

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ І. І. МЕЧНИКОВА  
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ  
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

## **АРХІТЕКТУРА КОМП'ЮТЕРІВ ТА НИЗЬКОРІВНЕВЕ ПРОГРАМУВАННЯ**

Методичні вказівки  
до виконання курсового проєкту для здобувачів  
першого (бакалаврського) рівня вищої освіти  
спеціальності 123 Комп'ютерна інженерія

ОДЕСА  
Олді+  
2024

УДК 004.424:004.27]:378.2:001.81(072)

A878

**Укладачі:**

**Ю. О. Гунченко**, завідувач кафедри комп'ютерних систем та технологій Одеського національного університету імені І. І. Мечникова;

**І. В. Шаріпова**, старший викладач кафедри комп'ютерних систем та технологій Одеського національного університету імені І. І. Мечникова;

**Ю. М. Берков**, старший викладач кафедри комп'ютерних систем та технологій Одеського національного університету імені І. І. Мечникова.

**Рецензенти:**

**Андрій Чепок**, кандидат фізико-математичних наук, доцент кафедри КСТ факультету математики, фізики та інформаційних технологій ОНУ імені І. І. Мечникова;

**Микола Малаксіано**, доктор технічних наук, професор, завідувач кафедри Технічна кібернетика й інформаційні технології ім. професора Р. В. Меркта Одеського національного морського університету.

*Рекомендовано вченою радою  
факультету математики, фізики та інформаційних технологій  
ОНУ імені І. І. Мечникова.  
Протокол №6 від 22 березня 2024 р.*

**Архітектура** комп'ютерів та низькорівневе програмування :  
A878 метод. вказівки до виконання курсового проєкту для здобувачів першого (бакалавр.) рівня вищої освіти спеціальності 123 Комп'ютерна інженерія / уклад.: Ю. О. Гунченко, І. В. Шаріпова, Ю. М. Берков. – Одеса : Олді+, 2024. – 40 с.

*У методичних вказівках розглянуто структуру проєкту, обсяг, порядок його виконання, вимоги до оформлення пояснювальної записки, правила захисту та оцінки курсового проєкту. Наведені перелік завдань з темами завдань, довідкові матеріали, необхідні для виконання окремих частин та список рекомендованої літератури.*

УДК 004.424:004.27]:378.2:001.81(072)

## Зміст

ВСТУП.....	4
1. МЕТА ТА ЕТАПИ КУРСОВОГО ПРОЄКТУВАННЯ.....	5
2. ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ .....	6
3. СКЛАД І СТРУКТУРА КУРСОВОГО ПРОЄКТУ .....	15
4. ВИМОГИ ДО ЗМІСТУ ЕЛЕМЕНТІВ КУРСОВОГО ПРОЄКТУ .....	16
5. ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ .....	22
5.1 Загальні вимоги.....	22
5.2 Побудова пояснювальної записки .....	22
5.3 Нумерація складових частин.....	23
5.4 Викладення тексту пояснювальної записки .....	23
5.5. Оформлення ілюстрацій .....	24
5.6. Побудова таблиць.....	25
6. ЗАХИСТ КУРСОВОГО ПРОЄКТУ .....	26
7. СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ .....	27
Додаток А Титульний аркуш пояснювальної записки .....	28
Додаток Б Формати команд.....	29
Додаток С Типи адресації.....	31

## ВСТУП

Основу сучасних інформаційних технологій (ІТ) та їх базис становлять апаратні засоби комп'ютерної техніки. У зв'язку з цим важливе значення в підготовці фахівців в області ІТ та комп'ютерної інженерії надається вивченню архітектури комп'ютерів, зокрема основних відомостей, щодо їх апаратних та програмних засобів, організації роботи в цілому, способів подання даних та системного і низькорівневого програмування. Таким чином метою курсу “Архітектура комп'ютерів та низькорівневе програмування” є вивчення методів та засобів проектування сучасних комп'ютерів, визначення ефективного використання у практиці, способи подання програм і даних, а також сучасні архітектурні рішення.

Курсовий проєкт (КП) з дисципліни “Архітектура комп'ютерів та низькорівневе програмування” виконується з метою закріплення та поглиблення теоретичних знань та вмінь, набутих у процесі засвоєння всього навчального матеріалу дисципліни. Виконання КП є важливим етапом у підготовці до виконання дипломного проєкту (роботи) майбутнього фахівця з комп'ютерної інженерії.

Час, потрібний для виконання курсового проєкту – до 60 годин самостійної роботи.

Курсовий проєкт присвячується розробці архітектури системи команд обчислювальної машини із виділеним доступом до пам'яті. Під архітектурою системи команд (АСК) прийнято визначати обсяг обчислювальної машини, які доступні програмісту.

## 1 МЕТА, ЕТАПИ ПРОВЕДЕННЯ ТА ЗАХИСТ ЛАБОРАТОРНИХ РОБІТ

Мета курсового проектування:

- закріплення, поглиблення та узагальнення теоретичних знань і розвиток навичок їх практичного застосування в галузі архітектури комп'ютерів та низькорівневого програмування;
- самостійне опрацювання здобувачом архітектурного, схемного та алгоритмічного описів ЕОМ;
- уміння користуватися відповідною довідковою літературою, державними стандартами.

Курсове проектування містить такі етапи:

- визначення теми і оформлення завдання на курсове проектування;
- безпосереднє виконання курсового проекту;
- оформлення пояснювальної записки та графічного матеріалу;
- захист курсового проекту.

Тематику курсових проектів та відповідних методичних вказівок затверджує завідувач кафедри.

Здобувачу по узгодженню з викладачем може бути надана індивідуальна тема КП, яка відповідає його перспективним інтересам.

Завдання на курсове проектування видається за підписом керівника, датується днем видачі та реєструється у журналі обліку і контролю на кафедрі.

Здобувач не рідше одного разу на два тижні зобов'язаний інформувати керівника про виконану роботу. Керівник повинен розвивати у здобувачів творчі навички. Необхідно пам'ятати, що відповідальність за всі прийняті в курсовому проекті рішення несе автор – здобувач.

Якщо керівник не допускає здобувача до захисту, то це питання обговорюється на засіданні кафедри у його присутності.

Захист курсових проектів – це форма перевірки якості виконання програми та знань, отриманих під час виконання лабораторних робіт та на лекціях.

Під час захисту здобувач робить доповідь по суті програми та відповідає на запитання.

Якість протоколу та його захист оцінюється в балах (0- 5), за шкалою ECTS (A, B, C, D, E, FX, F) та за національною шкалою «відмінно», «добре», «задовільно», «незадовільно».

## 2 ЗАВДАННЯ НА КУРСОВИЙ ПРОЄКТ

### 2.1. Предмет проєктування

Курсовий проєкт присвячується розробці архітектури системи команд обчислювальної машини із виділеним доступом до пам'яті. Під архітектурою системи команд (АСК) прийнято визначати обсяги обчислювальної машини, які і доступні програмісту.

АСК можна розглядати як лінію узгодження потреб розробників програмного забезпечення з можливостями творців апаратури обчислювальної машини.

### 2.2. Загальні вимоги до системи команд, що розробляються

Система команд має відповідати наступним вимогам:

- операції звернення до пам'яті відокремлені від операцій, пов'язаних з обробкою даних;
- операції, пов'язані з обробкою та перетворенням даних, виконуються за принципом «реєстр-реєстр».
- у загальному випадку, апаратно підтримуються операції над цілими числами зі знаком і без знака, а також числами, представленими у форматі з плаваючою точкою (ПТ);
- система команд має бути функціонально повною;
- процесор повинен мати систему переривань.

### 2.3. Вихідні дані для курсового проєктування

Вихідні дані для виконання курсового проєкту визначаються як загальними вимогами до процесора, що розробляється, так і варіантом завдання.

Варіанти завдань наведено у табл. 1.1.

Варіанти завдань включають такі вихідні дані для проєктування:

- перелік типів даних, що підтримуються апаратно;
- адресність операційних команд;
- способи адресації;
- основні характеристики регістрової пам'яті (кількість реєстрів, їх тип);
- ширина шини.

У процесі розробки архітектури визначаються:

1. Формати команд та даних.
2. Реєстрова модель процесора (включаючи користувальницькі та системні регістри).
3. Система команд (включаючи семантику їх виконання та встановлення прапорців).

Апаратно можуть підтримуватись такі формати:

- 8-розрядні цілі числа зі знаком та без знака;
- 16-розрядні цілі числа зі знаком та без знака;
- 32-розрядні цілі числа зі знаком і без знака.
- 64-розрядні цілі числа зі знаком і без знака,
- 32-розрядні числа з плаваючою точкою.

- 64-розрядні числа з плаваючою точкою.

Для тих варіантів, у яких передбачено 32 і 64 розрядні формати, потрібно реалізувати їх для цілих чисел зі знаком і без знака та у форматі з ПТ.

Система команд має бути функціонально повною та включати як мінімум наступні команди:

- 1) звернення до пам'яті з читання та запису;
- 2) цілочисленні арифметичні для чисел зі знаком і без знака (додавання, віднімання, множення, розподіл, порівняння);
- 3) арифметичні з ПТ (додавання, віднімання, множення, розподіл, порівняння);
- 4) логічні (порозрядне "І", "АБО" і "Виключає АБО");
- 5) зрушень на довільне число тактів;
- 6) умовних та безумовних переходів;
- 7) роботи з підпрограмами;
- 8) завантаження в реєстри безпосередніх операндів;
- 9) введення-виведення;
- 10) управління обчислювальним процесом.

Поняття адресності належить до операційним командам, тобто. командам, що виконують дії з обробки даних. Наприклад, немає сенсу використовувати триадресний формат для команд, що реалізують операції переходу. Таким чином, задана адресність для розробляється системи команд є максимальною.

Усього передбачено використання 7 основних способів адресації пам'яті: безпосередня (БА), пряма (ПА), непряма (НА), непряма реєстрова (НРА), базова реєстрова адресація (БРА), індексна адресація (ІА) та відносна адресація (ВА). Реалізація механізмів безпосередньої адресації тим варіантам, у яких це передбачено завданням, необхідна у разі, коли необхідно забезпечити функціональну повноту системи команд. Наприклад, якщо передбачається наявність керуючих реєстрів, слід передбачати механізми їх завантаження.

