

УДК 004.254.021

doi: 10.32620/reks.2020.4.06

В. О. ПУЙДЕНКО

Харківський радіотехнічний коледж

АВТОМАТНА МОДЕЛЬ, СИНТЕЗ ПРИСТРОЮ ТА АДАПТИВНОГО АЛГОРИТМУ ЗАМІЩЕННЯ ДЛЯ КЕШ-ПАМ'ЯТІ

Використання певних політик заміщення в асоціативній кеш-пам'яті та асоціативному кеш-буфері сторінкового перетворення обумовлене показниками ймовірностей влучать та промахів. Впровадження комбінованих політик заміщення може покращити роботу кеш-пам'яті і кеш-буфера в цілому за рахунок сумісності алгоритмів з односпрямованими або різноспрямованими політиками заміщення з можливістю перемикання з однієї політики на іншу. Адаптація алгоритмів заміщення для кеш-пам'яті процесорів базується на сумісності алгоритмів за декількома характеристиками, такими як: політика заміщення, швидкодія та складність реалізації. Перелічені характеристики зведені у відповідну таблицю і дозволяють побудувати так звану матрицю сумісності, яка дозволяє спостерігати не тільки пари сумісних адаптивних алгоритмів, але і їх тріади. Політика заміщення адаптивних алгоритмів поширюється і на асоціативну кеш-пам'ять процесорного ядра, і на асоціативний кеш-буфер сторінкового перетворення. В статті побудована автоматна модель адаптивного алгоритму, яка створюється парою сумісних алгоритмів. З одного боку, політика заміщення алгоритму має покладатися у пошуку і заміщенні найменш нещодавно або найменш часто використаного елемента адресованої множини блоку даних. З іншого боку, ця політика може покладатися у пошуку і заміщенні найбільш нещодавно або найбільш часто використаного елемента адресованої множини блоку даних. Автоматна модель описана відповідними дискретними функціями і структурними блок-схемами алгоритмів. Вирішено задачу синтезу апаратури адаптивного алгоритму для q -спрямованих асоціативних кеш-пам'яті та кеш-буфера сторінкового перетворення. В основу синтезу був покладений математичний апарат комбінаторного синтезу визначення дозволених умов селекції q -напрямків. Розроблено логічну модель селекції q -напрямків за адаптивним алгоритмом з відповідним апаратним рішенням.

Ключові слова: політика заміщення; адаптивний алгоритм; PLRU (Pseudo Least Recently Used); LRU (Least Recently Used); MRU (Most Recently Used); LFU (Least Frequently Used); MFU (Most Frequently Used); автоматна модель; комбінаторний синтез; асоціативна кеш-пам'ять; асоціативний кеш-буфер; швидкодія.

Вступ

Сумісність алгоритмів заміщення в основному покладається на подібності апаратних рішень з відповідними рівностями продуктивності, складності реалізації та надійності, що дає змогу поєднувати ці алгоритми в єдиний адаптивний алгоритм, при чому результуюча політика адаптивного, так би мовити, балансує між цими сумісними алгоритмами. Слід зазначити, що політики сумісних алгоритмів можуть бути різноспрямованими. Одночасно політика адаптивного алгоритму може працювати як на асоціативну кеш-пам'ять, так і на асоціативний кеш-буфер сторінкового перетворення процесорного ядра.

В архітектурі асоціативної кеш-пам'яті та асоціативного кеш-буфера сторінкового перетворення може бути інтегрований апаратний модуль з певним адаптивним алгоритмом заміщення достовірних елементів блоку даних. Відомо, що елементами блоку даних асоціативної кеш-пам'яті є інструкції процесора або дані. Елементами блоку даних асоціати-

вного кеш-буфера сторінкового перетворення є адресна інформація. При подіях промаху в асоціативній кеш-пам'яті “винна” адреса одразу потрапить до асоціативного кеш-буфера і політика адаптивного алгоритму стане в нагоді – виникає можливість усунення двоступеневого перетворення “винної” адреси, можливо за рахунок різноспрямованих політик сумісних алгоритмів. Це значно економить такти процесорного ядра, минаючи звертання останнього до каталогу сторінок та таблиць сторінок відповідно.

Публікації [1, 2] на детальному рівні містять досліджені апаратні реалізації алгоритмів PLRU та MFU з наведенням оцінок таких характеристик, як продуктивність, складність реалізації та надійність. Публікації [3, 4] на структурному рівні містять різноманітні апаратні реалізації алгоритму заміщення LRU та MRU. У публікаціях [5 - 8] наведений огляд, аналіз та дослідження алгоритмів заміщення кеш-пам'яті.

Мета і задачі

Мета статті полягає в створенні матриці сумісності алгоритмів заміщення з подальшим синтезом адаптивного алгоритму. Ця мета досягається шляхом розв'язання задач розроблення автоматної моделі, апаратної реалізації цього алгоритму та оцінкою досягнутого ефекту відповідно.

