

DOI [https://doi.org/10.15589/znp2021.1\(484\).7](https://doi.org/10.15589/znp2021.1(484).7)
УДК 681.5.013

LOGICAL SYNTHESIS OF DISCRETE AUTOMATIC CONTROL SYSTEMS USING LOW-LEVEL PROGRAMMED RELAYS

ЛОГІЧНИЙ СИНТЕЗ ДИСКРЕТНИХ СИСТЕМ АВТОМАТИЧНОГО КЕРУВАННЯ ПРИ ВИКОРИСТАННІ ПРОГРАМОВАНИХ РЕЛЕ НИЗЬКОГО РІВНЯ

Serhii O. Burian

sburyan18@gmail.com

ORCID: 0000-0002-4947-0201

Mykola V. Pechenik

pechenikmv@ukr.net

ORCID: 0000-0002-4527-1125

Hanna Y. Zemlianukhina

annzemlya@gmail.com

ORCID: 0000-0002-9653-8416

Ivan S. Yepifantsev

ye.sev00@gmail.com

ORCID: 0000-0001-5497-0609

С. О. Бур'ян,

канд. техн. наук, доцент

М. В. Печеник,

канд. техн. наук, професор

Г. Ю. Землянухіна,

асистент

І. С. Єпіфанцев,

студент

National Technical University of Ukraine “Igor Sikorsky Kyiv Polytechnic Institute”, Kyiv

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського», м. Київ

Abstract. To control complex automation systems, as a rule, intelligent industrial controllers are used, which have a rather complex structure and high cost characteristics. However, for local control systems that provide a certain production task either independently or within the general technological cycle is quite promising to use low-level software relays, which are characterized by low cost, ease of programming, ease of operation. However, the disadvantage of these relays is the limited number of memory cells and elements that are recorded in the program, which in turn leads to the use of more expensive controllers for local automation systems with a large number of states.

Purpose. Improving the method of synthesis of logic circuits based on graph transitions for its use in the development of discrete process control systems with a large number of states under the condition of technical implementation of the program on low-class logic relays.

Method. The improvement of the synthesis method is to transform a bulky graph transition, which describes the transitions between the states of a discrete circuit into a chain of several graph transitions, each of which describes part of the states of the automation system or process. The last signal that converts the current graph transition to the initial state is also the first signal of the next graph transition. This in turn removes the limitations of using low-level logic relays in complex automation systems, as it reduces the number of memory cells required to write a program.

Results. The proposed modification assumes that instead of one graph transition, which describes all the states of the automation system, a chain of small graph transitions is used, which successively pass from one to another. This reduces the number of conjunctive terms in each equation, which allows you to use the resulting equations to compile programs for logic programmable relays, which are limited by the number of memory cells and the number of elements recorded in the program.

Scientific novelty. The method of synthesis of discrete automation systems based on graph transitions is improved, which consists in transforming a bulky graph transition describing transitions between states of a discrete circuit into a chain of several graph transitions, each of which describes a part of automation system or process states.

Practical importance. The advanced method has elements of unification and can be used to synthesize logical equations describing automatic control systems that operate on a linear algorithm, i.e. the circuit sequentially passes from one state to another, which is also a characteristic feature of electromechanical systems of general industrial mechanisms of cyclic mode work.

Key words: logic relay; programming; logic synthesis; graph transition; automation system; trajectory; chain of graph junctions.

Анотація. Для керування комплексними системами автоматизації, як правило, використовуються інтелектуальні промислові контролери, які мають досить складну структуру і високі вартісні характеристики. Разом із тим для локальних систем керування, які забезпечують виконання певної виробничої задачі або самостійно, або в рамках загального технологічного циклу, досить перспективним є використання програмних реле низького рівня, яким властиві невисока вартість, простота в програмуванні, зручність в експлуатації. Однак недоліком зазначених реле можна відзначити обмежену кількість комірок пам'яті й елементів, які записуються в програмі, що у свою чергу призводить до використання більш дорогих контролерів для локальних систем автоматизації з великою кількістю станів.

Мета. Вдосконалення методу синтезу логічних схем на основі графопереходів для використання його під час розроблення дискретних систем керування технологічними процесами з великою кількістю станів за умови технічної реалізації програми на логічних реле низького класу.

