата. В этом случае преобразование кодов состояний будет по-прежнему осуществляться с помощью некоторого множества операций (назовем их операциями выходов (ОВ)), однако код

текущего состояния будет преобразовываться не в код следующего состояния, а в микрокоманду (набор микроопераций), соответствующую аргументам функции выходов автомата.

Структурные схемы блока СФМО для автоматов Мили (а) и Мура (б) показаны на рис. 3. Формирование выходных сигналов МПА осуществляется с помощью множества ОВ  $\{O_1, O_2, \dots, O_W\}$ , в котором каждая ОВ  $O_i$  представлена в виде отдельной комбинационной схемы и реализует функцию  $Y(O_i)$ , значениями которой служит некоторое подмножество наборов выходных сигналов автомата. Аргументами ОВ являются, в случае автомата Мили, код текущего состояния  $T$  и входные сигналы  $X$ , в случае автомата Мура – только код  $T$ . По аналогии с блоком ОЧ на рис. 2, набор комбинационных схем, реализующих ОВ, обозначен пунктиром.

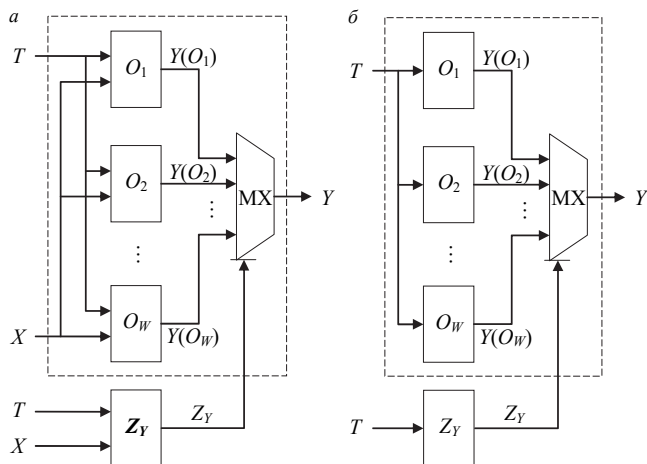


Рис. 3

Для выбора ОВ используется мультиплексор (MX) результата, управляемый кодом ОВ  $Z_Y$ , формируемым специальной  $Z_Y$ -подсхемой. На вход  $Z_Y$ -подсхемы поступают сигналы  $T$  и  $X$  (в случае автомата Мили) либо только сигналы  $T$  (в случае автомата Мура). Таким образом, в случае автомата Мили ОВ сопоставляется отдельному переходу автомата, в случае автомата Мура – отдельному состоянию автомата. На выходе мультиплексора формируется микрокоманда, поступающая в объект управления микропрограммного автомата.

Проанализируем структуры на рис. 3.

• Пусть МПА задан граф-схемой алгоритма (ГСА) [2, 3]. Как известно, при синтезе МПА микрокоманды, закрепленные за операторными вершинами ГСА, считаются заданными и не могут выбираться произвольно. В то же время структурные (двоичные) коды состояний не определяются исходной ГСА и могут выбираться любым удобным способом.

При операционной реализации функции переходов как ОП, так и коды состояний – вариативны. Это позволяет не только выбирать ОП на основании кодов состояний, но и задавать коды состояний с их учетом. Обязательным ограничением есть система переходов заданной ГСА.

При операционной реализации функции выходов коды состояний и формируемые микрокоманды инвариантны: коды состояний выбраны в процессе реализации функции переходов, а содержимое микрокоманд задано исходной ГСА. Единственным вариативным элементом есть множество операций, используемых для формирования микрокоманд.

Задача формирования множества операций схемы СФМО тривиально решается для случая, когда каждая микрокоманда формируется с помощью отдельной операции, т.е. когда число операций  $W$  в СФМО равно числу операторных вершин ГСА. Однако в этом случае следует ожидать, что получаемая в результате логическая схема не будет иметь преимуществ по аппаратным затратам в сравнении с канонической реализацией функции выходов по системе булевых уравнений. Это связано, в том числе, с необходимостью использования в составе схемы СФМО мультиплексора и  $Z_Y$ -подсхемы, которые, при большой разрядности микрокоманды и максимально возможном числе операций выходов, по суммарным затратам аппаратуры могут превышать затраты на каноническую реализацию СФМО.