Матриця сумісності

Дослідження у публікаціях [1 - 3] дозволяють

звести характеристики алгоритмів заміщення у таблицю 1. За подібністю їх апаратних рішень вони є сумісними. Апаратні рішення алгоритмів заміщення, які позначені *, наведені у публікаціях [1, 2]. З наведених у таблиці 1 характеристик видно, що апаратні рішення алгоритмів заміщення, які позначені *, мають значну меншу кількість вентилів на множину кеш-пам'яті, а значить більшу швидкодію і надійність відповідно.

У таблиці 2 наведена так звана матриця сумісності алгоритмів заміщення з урахуванням їх характеристик у таблиці 1.

Таблиця 1

Характеристики алгоритмів заміщення

Алгоритм	Складність реалізації (вентилів на множину)	Швидкодія (τ)	
PLRU*	29	$\tau_{PLRU} = 6 * \tau_{and-not}$	$\tau_{PLRU} = \tau_{and-not}$
PLRU* minimum	12	$\tau_{PLRU\ min} = 4 * \tau_{and-not}$	$\tau_{PLRU\ min} = \tau_{and-not}$
LRU Counter implementa- tion	760	$\tau_{LRU\ counter\ implementation} > \tau_{LFU}^*$ $\tau_{LRU\ counter\ implementation} > \tau_{MFU}^*$	$\tau_{LRU\ counter\ implementation} > \tau_{LFU}^*$ $\tau_{LRU\ counter\ implementation} > \tau_{MFU}^*$
LRU Skewed matrix	520	$\tau_{Skewed\ matrix} > \max(\tau_{PLRU}, \tau_{PLRU\ min})$	$\tau_{Skewed\ matrix} > \max(\tau_{PLRU}, \tau_{PLRU\ min})$
LRU Square matrix	720	$\tau_{Square\ matrix} > \max(\tau_{PLRU}, \tau_{PLRU\ min})$	$\tau_{Square\ matrix} > \max(\tau_{PLRU}, \tau_{PLRU\ min})$
LFU*	312	$\tau_{LFU} = \tau_{DC} + \tau_{CNT} +$ $+ \tau_{CMP} + \tau_{4OR} + \tau_{tbuf}$	$\tau_{LFU} = \tau_{3AND} + \tau_{2AND} + \tau_{4OR} (\tau_{3OR}) +$ $+ \tau_{4OR} + \tau_{tbuf}$
MFU*	312	$\tau_{MFU} = \tau_{DC} + \tau_{CNT} +$ $+ \tau_{CMP} + \tau_{4OR} + \tau_{tbuf}$	$\tau_{MFU} = \tau_{3AND} + \tau_{2AND} + \tau_{4OR} (\tau_{3OR}) +$ $+ \tau_{4OR} + \tau_{tbuf}$

Таблиця 2

Матриця сумісності алгоритмів заміщення

	PLRU*	PLRU* minimum	LRU Counter Implementation	LRU Skewed Matrix	LRU Square Matrix	LFU*	MFU*
PLRU*	x	1	0	1	1	0	0
PLRU* (minimum)	1	x	0	1	1	0	0
LRU Counter Implementation	0	0	x	0	0	1	1
LRU Skewed Matrix	1	1	0	x	1	0	0
LRU Square Matrix	1	1	0	1	x	0	0
LFU*	0	0	1	0	0	x	1
MFU*	0	0	1	0	0	1	x

Автоматна модель апаратури адаптивного алгоритму заміщення

Автоматна модель розрахована на адаптивний алгоритм за політиками заміщення LFU, LRU (Counter implementation) та MFU. До автоматної моделі входять елементи пам'яті типу синхронний двійковий лічильник та комбінаційні логіки CS₁, CS₂ та CS₃ (рис. 1). Комбінаційну логіку CS₁ представляє перемикальна функція $\varphi(\bullet)$, 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 комбінаційної логіки CS₁ має сформувані відповідні результати порівнянь у вигляді векторів $\bar{C}_j, j = \overline{1, m}$, що є аргументами перемикальної функції $\varphi(\bullet)$. Логіка комбінаційної схеми CS₂ описується складною перемикальною функцією $\mu(\bullet)$. Загальний алгоритм роботи автоматної моделі описується складною перемикальною функцією (1) (M – сигнал промаху, H – сигнал влучання, m – кількість компараторів, S – сигнал вирішувача, e, d – номери компараторів):

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

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

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

Комбінаційна схема CS₃ описується складною перемикальною функцією $\mu(\bullet)$ і являє собою комбінаційну логіку селекції напрямків за алгоритмами заміщення LFU та MFU відповідно. Вона спряжена з вирішувачем, логіка якого повинна вирішувати, політика заміщення якого алгоритму має працювати за значенням керуючого сигналу S.

