

7. Аврамова О.Д., Зуева С.Е., Наумова Т.А. и др. Автоматизированная информационная система «Учебный план». – М.: Изд-во МГУ, 2006. – 83 с.
8. Аврамова О.Д., Болотова И.Н., Владимиров А.М. и др. Автоматизированная информационная система «Педагогическая нагрузка». – М.: Изд-во МГУ, 2007. – 48 с.
9. Язык XML – Описание технологии. [Электронный ресурс]. – Режим доступа: <http://www.codenet.ru/webmast/xml/part2.php>.
10. Старых В.А., Дунаев С.Б., Коровкин С.Д. Спецификация и форматы обмена данными в разнородных информационных системах на базе XML-технологий. [Электронный ресурс]. – Режим доступа: <http://www.citforum.ru/internet/xml/xmltech/>.

Статью рекомендовал к опубликованию д.т.н., профессор Н.И. Витиска.

**Мунтян Евгения Ростиславна** – Южный федеральный университет; e-mail: [evgenia\\_muntyan@mail.ru](mailto:evgenia_muntyan@mail.ru); 347928, г. Таганрог, пер. Некрасовский, 44; тел.: 88634371550; кафедра вычислительной техники; старший преподаватель.

**Поленов Максим Юрьевич** – e-mail: [polenov@tgn.sfedu.ru](mailto:polenov@tgn.sfedu.ru); тел.: 88634371550; кафедра вычислительной техники; к.т.н.; доцент.

**Костюк Андрей Иванович** – e-mail: [a\\_kostyk@mail.ru](mailto:a_kostyk@mail.ru); тел.: 88634371608; кафедра вычислительной техники; к.т.н., доцент.

**Muntyan Evgenia Rostislavna** – Southern Federal University; e-mail: [evgenia\\_muntyan@mail.ru](mailto:evgenia_muntyan@mail.ru); 44, Nekrasovskiy, Taganrog, 347928, Russia; phone: +78634371550; the department of computer engineering; senior lecturer.

**Polenov Maxim Yuryevich** – e-mail: [polenov@tgn.sfedu.ru](mailto:polenov@tgn.sfedu.ru); phone: +78634371550; the department of computer engineering; cand. of eng. sc.; associate professor.

**Kostyuk Andrey Ivanovich** – e-mail: [a\\_kostyk@mail.ru](mailto:a_kostyk@mail.ru); phone: +78634371608; the department of computer engineering; cand. of eng. sc.; associate professor.

УДК 004.382.2

Д.Н. Ильченко

## ПРЕОБРАЗОВАНИЯ СТРУКТУРЫ АВТОМАТОВ ПОИСКА ШАБЛОНОВ С МАСКАМИ\*

*При реализации автомата поиска шаблонов с масками необходимо проводить оптимизацию его логической структуры для эффективной реализации в программируемых логических интегральных схемах (ПЛИС). В некоторых случаях проведение оптимизации приводит к нарушению корректной работы устройства, что связано с ошибочными находжениями шаблонов во входных данных, а также с пропусками шаблонов в связи с использованием масок. Необходимы дополнительные преобразования структуры цифровых автоматов поиска и правила, обеспечивающие корректную работу с масками. Для решения поставленной задачи при объединении группы автоматов поиска в общий автомат с помощью минимизации эквивалентных состояний в структуру автомата вводятся состояния инициализации шаблонов и устройство управления инициализацией. Это позволяет однозначно определять, поиск какого шаблона осуществляется в данный момент, и формировать корректные переходы из общих состояний в состояния, соответствующие конкретному шаблону. Для обеспечения работоспособности автомата при использовании масок в*

---

\* Исследования выполнены при финансовой поддержке Министерства образования и науки РФ.

шаблонах сформулированы правила формирования функций возвратов в состояния, соответствующие маскам шаблонов. Такие преобразования обеспечивают возможность проведения комплексной оптимизации структуры автомата поиска без ограничения его функциональных возможностей.