Снижение аппаратных затрат в схеме СФМО при операционной реализации функции выходов возможно в случае, если некоторые ОВ формируют несколько микрокоманд (по аналогии с тем, как одна ОП реализует несколько переходов автомата). При заданных

кодах состояний (аргументах функции) и содержанием операторных вершин (значениях функции) поиск таких операций сводится к математической задаче интерполяции по базовым точкам [6] на элементах некоторого разбиения множества операторных вершин. Для каждого из возможных разбиений может быть получено множество интерполирующих функций, соответствующее операциям  $O_1 - O_W$  (рис. 3). Минимизация аппаратных затрат в СФМО заключается в поиске такого разбиения множества операторных вершин и соответствующего этому разбиению множества ОВ, при котором суммарные аппаратные затраты в СФМО (с учетом  $Z_Y$ -подсхемы) оказываются минимально возможными.

• При операционной реализации функции переходов разрядность структурных кодов аргумента и результата любой ОП одинакова и равна разрядности  $R$  структурного кода состояния. Данная особенность позволяет использовать стандартные  $R$ -разрядные функциональные блоки, обладающие, как правило, регулярной внутренней структурой.

Формируемая схемой СФМО микрокоманда образована  $N$  структурными сигналами, соответствующими микрооперациям  $y_1 - y_N$ . Поскольку величина  $N$  определяется исходной ГСА и в общем случае не зависит от разрядности кода состояния  $R$  и количества логических условий  $L$ , при операционной реализации функции выходов разрядность аргументов ОВ, равная  $R$  в случае автомата Мура и  $R + L$  в случае автомата Мили, может не совпадать с разрядностью  $N$  результата. Это требует использования для реализации ОВ нестандартных функциональных блоков.

• В схеме мультиплексора результата блока ОЧ (рис. 2) разрядность каждого канала равна разрядности  $R$  структурного кода состояния автомата, а число каналов соответствует числу используемых ОП.

В схеме на рис. 3 число выходов каждой схемы  $O_i$  равно числу микроопераций  $N$ , формируемых управляющим автоматом. Если, например,  $N = 100$  (что типично, например, для МПА АЛУ (арифметико-логического устрой-

ства) [7]), а  $R = 10$ , то при одинаковом количестве ОП в блоке ОЧ ОАП и ОВ в схеме СФМО сложность мультиплексора результата в схеме будет в 10 раз выше сложности мультиплексора в блоке ОЧ. Если дополнительно предположить, что количество  $W$  ОВ в СФМО будет значительно больше количества ОП  $N_d$  переходов в блоке ОЧ ОАП, то сложность мультиплексора на рис. 3 на фоне сложности остальных блоков логической схемы МПА может оказаться неприемлемо большой.

Следует отметить, что операционная реализация функции выходов сама по себе не накладывает каких-либо требований на реализацию функции переходов автомата. При реализации функции выходов МПА операционным способом возможны следующие ситуации:

- Функция переходов реализуется каноническим способом по системе булевых уравнений. В этом случае целесообразно сначала выполнить синтез схемы СФМО, определив в его процессе структурные коды состояний, после чего построить систему канонических уравнений функции переходов с учетом известных кодов состояний. В данном случае процесс синтеза логической схемы автомата представляется наиболее простым.

- Функция переходов реализуется операционным способом, причем синтез ОАП выполняется до синтеза СФМО. В этом случае коды состояний, выбранные в процессе синтеза ОАП, становятся исходными данными для синтеза схемы СФМО. При этом следует ожидать, что инвариантность кодов состояний и наборов микроопераций при синтезе СФМО не будет способствовать экономии аппаратных затрат в сравнении с канонической реализацией СФМО.

