

МІНІСТЕРСТВО ОСВІТИ І НАУКИ
Національний авіаційний університет

КОМП'ЮТЕРНА СХЕМОТЕХНІКА ТА АРХІТЕКТУРА КОМП'ЮТЕРІВ

Лабораторний практикум
для студентів напрямку підготовки 6.050101
«Комп'ютерні науки»

Київ 2013

УДК 004.27(076.5)
ББК 3973.20я7
К 637

Укладачі : *В.М. Єфимець, І.А. Жуков, Є.В. Красовська,
С.О. Кудренко, В. В. Лукашенко*

Рецензент *д.т.н., проф. Зіатдінов Ю.К.*

*Затверджено методично-редакційною радою Національного
авіаційного університету (протокол № від р.).*

Комп'ютерна схемотехніка та архітектура комп'ютерів :
К 637 лабораторний практикум / Уклад.: В. М. Єфимець, І.А. Жуков,
Є. В. Красовська, С. О. Кудренко, В. В. Лукашенко. – К. : НАУ, 2013. –
64 с.

Лабораторний практикум містить роботи для набуття студентами
практичних навичок по схемотехніці та архітектурі комп'ютерів.
Для студентів напряму підготовки 6.050101 «Комп'ютерні науки».

ЗМІСТ

ВСТУП.....	4
Тема 1. СХЕМОТЕХНІКА КОМБІНАЦІЙНИХ ПРИСТРОЇВ.....	5
Лабораторна робота. Комбінаційні цифрові вузли.....	5
Тема 2. СХЕМОТЕХНІКА ЦИФРОВИХ ВУЗЛІВ.....	13
Лабораторна робота. Цифрові пристрої з пам'яттю.....	13
Тема 3. ІНТЕГРОВАНІ СИСТЕМИ ЕЛЕМЕНТІВ.....	26
Лабораторна робота. Керуючі автомати.....	26
Тема 4. ЦИФРОВІ КОМП'ЮТЕРИ.....	32
Лабораторна робота. Пристрій керування з програмованою логікою.....	32
Тема 5. МІКРОПРОЦЕСОРНІ СИСТЕМИ.....	41
Лабораторна робота. Блок обробки даних.....	41
Тема 6. ПАРАЛЕЛЬНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ.....	54
Лабораторна робота. Кластерна обчислювальна система.....	54
СПИСОК ЛІТЕРАТУРИ.....	69

ВСТУП

Навчальна дисципліна «Комп'ютерна схемотехніка та архітектура комп'ютерів» є складовою напрямку підготовки «Комп'ютерні науки» і повинна забезпечити теоретичну та практичну підготовку фахівців по схемотехніці, архітектурі комп'ютерів, комп'ютерним системам. Особливе значення має практичний напрямок підготовки, який забезпечує лабораторний практикум. В навчальній програмі виділяються теми, що визначають необхідний рівень знань та вмінь фахівців цього напрямку:

- схемотехніка комбінаційних пристроїв (вузлів). Досліджуються організація та використання логічних елементів, дешифраторів/шифраторів, мультиплексорів/демультиплексорів, суматорів;

- схемотехніка цифрових вузлів. Досліджується організація та використання цифрових пристроїв з пам'яттю – тригерів, регістрів, лічильників;

- інтегровані системи елементів. В ЛР досліджується синтез керуючих автоматів (КА) по графу мікропрограми;

- програмована логіка. Досліджується організація та проектування КА з програмованою логікою – кодування мікропрограм (МК), природня адресація МК, розробка закодованої мікропрограми (МП), структурна схема пристрою;

- мікропроцесорні пристрої. В ЛР на прикладі МПК К1804 досліджується розробка блоку обробки даних, програмування, формування закодованої МП;

- обчислювальні системи (ОС). Досліджуються принципи побудови ОС, проектується кластерна система з мінімальними зв'язками між вузлами.

В ЛР використовуються програмні додатки, які дозволяють оптимізувати виконання окремих елементів роботи.