Для реєстрової пам'яті задаються кількість реєстрів та їх тип, розрядність вибирається відповідно до найбільшої розрядності передбачуваних форматів даних. Реєстри можуть бути або універсальними (У), або функціонально орієнтованими (ФО). У першому випадку йдеться про реєстри загального призначення (РЗП), які призначені для зберігання як адрес, так і цілих чисел.

Функціональна орієнтація реєстрів загального призначення подразнює їх розбиття на групи залежно від функціонального призначення (наприклад, реєстри для зберігання даних, індексів (моді-фіксаторів), бази і т.ін.). У середині груп можливе додаткове розбиття (наприклад, реєстр бази даних, реєстр бази програмного коду і т.ін.).

На етапі визначення структури реєстрової пам'яті вибираються:

- кількість реєстрів різних типів та їх розрядність;
- склад та структура реєстру прапорів.

У завданні визначається обсяг оперативної пам'яті, до якого необхідно забезпечити доступ, в режимі пам'яті, що прямо адресується. Розрядність шини даних визначає розрядність пам'яті.

#### 2.4. Результати проектування

У процесі виконання курсового проекту розробляються:

- 1) формати команд та даних;
- 2) регістрова модель процесора, включаючи користувальницькі та системні регістри;
- 3) система команд, включаючи семантику їх виконання та встановлення прапорів;
- 4) приклади програм у кодах розробленої системи команд.

#### 2.5. Варіанти завдань

Варіанти завдань для курсового проектування наведено у табл. 1.1. та 1.2.

У таблиці 1.1 представлені такі вихідні дані для проектування:

- перелік типів даних, що підтримуються апаратно;
- адресність операційних команд;
- способи адресації;
- основні характеристики регістрової пам'яті (кількість реєстрів, їх тип).

Таблиця 1.1

Вихідні дані для курсового проектування

№	Формат даних				Адресність	Спосіб адресації							Регістрова пам'ять	
	8	16	32	64		БА	ПА	НА	НРА	БРА	ІА	ВА	Кількість	Тип
1	+	+	+	-	3	+	+	+	-	-	+	-	8	У
2	-	+	+	+	2	+	+	-	+	-	-	+	8	У
3	+	-	+	+	3	+	+	-	-	-	+	-	8	У
4	+	+	-	+	3	+	+	+	-	+	-	-	8	У
5	+	+	+	-	2	-	+	-	+	+	-	+	8	У
6	-	+	+	+	3	+	+	-	-	+	+	-	8	У
7	+	-	+	+	3	+	+	+	-	-	-	+	8	ФО
8	+	+	-	+	2	+	+	-	+	-	-	+	8	ФО
9	+	+	+	-	3	+	+	-	-	+	+	-	8	ФО
10	-	+	+	+	3	+	+	+	-	+	-	-	8	ФО
11	+	-	+	+	2	-	+	-	+	+	-	+	8	ФО
12	+	+	-	+	3	+	+	-	-	-	+	-	8	ФО
13	+	+	+	-	3	+	+	+	-	-	-	-	16	У
№	Формат даних				Адр	Спосіб адресації							Регістрова пам'ять	

	8	16	32	64		8	16	32	64		8	16	32	64
14	-	+	+	+	2	+	+	+	-	+	-	+	16	Y
15	+	-	+	+	3	+	+	-	+	+	+	-	16	Y
16	+	+	-	+	3	-	+	-	-	-	-	+	16	Y
17	+	+	+	-	2	+	+	+	-	-	-	+	16	Y
18	-	+	+	+	3	+	+	-	+	-	+	-	16	Y
19	+	-	+	+	3	+	+	-	-	-	-	+	16	ΦO
20	+	+	-	+	2	+	+	+	-	+	-	-	16	ΦO
21	+	+	+	-	3	-	+	-	+	+	+	-	16	ΦO
22	-	+	+	+	3	+	+	-	-	+	-	+	16	ΦO
23	+	-	+	+	2	+	+	+	-	-	-	-	16	ΦO
24	+	+	-	+	3	+	+	-	+	-	+	+	16	ΦO
25	+	+	+	-	3	+	+	-	-	+	-	+	32	Y
26	-	+	+	+	2	-	+	+	-	+	-	-	32	Y
27	+	+	-	+	3	+	+	-	+	+	+	-	32	Y
28	+	+	-	+	3	+	+	-	-	-	-	+	32	Y
29	-	+	+	+	2	+	+	+	-	-	-	-	32	Y
30	-	+	+	+	3	+	+	+	-	+	+	-	32	Y
31	+	+	+	-	3	-	+	-	+	+	-	+	32	ΦO
32	-	+	+	+	2	+	+	-	-	-	-	-	32	ΦO
33	+	-	+	+	3	+	+	+	-	+	+	+	32	ΦO
34	+	+	-	+	2	+	+	-	+	-	-	+	32	ΦO
35	+	+	+	-	3	+	+	-	-	-	-	-	32	ΦO
36	-	+	+	+	2	-	+	+	-	-	+	-	32	ΦO
37	+	-	+	+	2	+	+	-	+	+	-	+	64	Y
38	+	+	-	+	2	+	+	-	-	+	-	-	64	Y
39	+	+	+	-	2	+	+	+	-	+	+	-	64	Y
40	-	+	+	+	3	+	+	-	+	-	-	+	64	Y
41	+	-	+	+	2	-	+	-	-	-	-	-	64	Y
42	+	+	-	+	3	+	+	+	-	+	+	+	64	Y
43	+	+	+	-	2	+	+	-	+	+	-	+	64	ΦO
44	-	+	+	+	2	+	+	-	-	+	-	-	64	ΦO
45	+	-	+	+	3	+	+	+	-	+	+	-	64	ΦO
46	+	+	-	+	3	-	+	+	-	-	-	+	64	ΦO
47	+	+	+	-	2	+	+	-	+	-	-	-	64	ΦO
48	-	+	+	+	3	+	+	-	-	+	+	-	64	ΦO

## Завдання для програмування у кодах

№ варіанту	Лінійні обчислення	Цикли	Масиви
1	1	5	1
2	7	1	2
3	8	11	3
4	3	18	4
5	13	9	5
6	20	10	6
7	6	13	7
8	12	7	8
9	11	12	9
10	2	8	10
11	17	2	11
12	9	20	12
13	14	4	12
14	16	6	11
15	7	20	10
16	19	16	9
17	10	18	8
18	3	10	7
19	13	5	6
20	8	19	5
21	1	7	4
22	15	2	3
23	6	16	2
24	18	9	1
25	16	13	1
26	17	16	3
27	4	8	5
28	7	11	7
29	5	4	9
30	11	15	11
31	6	14	12
32	3	6	10
33	14	19	8
34	10	5	6
35	9	12	4

## ЗАВДАННЯ НА ЛІНІЙНІ ОБЧИСЛЕННЯ

1. Є дев'ятиповерховий будинок, на кожному сходову площадку якого виходить по 4 квартири. Визначити номер під'їзду, в якому розташована квартира з номером N.
2. Якщо ціле число M ділиться націло на ціле число N, то вивести на екран приватне від поділу, інакше встановити індикатор помилки в 1.
3. Дано двозначне натуральне число N. Визначити, яка його цифра більша. Вивести на екран 1, якщо перша, та 2, якщо друга.
4. Дано натуральне число N. Якщо воно закінчується парною цифрою, вивести на екран 1, якщо непарною – 0.
5. Дано два цілих числа. Обчислити суму цих чисел, якщо перше число менше другого, інакше – їх різницю.
6. Дано три цілих числа. Обчислити залишок від поділу перших двох чисел на третє.
7. Обчислити периметр паралелограма, сторона якого вдвічі більша за непаралельну їй сторону.
8. Перевірити, чи належить число X, введене з клавіатури, інтервалу (M; N). У разі позитивної відповіді вивести 1, у разі негативного - 0.
9. Визначити, чи є трикутник зі сторонами a, b та c рівно-боковим.
10. Дано двозначне натуральне число N. Визначити, чи входить до нього цифра M. У разі позитивної відповіді вивести 1, у разі негативної – 0.
11. Дано двозначне натуральне число N. Чи вірно, що цифри запису цього числа різні? У разі позитивної відповіді вивести 1, у разі негативної – 0.
12. Ввести довжину a та ширину b прямокутника. Обчислити площу та периметр прямокутника.
13. Ввести з клавіатури три натуральні числа і перевірити можливість існування трикутника з довжинами сторін, рівними введеним числам. У разі позитивної відповіді вивести 1, у разі негативної – 0.
14. Дано два цілих числа, не рівних нулю. Визначити, чи ці цифри мають однакові знаки. У разі позитивної відповіді вивести 1, у разі негативної – 0.
15. Знайти значення функції  $y = \frac{x+z^2}{z-x}$  при заданих значеннях x та z.
16. Знайти значення функції  $y = \frac{(a-2b)^3}{b-5}$  при заданих значеннях a та b.
17. Знайти значення функції  $y = \frac{4x^3+10z}{xz^2}$  при заданих значеннях x і z.
18. Знайти значення функції  $y = \frac{3z \% x}{4-x}$  при заданих значеннях x та z.

