

Тущ Є.В.

Тернопільський національний технічний університет імені Івана Пулюя

УЗАГАЛЬНЕНИЙ АЛГОРИТМ СИНТЕЗУ КОМПОНЕНТІВ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ МІКРОПРОГРАМНИХ АВТОМАТІВ

Синтез компонентів (логічних блоків) є невід'ємною частиною розробки комп'ютерних систем, оскільки саме на цьому етапі забезпечується створення функціональних вузлів, які спільно реалізують поставлені завдання системи. В загальному цей процес охоплює не лише побудову окремих блоків, але й їх оптимальну інтеграцію в загальну архітектуру, що сприяє підвищенню продуктивності та надійності. Завдяки синтезу забезпечується узгоджена взаємодія всіх компонентів системи, що є основою для досягнення її ефективного функціонування.

В статті обґрунтовано узагальнений алгоритм синтезу компонентів комп'ютерних систем на основі мікропрограмних автоматів, що містить в собі етапи абстрактного та структурного синтезу. Ґрунтовно розглянуто методику побудови керуючих автоматів з жорсткою логікою, починаючи від розробки граф-схеми алгоритму функціонування логічного блоку, створення абстрактної моделі у вигляді абстрактного автомату Мілі або автомата Мура та до представлення його у вигляді функціональної (структурної) схеми. Досліджено принципи побудови керуючих автоматів з програмованою логікою, що включає в себе вибір типу адресації та спосіб кодування мікрокоманд, написання мікропрограми в машинних мікрокодах та вибір елементів функціональної (структурної) схеми. Окремо описано алгоритм структурного синтезу операційних автоматів, що полягає у розробці граф-схеми алгоритму роботи логічного блоку та представленні кожної функції даного алгоритму відповідними комбінаційними або послідовними операційними елементами у структурній схемі розроблюваного компонента. Розглянутий канонічний синтез структурного синтезу операційних автоматів дозволяє оптимально поєднувати операційні елементи різних типів в єдину структурну схему.

Таким чином, запропонований в статті алгоритм об'єднує в єдине ціле вже відомі методики, що дозволяє підвищити ефективність та швидкість процесу синтезу компонентів комп'ютерних систем. Завдяки цьому процес синтезу стає більш структурованим, зрозумілим і доступним для реалізації. Це сприяє створенню надійних, функціональних та оптимізованих систем, що відповідають сучасним вимогам.

Ключові слова: компоненти комп'ютерних систем, мікропрограмні автомати, граф-схема алгоритму, абстрактний синтез, структурний синтез, керуючий автомат, операційний автомат, функціональна схема, структурна схема.

Постановка проблеми. В сучасному світі різноманітні комп'ютерні технології входять у всі сфери нашого життя. Їх бурхливий розвиток та впровадження зумовлює необхідність розробки ефективного апаратного забезпечення, вдосконалення вже існуючих технічних рішень, знаходження нових підходів та схемотехнічних варіантів в побудові компонентів комп'ютерних систем. Нові цифрові пристрої мають забезпечувати одночасно високу продуктивність, компактність та прийнятну вартість.

В процесі синтезу комп'ютерних систем різного призначення розробники часто стикаються з проблемою схемотехнічної реалізації та вибору стандартних компонент для таких систем або створення нетипових компонентів. Хоча

значна кількість досліджень вже була зосереджена на цій проблемі, існує прогалина у впровадженні загального уніфікованого алгоритму синтезу.

Аналіз останніх досліджень і публікацій. Фундаментальними у сфері розробки комп'ютерних систем є праці М. Уїлкса та В.Н. Глушкова, в яких було викладено принципи прикладної теорії цифрових автоматів та мікропрограмного керування, обґрунтовано концепції операційного та керуючого автоматів, що є основою для побудови компонентів та функціонально-логічних схем вузлів комп'ютерних систем [1-3]. В подальшому активну роботу в цій сфері було проведено плеядою вчених, серед яких можна відзначити наступних: Г. Мілі, Е. Мур, Дж. фонНейман, Д.А. Белл,

Дж. Ворден, Г. Бетке, Дж. деМічелі, С. Баранов, А. Кевалік, а також сучасних українських науковців, серед яких [4]: В.С. Харченко, А.М. Чеботарьов, В.М. Опанасенко, В.І. Хаханов, М.В. Лобур, О. Баркалов та ін.