ТЕМА 1. СХЕМОТЕХНІКА КОМБІНАЦІЙНИХ ПРИСТРОЇВ

Лабораторна робота

КОМБІНАЦІЙНІ ЦИФРОВІ ВУЗЛИ

Мета роботи. Дослідження та закріплення теоретичних знань по організації і використанню комбінаційних пристроїв: логічних елементів; дешифраторів/шифраторів; мультиплексорів/демультиплексорів; суматорів.

Основні завдання роботи

1. Отримати варіант домашнього завдання.
2. Вивчити теоретичну частину роботи.
3. Виконати аналіз та синтез комбінаційних пристроїв.

Основні теоретичні відомості

Комбінаційні цифрові схеми входять до складу цифрових пристроїв і визначаються як з'єднання логічних елементів (ЛЕ), в якому відсутні обернені зв'язки.

1. Булева алгебра

Аналіз та синтез комбінаційних вузлів (схем) базується на використанні співвідношень та законів булевої алгебри (БА) [1, 2]. Змінні в цій алгебрі приймають тільки два значення: стан логічної «1» (істина) і стан логічного «0» (не істина). БА використовує три операції: заперечення (операція «НІ»); логічні множення (операція «І») та додавання (операція «АБО»). Логічні операції «І» та «АБО» називають кон'юнкцією та диз'юнкцією відповідно.

Основні співвідношення та закони БА:

- подвійне заперечення $\overline{\overline{X}} = X$;
- правила де Моргана , $\overline{X \cdot Y} = \overline{X} + \overline{Y}$;

$x(xv)$

- правила поглинання ;
- співвідношення для констант , , $1 \cdot x = x$, $1vx = 1$, $0vx = x$;
- закон виключення третього $xv\bar{x} = 1$;
- закон протиріччя $x \cdot \bar{x} = 0$.

Для представлення булевих функцій (БФ) використовують, наприклад, диз'юнктивні нормальні форми (ДНФ) – диз'юнкції елементарних кон'юнкцій –

$$f(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \oplus x_1 x_2 x_3 .$$

Мінімізація БФ приводить до спрощення відповідної комбінаційної схеми (КС). Одним із способів спрощення БФ є

Карта Карно (КК). Це таблиця із клітин, в якій сусідні рядки та стовпці кодуються сусідніми кодами. На рис. 1 показані КК 3-х та 4-х змінних. Кожній клітині ставиться у відповідність двійковий n -мірний набір, на якому БФ приймає значення «1» («0»).

2. Логічні елементи

Реалізують прості БФ, наприклад, «І», «АБО», «НІ» (рис. 2, а). Сукупність логічних елементів (ЛЕ), що дозволяють реалізувати довільну БФ називають логічним базисом, наприклад, ЛЕ «І», «НІ», «АБО». Базис може складатись також із одного типу ЛЕ «І-НІ» або «АБО-НІ» (рис. 2, б). Використовують також інші ЛЕ, що реалізують прості БФ – сумування по $mod 2$ або «Виключення АБО» (рис. 2, в).

3. Комбінаційні схеми

Комбінаційні схеми (КС) це логічні схеми, що реалізують функції, значення яких визначаються вхідними змінними у даний момент часу.

3.1. Двійкові дешифратори та шифратори

Дешифратор (декодер) це n – перетворювач двійкового

n -розрядного коду в унітарний n -розрядний код (рис. 3). Функціонування дешифратора описується системою рівнянь:

$$\begin{aligned} F_0 &= \overline{x_0} \overline{x_1} \overline{x_2} \cdot \overline{E_N}, \\ F_1 &= x_0 \overline{x_1} \overline{x_2} \cdot \overline{E_N}, \\ &\dots\dots\dots \\ F_{n-1} &= x_0 x_1 x_2 \cdot \overline{E_N}. \end{aligned}$$

Шифратор. Виконує операцію обернену по відношенню до n

операції дешифратора. Повний шифратор має n входів та n виходів. При збудженні одного із входів шифратора на його виході формується двійковий код номера збудженого входу.

