

## ОБОБЩЕНИЕ МАТЕМАТИЧЕСКОЙ МОДЕЛИ МИКРОПРОГРАММНОГО АВТОМАТА НА СЧЕТЧИКЕ

**Актуальность.** Решена задача разработки структурных модификаций микропрограммного автомата на счетчике, которые могут способствовать снижению аппаратных затрат в логической схеме микропрограммного автомата по сравнению с известными структурами.

**Цель работы** – обобщение структурных особенностей микропрограммного автомата на счетчике с использованием математической модели на основе промежуточной алгебры переходов.

**Метод.** Проанализирована известная математическая модель микропрограммного автомата на счетчике, основанная на представлении функции переходов в виде двух частичных функций. Использование в структуре автомата инкрементного счетчика выражается в данной модели промежуточной алгеброй переходов, сигнатура которой образована единственной функцией инкремента. При этом аргументом функции является код текущего состояния автомата, интерпретируемый как целое число без знака. Для рассмотренной математической модели сделан ряд обобщений в части количества промежуточных алгебр переходов, их сигнатур и носителей. Проанализированы изменения в математической модели и структуре автомата на счетчике, являющиеся следствием сделанных обобщений.

**Результаты.** На основании сделанных обобщений получены обобщенная структурная схема и математическая модель микропрограммного автомата с неканоническим способом реализации функции переходов. Проведены экспериментальные исследования эффективности разработанной обобщенной структуры МПА по критерию аппаратных затрат.

**Выводы.** Полученные в данной работе результаты могут быть использованы при разработке новых структур и формальных методов синтеза микропрограммных автоматов с неканонической реализацией функции переходов, ориентированных на оптимизацию аппаратных затрат в логической схеме автомата.

**Ключевые слова:** микропрограммный автомат на счетчике, математическая модель, промежуточная алгебра переходов, аппаратные затраты.

### НОМЕНКЛАТУРА

МПА – микропрограммный автомат;  
 КС – комбинационная схема;  
 МХ – мультиплексор;  
 РП – регистр памяти;  
 СФМО – схема формирования микроопераций;  
 ГСА – граф-схема алгоритма;  
 ЛПС – линейная последовательность состояний;  
 ПК – прямой код;  
 ДК – дополнительный код;  
 $T$  – структурный (двоичный) код текущего состояния;  
 $T_i$  – разряд структурного кода текущего состояния;  
 $R$  – разрядность кода состояния;  
 $X$  – множество входных сигналов;  
 $Y$  – множество микроопераций, формируемых автоматом;  
 $\delta$  – абстрактная функция переходов;  
 $d$  – структурная функция переходов (структурный код состояния перехода);  
 $D_i$  – разряд структурного кода состояния перехода;  
 $\delta_i$  – частичная абстрактная функция переходов;  
 $d_i$  – частичная структурная функция переходов;  
 $G_\delta$  – абстрактная алгебра переходов;  
 $G_d$  – структурная алгебра переходов;  
 $G_{\delta_i}$  – абстрактная подалгебра переходов;  
 $G_{d_i}$  – структурная подалгебра переходов;  
 $G_I$  – промежуточная алгебра переходов;

$A$  – множество состояний автомата;  
 $a_i$  – состояние автомата;  
 $K_I(a_i)$  – промежуточный код состояния  $a_i$ ;  
 $A_{\delta_i}$  – множество состояний, входящих в носитель подалгебры  $G_{\delta_i}$ ;  
 $K_I(A_{\delta_i})$  – множество промежуточных кодов состояний из множества  $A_{\delta_i}$ ;  
 $O_{Inc}$  – операция инкремента;  
 $O_{Dec}$  – операция декремента;  
 $N_I$  – количество промежуточных алгебр переходов;  
 $K_S(a_i)$  – структурный (двоичный) код состояния  $a_i$ ;  
 $x_i$  – входной сигнал автомата;  
 $K_I(x_i)$  – промежуточный код входного сигнала  $x_i$ ;  
 $X_{\delta_i}$  – множество входных сигналов, входящих в носитель подалгебры  $G_{\delta_i}$ ;  
 $K_I(X_{\delta_i})$  – множество промежуточных кодов входных сигналов из множества  $X_{\delta_i}$ ;  
 $L_1$  – блок логического сдвига на один разряд влево;  
 $\Sigma$  – сумматор;  
 $Z$  – сигнал управления мультиплексором;  
 $Z$  – структурный блок, формирующий сигнал  $Z$  ( $Z$ -подсхема).  
 $M$  – количество состояний автомата;  
 $B$  – количество переходов автомата;

$N$  – количество микроопераций, формируемых автоматом;  
 $E$  – эффективность МПА по критерию аппаратурных затрат;  
 $k_1$  – коэффициент минимизации аппаратурных затрат;  
 $k_2$  – доля автоматных переходов, реализуемых операционным способом;  
 $k_3$  – коэффициент эффективности использования блочной памяти ПЛИС FPGA.

## ВВЕДЕНИЕ

Одним из центральных узлов современных вычислительных систем является устройство управления, координирующее работу всех блоков системы [1]. Одним из способов реализации устройства управления является МПА, логическая схема которого характеризуется относительно высоким быстродействием при значительных аппаратурных затратах [2]. При этом актуальной научной задачей является разработка новых структур МПА, обладающих меньшими затратами аппаратуры по сравнению с известными структурами [2–4].

Одной из структурных разновидностей МПА является микропрограммный автомат на счетчике (С-автомат) [3]. В нем часть микропрограммных переходов реализуется каноническим способом по системе булевых уравнений, часть – с помощью инкрементного счетчика. Поскольку аппаратурные затраты в схеме счетчика не зависят от числа реализуемых с его помощью автоматных переходов, выигрыш в аппаратурных затратах в логической схеме МПА со счетчиком возрастает с увеличением доли таких переходов.

В настоящей работе для микропрограммного автомата на счетчике предлагается ряд обобщений, приводящих к различным структурным модификациям. Целью работы является формализация данных обобщений на уровне математических и структурных моделей.

## 1 ПОСТАНОВКА ЗАДАЧИ

Основной научной задачей, решаемой в данной работе, является разработка новых модификаций формальной математической модели автомата на счетчике, при-

водящих к соответствующим модификациям структуры автомата. Модификации математической модели представлены в виде обобщений, логически следующих из структурных особенностей автомата. Данную задачу следует рассматривать как часть более общей научно-практической задачи разработки новых структур и методов синтеза микропрограммных автоматов, способствующих уменьшению аппаратурных затрат в логической схеме автомата.

