

## **КЕШ-ПАМ'ЯТЬ**

### **МЕТОДИЧНІ ВКАЗІВКИ**

до самостійних робіт з дисципліни  
“Архітектура комп’ютерів”  
для студентів заочної форми навчання спеціальності  
123 – “Комп’ютерна інженерія”

Затверджено на  
засіданні кафедри  
інформаційних і комп’ютерних систем  
*Протокол № 10*  
*від 30 березня 2017 р.*

Кеш-пам'ять. Методичні вказівки до самостійних робіт з дисципліни “Архітектура комп'ютерів” для студентів заочної форми навчання спеціальності 123 – "Комп'ютерна інженерія"./ Укл. А.І. Роговенко, О.В. Красножон, А.В. Красножон – Чернігів: ЧНТУ, 2017. – 11 с.. Укр. мовою.

Укладачі: РОГОВЕНКО Андрій ІВАНОВИЧ, старший викладач кафедри інформаційних та комп'ютерних систем  
КРАСНОЖОН ОЛЕКСІЙ ВАСИЛЬОВИЧ, старший викладач кафедри інформаційних та комп'ютерних систем  
КРАСНОЖОН Андрій ВАСИЛЬОВИЧ, кандидат технічних наук, доцент кафедри електричних систем та мереж

Відповідальний за випуск: ЗАЙЦЕВ СЕРГІЙ ВАСИЛЬОВИЧ, завідувач кафедри інформаційних і комп'ютерних систем, доктор технічних наук, доцент

Рецензент: НЕСТЕРЕНКО СЕРГІЙ ОЛЕКСАНДРОВИЧ, кандидат технічних наук, доцент кафедри інформаційних і комп'ютерних систем Чернігівського державного технологічного університету.

## ЗМІСТ

ВСТУП .....	4
1 ПРОГРАМНА МОДЕЛЬ КЕШ-ПАМ'ЯТІ.....	5
2 РОЗРАЗУНКОВО-ГРАФІЧНА РОБОТА№1. ПРИНЦИПИ РОБОТИ	
КЕШ-ПАМ'ЯТІ.....	7
2.1 Завдання 1 .....	7
2.1.1 Порядок виконання роботи.....	8
2.1.2 Зміст звіту .....	8
2.1.3 Контрольні питання .....	8
2.2 Алгоритми заміщення рядків кеш-пам'яті.....	8
2.3 Завдання 2 .....	9
2.3.1 Порядок виконання роботи.....	9
2.3.2 Зміст звіту .....	10
2.3.3 Контрольні питання .....	10
РЕКОМЕНДОВАНА ЛІТЕРАТУРА .....	11

## **ВСТУП**

Проводити початкове навчання програмуванню на низькому рівні з розглядом механізмів взаємодії пристроїв на реальну мову, наприклад x86 на персональній ЕОМ, не завжди зручно. У цьому випадку між користувачем і апаратурою ЕОМ присутній операційна система (ОС) яка суттєво обмежує бажання користувача експериментувати з апаратними засобами. Для подолання цих обмежень необхідно володіти глибокими знаннями як ОС, так і апаратних засобів ЕОМ.

# 1 ПРОГРАМНА МОДЕЛЬ КЕШ-ПАМ'ЯТІ

До описаної в розділі 1 програмної моделі навчальної ЕОМ може бути підключена програмна модель кеш-пам'яті. Кеш-пам'ять містить  $N$  комірок (в моделі  $N$  може вибиратися з безлічі  $\{4, 8, 16, 32\}$ ), кожна з яких включає трьохрозрядне поле тегу (адреси ОЗП), шестирозрядне поле даних і три одинітових ознаки (прапора):

- $Z$  – ознака зайнятості комірки;
- $U$  – ознака використання;
- $W$  – ознака запису в комірку.