3.2. Мультиплексори та демультимплексори

Мультиплексор МХ служить для підключення вхідного каналу до вихідного під керуванням адресного слова. Входи МХ діляться на дві групи: інформаційні X та адресні A (рис. 4). Робота МХ описується формулою

$$F = x_0 \overline{a_{n-1}} \dots \overline{a_0} \vee x_1 \overline{a_{n-1}} \dots a_0 \vee \dots \vee \overline{x_n} a_{n-1} \dots a_0,$$

де x_i – інформаційний вхід, $a_{n-1} \dots a_0$ – адреса інформаційного входу x_i .

Демультимплексор DMX. Виконує операцію оберненої операції MX, тобто передає дані із одного вхідного каналу в один із декількох каналів-приймачів (рис. 5). DC, який має вхід дозволу роботи, буде функціонувати в режимі DMX, якщо на вхід дозволу подавати інформаційний сигнал.

3.3. Суматори

Суматори SM виконують арифметичне додавання двійкових чисел. В основі багатьох схем SM лежить однорозрядний SM **a_i, b**

(OSM) (рис. 6). OSM має три входи (операнди a_i і b_i) та перенос із попереднього розряду c_{i-1} та два виходи (сума S_i і перенос в наступний розряд c_i). Таблиця істинності OSM має наступний вигляд (рис. 7). Вирази функцій суми S_i та переноса c_i :

$$S_i = \bar{a}_i \bar{b}_i c_{i-1} \vee \bar{a}_i b_i \bar{c}_{i-1} \vee a_i \bar{b}_i \bar{c}_{i-1} \vee a_i b_i c_{i-1};$$

$$c_i = a_i b_i \vee a_i c_{i-1} \vee b_i c_{i-1}.$$

Однорозрядні SM об'єднуються в багаторозрядні суматори (рис. 8). На виходах S_{n-1}, \dots, S_0 з'являється код, що відповідає сумі n-розрядних чисел.

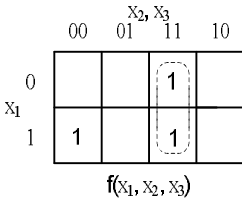


Рис. 1. Карти Карно 3-х та 4-х змінних

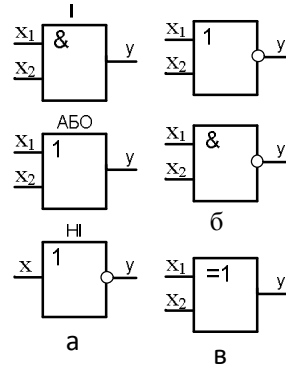
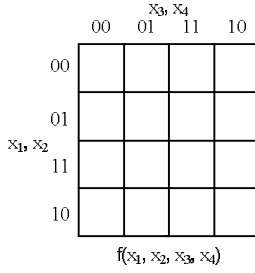


Рис. 2. Логічні елементи

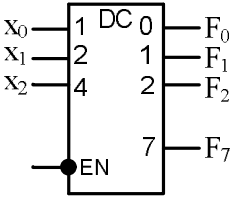


Рис. 3. DC 3x8

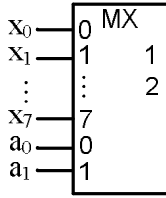


Рис. 4. MX 8x1

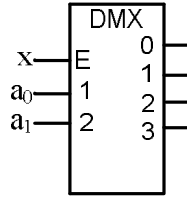


Рис. 5. DMX 1x4

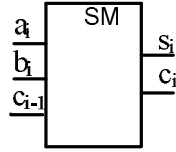


Рис. 6. OSM

a_i	b_i	c_{i-1}	s_i	c_i
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Рис. 7. Таблиця істинності OSM

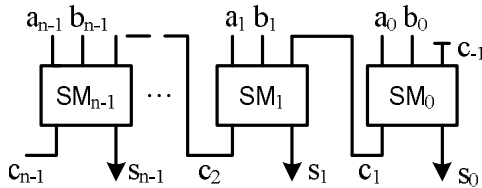


Рис. 8. Паралельний суматор із послідовним переносом

Обладнання, прилади та матеріали

ЛР пов'язана з розв'язанням задач аналізу та синтезу комбінаційних схем (пристроїв). Перевірку рішення задач забезпечує комп'ютерна програма «Інтерпретатор булевих функцій» [4]. Для проведення ЛР необхідний клас з ПК.

