

В. О. Пуйденко

Харківський радіотехнічний коледж, Харків, Україна

## АЛГОРИТМ ЗАМІЩЕННЯ MFU: АВТОМАТНА МОДЕЛЬ, СИНТЕЗ ТА ОЦІНКА АПАРАТНОЇ РЕАЛІЗАЦІЇ

**Анотація.** Політика заміщення алгоритму MFU (Most Frequently Used) поширюється на архітектуру асоціативного кеш буфер сторінкового перетворення. Подія влучання у кеш буфері сторінкового перетворення має спричинити подію фіксації використання відповідного елементу адресованої множини блоку даних. Подія промаху має залучити певну апаратну логіку для прийняття процесором рішення про заміщення достовірного найбільш часто використаного елементу у блоці даних з урахуванням аналізу фіксацій подій звертання до певних елементів при подіях влучань. В статті побудована автоматна модель політики з комбінаторним підходом компаративного аналізу фіксованих подій звертання. Автоматна модель описана відповідними дискретними функціями і структурною блок – схемою алгоритму. Створена й алгоритмізована автоматна модель спричинила синтез апаратури політики заміщення алгоритму MFU для  $q$  – спрямованого асоціативного кеш буфера сторінкового перетворення. В основу синтезу був покладений математичний апарат комбінаторного синтезу визначення дозволених умов селекції  $q$  - напрямків. Результатом синтезу стали логічні рівняння селекції  $q$  - напрямків зі схемотехнічним рішенням апаратури політики заміщення та детальна блок – схема керуючого апаратурою алгоритму. Синтезоване апаратне рішення дало змогу оцінити її складність за Квайном за кількістю полюсів схеми та за загальною кількістю вентилів. Це дозволило провести розрахунки ймовірностей безвідмовної роботи за час напрацювань на відмову до 100000 годин при ймовірності  $10^{-7}$  відмови одного вентиля. Стаття також містить оцінку швидкодії апаратного рішення як при подіях промахів, такі при подіях влучань.

**Ключові слова:** асоціативний кеш буфер сторінкового перетворення; політика заміщення; алгоритм MFU; автоматна модель; комбінаторний синтез; швидкодія; оцінка складності; ймовірність безвідмовної роботи.

### Вступ

Архітектура асоціативного кеш буфера сторінкового перетворення має містити у своєму складі апаратний модуль з певним алгоритмом заміщення достовірних елементів блоку даних. За такими критеріями як невелика кількість  $q$  - напрямків ( $q=4$ ) [1] та використання з більшою ймовірністю більш “застарілої” інформації є сенс використовувати алгоритм заміщення MFU. Відомо, що елементами блоку даних асоціативного кеш буфера сторінкового перетворення є адресна інформація. При подіях промаху в асоціативній кеш пам’яті “винна” адреса одразу потрапить до асоціативного кеш буфера і “застаріла” інформація стане в нагоді – мине необхідність двохсходинкового перетворення “винної” адреси. Це значно економить такти процесорного ядра, минаючи звертання останнього до каталогу сторінок та таблиць сторінок.

В публікації [2], присвяченій алгоритму заміщення MF-LRU, описана логіка роботи алгоритму MFU, але разом з тим відсутнє апаратне рішення алгоритму і такі його характеристики, як швидкодія, складність реалізації та надійність.

Публікації [3-9] містять дослідження алгоритмів заміщення PLRU, LRU, SF-LRU, MRU на структурному рівні, але тут також тут відсутні оцінки швидкодії, складності реалізації та ймовірностей безвідмовної роботи.

### Мета і задачі

Ймовірності промахів і влучань в асоціативних кеш пам’яті та кеш буфері сторінкового перетворення пов’язані з незбігом або збігом адреси, яку формує пристрій сегментації, з адресами, які зберігається у блоці тегів, і це не дозволяє у повній мірі

дослідити характеристики того чи іншого алгоритму заміщення. Синтезована апаратура алгоритму заміщення дозволяє більш детально дослідити характеристики алгоритму.

**Мета статті** полягає в оцінюванні таких характеристик алгоритму, як швидкодія при подіях промахів і влучань, складність реалізації та ймовірність безвідмовної роботи синтезованої апаратури алгоритму MFU. Ця мета досягається шляхом розв’язання задач розроблення автоматної моделі, схемної реалізації цього алгоритму та дослідження його характеристик.