- Функция переходов реализуется операционным способом, причем синтез ОАП выполняется после синтеза СФМО. В этом случае коды состояний, выбранные в процессе синтеза СФМО, становятся исходными данными для синтеза схемы ОАП. Следует ожидать, что в данном случае усложняется достижение выигрыша по аппаратным затратам в блоке ОАП в сравнении с канонической реализацией функции переходов.

- Функция переходов реализуется операционным способом, причем синтез ОАП выполняется совместно с синтезом СФМО. В этом случае коды состояний могут выбираться таким образом, чтобы способствовать экономии аппаратных затрат при реализации как функции переходов, так и функции выходов автомата. Однако такой подход может усложнить процесс синтеза МПА, поскольку все основные этапы синтеза будут взаимосвязаны и не могут быть разделены на отдельные этапы синтеза функции переходов и функции выходов.

### Пример операционной реализации функции выходов

Рассмотрим пример операционной реализации функции выходов МПА с ОАП, заданного ГСА  $\Gamma$  (рис. 4), отмеченной состояниями автомата Мура.

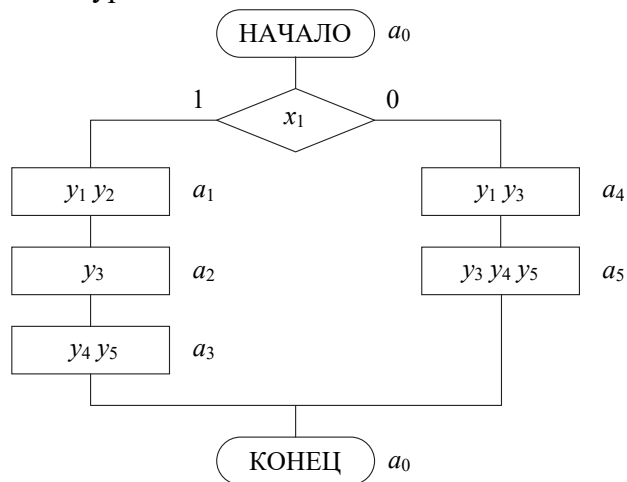


Рис. 4

Пусть для автомата выполнен синтез функции переходов, в соответствии с которым сформированы коды состояний  $K(a_0) - K(a_5)$  (табл. 1). Анализ рис. 4 показывает, что автомат формирует  $N = 5$  микроопераций  $y_1 - y_5$ .

Т а б л и ц а 1

$a_i$	$a_0$	$a_1$	$a_2$	$a_3$	$a_4$	$a_5$
$K(a_i)$	$011_2 =$ $= 3_{10}$	$010_2 =$ $= 2_{10}$	$001_2 =$ $= 1_{10}$	$000_2 =$ $= 0_{10}$	$101_2 =$ $= 5_{10}$	$100_2 =$ $= 4_{10}$

Поставим в соответствие каждой формируемой микрокоманде пятиразрядный двоичный вектор вида  $\langle y_1, y_2, y_3, y_4, y_5 \rangle$ , в котором каждый компонент  $y_i$  равен единице в том случае, если в данной МК микрооперация  $y_i$  фор-

мируется, и нулю в противном случае. Составим таблицу выходов МПА (табл. 2), содержащую следующие столбцы:

Таблица 2

$a_i$	$K(a_i)$	$Y$
$a_0$	$011_2 = 3_{10}$	0 0 0 0 0
$a_1$	$010_2 = 2_{10}$	1 1 0 0 0
$a_2$	$001_2 = 1_{10}$	0 0 1 0 0
$a_3$	$000_2 = 0_{10}$	0 0 0 1 1
$a_4$	$101_2 = 5_{10}$	1 0 1 0 0
$a_5$	$100_2 = 4_{10}$	0 0 1 1 1

где  $a_i$  – состояние автомата;  $K(a_i)$  – двоичное и десятичное значения кода состояния  $a_i$  в соответствии с табл. 1,  $Y$  – двоичный вектор вида  $\langle y_1, y_2, y_3, y_4, y_5 \rangle$ , соответствующий операторной вершине ГСА, отмеченной состоянием  $a_i$ .