Алгоритми роботи (рис. 2) представляє собою політику заміщення достовірного найменш часто використаного елемента через селекцію напрямку Li за знайденим мінімальним значенням вмісту лічильника Ci. Алгоритми роботи (рис. 3) представляє собою політику заміщення достовірного найбільш часто використаного елемента через селекцію напрямку Li за знайденим максимальним значенням вмісту лічильника Ci. Наведені алгоритми дають змогу синтезувати логічну модель апаратури (3) - (73) з подальшою побудовою відповідної комп'ютерної моделі (рис.4) адаптивного алгоритму заміщення.

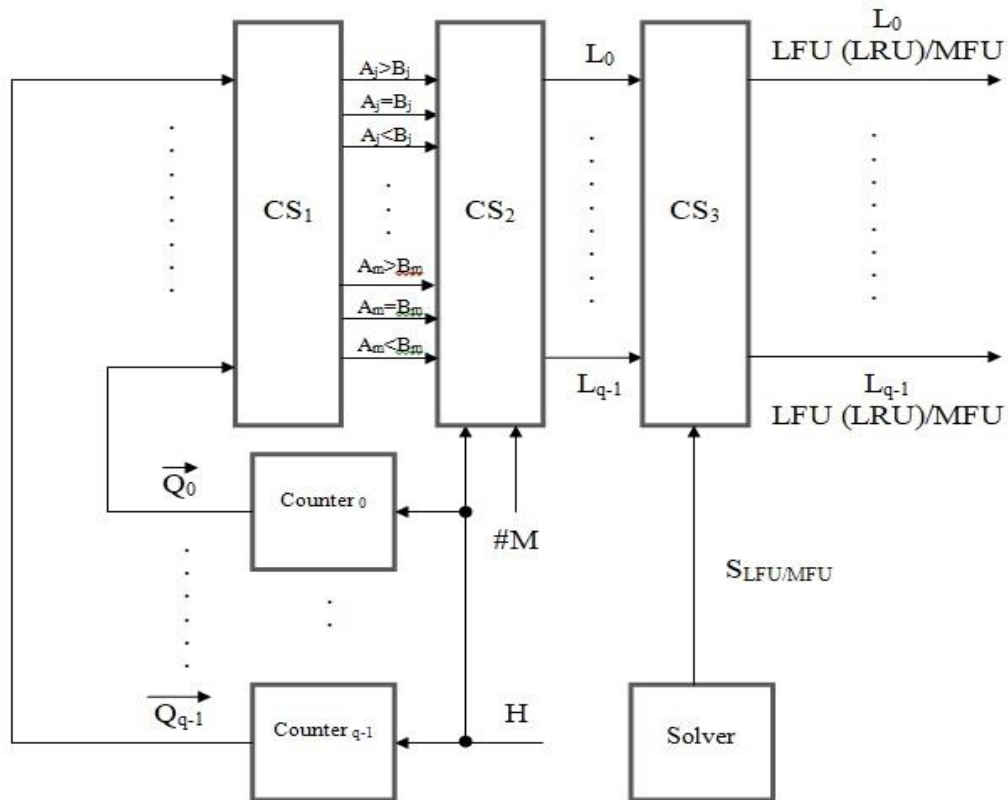


Рис. 1. Автоматна модель апаратури адаптивного алгоритму заміщення

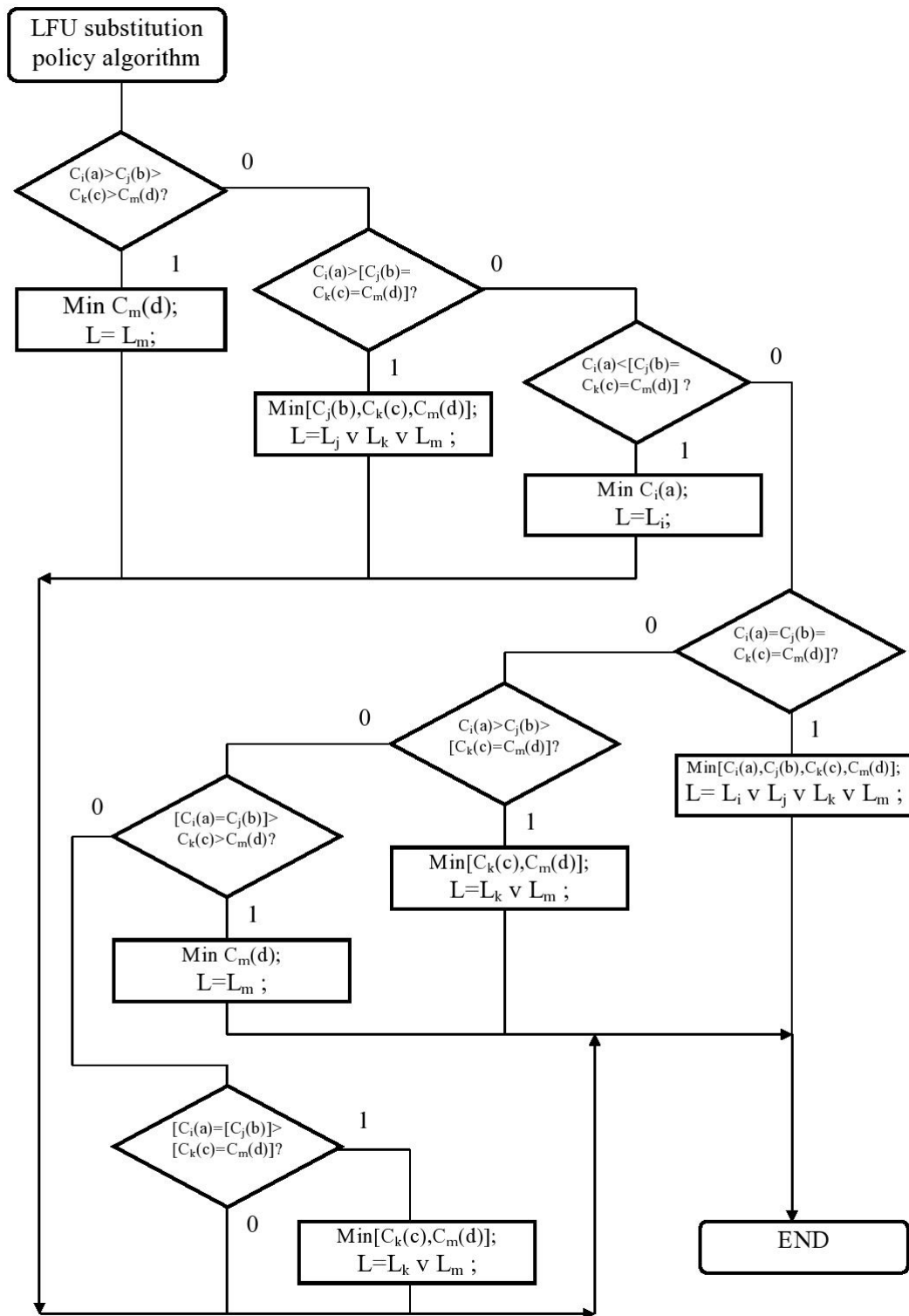


Рис. 2. Алгоритм заміщення LFU

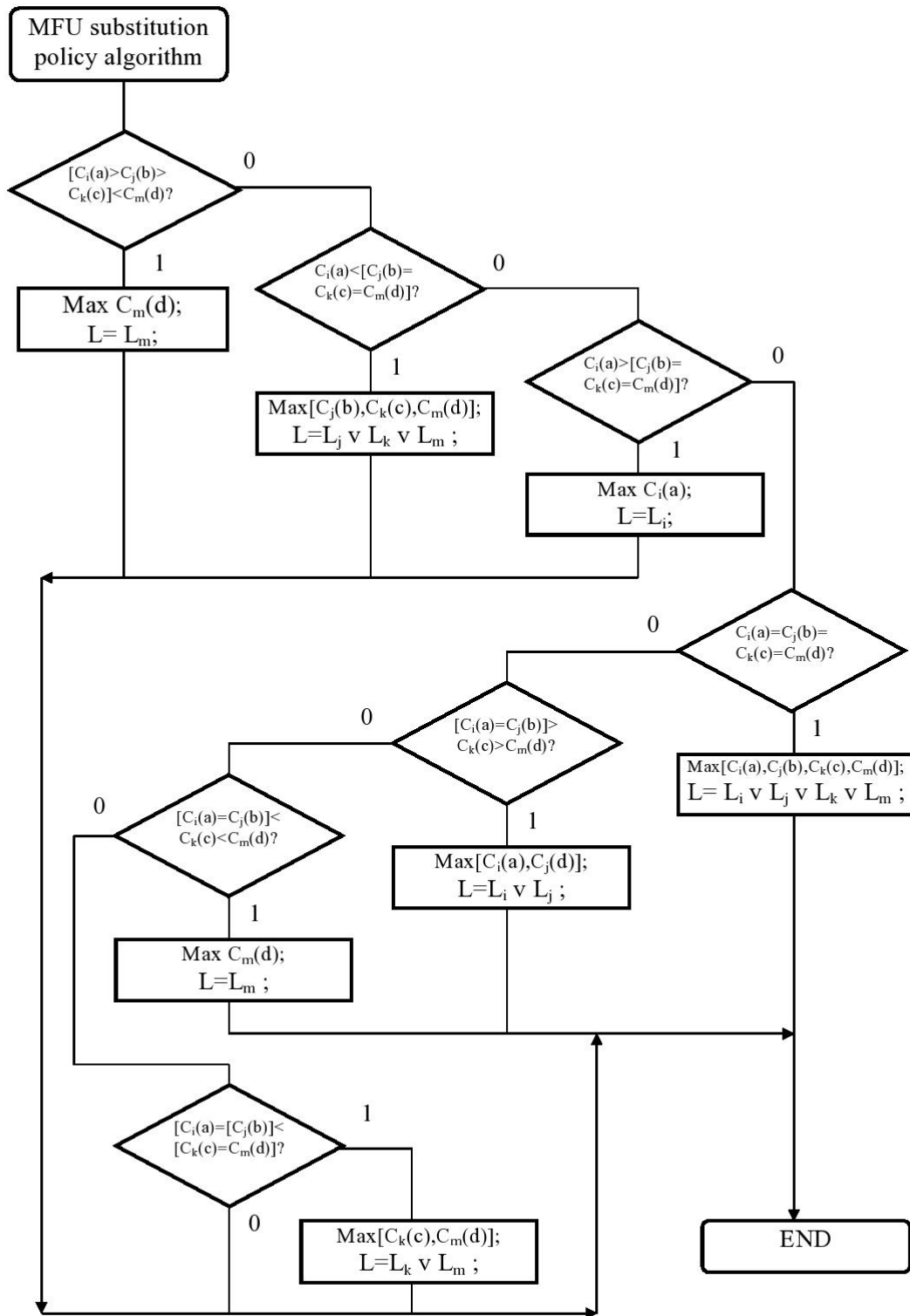


Рис. 3. Алгоритм заміщення MFU

Логічна модель адаптивного алгоритму (сумісних LFU(LRU) та MFU)

У виразах (3) - (73) позначено: C_i – вміст лічильника, L_i – селектований напрямок.

Логіка селекції напрямку L_0 алгоритму LFU (L_i MFU):

$$L_{01LFU}(L_{31MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (3)$$

$$L_{02LFU}(L_{36MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (4)$$

$$L_{03LFU}(L_{26MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (5)$$

$$L_{04LFU}(L_{16MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (6)$$

$$L_{05LFU}(L_{11MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (7)$$

$$L_{06LFU}(L_{21MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (8)$$

$$L_{07LFU}(L_{38MFU}) = (C_0 = C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (9)$$

$$L_{08LFU}(L_{37MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 = C_3) \& (C_2 = C_3) \quad (10)$$

$$L_{09LFU}(L_{114MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (11)$$