Постановка завдання. Метою статті є аналіз та зведення відомих методик синтезу компонентів комп'ютерних систем до єдиного загального алгоритму синтезу компонентів на основі концепції мікропрограмних операційного та керуючого автоматів, що дозволить зробити процес синтезу більш зрозумілим, ефективнішим та швидкісним у реалізації.

Виклад основного матеріалу. Сучасна функціональна та структурна організація компонентів комп'ютерних систем ґрунтується на моделі мікропрограмного пристрою, яка полягає у поєднанні операційного та керуючого автоматів, що взаємодіють між собою згідно з принципом зво-

ротного зв'язку [1-3,5]. Згідно цієї концепції, операційний автомат є структурою, що призначений для виконання дій над інформацією (зберігання слів, виконання наборів мікрооперацій, обчислення значень логічних умов). Керуючий автомат, у свою чергу, генерує послідовність керуючих сигналів, які забезпечують виконання в операційному автоматі послідовність мікрооперацій згідно алгоритму роботи цифрового пристрою.

Першим кроком у процесі розробки компонентів комп'ютерних систем (рис. 1) є створення мікропрограми, яка описує виконання операцій у синтезованому логічному блоці у вигляді алгоритму, що відомий як змістовна схема алгоритму. Надалі терміни цього алгоритму деталізуються у формі мікрооперацій та логічних умов, після чого формується граф-схема алгоритму (ГСА) [1, 5]. ГСА є скінченим орієнтованим