*Плюс; цифровой автомат; поиск шаблонов; маска шаблона; оптимизация структуры автомата; эквивалентные состояния; векторизация состояний; инициализация шаблонов.*

**D.N. Ilchenko**

### **STRUCTURE TRANSFORMATIONS OF DIGITAL STATE MACHINES FOR PATTERN SEARCH WITH MASKS**

*When implementing digital state machine for searching patterns with masks it is necessary to optimize its logical structure for effective implementation in field programmable gate arrays (FPGAs). In some instances, optimization leads to disruption of the correct operation of the device. This is manifested in the mistaken finding patterns in the input and in the pass pattern when using masks. There is a need in additional structure transformation of digital machines for search patterns and rules to ensure correct work with masks. To solve this problem by combining group machines search in the overall machine by minimizing the equivalent states of the structure are introduced initialization states of the patterns and the device for control initialization. This allows you to uniquely identify what pattern is searching in this moment and generate the correct transitions from states that are common in states corresponding to a specific pattern. For correct work of the search machine when using masks in patterns defined rules for the return functions to the states, which correspond to the masks of patterns. Such transformations allow for a complex optimization of the searching machine structure without limiting its functionality.*

*Fpga; digital state machine; pattern search; mask of pattern; optimization of digital state machine structure; equivalent states; vectoring states; initialization of patterns.*

В настоящее время поиск шаблонов в потоке данных в режиме реального времени является актуальной задачей. Наиболее эффективным для ее решения является использование автоматных моделей. Для повышения функциональности в шаблонах могут применяться маски. Такими масками являются метасимволы ‘\*’ и ‘?’ , где ‘?’ – соответствует любому произвольному символу, а ‘\*’ – множеству произвольных символов, однако на практике целесообразно использовать ‘\*’ с граничными значениями, например [1...n]. Применение масок усложняет структуру цифровых автоматов поиска и их синтез, поскольку каждому состоянию, соответствующему маске, необходимо определить функции прямых переходов, выходов и возвратов. Такие автоматы необходимо оптимизировать для их эффективной реализации в программируемых логических интегральных схемах (ПЛИС) [1]. Оптимизация заключается в необходимости минимизации внутренних состояний автоматов [2, 3], а также логической структуры автоматов с целью снижения количества используемых логических элементов, что связано с особенностью архитектуры последних семейств ПЛИС, где на одну логическую таблицу в ячейке реализовано два триггера [4]. При решении задачи поиска группы шаблонов с масками, где для каждого шаблона используется свой автомат, наиболее эффективными способами оптимизации являются объединения автоматов с помощью методов минимизации эквивалентных состояний [5–7] и последующее применение операции векторизации состояний [8]. Слияние эквивалентных состояний группы автоматов позволяет сформировать общий автомат поиска и снизить количество его внутренних состояний. Применение этого метода минимизации ограничено спецификой автомата-распознавателя, поэтому оно возможно только на первом этапе оптимизации при объединении автоматов. Операция векторизации является вторым этапом оптимизации и заключается в выделении группы последовательных состояний, переходы в которые осуществляются по одинаковым условиям, и объе-

динении их в вершину-массив состояний. Такие состояния соответствуют маскам шаблонов. При этом производится декомпозиция структуры автомата, формируется дополнительный автомат (счетчик), осуществляющий управление состояниями в вершине-массиве состояний. Данная операция не снижает общего количества состояний автомата, однако существенно упрощает его структуру и уменьшает количество аппаратных затрат на его реализацию [8].

Объединение группы автоматов поиска и дальнейшее применение векторизации состояний обобщенного автомата возможно только в том случае, если заданные шаблоны имеют одинаковые заголовки и разные окончания. Если же заголовки и окончания шаблонов отличаются и при этом автоматы поиска имеют эквивалентные состояния, то применение рассмотренных методов оптимизации невозможно. Таким образом, возникает необходимость в дополнительных преобразованиях структуры автомата, обеспечивающих возможность проведения подобных оптимизаций.