$$L_{010LFU}(L_{312MFU}) = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (12)$$

$$L_{011LFU}(L_{314MFU}) = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (13)$$

$$L_{012LFU}(L_{29MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3) \quad (14)$$

$$L_{013LFU}(L_{111MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (15)$$

$$L_{014LFU}(L_{112MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3) \quad (16)$$

$$L_{015LFU}(L_{216MFU}) = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3) \quad (17)$$

Логіка селекції напрямку L_1 алгоритму LFU (L_i MFU):

$$L_{111LFU}(L_{32MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (18)$$

$$L_{12LFU}(L_{35MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (19)$$

$$L_{13LFU}(L_{25MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (20)$$

$$L_{14LFU}(L_{06MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (21)$$

$$L_{15LFU}(L_{01MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (22)$$

$$L_{16LFU}(L_{22MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (23)$$

$$L_{17LFU}(L_{28MFU}) = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3) \quad (24)$$

$$L_{18LFU}(L_{27MFU}) = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3) \quad (25)$$

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

$$L_{110LFU}(L_{213MFU}) = (C_0 = C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (27)$$

$$L_{111LFU}(L_{313MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (28)$$

$$L_{112LFU}(L_{214MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3) \quad (29)$$

$$L_{113LFU}(L_{39MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 < C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 = C_3) \quad (30)$$