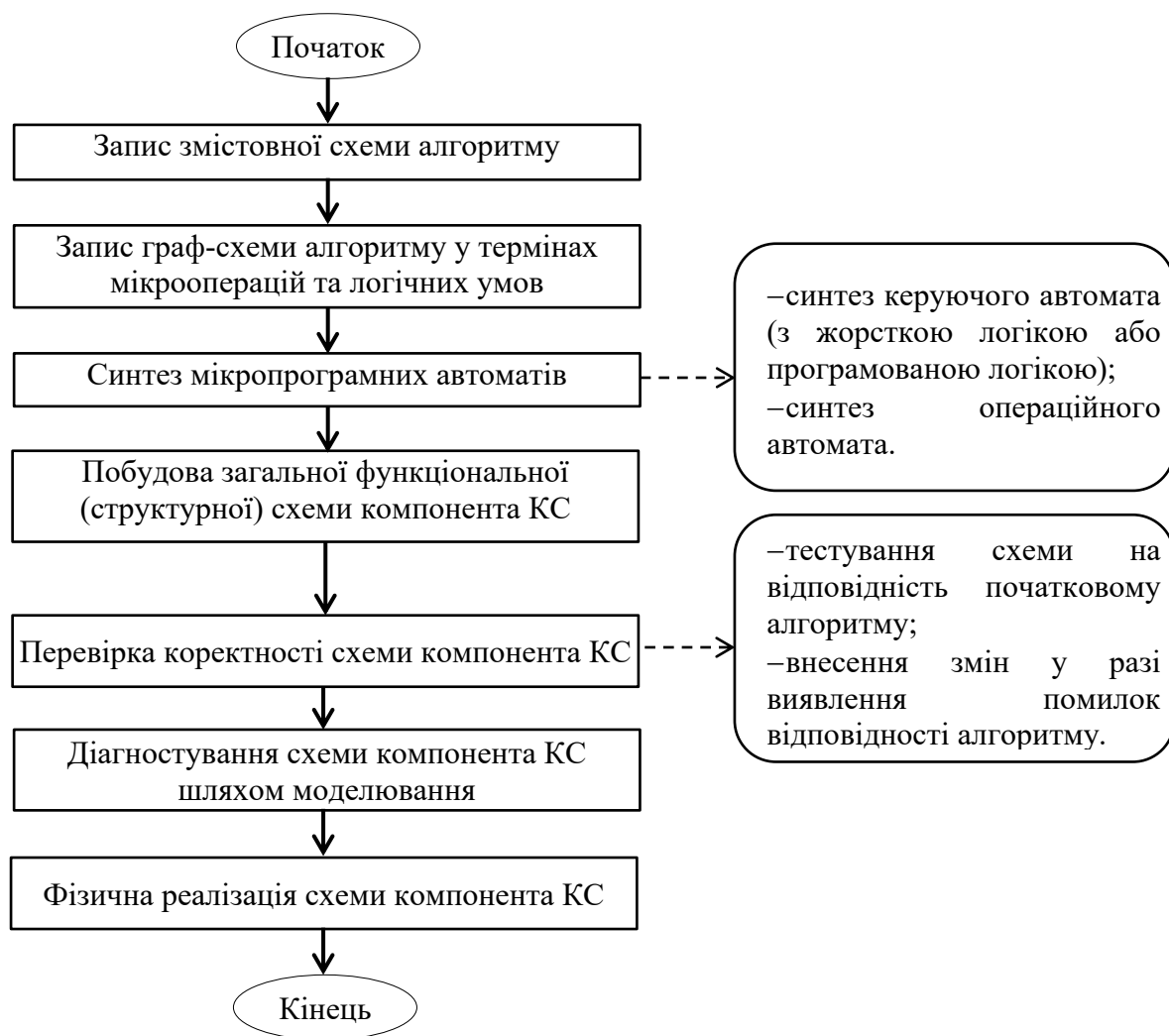


Рис. 1. Узагальнений алгоритм синтезу компонентів комп'ютерних систем

зв'язним графом, що включає вершини чотирьох типів: початкову, кінцеву, множину операторних $Y = \{y_1, \dots, y_K\}$ та умовних вершин $X = \{x_1, \dots, x_L\}$. ГСА має задовільняти коректним умовам побудови [1]. Зокрема, у операторній вершині записується мікрокоманда $Y_i \in Y$ або декілька сумісних мікрокоманд, у кожній умовній вершині записується тільки один з елементів множини логічних умов $X_s \in X$, тощо.

За розробленою ГСА в подальшому окремо проводиться синтез операційної та керуючої частини компонента комп'ютерної системи, результатом чого є отримання загальної функціональної (або структурної) схеми (див. рис. 1). Важливими етапами синтезу є верифікація отриманої схеми шляхом тестування схеми на відповідність початковому алгоритму та її діагностування з метою виявлення можливих несправностей, використовуючи програмне або апаратне моделювання [7,8].

Детально зупинимся на етапі побудови мікропрограмних автоматів.

Синтез керуючих автоматів залежить від типу організації їх пам'яті [1, 9]. До першої групи таких автоматів належать керуючі автомати з жорсткою логікою, які будуються на базі пам'яті станів, що реалізується сукупністю тригерів та формуванням вихідних (керуючих) сигналів в залежності від вхідних сигналів та поточного стану.

Алгоритм синтезу керуючого автомата з жорсткою логікою за ГСА відбувається у два етапи (рис. 2). На першому етапі – етапі абстрактного синтезу – здійснюється вибір абстрактної моделі логічного блоку та маркування ГСА. На практиці знайшли найбільшого використання два варіанти побудови: у вигляді моделі Мілі та Мура [1,2,5,9,10].

Маркування ГСА мітками $A = \{a_1, \dots, a_N\}$ здійснюється залежно від типу обраної моделі. У ГСА за моделлю Мілі маркують символом стану a_1 вихід початкової та вхід кінцевої вершин, а також всі виходи операторних вершин символами $a_2, \dots, a_N$. Згідно маркування за моделлю Мура