Таким чином, кожна комірка кеш-пам'яті може дублювати одну будь-яку комірку ОЗП, причому відзначається її зайнятість (на початку роботи моделі все комірки кеш-пам'яті вільні,  $VZ_i = 0$ ), факт запису інформації в комірку під час перебування її в кеш-пам'яті, а також використання комірки (т. е. будь-яке звернення до неї).

Поточний стан кеш-пам'яті відображається на екрані в окремому вікні в формі таблиці, причому кількість рядків відповідає обраному числу комірок кеш. Стовпці таблиці визначають вміст полів комірок, наприклад, так, як показано в табл. 1.1.

Таблиця 1.1 – Приклад поточного стану кеш-пам'яті

	Теги	Дані	$Z$	$U$	$W$
1	012	220152	1	0	0
2	013	211003	1	1	0
3	050	000025	1	1	1
4	000	000000	0	0	0

Для налаштування параметрів кеш-пам'яті можна скористатися діалоговим вікном кеш-пам'ять, що викликається командою Вид | Кеш-пам'ять, а потім натиснути першу кнопку на панелі інструментів відкритого вікна. Після цих дій з'явиться діалогове вікно Параметри кеш-пам'яті, що дозволяє вибрати розмір кеш-пам'яті, спосіб запису в неї інформації та алгоритм заміщення комірок.

Нагадаємо, що при наскрізній запису при кеш-попаданні в процесорних циклах запису здійснюється запис як в комірку кеш-пам'яті, так і в комірку ОЗП, а при зворотному запису - тільки в комірку кеш-пам'яті, причому ця комірка відзначається бітом запису ( $Wt: = 1$ ). При очищенні комірок, зазначених бітом запису, необхідно переписати змінене значення нуля даних в відповідному полі ОЗП.

При кеш-промаху слід помістити в кеш-пам'ять, певна комірка яка адресується процесором. При наявності вільних комірок кеш-пам'яті потрібне слово поміщається в одну з них (в порядку черги). При відсутності вільних комірок слід відшукати комірку кеш-пам'яті, вміст якої можна видалити, записавши на його місце необхідні дані (команду). Пошук такої комірки

здійснюється з використанням алгоритму заміщення рядків. У моделі реалізовані три різних алгоритма заміщення рядків:

— Випадкове заміщення, при реалізації якого номер комірки кеш-пам'яті вибирається випадковим чином;

— Черга, при якій вибір змінюваній комірки визначається часом перебування її в кеш-пам'яті;

— Біт використання, випадковий вибір здійснюється тільки з тих комірок, які мають нульове значення прапора використання.

Нагадаємо, що біт використання встановлюється в 1 при будь-якому зверненні до комірки, однак, як тільки всі біти  $U_i$ , встановляться в 1, всі вони тут же скидаються в 0, так що в кеш завжди комірки розбиті на два непересічних підмножини за значенням біта  $U$ , звернення до яких відбулося відносно недавно (після останнього скидання вектора  $U$ ) мають значення  $U = 1$ , інші зі значенням  $U = 0$  є "кандидатами на видалення" при використанні алгоритму заміщення "біт використання".

Якщо в параметрах кеш-пам'яті встановлений прапор "з урахуванням біта запису", то всі три алгоритму заміщення здійснюють пошук "кандидата на видалення" перш за все серед тих комірок, ознака запису яких не встановлено, а при відсутності таких комірок (що вкрай мало ймовірно) - серед усіх комірок **кешпам'яті**. При знятому прапорі "з урахуванням біта записи" пошук здійснюється за всіма комітками кеш-пам'яті без урахування значення  $W$ .