Заходи безпеки при виконанні ЛР

Заходи безпеки відповідають вимогам, яких необхідно дотримуватись при роботі в комп'ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання, основні теоретичні положення та розв'язки задач аналізу та синтезу цифрових комбінаційних схем, що виникають при проектуванні вузлів ЕОМ.

2. Результати домашнього завдання необхідно представити викладачу та захистити їх.

3. Правильний розв'язок задач контролюється «Інтерпретатором булевих функцій». При наявності помилок необхідно їх ідентифікувати.

4. Об'єктами задач є логічні елементи; дешифратори/шифратори; мультиплексори/демультиплексори; комбінаційні суматори.

5. Елементна база – ІМС серії КР1533 [3].

6. Варіанти завдань представлені в табл.1.

Таблиця 1

Варіанти завдань

№	БФ	DC/CD	MX/DMX	SM
1	2	3	4	5
1	1.1	2.1	3.1	4.1
2	1.2	2.2	3.2	4.2
3	1.3	2.1	3.1	4.3
4	1.4	2.3	3.2	4.4
5	1.5	2.4	3.1	4.2
6	1.8	2.2	3.2	4.5

Продовження таблиці 1

1	2	3	4	5
7	1.7	2.1	3.1	4.7
8	1.6	2.3	3.2	4.4
9	1.9	2.4	3.1	4.2
10	1.10	2.1	3.2	4.3
11	1.6	2.2	3.3	4.6
12	1.7	2.3	3.1	4.8
13	1.9	2.4	3.2	4.3

В таблиці 1 рядки визначають варіанти завдань, а стовпці – групи задач: БФ – булева функція; DC/CD – дешифратор/шифратор; МХ/DMX – мультиплексор/демультиплексор; SM – суматор.

1. Задачі групи БФ

Реалізують булеву функцію $y(x_1, x_2, \dots, x_n)$. Базові операції: «І, АБО, НІ», «АБО-НІ», «І-НІ».

1.1. Визначити ПФ, яка реалізується КС (рис.9). Спростити БФ та побудувати нову схему в базисі «АБО-НІ» на ІМС КР 1533.

1.2. Спростити комбінаційну схему (рис.10) та побудувати нову в базисі «АБО-НІ».

1.3. Синтезувати КС у базисі «І, АБО, НІ» у якій сигнал на виході $y=1$ при різночинних сигналах на виходах x_1, x_2 .

1.4. Вирішити задачу №1.3, використовуючи тільки логічний елемент «І-НІ».

1.5. Задача №1.3, але використаний логічний елемент «АБО-НІ».

1.6. На входи КС (задача 1.3) поступають імпульси (рис.11). Визначить діаграму вихідного сигналу $y(t)$.

1.7. Синтезувати КС у базисі «І-АБО-НІ», у якій вихідний $y=1$ при однакових сигналах x_1, x_2 на входах.

1.8. На входи КС (задача 1.7) поступають вхідні імпульсні сигнали (рис.12). Визначити часову діаграму вихідного сигналу $y(t)$.

1.9. Комбінаційна схема, що має 4 входи $\{(x_1, x_2, x_3, x_4)\}$ генерує вихідний сигнал $y(t)=I$, якщо тільки на входах x_1, x_2, x_3 одночасно з'являються «1». КС реалізується на ЛЕ «І, АБО, НІ».

1.10. Синтезувати КС в базисі «АБО-НІ», якщо ПФ задана у числовому вигляді $y\{(x_1, x_2, x_3, x_4)\} = \sum(1, 4, 8, 10, 11, 14)$.

1.11. Задача 1.10 в базисі «І-НІ».

2. Задачі групи DC/CD

2.1. Створити повний DC (3×8) на ЛЕ «І, АБО, НІ».

2.2. В задачі 2.1 схему DC перетворити в декодер-демультиплексор (стробований DC).

2.3. Декодер – демультиплексор типу «ИДЗ» (ІМС КР15.33) використати для побудови повного DC на п'ять входів.

2.4. Синтезувати шифратор CD на п'ять входів (рис.13) на ЛЕ «АБО-НІ».

2.5. Задачу 2.4 реалізувати на ЛЕ «І-НІ».

