

# ОСОБЛИВОСТІ ЛОГІЧНОГО СИНТЕЗУ ДИСКРЕТНИХ СИСТЕМ АВТОМАТИЗАЦІЇ З ВЕЛИКОЮ КІЛЬКІСТЮ СТАНІВ МЕТОДОМ ГРАФОПЕРЕХОДІВ

**Бур'ян С.О., к.т.н., доц., Печеник М.В., к.т.н., проф., Єпіфанцев І.С., студент**  
*КПІ ім. Ігоря Сікорського, кафедра автоматизації електромеханічних систем та електроприводу*

**Вступ.** Дискретні системи автоматизації широко розповсюджені на промислових підприємствах, у комунальному, сільському господарстві та стрімко продовжують поширюватися. Комплексна автоматизація будь-якого підприємства складається з низки автоматизованих технологічних процесів, кожен з яких може бути автономним або підпорядкований іншим. Для здійснення керування та моніторингу такими процесами поширені інтелектуальні промислові контролери. Проте, нескладні автономні процеси можуть обмежитися програмованими реле низького рівня. Перевагою таких реле є низька вартість, відносна простота програмування на мовах LD та FBD. Однак, недоліком простих логічних реле є обмежена кількість комірок пам'яті та кількість елементів, що записуються у програмі. Наприклад, логічне реле Lovato серії LRX [1] дозволяють записати у програму кон'юнкцію з максимум трьох елементів, а реле Schneider серії Zelio Logic [2] – всього п'ять. За таких умов на логічних синтез [3] накладаються досить жорсткі обмеження: логічні рівняння мають містити обмежену кількість елементів у кожній кон'юнкції, а кількість проміжних змінних не може перевищувати кількість комірок пам'яті контролера. Ці обмеження не є суттєвими для логічних схем з невеликою кількістю станів, проте, збільшення кількості станів унеможливає використання простих логічних реле, так як збільшується кількість проміжних змінних та кон'юнктивних компонентів рівнянь.

Оптимальним методом синтезу логічних схем, що з однієї сторони дозволяє графічно зобразити всі стани схеми, а з іншого – врахувати всі можливі переходи між станами є синтез методом графопереходів [3-5]. Проте вказаний вище недолік логічних реле унеможливає їх використання коли кількість станів збільшується хоча б на один відносно 16. У такому випадку необхідно використовувати графоперехід на 32 вершини або більше, що вносить до кожного рівняння мінімум 5 кон'юнктивних компонент за умови одного сигналу на ребрах графопереходу.

Для реалізації таких схем можна використати контролери високого рівня проте їх вартість значно вища, і такий вихід є неефективним з точки зору техніко-економічного обґрунтування проекту. Іншим виходом є модифікація методу синтезу на основі графопереходів, що передбачає зменшення кон'юнктивних членів у рівняннях, що описують логічну схему. Такий підхід дає можливість використовувати навіть логічні реле з трьома елементами у одній кон'юнкції при кількості станів схеми більшій за 16, що є актуальним завданням.

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

**Матеріали та результати досліджень.** Для опису вдосконаленої методики розглянемо простий технологічний процес керуванням асинхронним двигуном від перетворювача частоти при відпрацюванні заданої траєкторії руху [6-9], що представлена на рис. 1. Керування перетворювачем буде здійснюватися логічних реле Lovato серії LRX.