**1. Инициализация шаблонов.** При слиянии группы автоматов поиска шаблонов с разными заголовками и объединении всех эквивалентных состояний возникает необходимость однозначного определения переходов из общих состояний в состояния, соответствующие определенному шаблону. Неоднозначность заключается в том, что переход возможен как в состояние, соответствующее следующему символу искомого шаблона, так и в состояние, соответствующее другому шаблону, что не будет соответствовать действительности. Например, если заданы шаблоны  $P = \{P_0, P_1\} = \{ab^*e, cb^*d\}$ , то при поступлении на вход автомата последовательности  $X = \{a, b, f, j, k, d\}$  выход из общего состояния будет осуществлен в состояние, соответствующее символу "d", что будет ошибочно соответствовать нахождению шаблона  $P_1$ . Таким образом, нарушается корректная работа автомата поиска. Необходимо точно определять, при поиске какого шаблона автомат перешел в общее для группы шаблонов состояние, и в какое состояние будет возможен дальнейший переход.

Для решения этой проблемы необходимо инициализировать шаблоны. Такая процедура заключается в выделении некоторых состояний автомата, которые позволяют однозначно определять, поиск какого шаблона осуществляется в данный момент времени. Такие состояния называются состояниями инициализации и определяются как ближайшие к начальному неэквивалентному состоянию шаблонов. Для каждого заданного шаблона используется одно состояние инициализации. Информация об инициализации должна быть учтена при формировании функций переходов автомата из всех общих состояний в состояния, соответствующие определенному шаблону, что обеспечит корректную работу автомата поиска.

Для хранения информации об инициализации шаблона формируется устройство управления инициализацией, которое представляет собой автомат с хранением состояний. Количество состояний такого автомата будет определяться количеством шаблонов с разной инициализацией. Состояния в устройстве управления инициализацией будут меняться только при переходе автомата поиска в состояния инициализации шаблонов.

Устройство управления инициализацией шаблонов обеспечивает не только корректное формирование переходов на этапе минимизации эквивалентных состояний при объединении группы автоматов, но и позволяет использовать одно общее устройство управления вершиной-массивом при проведении векторизации состояний в том случае, если множество заданных шаблонов имеет маски разного размера.

Применение устройства управления инициализацией на этапе минимизации эквивалентных состояний автоматов увеличивает количество используемого аппаратного ресурса. Количество триггеров  $N$  на его реализацию можно оценить по формуле

$$N = \lceil \log_2 I \rceil,$$

где  $I$  – количество шаблонов с разной инициализацией.

Количество логических элементов  $L$  будет определяться по формуле

$$L \approx 2 \cdot I.$$

Если все  $I$  шаблонов будут иметь разное количество символов в масках, то на этапе применения векторизации состояний количество логических элементов для реализации устройства управления состояниями в вершине-массиве увеличится на  $I$  элементов, что связано с необходимостью формирования дополнительных выходных функций.

Рассмотрим пример минимизации автоматов поиска шаблонов  $P = \{P_0, P_1\} = \{a^*e, c^*d\}$ , где  $*[1..3]$ ,  $\{ "a", "c" \} \notin *$ . Исключение символов инициализации шаблонов является необходимым ограничением для обеспечения возможности проведения оптимизации автоматов поиска шаблонов с масками. Графы автоматов для поиска шаблонов  $P_0$  и  $P_1$  представлены на рис. 1, для простоты указаны только прямые переходы.