## ЗАВДАННЯ НА ЦИКЛИ

1. Обчислити  $y = \prod_{i=1}^n \left( \frac{10}{(a+i)^2} + 5 \right)$

2. Обчислити  $y = \begin{cases} \prod_{i=1}^n \frac{i^3}{a-b} & \text{при } a \neq b \\ \frac{a-b}{a+b} & \text{в противном случае} \end{cases}$

3. Обчислити  $y = \begin{cases} \sum_{i=1}^n \frac{a+i}{b^4} & \text{при } b > 0 \\ \frac{b}{a^2+10} & \text{в противном случае} \end{cases}$

4. Обчислити  $y = \sum_{i=1}^n \frac{(a+b)^2}{b-a*i}$

5. Обчислити  $y = \prod_{i=1}^n \frac{a^2 + b \% i}{a - 16}$

6. Обчислити  $y = \sum_{i=2}^n (i+1)^2 \% (i-1)^2$

7. Обчислити  $y = \prod_{i=1}^n \frac{100}{(a-i)^2 + b}$

8. Обчислити  $y = \sum_{i=1}^n \frac{(a+b)*i^2}{a}$

9. Обчислити  $y = \prod_{i=1}^n \frac{i-a}{(i+b)^2 - 1}$

10. Обчислити  $y = \sum_{i=1}^n \left( \frac{a+i}{a^2 - b^2} * i \right)$

11. Обчислити  $y = \sum_{i=1}^n \frac{a + \frac{b}{i}}{b^2 - 5}$

12. Обчислити  $y = \prod_{i=1}^n (i \% a + 2) + \sum_{i=1}^n \frac{4i}{a}$

13. Обчислити  $y = \prod_{i=1}^n \frac{a^4 + 1}{a - i}$

14. Обчислити  $y = \sum_{i=1}^n (3i^2 + 2a) \% i$

15. Обчислити  $y = \prod_{i=1}^n \left( \frac{a+b}{a-i} * i \right)$

16. Обчислити  $y = \begin{cases} \prod_{i=1}^n \frac{a+i}{a*i} & \text{при } a \neq 0 \\ (a+b)^3 & \text{в противном случае} \end{cases}$

17. Обчислити  $y = \begin{cases} \sum_{i=1}^n \left( \frac{100}{a} + b + i \right) & \text{при } a \neq 0 \\ (a+b) \% (a-b) & \text{в противном случае} \end{cases}$

18. Обчислити  $y = \sum_{i=1}^n (i-1)^2 + \prod_{i=1}^n (i+1)^2$

19. Ввести з клавіатури 2 числа – M та N (M < N). Знайти суму кубів всіх парних позитивних цілих чисел в інтервалі (M; N).

20. Ввести з клавіатури 2 числа – M та N (M < N). Знайти проведення квадратів всіх непарних негативних цілих чисел в інтервалі (M; N).

### ЗАВДАННЯ НА МАСИВИ

*Примітка: кількість елементів масиву n вибирається самостійно.*

1. Скласти програму для обчислення суми парних елементів цілісного масиву, кожен із яких перед обчисленням суми має бути зменшений удвічі.
2. Скласти програму для пошуку в цілісному масиві кількості і добутку елементів, що стоять перед першим елементом із заданим значенням.
3. Скласти програму для обчислення непарних елементів цілісного масиву, кожен з яких перед обчисленням повинен бути зменшений на 5.
4. Скласти програму для пошуку в цілісному масиві кількості та суми елементів, що стоять після елемента із заданим значенням.
5. Скласти програму для обчислення твору позитивних елементів цілісного масиву, що мають парні номери.
6. Скласти програму для підрахунку в цілісному масиві суми парних і суми непарних елементів.

7. Скласти програму для обчислення суми негативних елементів цілісного масиву, що мають непарні номери.
8. Скласти програму для обчислення суми невід'ємних елементів целочисленного масиву, малого заданого невід'ємного цілого числа.
9. Скласти програму для підрахунку в цілісному масиві кількості елементів, рівних заданому цілому числу.
10. Скласти програму для підрахунку в цілісному масиві добутку елементів, наступних за першим елементом із заданим значенням.
11. Скласти програму для підрахунку в цілісному масиві кількості нулів, що стоять за елементом із заданим номером.
12. Скласти програму для підрахунку в цілісному масиві добутку квадратів елементів, номери яких кратні заданому числу.

### 3 СКЛАД І СТРУКТУРА КУРСОВОГО ПРОЄКТУ

Курсовий проєкт являє собою сукупність розроблених програм, схем, описів спеціалізованою мовою та документів, що розкривають суть розробленого: пояснювальної записки, креслень, схем, ілюстративного матеріалу, які виконуються з обов'язковим додержанням вимог чинних стандартів.

Структура курсового проєкту з курсу “Архітектура комп'ютерів та низькорівневе програмування”:

- титульний аркуш;
- анотація;
- завдання на виконання курсового проєкту;
- пояснювальна записка.

Структура пояснювальної записки:

- зміст;
- перелік умовних позначень (за потреби);
- вступ;
- основна частина;
- висновки та рекомендації;
- список використаних джерел;
- додатки.

## 4 ВИМОГИ ДО ЗМІСТУ СТРУКТУРНИХ ЕЛЕМЕНТІВ КУРСОВОГО ПРОЄКТУ

Зразок оформлення титульного аркуша курсового проєкту наведено в додатку А.

Анотація пояснювальної записки – це коротка характеристика роботи, призначена для ознайомлення зі змістом курсового проєкту. Вона виконується державною мовою. Анотація показує відмінні риси та переваги виконаної роботи, містить тему та основну мету роботи, крім цього вона може перераховувати основні положення роботи.

Анотація має бути стислою, але інформативною, і містити відомості, які дозволяють мати певне уявлення про проєкт. Вона повинна містити:

- відомості про обсяг пояснювальної записки, кількість ілюстрацій, таблиць, додатків, літературних джерел;
- основний текст анотації;
- перелік ключових слів (словосполучень).

Текст анотації має відображати подану в пояснювальній записці інформацію в такій послідовності:

- об'єкт дослідження або розроблення;
- мета роботи, методи дослідження, застосовані технічні та програмні засоби;
- основні конструктивні, технологічні та інші характеристики показники;
- результати та стислі висновки.

Анотація займає не більше однієї сторінки пояснювальної записки.

Перелік ключових слів (словосполучень), що є визначальними для розкриття суті курсового проєкту, наводять після основного тексту анотації. Цей перелік повинен містити від 5 до 15 слів (словосполучень) в називному відмінку через кому.

Зміст пояснювальної записки розміщують безпосередньо після анотації, починаючи з нової сторінки.

До змісту включають заголовки таких структурних елементів пояснювальної записки: ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ; ВСТУП, послідовно назви (заголовки) всіх розділів, підрозділів; ВИСНОВКИ; СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ; назви додатків. Справа наводять номери сторінок, з яких починається зазначений матеріал.

Усі прийняті у пояснювальній записці малопоширені умовні позначення, символи, одиниці, скорочення та терміни пояснюють у відповідному переліку.

Вступ –орієнтує на подальше розкриття теми проєкту і містить усі необхідні її кваліфікаційні характеристики. Вступ має обсяг 2–3 сторінки і починається з нового аркуша.

У вступі відображаються:

- актуальність, наукове та практичне значення теми проєкту, сучасний стан проблеми, причому відзначаються практично розв'язані задачі, відомі світові тенденції розв'язання задач, обґрунтовується доцільність виконання роботи для розвитку науки, виробництва або навчального процесу, формулюється авторське розуміння проблеми;
- мета та завдання курсового проєктування;
- об'єкт та предмет розгляду;
- посилання на відповідні нормативні документи;
- положення, які виносяться на захист.

Загальна характеристика архітектури системи команд обчислювальної машини складається з відповідей на такі питання:

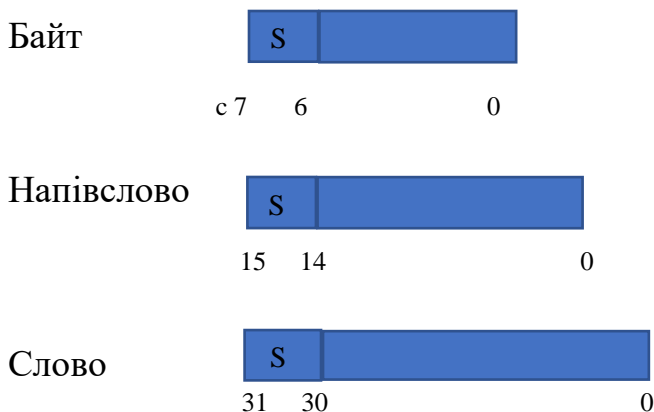
1. Якого виду дані будуть представлені у ВМ та в якій формі?
2. Де ці дані можуть зберігатися, крім основної пам'яті?
3. Яким чином здійснюватиметься доступ до даних?
4. Які операції можуть бути здійснені над даними?
5. Скільки операндів може бути у команді?
6. Як визначатиметься адреса чергової команди?
7. Як буде закодовано команди?

Основна частина пояснювальної записки складається з наступних розділів:

- розділ опису теоретичних засад або аналізу аналогічних алгоритмів, відповідних існуючих інженерних методик та технічних рішень;
- розділ, що присвячений власне розробці;
- інструкцію користувача програмного або апаратного забезпечення, яке було розроблено;
- результати випробувань розроблених моделей, їх порівняння та аналіз;
- висновки.

## 2.1 Вибір форматів даних

На рис. 2.1 показані основні типи даних, що використовуються.



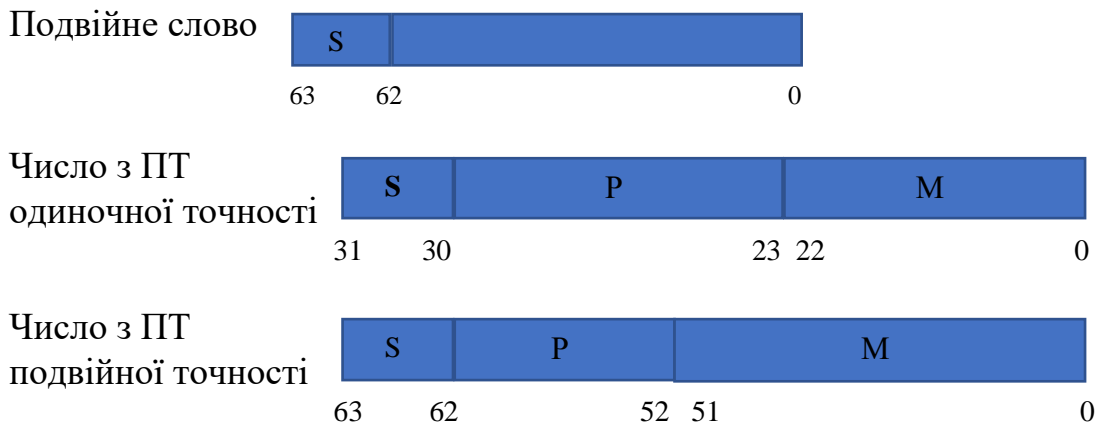


Рис. 2.1 - Основні типи даних, що використовуються

Ціле - числа зі знаком або без знака. Цілі зі знаком подаються у додаткових кодах. Цілі без знака використовуються також для подання адрес. Для представлення чисел з ПТ рекомендується дотримуватися вимог стандарту (ANSI/IEEE Standart 754). При цьому слід дати докладний опис подання особливих ситуацій (нуль, переповнення, NAN).