## 2 ОБЗОР ЛИТЕРАТУРЫ

Структурная схема автомата на счетчике приведена на рис. 1 [5]. Здесь комбинационная схема  $КС_{d_1}$  представляет собой  $R$ -разрядный инкрементор, где  $R$  – разрядность структурного (двоичного) кода состояния автомата. На вход инкрементора подается код текущего состояния  $T$ , на выходе формируется т.н. частичная структурная функция переходов, обозначенная символом  $d_1$ . Термин «частичная» подразумевает, что с помощью инкрементора реализуется лишь часть переходов автомата, причем, согласно [3], выигрыш в аппаратурных затратах при использовании С-автомата по сравнению с каноническим МПА [2] оказывается тем выше, чем большая доля переходов реализуется данным способом. Под термином «структурная» следует понимать представление аргументов и значений функции в виде векторов структурного (двоичного) алфавита.

Вторая часть переходов реализуется блоком  $КС_{d_2}$ , который синтезируется по системе булевых уравнений аналогично схеме формирования переходов в каноническом МПА [2] и формирует частичную структурную функцию переходов  $d_2$  на основании кода  $T$  и множества структурных входных сигналов  $X = \{x_1, \dots, x_L\}$ .

Выходы блоков  $КС_{d_1}$  и  $КС_{d_2}$  поступают в  $R$ -разрядный мультиплексор  $MX$ , формирующий структурную функцию перехода  $d$ , соответствующую коду состояния перехода автомата. Управление мультиплексором осуществляется блоком  $КС_{inc}$  в соответствии со значениями сигналов  $T$  и  $X$ . Сигнал  $Inc$  принимает нулевое значе-

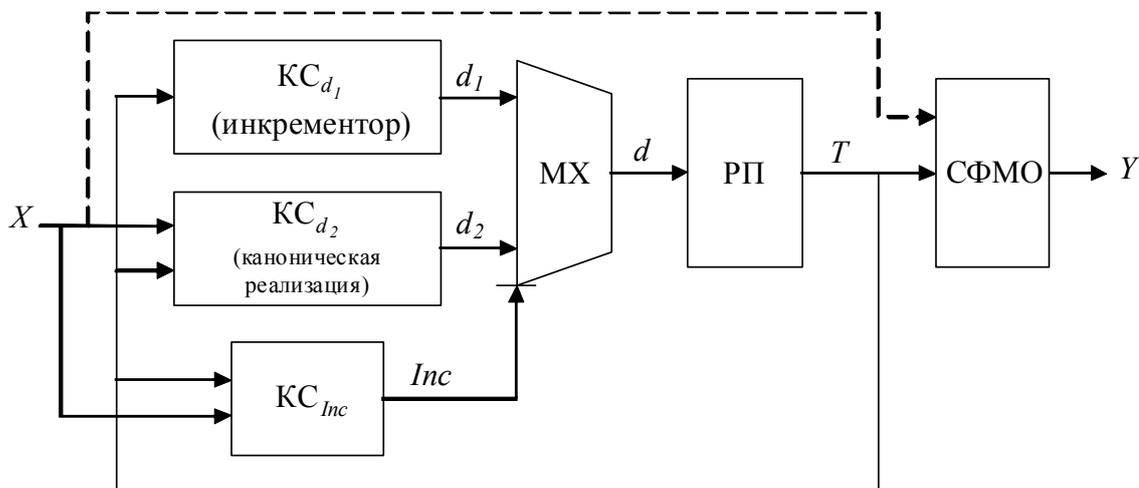


Рисунок 1 – Структурная схема автомата на счетчике

ние при переходах, реализуемых функцией  $d_1$ , и единичное значение при переходах, реализуемых функцией  $d_2$ .

Также в структуру входят регистр памяти и схема формирования микроопераций, назначение и архитектура которых аналогичны канонической структуре МПА [2]. Наличие связи, показанной пунктиром, позволяет считать данную структуру автоматом Мили, отсутствие – автоматом Мура.

В работе [5] предложена математическая модель автомата на счетчике, основанная на представлении функции переходов автомата в виде двух частичных функций и представляемая следующей системой изоморфизмов алгебр:

$$\begin{cases} G_{\delta} \leftrightarrow G_d; \\ G_{\delta_1} \leftrightarrow G_I \leftrightarrow G_{d_1}. \end{cases} \quad (1)$$

Здесь каждая алгебра представляет собой совокупность сигнатуры и носителя [6, 7]. Структура алгебр подробно рассмотрена в [5]. Сигнатура каждой алгебры образована соответствующей функцией, носитель – множествами ее аргументов и значений.

Изоморфизм  $G_{\delta} \leftrightarrow G_d$  устанавливает взаимно-однозначное соответствие между абстрактным автоматом (задаваемым, например, в форме ГСА, [2]) и структурным  $C$ -автоматом. Это выражается в том, что каждому состоянию и входному сигналу автомата ставится в соответствие уникальный структурный код, а каждому переходу в абстрактном автомате – единственный переход в структурном автомате. В  $C$ -автомате существование данного изоморфизма не зависит от способа разбиения функции переходов автомата на частичные функции [5].

Второй изоморфизм в системе (1) включает в себя три изоморфизма:  $G_{\delta_1} \leftrightarrow G_{d_1}$ ,  $G_{\delta_1} \leftrightarrow G_I$  и  $G_I \leftrightarrow G_{d_1}$ . Подалгебры  $G_{\delta_1}$  и  $G_{d_1}$  соответствуют той части переходов автомата, которая реализуется с помощью инкрементора. Возможность использования инкрементора обеспечивается существованием изоморфизмов данных подалгебр с промежуточной алгеброй переходов  $G_I$ , сигнатура которой образована функцией инкремента, а носитель представляет собой подмножество натуральных чисел. Это вызвано тем, что в используемая в  $C$ -автомате операция инкремента  $O_{Inc}$  определена для скалярных величин. Согласно [3], в процессе синтеза  $C$ -автомата на этапе формирования линейных последовательностей состояний каждое состояние  $a_i \in A$ , входящее в носитель подалгебры  $G_{\delta_1}$ , получает уникальный «скалярный» (промежуточный) код  $K_I(a_i)$  из множества натуральных чисел. «Скалярные» коды назначаются всем состояниям, переходы в которые или переходы из которых выполняются с помощью операции инкремента. Таким образом, результатом кодирования состояний, присутствующих в носителе подалгебры  $G_{\delta_1}$ , является множество скалярных кодов  $K_I(A_{\delta_1})$ . Состояния, отсутствующие в носителе  $G_{\delta_1}$ , «скалярных» кодов не получают.