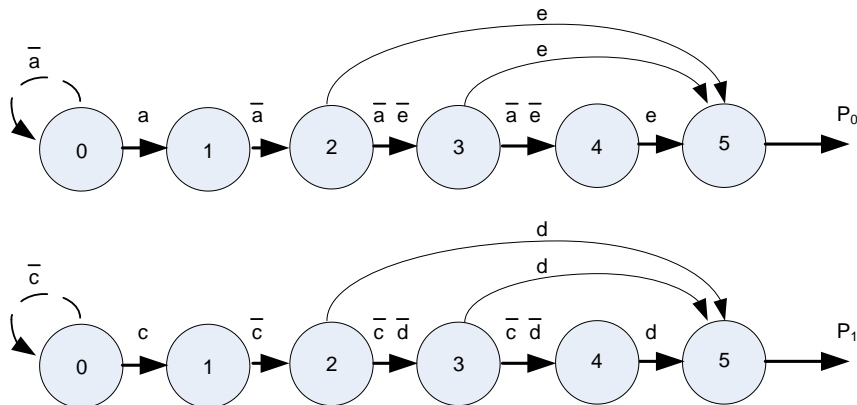


Рис. 1. Графы автоматов поиска шаблонов

В данном примере состояния 2, 3, 4 каждого автомата соответствуют маскам и являются эквивалентными. Общий автомат поиска шаблонов  $P_0, P_1$  после объединения эквивалентных состояний представлен на рис. 2. Состояния 5-1, 5-2 являются конечными состояниями. Очевидно, что после перехода в состояние 2 возникает неоднозначность дальнейшего поиска шаблонов, т.е. теряется информация о том, из какого состояния был осуществлен переход в это состояние. Последующий выход возможен как по входному символу “e”, так и по символу “d”. Но после перехода в состояние 2 из состояния 1-1, что соответствует возможному нахождению шаблона  $P_0$ , и поступления на вход автомата воздействия “d”, автомат перейдет в состояние 5-2. Такой переход будет соответствовать найденному шаблону  $P_1$ , что некорректно.

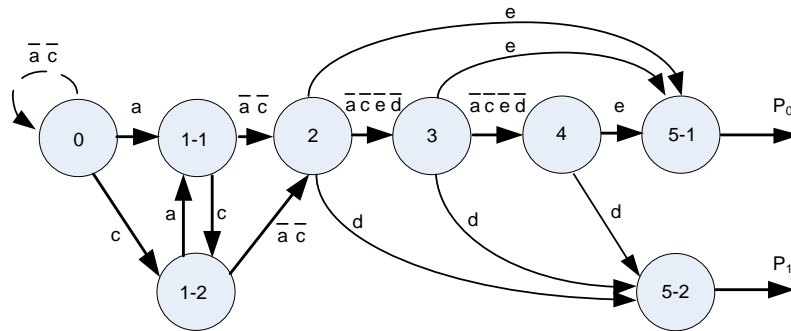


Рис. 2. Граф автомата после минимизации эквивалентных состояний

Автомат после формирования устройства управления и применения операции векторизации состояний представлен на рис. 3. Вершина 2-4 является вершиной-массивом состояний, в которую входят состояния 2,3,4, Counter – устройство управления состояниями в этой вершине, начало работы которого определяется функцией Z, а R – выходная функция устройства управления, формирующаяся при переходе устройства управления в состояние, соответствующее последнему состоянию в вершине-массиве, т.е. последнему символу маски.

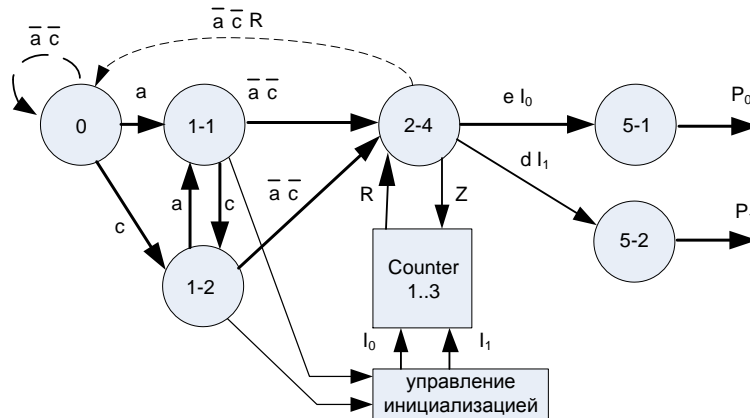


Рис. 3. Формирование устройства управления инициализацией и применение векторизации состояний