## 2.2 Визначення моделі пам'яті та структури регістрової пам'яті.

На етапі визначення структури реєстрової пам'яті вибираються:

- число регістрів різних типів та їх розрядність;
- склад та структура регістру прапорців.

Регістри, в свою чергу, можуть поділятися на адресні, цілечисленні і регістри з ПТ. Часто теж самі регістри використовуються як адресні, так і як цілочисленні (РОНи). Крім того, зазвичай є кілька управляючих регістрів.

До складу системних регістрів можуть входити такі регістри, як регістр фізичної адреси таблиці векторів переривання. Один з можливих варіантів організації користувальницької реєстрової пам'яті ЦП показаний на рис. 2.2.

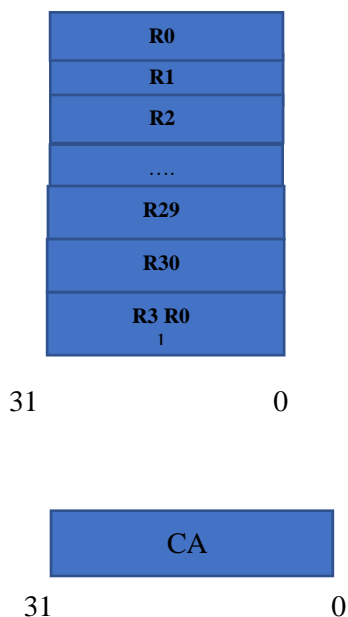




Рисунок 2.2 – Приклад моделі регістрової пам'яті

У структурі, наведеній на рис. 2.2, регістрова пам'ять включає тридцять два 32-розрядних РВН та 32-розрядний лічильник команд. Регістр прапорців містить 4 прапори, які використовуються таким чином:

- Z – ознака нульового результату;
- C - ознака перенесення зі старшого розряду;
- S – знак результату;
- O – ознака переповнення результату.

У регістрі можуть бути й інші прапори, наприклад, дозвіл переривання (I), покроковий режим (T) або режим супервізор/користувач (U).

### 2.3 вибір форматів команд

Формат команди визначає її структуру, тобто кількість двійкових розрядів, що відводяться під всю команду, а також кількість та розташування окремих полів команди. При створенні ВМ вибір формату команди впливає на багато характеристик майбутньої машини. Оцінюючи можливі формати, слід враховувати такі фактори:

- загальна кількість різних команд;
- загальну довжину команди;
- тип полів команди (фіксованої або змінної довжини) та їх довжину;
- простоту декодування;
- адресність та способи адресації.

Для спрощення апаратури та підвищення швидкодії ВМ довжину команди зазвичай вибирають кратну байту, оскільки в більшості ВМ основна пам'ять організована у вигляді 8-бітових осередків.

Кількість двійкових розрядів, що відводяться під код операції, вибирається так, щоб можна було уявити будь-яку з операцій. Якщо система команд передбачає  $N_{op}$  операцій, то мінімальна розрядність поля коду операції  $R_{op}$  визначається наступним чином:  $R_{op} = \lceil \log_2 N_{op} \rceil$ , де результат означає округлення у бік до цілого числа.

При виборі форматів слід виходити з принципів RISC-обробки, відповідно до яких використовується мінімальна кількість форматів команд. Дозволяється використання одного (32-розрядного) формату. Прийнятним можна вважати варіант, коли команди мають різну довжину.

На рис. 2.3 та 2.4 наведено можливі формати команд. На рис. 2.3 показані формати команд, що відповідають другому випадку. Операційні команди (рис. 2.3, а) мають довжину 16 розрядів і містять 4 поля: 7-розрядне поле коду операції (OP) та три 3-розрядні поля для завдання регістрів джерел та приймача. Типова двомісна операція типу додавання виконується за схемою  $R1 \text{ Я } \langle R2 \rangle * \langle R3 \rangle$ . Команди звернення до пам'яті, звернення до підпрограм, переходів, і навіть команди, використовують безпосередні операнди мають довжину 32 розряду (рис. 2.3,б).

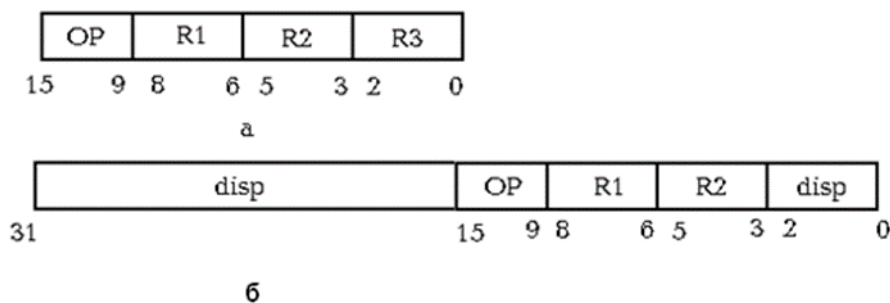


Рис. 2.3 – Приклад із двома форматами команд

Крім полів OP, R1 і R2 є 19-розрядне поле зміщення (disp). Адреса пам'яті визначається додаванням вмісту R2 і disp:  $A = \langle R2 \rangle + \text{disp}$ .

На рис. 2.4 показаний приклад 32-розрядного формату.

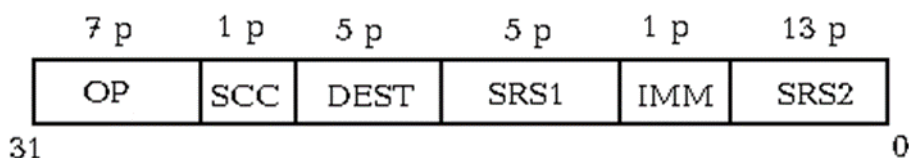


Рис. 2.4 - Приклад 32-розрядного формату

Команда на рис. 2.4 містить 6 полів: 7-розрядне поле коду операції (OP), 1-розрядне поле SCC дозволяє, або забороняє встановлення прапорців. Поля DEST, SRS1 і SRS2 задають, відповідно, регістр-приймач і регістри-джерела даних. Однорозрядне поле IMM визначає вміст 13-розрядного поля SRS2. Якщо  $\text{IMM} = 0$ , то вміст зазначеного поля інтерпретується як номер регістру, в іншому випадку розглядається як 13-розрядна константа. У командах звернення до пам'яті воно використовується як зміщення. При цьому адреса пам'яті визначається як  $A = \text{SRS1} + \text{SRS2}$ .

#### 2.4 Розробка системи команд

Результати проектування слід оформити у вигляді таблиці:

### Система команд

№	Мне- моника	Назва	Зміст	Прапор			Код
				C	Z	S	
1	ADD	Додавання з ФТ	$R1 = \leftarrow \langle R2 \rangle + \langle R3 \rangle$	+	+	+	0000001
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	JMP	Безумовний перехід	$A = \leftarrow \langle R2 \rangle + disp$	-	-	-	1010101

За результатами виконання курсового проєкту необхідно зробити висновки та дати рекомендації по використанню і впровадженню розробленого модуля. Ця частина записки є завершальною, зумовленою логікою виконаного курсового проєктування. Висновки повинні відповідати поставленим раніше завданням, носити конкретний характер, та містити також відомості про виконання, оформлення наукових статей, тез доповідей на конференціях. Його обсяг не повинен перевищувати трьох сторінок.

У переліку використаних джерел наводять усі літературні джерела, які були використані під час курсового проєктування.

## **5 ПРАВИЛА ОФОРМЛЕННЯ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

Результатом виконання курсового проєкту є технічні описи, розрахунки, таблиці, графіки, лістинги, схеми алгоритмів і пояснення до них тощо. Ці матеріали оформляють у вигляді пояснювальної записки і відповідного графічного матеріалу.

Основна з систем державних стандартів, які визначають правила оформлення технічної документації – Єдина система конструкторської документації, яка містить комплекс державних стандартів, що встановлюють правила й положення з розробки, оформлення, обігу конструкторських документів.

Конструкторську документацію складають текстові та графічні документи, які визначають конструкцію технічного виробу і містять дані, потрібні для його розробки, виготовлення, контролю, приймання, експлуатації та ремонту.

### **5.1 Загальні вимоги**

Пояснювальну записку до курсового проєкту оформлюють у відповідності до вимог таких стандартів:

- ДСТУ 3008–2015 «Звіти у сфері науки і техніки. Структура і правила оформлення»;
- ГОСТ 19.105–78 «Загальні вимоги до програмних документів»;
- ГОСТ 19.404–79 «Пояснювальна записка. Вимоги до змісту і оформлення».

Текстовий та графічний матеріали записки друкують комп'ютерним способом на одному боці односторонніх білих аркушів (аркушів) формату А4 (розмір 210 Н 297 мм) через 1,5 міжрядковий інтервал, текст вирівнюють по ширині аркуша.

Текстовий редактор – Word з пакета Microsoft Office, Open Office Writer, Star Office Writer та ін. Шрифт – Times New Roman Cyr, 14.

Обсяг пояснювальної записки для курсового проєкту становить 30–35 сторінок, (без урахування додатків). Виконуючи пояснювальну записку, потрібно витримувати рівномірну щільність, контрастність і чіткість зображення по всьому тексту. Всі лінії, букви, цифри та знаки мають бути однакового чорного кольору по всій записці.

Абзаци в тексті починаються з відступом 15–17 мм (п'ять знаків).

### **5.2 Побудова пояснювальної записки**

Загалом пояснювальна записка містить такі структурні елементи:

«ЗМІСТ», «ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ», «ВСТУП», розділи основної частини, «ВИСНОВКИ», «СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ», додатки.

Розділи, за потреби, поділяють на підрозділи, пункти на підпункти. Розділи і підрозділи повинні мати заголовки, пункти і підпункти можуть мати заголовки.

Заголовки розділів основної частини записки розміщують симетрично в середині рядка і друкують великими буквами (подібно до інших структурних елементів).

Заголовки підрозділів, пунктів і підпунктів починають з абзацу і друкують малими буквами, крім першої великої. Абзац має бути однаковим по всьому тексту записки. Усі заголовки в тексті записки не підкреслюють і в їх кінці крапку не ставлять. Кожний розділ пояснювальної записки, як і інших структурних елементів, починають з нового аркуша (сторінки).