Таким образом, носитель алгебры переходов  $G_I$  содержит множество  $K_I(A_{\delta_1})$ , сигнатура – операцию  $O_{Inc}$ :

$$G_I = \langle \{K_I(A_{\delta_1})\}, \{O_{Inc}\} \rangle. \quad (2)$$

В данном выражении операция  $O_{Inc}$  есть множество кортежей вида

$$\langle K_I(a_i), K_I(a_j) \rangle, \quad (3)$$

где  $K_I(a_i), K_I(a_j) \in K_I(A_{\delta_1})$ . Поскольку множество входных сигналов  $X$  не является аргументом операции  $O_{Inc}$ , оно отсутствует в носителе алгебры  $G_I$ , а его элементы – в кортеже (3).

### 3 МАТЕРИАЛЫ И МЕТОДЫ

Сделаем для автомата на счетчике ряд обобщений.

**Обобщение 1.** В сигнатуре промежуточной алгебры переходов может использоваться операция, отличная от операции инкремента.

В процессе синтеза  $C$ -автомата при кодировании состояний в исходной ГСА выделяют т.н. линейные последовательности состояний (ЛПС), в пределах которых переходы выполняются с помощью операции инкремента [4]. Кодирование состояний в пределах ЛПС выполняется в естественном порядке – так, чтобы десятичная интерпретация кода каждого следующего состояния была на единицу больше, чем у предыдущего состояния.

Заметим, что ничто не мешает в пределах ЛПС выполнить кодирование состояний в порядке убывания десятичных значений кодов состояний, что позволит реализовывать соответствующие переходы с помощью операции декремента. При этом в выражении (2) операция инкремента  $O_{Inc}$  будет заменена операцией декремента  $O_{Dec}$ , а в структуре на рис. 1 блок  $KC_{d_1}$  будет представлять собой  $R$ -разрядный декрементор. Такой автомат может быть назван  $D$ -автоматом, а методика его синтеза подобна  $C$ -автомату.

В более общем случае можно утверждать, что в сигнатуре промежуточной алгебры переходов может использоваться любая операция над кодами состояний (не только инкремент или декремент). Операция может быть определена как на множестве промежуточных кодов состояний (сложение или вычитание с константой, умножение или деление на константу, возведение в квадрат, извлечение корня и т.п.), так и на множестве структурных (двоичных) кодов состояний (конъюнкция или дизъюнкция с константой, логический или циклический сдвиг, инверсия всех или отдельных битов и т.п.) Допустимо выполнение таких операций над частью двоичных разрядов кода состояния. Таким образом, никаких принципиальных ограничений на выбор операции, используемой в сигнатуре промежуточной алгебры, не существует.

Как было отмечено, эффективность  $C$ -автомата по аппаратным затратам обусловлена тем, что затраты аппаратуры в инкременторе не зависят (при фиксированной разрядности структурного кода состояния) от числа автоматных переходов, реализуемых данным блоком. Выдвинем подобное требование к любой операции, используемой в сигнатуре промежуточной алгебры переходов. Как и в случае  $C$ -автомата, эффективность

по аппаратурным затратам будет тем выше, чем большее число переходов реализуется посредством данной операции.

**Обобщение 2.** Алгебраический синтез автомата допускает использование нескольких промежуточных алгебр.

Пусть микропрограммный автомат Мура задан ГСА  $\Gamma_1$  (рис. 2).

Выделим в  $\Gamma_1$  две ЛПС:  $\langle a_0, a_1, a_2, a_3 \rangle$  и  $\langle a_0, a_4, a_5 \rangle$ . Зададим состояниям промежуточные коды из множества натуральных чисел следующим образом:  $K_I(a_0) = 2$ ,  $K_I(a_1) = 3$ ,  $K_I(a_2) = 4$ ,  $K_I(a_3) = 5$ ,  $K_I(a_4) = 1$ ,  $K_I(a_5) = 0$ . Можно заметить, что при данных значениях промежуточных кодов состояния переходы в пределах первой ЛПС могут быть выполнены с помощью операции инкремента, в пределах второй ЛПС – с помощью операции декремента. Переход из состояния  $a_5$  с кодом 0 в состояние  $a_0$  с кодом 2 не может быть выполнен с помощью одной из этих операций и должен быть реализован каноническим способом по системе булевых уравнений.

Таким образом, можно говорить о трех частичных функциях переходов: функция  $\delta_1$  определяет переходы внутри первой ЛПС,  $\delta_2$  – внутри второй ЛПС,  $\delta_3$  определяет «канонический» переход  $a_5 \rightarrow a_0$ . Подобное разбиение позволяет сформировать две промежуточные алгебры переходов, первая из которых определяется выражением (1), вторая – выражением (1), в котором  $O_{Inc}$  заменена на  $O_{Dec}$ . Соответственно, математическая модель полученного автомата, по аналогии с (1), представляется выражением (4).

$$\begin{cases} G_\delta \leftrightarrow G_d; \\ G_{\delta_1} \leftrightarrow G_{I_1} \leftrightarrow G_{d_1}; \\ G_{\delta_2} \leftrightarrow G_{I_2} \leftrightarrow G_{d_2}. \end{cases} \quad (4)$$

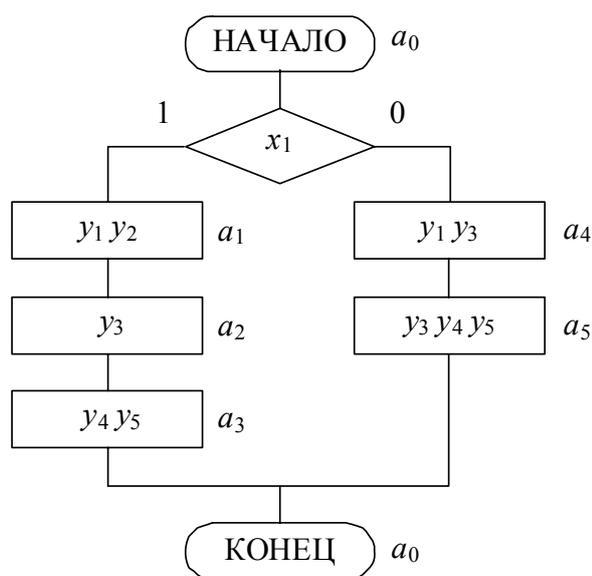


Рисунок 2 – Граф-схема алгоритма  $\Gamma_1$

В общем случае количество промежуточных алгебр может быть больше двух (в предельном случае – равно количеству переходов автомата). При этом система (4) примет следующий вид:

$$\begin{cases} G_\delta \leftrightarrow G_d; \\ G_{\delta_1} \leftrightarrow G_{I_1} \leftrightarrow G_{d_1}; \\ \dots \\ G_{\delta_{N_I}} \leftrightarrow G_{I_{N_I}} \leftrightarrow G_{d_{N_I}}. \end{cases} \quad (5)$$

Отметим, что целесообразность использования нескольких промежуточных алгебр определяется, помимо выигрыша в аппаратурных затратах, сложностью синтеза автомата, связанной с поиском необходимого разбиения функции переходов на частичные функции, выбором операций и специальным кодированием состояний.