Состояния 1-1 и 1-2 являются состояниями инициализации шаблонов. При переходе автомата в состояние 1-1 в блоке управления инициализацией формируется выходная функция  $I_0$ , при переходе в состояние 1-2 – выходная функция  $I_1$ . Функции  $I_0, I_1$  позволяют определять, поиск какого шаблона осуществляется в данный момент времени  $P_0$  либо  $P_1$ . Информация об инициализации используется при формировании переходов из вершины-массива состояний 2-4 в конечные состояния 5-1 и 5-2, что позволяет решить проблему с некорректным нахождением шаблонов.

Инициализация шаблонов обеспечивает возможность проведения дальнейшей минимизации структуры автомата. Такая минимизация заключается в объединении изоморфных подграфов после общих состояний автомата, где каждый подграф является окончанием отдельного шаблона. В этом случае все переходы между состоя-

ниями после объединения должны быть сформированы с учетом информации об инициализации. Такое преобразование структуры компенсирует аппаратные затраты на формирование устройства управления инициализацией и приводит к дополнительной минимизации аппаратного ресурса на реализацию автомата.

Таким образом, управление инициализацией позволяет проводить минимизацию автоматов поиска шаблонов с масками, сохраняя при этом их корректную работу без ограничения функциональных возможностей.

**2. Формирование функций возвратов автомата поиска при использовании масок в шаблонах.** Если шаблоны после маски содержат один символ, являющийся последним, то формирование функций переходов и возвратов такого автомата поиска не вызывает сложностей и неопределенностей при минимизации автомата с помощью векторизации состояний. Автомат, находясь в любом состоянии в вершине-массиве, при входном воздействии, соответствующем переходу в конечное состояние, однозначно определяет нахождение шаблона, в противном случае при переходе в конечное состояние в вершине-массиве и дальнейшей неудаче поиска автомат осуществляет переход в начальное состояние (рис. 3). Однако если после маски в шаблоне следует некоторая комбинация символов, приводящая к конечному состоянию автомата, то в случае функции неудачи пропадает однозначность в определении состояния, в которое автомат должен вернуться. Возврат может быть осуществлен в начальное состояние, в состоянии начальной инициализации, в одно из состояний в вершине-массиве, а также в первое состояние после вершины-массива. Таким образом, возникает необходимость определения правил формирования функций возвратов, которые обеспечат корректную работу автоматов поиска шаблонов с масками при их оптимизации.

Для того чтобы отслеживать все возможные вхождения символов в маску при выходе из нее и дальнейшем попадании в функцию неудачи, устройство управления состояниями в вершине-массиве состояний (счетчик) должно продолжать осуществлять переход по состояниям, даже если автомат вышел из вершины-массива состояний, имитируя при этом возможное попадание входного символа в маску шаблона. Начальная инициализация счетчика должна осуществляться либо после перехода автомата в начальное состояние, либо после выполнения в счетчике выходной функции  $R$ , определяющей выход из последнего состояния в вершине-массиве и соответствующей шаблону, поиск которого осуществляется в настоящий момент.

Чтобы обеспечить корректную работу автомата после неудачи поиска функции возвратов должны быть определены для  $m-1$  состояния после вершины-массива с учетом функции выхода  $R$  счетчика. В данном случае  $m$  – максимальное количество символов в маске соответствующего шаблона, что определяет количество состояний в вершине-массиве. Функции возвратов для состояний, начиная с  $m$ -состояния, определяются однозначно без анализа функции выхода счетчика, поскольку такие состояния уже не попадают в маску шаблона.