Якщо заголовок складається з двох і більше речень, то їх розділяють крапкою. Перенесення слів у заголовках розділів не допускається. Відстань між заголовком і наступним текстом – 15 мм (1 рядок).

Не допускається розміщувати найменування підрозділу, пункту, а також підпункту, якщо нижче є тільки один рядок тексту.

### **5.3 Нумерація складових частин**

Розділи, підрозділи, пункти і підпункти записки нумерують арабськими цифрами. Порядкові номери записують з абзацу і в кінці їх ставлять крапку.

Розділи записки послідовно нумерують у межах документа, наприклад, 1, 2, 3 і т. д. Підрозділи нумеруються у межах кожного розділу, наприклад, 2.1 (перший підрозділ другого розділу).

Розділи, як і підрозділи, можуть складатися із декількох пунктів. Якщо розділ не має підрозділів, то пункти нумерують в межах розділу; при цьому пункти не мають найменування і записуються з абзацу, наприклад, 4.1. (четвертий пункт першого розділу). Якщо записка має підрозділи, то номер пункту в них складається з номера розділу, підрозділу і порядкового номера пункту, наприклад, 1.2.1, 1.2.2 і т. д.

Заголовки інших структурних елементів пояснювальної записки («ЗМІСТ», «ВСТУП» і т. ін.) не нумерують.

Сторінки записки наскрізно нумерують арабськими цифрами. Титульний аркуш не нумерують. Ілюстрації, схеми, таблиці, які розміщені на окремих сторінках, включають у загальну нумерацію записки.

### **5.4 Викладення тексту пояснювальної записки**

Текст записки має бути чітким і не допускати різних тлумачень.

Пояснювальна записка має бути написана логічно послідовно, грамотною технічною мовою. Варто звернути увагу на вибір потрібних формулювань, які б якомога простіше, чітко, стисло і доступно виражали б зміст питання, що викладається.

Викладаючи обов'язкові вимоги у записці, потрібно використовувати слова «повинен», «потрібно», «вимагається щоб», «дозволяється тільки», «не допускається». Використовуючи інші положення потрібно вживати слова «можна», «як правило», «за потреби» і т. ін.

Назви, які наводяться в тексті записки і на ілюстраціях, повинні бути однаковими.

У записці належить застосовувати науково-технічні терміни, позначення і визначення, установлені чинними стандартами, а за їх відсутності – загальноприйняті у науково-технічній літературі.

Усі скорочені найменування потрібно пояснювати в тексті з їх першою появою, наприклад, «Паралельна комп'ютерна система (ПКС)...», далі можна користуватися скороченням «ПКС».

Якщо у документі прийнято специфічну термінологію, то в кінці його перед списком літератури наводять перелік використаних термінів з відповідними поясненнями.

У тексті записки не допускається:

- скорочувати позначення одиниць фізичних величин, якщо їх використовують без цифр, за винятком фізичних величин в головках і боковиках таблиць і в розшифровках буквених позначень, які входять у формули;
- використовувати без числових значень математичні знаки, наприклад,  $\leq$  (менше або дорівнює),  $\geq$  (більше або дорівнює),  $\neq$  (не дорівнює), а також № (номер), % (відсоток);
- використовувати знак «о» для позначення діаметра (треба писати слово «діаметр»);
- застосовувати індекси стандартів, технічних умов і інших документів (ГОСТ, ДСТУ, ТУ тощо) без реєстраційного номера.

Числові значення величин в тексті вказують з необхідним ступенем точності, при цьому в ряду величин вирівнювати число знаків після кожної коми не обов'язково.

### **5.5. Оформлення ілюстрацій**

Усі ілюстрації в пояснювальній записці (креслення, схеми, фотографії, діаграми, графіки) називають рисунками.

Кількість ілюстрацій має бути достатньою для пояснення тексту, який викладається. Ілюстрації потрібно розміщувати як по тексту записки (якомога ближче до відповідних частин тексту), так і в кінці його або наводити в додатках. Ілюстрації належить виконувати у відповідності до вимог стандартів ЄСКД і ЕСПД за допомогою різних графічних редакторів та систем автоматизованого проектування.

Усі ілюстрації послідовно нумерують у межах розділу арабськими цифрами. Номер ілюстрації складається з номера розділу і порядкового номера ілюстрації, наприклад, «Рисунок 2.5 — Граф алгоритму». Посилання на ілюстрації подають так: «... на рис. 2.5 ...». Повторне посилання на ілюстрацію наводять із скороченням слова «дивись», наприклад, «... див. рис.2.5 ...». Допускається нумерація ілюстрацій у межах усієї записки.

Розмір шрифту всіх без винятку надписів у рисунках має бути таким самим, як і в тексті пояснювальної записки.

Ілюстрації розміщують так, щоб їх можна було розглядати, не повертаючи або повертаючи за ходом стрілки годинника.

## **5.6. Побудова таблиць**

Цифровий матеріал, як правило, оформляють у вигляді таблиць.

Горизонтальні лінії, які розмежовують рядки таблиці, можна не використовувати, якщо це не затрудняє користування таблицею.

Таблицю розміщують безпосередньо після тексту, в якому вона згадується вперше або на наступній сторінці. На всі таблиці мають бути посилання в тексті записки.

Таблиці послідовно нумерують арабськими цифрами в межах розділу, за винятком таблиць у додатках. Номер таблиці складається з номера розділу і порядкового номера таблиці, розділених крапкою, наприклад, «таблиця 2.1» – перша таблиця другого розділу.

Якщо у записці тільки одна таблиця, то її також нумерують за наведеними правилами.

Таблиця може мати назву, яку друкують малими літерами, окрім першої великої, і розміщують над таблицею. Назва повинна бути короткою і відображати зміст таблиці.

Заголовки граф і рядків таблиці пишуть з великої букви, а підзаголовки – з малої букви, якщо вони складають одне речення із заголовком, або з великої букви, якщо вони мають самостійне значення. Заголовки і підзаголовки вказують в однині, в кінці їх крапку не ставлять. Розділяти заголовки і підзаголовки боковика і граф діагональними лініями не допускається. Заголовки граф записують паралельно рядкам таблиці, але за потреби допускається їх вертикальне розміщення. Головка відділяється лінією від решти таблиці. Висота рядків повинна бути не менше 8 мм. Допускається розміщувати таблицю уздовж довгої сторони аркуша.

## 6. ЗАХИСТ КУРСОВОГО ПРОЄКТУ

Виконаний проєкт підписує здобувач і після дозволу керівника він допускається до захисту. Якщо керівник проєкту не допускає здобувача до захисту, то це питання обговорюється на засіданні кафедри у його присутності.

Захист курсового проєкту – це форма перевірки якості його виконання. Захист відбувається перед комісією у складі двох-трьох викладачів за участю керівника та здобувачів групи.

Під час захисту здобувач робить доповідь (6–8 хв) по суті проєкту та відповідає на запитання. Для демонстрації графічного матеріалу можуть застосовуватись мультимедійні засоби ПЕОМ та використовуватись різні презентаційні програми, наприклад Microsoft Power Point. При цьому здобувач може використовувати додаткові ілюстративні матеріали, технічні розробки, які відображають суть проєкту.

Якість проєкту та його захист оцінюється в балах (0-100), за шкалою ECTS (A, B, C, D, E, FX, F) та за національною шкалою «відмінно», «добре», «задовільно», «незадовільно». Здобувач, який не подав до захисту курсовий проєкт у встановлений графіком термін або не захистив його з позитивною оцінкою, вважається таким, що має академічну заборгованість, яку він повинен ліквідувати в установленому порядку.

Захищений курсовий проєкт (робота) зберігається на кафедрі протягом трьох років.

## 7. СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

### Основна література

1. Основи комп'ютерної техніки та програмування мікропроцесорів: навч. посіб. / Д.О. Гололобов. – К. : Редакційно-видавничий центр Державного університету телекомунікацій, 2019. – 58с.

2. Мікропроцесорна техніка: Лабораторний практикум [Електронний ресурс]: навч. посіб. для студ. спеціальності 171 «Електроніка» / КПІ ім. Ігоря Сікорського; уклад.: Т. О. Терещенко, Л. М. Батрак, Ю. С. Ямненко. – Електронні текстові дані (1 файл: 1,51 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2021. – 76 с.

3. Мельник А. О. Архітектура комп'ютера: підручник для здобувачів вузів. 3-вид., – Луцьк : Волинська обласна друкарня, 2018. – 470 с.

4. Тонкошкур О.С., Гниленко О.Б., Матвєєва Н.О., Морозов О.С. Архітектура комп'ютерів. Машинні команди та програмування на асемблері. Навчальний посібник, – Дніпро: «Нова Ідеологія», 2018. – 179 с.

5. Advanced Micro Devices, Inc. AMD64 Architecture Programmer's Manual Volume 1: Application Programming. Publication No. 24592. Revision Date 3.22. December 2017. – Режим доступу:

[https://developer.amd.com/wordpress/media/2012/10/24592\\_APM\\_v11.pdf](https://developer.amd.com/wordpress/media/2012/10/24592_APM_v11.pdf)

6. ДСТУ 8302:2015. Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання. Чинний від 2016-07-01. Вид. офіц. Київ : УкрНДНЦ, 2016. 16 с

**Додаток А**  
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА  
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА ТЕХНОЛОГІЙ

**Курсовий проєкт**

з дисципліни «Архітектура комп'ютерів та низькорівневе програмування»

Виконав здобувач денної (заочної) форми навчання  
спеціальності 123 «Комп'ютерна інженерія»

Прізвище Ім'я \_\_\_\_\_

Керівник: д.т.н., проф.

Прізвище Ім'я \_\_\_\_\_

Члени комісії : Прізвище Ім'я \_\_\_\_\_

Прізвище Ім'я \_\_\_\_\_

Прізвище Ім'я \_\_\_\_\_

Одеса – 2024

## Додаток Б

### ФОРМАТИ КОМАНД

Типова команда, в загальному випадку, повинна зазначати:

- операцію, що підлягає виконанню;
- адреси вихідних даних (операндів), над якими виконується операція;
- адресу, за якою має бути поміщений результат операції.

Відповідно до цього команда складається з двох частин: операційної та адресної.