Методика. Вдосконалення методу синтезу полягає в перетворенні громіздкого графопереходу, який описує переходи між станами дискретної схеми в ланцюг із декількох графопереходів, кожен з яких описує частину станів системи автоматизації або технологічного процесу. Останній сигнал, який переводить поточний графоперехід у вихідний стан, є одночасно першим сигналом наступного графопереходу. Це у свою чергу знімає обмеження застосування в складних системах автоматизації логічних реле низького рівня, оскільки зменшується кількість комірок пам'яті, необхідних для написання програми.

Результати. Запропонована модифікація передбачає, що замість одного графопереходу, що описує всі стани системи автоматизації, застосовується ланцюг із невеликих графопереходів, які послідовно переходять з одного на інший. При цьому зменшується кількість кон'юнктивних членів у кожному рівнянні, що дозволяє використовувати отримані рівняння для складання програм для логічних програмованих реле, що мають обмеження за кількістю комірок пам'яті та кількістю елементів, що записуються в програмі.

Наукова новизна. Вдосконалено метод синтезу дискретних систем автоматизації на основі графопереходів, який полягає в перетворенні громіздкого графопереходу, що описує переходи між станами дискретної схеми, в ланцюг із декількох графопереходів, кожен з яких описує частину станів системи автоматизації або технологічного процесу.

Практична значимість. Вдосконалений метод має елементи уніфікації, і його можна використовувати для синтезу логічних рівнянь, що описують системи автоматичного керування, які працюють за лінійним алгоритмом, тобто схема послідовно переходить з одного стану в інший, що є характерною рисою також і для електромеханічних систем механізмів загальнопромислового призначення циклічного режиму роботи.

Ключові слова: логічне реле; програмування; логічний синтез; графоперехід; система автоматизації; траєкторія; ланцюг графопереходів.

ПОСТАНОВКА ЗАДАЧІ

Дискретні системи керування набули широкого поширення як під час комплексної автоматизації промислових підприємств і окремих технологічних процесів, так і локальних виробничих циклів із замкнутим характером роботи. Для керування даними процесами, як правило, використовуються інтелектуальні промислові контролери [1; 2], які мають досить складну структуру і високі вартісні характеристики.

АНАЛІЗ ОСТАННІХ ДОСЛІДЖЕНЬ І ПУБЛІКАЦІЙ

Програмування таких контролерів зазвичай здійснюється на основі попередньо виконаного логічного синтезу на основі мереж Петрі [3–5], які дозволяють будувати розподілені системи автоматизації, з використанням об'єктно-орієнтованого підходу [6] або мов програмування ST (Structured Text – мова структурованого тексту) та FBD (Function Block Diagram – мова функціональних блоків). Разом із тим для локальних систем керування, які забезпечують виконання певної виробничої задачі або самостійно, або в рамках загального технологічного циклу, досить перспективним є використання програмних реле низького рівня, яким

властиві невисока вартість, простота в програмуванні на базі мови LD (Ladder Diagram – мова релейної логіки), зручність в експлуатації. Однак недоліком зазначених реле можна відзначити обмежену кількість комірок пам'яті й елементів, які записуються в програмі. Так, логічне реле Lovato серії LRX [7] допускає записати в програму кон'юнкцію, що складається з трьох елементів, реле Schneider серії Zelio Logic [8] – з п'яти. З огляду на такі умови до логічного синтезу [9; 10] пред'являються обмеження в кількості логічних елементів у кожній кон'юнкції логічного рівняння, при цьому число проміжних змінних не може перевищувати обсяг комірок пам'яті контролера. Зазначені обмеження є не суттєвими для схем із невеликою кількістю станів, але в разі збільшення їх кількості може виникнути ситуація, за якої розроблена програма не вміститься в пам'ять контролера.

ВІДОКРЕМЛЕННЯ НЕ ВИРІШЕНИХ РАНІШЕ ЧАСТИН ЗАГАЛЬНОЇ ПРОБЛЕМИ

Досить перспективним напрямом синтезу логічних схем є метод графопереходів, який дозволяє врахувати всі можливі стани схеми і переходи між ними [9; 11]. Разом із тим зазначений вище недолік програмованих реле обмежує їх застосування в разі,

коли кількість станів перевищує 16. Зняття цього обмеження можливе або за допомогою контролерів високого рівня, або в результаті вдосконалення методів синтезу логічних систем. З огляду на вищевикладене досить актуальною є задача модифікації методу синтезу логічних схем на базі графопереходів, який за рахунок зменшення кон'юнктивних членів у рівняннях дозволяє використовувати реле низького рівня, в тому числі з трьома і більше елементами в одній кон'юнкції і кількістю станів у схемі більше 16-ти.