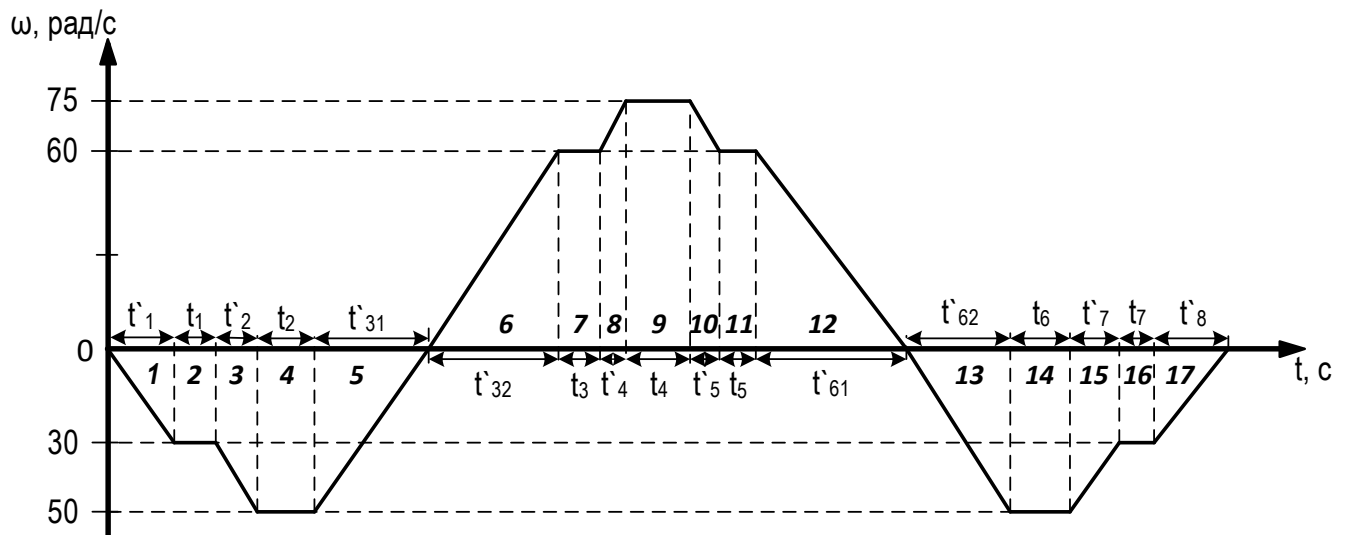


Рисунок 1 – Задана траєкторія відпрацювання кутової швидкості двигуна

Використовуючи дискретні входи перетворювача, логічне реле буде подавати команди, відповідно до табл. 1.

Таблиця 1 – Призначення дискретних входів перетворювача

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

Налаштування задатчика інтенсивності перетворювача частоти приймемо наступні: 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 наступні:  $t_2=t_4=t_6=1,2$  с;  $t_1=t_3=t_5=t_7=0,8$  с. У відповідності до вихідних даних виконаємо розрахунок часових ділянок розгону та сповільнення.

Час роботи на окремій ділянці визначається як

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

де  $t_p$  – час розгону/сповільнення до синхронної частоти;  $\omega_0$  – початкова швидкість ділянки;  $\omega_1$  – кінцева швидкість ділянки.

Відповідно до виразу (1) визначаємо значення часу на пологих ділянках

$$\begin{aligned} t_{2}^{\wedge} &= \frac{7 \cdot (50 - 30)}{157} = 0,89c; & t_{31}^{\wedge} &= \frac{8 \cdot (50 - 0)}{157} = 2,54c; & t_{32}^{\wedge} &= \frac{7 \cdot (60 - 0)}{157} = 2,67c; \\ t_{4}^{\wedge} &= \frac{7 \cdot (75 - 60)}{157} = 0,66c; & t_{5}^{\wedge} &= \frac{8 \cdot (75 - 60)}{157} = 0,76c; & t_{61}^{\wedge} &= \frac{8 \cdot (60 - 0)}{157} = 3,05c; \\ t_{62}^{\wedge} &= \frac{7 \cdot (50 - 0)}{157} = 2,22c; & t_{7}^{\wedge} &= \frac{8 \cdot (50 - 30)}{157} = 1,01c; & t_{8}^{\wedge} &= \frac{8 \cdot (30 - 0)}{157} = 1,52c. \end{aligned}$$

Після виконання розрахунків часових проміжків перейдемо безпосередньо до процедури синтезу методом графопереходів на основі RS-тригерів. Графоперехід – це графічне зображення послідовності роботи багатотактної схеми. Елементами графу є вершини і ребра. Вершини відповідають станам схеми і позначаються кружками. Ребра – це лінії зі стрілками, що з'єднують вершини і показують напрям переходу з одного стану схеми в інший.

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

Вихідні сигнали тригерів виконують роль проміжних змінних, тому тригери та їх 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 обирають так, щоб для сусідніх вершин вони розрізнялися значенням тільки однієї змінної. У вихідному стані схеми (стан очікування) зазвичай припускають, що всі проміжні змінні дорівнюють нулеві. Значення проміжних змінних для кожної вершини записують над кружками у тій черговості, у якій записано позначення тригерів.

Всередині 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 «одиниці») 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 введемо поняття «ланцюга графопереходів». Суть цього методу в 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 тригера кожний, як показано на рис. 2. Стани позначені квадратом означають перехід на наступний графоперехід. В 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 у табл. 2.

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 $t_1-t_7$	проміжні змінні (для графопереходу – вхідні змінні)
Tx1 – Tx8	сигнали таймерів, що дають затримки $t_1^{\wedge}-t_8^{\wedge}$	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 рис. 2 наведений логічний ланцюг з шести графопереходів по 4 вершини кожний. В кожному графопереході один проміжний стан використовується для переходу на інший графоперехід, тому в підсумку маємо 24 вершини (4 вершини на 6 графопереходів) на яких зображено 18 станів.