**Обобщение 3.** В промежуточных алгебрах допустимы различные интерпретации структурных кодов состояний.

В структурной алгебре переходов  $G_d$  код состояния есть битовый вектор, уникальный в рамках используемого множества структурных кодов состояний. В вычислительной технике битовый вектор рассматривается не только как уникальный набор битов, но и допускает различную числовую (скалярную) интерпретацию [8]. Такими интерпретациями могут быть: число с фиксированной запятой (без знака, в прямом, обратном или дополнительном кодах), число с плавающей запятой (нормализованное, денормализованное, различной точности), число в двоично-десятичном формате (упакованное, неупакованное) и другие.

При фиксированном количестве двоичных разрядов каждый из данных форматов соответствует некоторому множеству чисел, над которым может быть определено множество операций различной аности. При этом любая операция может быть реализована схемным путем в большинстве известных элементных базисов, зачастую допуская различные архитектурные варианты реализации, характеризующиеся затратами аппаратуры, быстротой действия, регулярностью, универсальностью и т.д.

Один и тот же автоматный переход, рассматриваемый как преобразование одного структурного (двоичного) вектора в другой, может быть реализован при различных интерпретациях структурных кодов состояний. Пусть, например, для состояния  $a_i$  известен структурный код  $K_S(a_i) = 11101011_2$ . Данный код может интерпретироваться как целое без знака, равное  $235_{10}$ , либо как целое в прямом коде, равное  $-107_{10}$ , либо как целое в дополнительном коде, равное  $-21_{10}$ .

Пусть в автомате выполняется переход из состояния  $a_i$  в состояние  $a_j$ , при котором требуется преобразовать код состояния  $K_S(a_i) = 11101011_2 = 235_{10} = -107_{10} = -21_{10}$  в код  $K_S(a_j) = 00010100_2 = 20_{10} = +20_{10} = +20_{10}$ . Понятно, что в зависимости от числового формата кодов состояний будут использованы различные операции. Так, в беззнаковом десятичном формате такой опера-

ций будет сложение с константой  $41_{10}$ , в прямом коде – сложение с константой  $+127_{\text{ПК}}$ , в дополнительном коде – сложение с константой  $+41_{\text{ДК}}$ . При этом каждая из этих операций дает в результате нужный битовый вектор  $K_S(a_i) = 11101011_2$ .

Более того, при использовании некоторого числового формата одно и то же преобразование двоичных векторов может быть выполнено с помощью различных операций. Так, преобразование кода  $K_S(a_i) = -21_{\text{ДК}}$  в код  $K_S(a_j) = +20_{\text{ДК}}$  может быть выполнено с помощью операций « $K_S(a_i) + 41_{\text{ДК}}$ », « $(K_S(a_i) + 1) \cdot (-1)$ », « $K_S(a_i) \cdot (-1) + 1$ ». Все эти операции, фактически, выполняют преобразование двоичного вектора  $11101011$  в вектор  $00010100$ .

Заметим, что в рассматриваемом примере наиболее быстрым и экономным способом преобразования  $K_S(a_i) = 11101011_2$  в  $K_S(a_j) = 00010100_2$  является операция поразрядной инверсии. Однако целесообразность использования той или иной операции зависит от того, какое суммарное количество автоматных переходов может быть реализовано с помощью данной операции в проектируемом МПА.

**Обобщение 4.** Входные сигналы автомата могут выступать в качестве аргументов операции, образующей сигнатуру промежуточной алгебры переходов.

В соответствии с обобщением 3, в промежуточной алгебре переходов структурный код состояния интерпретируется как некоторая скалярная или векторная величина. Вместе с тем свою интерпретацию могут иметь и структурные коды входных сигналов, причем интерпретации кодов состояний и входных сигналов могут как совпадать, так и быть различными.

Пусть некоторая ГСА  $\Gamma_2$ , отмеченная состояниями автомата Мили, содержит фрагмент, изображенный на рис. 3. Данный фрагмент содержит два условных перехода из состояния  $a_1$ , которым в каноническом МПА соответствуют два различных термина в системе булевых уравнений функции переходов.

В автомате на счетчике любой из этих переходов (но не оба сразу) может быть реализован с помощью операции инкремента, другой – каноническим способом. Способ реализации того или иного перехода определяет блок  $KC_{Inc}$ , реализующий функцию

$$Inc = Inc(T, X). \tag{6}$$

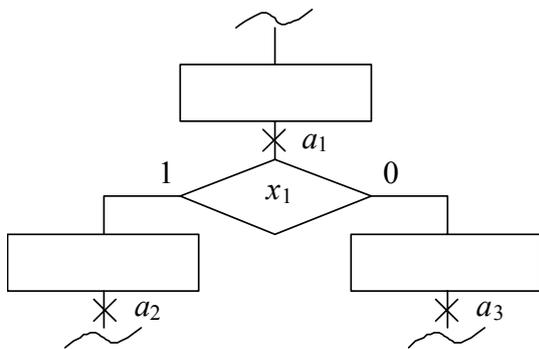


Рисунок 3 – Фрагмент ГСА  $\Gamma_2$

Как и в каноническом МПА, в системе уравнений блока  $KC_{Inc}$  каждому переходу соответствует уникальный булев терм, включающий прямое или инверсное значение переменной  $x_j$ . В то же время на вход инкрементора сигналы  $X$  не поступают, поскольку не являются аргументами операции инкремента.

По аналогии с С-автоматом дадим кодам состояний скалярную интерпретацию в формате целых чисел без знака, получив таким образом множество промежуточных кодов состояний  $K_I(a_i)$ . Реализуем оба перехода из состояния  $a_1$  с помощью одной операции  $O_1$ , представляемой в виде интервальной функции (7).

$$O_1 = \begin{cases} K_I(a_i) \cdot 2, & x_1 = 1; \\ K_I(a_i) \cdot 2 + 1, & x_1 = 0. \end{cases} \tag{7}$$

Интерпретируем логическое значение переменной  $x_1$  как скалярную величину в том же формате, что и промежуточные коды состояний. Таким образом, промежуточными кодами переменной  $x_1$  могут быть коды  $K_I(x_1) = 0_{10}$  или  $K_I(x_1) = 1_{10}$ . Это позволяет преобразовать выражение (7) к неинтервальному виду:

$$O_1 = K_I(a_i) \cdot 2 + K_I(x_1). \tag{8}$$

Пусть  $K_I(a_1) = 5$ . Тогда могут быть определены промежуточные коды состояний переходов  $K_I(a_2)$  и  $K_I(a_3)$ : при  $x_1 = 1$  получаем  $K_I(a_2) = K_I(a_1) \cdot 2 + K_I(x_1) = 5 \cdot 2 + 1 = 11_{10}$ , при  $x_1 = 0$  получаем  $K_I(a_3) = K_I(a_1) \cdot 2 + K_I(x_1) = 5 \cdot 2 + 0 = 10_{10}$ . Результаты использования выражения (8) должны учитываться при кодировании остальных состояний ГСА.