### 1. Автоматна модель апаратури алгоритму заміщення

До автоматної моделі входять елементи пам’яті типу синхронний двійковий лічильник та комбінатійні логіки  $CS_1$  та  $CS_2$  (рис. 1). Комбінатійну логіку  $CS_1$  представляє перемикальна функція  $\mu(\bullet)$ , яка описує роботу цифрових компараторів, які мають забезпечити порівняння вмістів елементів пам’яті  $\bar{Q}_1, \bar{Q}_2, \dots, \bar{Q}_q$  комбінаторним методом. Вихід комбінатійної логіки  $CS_1$  має сформувати відповідні результати порівнянь у вигляді векторів  $\bar{C}_j, j = \bar{1}, \bar{m}$ , що є аргументами перемикальної функції  $\varphi(\bullet)$ . Загальний алгоритм роботи автоматної моделі описується виразом ( $M$  – сигнал промаху,  $m$  – кількість компараторів,  $e, d$  – номери компараторів):

$$L_i = \lambda\left(\varphi\left(\bar{C}_1, \bar{C}_2, \dots, \bar{C}_m\right), M\right), \quad (1)$$

де  $\bar{C}_j = \mu\left(\bar{Q}_e, \bar{Q}_d\right) \in \{A > B, A = B, A < B\}$ ;  $e \neq d$ .

Кількість компараторів залежить від кількості напрямків  $q$  і визначається як

$$m = S_q^2 = q! / (2! \cdot (q-2)!). \quad (2)$$

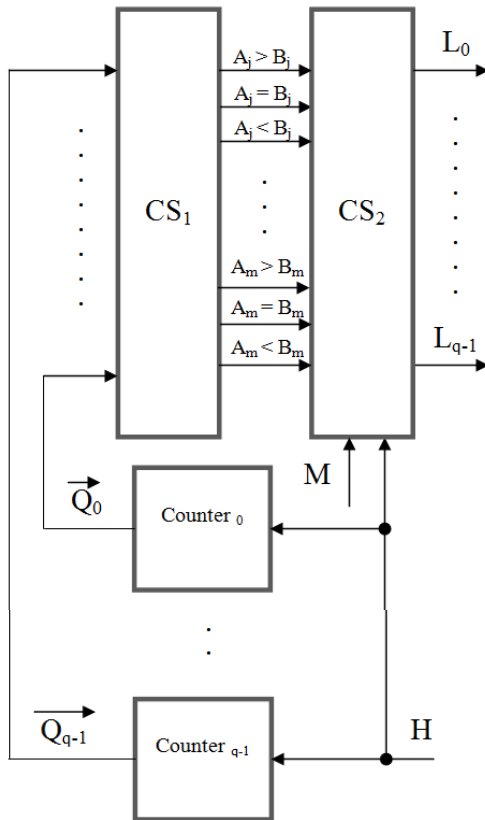


Рис. 1. Автоматна модель апаратури алгоритму заміщення з елементами пам'яті типу синхронний двійковий лічильник

Вихідна комбінаційна логіка CS<sub>2</sub> представляє собою складну перемикальну функцію λ(μ(•)) селекції q напрямків (H – сигнал влучання).

**2. Алгоритм функціонування автоматної моделі**

Алгоритм роботи (рис. 2) автоматної моделі (рис. 1) представляє собою політику заміщення достовірного найбільш нещодавно використаного елемента через селекцію напрямку L<sub>i</sub> за знайденим максимальним значення вмісту лічильника C<sub>i</sub> та дає змогу синтезувати логічну модель апаратури (3) - (69) з подальшою побудовою схемотехнічного рішення (рис. 3) і Результат селекції напрямку L<sub>0</sub> (8) (рис. 4).

**3. Логічна модель апаратури політики заміщення MFU**

У виразах (3)-(69) позначено: C<sub>i</sub> – вміст лічильника, L<sub>i</sub> – селектований напрямок.

**3.1. Логіка селекції напрямку L<sub>0</sub>:**

$$L_{01} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (3)$$

$$L_{02} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (4)$$

$$L_{03} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (5)$$

$$L_{04} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (6)$$

$$L_{05} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (7)$$

$$L_{06} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (8)$$

$$L_{07} = (C_0 = C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (9)$$

$$L_{08} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 = C_3) \& (C_2 = C_3); \quad (10)$$

$$L_{09} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (11)$$

$$L_{10} = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (12)$$

$$L_{1011} = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (13)$$

$$L_{1012} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3); \quad (14)$$

$$L_{1013} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (15)$$

$$L_{1014} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3); \quad (16)$$

$$L_{1015} = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3). \quad (17)$$