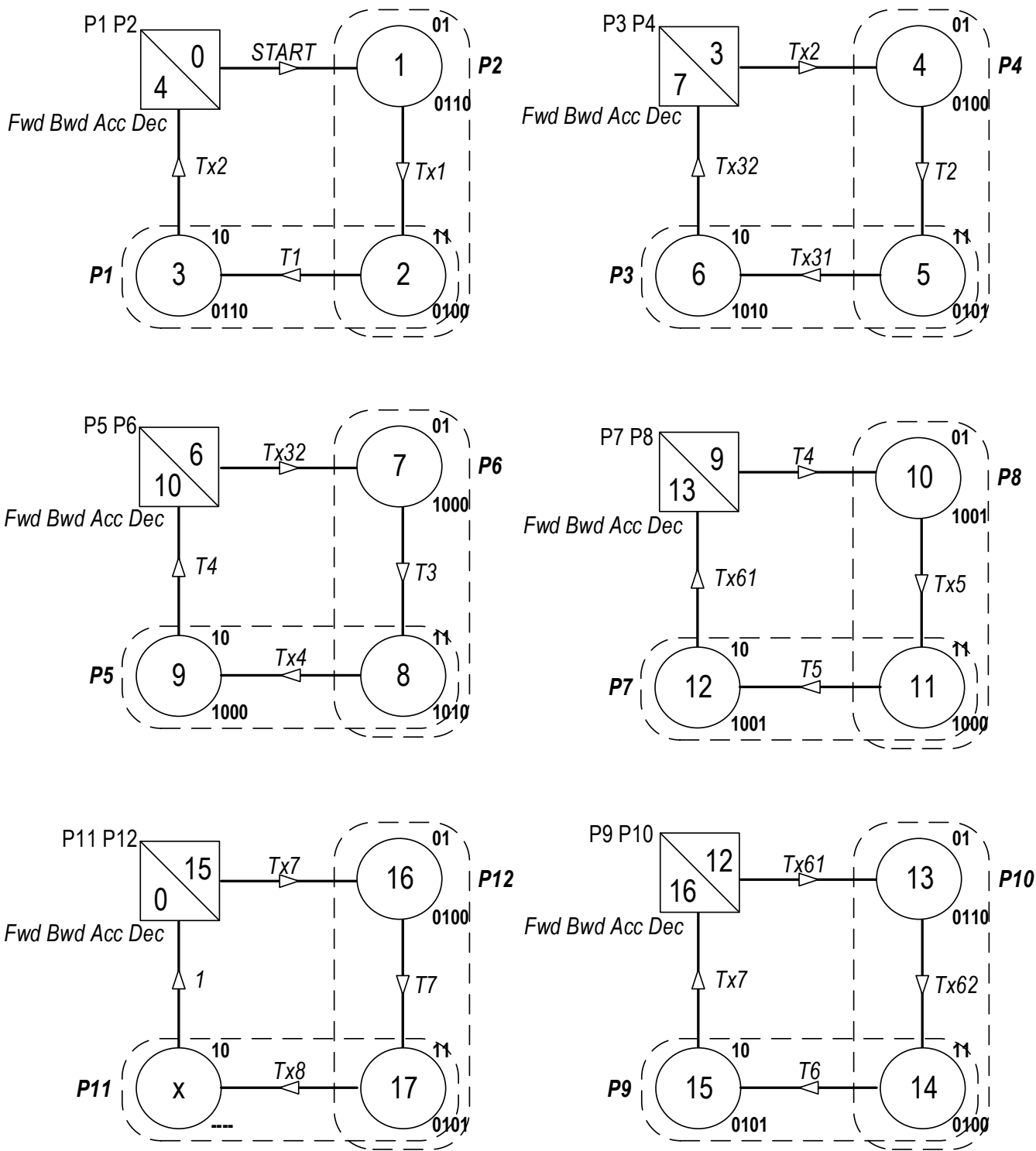


Рисунок 2 – Логічний ланцюг з шести графопереходів 2x2

За графопереходами складемо рівняння умов вмикання і скидання кожного тригера.

Умови вмикання тригерів:

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

Умови скидання тригерів:

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

Рівняння для вихідних сигналів:

$$\begin{aligned}
 Fwd &= p3\overline{p4} + p5 + p6 + p7 + p8; \\
 Bwd &= p1 + p2 + p4 + p9 + p10 + p12; \\
 Acc &= \overline{p1}p2 + p1\overline{p2} + p3\overline{p4} + p5p6 + \overline{p9}p10; \\
 Dec &= p3p4 + \overline{p7}p8 + p7\overline{p8} + p9\overline{p10} + p11.
 \end{aligned}$$