Логическая схема блока  $KC_{O_1}$ , реализующая операцию  $O_1$ , может быть такой, как на рис. 4. В ней блок « $L_1$ » реализует операцию логического сдвига  $R$ -разрядного структурного кода состояния на один разряд влево, что для беззнаковых целых чисел эквивалентно умножению на 2. Сумматор (блок « $\Sigma$ ») организован таким образом,

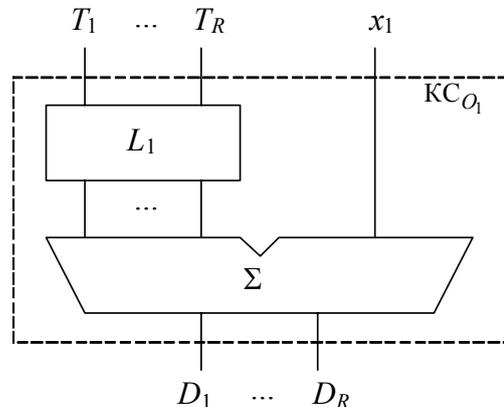


Рисунок 4 – Функциональная схема, соответствующая операции  $O_1$

что значение сигнала  $x_1$  прибавляется к младшему разряду кода текущего состояния.

Особенностью блока  $KC_{O_1}$  является то, что на его вход, помимо кода состояния  $T$ , подается входной сигнал  $x_1$ , используемый в операции  $O_1$  в качестве слагаемого и непосредственно влияющий на ее результат. Таким образом, в отличие от операции инкремента,  $O_1$  является операцией условного перехода и должна выполняться автоматом при переходе из состояния  $a_1$  независимо от текущего значения  $x_1$ .

При использовании операций, подобных  $O_1$ , в структуре автомата должен использоваться блок, управляющий мультиплексором подобно блоку  $KC_{Inc}$  в автомате на счетчике. Поскольку в нашем примере операция  $O_1$  сопоставляется не отдельному автоматному переходу, а всем переходам из состояния  $a_1$ , блок управления мультиплексором (назовем его  $Z$ -подсхемой) инициирует ее выполнение лишь по коду состояния  $a_1$ , что в системе булевых уравнений  $Z$ -подсхемы соответствует единственному терму, образованному переменными  $T_1, \dots, T_R$  и не содержащему переменных из множества  $X$ . Если бы переходы из состояния  $a_1$  (рис. 3) реализовывались двумя различными операциями, каждой из них соответствовал в системе уравнений  $Z$ -подсхемы соответствовал бы отдельный терм, образованный переменными  $T_1, \dots, T_R, x_1$ . Таким образом, применение условных операций, использующих входные сигналы в качестве аргументов, может способствовать упрощению схемы управления мультиплексором. Однако при этом следует учитывать аппаратные затраты в схеме, реализующей условную операцию, которые могут нивелировать выигрыш в  $Z$ -подсхеме.

Как было отмечено при изложении обобщения 1, выигрыш в аппаратных затратах от использования операций оказывается тем выше, чем больше переходов реализует каждая операция. В отношении условных операций переходов, подобных  $O_1$ , следует ожидать, что количество переходов, реализуемых такой операцией в рамках заданной ГСА, будет относительно небольшим (по сравнению с той же операцией инкремента). Однако, поскольку (исходя из обобщения 2) использование условных операций не исключает одновременного использования операций, не использующих входные сигналы в качестве аргументов, целесообразность подхода, определяемого обобщением 4, может следовать только из результирующих характеристик логической схемы автомата.

Способы использования входных сигналов в операциях переходов могут быть различными. Все способы интерпретации структурных кодов состояний (см. обобщение 3) в той или иной степени применимы и к структурным входным сигналам. В общем случае выбор способа интерпретации входных сигналов не является очевидным.

Если входные сигналы являются аргументам операции из сигнатуры промежуточной алгебры переходов, выражение (2), задающее данную алгебру, принимает следующий вид:

$$G_I = \langle \{K_I(A_{\delta_1}), K_I(X_{\delta_1})\}, \{O_1\} \rangle, \quad (9)$$

где операция  $O_1$  формально представлена множеством векторов вида (10).

$$\langle K_I(a_i), K_I(a_j), K_I(x_k) \rangle, \quad (10)$$

в которых где  $K_I(a_i), K_I(a_j) \in K_I(A_{\delta_1}), K_I(x_k) \in K_I(X_{\delta_1})$ .

**Обобщение 5.** Количество промежуточных алгебр может совпадать с количеством частичных функций переходов.

В примере, приведенном для обобщения 2 (ГСА  $G_1$ , рис. 2) переход из состояния  $a_5$  в состояние  $a_0$  при выбранных значениях промежуточных кодов состояний ( $K_I(a_5) = 0, K_I(a_0) = 2$ ) не может быть выполнен с помощью операций инкремента или декремента, вследствие чего должен быть реализован каноническим способом по системе булевых уравнений.

Однако ничто не мешает для реализации данного перехода использовать третью операцию, результатом которой над аргументом 0 является значение 2. Согласно обобщению 2, данное преобразование может быть выполнено с помощью различных операций: сложение с константой 2, инверсия второго справа двоичного разряда и т.п. При этом в математическую модель автомата добавится третья промежуточная алгебра переходов, носитель которой образован промежуточными кодами состояний  $a_0$  и  $a_5$ , а сигнатура содержит выбранную операцию.

В результате все переходы автомата будут реализованы операционным способом, и не один переход не будет реализован каноническим способом по системе булевых уравнений. Хотя реализация единственного перехода с помощью отдельной операции с точки зрения аппаратных затрат обычно менее оптимальна, чем его каноническая реализация, ничто не мешает использовать данную операцию для реализации других переходов автомата. В общем случае допустимы ситуации, когда в автомате, содержащем сотни переходов, все без исключения переходы реализуются с помощью нескольких операций, каждая из которых обеспечивает реализацию нескольких десятков переходов. Поскольку затраты аппаратуры в логической схеме каждой из операций не зависят от количества реализуемых переходов, следует ожидать существенной экономии аппаратных затрат по сравнению с каноническим автоматом. Безусловно, данный подход требует разработки специальных методов синтеза МПА, включающих такие этапы, как выбор множества операций, разбиение функции переходов на частичные функции, кодирование состояний, схемная реализация операций и другие.