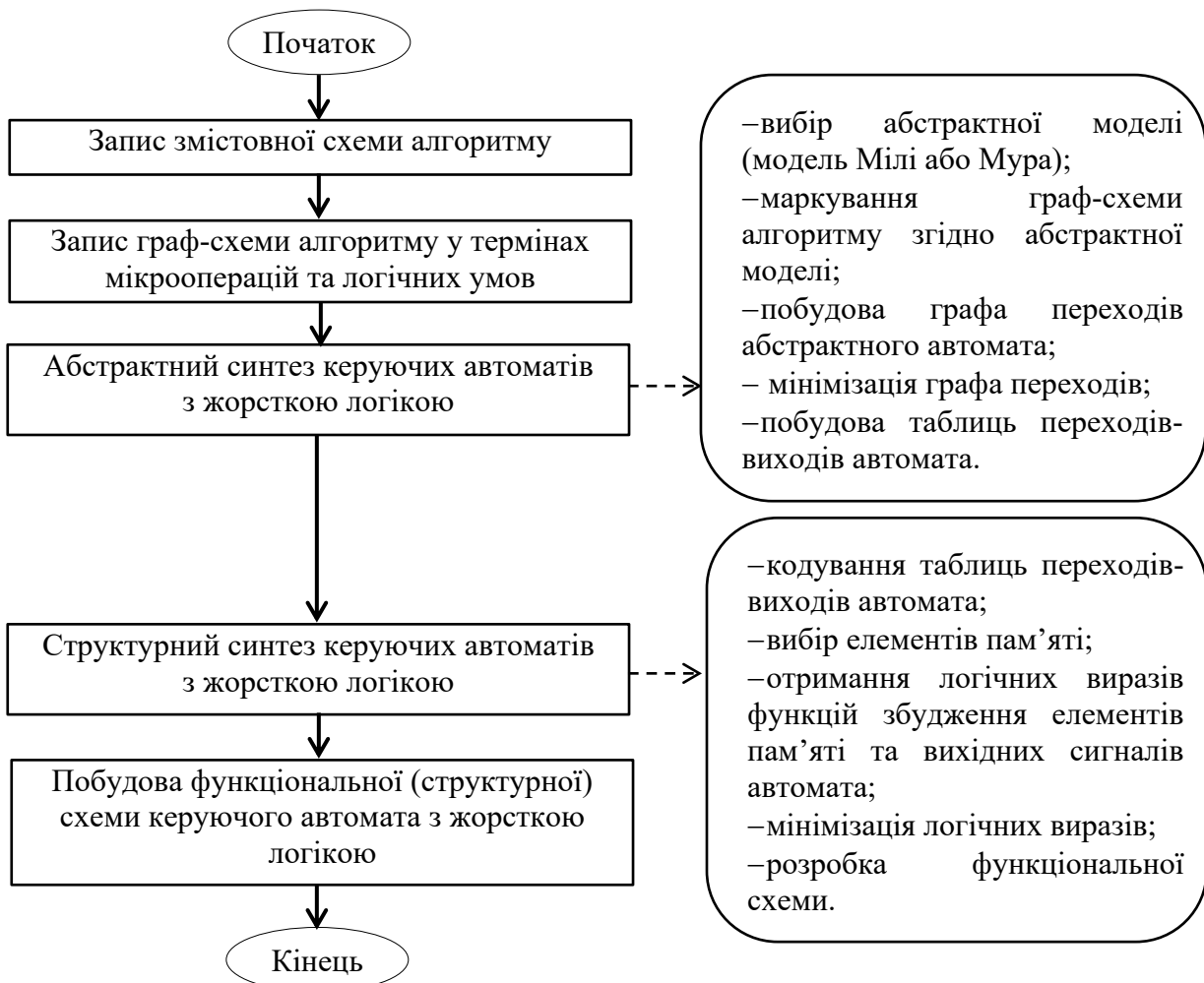


Рис. 2. Алгоритм синтезу керуючих автоматів з жорсткою логікою

позначаються символом стану a_1 вершина початок та кінець, всі операторні вершини маркуються як $a_2, \dots, a_M$. Множина A отриманих таким чином символів формує кількість вершин графа переходів абстрактного автомата. Кожному ребру такого автомата з одного стану в інший відповідає перехід по ГСА. Ребру автомата згідно моделі Мілі приписується набір логічних умов, що викликає цей перехід, а також набір керуючих сигналів, що відповідає даному переходу. При абстрактному синтезі керуючого автомата згідно моделі Мура ребру приписується лише набір логічних умов, а керуючі сигнали зазначаються біля вершин створеного графа.

Завершальним етапом абстрактного синтезу є побудова таблиць переходів-виходів абстрактного автомата, в яких зазначається відповідність між станами автомата, вхідними та вихідними сигналами.