Сопоставим состояниям  $a_0, a_1$  и  $a_5$  следующую ОВ:

$$Y = 00.K(a_i) \oplus 00011_2. \quad (3)$$

В данном выражении двоичный код состояния  $K(a_i)$  приводится к пятиразрядному формату путем добавления двух старших разрядов, равных нулю. Полученный вектор складывается по модулю 2 с вектором–константой  $00011_2$ . Результат операции – пятиразрядный вектор, отождествляемый с формируемой микрокомандой. Например, формирование микрокоманды для состояния  $a_5$  осуществляется в следующем порядке:

$$00.K(a_5) = 00100;$$

$$00100 \oplus 00011 = 00111;$$

$$00111 \equiv \{y_3, y_4, y_5\}.$$

Состояниям  $a_1, a_2, a_4$  поставим в соответствие следующую ОВ:

$$Y = (L_2(K(a_i)) + 10_{10}) \oplus 01010_2. \quad (4)$$

Код  $K(a_i)$  сначала приводится к пятиразрядному двоичному формату путем логического сдвига влево на два разряда (операция  $L_2$ ). Полученный пятиразрядный двоичный вектор интерпретируется как целое число и складывается с константой  $10_{10}$ . Результат сложения рассматривается как пятиразрядный двоичный вектор и складывается по модулю 2 с вектором–константой  $01010_2$ . Например, формиро-

вание микрокоманды для состояния  $a_i$  происходит в таком порядке:

$$L_2(K_2(a_1)) = 01000;$$

$$01000_2 = 8_{10};$$

$$8_{10} + 10_{10} = 18_{10};$$

$$18_{10} = 10010_2;$$

$$10010 \oplus 01010 = 11000;$$

$$11000 \equiv \{y_1, y_2\}.$$

Таким образом, для реализации функции выходов МПА, заданного ГСА  $\Gamma$ , достаточно  $W = 2$  ОВ. Закодируем данные операции двоичным кодом  $Z$  разрядности  $\log_2(W) = 1$ , который в схеме на рис. 3, б будем представлять переменной  $z_1$ . Пусть  $K(O_1) = 0$ ;  $K(O_2) = 1$ . Составим таблицу истинности  $Z_Y$ -подсхемы (табл. 3), состоящую из столбцов

$a_i$  – состояние автомата;

$K(a_i)$  – код состояния  $a_i$  (табл. 1);

$z_1$  – код ОВ, сопоставленной состоянию  $a_i$ .

Таблица 3

$a_i$	$K(a_i)$	$Z_1$
$a_0$	0 1 1	0
$a_1$	0 1 0	1
$a_2$	0 0 1	1
$a_3$	0 0 0	0
$a_4$	1 0 1	1
$a_5$	1 0 0	0

В соответствии с данной таблицей, функции  $z_1$  соответствует дизъюнктивная нормальная форма (ДНФ)  $z_1 = \bar{T}_1 T_2 \bar{T}_3 \vee \bar{T}_1 \bar{T}_2 T_3 \vee T_1 \bar{T}_2 T_3$ . При реализации  $Z_Y$ -подсхемы в базе запоминающих устройств (ЗУ) таблица истинности может быть легко преобразована в таблицу ЗУ.

**Заключение.** Предлагаемый в статье принцип операционной реализации функции выходов не зависит от способа реализации функции переходов автомата, что позволяет применять его не только в МПА с ОАП, но и в МПА с канонической структурой. Анализ особенностей применения принципа операционного преобразования кодов состояний при реализации функции выходов микропрограммного автомата демонстрирует принципиальную возмож-

ность использования данного подхода, но не позволяет однозначно судить о его эффективности. Вопрос целесообразности построения схемы формирования микроопераций со структурой, приведенной на рис. 3, требует исследований, включающих в себя разработку соответствующих структур и методов синтеза МПА, а также оценку их эффективности в сравнении с известными структурами.

1. Глушков В.М. Синтез цифровых автоматов. – М.: Физматгиз, 1962. – 476 с.
2. Баранов С.И. Синтез микропрограммных автоматов. – Л.: Энергия, 1979. – 232 с.
3. Баркалов А.А. Синтез устройств управления на программируемых логических устройствах. – Донецк: ДонНТУ, 2002. – 262 с.

