

В. О. ПУЙДЕНКО<sup>1</sup>, В. С. ХАРЧЕНКО<sup>2</sup>

<sup>1</sup>Харківський радіотехнічний коледж

<sup>2</sup>Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ»

## МІНІМІЗАЦІЯ ЛОГІЧНОЇ СХЕМИ ДЛЯ РЕАЛІЗАЦІЇ PSEUDO LRU ШЛЯХОМ МІЖТИПОВОГО ПЕРЕХОДУ У ТРИГЕРНИХ СТРУКТУРАХ

Принцип програмного управління передбачає звертання процесорного ядра до основної пам'яті комп'ютеру за операндами або інструкціями, причому операнди зберігаються у сегментах даних, а інструкції зберігаються у сегментах коду основної пам'яті. Разом з сегментною організацією пам'яті також використовується і сторінкова організація. Сторінкова організація пам'яті завжди відображається на сегментну організацію. Завдяки кешованим циклам процесорного ядра копії сторінок основної пам'яті зберігаються у внутрішній асоціативній кеш-пам'яті. Асоціативна кеш-пам'ять складається з трьох блоків: блоку даних, блоку тегів та блоку LRU. У блоці даних зберігаються операнди або інструкції, блок тегів містить фрагменти адресної інформації, а блок LRU містить логіку політики заміщення рядків. Подія промаху в асоціативній кеш-пам'яті залучає логіку блоку LRU для прийняття рішення про заміщення достовірного рядка у блоці даних. Алгоритм pseudo LRU представляє собою просту і кращу політику заміщення серед відомих політик. В статті реалізовано два варіанти мінімізації апаратури політики заміщення рядків за алгоритмом pseudo LRU для  $q$  – спрямованої асоціативної кеш-пам'яті. В обох варіантах здійснюється перехід від тригерної структури типу синхронний D-тригер до тригерної структури типу синхронний JK-тригер. Перший варіант мінімізації базується на послідовності оновлення бітів блоку LRU за алгоритмом pseudo LRU, що дозволяє скоротити комбінаційну логіку оновлення цих бітів. Другий варіант мінімізації базується на послідовності зміни  $q$  - індексу напрямку внаслідок оновлення бітів блоку LRU за алгоритмом pseudo LRU та додатково дозволяє зменшити кількість елементів пам'яті. Обидва варіанти мінімізації дозволяють підвищити такі характеристики як швидкодію і надійність логічної схеми блоку LRU.

**Ключові слова:** алгоритм pseudo LRU, тип тригерної структури, асоціативна кеш-пам'ять, блок LRU, оцінка складності за Квайном.

### Вступ

Однією із складових значного підвищення продуктивності комп'ютерної системи є організація операційною системою кешованих циклів звертання процесорного ядра до фізичних сторінок в оперативній пам'яті, внаслідок чого копії читаних сторінок залишаються 64-байтними блоками у внутрішній асоціативній кеш-пам'яті процесорного ядра, що в подальших циклах усуває необхідність звертання в оперативній пам'яті. Подія відсутності потрібного 64-байтного блока у кеш-пам'яті (промах), за умови повної достовірності інформації усіх рядків обраної множини блоку даних асоціативної кеш-пам'яті, змушує процесорне ядро приймати рішення про обґрунтоване обрання певного рядка у відповідності з політикою заміщення за алгоритмом pseudo LRU (рис. 2). Політика заміщення LRU [1, 2] є простою і кращою політикою, політика заміщення pseudo LRU [1] набагато простіша за звичайну політику LRU. Політика заміщення pseudo LRU представлена алгоритмом pseudo LRU (рис. 2) і у першому наближенні

описується у вигляді моделі синхронного цифрового автомату (рис. 1), яка складається з комбінаційних схем  $CS_1$ ,  $CS_2$  та елементів пам'яті типу синхронний D-тригер. Робота автомату описується складною перемикальною функцією:

$$B^+ = f(B, \lambda(B)), \quad (1)$$

де  $B \in \{B_0, B_1, \dots, B_{m-1}\}$  – множина внутрішніх станів автомату у поточний момент часу  $t_d$ ;

$B^+ \in \{B_0^+, B_1^+, \dots, B_{m-1}^+\}$  – множина внутрішніх станів автомату у наступний момент часу  $t_{d+1}$ ;

$\lambda(B) \in \{\lambda_0(B), \lambda_1(B), \dots, \lambda_{q-1}(B)\}$  – множина простих перемикальних функцій селекції  $q$ -напрямків;

$$\lambda_r(B), B_i, B_i^+ \in \{0, 1\};$$

$$i \in \{0, 1, 2, \dots, m-1\};$$

$$r \in \{0, 1, 2, \dots, q-1\};$$

$m = q - 1$  – число елементів пам'яті синхронний тригер D-типу;  
 $m \in \mathbb{N}$ ;  
 $q = 2^n$  – число напрямків асоціативної кеш-пам'яті;  
 $n, q \in \mathbb{N}$ .

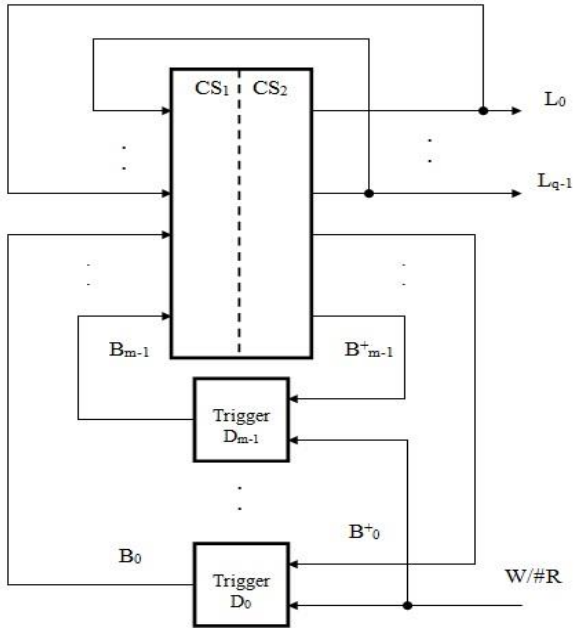


Рис. 1. Модель синхронного цифрового автомату з елементами пам'яті синхронний тригер D-типу

Логічні рівняння (2) – (9) описують роботу декодера q-напрямку в асоціативній кеш-пам'яті.

$$L_0 = \overline{B_3} \& \overline{B_1} \& \overline{B_0}, \quad (2)$$

$$L_1 = B_3 \& \overline{B_1} \& \overline{B_0}, \quad (3)$$

$$L_2 = \overline{B_4} \& B_1 \& \overline{B_0}, \quad (4)$$

$$L_3 = B_4 \& B_1 \& \overline{B_0}, \quad (5)$$

$$L_4 = \overline{B_5} \& \overline{B_2} \& B_0, \quad (6)$$

$$L_5 = B_5 \& \overline{B_2} \& B_0, \quad (7)$$

$$L_6 = \overline{B_6} \& B_2 \& B_0, \quad (8)$$

$$L_7 = B_6 \& B_2 \& B_0. \quad (9)$$

Логічні рівняння (10) – (16) описують логіку оновлення бітів блоку LRU за алгоритмом pseudo LRU.

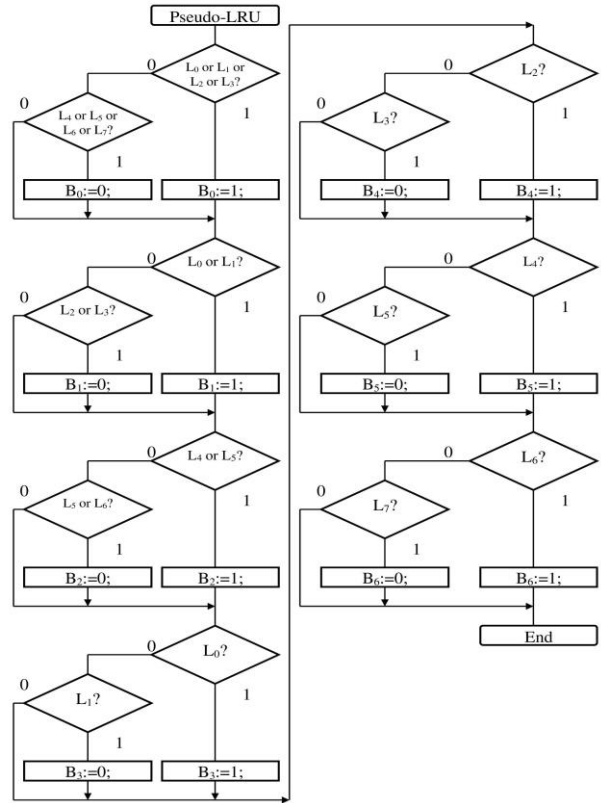


Рис. 2. Алгоритм pseudo LRU з q=8

$$B_0^+ = \overline{L_5} \& \overline{L_7} \& \overline{L_6} \& \overline{L_4} \& B_0 \& \overline{L_3} \& \overline{L_2} \& \overline{L_1} \& \overline{L_0}, \quad (10)$$

$$B_1^+ = \overline{L_3} \& \overline{L_2} \& B_1 \& \overline{L_1} \& \overline{L_0}, \quad (11)$$

$$B_2^+ = \overline{L_7} \& \overline{L_6} \& B_2 \& \overline{L_5} \& \overline{L_4}, \quad (12)$$

$$B_3^+ = \overline{L_1} \& B_3 \& \overline{L_0}, \quad (13)$$

$$B_4^+ = \overline{L_3} \& B_4 \& \overline{L_2}, \quad (14)$$

$$B_5^+ = \overline{L_5} \& B_5 \& \overline{L_4}, \quad (15)$$

$$B_6^+ = \overline{L_7} \& B_6 \& \overline{L_6}. \quad (16)$$

Формула (17) дає можливість оцінити складність апаратури К за Квайном з урахуванням розрядностей компонент комбінаційних схем CS<sub>1</sub>, CS<sub>2</sub> та елементів пам'яті типу синхронний D-тригер:

$$K = \left[ p * \left( 2^p + 2^{p-1} + 1 \right) + 2^p - (p-1) \right] + \left[ n * \left( 2^{2 * n-1} + 2^{n-1} \right) + \left( 2^{n+1} - 1 \right) \right] + \left[ m * 2^{m-3} + \left( 2^{m-1} - 1 \right) \right] + \left[ 2^m + (m-1) * 2^{m-2} \right]. \quad (17)$$