**3.2. Логіка селекції напрямку L<sub>1</sub>:**

$$L_{11} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (18)$$

$$L_{12} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (19)$$

$$L_{13} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (20)$$

$$L_{14} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (21)$$

$$L_{15} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (22)$$

$$L_{16} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (23)$$

$$L_{17} = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3); \quad (24)$$

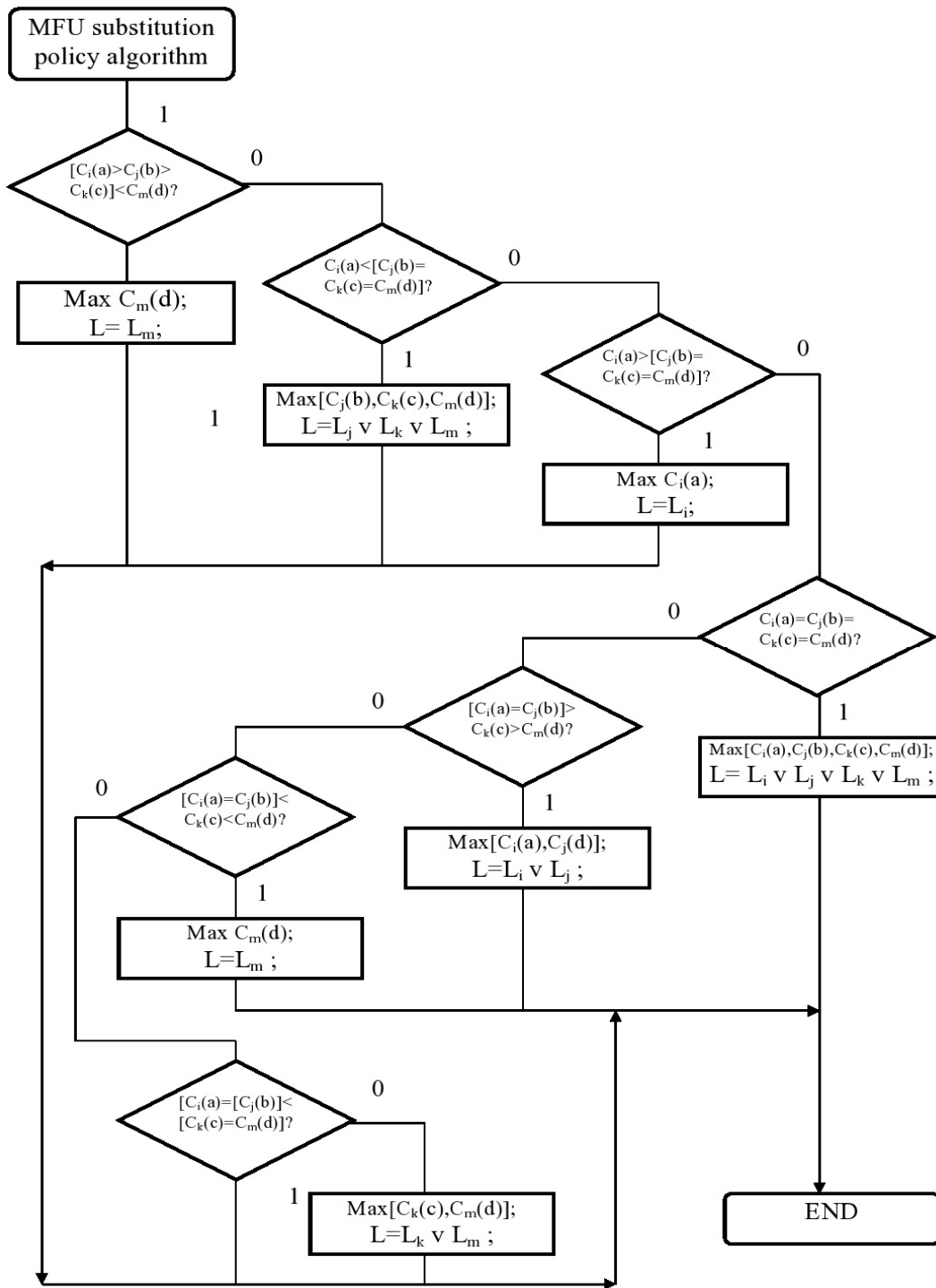


Рис. 2. Алгоритм роботи автоматної моделі алгоритму заміщення MFU з  $q = 4$

$$L_{18} = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3); \quad (25)$$

$$L_{19} = (C_0 = C_1) \& (C_0 = C_2) \& (C_0 = C_3) \& (C_1 = C_2) \& (C_1 = C_3) \& (C_2 = C_3); \quad (26)$$