Після етапу абстрактного синтезу слідує етап структурного синтезу, метою якого є створення логічної схеми автомата з використанням елементів заданого базису. Основною моделлю на цьому етапі є структурний автомат – пристрій, який реалізує закон функціонування абстрактного автомата [1,2,5,6].

Структурний автомат задається п'ятіркою:

$$S = \langle X, Y, Q, f, g \rangle,$$

де $X = \{x_1, \dots, x_L\}$, $x_i \in \{0,1\}$ – множина вхідних сигналів, $L \geq \log_2 Z$, Z – вхідний алфавіт абстрактного автомата; $Y = \{y_1, \dots, y_K\}$, $y_k \in \{0,1\}$ – множина вихідних сигналів, $K \geq \log_2 W$, W – вихідний алфавіт абстрактного автомата; Q – кількість елементів пам'яті структурного автомата, $Q \geq \log_2 A$, A – кількість станів абстрактного автомата; f – функція переходів; g – функція виходів.

Основним підходом до розробки функціональної схеми керуючого автомата є канонічний метод структурного синтезу, що дозволяє створювати схему структурного автомата на основі таблиць переходів-виходів чи графа абстрактного автомата. Цей метод передбачає використання структурно повної системи, яка складається з двох типів елементарних автоматів: автоматів з пам'яттю (елементів пам'яті), що мають більше одного стану, і автоматів без пам'яті – логічних елементів [1,2,5].

У результаті застосування канонічного методу із закодованих таблиць переходів-виходів абстрактного автомата отримується система булевих рівнянь. Ці рівняння описують залежність вихідних сигналів й сигналів, що подаються на входи еле-

ментів пам'яті, від вхідних сигналів автомата та сигналів, знятих із виходів елементів пам'яті.

Після мінімізації булевих рівнянь та вибору елементного базису будується функціональна схема керуючого автомату з жорсткою логікою.

Як вже зазначалося, функція будь-якого керуючого автомата полягає у створенні послідовності керуючих слів (мікрокоманд), що визначена алгоритмом, який він реалізує, з урахуванням значень інформаційних сигналів. Якщо заздалегідь записати всі необхідні мікрокоманди для виконання заданого алгоритму (або групи алгоритмів) у запам'ятовуючому пристрої та потім вибирати їх у передбаченому алгоритмом порядку, отримаємо керуючий автомат, структура якого мало залежить від реалізованих алгоритмів. Його поведінка визначається переважно вмістом пам'яті. При зміні алгоритму в межах певного діапазону структура такого автомата залишається незмінною, і для адаптації достатньо змінити вміст комірок пам'яті. Керуючий автомат, побудований за таким принципом, називається керуючим автоматом з програмованою логікою та включає в себе запам'ятовуючий пристрій мікрокоманд, регістр мікрокоманд та пристрій формування адреси мікрокоманди [1,2].

При структурному синтезі автоматів з програмованою логікою (рис. 3) спочатку проводиться вибір типу адресації мікрокоманд (з примусовою або природньою адресацією) та вибір способу кодування мікрооперацій (горизонтальний, вертикальний та змішаний способи кодування). Далі відповідно до ГСА функціонування компонента комп'ютерної системи проводиться написання мікропрограми в машинних мікрокодах, після чого формується на основі мікропрограми сама структурна схема [1,9,11,12].

Алгоритм синтезу операційних автоматів залежить від принципів їх структурної організації [1,5]. До основних структур належать операційні автомати з канонічною структурою, І-автомати, М-автомати, ІМ-автомати з паралельними та послідовними комбінаційними частинами, а також S-автомати. Методи, що застосовуються для синтезу керуючих автоматів, не підходять для операційних автоматів, оскільки вони базуються на абстрактній теорії автоматів, а кількість можливих станів операційного автомата є надзвичайно великою. Структури операційних автоматів у заданому базисі, що складається з шин, регістрів і комбінаційних схем, можуть бути синтезовані безпосередньо на основі функції, яку реалізує автомат. При цьому функцію операційного автомата можна отримати без проміжних етапів