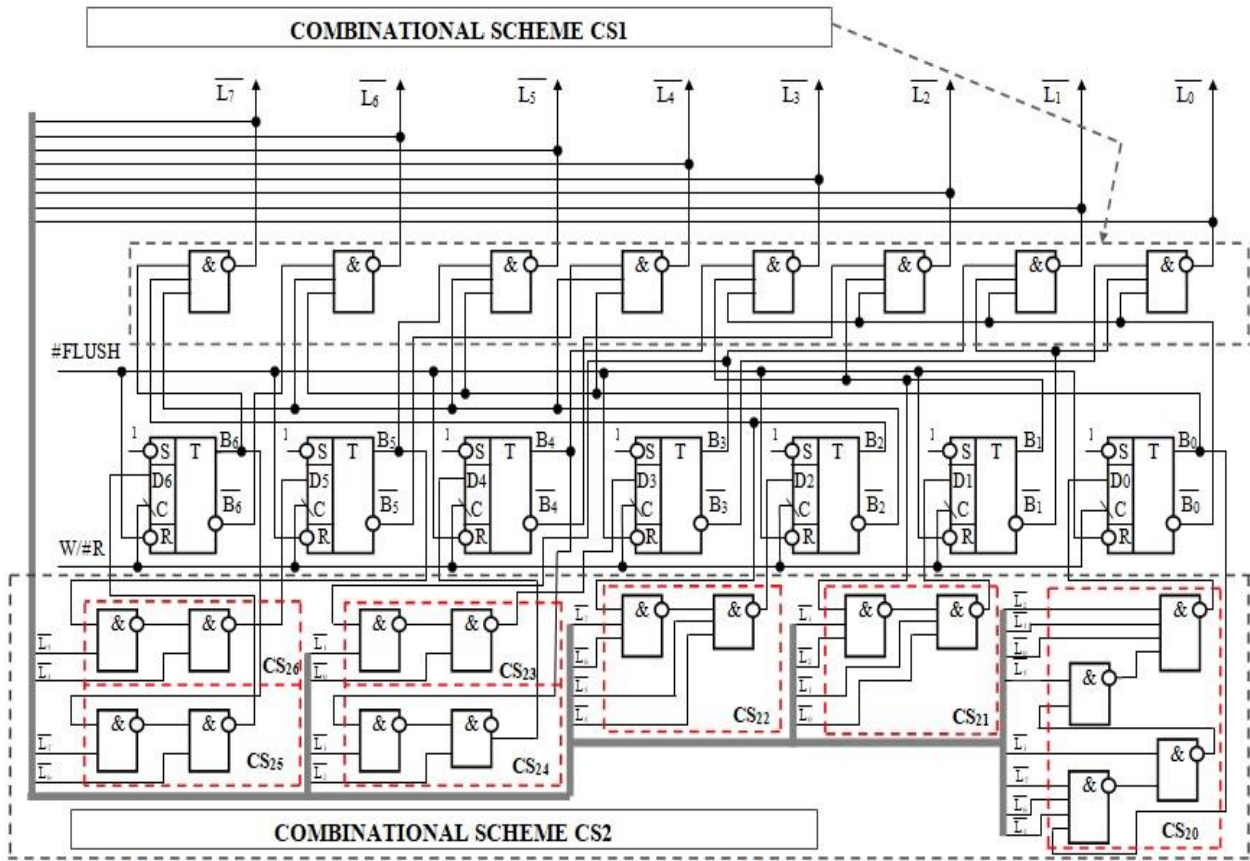


Рис. 3. Мінімальна апаратура політики заміщення pseudo LRU

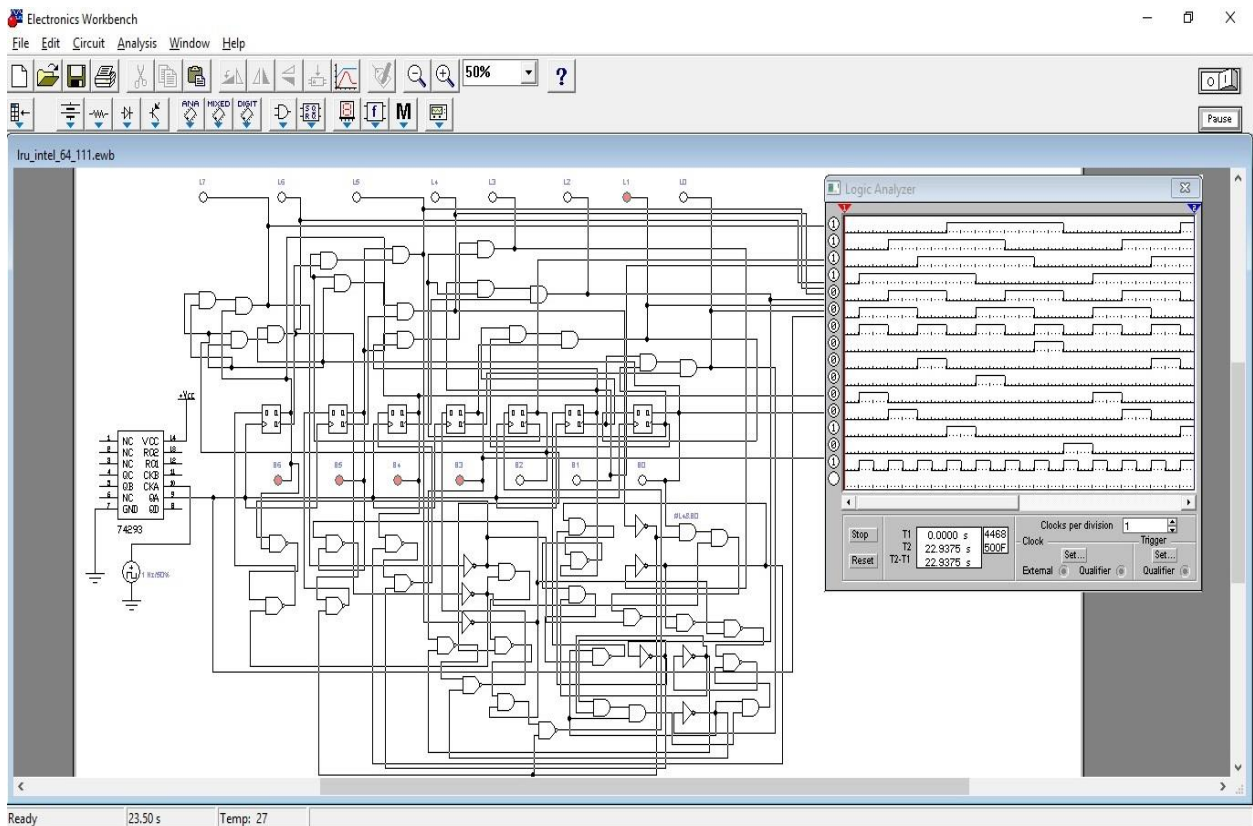


Рис. 4. Комп'ютерна модель мінімальної апаратури політики заміщення pseudo LRU

Діаграма на рисунку 5 демонструє лінійне зростання оцінки складності  $K$  за Квайном при розрядності  $n = 2, p = 3, m = 4$  компонент комбінаційних схем  $CS_1, CS_2$  та елементів пам'яті синхронний тригер D-типу відповідно.

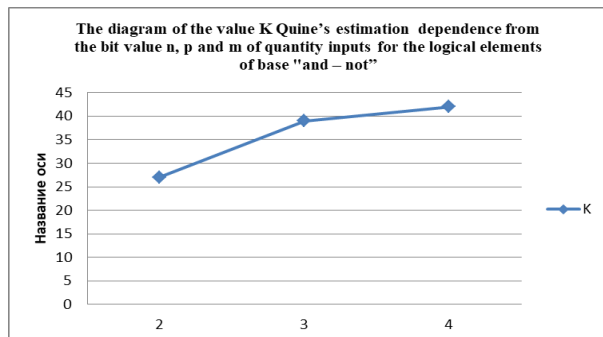


Рис. 5. Діаграма лінійного зростання оцінки складності  $K$  за Квайном при розрядностях електронних компонент  $n = 2, p = 3, m = 4$  відповідно

### Формулювання цілей та задач

Доцільність між типового переходу у тригерних структурах ґрунтується на подібності логічних організацій та однаковій або кращій швидкодії перемикавання. Також слід враховувати таку вагому складову, як оцінку складності за Квайном.

Логічна організація синхронного тригера D-типу показана на рис. 6. Ця логічна організація побудована за схемою трьох тригерів RS-типу. Швидкодія перемикавання такої тригерної структури становить  $4\tau$ , де  $\tau$  – час перемикавання логічного елемента базису «і - ні».