МЕТА ДОСЛІДЖЕННЯ

Вдосконалення методу синтезу логічних схем на основі графопереходів для використання його про розроблення дискретних систем керування технологічними процесами з великою кількістю станів за умови технічної реалізації програми на логічних реле низького класу.

МЕТОДИ, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

У роботі використані методи логічного синтезу дискретних систем автоматизації, методи програмування логічних контролерів низького класу. Об'єктом дослідження є процеси дискретного керування в системах автоматизації. Предметом дослідження є дискретні системи автоматичного керування електроприводами типових загальнопромислових механізмів.

ОСНОВНИЙ МАТЕРІАЛ

Опис удосконаленої методики розглянемо на прикладі технологічного процесу частотного керування асинхронним двигуном за відпрацювання заданої траєкторії руху [12], що представлена на рис. 1. Керування електроприводом здійснюється за допомогою логічних реле типу Lovato серії LRX.

Використовуючи дискретні входи перетворювача, логічне реле буде подавати команди (табл. 1).

Налаштування задатчика інтенсивності перетворювача частоти приймемо такі: час розгону $t_p=8$ с, час сповільнення $t_c=7$ с. Синхронна швидкість обертання двигуна $\omega_c=157$ рад/с. Значення часу на усталених

ділянках такі: $t_2=t_4=t_6=1,2$ с; $t_1=t_3=t_5=t_7=0,8$ с. У відповідності до вихідних даних виконаємо розрахунок часових ділянок розгону та сповільнення.

Таблиця 1. Призначення дискретних входів перетворювача

Позначення входу	Призначення
I1	Дозвіл роботи вперед
I2	Дозвіл роботи назад
I3	Розгін
I4	Сповільнення

Час роботи на окремій ділянці визначається як

$$t = \frac{t_p(\omega_1 - \omega_0)}{\omega_c}, \quad (1)$$

де t_p – час розгону/сповільнення до синхронної частоти; ω_0 – початкова швидкість ділянки; ω_1 – кінцева швидкість ділянки.

Відповідно до виразу (1) визначаємо час роботи на пологих ділянках траєкторії руху

$$t'_{2} = \frac{7 \cdot (50 - 30)}{157} = 0,89 \text{ с}; \quad t'_{31} = \frac{8 \cdot (50 - 0)}{157} = 2,54 \text{ с};$$

$$t'_{32} = \frac{7 \cdot (60 - 0)}{157} = 2,67 \text{ с};$$

$$t'_{4} = \frac{7 \cdot (75 - 60)}{157} = 0,66 \text{ с}; \quad t'_{5} = \frac{8 \cdot (75 - 60)}{157} = 0,76 \text{ с};$$

$$t'_{61} = \frac{8 \cdot (60 - 0)}{157} = 3,05 \text{ с};$$

$$t'_{62} = \frac{7 \cdot (50 - 0)}{157} = 2,22 \text{ с}; \quad t'_{7} = \frac{8 \cdot (50 - 30)}{157} = 1,01 \text{ с};$$

$$t'_{8} = \frac{8 \cdot (30 - 0)}{157} = 1,52 \text{ с}.$$

Наступним етапом виконаємо синтез методом графопереходів на основі RS-тригерів [9].