Рис. 3. Алгоритм синтезу керуючих автоматів з програмованою логікою

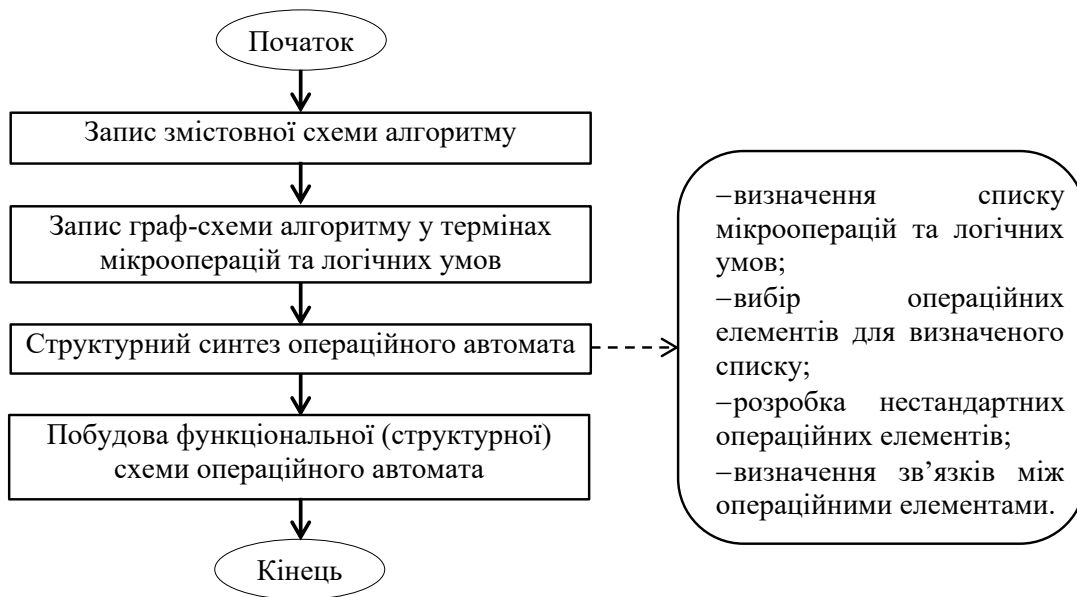


Рис. 4. Алгоритм синтезу операційних автоматів

зі змістовної граф-схеми, яка відображає логіку роботи компонента комп'ютерної системи. Таким чином, операційний автомат можна розглядати як сукупність операційних елементів, кількість яких відповідає кількості внутрішніх слів, що обробляється мікропрограмою.

Отже, структурний синтез операційного автомата (рис. 4) зводиться до [1,13,14]: представлення кожного слова регістром, що має розмірність рівну довжині цього слова; визначення списку мікрооперацій та логічних умов, яким ставляться у відповідність визначені комбінаційні схеми;

розробки нестандартних операційних елементів; визначення та технічна імплементація зв'язків між операційними елементами. Результат вирішення перелічених задач призводить до синтезу функціональної (структурної) схеми операційної частини компонента комп'ютерних систем.

Висновки. В статті розглянуто сучасні методи розробки компонентів комп'ютерних систем із їх представленням у вигляді мікропрограмних автоматів з операційною та керуючою частиною. Запропоновано узагальнений алгоритм синтезу компонентів, який включає побудову граф-схеми

алгоритму, абстрактний та структурний синтез, а також верифікацію отриманих схем. Описано особливості побудови керуючих автоматів із жорсткою логікою та програмованою логікою, а також принципи синтезу операційних автоматів.

Результати дослідження дають можливість об'єднати в єдине ціле весь процес синтезу компонентів комп'ютерних систем, що спрямовано на підвищення ефективності та швидкості їх проектування.