3. Задачі групи МХ/DMX

3.1. Синтезувати перетворювач десяткового коду у двійково-десятковий код. Базис «І-НІ» .

3.2. Синтезувати мультиплексом МХ на 8 входів на ЛЕ «І, АБО, НІ» (рис.14).

4. Задачі групи SM

4.1. Розробити схему порівняння двох 5-розрядних двійкових чисел, ЛЕ – «Виключення АБО».

4.2. Синтезувати однорозрядний суматор (OSM) на ЛЕ «І, АБО, НІ».

4.3. Розробити 4-розрядний паралельний суматор із послідовним переносом.

4.4. Синтезувати OSM на ЛЕ «І-НІ» мікросхема типу (ЛАЗ).

4.5. Задача 4.3. Визначити максимальний час виконання операції, якщо час затримки одного OSM $t_{3max}=100нс$.

4.6. Задача 4.3. Чи можливо знімати дані із суматора в момент часу $t < t_{3max}$ після подачі вхідних сигналів.

4.7. Задача 4.3 Для який цілей може бути використаний 3-й вхід суматора початкового (першого) розряду.

4.8. Задача 4.3 Як можна прискорити виконання операції додавання?

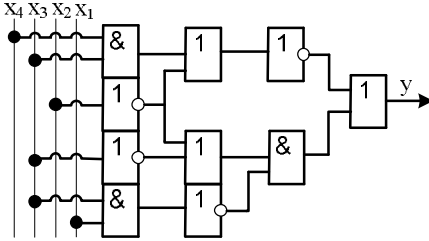


Рис. 9. Задача 1.1

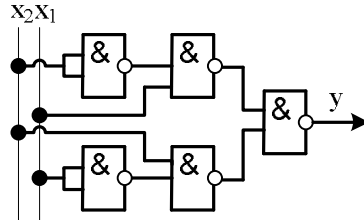


Рис. 10. Задача 1.2

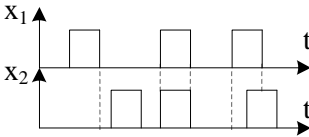


Рис. 11. Задача 1.6

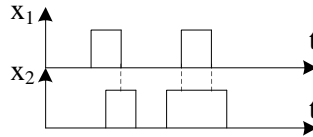


Рис. 12. Задача 1.8

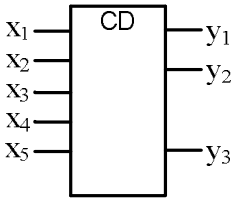


Рис. 13. Задача 2.4

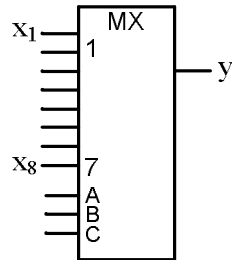


Рис. 14. Задача 3.2

Оформлення результатів експерименту

Результати експерименту оформлюються у вигляді розв'язків задач: аналітичний вираз; спрощена комбінаційна схема; комбінаційний вузол.

Обробка експериментальних даних

Аналізуються розв'язки задач: багатовимірність; мінімальність аналітичного виразу; ціна комбінаційної схеми відповідно до числа ЛЕ або числа корпусів ІМС.

Аналіз одержаних результатів

Формуються висновки по результатам виконання ЛР.

Запитання для самоперевірки

1. Визначіть поняття - комбінаційна схема.
2. Скільки стовпців має таблиця істинності OSM?
3. Назвіть базові операції алгебри Буля.
4. Таблиця істинності схеми порівняння двійкових чисел.
5. Назвіть функції груп входів мультиплексора.
6. Карта Карно. Призначення, принцип побудови.
7. Форми представлення булевих функцій.
8. Шифратор. Призначення, реалізація.
9. Повний та неповний дешифратор.

Джерела

[1, 4].

ТЕМА 2. СХЕМОТЕХНІКА ЦИФРОВИХ ВУЗЛІВ

Лабораторна робота

ЦИФРОВІ ПРИСТРОЇ З ПАМ'ЯТТЮ

Мета роботи. Дослідження та закріплення теоретичних знань по організації і використанню цифрових пристроїв з пам'яттю: тригерів; регістрів; лічильників.