4. Бабаков Р.М., Баркалов А.А. Реализация функции переходов микропрограммного автомата на базе операционного автомата // УСиМ. – 2015. – № 5. – С. 22–29.
5. Бабаков Р.М., Баркалов А.А., Ярош И.В. Операционный автомат переходов // Сб.научн. тр. ДонНТУ. Серия: «Вычислительная техника и автоматизация». – Красноармейск: ДонНТУ, 2015. – 1 (28). – С. 33–40.
6. Половко А.М., Бутусов П.Н. Интерполяция. Методы и компьютерные технологии их реализации. – СПб.: БХВ–Петербург, 2004. – 320 с.
7. Жмалкин А.П. Архитектура ЭВМ. – Там же, 2006. – 320 с.

Поступила 07.03.2017

Тел. для справок: +48 68 328-2693 (Винница, Зеленая Гора, Польша)

E-mail: r.babakov@donnu.edu.ua, cpld@mail.ru, a.barkalov@imei.uz.zgora.pl

© Р.М. Бабаков, А.А. Баркалов, 2017

UDC 004.2

R.M. Babakov<sup>1</sup>, A.A. Barkalov<sup>2</sup>

## Operational Realization of Output Function of Microprogram Final-State Machine

<sup>1</sup> PhD in Techn. Sciences, associate professor, Department of applied mechanics and computer technologies, Vasyl' Stus Donetsk National Univ., 600-richa str., 21, 21021, Vinnytsia UKRAINE, E-mail: r.babakov@donnu.edu.ua, cpld@mail.ru

<sup>2</sup> Doctor of Technical Sciences, Professor of Institute of Computer Engineering and Electronics, University of Zielona Gora (Poland), E-mail: a.barkalov@imei.uz.zgora.pl

**Keywords:** microprogram final-state machine, datapath of transitions, output function, hardware expenses.

**Introduction.** The object of the research is the microprogram final-state machine with the datapath of transitions. In the digital devices, the microprogram final-state machine performs the functions of control unit and coordinates the work of other units of the system. One of the topical scientific and practical problems is the reduction of the hardware expenses in the logical circuit of the microprogram final-state machine.

**Methods.** One of the way to solve this problem is to develop new microprogram final-state machine structures and the methods for their synthesis. One of such structures is a microprogram final-state machine with datapath of transitions. In this structure, the transition formation circuit is implemented in the form of an datapath consisting of the number of the separate operational blocks. Each operational block implements its law of converting state codes and input signals, realizing a disjoint subset of microprogram transitions. Herewith, the hardware expenses in the operational block do not depend or depend insignificantly on the number of microprogram transitions it implements. This allows under the certain conditions to achieve the reducing of hardware expenses in comparison with the implementation of the transition function of the final-state machine by the canonical method using the system of Boolean equations. The approach is called the operational realization of the transition function of the final-state machine, and consists of the representation of the circuit for forming the transitions of the microprogram final-state machine in the form of an datapath.

**Results.** It is proposed to use the principle of operational realization of the transition function of the final-state machine to realize the output function, which consists of the microinstructions (sets of microoperations) formation. In this case, the circuit for forming microoperations is represented as a set of operational blocks, each of which implements its own law of transformation of the current state code and input signals into a set of microoperations. If each operational unit is responsible for the formation of a subset of microinstructions and the hardware expenses for its implementation are less than the expenses of implementing this subset in a canonical way, there is a savings in hardware expenses in the circuit for generating microoperations compared to its canonical implementation.

**Conclusion.** It is shown that the operational realization of the output function of the microprogram final-state machine is possible both for the canonical and for the operational implementation of the transition function. In the first case, state codes are defined during the operational realization of the output function. In the second case, state codes are defined in the process of joint implementation of the transition and output functions by the operational method. At the same time, the formalization of the process of operational realization of the output function is not obvious and it forms a separate line of the future research.