Оцінка ефективності роботи системи з кеш-пам'яттю визначається числом кеш-влучень по відношенню до загальної кількості звернень до пам'яті. З огляду на різницю в алгоритмах запису в режимах наскрізною і зворотного запису, ефективність використання кеш-пам'яті обчислюється за виразами 1.1 і 1.2 (відповідно для наскрізною і зворотного запису):

$$K = \frac{S_k - S_{k_w}}{S_0}, \quad (1.1)$$

$$K = \frac{S_k - S_{k_w}^i}{S_0}, \quad (1.2)$$

- Де:  $K$  – коефіцієнт ефективності роботи кеш-пам'яті;
- $S_0$  – загальна кількість звернень до пам'яті;
- $S_k$  – число кеш-влучень;
- $S_{k_w}$  – число наскрізних записів при кеш-попаданні (в режимі наскрізний запису);
- $S_{k_w}^i$  – число зворотних записів (в режимі зворотного запису).

## 2 РОЗРАЗУНКОВО-ГРАФІЧНА РОБОТА №1. ПРИНЦИПИ РОБОТИ КЕШ-ПАМ'ЯТІ

### Мета роботи

Перевірити роботу різних алгоритмів заміщення при різних режимах запису.

### 2.1 Завдання 1

В якості завдання пропонується деяка коротка програма (таблиця 2.2), яку необхідно виконати з підключеною кеш-пам'яттю (розміром 4 і 8 комірок) в покроковому режимі для таких двох варіантів алгоритмів заміщення (2.1) див. так само теоретичні відомості.

Таблиця 2.1 – Пояснення до варіантів завдання

Номер варіанта	Режим запису	Алгоритм заміщення
1,7,11	Наскрізна	СЗ, без урахування біта запису
	Зворотня	О, з урахування біта запису
2,5,9	Наскрізна	БИ, без урахування біта запису
	Зворотня	О, з урахування біта запису
3,6,12	Наскрізна	О, без урахування біта запису
	Зворотня	СЗ, з урахування біта запису
4,8,10	Наскрізна	БИ, без урахування біта запису
	Зворотня	БИ, з урахування біта запису

Таблиця 2.2 – Варіанти завдань

№ вар.	Номера команд програми						
	1	2	3	4	5	6	7
1	RD #12	WR 10	WR @10	ADD 12	WR R0	SUB 10	PUSH R0
2	RD #65	WR R2	MOV R4,R2	WR 14	PUSH R2	POP R3	CALL 002
3	RD #16	SUB #5	WR 9	WR @9	WR R3	PUSH R3	POP R4
4	RD #99	WR R6	MOV R7,R6	ADD R7	PUSH R7	CALL 006	POP R8
5	RD #11	WR R2	WR -@R2	PUSH R2	CALL 005	POP R3	RET
6	RD #19	SUB #10	WR 9	ADD #3	WR @9	CALL 006	POP R4
7	RD #6	CALL 006	WR 11	WR R2	PUSH R2	RET	JMP 002
8	RD #8	WR R2	WR @R@+	PUSH R2	POP R3	WR -@R3	CALL 003
9	RD #13	WR 14	WR @14	WR @13	ADD 13	CALL 006	RET
10	RD #42	SUB #54	WR 16	WR @16	WR R1	ADD @R1	PUSH R1
11	RD #10	WR R5	ADD R5	WR R6	CALL 005	PUSH R6	RET
12	JMP 006	RD #76	WR 14	WR R2	PUSH R2	RET	CALL 001

Не слід розглядати задану послідовність команд як фрагмент програми. Деякі конструкції, наприклад, послідовність команд push R6, ret в загальному випадку не повертає програму в точку виклику підпрограми. Такі групи команд

введені в завдання для того, щоб звернути увагу на особливості функціонування стеку.

### **2.1.1 Порядок виконання роботи**

Ввести в модель навчальної ЕОМ текст свого варіанту програми (таблиця 2.2), асемблювати його і зберегти на диску у вигляді txt-файл.

Встановити параметри кеш-пам'яті розміром 4 комірки, вибрати режим запис та алгоритм заміщення згідно з першим рядком свого варіанту з таблиці 2.1.

В пошаговому режимі виконати програму, фіксуєючи після кожного кроку стан кеш-пам'яті.