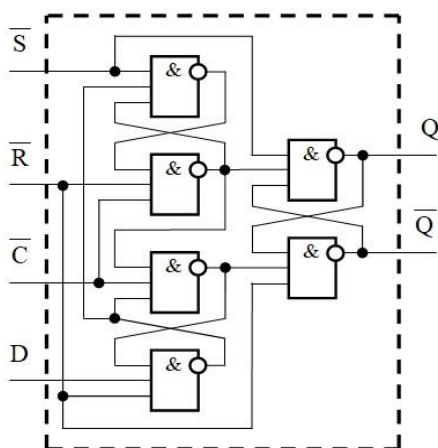


Рис. 6. Логічна організація синхронного тригера D-типу

Логічна організація синхронного тригера JK-типу (рис. 7) подібна логічній організації синх-

ронного тригера D-типу (див. рис. 6). Швидкодія перемикавання тригера JK-типу становить  $3\tau$ , де  $\tau$  – час перемикавання логічного елемента базису «і - ні».

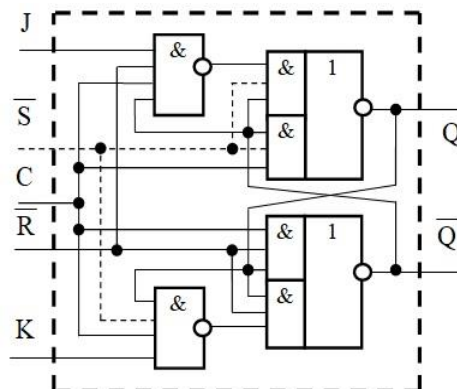


Рис. 7. Логічна організація синхронного тригера JK-типу

Синхронний тригер D-типу має у своїй логічній організації 6 логічних елементів базису «і-ні» на три входи. Отже, ціна за Квайном логічної структури синхронного тригера D-типу з урахуванням входів-виходів інверторів буде такою:

$$K_{D\_тригер} = 3 \cdot 6 + 6 = 24.$$

Синхронний тригер JK-типу у своїй логічній організації містить 2 логічних елементів базису «і-ні» на 4 входи та 2 логічних елементи базису «і-або-ні» на 6 входів. Отже, ціна за Квайном логічної організації синхронного тригера JK-типу з урахуванням входів-виходів інверторів буде такою:

$$K_{JK\_тригер} = 4 \cdot 2 + 2 \cdot 6 + 2 = 18.$$

Розрахунок показує, що оцінювання за Квайном синхронного тригера JK-типу дає меншу складність ніж складність синхронного тригера D-типу:

$$K_{D\_тригер} > K_{JK\_тригер}.$$

Мета цієї статті складається у доведенні доцільності мінімізації логічної схеми pseudo LRU з використанням між типового переходу у тригерних структурах.

### Варіант мінімізації логічної схеми pseudo LRU на базі оновлення бітів В за алгоритмом pseudo LRU

Мінімальна модель автомату (рис. 8) складається з комбінаційної схеми  $CS_1$  та елементів пам'яті типу синхронний JK-тригер. Робота автомату описується перемикальними функціями:

$$B^+ = f(\varphi(B), \psi(B)), \quad (18)$$

$$L = \lambda(B), \quad (19)$$

де  $B \in \{B_0, B_1, \dots, B_{m-1}\}$  – множина внутрішніх станів автомату у поточний момент часу  $t_d$ ;

$B^+ \in \{B_0^+, B_1^+, \dots, B_{m-1}^+\}$  – множина внутрішніх станів автомату у наступний момент часу  $t_{d+1}$ ;

$\varphi(B) \in \{\varphi_0(B), \varphi_1(B), \dots, \varphi_{m-1}(B)\}$  – множина простих перемикальних функцій збудження інформаційних входів J синхронних тригерів;

$\psi(B) \in \{\psi_0(B), \psi_1(B), \dots, \psi_{m-1}(B)\}$  – множина простих перемикальних функцій збудження інформаційних входів K синхронних тригерів;

$\lambda(B) \in \{\lambda_0(B), \lambda_1(B), \dots, \lambda_{q-1}(B)\}$  – множина простих перемикальних функцій селекції напрямків L;

$$\varphi_i(B), \psi_i(B), \lambda_r(B), B_i, B_i^+ \in \{0, 1\};$$

$$i \in \{0, 1, 2, \dots, m-1\};$$

$$r \in \{0, 1, 2, \dots, q-1\};$$

$m = q - 1$  – число синхронних JK-тригерів;

$m \in \mathbb{N}$ ;

$q = 2^n$  – число напрямків асоціативної кеш-пам'яті;

$n, q \in \mathbb{N}$ .

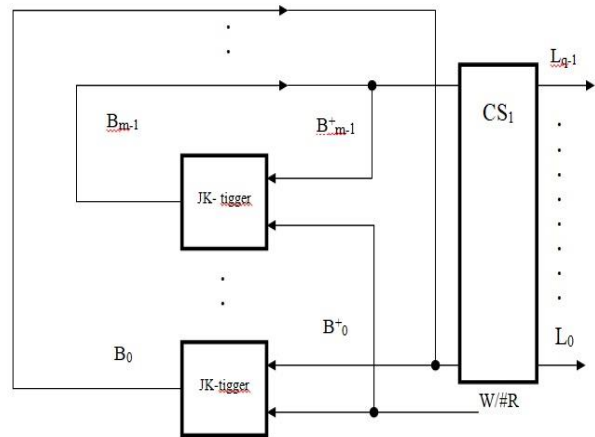


Рис. 8. Мінімальна модель синхронного цифрового автомату за варіантом мінімізації на базі оновлення бітів B за алгоритмом pseudo LRU

Згідно з результатами попередніх досліджень комп'ютерної моделі мінімальної апаратури (див. рис. 4) побудуємо спрямований граф (рис. 9) з вершинами переходів зміни станів бітів  $B_6, B_5, B_4, B_3, B_2, B_1, B_0$  згідно алгоритму pseudo LRU (див. рис. 2) під впливом сигналу запису-читання при умовах промаху або влучанні на циклах звертання:

Перенесемо значення вершин спрямованого графу у відповідні карти Карно відповідних перемикальних функцій (18) збудження інформаційних входів J і K тригерів (рис. 10 – 16).

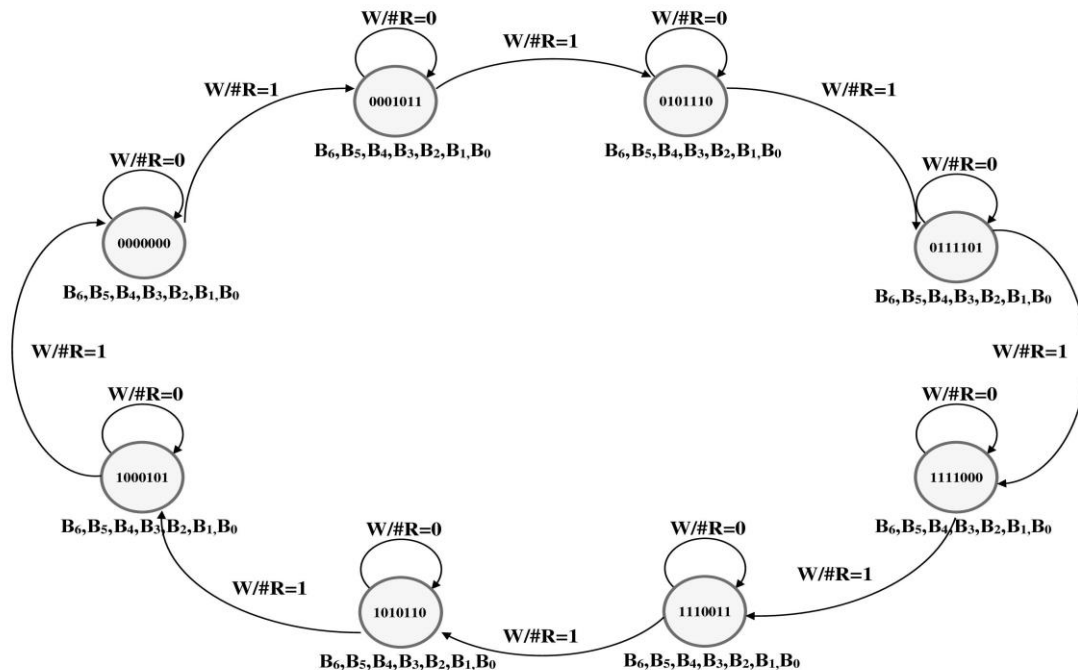


Рис. 9. Спрямований граф переходів зміни стану бітів  $B_6, B_5, B_4, B_3, B_2, B_1, B_0$



		V <sub>4</sub> <sup>+</sup> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>																
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000	
V <sub>4</sub> <sup>+</sup> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	1
	101	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x

		J <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>																
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000	
J <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	101	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x

		K <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
K <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	0
	101	x	x	x	x	1	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Рис. 14. Стани переходів бітів V<sub>4</sub> → V<sub>4</sub><sup>+</sup> неповністю визначених перемикальних функцій збудження

$$J_4 = \varphi_4(B), K_4 = \psi_4(B)$$

		V <sub>6</sub> <sup>+</sup> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>																
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000	
V <sub>6</sub> <sup>+</sup> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x	x	1
	101	x	x	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x

		J <sub>6</sub> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>																
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000	
J <sub>6</sub> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	101	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

		K <sub>6</sub> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
K <sub>6</sub> V <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	0
	101	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	1	x	x	x	x	x	x	x	x	x

Рис. 16. Стани переходів бітів V<sub>6</sub> → V<sub>6</sub><sup>+</sup> неповністю визначених перемикальних функцій збудження

$$J_6 = \varphi_6(B), K_6 = \psi_6(B)$$

		V <sub>5</sub> <sup>+</sup> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
V <sub>5</sub> <sup>+</sup> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	1	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	1	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	0	x	x	x	x	x	x	x	x	x	x	x	x	1
	101	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x

		J <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
J <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	0	x	x	x	x	x	x	x	x	x	x	x	x	1	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	101	x	x	x	x	0	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	0	x	x	x	x	x	x	x	x	x