$$L_{110} = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (27)$$

$$L_{111} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (28)$$

$$L_{112} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3); \quad (29)$$

$$L_{113} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3); \quad (30)$$

$$L_{114} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (31)$$

$$L_{115} = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3); \quad (32)$$

$$L_{116} = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3). \quad (33)$$

### 3.3. Логіка селекції напрямку $L_2$ :

$$L_{21} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (34)$$

$$L_{22} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (35)$$

$$L_{23} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (36)$$

$$L_{24} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (37)$$

$$L_{25} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (38)$$

$$L_{26} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (39)$$

$$L_{27} = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3); \quad (40)$$

$$L_{28} = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3); \quad (41)$$

$$L_{29} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3); \quad (42)$$

$$L_{210} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (43)$$

$$L_{211} = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (44)$$

$$L_{212} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (45)$$

$$L_{213} = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3); \quad (46)$$

$$L_{214} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3); \quad (47)$$

$$L_{215} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3); \quad (48)$$

$$L_{216} = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3). \quad (49)$$

### 3.4. Логіка селекції напрямку $L_3$ :

$$L_{31} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (50)$$

$$L_{32} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (51)$$

$$L_{33} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (52)$$

$$L_{34} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (53)$$

$$L_{35} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (54)$$

$$L_{36} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (55)$$

$$L_{37} = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 = C_3) \& (C_2 = C_3); \quad (56)$$

$$L_{38} = (C_0 = C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (57)$$

$$L_{39} = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3); \quad (58)$$

$$L_{310} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (59)$$

$$L_{311} = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3); \quad (60)$$

$$L_{312} = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (61)$$

$$L_{313} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (62)$$

$$L_{314} = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3); \quad (63)$$

$$L_{315} = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3); \quad (64)$$

$$L_{316} = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3). \quad (65)$$

### 3.5. Логіка остаточної селекції напрямків $L_3$ , $L_2$ , $L_1$ , $L_0$ :

$$L_0 = L_{01} \vee L_{02} \vee L_{03} \vee L_{04} \vee L_{05} \vee L_{06} \vee L_{07} \vee L_{08} \vee L_{09} \vee L_{010} \vee L_{011} \vee L_{012} \vee L_{013} \vee L_{014} \vee L_{015}; \quad (66)$$

$$L_1 = L_{11} \vee L_{12} \vee L_{13} \vee L_{14} \vee L_{15} \vee L_{16} \vee L_{17} \vee L_{18} \vee L_{19} \vee L_{110} \vee L_{111} \vee L_{112} \vee L_{113} \vee L_{114} \vee L_{115} \vee L_{116}; \quad (67)$$

$$L_2 = L_{21} \vee L_{22} \vee L_{23} \vee L_{24} \vee L_{25} \vee L_{26} \vee L_{27} \vee L_{28} \vee L_{29} \vee L_{210} \vee L_{211} \vee L_{212} \vee L_{213} \vee L_{214} \vee L_{215} \vee L_{216}; \quad (68)$$

$$L_3 = L_{31} \vee L_{32} \vee L_{33} \vee L_{34} \vee L_{35} \vee L_{36} \vee L_{37} \vee L_{38} \vee L_{39} \vee L_{310} \vee L_{311} \vee L_{312} \vee L_{313} \vee L_{314} \vee L_{315} \vee L_{316}. \quad (69)$$