С учетом функции выхода  $R$  устройства управления для  $m-1$  состояния ПОСЛЕ вершины-массива можно определить следующие правила возвратов:

- ◆ если  $R=0$ , то автомат переходит в вершину-массив из любого  $m-2$  состояния. Для  $m-1$  состояния возврат в вершину-массив невозможен;
- ◆ если  $R=1$  и входное воздействие на автомат соответствует переходу в первое состояние после вершины-массива состояний, то осуществляется возврат автомата в это состояние, иначе – в начальное состояние автомата, либо в состояние начальной инициализации шаблона в зависимости от входного воздействия на автомат;

- ♦ если автомат находится в первом состоянии после вершины-массива и на автомат подается последовательность одинаковых входных воздействий, переводящих автомат в это состояние, то автомат остается в этом состоянии до выполнения функции возврата устройства управления в исходное состояние  $R_{reset}$ . Функция  $R_{reset}$  выполняется на следующем шаге цикла работы устройства управления после формирования выходной функции  $R$ .

Данные правила обеспечивают корректное нахождение всех попаданий шаблона во входной последовательности при проведении минимизации структуры автоматов поиска.

Пусть заданы шаблоны  $P = \{P_0, P_1\} = \{a^*bcde, m^*ghij\}$ , где  $*[1..3]$ ,  $\{a, m\} \notin *$ . Графы автоматов для их поиска представлены на рис. 4. Для упрощения указаны не все функции возвратов.

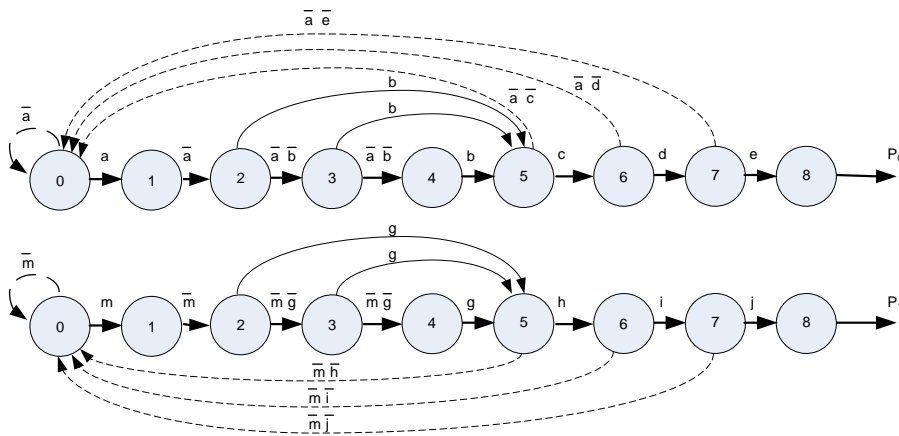


Рис. 4. Графы автоматов поиска шаблонов  $P = \{a^*bcde, m^*ghij\}$

Состояния 2, 3, 4 автоматов являются эквивалентными и соответствуют маскам шаблонов. Минимизировав эквивалентные состояния, получим общий автомат поиска шаблонов. Проведем операцию векторизации состояний 2,3,4 и объединим их в вершину-массив состояний, сформировав при этом устройство управления состояниями в вершине-массиве. После объединения автоматов состояния 1 заменим состояниями 1-1 и 1-2, которые являются состояниями инициализации шаблонов, а состояния 5, 6, 7, 8 автомата поиска шаблона  $P_1$  обозначим как 9, 10, 11, 12. После использования устройства управления инициализацией и применения правил формирования функций возвратов из состояний после вершины-массива граф автомата поиска шаблонов представлен на рис. 5.