Кількість вершин графу при синтезі схем визначають з умови $2^n \geq S$, де S – кількість станів схеми; 2^n – кількість вершин графу; n – кількість тригерів. Вершини

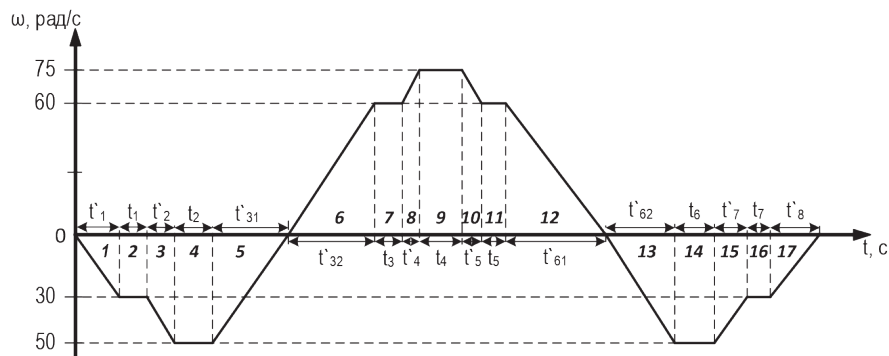


Рис. 1. Задана траєкторія відпрацювання кутової швидкості двигуна

графу рекомендується розміщувати так, щоб коли $n=2$, вони створювали конфігурацію 2×2 , якщо $n=3$, – конфігурацію 4×2 , якщо $n=4$, – конфігурацію 4×4 і так далі.

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їхні вихідні сигнали позначають буквами P_1, P_2, P_3 . Ці позначення записують зверху ліворуч від графу. Кожна вершина графу кодується набором значень вихідних сигналів тригерів. Коди вибирають так, щоб для сусідніх вершин вони розрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) зазвичай припускають, що всі проміжні змінні дорівнюють нулю. Значення проміжних змінних для кожної вершини записують над кружками в тій послідовності, в якій записано позначення тригерів.

У середині вершин записують номери станів схеми. Стани, між якими має відбуватися перехід згідно з умовами роботи схеми, розміщуються в сусідніх вершинах, якщо це можливо. Під вершинами з номерами станів записують значення вихідних змінних, що відповідають цим станам. Позначення вихідних змінних записують ліворуч від графу під позначеннями тригерів. Вершини, між якими повинні відбуватися переходи, з'єднують ребрами зі стрілками. Над стрілками або праворуч від них, якщо ребра спрямовано вертикально, записують позначення вхідних сигналів, що спричиняють ці переходи.

У схемах на асинхронних RS-тригерах переходи можна робити тільки між сусідніми вершинами. Якщо ця умова не виконується, то необхідно передбачити природні переходи (за рахунок подавання вхідного сигналу логічної «оддиниці») через проміжні нестійкі стани.

Відповідно до заданої тахограми руху (рис. 1) логічна схема має 17 робочих станів та 1 вихідний стан. За методом синтезу необхідно побудувати графоперехід на 5 тригерів ($2^n \geq 18$, звідки $n=5$). Проте такий графоперехід буде мати 32 вершини, а всі рівняння будуть мати мінімум 5 елементів в одній кон'юнкції.

Для задач з великою кількістю станів уведемо поняття «ланцюга графопереходів». Суть цього методу – у застосуванні декількох графопереходів для

запису однієї послідовності станів. Останній сигнал, який переводить поточний графоперехід у вихідний стан, є одночасно першим сигналом наступного графопереходу.

За умовами роботи схеми маємо 18 станів системи (17+1). Складемо 6 графопереходів по 2 тригери кожний, як показано на рис. 2. Стани, позначені квадратом, означають перехід на наступний графоперехід. У цьому стані вихідні сигнали завжди дорівнюють 0.

Позначення вхідних і вихідних сигналів, а також сигналів таймерів, які необхідно розглядати як вхідні сигнали для графопереходів, наведені в табл. 2.

На рис. 2 наведений логічний ланцюг із шести графопереходів по 4 вершини кожний. У кожному графопереході один проміжний стан використовується для переходу на інший графоперехід, тому в підсумку маємо 24 вершини (4 вершини на кожен із 6 графопереходів), на яких зображено 18 станів.

За графопереходами складемо рівняння умов вмикання і скидання кожного тригера.