$$L_{114LFU}(L_{09MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (31)$$

$$L_{115LFU}(L_{011MFU}) = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 < C_3) \& (C_2 > C_3) \quad (32)$$

$$L_{116LFU}(L_{316MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (33)$$

**Логіка селекції напрямку L_2
алгоритму LFU (L_i MFU):**

$$L_{21LFU}(L_{33MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (34)$$

$$L_{22LFU}(L_{34MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (35)$$

$$L_{23LFU}(L_{15MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (36)$$

$$L_{24LFU}(L_{05MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (37)$$

$$L_{25LFU}(L_{02MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (38)$$

$$L_{26LFU}(L_{12MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (39)$$

$$L_{27LFU}(L_{18MFU}) = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3) \quad (40)$$

$$L_{28LFU}(L_{17MFU}) = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3) \quad (41)$$

$$L_{29LFU}(L_{113MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3) \quad (42)$$

$$L_{210LFU}(L_{013MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (43)$$

$$L_{211LFU}(L_{115MFU}) = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (44)$$

$$L_{212LFU}(L_{310MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 = C_3) \& (C_1 > C_2) \& (C_1 < C_3) \& (C_2 < C_3) \quad (45)$$

$$L_{213LFU}(L_{010MFU}) = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 < C_3) \quad (46)$$

$$L_{214LFU}(L_{311MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3) \quad (47)$$

$$L_{215LFU}(L_{315MFU}) = (C_0 < C_1) \& (C_0 = C_2) \& (C_0 < C_3) \& (C_1 > C_2) \& (C_1 = C_3) \& (C_2 < C_3) \quad (48)$$

$$L_{216LFU}(L_{116MFU}) = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3) \quad (49)$$