Операційна частина	Адресна частина
--------------------	-----------------

Формат команди визначає її структуру, тобто кількість двійкових розрядів, що відводяться під всю команду, а також кількість та розташування окремих полів команди. Полем називається сукупність двійкових розрядів, що кодують складову частину команди.

У межах системи команд однієї ЄОМ можна використовувати різні формати команд. Зазвичай це пов'язано із застосуванням різних способів адресації. У такому випадку до складу коду команди вводиться поле для завдання способу адресації (СА).

КОП	СА	Адресна частина
-----	----	-----------------

В адресній частині команди міститься інформація про місцезнаходження вихідних даних і місце збереження результату операції. Зазвичай місцезнаходження кожного з операндів та результату задається в команді шляхом вказівки адреси відповідної комірки основної пам'яті або номера регістру процесора. Принципи використання інформації з адресної частини команди визначає система адресації. Система адресації задає число адрес в команді і прийняті способи адресації.

Для визначення кількості адрес, що включаються до адресної частини, будемо використовувати термін адресність. У «максимальному» варіанті необхідно вказати три компоненти: адреса першого операнда, адреса другого операнда і адреса осередку, куди заноситься результат операції (триадресний формат команди).

Операція	Адреса		
КОП	1-й операнд	2-й операнд	3-й операнд

У триадресному форматі довжина команди може бути дуже великою. Так, якщо адреса осередку основної пам'яті має довжину 32 біти, а довжина коду операції - 8біт, то довжина команди складе 104 біта (13 байт).

Якщо за умовчанням взяти як адресу результату адресу одного з операндів (зазвичай першого), то можна обійтися без третьої адреси, в результаті отримуємо двоадресний формат команди.

Операція	Адреса	
КОП	1-й операнд	2-й операнд

У цьому випадку відповідний операнд після виконання операції втрачається.

Команду можна ще більше скоротити, перейшовши до одноадресного формату, що можливо при виділенні певного стандартного місця для зберігання першого операнда та результату. Зазвичай для цієї мети використовується спеціальний регістр центрального процесора (ЦП), відомий під назвою акумулятора, оскільки тут акумулюється результат.

Операція	Адреса
КОП	1-й операнд або 2-й операнд

Застосування єдиного регістру для зберігання одного з операндів і результату є обмежуючим фактором, тому крім акумулятора часто використовують інші регістри ЦП. Так як число регістрів у ЦП невелике, для вказівки одного з них у команді достатньо мати порівняно коротке адресне поле. Відповідний формат носить назву півтораадресного або регістрового формату.

Операція	Адреса	
КОП	Регистр	2-й операнд

Нарешті, якщо для обох операндів вказати чітко задане місце-положення, а також у разі команд, які не вимагають операнда, можна отримати нуль-адресний формат команди.

Операція
КОП

## ДОДАТОК С. ТИПИ АДРЕСАЦІЇ

Для оцінки ефективності різних способів адресації використовуються витрати обладнання (С) і витрати часу (Т) на доступ до адресованих даних.

### Безпосередня адресація

При безпосередній адресації адресному полі команди замість адреси міститься безпосередньо сам операнд. Цей спосіб може застосовуватися при виконанні арифметичних операцій, операцій порівняння, а також завантаження констант в регістри.

КОП	СА	Безпосередній операнд
-----	----	-----------------------

При записі в регістр, що має розрядність, що перевищує довжину безпосереднього операнда, операнд розміщується в молодшій частині регістра, а позиції, що залишилися, заповнюються значенням знакового біта операнда.

Недоліком даного способу адресації є те, що розмір безпосереднього операнда обмежений довжиною адресного поля команди, яке в більшості випадків менше довжини машинного слова.

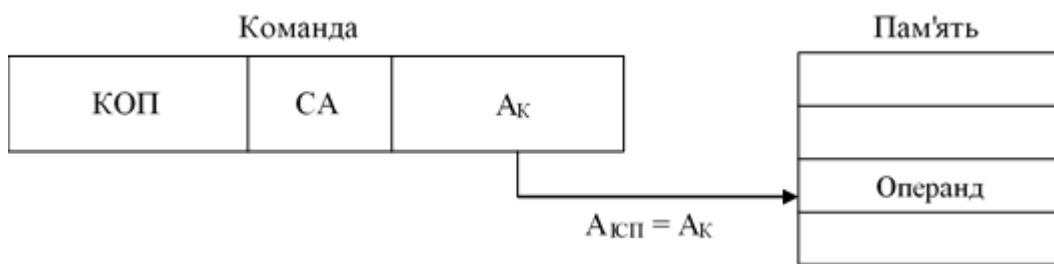
Найбільш інтенсивно даний вид адресації використовується в арифметичних операціях і командах порівняння.

Безпосередня адресація скорочує час виконання команди, оскільки не потрібно звернення до пам'яті за операндом. Крім того, економиться пам'ять, оскільки відпадає потреба в осередку для зберігання операнда. У плані ефективності цей спосіб можна вважати «ідеальним» (СНА = 0, ТНА = 0), і його можна рекомендувати до використання у всіх ситуаціях, коли тому не перешкоджають вищезгадані обмеження.

## ПРЯМА АДРЕСАЦІЯ

При прямій або абсолютній адресації (ПА) адресний код прямо вказує номер осередку пам'яті, до якої здійснюється звернення, тобто адресний код збігається з виконавчою адресою.

Спосіб має важливий недолік - обмежений розмір адресного простору, так як для адресації до пам'яті великої ємності потрібно "довге" адресне поле. Ще більш істотною недосконалістю можна вважати те, що адреса, вказана в команді, не може бути змінена в процесі обчислень (принаймні, така зміна не рекомендується). Це обмежує можливість довільного розміщення програми в пам'яті.

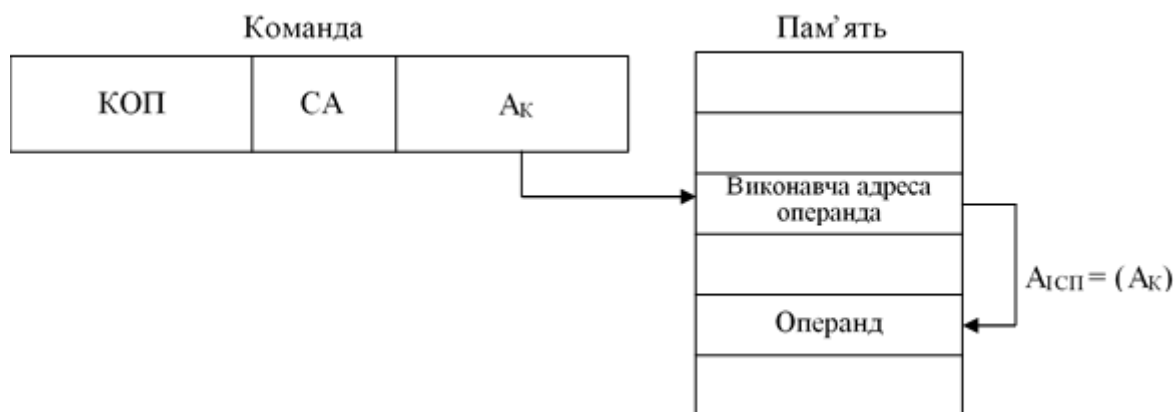


Прямую адресацію характеризують такі показники ефективності:

$C_{ПА} = \text{int}(\log_2 N_i)$ ,  $T_{ПА} = t_{зп}$ , де  $N_i$  - кількість адресованих операндів,  $t_{зп}$  - час вибірки або запису операнда.

Косвена (непряма) адресація

Непряма адресація (КА) – прийом, коли за допомогою обмеженого адресного поля команди вказується адреса комірки, що у свою чергу містить повнорозрядну адресу операнда. Запис (АК) означає вміст комірки, адреса якої вказаний в дужках.



При непрямій адресації вміст адресного поля команди залишається незмінним, тоді як непрямий адресу процесі виконання програми можна змінювати. Це дозволяє проводити обчислення, коли адреси операндів заздалегідь невідомі і з'являються лише в процесі вирішення задачі. Даний прийом спрощує обробку масивів та списків, а також передачу параметрів підпрограм.

Недоліком є необхідність у дворазовому зверненні до пам'яті: спочатку для отримання адреси операнда, а потім для звернення до операнда ( $T_{КА} = 2t_{зп}$ ). Крім того задіяний зайвий осередок пам'яті для зберігання виконавчої адреси операнда.

Способу властиві такі витрати устаткування:

$$C_{КА} = R_{яч} + \text{int}(\log_2 N_A) \geq \text{int}(\log_2(N_i + N_A))$$

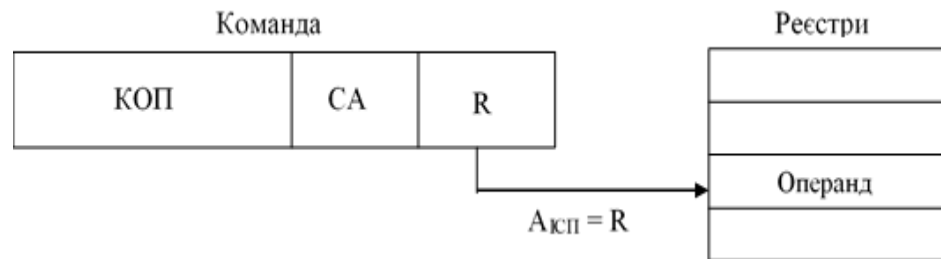
де  $R_{яч}$  - розрядність осередку пам'яті, що зберігає виконавчу адресу,  $N_A$  - кількість осередків для зберігання виконавчих адрес,  $N_i$  - кількість адресованих

операндів. Тут вираз  $int(\log_2 N_A)$  визначає розрядність скороченого адресного поля команди (зазвичай  $N_A \ll N_i$ ).

## РЕЄСТРОВА АДРЕСАЦІЯ

Реєстрова адресація (РА) нагадує пряму адресацію. Різниця полягає в тому, що адресне поле інструкції вказує не на комірку пам'яті, а на реєстр загального призначення процесора (РОН), що позначається буквою R.

Двома основними перевагами реєстрової адресації є: коротке адресне поле в команді та виключення звернень до пам'яті. Мале число РОН у складі процесора (зазвичай 8 або 16) дозволяє скоротити довжину адресного поля команди, тобто  $C_{РА} \ll C_{ПА}$ . Крім того,  $T_{РА} = t_{РОН}$ , де  $t_{РОН}$  - час вибірки операнда з реєстра загального призначення, причому  $t_{РОН} \ll t_{ЗУ}$ .