Умови вмикання тригерів:

$$\begin{aligned} S_{p1} &= Tx1 \cdot \underline{p2}; & S_{p2} &= START \cdot \overline{p1}; & S_{p3} &= T2 \cdot p4; \\ S_{p4} &= Tx2 \cdot \underline{p3}; & S_{p5} &= T3 \cdot \underline{p6}; & S_{p6} &= Tx32 \cdot \underline{p5}; \\ S_{p7} &= Tx5 \cdot \underline{p8}; & S_{p8} &= T4 \cdot \underline{p7}; & S_{p9} &= Tx62 \cdot \underline{p10}; \\ S_{p10} &= Tx61 \cdot \underline{p9}; & S_{p11} &= T7 \cdot \underline{p12}; & S_{p12} &= Tx7 \cdot \underline{p11}. \end{aligned}$$

Умови скидання тригерів:

$$\begin{aligned} R_{p1} &= Tx2 \cdot \underline{p2}; & R_{p2} &= T1 \cdot \underline{p1}; & R_{p3} &= Tx32 \cdot \overline{p4}; \\ R_{p4} &= Tx31 \cdot \underline{p3}; & R_{p5} &= T4 \cdot \underline{p6}; & R_{p6} &= Tx4 \cdot \underline{p5}; \\ R_{p7} &= Tx61 \cdot \underline{p8}; & R_{p8} &= T5 \cdot \underline{p7}; & R_{p9} &= Tx7 \cdot \underline{p10}; \\ R_{p10} &= T6 \cdot \underline{p9}; & R_{p11} &= \underline{p12}; & R_{p12} &= Tx8 \cdot \underline{p11}. \end{aligned}$$

Рівняння для вихідних сигналів:

$$\begin{aligned} Fwd &= p3p4 + p5 + p6 + p7 + p8; \\ Bwd &= \underline{p1} + p2 + \underline{p4} + \underline{p9} + p10 + p12; \\ Acc &= \underline{p1p2} + \underline{p1p2} + \underline{p3p4} + \underline{p5p6} + \underline{p9p10}; \\ Dec &= p3p4 + \underline{p7p8} + \underline{p7p8} + \underline{p9p10} + p11. \end{aligned}$$

Рівняння умов вмикання таймерів:

$$\begin{aligned} Tx1 &= \underline{p1p2}; & T1 &= p1p2; & Tx2 &= \underline{p1p2}; \\ T2 &= \underline{p3p4}; & Tx31 &= p3p4; & Tx32 &= \underline{p3p4}; \\ T3 &= \underline{p5p6}; & Tx4 &= p5p6; & T4 &= \underline{p5p6}; \\ Tx5 &= \underline{p7p8}; & T5 &= p7p8; & Tx61 &= \underline{p7p8}; \\ Tx62 &= \underline{p9p10}; & T6 &= p9p10; & Tx7 &= \underline{p9p10}; \\ T7 &= \underline{p11p12}; & Tx8 &= p11p12. \end{aligned}$$

Таблиця 2. Позначення вхідних та вихідних сигналів на графопереходах

Назва сигналу	Призначення сигналу	Тип сигналу
START	команда «Пуск»	вхідна змінна
T1 – T7	сигнали таймерів, що дають затримки t1-t7	проміжні змінні (для графопереходу – вхідні змінні)
Tx1 – Tx8	сигнали таймерів, що дають затримки $t'_1 - t'_8$	проміжні змінні (для графопереходу – вхідні змінні)
Fwd	дозвіл роботи вперед	вихідна змінна
Bwd	дозвіл роботи назад	вихідна змінна
Acc	розгін	вихідна змінна
Dec	сповільнення	вихідна змінна
P1 – P12	сигнали тригерів	проміжні змінні

На основі отриманих рівнянь складаємо програму на мові LD для програмованого реле Lovato серії LRX, що представлена на рис. 3. Адресацію змінних наведено в табл. 3.

Таблиця 3. Адресація змінних

Назва сигналу	Позначення у програмі
START	И
T1 – T7	TA – T10
Tx1 – Tx8	T01 – T09
Fwd, Bwd, Acc, Dec	Q1, Q2, Q3, Q4
P1 – P12	M01 – M12

ОБГОВОРЕННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Наведена програма задовольняє всім обмеженням, що мають програмовані логічні реле Lovato серії LRX, та може бути використана для будь-якого програмованого реле низького рівня.