**Логіка селекції напрямку L_3
алгоритму LFU (L_i MFU):**

$$L_{31LFU}(L_{23MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (50)$$

$$L_{32LFU}(L_{24MFU}) = (C_0 > C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (51)$$

$$L_{33LFU}(L_{14MFU}) = (C_0 < C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (52)$$

$$L_{34LFU}(L_{04MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (53)$$

$$L_{35LFU}(L_{03MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (54)$$

$$L_{36LFU}(L_{13MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (55)$$

$$L_{37LFU}(L_{08MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 = C_3) \& (C_2 = C_3) \quad (56)$$

$$L_{38LFU}(L_{07MFU}) = (C_0 = C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (57)$$

$$L_{39LFU}(L_{012MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 = C_3) \quad (58)$$

$$L_{310LFU}(L_{212MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (59)$$

$$L_{311LFU}(L_{014MFU}) = (C_0 > C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3) \quad (60)$$

$$L_{312LFU}(L_{110MFU}) = (C_0 = C_1) \& (C_0 > C_2) \& (C_0 > C_3) \& (C_1 > C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (61)$$

$$L_{313LFU}(L_{210MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 > C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (62)$$

$$L_{314LFU}(L_{211MFU}) = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (63)$$

$$L_{315LFU}(L_{015MFU}) = (C_0 > C_1) \& (C_0 = C_2) \& (C_0 > C_3) \& (C_1 < C_2) \& (C_1 = C_3) \& (C_2 > C_3) \quad (64)$$

$$L_{316LFU}(L_{215MFU}) = (C_0 < C_1) \& (C_0 < C_2) \& (C_0 = C_3) \& (C_1 = C_2) \& (C_1 > C_3) \& (C_2 > C_3) \quad (65)$$

Логіка остаточної селекції напрямків L_0, L_1, L_2, L_3 за алгоритмом LFU:

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

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

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

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

Логіка остаточної селекції напрямків L_0, L_1, L_2, L_3 за алгоритмом MFU:

$$L_0 = L_{14} \vee L_{15} \vee L_{114} \vee L_{115} \vee L_{24} \vee L_{25} \vee L_{210} \vee L_{213} \vee L_{34} \vee L_{35} \vee L_{37} \vee L_{38} \vee L_{39} \vee L_{311} \vee L_{315} \quad (70)$$

$$L_2 = L_{03} \vee L_{06} \vee L_{012} \vee L_{015} \vee L_{13} \vee L_{16} \vee L_{17} \vee L_{18} \vee L_{110} \vee L_{112} \vee L_{31} \vee L_{32} \vee L_{310} \vee L_{313} \vee L_{314} \vee L_{316} \quad (71)$$

$$L_1 = L_{04} \vee L_{05} \vee L_{09} \vee L_{013} \vee L_{014} \vee L_{19} \vee L_{23} \vee L_{26} \vee L_{27} \vee L_{28} \vee L_{29} \vee L_{211} \vee L_{216} \vee L_{33} \vee L_{36} \vee L_{312} \quad (72)$$

$$L_3 = L_{01} \vee L_{02} \vee L_{07} \vee L_{08} \vee L_{010} \vee L_{011} \vee L_{11} \vee L_{12} \vee L_{111} \vee L_{113} \vee L_{116} \vee L_{21} \vee L_{22} \vee L_{212} \vee L_{214} \vee L_{215} \quad (73)$$

Висновки

Синтезовано і досліджено автоматну модель і апаратну реалізацію адаптивного алгоритму заміщення достовірних рядків при умовах промахів. На підставі досліджень можна зробити висновки, що синтезоване апаратне рішення адаптивного алгоритму працює у відповідності з кожним із сумісних алгоритмів, а також:

- складність апаратури адаптивного алгоритму залишається практично рівною складності апаратури кожної з незначним ускладненням за рахунок логічних схем CS3, які описані перемикальними функціями (66) - (69) або (70) - (73) кожного із сумісних алгоритмів і складності перемикача, який вирішує доцільність виконання відповідного алгоритму;

- швидкодія апаратури адаптивного алгоритму при події влучань і промахів залишається рівною швидкодії апаратури кожного із сумісних алгоритмів;

- складність реалізації і швидкодія апаратури адаптивного алгоритму є кращою за швидкодію сумісного алгоритму заміщення LRU (counter implementation) [3].

Література

1. Пуйденко, В. О. Алгоритм заміщення MFU: автоматна модель, синтез та оцінка апаратної реалізації [Text] / В. О. Пуйденко // Сучасні інформаційні системи. - 2020. - Т. 4, № 4. - С. 57-63. DOI: 10.20998/2522-9052.2020.4.08.

2. Пуйденко, В. О. Мінімізація логічної схеми для реалізації pseudo - LRU шляхом між типового переходу у тригерних структурах [Text] / В. О. Пуйденко, В. С. Харченко // Радіоелектронні і комп'ютерні системи. - 2020. - №2. - С. 33-47. DOI: 10.32620/reks.2020.2.03.

3. 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>. - 10.05.2020.

4. 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

5. 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.