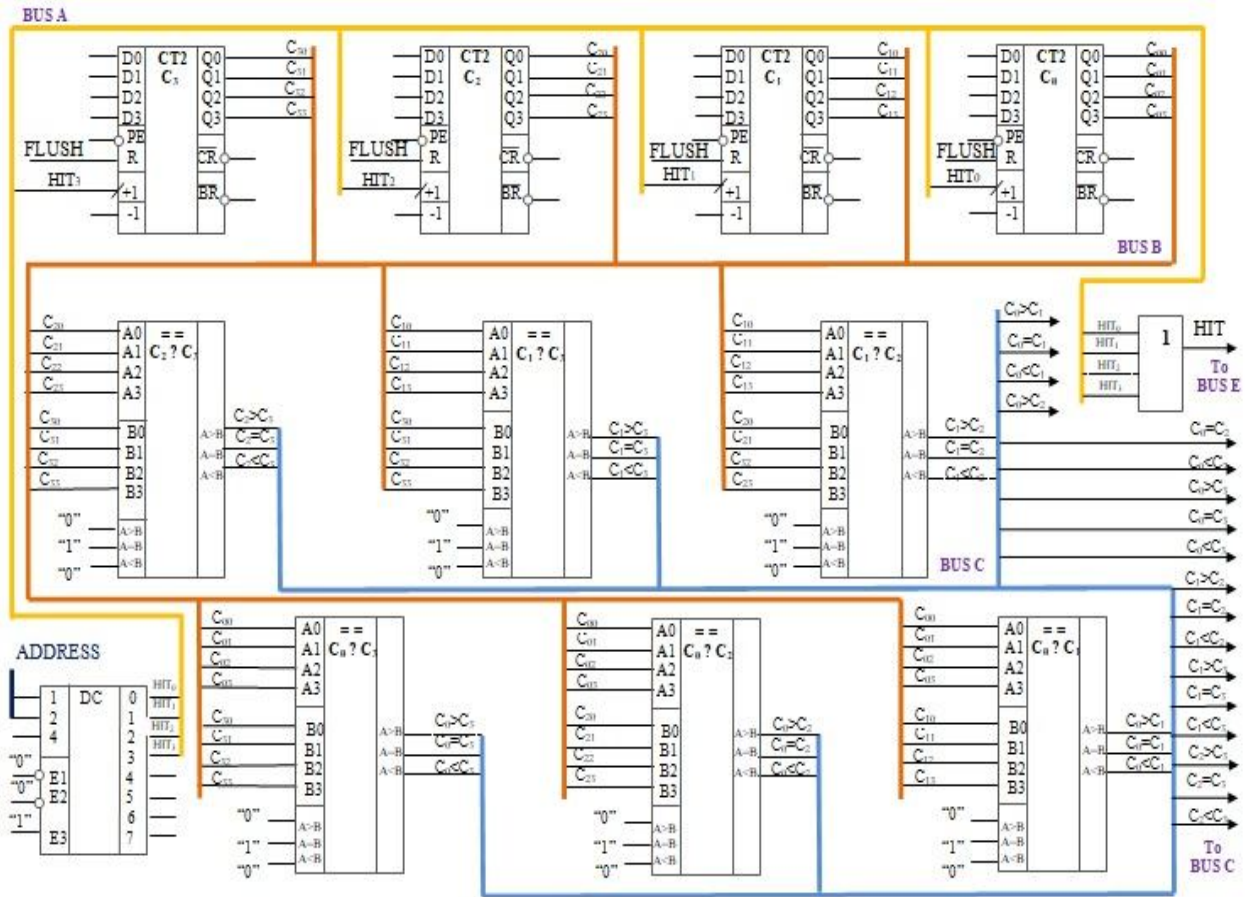


Рис. 3. Схематехнічне рішення апаратури алгоритму заміщення MFU

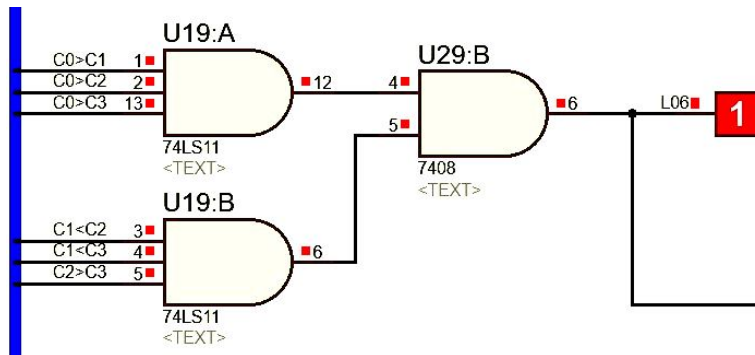


Рис. 4. Результат селекції напрямку  $L_0$  (8) апаратури алгоритму заміщення MFU