### Непряма реєстрова адресація

*Непряма реєстрова адресація* (НРА) являє собою непряму адресацію, де виконавча адреса операнда зберігається не в осередку основної пам'яті, а в реєстрі процесора. Відповідно, адресне поле команди вказує не на комірку пам'яті, а на реєстр.



Переваги та обмеження непрямої реєстрової адресації ті ж, що і у звичайної непрямої адресації, але завдяки тому, що непряма адреса зберігається не в пам'яті, а в реєстрі, для доступу до операнда потрібно на одне звернення до пам'яті менше.

Ефективність непрямої реєстрової адресації можна оцінити за формулами:

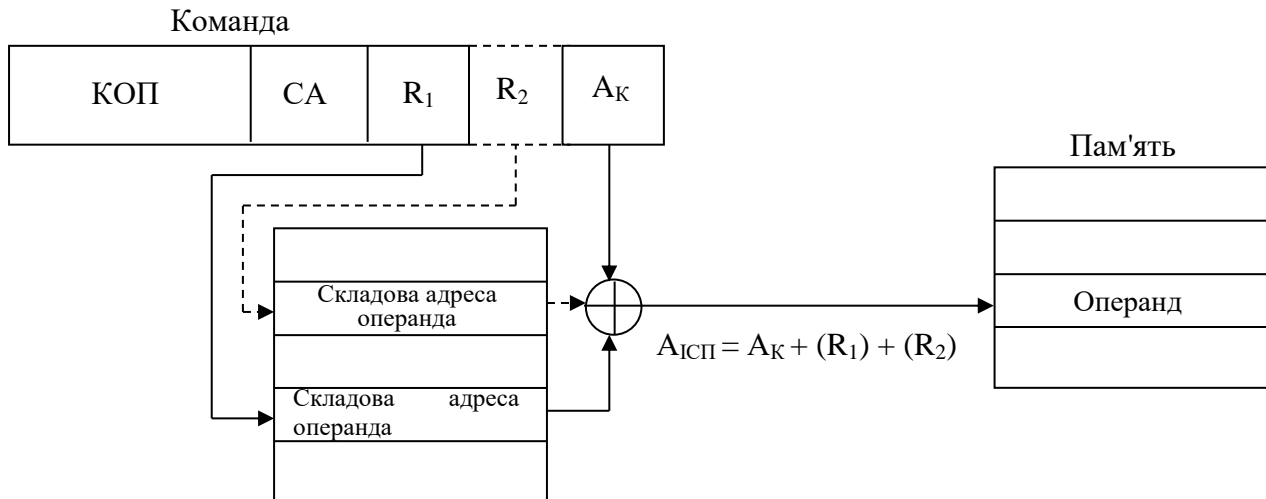
$$T_{КРА} = t_{РОН} + t_{ЗУ}$$

$$C_{КРА} = R_{РОН} + int(\log_2 N_A) \geq int(\log_2 (N_i + N_A))$$

де  $R_{РОН}$  – розрядність реєстру загального призначення.

## АДРЕСАЦІЯ ЗІ ЗМІЩЕННЯМ

При адресації зі зміщенням виконавчий адресу формується в результаті підсумовування вмісту адресного поля вмістом одного або декількох регістрів процесора.



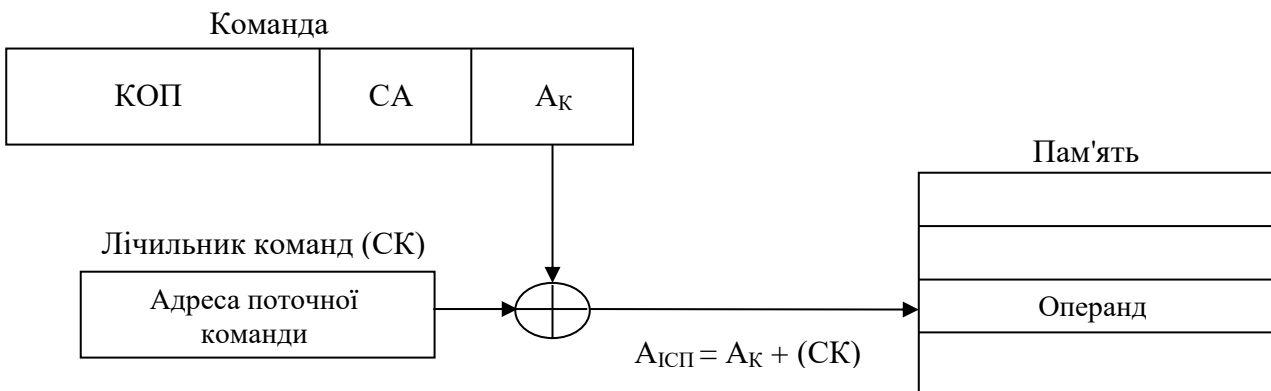
Адресація зі зміщенням передбачає, що адресна частина команди включає як мінімум одне поле ( $A_K$ ). У ньому міститься константа, сенс якої в різних варіантах адресації зі зміщенням може змінюватися. Константа може представляти собою якийсь базовий адресу, до якого додається зміщення, що зберігається в регістрі. Допустимо і прямо протилежний підхід: базова адреса знаходиться в регістрі процесора, а в полі  $A_K$  вказується зміщення щодо цієї адреси. У деяких процесорах для реалізації певних варіантів адресації зі зміщенням передбачені спеціальні регістри, наприклад базовий або індексний. У деяких процесорах для реалізації певних варіантів адресації зі зміщенням передбачені спеціальні реєстри, наприклад базовий або індексний. Використання таких регістрів передбачається за умовчанням, тому адресна частина команди містить лише поле  $A_K$ . Якщо ж складова адреси може розташовуватися в довільному регістрі загального призначення, то для вказівки конкретного регістру в команду включається додаткове поле R (при складанні адреси більш ніж з двох складових у команді буде кілька таких полів). Ще одне поле R може з'явитися в командах, де зміщення перед обчисленням виконавчої адреси множиться на мас-штабний коефіцієнт. Такий коефіцієнт заноситься в один із регістрів загального призначення, на який і вказує це додаткове поле. У найбільш загальному випадку адресація зі зміщенням має на увазі наявність двох адресних полів:  $A_K$  і R.

У рамках адресації зі зміщенням є ще один варіант, при якому виконавча адреса обчислюється не підсумовуванням, а кон-тенацією складових адреси. Тут одна складова є старшою частиною виконавчої адреси, а друга — молодшою.

Нижче розглядаються основні способи адресації зі зміщенням, кожен з яких, втім, має власну назву.

## ВІДНОСНА АДРЕСАЦІЯ

При відносній адресації (ВА) для отримання виконавчої адреси операнда вміст поля АК складається з вмістом лічильника команд. Таким чином, адресний код в команді являє собою зсув щодо адреси команди, що текує. Слід зазначити, що в момент обчислення виконавчої адреси операнда в лічильнику команд може вже бути сформований адресу наступної команди, що потрібно враховувати при виборі величини усунення. Зазвичай поле  $A_K$  трактується як двійкове число додаткового коду.



Адресація щодо лічильника команд базується на властивості локальності, що виражається в тому, що більшість звернень відбувається до осередків, розташованих у безпосередній близькості від виконуваної команди. Це дозволяє заощадити на довжині адресної частини команди, оскільки розрядність поля АК може бути невеликою. Головне достоїнство даного способу адресації полягає в тому, що він робить програму переміщується в пам'яті: незалежно від поточного розташування програми в адресному просторі взаємне положення команди і операнда залишається незмінним, тому адресація операнда залишається коректною.

Ефективність даного способу адресації (позначимо його як СА – «щодо Лічильника Адресація») можна описати виразами:

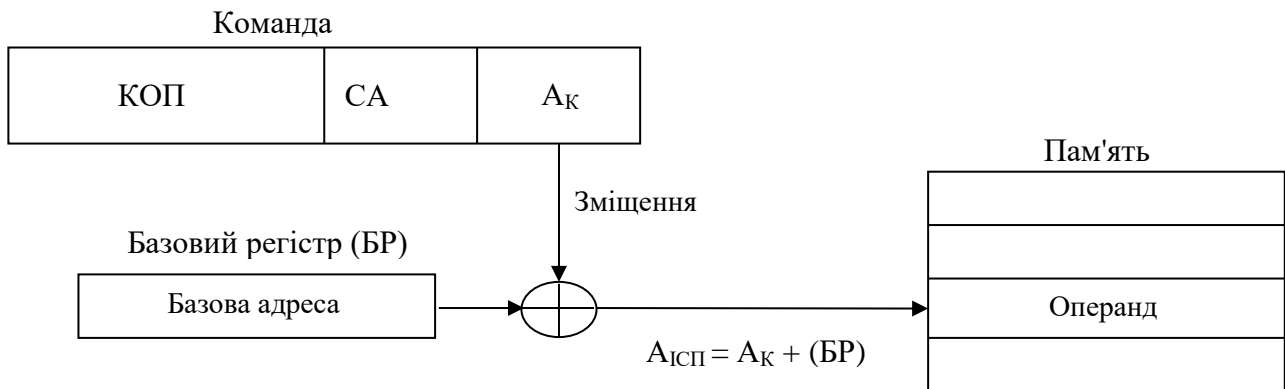
$$T_{СА} = t_{РОН} + t_{СЛ} + t_{ЗУ}$$

$$C_{СА} = R_{РОН} + \text{int}(\log_2 N_i - R_{СК})$$

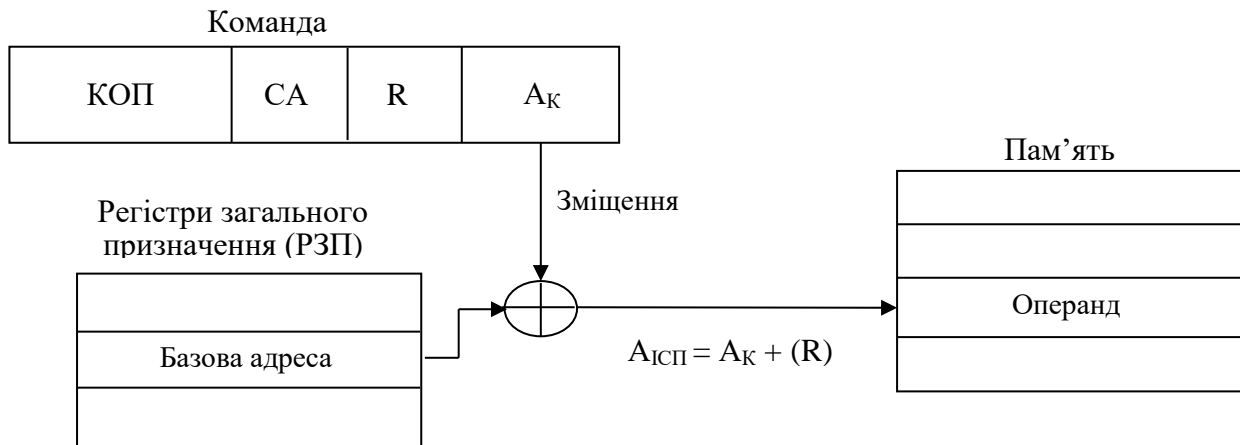
де  $t_{СЛ}$  - час складання складових виконавчої адреси,  $R_{СК}$  - розрядність лічильника команд.