		K <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>															
		0000	0001	0011	0010	0110	0111	0101	0100	1100	1101	1111	1110	1010	1011	1001	1000
K <sub>5</sub> V <sub>4</sub> V <sub>3</sub> V <sub>2</sub> V <sub>1</sub> V <sub>0</sub>	000	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	001	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	011	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x	x
	010	x	x	x	x	x	x	x	x	x	x	0	x	x	x	x	x
	110	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	111	x	x	1	x	x	x	x	x	x	x	x	x	x	x	x	0
	101	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
	100	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Рис. 15. Стани переходів бітів V<sub>5</sub> → V<sub>5</sub><sup>+</sup> неповністю визначених перемикальних функцій збудження

$$J_5 = \varphi_5(B), K_5 = \psi_5(B)$$

Знайдені відповідні контури на певних картах Карно функцій збудження (див. рис. 10 – 16) дадуть мінімальні логічні рівняння (20) – (26), за якими побудований фрагмент мінімальної логічної схеми pseudo LRU (рис. 17):

$$J_0 = \varphi_0(B) = K_0 = \psi_0(B) = 1; \tag{20}$$

$$J_1 = \varphi_1(B) = \overline{B_2}; K_1 = \psi_1(B) = B_2 \tag{21}$$

$$J_2 = \varphi_2(B) = B_1; K_2 = \psi_2(B) = \overline{B_1} \tag{22}$$

$$J_3 = \varphi_3(B) = \overline{B_6}; K_3 = \psi_3(B) = B_6; \tag{23}$$

$$J_4 = \varphi_4(B) = B_5; K_4 = \psi_4(B) = \overline{B_5}; \tag{24}$$

$$J_5 = \varphi_5(B) = B_3; K_5 = \psi_5(B) = \overline{B_3}; \tag{25}$$

$$J_6 = \varphi_6(B) = B_4; K_6 = \psi_6(B) = \overline{B_4}; \tag{26}$$

Дослідження комп'ютерної моделі отриманої мінімальної логічної схеми pseudo LRU дає можливість спостерігати послідовність відновлень значень бітів V<sub>6</sub>, V<sub>5</sub>, V<sub>4</sub>, V<sub>3</sub>, V<sub>2</sub>, V<sub>1</sub>, V<sub>0</sub>: 0000001, 0001010, 0101111, 0111100, 1111001, 1110010, 1010111, 1000100. Отримані значення забезпечують селекцію напрямків асоціативної кеш-пам'яті L<sub>4</sub> → L<sub>2</sub> → L<sub>6</sub> → L<sub>1</sub> → L<sub>5</sub> → L<sub>3</sub> → L<sub>7</sub> → L<sub>0</sub> при умовах промаху або влучання на циклах звертання.

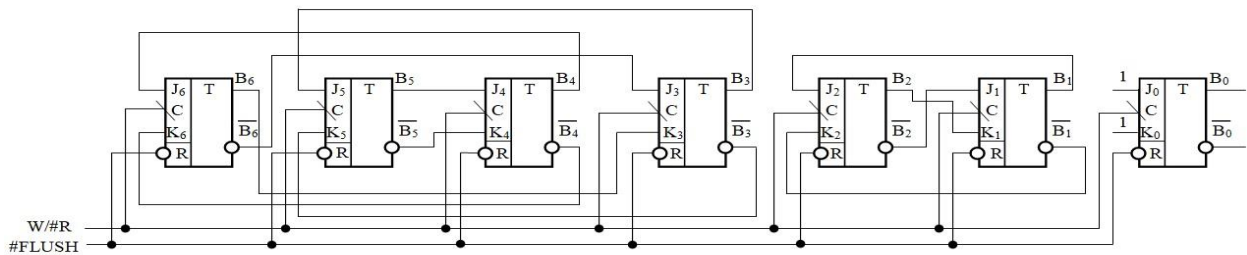


Рис. 17. Фрагмент мінімальної логічної схеми pseudo LRU за варіантом мінімізації логічної схеми pseudo LRU на базі оновлення бітів В за алгоритмом pseudo LRU

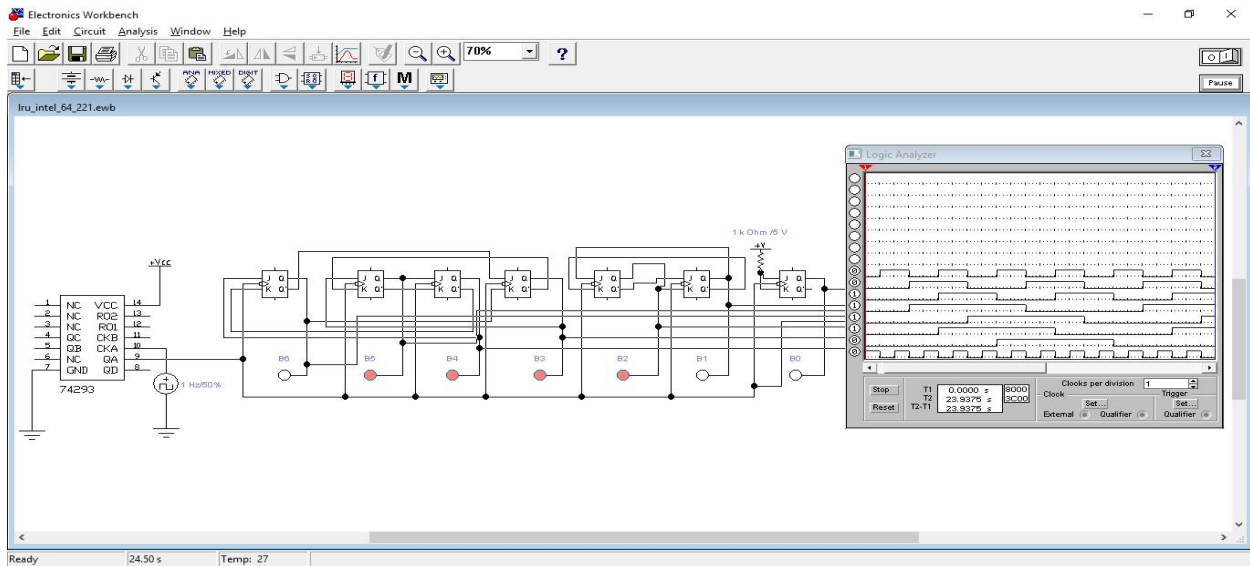


Рис. 18. Комп'ютерна модель фрагменту мінімальної логічної схеми pseudo LRU за варіантом мінімізації на базі оновлення бітів В за алгоритмом pseudo LRU

Якщо перенести послідовність відновлень значень бітів  $B_6, B_5, B_4, B_3, B_2, B_1, B_0$ , отриманих в результаті дослідження комп'ютерної моделі (рис. 18) у таблицю 1, то виділивши стани бітів  $B_3, B_2, B_1$  можна побачити аргументи перемикальних функцій  $\lambda_0(B), \lambda_1(B), \dots, \lambda_{q-1}(B)$  роботи декодера  $q = 2^n$  селекції напрямків  $L, n = 3$ .

Таблиця 1

Логіка роботи декодера  $q = 2^n, n = 3$

Біти LRU							Напрямок $L = \lambda(B)$
$B_6$	$B_5$	$B_4$	$B_3$	$B_2$	$B_1$	$B_0$	
1	0	0	0	1	0	1	$L_0 = \lambda_0(B)$
0	1	1	1	1	0	1	$L_1 = \lambda_1(B)$
0	0	0	1	0	1	1	$L_2 = \lambda_2(B)$
1	1	1	0	0	1	1	$L_3 = \lambda_3(B)$
0	0	0	0	0	0	0	$L_4 = \lambda_4(B)$
1	1	1	1	0	0	0	$L_5 = \lambda_5(B)$
0	1	0	1	1	1	0	$L_6 = \lambda_6(B)$
1	0	1	0	1	1	0	$L_7 = \lambda_7(B)$

У відповідності з таблицею 1 мінімальні логічні рівняння селекції напрямків  $L_4 \rightarrow L_2 \rightarrow L_6 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3 \rightarrow L_7 \rightarrow L_0$  набудуть вигляду:

$$L_0 = \lambda_0(B) = \overline{B_3} \& \overline{B_2} \& \overline{B_1}, \tag{27}$$

$$L_1 = \lambda_1(B) = B_3 \& B_2 \& \overline{B_1}, \tag{28}$$

$$L_2 = \lambda_2(B) = B_3 \& \overline{B_2} \& B_1, \tag{29}$$

$$L_3 = \lambda_3(B) = \overline{B_3} \& \overline{B_2} \& B_1, \tag{30}$$

$$L_4 = \lambda_4(B) = \overline{B_3} \& \overline{B_2} \& \overline{B_1}, \tag{31}$$

$$L_5 = \lambda_5(B) = B_3 \& \overline{B_2} \& \overline{B_1}, \tag{32}$$

$$L_6 = \lambda_6(B) = B_3 \& B_2 \& B_1, \tag{33}$$

$$L_7 = \lambda_7(B) = \overline{B_3} \& B_2 \& B_1. \tag{34}$$

З урахуванням (20) – (34) представимо повну мінімальну логічну схему pseudo LRU та її комп'ютерну модель (рис. 19, 20), де спостерігається відсутність комбінаційної схеми  $CS_2$ .