6. 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.

7. 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.

8. 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.

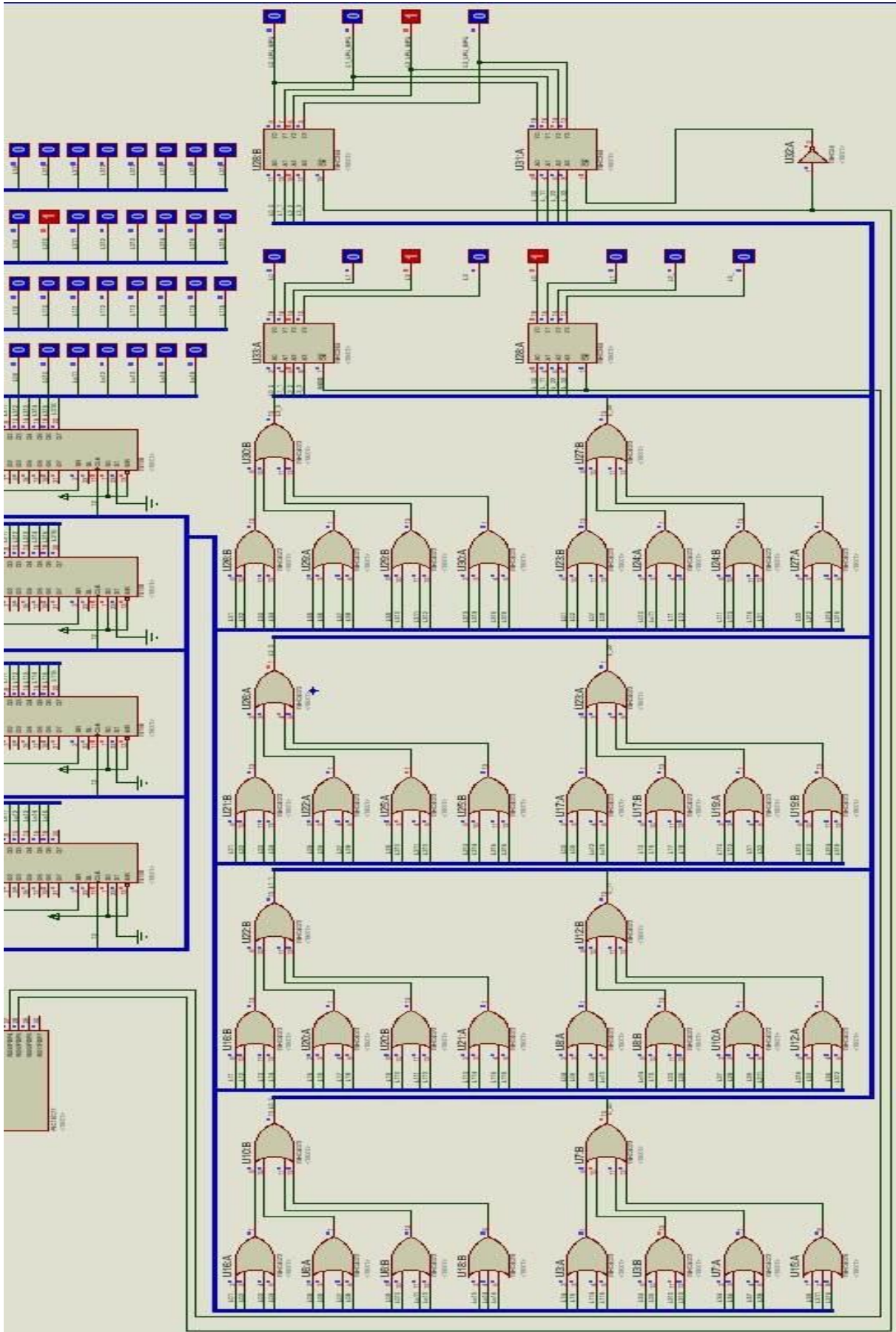


Рис. 4. Комп'ютерна модель адаптивного алгоритму з перемиканням селекції напрямків L за алгоритмом LFU

References

1. Puydenko, V. O. Alhorytm zamishchennya MFU: avtomatna model', syntez ta otsinka aparatnoyi realizatsiyi [MFU substitution algorithm: automatic model, synthesis and evaluation of hardware implementation]. *Suchasni informatsiyini systemy – Advanced information systems*, vol. 4, no. 4, 2020, pp. 57-63. DOI: 10.20998/2522-9052.2020.4.08.
2. Puydenko, V. O., Kharchenko, V. S. Minimizatsiya lohichnoyi skhemy dlya realizatsiyi pseudo – LRU shlyakhom mizh tipovoho perekhodu u tryhernykh strukturakh [The minimizing of logical scheme for implementation of pseudo lru by inter-type transition in trigger structures]. *Radioelektronni i komp'uterni sistemi – Radioelectronic and computer systems*, 2020, no. 2, pp. 33-47. DOI: 10.32620/reks.2020.2.03.
3. 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 10.05.2020).
4. Omran, Safaa S. 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.
5. 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.
6. Kumar, Swadesh., 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.
7. 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.
8. Reineke, J., Grund, D., Berg, C., Wilhelm, R. Timing predictability of cache replacement policies. *Real-Time Syst.*, vol. 37, no. 2, 2007, pp. 99-122.