ВИСНОВКИ

У роботі представлений удосконалений метод синтезу логічних схем на основі графопереходів для розроблення дискретних систем керування технологічними процесами з великою кількістю станів за умови технічної реалізації програми на логічних реле

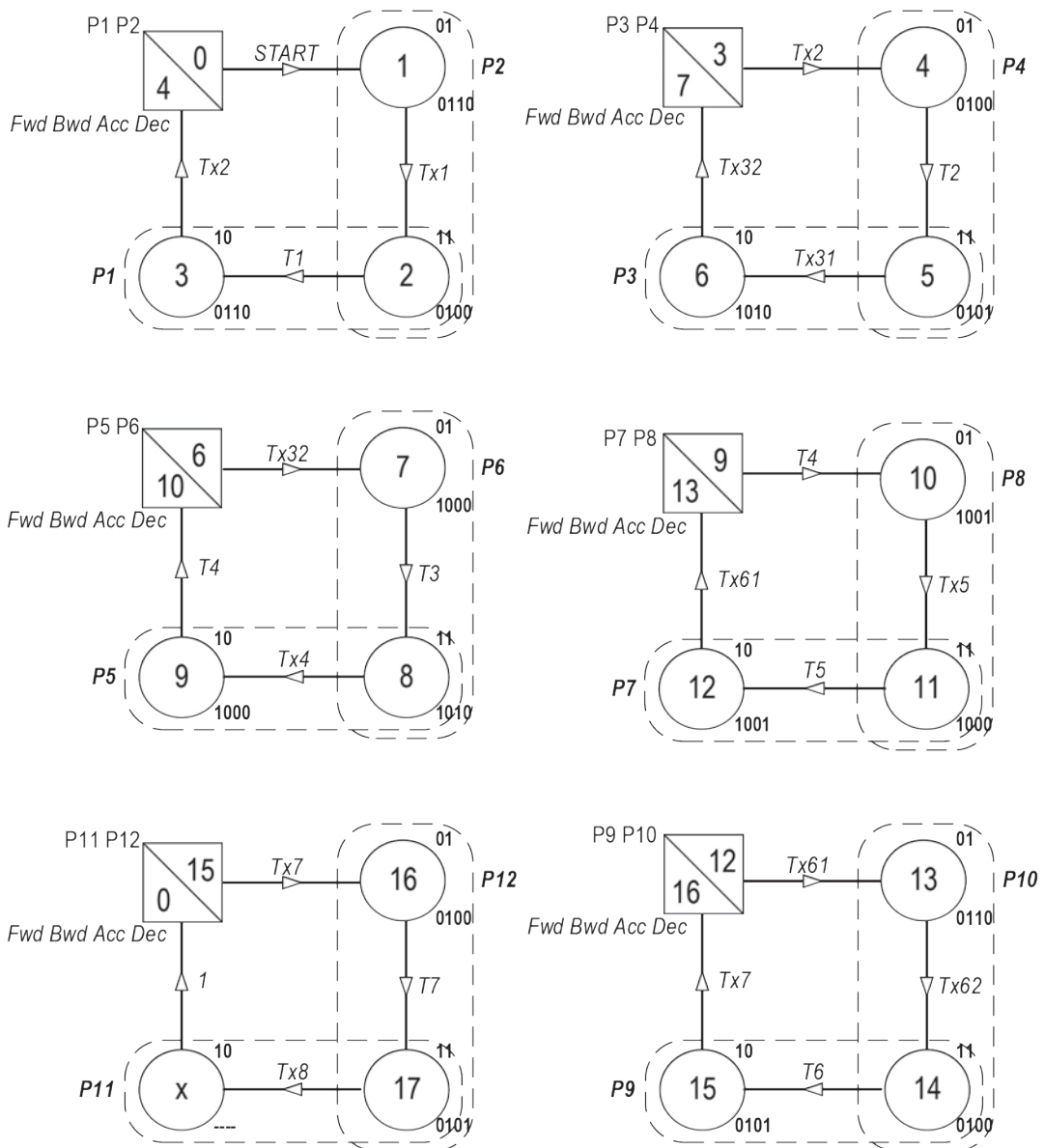


Рис. 2. Логічний ланцюг із шести графопереходів 2x2

низького класу. Запропонована модифікація передбачає, що замість одного графопереходу, що описує всі стани системи автоматизації, застосовується ланцюг із невеликих графопереходів, які послідовно переходять з одного на інший. При цьому зменшується кількість кон'юнктивних членів у кожному рівнянні, що дозволяє використовувати отримані рівняння для складання програм для логічних програмованих реле, що мають обмеження за кількістю комірок пам'яті

та кількістю елементів, що записуються в програмі. Даний метод має елементи уніфікації, і його можна використовувати для синтезу логічних рівнянь, що описують системи автоматичного керування, які працюють за лінійним алгоритмом, тобто схема послідовно переходить з одного стану в інший, що є характерною рисою також і для електромеханічних систем механізмів загальнопромислового призначення циклічного режиму роботи.