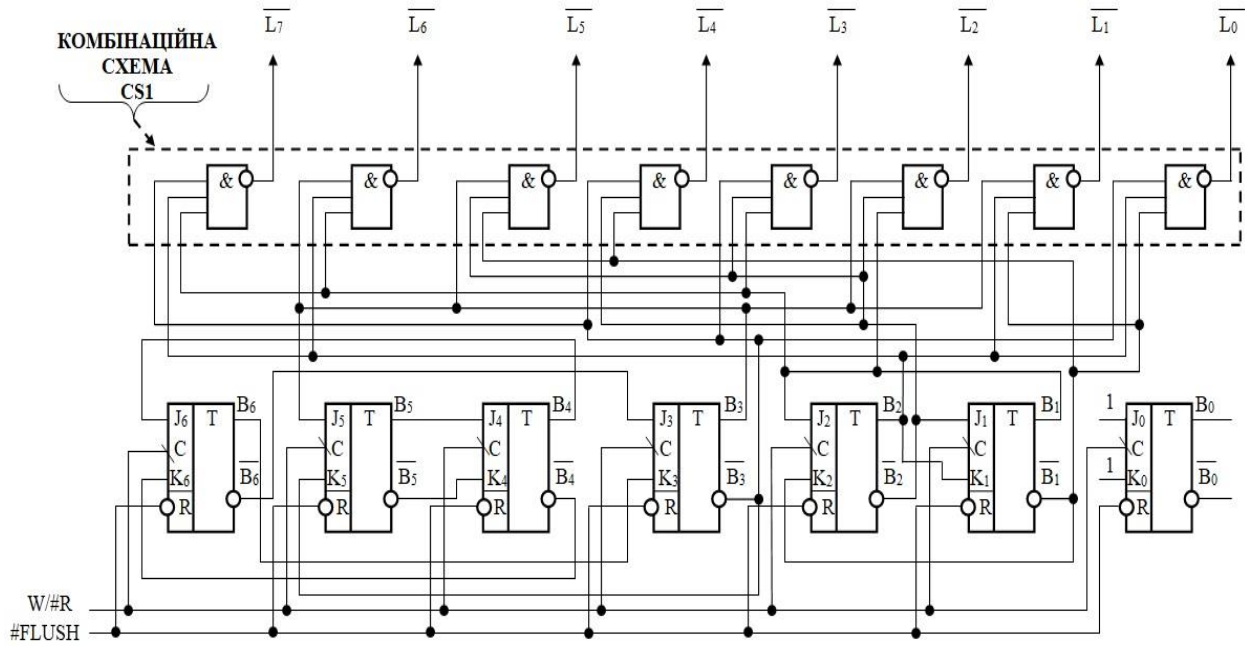


Рис. 19. Мінімальна логічна схема pseudo LRU за варіантом мінімізації на базі оновлення бітів В за алгоритмом pseudo LRU

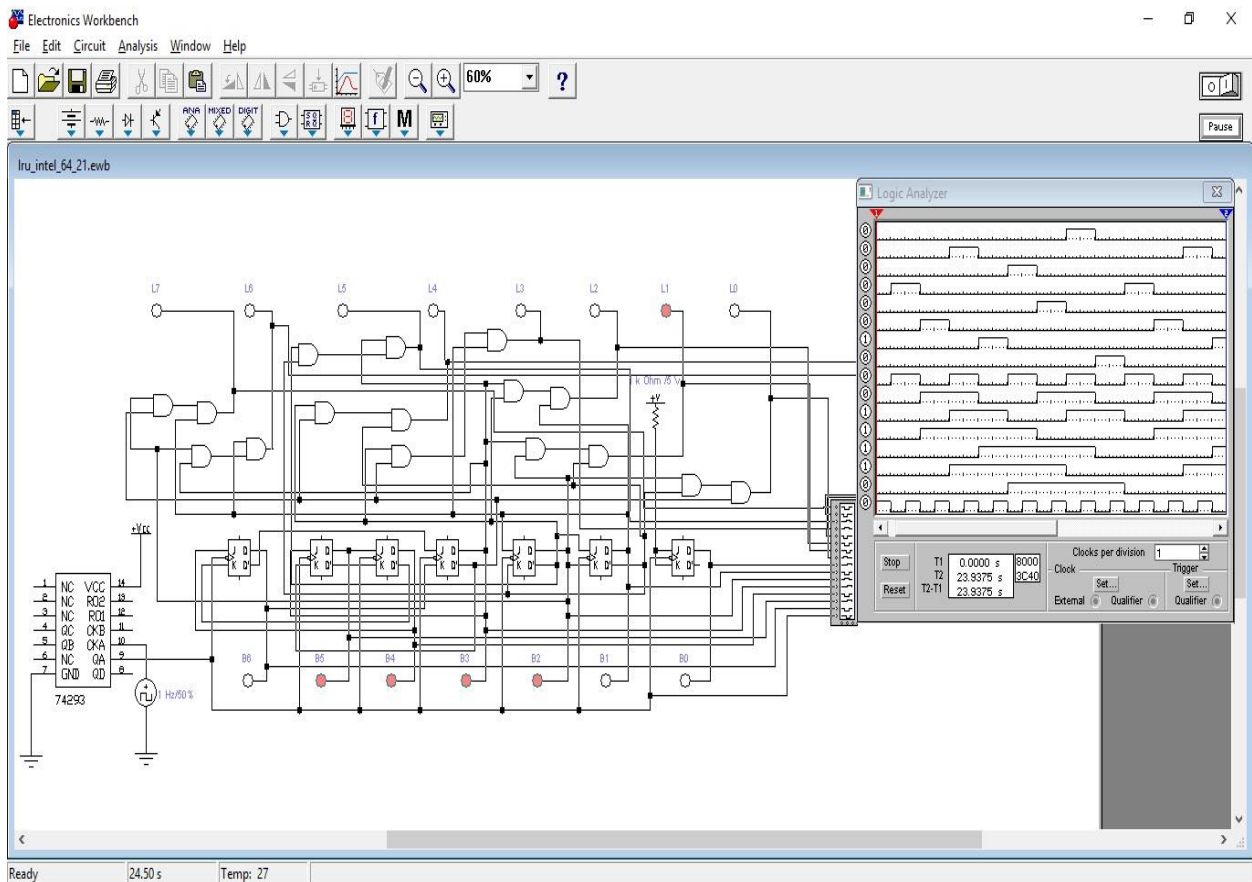


Рис. 20. Комп'ютерна модель мінімальної логічної схеми pseudo LRU за варіантом мінімізації на базі оновлення бітів В за алгоритмом pseudo LRU

### Варіант мінімізації логічної схеми pseudo LRU на базі послідовності зміни q - індексу напрямку

Мінімальна модель автомату (рис. 21) складається з комбінаційної схеми CS<sub>1</sub> та елементів пам'яті синхронний тригер JK-типу. Робота автомату описується перемикальними функціями:

$$Q^+ = f(\varphi(Q), \psi(Q)), \quad (35)$$

$$L = \lambda(Q), \quad (36)$$

де  $Q \in \{Q_0, Q_1, \dots, Q_{m-1}\}$  – множина внутрішніх станів автомату у поточний момент часу  $t_d$ ;

$Q^+ \in \{Q_0^+, Q_1^+, \dots, Q_{m-1}^+\}$  – множина внутрішніх станів автомату у наступний момент часу  $t_{d+1}$ ;

$\varphi(Q) \in \{\varphi_0(Q), \varphi_1(Q), \dots, \varphi_{m-1}(Q)\}$  – множина простих перемикальних функцій збудження інформаційних входів  $J$  синхронних тригерів;

$\psi(Q) \in \{\psi_0(Q), \psi_1(Q), \dots, \psi_{m-1}(Q)\}$  – множина простих перемикальних функцій збудження інформаційних входів  $K$  синхронних тригерів;

$\lambda(Q) \in \{\lambda_0(Q), \lambda_1(Q), \dots, \lambda_{q-1}(Q)\}$  – множина простих перемикальних функцій селекції q - індексу напрямків  $L$ ;

$$\varphi_i(Q), \psi_i(Q), \lambda_r(Q), Q_i, Q_i^+ \in \{0, 1\};$$

$$i \in \{0, 1, 2, \dots, m-1\};$$

$$r \in \{0, 1, 2, \dots, q-1\};$$

$m'$  – число елементів пам'яті типу синхронний JK-тригер;

$q = 2^{m'}$  – число напрямків асоціативної кеш-пам'яті;  
 $m', q \in N$ .

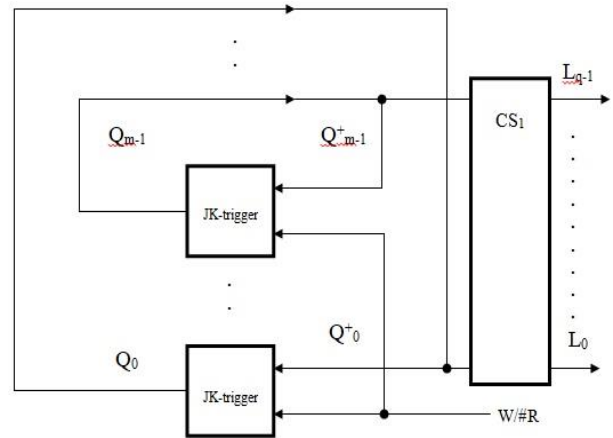


Рис. 21. Мінімальна модель синхронного цифрового автомату за варіантом мінімізації на базі послідовності зміни q - індексу напрямку

В цьому варіанті аргументами перемикальних функцій  $\lambda_0(Q), \lambda_1(Q), \dots, \lambda_{q-1}(Q)$  роботи декодера

$q = 2^{m'}$  селекції напрямків є значення внутрішнього стану автомату  $Q_0, Q_1, \dots, Q_{m-1}$  (табл. 2) (збережена комбінаційна логіка CS<sub>1</sub>) (див. рис. 3). Вершини графу (рис. 22) описують значення внутрішнього стану автомату при  $m' = 3$ , які несуть інформацію про зміну q - індексу напрямків  $L$  у відповідності з попереднім дослідженням комп'ютерної моделі (рис. 20).