#### 4. Результати оцінювання

##### 4.1. Оцінювання складності за Квайном.

Оцінювання складності за Квайном відбувається у відповідності з виразами (70) - (75):

$$K = K_{CS1} + K_{CS2} + K_{cnt}; \quad (70)$$

$$K_{CS1} = K_{cmp} + K_{dc}, \quad (71)$$

де  $K_{CS1}$  – оцінка складності комбінаційної схеми  $CS1$ ,  $K_{CS2}$  – оцінка складності комбінаційної схеми  $CS2$ ,  $K_{cnt}$  – оцінка складності схеми елементів пам'яті типу синхронний двійковий лічильник,  $K_{cmp}$  – оцінка складності компоненти на базі компараторів,  $K_{dc}$  – оцінка складності компоненти на базі декодера;

$$K_{dc} = j \cdot 2^{j-6} \quad (j = 6), \quad (72)$$

$$K_{cnt} = i \cdot 2^{i-6}, \quad (i = 8), \quad (73)$$

$$K_{cmp} = (l-3) \cdot 2^{(l-8)} + 2, \quad (l = 11). \quad (74)$$

Наступний вираз описує складність  $K$  комбінаційної схеми  $CS_2$  через залежність від розрядності логічних елементів, які саме і складають комбінаційну схему:

$$K_{CS2} = [m \cdot 2^m + 2^m] + [2^{n^{n+1}-1} + 2^{n+1} + 2^n + n] + [2^{(k-1)^k} + 2^{(k-1)^*k} + 2^{(k-1)^*k-1} + 2^{k+1} + 2^k + 2^{k-1} + k - 1], \quad (75)$$

де  $n = 2, k = 3, m = 4$

Вирази (72) – (74) є підставою побудови діаграми зміни складності (рис. 5):

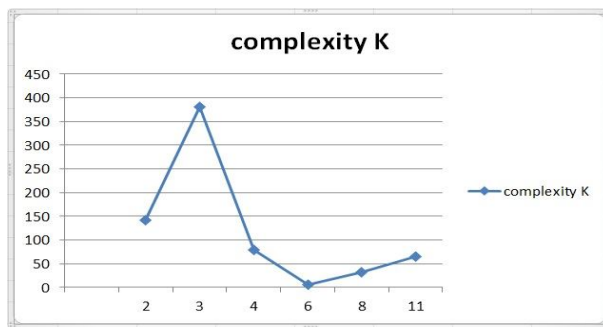


Рис. 5. Діаграма зміни складності апаратури алгоритму заміщення MFU в залежності від розрядності компонент

**4.2. Оцінювання ймовірності безвідмовної роботи.** Ймовірність безвідмовної роботи визначається за таким виразом:

$$P(t) = e^{-\lambda * t}, \tag{76}$$

де  $e$  - Неперове число,  $\lambda$  – ймовірність безвідмовної роботи одного вентиля,  $t$  – час напрацювання на відмову (години).

У табл. 1 наведена складність синтезованої апаратури алгоритму MFU у вентилях.

Таблиця 1 – Складність апаратури алгоритму MFU ( $V$  – кількість вентилів)

№	Компоненти	$V$
1	Синхронний двійковий лічильник на (mod 16)	96
2	Компаратор	30
3	Дешифратор	6
4	Логічний елемент «і» на 3 входи	96
5	Логічний елемент «або» на 3 входи	1
6	Логічний елемент «і» на 2 входи	63
7	Логічний елемент «або» на 4 входи	20
		$V=312$

Формула (76) дозволяє обчислити ймовірності безвідмовної роботи апаратури алгоритму MFU при  $t=100, t=1000, t=10000, t=100000$  годин відповідно:

$$P(100) = 2,72^{-312 * 10^{-7} * 100} = 0,9969;$$

$$P(1000) = 2,72^{-312 * 10^{-7} * 1000} = 0,9693;$$

$$P(10000) = 2,72^{-312 * 10^{-7} * 10000} = 0,7320;$$

$$P(100000) = 2,72^{-312 * 10^{-7} * 100000} = 0,0442.$$

Вищенаведені обчислення дозволяють побудувати відповідну діаграму (рис. 6).

**4.3. Оцінювання швидкодії.** Швидкодія апаратури при події промаху:

$$\tau = \tau_{3AND} + \tau_{2AND} + \tau_{4OR}(\tau_{3OR}) + \tau_{4OR} + \tau_{tbuf}, \tag{77}$$

де  $\tau_{3AND}$  - швидкодія логічної схеми типу AND на 3 входи,  $\tau_{2AND}$  - швидкодія логічної схеми типу AND на 2 входи,  $\tau_{4OR}(\tau_{3OR})$  - швидкодія логічної схеми типу OR на 4/3 входи,  $\tau_{4OR}$  - швидкодія логічної схеми типу OR на 4 входи,  $\tau_{tbuf}$  - час відкриття тристабільного буфера.