T01	—	M02	—	—	—	↑M01	T08	—	M11	—	—	—	↓M12
T02	—	m02	—	—	—	↓M01	M03	—	m04	—	—	┐	(Q1
П	—	m01	—	—	—	↑M02	M05	—	—	—	—	┐	
T0A	—	M01	—	—	—	↓M02	M06	—	—	—	—	┐	
T0B	—	M04	—	—	—	↑M03	M07	—	—	—	—	┐	
T103	—	m04	—	—	—	↓M03	M08	—	—	—	—	┘	
T0C	—	m03	—	—	—	↑M04	M01	—	—	—	—	┐	(Q2
T0D	—	M03	—	—	—	↓M04	M02	—	—	—	—	┐	
T13	—	M06	—	—	—	↑M05	M04	—	—	—	—	┐	
T04	—	m06	—	—	—	↓M05	M09	—	—	—	—	┐	
T05	—	m05	—	—	—	↑M06	M10	—	—	—	—	┐	
T06	—	M05	—	—	—	↓M06	M12	—	—	—	—	┘	
T0D	—	M08	—	—	—	↑M07	m01	—	M02	—	—	┐	(Q3
T0E	—	m08	—	—	—	↓M07	M01	—	m02	—	—	┐	
T16	—	m07	—	—	—	↑M08	M03	—	m04	—	—	┐	
T07	—	M07	—	—	—	↓M08	M05	—	M06	—	—	┐	
T06	—	M10	—	—	—	↑M09	m09	—	M10	—	—	┘	
T07	—	m10	—	—	—	↓M09	M03	—	M04	—	—	┐	(Q4
T06	—	m09	—	—	—	↑M10	07	—	M08	—	—	┐	
T0F	—	M09	—	—	—	↓M10	M07	—	m08	—	—	┐	
T10	—	M12	—	—	—	↑M11	m09	—	M10	—	—	┐	
—	—	m12	—	—	—	↓M11	M11	—	—	—	—	┘	
T07	—	m11	—	—	—	↑M12	m01	—	M02	—	—	—	(T01
M01	—	m02	—	—	—	(T02	M11	—	M12	—	—	—	(T08
M03	—	M04	—	—	—	(T03	M01	—	M02	—	—	—	(T0A
M03	—	m04	—	—	—	(T13	m03	—	M04	—	—	—	(T0B
M05	—	M06	—	—	—	(T04	m05	—	M06	—	—	—	(T0C
m07	—	M08	—	—	—	(T05	M05	—	m06	—	—	—	(T0D
M07	—	m08	—	—	—	(T06	M07	—	M08	—	—	—	(T0E
m09	—	M10	—	—	—	(T16	M09	—	M10	—	—	—	(T0F
M09	—	m10	—	—	—	(T07	m11	—	M12	—	—	—	(T10

Рис. 3. Програма керування двигуном на мові LD для програмованого реле Lovato серії LRX