Список літератури:

1. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Львів: Видавництво «Магнолія-2006», 2015. 354 с.
2. Baranov S. Logic Synthesis for Control Automata. Springer; Softcover reprint of the original 1st ed. 1994. 407 p.
3. Baranov S. Logic and System Design of Digital Systems. Tallin, TUTPress. 2008. 267 p.
4. Баркалов О.О., Бабаков Р.М. Прикладна теорія цифрових автоматів в сучасній українській науці. Матеріали наукової конференції професорсько-викладацького складу, наукових працівників і здобувачів наукового ступеня за підсумками науководослідної роботи за період 2019–2020 рр. (квітень–травень 2021 р.). Вінниця: Донецький національний університет імені Василя Стуса. 2021. С. 318–320.
5. Петрушенко А.М. Принцип мікропрограмовного керування та автоматизація проектування операційних пристроїв. I. Control systems & computers. 2020. № 1. С. 3–22.
6. Barkalov A., Titarenko L. Basic principles of logic design. UZ Press, Zielona Gora. 2010.
7. Keavallik A., Kruus M., Lensen H. Microprogram Automaton: Design for Testability. Proceedings of 6th Baltic Electronic Conference, Oct. 7–9, Tallinn. 1998.
8. Лахно В.А., Гусев Б.С., Смолий В.В., Місюра М.Д., Касаткін Д.Ю. Технології проектування комп'ютерних систем (частина 1). К.: НУБіП України, 2019. 205 с.
9. Жабін В.І., Жуков І.А., Клименко І.А., Стіренко С.Г. Арифметичні та управляючі пристрої цифрових ЕОМ: Навчальний посібник. – К.: БЕК +, 2008. 176 с.
10. Czerwinski R., Kania D. Finite state machines logic synthesis for complex programmable logic devices. Berlin: Springer, 2013. 172 p.
11. Bulatowa I., Radziwoniuk M. Design of pseudo-equivalent microprogram automata on programmable logic devices. Zeszyty Naukowe Politechniki Białostockiej. Informatyka, vol. 7. 2011. pp. 17–29.
12. Мартинюк Т.Б., Круківський Б.І., Богомолов С.В., Кузіна А.О. Синтез пристрою керування на базі R-автомата для асоціативного процесора. “інформаційні технології та комп'ютерна інженерія”, 2022, № 2. С. 79–85.
13. Barkalov A., Babakov R. Operational formation of state codes in microprogram automata. Cybernetics and System Analysis. V. 47, 2. 2011. pp. 193–197.
14. Barkalov A., Titarenko L. Logic Synthesis for Compositional Microprogram Control Units. Springer, Berlin. 2008. 272 pp.

Tysh Ie.V. GENERALIZED ALGORITHM FOR THE SYNTHESIS OF COMPUTER SYSTEMS COMPONENTS BASED ON MICROPROGRAM AUTOMATA

The synthesis of components (logical blocks) is an integral part of computer system development, as it ensures the creation of functional units that collectively implement the system's tasks. This process generally encompasses not only the construction of individual blocks but also their optimal integration into the overall architecture, enhancing both productivity and reliability. Through synthesis, the coordinated interaction of all system components is achieved, forming the foundation for effective system functionality.

The article substantiates a generalized algorithm for the synthesis of computer system components based on microprogrammed automata, encompassing the stages of abstract and structural synthesis. It thoroughly examines the methodology for constructing control automata with rigid logic, starting with the development of a graph-scheme of algorithm for the functioning of a logical block, followed by the creation of its abstract model in the form of an abstract Mealy or Moore automaton, and its presentation as a functional (structural) diagram. The principles of constructing control automata with programmable logic are explored, including the selection of the addressing type, the method of encoding microinstructions, writing the microprogram in machine microcodes, and choosing the elements for the functional (structural) diagram. Additionally, the algorithm for the structural synthesis of operational automata is described. This involves developing a graph diagram of the logical block's algorithm and representing each function within the algorithm using the corresponding combinational or sequential operational elements in the component's structural diagram. The canonical method for the structural synthesis of operational automata discussed in the article facilitates the optimal combination of operational elements of different types into a unified structural diagram.

Thus, the algorithm proposed in the article integrates existing methods into a cohesive framework, enhancing the efficiency of the computer system component synthesis process. This approach makes the synthesis process more structured, comprehensible, and accessible for implementation, contributing to the development of reliable, functional, and optimized systems that meet modern requirements.

Key words: computer system components, microprogram automaton, graph-scheme of algorithm, abstract synthesis, structural synthesis, control automaton, operational automaton, functional diagram, structural diagram.