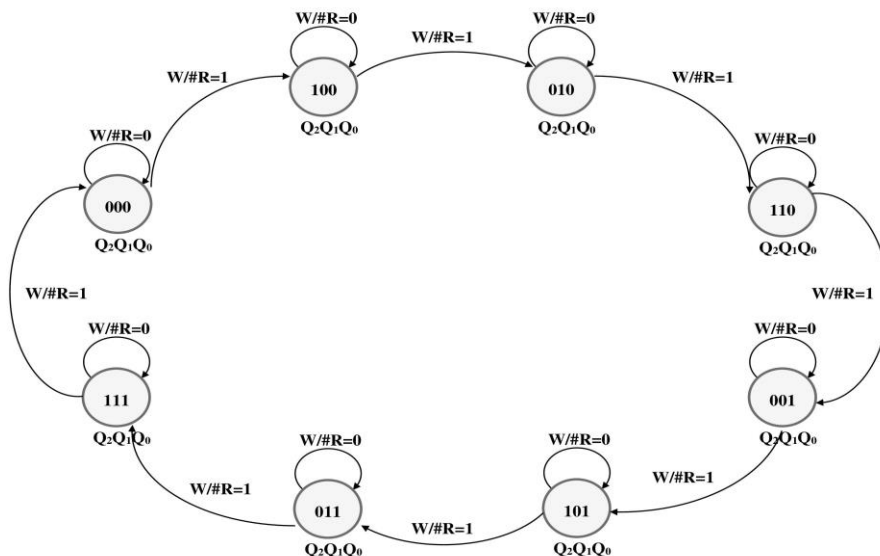


Рис. 22. Спрямований граф переходів зміни q - індексу напрямку  $L$  за станом бітів з  $Q_0, Q_1, \dots, Q_{m-1}$  з  $m' = 3$

Таблиця 2

Логіка роботи декодера  $q = 2^{m'}$ ,  $m' = 3$

Аргументи простої перемикальної функції $\lambda(Q)$			Напрямок $L_q = \lambda(Q)$							
			$q$							
$Q_2$	$Q_1$	$Q_0$	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	1	0	0	0	0	0	0	1	0	0
0	1	1	0	0	0	0	1	0	0	0
1	0	0	0	0	0	1	0	0	0	0
1	0	1	0	0	1	0	0	0	0	0
1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0

Перенесемо значення вершин спрямованого графу (рис. 22) у відповідні карти Карно перемикальної функції (35) збудження інформаційних входів J і K синхронних тригерів (рис. 23):

Знайдені відповідні контури на відповідних картах Карно функцій збудження (рис. 23) дадуть наступні мінімальні логічні рівняння:

$$L_0 = \lambda_0(Q) = \overline{Q_2} \& \overline{Q_1} \& \overline{Q_0}, \quad (37)$$

$$L_1 = \lambda_1(Q) = \overline{Q_2} \& \overline{Q_1} \& Q_0, \quad (38)$$

$$L_2 = \lambda_2(Q) = \overline{Q_2} \& Q_1 \& \overline{Q_0}, \quad (39)$$

$$L_3 = \lambda_3(Q) = \overline{Q_2} \& Q_1 \& Q_0, \quad (40)$$

$$L_4 = \lambda_4(Q) = Q_2 \& \overline{Q_1} \& \overline{Q_0}, \quad (41)$$

$$L_5 = \lambda_5(Q) = Q_2 \& \overline{Q_1} \& Q_0, \quad (42)$$

$$L_6 = \lambda_6(Q) = Q_2 \& Q_1 \& \overline{Q_0}, \quad (43)$$

$$L_7 = \lambda_7(Q) = Q_2 \& Q_1 \& Q_0, \quad (44)$$

$$J_0 = \varphi_0(Q) = K_0 = \psi_0(Q) = Q_2 \& Q_1, \quad (45)$$

$$J_1 = \varphi_1(Q) = K_1 = \psi_1(Q) = Q_2, \quad (46)$$

$$J_2 = \varphi_2(Q) = K_2 = \psi_2(Q) = 1. \quad (47)$$

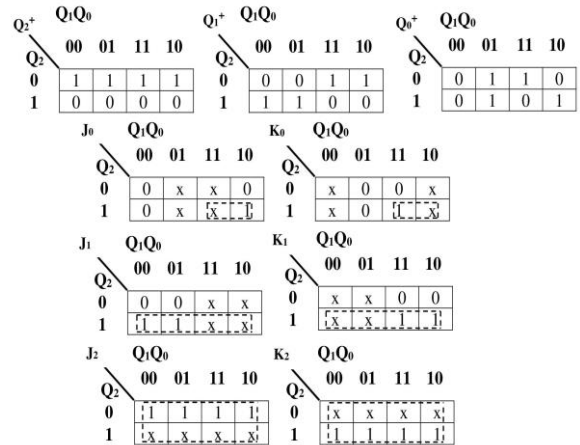


Рис. 23. Стани переходів бітів

$Q_2, Q_1, Q_0 \rightarrow Q_2^+, Q_1^+, Q_0^+$  неповністю визначених перемикальних функцій збудження  $J_0 = \varphi_0(Q)$ ,

$$K_0 = \psi_0(Q); J_1 = \varphi_1(Q), K_1 = \psi_1(Q); J_2 = \varphi_2(Q), K_2 = \psi_2(Q)$$

З урахуванням (37) - (47) представимо повну мінімальну логічну схему та її комп'ютерну модель (рис. 24, 25), у якій відсутня комбінаційної схеми CS<sub>2</sub>, а кількість синхронних тригерів JK типу зменшена з 7-ми до 3-х.

В процесі дослідження спостерігається зміна стану виходів  $Q_2, Q_1, Q_0 \rightarrow Q_2^+, Q_1^+, Q_0^+$ :  $100 \rightarrow 010 \rightarrow 110 \rightarrow 001 \rightarrow 101 \rightarrow 011 \rightarrow 111 \rightarrow 000$ , що відповідає послідовності обрання рядків  $L_4 \rightarrow L_2 \rightarrow L_6 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3 \rightarrow L_7 \rightarrow L_0$  певної множини.

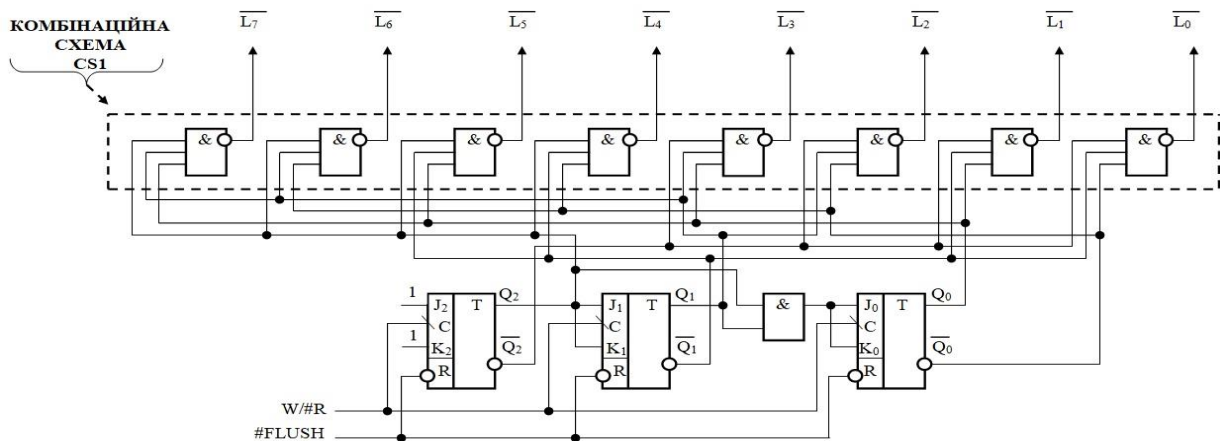


Рис. 24. Мінімальна логічна схема pseudo LRU за варіантом мінімізації її на базі послідовності зміни  $q$  - індексу напрямку

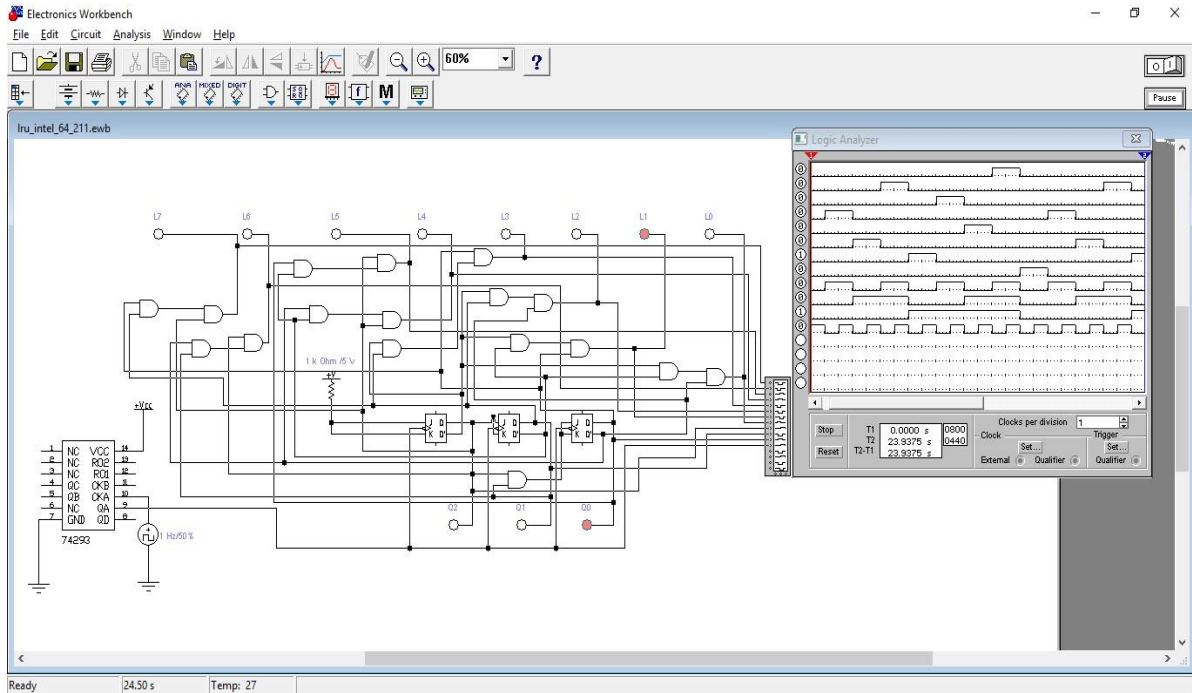


Рис. 25. Комп'ютерна модель мінімальної логічної схеми pseudo LRU за варіантом мінімізації на базі послідовності зміни  $q$  - індексу напрямку