REFERENCES

- [1] Basile, F., Chiacchio, P., & Gerbasio, D. (2012). On the implementation of industrial automation systems based on PLC. *IEEE Transactions on Automation Science and Engineering*, 10(4), 990-1003.
- [2] Zhou, M., & Twiss, E. (1998). Design of industrial automated systems via relay ladder logic programming and Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 28(1), 137-150.
- [3] Frey, G. (2000, June). Automatic implementation of Petri net based control algorithms on PLC. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)* (Vol. 4, pp. 2819-2823). IEEE.
- [4] Peng, S., & Zhou, M. (2003). Sensor-based stage Petri net modelling of PLC logic programs for discrete-event control design. *International Journal of Production Research*, 41(3), 629-644.
- [5] Dai, W., & Vyatkin, V. (2012). Redesign distributed PLC control systems using IEC 61499 function blocks. *IEEE Transactions on Automation Science and Engineering*, 9(2), 390-401.
- [6] Schunemann, U. (2007). Programming Plcs with an object-oriented approach. *Automation Technology in Practice*, 2(B3654), 59-63.
- [7] *Base modules | Lovato Electric*. (2015, May). Lovato Electric Spa. <https://www.lovatoelectric.com/ProductsList.aspx?id=430005&text=Base-modules>.
- [8] *Compact smart relays - Zelio Logic SR2/SR3 | Schneider Electric Global*. (2020, October). Schneider Electric Global. <https://www.se.com/ww/en/product-range/531-zelio-logic-sr2-sr3/#>.
- [9] Kovalchuk, O. V. (2008). Lohichnyi syntez dyskretnykh skhem avtomatyky navch. posib. K.: NTUU «KPI». [in Ukrainian]
- [10] Kovalchuk, O. V., & Burian, S. O. (2010). Zastosuvannia riznykh metodiv v syntezu dlia skladnykh prohram dlia lohichnykh prohramovanykh kontroleriv [Application of various methods in synthesis for complex programs for logical programmable controllers]. *Informatsiyni zbirnyk Promelektro. «Promyslova elektronika ta elektrotehnika»*, (4), 51-53. [in Ukrainian]
- [11] Vingron, S. P. (2012). *Logic circuit design: Selected methods*. Springer Science & Business Media.
- [12] Burian, S. O., Voroshchenko, V. Yu., Korol, S. V., Savych, O. Yu., & Smirnov, S. A. (2011). Avtomatyzatsiia formuvannia traiektorii pry eksperymentalnomu doslidzhenni elektromekhanichnykh system [Automation of trajectory formation in experimental study of electromechanical systems]. *Elektrotehnika i elektroenerhetyka*, (2), 30-37. [in Ukrainian]

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

- [1] Basile, F., Chiacchio, P., Gerbasio, D. (2012). On the implementation of industrial automation systems based on PLC. *IEEE Transactions on Automation Science and Engineering*.10(4). 990–1003.
- [2] Zhou, M., Twiss, E. (1998). Design of industrial automated systems via relay ladder logic programming and Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*. 28(1). 137–150.
- [3] Frey, G. (2000, June). Automatic implementation of Petri net based control algorithms on PLC. In *Proceedings of the 2000 American Control Conference. ACC (IEEE Cat. No. 00CH36334)*. Vol. 4. P. 2819–2823. IEEE.
- [4] Peng, S., Zhou, M. (2003). Sensor-based stage Petri net modelling of PLC logic programs for discrete-event control design. *International Journal of Production Research*. 41(3). 629–644.
- [5] Dai, W., Vyatkin, V. (2012). Redesign distributed PLC control systems using IEC 61499 function blocks. *IEEE Transactions on Automation Science and Engineering*. 9(2). 390–401.
- [6] Schunemann, U. (2007). Programming Plcs with an object-oriented approach. *Automation Technology in Practice*. 2(B3654). 59–63.
- [7] *Base modules | Lovato Electric*. (2015, May). Lovato Electric Spa. URL: <https://www.lovatoelectric.com/ProductsList.aspx?id=430005&text=Base-modules>
- [8] *Compact smart relays - Zelio Logic SR2/SR3 | Schneider Electric Global*. (2020, October). Schneider Electric Global. URL: <https://www.se.com/ww/en/product-range/531-zelio-logic-sr2-sr3/#>
- [9] Ковальчук, О. В. (2008). Логічний синтез дискретних схем автоматики : навч. посіб. Київ : НТУУ «КПІ».
- [10] Ковальчук, О. В., Бур'ян, С. О. (2010). Застосування різних методів в синтезу для складних програм для логічних програмованих контролерів. *Інформаційний збірник Промелектро. «Промислова електроніка та електротехніка»*. (4). 51–53.
- [11] Vingron, S. P. (2012). *Logic circuit design: Selected methods*. Springer Science & Business Media.
- [12] (2011). Автоматизація формування траєкторій при експериментальному дослідженні електромеханічних систем. *Електротехніка і електроенергетика / С. О. Бур'ян, та ін. № (2). С. 30–37.*

© Бур'ян С. О., Печеник М. В., Землянхуна Г. Ю., Спіфанцев І. С.
 Дата надходження статті до редакції: 09.03.2021
 Дата затвердження статті до друку: 23.03.2021