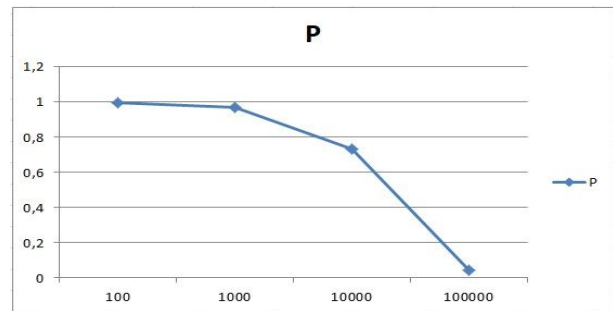


Рис. 6. Діаграма ймовірностей безвідмовної роботи апаратури алгоритму заміщення MFU у часі

Швидкодія апаратури при умові влучання:

$$\tau = \tau_{DC} + \tau_{CNT} + \tau_{CMP} + \tau_{4OR} + \tau_{tbuf}, \tag{78}$$

де  $\tau_{DC}$  - швидкодія логічної схеми комбінаційного типу декодер,  $\tau_{CNT}$  - швидкодія схеми накопичувального типу лічильник,  $\tau_{CMP}$  - швидкодія логічної схеми комбінаційного типу компаратор,  $\tau_{4OR}$  - швидкодія логічної схеми типу OR на 4 входи,  $\tau_{tbuf}$  - час відкриття тристабільного буфера.

Формули (77) та (78) дозволяють оцінити швидкодію апаратури MFU з урахуванням перехідних процесів обраної елементної бази та технологій.

**Висновки**

Досліджено швидкодію апаратної реалізації алгоритму MFU при подіях промахів та влучань, складність апаратної реалізації та ймовірності безвідмовної роботи апаратури. На підставі досліджень можна зробити висновки, що:

- швидкодія апаратури при події влучання повільніше швидкодії при події промаху;
- складність змінюється в залежності від розрядності компонент апаратури;
- з плином часу безвідмовність апаратури експоненційно спадає на проміжку між 100 та 100000 годин напрацювання відповідно.

REFERENCES

1. Puidenko, V. and Kharchenko, V. (2020), "The Pseudo LRU Hardware Complexity Decreasing for Associative Cache Memory and Translation Look-a-Side Buffer", *CERes Journal*, Vol. 6, Is. 1, pp. 142-153.
2. Sabarinathan, Sayiraman, Senthil Kumar, Dayalan and Shanmugavel Mani, Subbiah (2002), *A Framework for MF-LRU (Most Frequently-Least Recently Used) Replacement Policy*, available at: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.618.3125&rep=rep1&type=pdf>
3. Hsin-Chuan, Chen and Jen-Shiun, Chiang (2007), "A high-performance sequential MRU cache using valid-bit assistant search algorithm", *Journal of Circuits, Systems, and Computers*, Vol. 16, No. 4, pp. 613-626.
4. Sudarshan, T.S.B. Rahil Abbas, Mir and Vijayalakshmi, S. (2017), "Highly Efficient LRU Implementations for High Associativity Cache Memory", *Birla Institute of Technology and Science, Pilani, Rajasthan 330331 INDIA*.



5. Safaa S., Omran and Ibrahim A., Amory (2018), "Implementation of LRU Replacement Policy for Reconfigurable Cache Memory Using FPGA", *2018 International Conference on Advanced Science and Engineering*, Kurdistan Region, Iraq, November, 12-14, pp.13-18.
6. Jaafar, Alghazo, Adil, Akaaboune and Nazeih, Botros (2004), "Cache Replacement Algorithm Records", *2004 International Workshop on Memory Technology, Design and Testing*, Illinois, USA, August, pp.19-24.
7. Alejandro, Valero, Julio, Sahuquillo, Salvador, Petit, Pedro, Lopez, and Jose, Duato (2019), "MRU-Tour-based Replacement Algorithms for Last-Level Caches", *Conference: 23rd International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, Vitória (Brazil), DOI: <https://doi.org/10.1109/SBAC-PAD.2011.13>
8. Burhan U.I. Islam, Khan, Rashidah, Olanrewaju, Roohie, Naaz Mir, Abdul Raouf, Khan and Yusoff, S.H. (2017), "A Computationally Efficient P-LRU based Optimal Cache Heap Object Replacement Policy", *Int. Journal of Advanced Computer Science and Applications*, Vol. 8, No. 1, pp. 128-139, DOI: <https://doi.org/10.14569/IJACSA.2017.080118>
9. Swadhash, Kumar and Singh, P.K. (2016), "An Overview of Modern Cache Memory and Performance Analysis of Replacement Policies", *2nd IEEE International Conference on Engineering and Technology*, India, pp. 4145-4148, DOI: <https://doi.org/10.1109/ICETECH.2016.7569243>