### Порівняння складності і швидкодії варіантів мінімізації

Для обох варіантів мінімізації загальне оцінювання складності за Квайном виглядатиме як:

$$K = K_{cs1} + K_{JK}, \quad (48)$$

де  $K_{cs1}$  – оцінка складності за Квайном комбінаційної схеми  $CS_1$ ,  $K_{JK}$  – оцінка складності за Квайном синхронних тригерів JK-типу.

Обчислимо оцінку складності за Квайном для варіанту мінімізації на базі оновлення бітів  $B$  за алгоритмом pseudo LRU з урахуванням розрядності компонент.

Для комбінаційної схеми  $CS_1$  оцінка складності за Квайном буде такою:

$$K_{cs1} = p * 2^p + I, \quad (49)$$

де  $p = 3$  – розрядність елемента базису «і-ні»,  $I = 3$  – урахування інверсних виходів – входів.

Для синхронних тригерів JK-типу оцінка складності за Квайном буде такою:

$$K_{JK} = 2^m + (m-1) * 2^{m-2}, \quad (50)$$

де  $m = 4$  – кількість входів синхронного тригеру JK-типу.

На підставі виразів (48) – (50) діаграма на рисунку 26 дозволяє спостерігати лінійне зростання складності з розрядністю компонент:

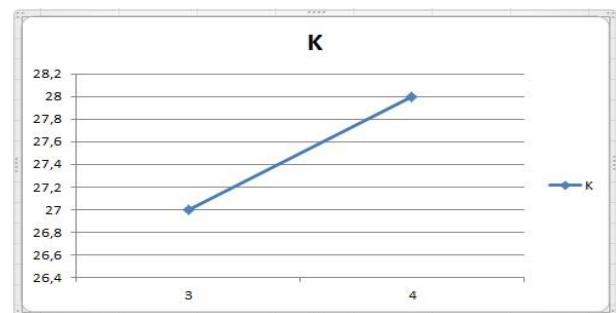


Рис. 26. Зростання складності за Квайном при  $p = 3, m = 4$

Далі обчислимо оцінку складності за Квайном для варіанту мінімізації на базі послідовності зміни  $q$  - індексу напрямку.

Для комбінаційної схеми  $CS_1$  оцінка складності за Квайном буде такою:

$$K_{cs1} = p * 2^p + I, \quad (51)$$

де  $p = 3$  – розрядність логічного елемента базису «і-ні»,  $I = 3$  – урахування інверсних виходів – входів.

Для синхронних тригерів JK-типу оцінка складності за Квайном з урахуванням одного елементу базису «і» та два входи буде такою:

$$K_{JK} = (m - 2) * 2^{m-4} + 2^{m-1} + 2^{m-2}, \quad (52)$$

де  $m = 4$  – кількість входів синхронного тригеру JK-типу,  $n = m - 2$  – розрядність логічного елементу базису «і».

На підставі виразів (48), (51), (52) діаграма на рисунку 27 дозволяє спостерігати лінійне спадання складності з розрядністю компонент  $p = 3, m = 4$ .

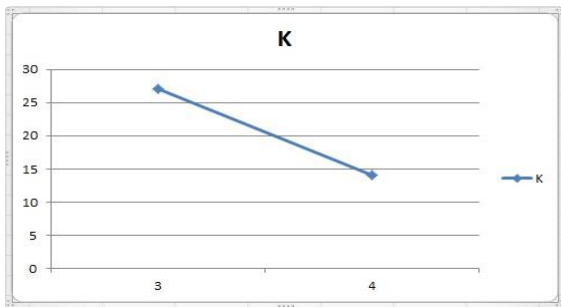


Рис. 27. Спадання складності за Квайном при  $p = 3, m = 4$

Мінімізація логічної схеми pseudo LRU з обґрунтованим переходом від синхронних тригерів D-типу на синхронні тригери JK-типу (на базі оновлення бітів В за алгоритмом pseudo LRU та на базі послідовності зміни q - індексу напрямку) дала змогу позбутися комбінаційної логіки CS<sub>2</sub> (рис. 19, 24), що дозволяє досягнути швидкодії зміни станів бітів  $V_6, V_5, V_4, V_3, V_2, V_1, V_0$  у  $3\tau$  (швидкодія мінімальної апаратури на рис. 3 складає  $7\tau$ :  $4\tau$  – швидкодія найбільш повільної компоненти CS<sub>20</sub> комбінаційної логіки CS<sub>2</sub>,  $3\tau$  – швидкодія перемикання синхронного тригера D-типу).

### Висновки

Мінімальні логічні схеми pseudo LRU, які отримані двома варіантами мінімізації, повністю задовольняють політиці алгоритму заміщення pseudo LRU через досліджені діаграми часу у комп'ютерному середовищі моделювання. Дослідження комп'ютерної моделі мінімальної логічної схеми pseudo LRU на базі оновлення бітів В за алгоритмом pseudo LRU дає можливість спостерігати послідовність відновлень значень бітів  $V_6, V_5, V_4, V_3, V_2, V_1, V_0$  – виходів тригерів JK-типу : 0000001, 0001010, 0101111, 0111100, 1111001, 1110010, 1010111, 1000100. Ця послідовність зміни станів забезпечують селекцію напрямків асоціативної кеш-пам'яті

$$L_4 \rightarrow L_2 \rightarrow L_6 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3 \rightarrow L_7 \rightarrow L_0$$

на циклах звертання при умовах промаху або влучанні.

Усунення комбінаційної логіки CS<sub>2</sub> в результаті мінімізації логічної схеми на базі оновлення бітів В за алгоритмом pseudo LRU дозволяє досягти швидкодію у  $3\tau$  і підвищити надійність пристрою у цілому (з урахуванням заміни 7 синхронних тригерів D-типу на 7 синхронних тригерів JK-типу, вважаючи майже рівною їх надійність).

Дослідження комп'ютерної моделі мінімальної логічної схеми pseudo LRU за варіантом мінімізації на базі послідовності зміни q - індексу напрямку дає можливість спостерігати зміну станів виходів  $Q_2, Q_1, Q_0 \rightarrow Q_2^+, Q_1^+, Q_0^+$  тригерів JK-типу:  $100 \rightarrow 010 \rightarrow 110 \rightarrow 001 \rightarrow 101 \rightarrow 011 \rightarrow 111 \rightarrow 000$  – зміни q - індексу напрямку, як слідства оновлення бітів  $V_6, V_5, V_4, V_3, V_2, V_1, V_0$ , що відповідає послідовності обрання рядків

$$L_4 \rightarrow L_2 \rightarrow L_6 \rightarrow L_1 \rightarrow L_5 \rightarrow L_3 \rightarrow L_7 \rightarrow L_0$$

певної множини на циклах звертання при умовах промаху або влучанні.

Мінімізація логічної схеми блоку LRU на базі послідовності зміни q - індексу напрямку забезпечує скорочення кількості синхронних тригерів JK-типу (з 7-ми до 3-х), що дає змогу додатково підвищити надійність пристрою при швидкодії у  $3\tau$ .

Мінімізації логічної схеми блоку LRU на базі послідовності зміни q - індексу напрямку дає замість формули визначення кількості синхронних тригерів JK-типу  $m = q - 1$  нову формулу визначення кількості синхронних тригерів JK-типу  $m' = \log_2 q$ .

Синтезовані мінімальні логічні схеми pseudo LRU є апаратними, що безперечно має переваги перед програмним VHDL – кодом [2] за продуктивністю: відновлення бітів  $V_6, V_5, V_4, V_3, V_2, V_1, V_0$  у  $3\tau$  та селекції відповідного рядка множини асоціативної кеш – пам'яті на 8 напрямків у  $\tau$ , де  $\tau$  – час перехідного процесу перемикання логічного елементу базису «і – ні».

Синтезовані мінімальні логічні схеми pseudo LRU значно простіші на відміну від апаратних рішень реалізації LRU [2], де зокрема певної комбінаційної логіки додатково задіяні такі складні цифрові пристрої, як лічильники, регістри та компаратори, що негативно впливає на такі характеристики, як: простота реалізації, продуктивність і надійність.