Рівняння умов вмикання таймерів:

$$\begin{aligned}
 Tx1 &= \overline{p1}p2; & T1 &= p1p2; & Tx2 &= p1\overline{p2}; \\
 T2 &= \overline{p3}p4; & Tx31 &= p3p4; & Tx32 &= p3\overline{p4}; \\
 T3 &= \overline{p5}p6; & Tx4 &= p5p6; & T4 &= p5\overline{p6}; \\
 Tx5 &= \overline{p7}p8; & T5 &= p7p8; & Tx61 &= p7\overline{p8}; \\
 Tx62 &= \overline{p9}p10; & T6 &= p9p10; & Tx7 &= p9\overline{p10}; \\
 T7 &= \overline{p11}p12; & Tx8 &= p11p12.
 \end{aligned}$$

На основі отриманих рівнянь складаємо програму на мові LD для програмованого реле Lovato серії LRX, що представлена на рис. 3. Адресація змінних наведена у табл. 3.

Таблиця 3 – Адресація змінних

Назва сигналу	Позначення у програмі
START	I1
T1 – T7	TA – T10
Tx1 – Tx8	T01 – T09
Fwd, Bwd, Acc, Dec	Q1, Q2, Q3, Q4
P1 – P12	M01 – M12

T01	—	M02	—	—	—	↑M01	T08	—	M11	—	—	—	↓M12
T02	—	m02	—	—	—	↓M01	M03	—	m04	—	—	—	(Q1
I1	—	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

Рисунок 3 – Програма керування двигуном на мові LD для програмованого реле Lovato серії LRX

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 вдосконалений методу синтезу логічних схем на основі графопереходів при розробці дискретних систем керування технологічними процесами з великою кількістю станів за умови технічної реалізації

програми на логічних реле низького класу. Запропонована модифікація передбачає, що замість одного графопереходу, що описує всі стани системи автоматизації, застосовується ланцюг з невеликих графопереходів, які послідовно переходять з одного на інший. При цьому зменшується кількість кон'юнктивних членів в 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 у програмі. Даний метод можна використовувати для синтезу логічних рівнянь, що описують будь-які системи автоматизації, що працюють за лінійним алгоритмом, тобто схема послідовно переходить з одного стану в інший. При виникненні розгалужень у станах, метод може бути застосований для тієї частини схеми, що не має розгалужень.

#### Перелік посилань

1. Програмовані логічні реле Lovato серії LRX. URL: <https://www.lovatoelectric.com/ProductsList.aspx?id=430005&text=Base-modules> (дата звернення 29.11.2020).
2. Програмовані логічні реле Zelio Logic. URL: <https://www.se.com/ua/ru/product-range/531-zelio-logic/> (дата звернення 29.11.2020).
3. Ковальчук О.В. Логічний синтез дискретних схем автоматики: навч. посіб. – К.: НТУУ «КПІ», 2008. – 168 с. ISBN 978-966-622-294-0.
4. Ковальчук О.В., Бур'ян С.О. Застосування різних методів в синтезу для складних програм для логічних програмованих контролерів. *Інформаційний збірник Промелектро. «Промислова електроніка та електротехніка»*. Київ, 2010. №4. С. 51-53.
5. Shimon P. Vingron Logic Circuit Design. Selected Methods. Springer – Verlag Berlin Heidelberg, 2012. 258 P.
6. Бур'ян С.О., Ворощенко В.Ю., Король С.В., Савич О.Ю., Смірнов С.А. Автоматизація формування траєкторій при експериментальному дослідженні електромеханічних систем. *Електротехніка та електроенергетика*. Запоріжжя: ЗНТУ, 2011. №2. С. 30-37.
7. Савич О.Ю., Ворощенко В.Ю., Бур'ян С.О. Синтез логічних виразів для складання програми відпрацювання траєкторії руху для логічного програмованого контролера. *Сучасні проблеми електроенергетехніки та автоматики*: доповіді за матеріалами Міжнародної науково-технічної конференції молодих учених, аспірантів і студентів, м. Київ, грудень. 2010 р. Київ, «Політехніка». С. 362-364.
8. Бур'ян С.О., Савич О.Ю. Порівняння різних методів синтезу відпрацювання заданої траєкторії руху для логічних програмованих контролерів. *Електромеханічні та енергетичні системи, методи моделювання та оптимізації*. Збірник наукових праць ІХ Міжнародної науково-технічної конференції молодих учених і спеціалістів, м. Кременчук, 07-08 квітня 2011 р. Кременчук, КНУ. С. 87-88.
9. Диптан В.В. Бур'ян С.О. Лабораторний стенд для дослідження систем відпрацювання траєкторій на основі програмованого логічного контролера та перетворювача частоти. *Сучасні проблеми електроенергетехніки та автоматики*: доповіді за матеріалами Міжнародної науково-технічної конференції молодих учених, аспірантів і студентів, м. Київ, грудень. 2012 р. Київ, «Політехніка». С. 375-377.