#### 4 ЭКСПЕРИМЕНТЫ

Сделанные обобщения позволяют построить обобщенную структуру микропрограммного автомата, использующего неканонические способы преобразования кодов состояний (рис. 5).

В данной структуре блоки  $KC_{d_1}, \dots, KC_{d_{N_I}}$  соответствуют операциям, образующим сигнатуры  $N_I$  промежуточных алгебр.

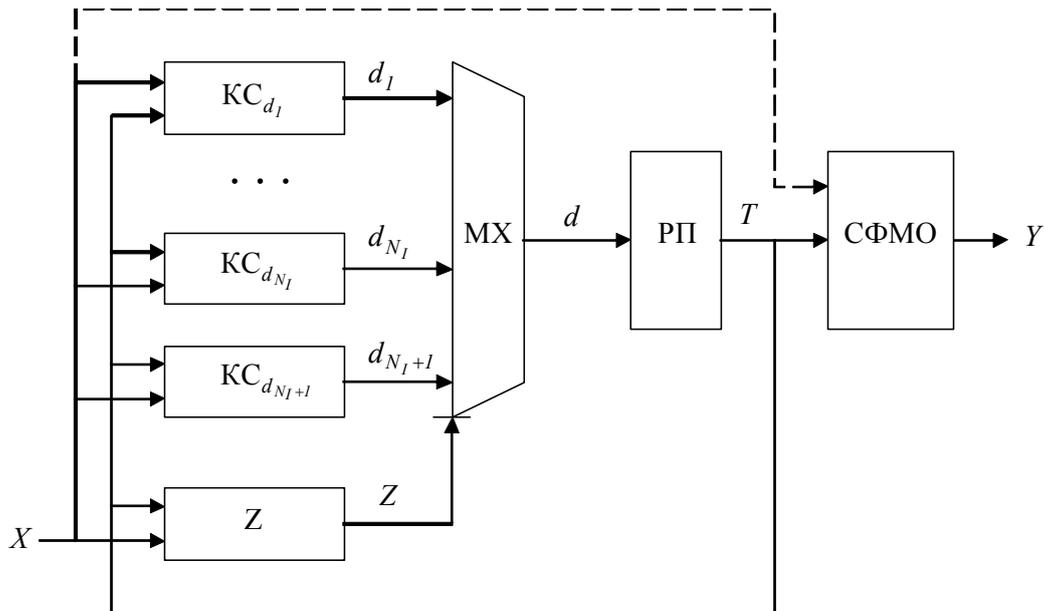


Рисунок 5 – Обобщенная структурная схема микропрограммного автомата, использующего неканонические способы преобразования кодов состояний

жучочных алгебр переходов. Согласно обобщению 1, каждый из этих блоков может имплементировать любую операцию, которая входит в сигнатуру промежуточной алгебры переходов. При этом аппаратные затраты в таком блоке не должны зависеть от количества переходов, реализуемых данным блоком. Блок  $KC_{d_{N_I+1}}$  не отождествляется с промежуточной алгеброй переходов и реализует подмножество переходов автомата каноническим способом по системе булевых уравнений. Согласно обобщению 4, каждый из блоков  $KC_{d_1}, \dots, KC_{d_{N_I}}$  может использовать входные сигналы автомата в качестве аргументов операции, реализуемой данным блоком.

Отметим, что в соответствии с обобщением 3 в каждой промежуточной алгебре переходов может использоваться свой способ интерпретации структурных кодов состояний, определяющий внутреннюю схемную реализацию соответствующего блока  $KC_{d_i}$ . Следует, однако, отметить, что, если несколько блоков используют одну и ту же интерпретацию структурных кодов состояний, существует возможность совместной реализации блоков, при которой один и тот же функциональный узел (например, сумматор) используется обоими блоками одновременно (метод обобщенных операторов, [9]). Данный подход может способствовать уменьшению аппаратных затрат в схеме автомата, а на структурном уровне приводит к представлению двух совместно реализуемых блоков  $KC_{d_i}$  и  $KC_{d_j}$  в виде одного блока с двумя выходами, каждый из которых соответствует своей частичной структурной функции переходов.

На основании обобщения 5 блок  $KC_{d_{N_I+1}}$ , реализующий часть переходов автомата каноническим способом, в частном случае может отсутствовать. Но поскольку на структурном уровне внутренняя архитектура блоков не детализируется, данный блок никак не выделяется среди других блоков  $KC_{d_i}$ .

Z-подсхема подобна блоку *Inc* в автомате на счетчике и формирует код  $Z$ , поступающий на управляющие входы мультиплексора. В результате, в каждом такте работы структурная функция перехода  $d$  принимает одно из значений  $d_1, \dots, d_{N_I+1}$ , представляющее собой структурный код состояния перехода.

В остальном данный автомат подобен автомату на счетчике. Наличие связи, показанной пунктиром, позволяет считать данную структуру автоматом Мили, отсутствие связи – автоматом Мура. Математическая модель данного автомата в части функции переходов задается выражением (5), в котором каждая промежуточная алгебра переходов формально представляется аналогично выражению (9).

Для МПА, изображенного на рис. 5, проведены экспериментальные исследования эффективности по критерию аппаратных затрат в сравнении с МПА с канонической структурой. Методология исследований изложена в [10] и тезисно определяется следующими положениями:

- эффективность структуры определяется отношением численно выраженной величины аппаратных затрат в эквивалентном каноническом МПА к величине аппаратных затрат исследуемой структуре МПА;
- экспериментальные исследования проведены с помощью VHDL-моделирования; в качестве среды моделирования использован САПР Xilinx Webpack v. 9.2;
- в качестве элементного базиса выбраны ПЛИС FPGA компании Xilinx; в качестве единицы измерения аппаратных затрат – таблицы поиска (LUT-элементы), являющиеся регулярными функциональными узлами данного класса ПЛИС;
- затраты аппаратуры в логической схеме МПА есть сумма затрат в каждом блоке структуры;
- для блоков, которые могут быть синтезированы в базисе запоминающих устройств, допускается реализация в базисе блочной памяти ПЛИС FPGA;

– аппаратурные затраты в блоке СФМО для обеих структур приняты одинаковыми.