## Література

1. Omran, Safaa S. *Implementation of LRU Replacement Policy for Reconfigurable Cache Memory Using FPGA [Text]* / Safaa S. Omran, Ibrahim A. Amory // *International Conference on Advanced Science and Engineering, Kurdistan Region, Iraq, November, 12-14, 2018.* – P. 13-18.
2. Sudarshan, T. S. B. *Highly Efficient LRU Implementations for High Associativity Cache Memory [Electronic resource]* / T. S. B. Sudarshan, Rahil Abbas Mir, S. Vijayalakshmi // *Birla Institute of Technology and Science, Pilani, Rajasthan 330331 INDIA.* – Available at: <http://www.semanticscholar.org/paper/Highly-efficient-LRU-implementations-for-high-cache-Sudarshan-Mir/e9a6b5b9cb70fc3782b2709ebcf1414051ed6e4c> (Accessed 2004)
3. Puidenko, V. *The Minimizing of Hardware for Implementation of Pseudo LRU Algorithm for Cache Memory [Text]* / Vadym Puidenko, Vyacheslav Kharchenko // *The 11th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2020 14-18 May, 2020, Kyiv, Ukraine.* – P. 63-71.
4. *A Computationally Efficient P-LRU based Optimal Cache Heap Object Replacement Policy [Text]* / Burhan Ul Islam Khan, Rashidah F. Olanrewaju, Roohie Naaz Mir, Abdul Raouf Khan, S. H. Yusoff // *International Journal of Advanced Computer Science and Applications.* – 2017. – Vol. 8, No. 1. – P. 128-138.
5. Kumar, Swadhesh. *An Overview of Modern Cache Memory and Performance Analysis of Replacement Policies [Text]* / Swadhesh Kumar, P. K. Singh // *2nd IEEE International Conference on Engineering and Technology, India, 2016.* – P. 4145-4148.
6. Alghazo, Jaafar. *Cache Replacement Algorithm Records" [Text]* / Jaafar Alghazo, Adil Akaaboune, Nazeih Botros // *International Workshop on Memory Technology, Design and Testing, Illinois, USA, August, 2004.* – P. 19-24.
7. *Timing predictability of cache replacement policies [Text]* / J. Reineke, D. Grund, C. Berg, R. Wilhelm // *Real-Time Syst.* – 2007. – vol. 37, no. 2. – P. 99-122.
8. Пухальський, Г. І. *Проектирование дискретных устройств на интегральных микросхемах [Текст] : Справочник / Г. И. Пухальский, Т. Я. Новосельцева.* – М. : Радио и связь, 1990. – 304 с. – С. 61-91.

Поступила в редакцію 12.03.2020, рассмотрена на редколлегии 15.04.2020

## МИНИМИЗАЦИЯ ЛОГИЧЕСКОЙ СХЕМЫ ДЛЯ РЕАЛИЗАЦИИ PSEUDO LRU ПУТЕМ МЕЖТИПОВОГО ПЕРЕХОДА В ТРИГГЕРНЫХ СТРУКТУРАХ

**В. А. Пуйденко, В. С. Харченко**

Принцип программного управления предполагает обращение процессорного ядра к основной памяти компьютера за операндами или инструкциями. При этом операнды хранятся в сегментах данных, а инструкции хранятся в сегментах кода основной памяти. Наряду с сегментной организацией памяти также используется и страничная организация памяти. Страничная организация памяти всегда отображается на сегмент-

## References

1. Omran, Safaa S., Amory, Ibrahim A. *Implementation of LRU Replacement Policy for Reconfigurable Cache Memory Using FPGA. International Conference on Advanced Science and Engineering, Kurdistan Region, Iraq, November, 12-14, 2018.* pp. 13-18.
2. Sudarshan, T. S. B., Mir, Rahil Abbas., Vijayalakshmi, S. *Highly Efficient LRU Implementations for High Associativity Cache Memory. Birla Institute of Technology and Science, Pilani, Rajasthan 330331 INDIA, 2017.* Available at: <http://www.semanticscholar.org/paper/Highly-efficient-LRU-implementations-for-high-cache-Sudarshan-Mir/e9a6b5b9cb70fc3782b2709ebcf1414051ed6e4c> (Accessed 2004)
3. Puidenko, Vadym., Kharchenko, Vyacheslav. *The Minimizing of Hardware for Implementation of Pseudo LRU Algorithm for Cache Memory. The 11th IEEE International Conference on Dependable Systems, Services and Technologies, DESSERT'2020, 14-18 May, 2020, Kyiv, Ukraine,* pp. 63-71.
4. Khan, Burhan Ul Islam., Olanrewaju, Rashidah F., Mir, Roohie Naaz., Khan, Abdul Raouf., Yusoff, S. H. *A Computationally Efficient P-LRU based Optimal Cache Heap Object Replacement Policy. International Journal of Advanced Computer Science and Applications,* vol. 8, no. 1, 2017, pp. 128-138.
5. Kumar, Swadhesh., Singh, P. K. *An Overview of Modern Cache Memory and Performance Analysis of Replacement Policies. 2nd IEEE International Conference on Engineering and Technology, India, 2016,* pp. 4145-4148.
6. Alghazo, Jaafar., Akaaboune, Adil., Botros, Nazeih. *Cache Replacement Algorithm Records. International Workshop on Memory Technology, Design and Testing, Illinois, USA, August, 2004,* pp. 19-24.
7. Reineke, J., Grund, D., Berg, C., Wilhelm, R. *Timing predictability of cache replacement policies. Real-Time Syst.,* vol. 37, no. 2, Nov. 2007, pp. 99-122.
8. Pukhalskiy, G. I, Novoseltseva, T. Ya. *Proektirovanie diskretnykh ustroystv na integral'nykh mikroskhemakh: Spravochnik [Designing discrete devices to integrated circuits: Directory].* Moscow, Radio i svyaz' Publ., 1990. 304 p., pp. 61-91.

ную організацію. Благодаря кэшируемым пакетным циклам процессорного ядра копии страниц основной памяти сохраняются во внутренней ассоциативной кэш-памяти. Ассоциативная кэш-память состоит из трёх блоков: блока данных, блока тегов и блока LRU. В блоке данных хранятся операнды или инструкции, блок тэгов содержит фрагменты адресной информации, а блок LRU содержит логику политики замещения строк. Событие промаха в ассоциативной кэш-памяти привлекает логику блока LRU для принятия решения о замещении достоверной строки в блоке данных. Алгоритм pseudo LRU представляет собой простую и лучшую политику замещения среди известных политик. В статье реализованы два варианта минимизации аппаратуры политики замещения строк по алгоритму pseudo LRU для  $q$  – направленной ассоциативной кэш-памяти. Для обоих вариантов обоснован переход от триггерной структуры типа синхронный D - триггер к триггерной структуре типа синхронный JK-триггер. Первый вариант минимизации основан на последовательности обновления битов блока LRU по алгоритму pseudo LRU, что позволяет сократить комбинационную логику обновления этих битов. Второй вариант минимизации основан на последовательности изменения  $q$  - индекса направления вследствие обновления битов блока LRU по алгоритму pseudo LRU, что дополнительно позволяет уменьшить количество элементов памяти. Оба варианта минимизации позволяют улучшить такие характеристики, как быстродействие и надёжность логической схемы блока LRU.

**Ключевые слова:** алгоритм pseudo LRU; тип триггерной структуры; ассоциативная кэш - память; блок LRU; оценка сложности по Квайну.

### THE MINIMIZATING OF LOGICAL SCHEME FOR IMPLEMENTATION OF PSEUDO LRU BY INTER-TYPE TRANSITION IN TRIGGER STRUCTURES

*V. O. Puidenko, V. S. Kharchenko*

The principle of program control means that the processor core turns to the main memory of the computer for operands or instructions. According to architectural features, operands are stored in data segments, and instructions are stored in code segments of the main memory. The operating system uses both page memory organization and segment memory organization. The page memory organization is always mapped to the segment organization. Due to the cached packet cycles of the processor core, copies of the main memory pages are stored in the internal associative cache memory. The associative cache memory consists of three units: a data unit, a tag unit, and an LRU unit. The data unit stores operands or instructions, the tag unit contains fragments of address information, and the LRU unit contains the logic of policy for replacement of string. The missing event attracts LRU logic to decide for substitution of reliable string in the data unit of associative cache memory. The pseudo-LRU algorithm is a simple and better substitution policy among known substitution policies. Two options for the minimization of the hardware for replacement policy by the pseudo-LRU algorithm in  $q$  - directed associative cache memory is implemented. The transition from the trigger structure of the synchronous D-trigger to the trigger structure of the synchronous JK-trigger is carried out reasonably in both options. The first option of minimization is based on the sequence for updating of the by the algorithm pseudo LRU, which allows deleting of the combinational logic for updating bits of LRU unit. The second option of minimization is based on the sequence for changing of the  $q$  - index of direction, as the consequence for updating the bits of LRU unit by the algorithm pseudo LRU. It allows additionally reducing the number of memory elements. Both options of the minimization allow improving such characteristics as productivity and reliability of the LRU unit.

**Keywords:** algorithm pseudo LRU; type of trigger structure; associative memory cache; LRU unit; assessment of complexity by Quine.

**Пуйденко Вадим Олексійович** – заступник директора з навчально-виробничої роботи, спеціаліст вищої категорії, Харківський радіотехнічний коледж, Харків, Україна.

**Харченко Вячеслав Сергійович** – заслужений винахідник України, д-р техн. наук, професор, завідувач кафедри комп'ютерних систем, мереж та кібербезпеки, Національний аерокосмічний університет ім. М. С. Жуковського «Харківський авіаційний інститут», Харків, Україна.

**Vadym Puidenko** – Deputy Director of Production and Training, Expert of the higher category, Kharkiv RadioTechnical College, Kharkiv, Ukraine, e-mail: VAPuydenko@gmail.com.

**Vyacheslav Kharchenko** – Honored inventor of Ukraine, Doctor of Science on Engineering, Professor, Head of the Department of Computer systems, network and cybersecurity, National Aerospace University «Kharkiv Aviation Institute», Kharkiv, Ukraine, e-mail: V.Kharchenko@csn.khai.edu,

ORCID Author ID: 0000-0001-5352-077X, Scopus Author ID: 22034616000, ResearcherID: A-7719-2017