## БАЗОВА РЕЄСТРОВА АДРЕСАЦІЯ

У разі базової регістрової адресації (БРА) регістр, званий базовою, містить повнорозрядну адресу, а поле АК – зсув щодо цієї адреси. Посилання на базовий регістр може бути явним або неявним. У деяких обчислювальних машинах є спеціальний базовий регістр і його використання є неявним, тобто підпол R в команді відсутня.



Найбільш типовий випадок, коли у ролі базового регістру виступає один з РОН, тоді його номер явно вказується в полі R команди.



Базову регістрову адресацію зазвичай використовують для доступу до елементів масиву, положення якого пам'яті в процесі обчислень може змінюватися. У базовий регістр заноситься початкова адреса масиву, а адреса елемента масиву вказується в полі АК команди у вигляді зміщення щодо початкової адреси масиву. Перевага цього способу в тому, що зсув має меншу довжину, ніж повна адреса, і це дозволяє скоротити довжину адресного поля команди. Коротке зміщення розширюється до повної довжини виконавчої адреси шляхом додавання зліва бітів, що збігаються зі значенням знакового розряду зміщення.

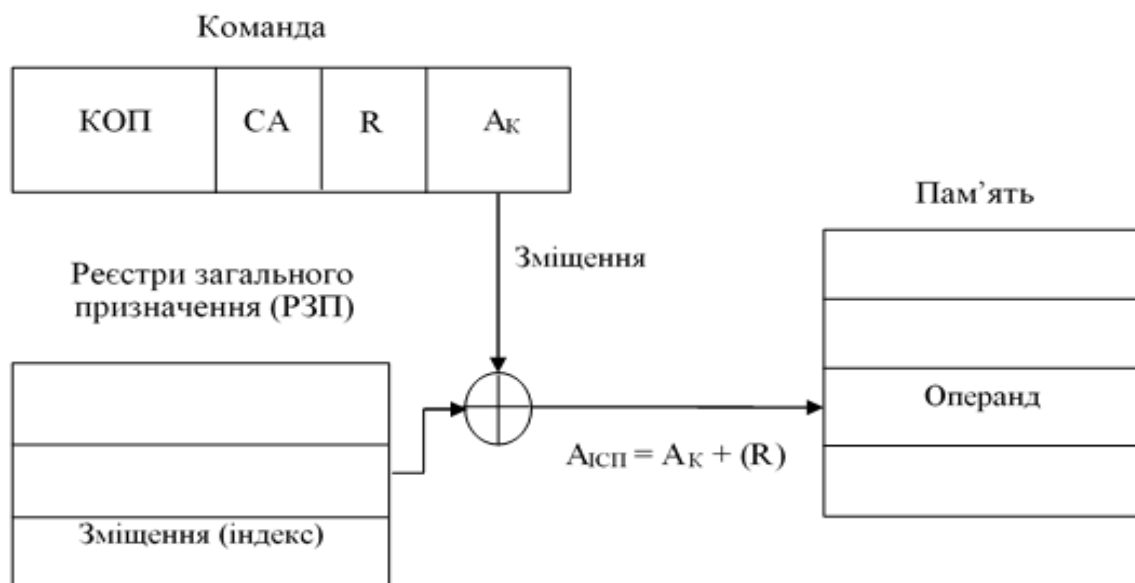
Розрядність зміщення  $R_{CM}$  і, відповідно, витрати обладнання визначаються з умови  $R_{CM} = C_{БРА} = \text{int}(\log_2(\max(N_{ОПi})))$ , де  $N_{ОПi}$  - кількість операндів  $i$ -ї програми.

Витрати часу становлять  $T_{БРА} = t_{РОН} + t_{СЛ} + t_{ЗВ}$

## ІНДЕКСНА АДРЕСАЦІЯ

При індексній адресації (ІА) поле  $A_K$  містить адресу осередку пам'яті, а регістр (зазначений явно чи неявно) - зміщення щодо цієї адреси. Як видно, цей спосіб адресації схожий на базову регістрову адресацію. Оскільки при індексній адресації в полі АК знаходиться повнорозрядна адреса осередку пам'яті, що грає роль бази, довжина цього поля більша, ніж при базовій регістровій адресації. Тим не менш, обчислення виконавчої адреси операнда проводиться ідентично.

Індексна адресація з використанням одного з РОН



Індексна адресація надає зручний механізм організації ітеративних обчислень. Нехай, наприклад, є масив чисел, розташованих у пам'яті послідовно, починаючи з адреси  $N$ , і хочемо збільшити на одиницю все елементи даного масиву. Для цього потрібно витягти кожне число з пам'яті, додати до нього 1 і повернути назад, а послідовність виконавчих адрес буде наступною:  $N, N+1, N+2$  і т. д., аж до останнього осередку, займаного масивом. Значення  $N$  береться з підполя команди, а обраний регістр, званий індексним регістром, спочатку заноситься 0. Після кожної операції вміст індексного регістру збільшується на одиницю.

Так як це досить типовий випадок, у більшості збільшення або зменшення вмісту індексного регістру до або після звернення до нього здійснюється автоматично як частина машинного циклу. Такий прийом називається автоіндексуванням. Якщо для індексної адресації використовуються спеціально

виділені регістри, автоіндексування може відбуватися неявно і автоматично. При залученні для зберігання індексів регістрів загального призначення необхідність операції автоіндексування повинна вказуватися в команді спеціальним бітом.

Автоіндексування зі збільшенням вмісту індексного регістру носить назву автоінкрементної адресації і може бути описано наступним чином:

$$A_{ICP}=A_k+(R), R\leftarrow(R)+1 \text{ або } R\leftarrow(R)+1, A_{ICP}=A_k+(R)$$

де  $(R)$  - вміст індексного регістру з адресою  $R$ .

У першому варіанті збільшення вмісту індексного регістру відбувається після формування виконавчої адреси, і цей спосіб називається постінкрементним автоіндексуванням. У другому випадку спочатку проводиться збільшення вмісту індексного регістру, і вже нове значення використовується для формування виконавчої адреси. Тоді говорять про преінкрементне автоіндексування.

Автоіндексування зі зменшенням вмісту індексного регістру носить назву автодекрементної адресації і може бути описано так:

$$A_{ICP}=A_k+(R), R\leftarrow(R)-1 \text{ або } R\leftarrow(R)-1, A_{ICP}=A_k+(R)$$

Тут також можливі два варіанти, що відрізняються послідовністю виконання операцій зменшення вмісту індексного регістру та обчислення виконавчої адреси: постдекрементне автоіндексування та передекрементне автоіндексування.

Цікавим і дуже корисним є ще один варіант індексної адресації - індексна адресація з масштабуванням і усуненням: вміст індексного регістру множиться на масштабний коефіцієнт і підсумовується з  $A_k$ . Масштабний коефіцієнт може набувати значення 1, 2, 4 або 8, для чого в частині команди виділяється додаткове поле.

Слід особливо відзначити, що система команд багатьох обчислювальних машин надає можливість різним чином поєднувати базову і індексну адресації як додаткові способи адресації.

### **Способи адресації у командах управління потоком команд**

Основними способами адресації у командах управління потоком команд є пряма та відносна.

Для команд безумовного і умовного переходу (розгалуження) найбільш типова відносна адресація, коли в адресній частині команди вказується зміщення адреси точки переходу щодо поточної команди, тобто зміщення щодо поточного вмісту лічильника команд. Використання цього способу адресації дозволяє програмі виконуватися в будь-якому місці пам'яті - програми стають переміщуваними. Серед команд безперечного переходу частка відносної адресації становить близько 90%.

ДЛЯ НОТАТОК

*Навчальне видання*

# **АРХІТЕКТУРА КОМП'ЮТЕРІВ ТА НИЗЬКОРІВНЕВЕ ПРОГРАМУВАННЯ**

**Методичні вказівки**

до виконання курсового проєкту для здобувачів  
першого (бакалаврського) рівня вищої освіти  
спеціальності 123 Комп'ютерна інженерія

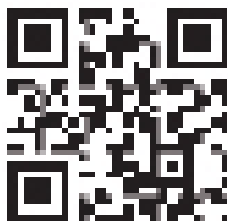
***Укладачі:***

**Гунченко Юрій Олександрович**

**Шаріпова Ільнара Вільївна**

**Берков Юрій Миколайович**

*В авторській редакції*



Підписано до друку 22.03.2024 р. Формат 60x84/16.  
Папір офсетний. Гарнітура Times. Цифровий друк.  
Ум. друк. арк. 2,32. Наклад 30 пр. Замовлення № 0524-22.  
Віддруковано з готового оригінал-макету у друкарні ТакиБук.

Видавництво: Олді+  
65101, Україна, м. Одеса, вул. Інглезі, 6/1  
Свідоцтво ДК № 7642 від 29.07.2022 р.

Тел.: +38 (098) 559-45-45,  
+38 (095) 559-45-45, +38 (093) 559-45-45  
Для листування: 65101, Україна, м. Одеса, вул. Інглезі, 6/1  
E-mail: office@oldiplus.ua

**ОЛДІ  
ПЛЮС**