Исследования проведены отдельно для автоматов Мили и Мура. Для каждого типа автоматов рассмотрены четыре класса сложности, определяемые значениями параметров  $M$ ,  $R$  и  $B$ : автоматы малой сложности ( $M=100$ ,  $R=7$ ,  $B=200$ ), средней сложности ( $M=500$ ,  $R=9$ ,  $B=1000$ ), большой сложности ( $M=1000$ ,  $R=10$ ,  $B=2000$ ) и сверхбольшой сложности ( $M=2000$ ,  $R=11$ ,  $B=4000$ ).

## 5 РЕЗУЛЬТАТЫ

В табл. 1 приведены результаты исследований, отражающих зависимость эффективности МПА Мура со структурой, изображенной на рис. 5, от ряда параметров автомата [10].

Содержимое таблицы представляет собой выборку наборов значений параметров, при которых значение эффективности оказывается не ниже 1,10, что соответствует превышению аппаратурных затрат в каноническом МПА на 10% по сравнению с затратами в исследуемой структуре. Таким образом, для всех наборов значений параметров, приведенных в табл. 1, структура МПА, изображенная на рис. 5, по критерию аппаратурных затрат является эффективной, т.е. более предпочтительной, чем каноническая структура.

Анализ табл. 1 позволяет сделать следующие выводы:

– с увеличением сложности автомата возрастает количество микроопераций, при которых исследуемая структура остается эффективной (столбец  $N$ );

– в автоматах более высокой сложности допустимо использование большего количества операций переходов (столбец  $N_I$ );

– исследуемая структура остается эффективной в том случае, если после использования известных методов минимизации результирующие затраты аппаратуры в структурных блоках, синтезируемых по системе булевых уравнений, составляют около 90% от исходных (столбец  $k_1$ );

– обязательным условием эффективности исследуемой структуры является возможность операционной реализации не менее 90% автоматных переходов (столбец  $k_2$ );

– для исследуемой структуры относительная стоимость базиса блочной памяти ПЛИС FPGA по отношению к базису LUT-элементов составляет 0,1 (столбец  $k_3$ ), что говорит о целесообразности использования данного базиса при синтезе блоков с малым числом входных сигналов, которые могут быть реализованы в базисе запоминающих устройств.

Сделанные выводы позволяют оценить область эффективного применения структуры МПА, изображенной на рис. 5, а также могут быть учтены при разработке формализованных методов синтеза данной структуры.

## 6 ОБСУЖДЕНИЕ

Сделанные в работе обобщения для автомата на счетчике могут быть использованы при разработке новых структур и методов синтеза микропрограммных автоматов, направленных на оптимизацию аппаратурных затрат. Каждое из сделанных обобщений является самостоятельным и не связано с другими, вследствие чего проектируемый автомат может базироваться на одном или нескольких обобщениях. В общем случае логический синтез автомата сводится к формированию системы изоморфизмов (5), структурный синтез – к построению логической схемы автомата со структурой, изображенной на рис. 5.

Отметим, что результаты, полученные в данной работе (обобщения и модификации математической и структурной моделей), относятся к функции переходов автомата и не затрагивают функцию выходов, реализуемую схемой формирования микроопераций СФМО.

## 7 БЛАГОДАРНОСТИ

Исследования выполнены в соответствии с приоритетным направлением развития науки и техники в Украине до 2020 года «Информационные и коммуникационные технологии» и содержат некоторые результаты НИР «Методологические аспекты синтеза микропрограммного автомата с операционным автоматом переходов» (номер государственной регистрации 0117U004098), финансируемой за счет средств исполнителей.

## ВЫВОДЫ

В настоящей работе для микропрограммного автомата на счетчике сделан ряд обобщений в отношении

Таблица 1 – Результаты экспериментальных исследований МПА Мура

Сложность МПА	Параметры					$E$
	$N$	$N_I$	$k_1$	$k_2$	$k_3$	
Малая	50	4	0,8	0,95	0,1	1,16
	30	4	0,9	0,95	0,1	1,12
	30	4	0,9	1	1	1,11
	30	4	0,8	0,95	0,1	1,10
Средняя	30	8	0,9	0,95	0,1	1,20
	100	8	0,9	0,95	0,1	1,16
	100	8	0,9	0,9	0,1	1,10
	90	16	0,9	0,97	0,1	1,10
Большая	100	16	0,9	0,95	0,1	1,13
	200	16	0,5	0,95	0,1	1,12
	100	8	0,8	0,9	0,1	1,10
	100	32	0,8	0,97	0,1	1,10
Сверх-большая	200	16	0,8	0,95	0,1	1,15
	100	32	0,8	0,95	0,1	1,10
	100	16	0,8	0,91	0,1	1,10
	300	8	0,85	0,9	0,1	1,10

возможных модификаций его математической модели и структуры. Показано, что в общем случае для преобразования кодов состояний могут использоваться несколько различных операций, каждая из которых предполагает собственную интерпретацию структурных кодов состояний и входных сигналов. Допускается реализация всех переходов автомата операционным способом, а также реализация части переходов реализуется каноническим способом по системе булевых уравнений.

В процессе исследований получены обобщенная математическая модель и структурная схема микропрограммного автомата с неканонической реализацией переходов. Данные результаты могут стать основой для разработки новых структур микропрограммных автоматов и формализованных методов их синтеза, ориентированных на уменьшение аппаратных затрат в логической схеме автомата.

Проведенные экспериментальные исследования предложенной обобщенной структуры МПА показали, что ее эффективность по критерию аппаратных затрат в сравнении с каноническим МПА находится в диапазоне 1,10–1,20. Данный факт позволяет обозначить область эффективного использования данной структуры в виде наборов или диапазонов значений некоторых параметров микропрограммного автомата.

#### СПИСОК ЛІТЕРАТУРИ

1. Глушков В. М. Синтез цифровых автоматов / В. М. Глушков. – М. : Физматгиз, 1962. – 476 с.
2. Баранов С. И. Синтез микропрограммных автоматов / С. И. Баранов. – Л. : Энергия, 1979. – 232 с.
3. Баркалов А. А. Синтез устройств управления на программируемых логических устройствах / А. А. Баркалов. – Донецк, ДонНТУ, 2002. – 262 с.
4. DeMicheli G. Synthesis and Optimization of Digital Circuits / G. DeMicheli. – NY. : McGraw-Hill, 1994. – 576 p.
5. Бабаков Р. М. Промежуточная алгебра переходов в микропрограммном автомате // Радиоэлектроника, информатика, управление. – 2016. – № 1. – С. 64–73.
6. Мальцев А. И. Алгебраические системы / А. И. Мальцев. – М. : Наука, 1970. – 392 с.
7. Плоткин Б. И. Элементы алгебраической теории автоматов : учеб. пособие для вузов / Б. И. Плоткин, Л. Я. Гринглаз, А. А. Гварамия. – М. : Высш. шк., 1994. – 191 с.
8. Поспелов Д. А. Арифметические основы вычислительных машин дискретного действия. Учебное пособие для вузов / Д. А. Поспелов. – М. : Высш. школа, 1970. – 308 с.
9. Майоров С. А. Структура электронных вычислительных машин / С. А. Майоров, Г. И. Новиков. – Л. : Машиностроение, 1979. – 384 с.
10. Babakov R. Research of Efficiency of Microprogram Final-State Machine with Datapath of Transitions / Roman Babakov, Alexander Barkalov, Larysa Titarenko // Proceedings of 14<sup>th</sup> International Conference “The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)”. – February 21–25, 2017. – Polyna, Ukraine. – P. 203–206.