Поступила в редакцию 4.09.2020, рассмотрена на редколлегии 16.11.2020

АВТОМАТНАЯ МОДЕЛЬ, СИНТЕЗ УСТРОЙСТВА И АДАПТИВНОГО АЛГОРИТМА ЗАМЕЩЕНИЯ ДЛЯ КЭШ-ПАМЯТИ

В. А. Пуйденко

Использование определенных политик замещения в ассоциативной кэш-памяти и ассоциативном кэш-буфере страничного преобразования обусловлено показателями вероятностей попаданий и промахов. Внедрение комбинированных политик замещения может улучшить работу кэш-памяти и кэш-буфера в целом за счет совместимости алгоритмов с однонаправленными или разнонаправленными политиками замещения с возможностью переключения с одной политики на другую. Адаптация алгоритмов замещения основана на совместимости алгоритмов по нескольким характеристикам, таким как: политика замещения, быстродействие, и сложность реализации. Все перечисленные характеристики сведены в соответствующую таблицу и позволяют построить так называемую матрицу совместимости, которая позволяет наблюдать не только пары совместимых адаптивных алгоритмов, но и их триады. Политика замещения адаптивных алгоритмов распространяется и на ассоциативную кэш-память процессорного ядра, и на ассоциативный кэш-буфер страничного преобразования. В статье построена автоматная модель адаптивного алгоритма, которая создается парой совместимых алгоритмов. С одной стороны политика замещения алгоритма должна состоять в поиске и замещении наименее недавно или наименее часто использованного элемента адресованного множества блока данных. С другой стороны, эта политика может состоять в поиске и замещении наиболее недавно или наиболее часто использованного элемента адресованного множества блока данных. Автоматная модель описана соответствующими дискретными функциями и структурными блок - схемами алгоритмов. Созданная и алгоритмизированная автоматная модель стала причиной синтеза аппаратуры адаптивного алгоритма для q - направленных ассоциативных кэш-памяти и кэш-буфера страничного преобразования. В основу синтеза был положен математический аппарат комбинаторного синтеза определения разрешенных условий селекции q – направлений. Результатом синтеза стала логическая модель селекции q – направлений по адаптивному алгоритму с соответствующим аппаратным решением.

Ключевые слова: политика замещения; адаптивный алгоритм; PLRU (Pseudo Least Recently Used); LRU (Least Recently Used); MRU (Most Recently Used); LFU (Least Frequently Used); MFU (Most Frequently Used); автоматная модель; комбинаторный синтез; ассоциативная кэш-память; ассоциативный кэш-буфер; быстродействие.

AUTOMATON MODEL, DEVICE SYNTHESIS AND ADAPTIVE SUBSTITUTION ALGORITHM FOR CACHE MEMORY

V. Puidenko

The probability indicators of the hits or misses events have conditioned the application of the certain substitution policies in the associative cache and the associative translation look-a-side buffer. The implementation of combined substitution policies can improve cache memory and cache buffer performance in general by the interoperability of algorithms with unidirectional or multidirectional substitution policies with the ability to switch from one policy to another. Adaptation of substitution algorithms is based on the compatibility of algorithms according to several characteristics, such as substitution policy, productivity, and implementation complexity. All listed characteristics are summarized in the corresponding table and they are allowed to create the construction for the so-called compatibility matrix, which allows observing not only pairs of compatible algorithms, but also their triads. The substitution policy of the adaptive algorithms extends as to the associative cache memory as to the associative translation look-a-side buffer of the processor core. In the paper, the automaton model of an adaptive algorithm was built and was created by a pair of compatible algorithms. On the one side, the substitution policy algorithm policy should rely on finding and replacing the least recently used or least frequently used element of the addressed set of the data unit. On the other side, this policy may occur in searching and replacing the most recently used or most frequently used element in the addressed set of a data unit. The automaton model is described by the corresponding discrete functions and structural block diagrams of algorithms. The automaton model was created and algorithmized and was the reason for the synthesis of adaptive algorithm hardware for q – directional associative cache memory and associative translation look-a-side buffer. The synthesis was based on the mathematical apparatus of combinatorial synthesis for determining the enabling conditions for selecting q - directions. The result of the synthesis was the logic model of a selection of q – directions according to an adaptive algorithm with the corresponding hardware solution.

Keywords: substitution policy, adaptive algorithm, PLRU (Pseudo Least Recently Used), LRU (Least Recently Used), MRU (Most Recently Used), LFU (Least Frequently Used), MFU (Most Frequently Used), automaton model, combinatorial synthesis, associative cache memory, associative cache buffer, productivity.

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, ORCID: 0000-0002-6393-034X.