Received (Надійшла) 25.08.2020

Accepted for publication (Прийнята до друку) 28.10.2020

#### ABOUT THE AUTHOR / ВІДОМОСТІ ПРО АВТОРА

**Пуйденко Вадим Олексійович** – заступник директора з навчально-виробничої роботи, спеціаліст вищої категорії, Харківський радіотехнічний коледж, 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](mailto:VAPuydenko@gmail.com), ORCID ID: <https://orcid.org/0000-0002-6393-034X>

#### Алгоритм замещения MFU: автоматная модель, синтез и оценка аппаратной реализации

В. А. Пуйденко

**Аннотация.** Политика замещения адаптивного алгоритма MFU (Most Frequently Used) распространяется на архитектуру ассоциативного кэш - буфер страничного преобразования. Событие попадания в кэш - буфере страничного преобразования должно вызвать событие фиксации использования соответствующего элемента адресованного множества блока данных. Событие промаха должно привлечь определенную аппаратную логику для принятия процессором решения о замещении достоверного наиболее часто использованного элемента в блоке данных с учетом анализа фиксаций событий обращения к определенным элементам при событиях попаданий. В статье построена автоматная модель политики с комбинаторным подходом компаративного анализа фиксированных событий обращения. Автоматная модель описана соответствующими дискретными функциями и структурной блок - схемой алгоритма. Созданная и алгоритмированная автоматная модель вызвала синтез аппаратуры политики замещения алгоритма MFU для  $q$  - направленного ассоциативного кэш - буфера страничного преобразования. В основу синтеза был положен математический аппарат комбинаторного синтеза определения разрешенных условий селекции  $q$  - направлений. Результатом синтеза стали логические уравнения селекции  $q$  - направлений со схмотехническим решением аппаратуры политики замещения и детальная блок - схема управляющего аппаратуры алгоритма. Синтезированное аппаратное решение аппаратуры позволило оценить ее сложность по Квайну: по количеству полюсов схемы и по общему количеству вентилей, что позволило провести расчеты вероятностей безотказной работы за время наработок на отказ до 100000 часов при вероятности  $10^{-7}$  отказа одного вентиля. Статья также содержит оценку быстродействия аппаратного решения как при событиях промахов, такие при событиях попаданий.

**Ключевые слова:** ассоциативный кэш буфер страничного преобразования, политика замещения, алгоритм MFU, автоматная модель, комбинаторный синтез, быстродействие, оценка сложности, вероятность безотказной работы.

#### MFU substitution algorithm: automaton model, synthesis and assessment of the hardware implementation

Vadym Puidenko

**Abstract.** The substitution policy of the MFU (Most Frequently Used) algorithm is extending to the associative translation look-a-side buffer architecture. The hit event in the translation look-a-side buffer should call the event of fixing and using the corresponding element in the multitude of the data unit, which was addressed. The miss event should involve certain hardware logic to allow the processor core to make a decision about of the most frequently used element substitution in the data unit, taking into account the analysis of events fixing on access to certain elements at the hit events. The paper has developed automaton model of the policy with combinatorial approach of comparative analysis for fixed events of access. The automaton model is described by the corresponding discrete functions and the structural block diagram of the algorithm. The automaton model which was created and algorithmized caused the synthesis of the hardware substitution policy of the MFU algorithm for  $q$  - directed associative translation look-a-side buffer. The synthesis was founded on the mathematical apparatus of combinatorial synthesis of determining the enabled conditions for selecting  $q$  - directions. The result of the synthesis was logical equations of selection for  $q$  - directions with technical scheme solution of the substitution policy hardware and the detailed diagram block of the control algorithm. The synthesized hardware solution made it possible to make an assessment of complexity according to Quine by the number of poles of the circuit and by the total number of gates, that is allow to compute the probability of failure-free operation during the time of failures up to 100,000 hours with a probability of failures in  $10^{-7}$  for a one gate. The paper also contains an assessment of the productivity for the developed hardware solution as for the miss event, as for the hit event.

**Keywords:** associative translation look-a-side buffer; substitution policy; MFU algorithm; automaton model; combinatorial synthesis; productivity; complexity assessment; probability of failure-free operation.