Статья поступила в редакцию 02.08.2017.  
После доработки 25.09.2017.

Бабаков Р. М.

Канд. техн. наук, доцент, доцент кафедры прикладной механики та комп'ютерних технологій Донецького національного університету, м. Вінниця, Україна

#### УЗАГАЛЬНЕННЯ МАТЕМАТИЧНОЇ МОДЕЛІ МІКРОПРОГРАМНОГО АВТОМАТА НА ЛІЧИЛЬНИКУ

**Актуальність.** Вирішено задачу розробки структурних модифікацій мікропрограмного автомата на лічильнику, які можуть сприяти зниженню апаратних витрат в логічній схемі мікропрограмного автомата у порівнянні з відомими структурами.

**Мета роботи** – узагальнення структурних особливостей мікропрограмного автомата на лічильнику з використанням математичної моделі на основі проміжної алгебри переходів.

**Метод.** Проаналізовано відому математичну модель мікропрограмного автомата на лічильнику, засновану на представленні функції переходів у вигляді двох часткових функцій. Використання в структурі автомата інкрементного лічильника виражається в даній моделі проміжною алгеброю переходів, сигнатура якої утворена єдиною функцією інкременту. При цьому аргументом функції є код поточного стану автомата, який інтерпретується як ціле число без знака. Для розглянутої математичної моделі зроблений ряд узагальнень в частині кількості проміжних алгебр переходів, їх сигнатур та носіїв. Проаналізовані зміни в математичній моделі та структурі автомата на лічильнику, які є наслідком зроблених узагальнень.

**Результати.** На підставі зроблених узагальнень отримані узагальнені структурна схема та математична модель мікропрограмного автомата з неканонічним способом реалізації функції переходів. Проведені експериментальні дослідження ефективності розробленої узагальненої структури МПА за критерієм апаратних витрат.

**Висновки.** Отримані в даній роботі результати можуть бути використані при розробці нових структур і формальних методів синтезу мікропрограмних автоматов з неканонічною реалізацією функції переходів, орієнтованих на оптимізацію апаратних витрат в логічній схемі автомата.

**Ключові слова:** мікропрограмний автомат на лічильнику, математична модель, проміжна алгебра переходів, апаратні витрати.

Babakov R. M.

PhD, Associate professor of the Department of Applied Mechanics and Computer Technologies of the Donetsk national university, Vinnytsya, Ukraine

#### GENERALIZING OF THE MATHEMATICAL MODEL OF MICROPROGRAM FINITE STATE MACHINE ON COUNTER

**Context.** The problem of developing of structural modifications of the microprogram finite state machine on the counter has been solved, which can help reduce hardware expenses in the logical circuit of the microprogram finite state machine in comparison with the known structures.

**Objective.** The goal of the work is to generalize the structural features of the microprogram finite state machine on the counter using a mathematical model based on the intermediate algebra of transitions.

**Method.** The known mathematical model of a microprogram finite state machine on a counter, based on the representation of the transition function in the form of two partial functions, is analyzed. The use of an incremental counter in the structure of the finite state

machine is expressed in this model by an intermediate algebra of transitions whose signature is formed by a single incremental function. In this case, the argument of the function is the code of the current state of the finite state machine, interpreted as an unsigned integer. For the considered mathematical model, a number of generalizations are made regarding the number of intermediate algebras of transitions, their signatures and carriers. Changes in the mathematical model and the structure of the finite-state machine on the counter, which are a consequence of the generalizations made, are analyzed.

**Results.** On the basis of the generalizations made, a generalized structural scheme and a mathematical model of a microprogram finite-state machine with a noncanonical way of realizing the transition function are obtained. Experimental research of the effectiveness of the developed generalized structure of MPA on the criterion of hardware costs has been carried out.

**Conclusions.** The results obtained in this paper can be used in the development of new structures and formal methods for the synthesis of microprogram finite state machines with noncanonical realization of the transition function, oriented to optimizing the hardware expenses in the logical circuit of the automaton.

**Keywords:** microprogram finite state machine on counter, mathematical model, intermediate algebra of transitions, hardware expenses.

## REFERENCES

1. Glushkov V. M. Sintez tsifrovih avtomatov. Moscow, Fizmatgiz, 1962, 476 p.
2. Baranov S.I. Sintez mikroprogramnih avtomatov. Leningrad, Energiya, 1979, 232 p.
3. Barkalov A. A., Palagin A. V. Sintez mikroprogrammnih ustroystv upravleniya. Kiev, Institut kibernetiki NAN Ukraini, 1997, 135 p.
4. Barkalov A.A. Sintez ustroystv upravleniya na programmuemih logicheskikh ustroystvah. Donetsk, DonNTU, 2002, 262 p.
5. Babakov R. M. Promezhutochnaya algebra perehodov v mikroprogrammnom avtomate, *Radio Electronics, Computer Science, Control*, 2016, No. 1, pp. 64–73.
6. Maltsev A.I. Algebraicheskie sistemi. Moscow, Nauka, 1970, 392 p.
7. Plotkin B. I., Gringlaz L. Ya., Gvaramiya A. A. Elementi algebraicheskoy teorii avtomatov: Ucheb. posobie dlya vuzov. Moscow, Vish. Shk., 1994, 191 p.
8. Pospelov D. A. Arifmeticheskie osnovi vichislitelnih mashin diskretnogo deistviya. Uchebnoe posobie dlya vtuzov. Moscow, Vish. shkola, 1970, 308 p.
9. Majorov S. A., Novikov G. I. Struktura elektronnih vichislitelnih mashin. Leningrad, Mashinostroenie, 1979, 384 p.
10. Babakov R., Barkalov A.I., Titarenko L. Research of Efficiency of Microprogram Final-State Machine with Datapath of Transitions, *Proceedings of 14<sup>th</sup> International Conference "The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM)*, February 21–25, 2017. Polyana, Ukraine, pp. 203–206.