Для однієї з команд запису (WR) перейти в режим Такт та зазначити, в яких мікрокомандах відбувається зміна кеш-пам'яті.

Для кеш-пам'яті розміром 8 комірок встановити параметри відповідно з другим рядком свого варіанту з таблиці 2.1 і виконати програму в по-шаговому режимі ще раз, фіксуєючи послідовність номерів заміщаючих комірок кеш-пам'яті.

### **2.1.2 Зміст звіту**

1. Варіант завдання – текст програми та режими кеш-пам'яті.
2. Послідовність станів кеш-пам'яті розміром 4 комірки при одноразовому виконанні програми (команди 1-7).
3. Послідовність мікрокоманд при виконанні команди wr з відміткою тих мікрокоманд, в яких можлива модифікація кеш-пам'ять.
4. Для варіанта кеш-пам'яті розміром 8 комірок - послідовність номерів заміщуються комірки кеш-пам'яті для другого варіанту параметрів кеш пам'яті при дворазовому виконанні програми (команди 1-7).

### **2.1.3 Контрольні питання**

5. У чому сенс включення кеш-пам'яті до складу ЕОМ?
6. Як працює кеш-пам'ять в режимі зворотного запису? Наскрізнний запису?
7. Як залежить ефективність роботи ЕОМ від розміру кеш-пам'яті?
8. В яку комірку кеш-пам'яті буде поміщатися чергове слово, якщо вільні комірки відсутні?
9. Які алгоритми заміщення комірок кеш-пам'яті вам відомі?

### **2.2 Алгоритми заміщення рядків кеш-пам'яті**

Мета – вивчення впливу параметрів кеш-пам'яті і вибраного алгоритму заміщення на ефективність роботи системи.

Ефективність у даному випадку оцінюється числом кеш-включень по відношенню до загального числа звернень до пам'яті. Враховуючи різницю в



алгоритмах в режимах наскрізного і зворотнього запису, ефективність використання кеш-пам'яті обчислюється виразами (1.1) і (1.2) відповідно для наскрізного і зворотнього запису (розділ 1.8).

Очевидно, ефективність роботи системи з кеш-пам'яттю буде залежати не тільки від параметрів кеш-пам'яті і вибраного алгоритму заміщення, але і від класу задачі. Так, лінійні програми повинні добре працювати з алгоритмами заміщення типу чергу, а програми з великим числом умовних переходів, що залежать від випадкових вхідних даних, можуть давати непогані результати з алгоритмами випадкового заміщення. Можна припустити, що програми, що мають велике число повторюваних ділянок (часто викликаються підпрограми та/або циклів) при інших рівних умовах забезпечать більш високу ефективність застосування кеш-пам'яті, ніж лінійні програми. І, зрозуміло, на ефективність безпосередньо повинен впливати розмір кеш-пам'яті.

Для перевірки висловлених вище припущень виконується дана лабораторна робота.

## **2.3 Завдання 2**

В даній частині лабораторної роботи всі варіанти однакові завдання: дослідити ефективність роботи кеш-пам'яті при виконанні двох різнотипних програм, написаних і налагоджених вами при виконанні лабораторної роботи № 2 і 3.

### **2.3.1 Порядок виконання роботи**

1. Завантажити модель навчальної ЕОМ налагоджену програму з лабораторної роботи № 2.

2. У меню Робота встановити режим Кеш-пам'ять.

3. У меню Вид вибрати команду Кеш-пам'ять, відкривши тим самим вікно Кеш пам'ять, в ньому натиснути першу зліва кнопку на панелі інструментів, відкривши діалогове вікно Параметри кеш-пам'яті і встановити наступні параметри кеш-пам'яті: розмір – 4, режим запису - наскрізна, алгоритм заміщення – випадкове, без урахування біта запису (W).

4. Запустити програму в автоматичному режимі; по закінченні роботи переглянути результати роботи кеш-пам'яті у вікні Кеш-пам'ять, обчислити значення коефіцієнта ефективності і записати в комірку таблиці 6.3, позначену зірочкою.