Пунктирными линиями на рис. 5 обозначены основные функции возвратов, которые возможны после выхода из последнего состояния в вершине-массиве состояний в следующее состояние после вершины-массива при дальнейшей функции неудачи поиска. Поскольку максимальное количество символов в маске шаблона равно трем, то дополнительные правила формирования функций возвратов должны быть применены для двух состояний после вершины-массива для каждого шаблона, в данном случае – для состояний 5, 6 и 9, 10. Устройство управления инициализацией позволяет однозначно определять переходы из общей вершины-массива состояний 2-4 в состояния, соответствующие шаблонам  $P_0$  и  $P_1$ , т.е. состояния 5 и 6. При этом информация об инициализации используется при форми-

ровании выходных функций устройства управления вершиной-массивом. В данном примере используются две выходные функции  $R_0$  и  $R_1$ , но поскольку размерность масок шаблонов одинакова, достаточно одной общей выходной функции  $R$ .

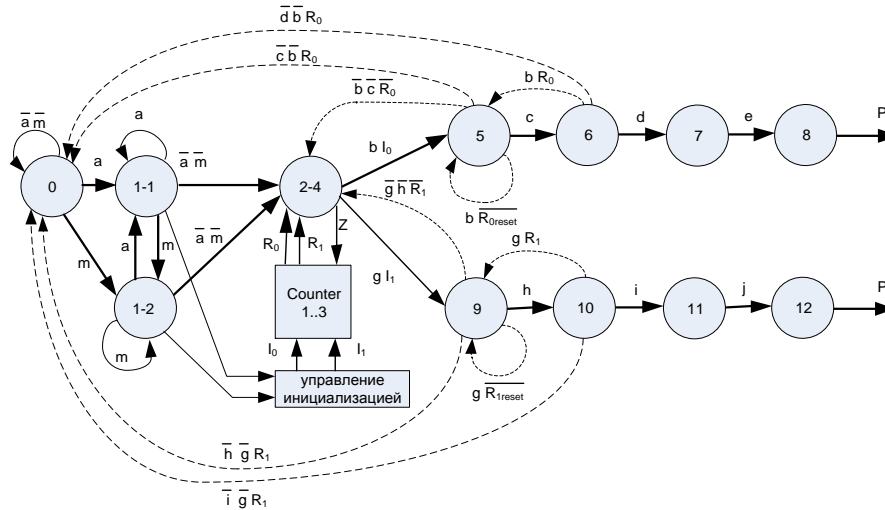


Рис. 5. Граф автомата поиска шаблонов после объединения эквивалентных состояний и операции векторизации состояний

На рис. 5 подграфы автомата, содержащие состояния 5, 6, 7, 8 и 9, 10, 11, 12, являются изоморфными, что позволяет при дальнейшей оптимизации объединить их в один подграф, содержащий, например, состояния 5, 6, 7, 8, где переходы в каждое состояние будут объединены и сформированы с учетом информации об инициализации.

Комплексная минимизация, проведение которой возможно благодаря рассмотренным методам и правилам, обеспечивает в лучшем случае до 90 % сокращения аппаратного ресурса, в среднем – до 30 %, что определяется структурой заданных шаблонов.

**Заключение.** Инициализация шаблонов и применение правил формирования функций возвратов обеспечивают возможность оптимизации логической структуры автоматов поиска с целью их эффективной реализации в ПЛИС. Инициализация шаблонов приводит к незначительному увеличению используемого аппаратного ресурса, однако при этом сохраняется корректная работа автомата при поиске группы шаблонов с разными заголовками после минимизации эквивалентных состояний и формировании общего автомата поиска. В ряде случаев объединение изоморфных подграфов при инициализации шаблонов компенсирует аппаратные затраты на реализацию устройства управления инициализацией.

Дополнительные правила формирования возвратов при работе с масками не увеличивают количество аппаратных ресурсов на их применение, но позволяют находить все попадания шаблонов во входную последовательность при использовании масок и обеспечивают возможность минимизации логической структуры автомата поиска с помощью операции векторизации состояний.

Таким образом, разработанный метод и правила могут быть эффективно применены при реализации устройств сигнатурного анализа данных в режиме реального времени на ПЛИС, например аппаратных антивирусов, устройств обнаружения вторжений, устройств предотвращения вторжений, сетевых экранов и т.п.



## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Каляев А.В., Левин И.И.* Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. – М.: Янус-К, 2003. – 380 с.
2. *Глушков В.М.* Синтез цифровых автоматов. – М.: Физматгиз: 1962. – 476 с.
3. *Поликарпова Н.И., Шалыто А.А.* Автоматное программирование. – СПб.: СПбГПУ, 2008. – 227 с.
4. [www.xilinx.com/support/documentation/user\\_guides/ug364.pdf](http://www.xilinx.com/support/documentation/user_guides/ug364.pdf) (дата обращения: 17.02.2014).
5. *Хопкрофт Д., Мотвани Р., Ульман Дж.* Введение в теорию автоматов, языков и вычислений: Пер. с англ. – 2-е изд. – М.: Вильямс, 2002. – 528 с.
6. *Мельников Б.Ф., Мельникова А.А.* Многоаспектная минимизация недетерминированных конечных автоматов (Ч. I. Вспомогательные факты и алгоритмы) // Изв. вузов. Поволжский регион. Физико-математические науки. – 2011. – № 4. – С. 59-69.
7. *Мельников Б.Ф., Сайфуллина М.Р.* О некоторых алгоритмах эквивалентного преобразования недетерминированных конечных автоматов // Изв. вузов. Математика. – 2009. – № 4. – С. 67-71.
8. *Ильченко Д.Н.* Применение операции векторизации состояний для синтеза цифровых автоматов [Электронный ресурс] // Инженерный вестник Дона. – 2013. – № 4 – Режим доступа: <http://www.ivdon.ru/magazine/archive/n4y2013/2028> (доступ свободный) – Загл. с экрана. – Яз. рус.

Статью рекомендовал к опубликованию к.т.н. Е.А. Семерников.

**Ильченко Дмитрий Николаевич** – Научно-исследовательский институт многопроцессорных вычислительных систем им. акад. А.В. Каляева Южного федерального университета; e-mail: [dimas\\_doct@mail.ru](mailto:dimas_doct@mail.ru); 347900, г. Таганрог, ул. Транспортная, 11а, кв. 9; тел.: 89043470674; младший научный сотрудник.

**Pchenko Dmitry Nirkolaevich** – Scientific Research Institute of Multiprocessor Computer Systems named after Acad. A.V. Kalyaev of Southern Federal University; e-mail: [dimas\\_doct@mail.ru](mailto:dimas_doct@mail.ru); 11a – 9, Transportnaya street, Taganrog, 347900, Russia; phone: +79043470674; researcher.

УДК 004.4:004.9

**С.А. Кучеров**

**МЕТОД КОНФИГУРИРОВАНИЯ ДИНАМИЧЕСКИХ БАЗ ДАННЫХ\***

*При создании конфигурируемых информационных систем существует задача обеспечения гибкости баз данных. Для ее решения применяются базы данных с динамической структурой. В качестве исходного материала динамические базы данных предоставляют: множество характеристик, в терминах которых описывается структура данных. Примером баз данных с динамической структурой являются структурно-независимые базы данных. Задача статьи – предложить метод перехода от динамических баз данных к статической структуре, то есть метод конфигурирования структурно-независимых баз данных. Описана исходная динамическая база данных на примере структурно-независимой базы данных, сформулирован метод в виде правил конфигурирования структурно-независимой базы данных и приведен пример механизма реализации правил конфигурирования структурно-независимых баз данных. Наиболее удобным средством представления правил, по мнению авторов, является совмещение графического и текстового описания. В качестве графического представления используется язык блок-схем. Текстовым представлением правила является модифицированная формулировка механизма.*

*Структурно-независимая база данных; конфигурируемая информационная система.*

\* Исследование выполнено при поддержке государственного задания 0110021005901621. Тема № 213.01-11/2014-17.