5. Вимкнути кеш-пам'ять моделі (Робота | Кеш-пам'ять) і змінити один з її параметрів - встановити прапор з урахуванням біта запису (у вікні Параметри кеш-пам'яті).

6. Повторити п. 4, помістивши значення отриманого коефіцієнта ефективності у наступну комірку праворуч.

7. Послідовно змінюючи параметри кеш-пам'яті, повторити пп. 3-5, заповнюючи всі комірки таблиці 6.3.

8. Повторити всі дії, описані в пп. 1-7 для програми з лабораторної роботи № 4, заповнюючи другу таблицю за формою таблиці 2.3.

### 2.3.2 Зміст звіту

Дві таблиці за формою таблиці 2.3 з результатами моделювання програм з лабораторних робіт № 2 і 3 при різних режимах роботи кеш-пам'яті.

#### Висновки, що пояснюють отримані результати

### 2.3.3 Контрольні питання

1. Як працює алгоритм заміщення черга при встановленому прапорці з урахуванням біта запису в діалоговому вікні Параметри кеш-пам'яті?

2. Який алгоритм заміщення буде найбільш ефективним у разі застосування кеш-пам'яті великого обсягу (в кеш-пам'яті цілком поміщається програма)?

3. Як позначиться на ефективності алгоритмів заміщення облік значення біта запису  $W$  при роботі кеш-пам'яті в режимі зворотнього запису? Наскрізного запису?

4. Для яких цілей в структуру комірки кеш-пам'яті включений біт використання. Як встановлюється і скидається цей біт?

Таблиця 2.3 – Результати експерименту

Спосіб	Наскрізний запис					
	Випадковий запис		Черга		Біт U	
Розмір	без W	з W	без W	з W	без W	з W
4						
8						
16						
32						
Спосіб	Зворотній запис					
	Випадковий запис		Черга		Біт U	
Розмір	без W	з W	без W	з W	без W	з W
4						
8						
16						
32						

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

- 1 Акушский І. Я., Юдицький Д. І. Машинна арифметика в залишкових класах. — М: Сов. радіо, 1968.
- 2 Баранов С. В. Синтез мікропрограмних автоматів. — Л.: Енергія, 1974.
- 3 Григор'єв Ст. Л. Мікропроцесор і486. Архітектура та програмування. в чотирьох книгах. — М.: ГРАНАЛ, 1993.
- 4 Жмакин А. П. Архітектура ЕОМ. — СПб.:БХВ-Петербург, 2006.
- 5 Жмакин А. П., Тітов В. С. Однокристалні мікроеом в системах управління: Навчальний посібник. — Курськ: Курськ, держ. тех. ун-т, 2002.
- 6 Закревський А. Д. Алгоритми синтезу дискретних автоматів. — М: Наука, 1971.
- 7 Каган Б. М. ЕОМ і системи. — М: Вища Школа, 1985.
- 8 Майоров С. А., Новіков Р. А. Принципи організації ЦВМ.— Л.: Машинобудування, 1974.
- 9 Савельєв А. Я. Прикладна теорія цифрових автоматів. — М: Вища школа, 1987.
- 10 Соловійов Р. Н. Арифметичні пристрої ЦОМ. — М: Енергія, 1978.
- 11 Стариченко Б. Е. Теоретичні основи інформатики.— М.: Гаряча лінія — Телеком, 2003.
- 12 Таненбаум Е Архітектура комп'ютера. 4-е изд. — СПб.: Пітер, 2003.
- 13 Хамахер К., Вранешич З., Заки С. Організація ЕОМ. 5-е изд.— СПб.: Пітер, 2003.
- 14 Хвоц С. Т. та ін Мікропроцесори і мікроеом в системах автоматичного управління: Довідник. — Л.: Машинобудування, 1987.
- 15 Юров В. І. Assembler. — СПб.: Пітер, 2003.