Основні завдання роботи

1. Отримати варіант домашнього завдання.
2. Вивчити теоретичну частину лабораторної роботи.
3. Виконати аналіз та розробку цифрового пристрою.

Основні теоретичні відомості

1. Тригер. Це пристрій, які може знаходитись в одному із двох стійких станів та переходити з одного стану в інший під дією зовнішніх сигналів. Функція, яка встановлює залежність нового стану тригера від поточного стану та заданих сигналів керування називається функцією переходів. Вона може задаватись логічною (булевою) формулою або у вигляді таблиці переходів.

Класифікація тригерів може виконуватись по ознаці логічного функціонування (RS, -D, -T, -JK та ін.) та способу запису (асинхронні, синхронні) [1].

В асинхронних (нетактованих) тригерах зміна стану викликається зміною входних інформаційних сигналів. В синхронних (тактованих) тригерах перехід відбувається тільки при подачі на спеціальний вхід тактових сигналів («С»). По способу прийняття тактових сигналів тригери діляться на керовані рівнем та керовані фронтом. У першому випадку тригер сприймає входні сигнали тільки при одному рівні тактового сигналу, а при керуванні фронтом переключення стану відбувається тільки в момент перепаду синхросигналу наростання або спаду. В інший час тригер не сприймає входні сигнали і залишається в незмінному стані.

Діаграми на рис. 1 пояснюють дію синхросигналу С та умовні позначення синхровходів.

1.1. Асинхронний RS- тригер (рис.2). Має два входи «S» - встановлення в стан «1» та «R» встановлення в стан 0. Вихідні сигнали Q та \bar{Q} визначають поточний стан тригера. Через Q' в таблицях переходів позначено стан, в який тригер переходить.

На рис. 2.а показані схема реалізації асинхронного RS- тригера на мікроелементах ЛЕ «І - НІ», умовне позначення та таблиця переходів. Тригер управляється нульовими сигналами на входах \bar{S}, \bar{R} . При $\bar{R}=0$ тригер встановлюється у стан $Q'=0$, при $\bar{S}=0$ - у стан $Q'=1$. Комбінація сигналів на входах $\bar{S}=\bar{R}=0$ є забороненою.

Тригер на рис. 2.б реалізований на ЛЕ «І - НІ» і керується сигналами на одному із його входів S або R. При S=1 тригер встановлюється у стан $Q'=1$ при R=1 у стан $Q'=0$. При значенні сигналів R=S=1 стан Q' буде невизначеним.

1.2. Синхронний RS-тригер (рис. 3). Має синхровхід C та інформаційні входи S , R . При $C=0$ стан тригера не змінюється при, $C=1$ в залежності від значення сигналів S , R стан тригера змінюється. Крім входів, що синхронізуються синхронний тригер може мати асинхронні входи \bar{S} , \bar{R} (рис. 3.б.). Керування станом тригера по входах \bar{S} , \bar{R} виконується незалежно від синхросигналу C .

1.3. JK-тригер. На рис. 4 показані його умовне позначення, таблиця переходів та функція переходів. JK-тригер не змінює свого стану при $C=1$, якщо $J=K=0$. При $J=1$, $K=0$ тактовий сигнал C змінює стан тригера $Q' = 1$, а при $J=0$, $K=1$ тактовий сигнал C – в $Q' = 0$. Тригер не змінює свого стану при $C=0$.

1.4. T-тригер. Умовне позначення та таблиця переходів показані на рис. 5. T- тригер реалізується із JK-тригера, якщо в останньому об'єднати входи J та K . JK-тригер є універсальним, тому що з нього легко реалізувати RS- та T- тригери.

1.5. D-тригер. Умовне позначення, таблиця переходів показані на рис.6. По синхросигналу C тригер приймає стан, що відповідає значенню сигналу на його вході D . Тому його називають ще тригером затримки.

Розглянуті тригери випускаються у вигляді ІМС, наприклад серії КР 1533 [3].

Таблиці істинності та функції переходів повністю відображають функціонування тригера. Кожний рядок у таблиці визначає стан Q' , в який переходить тригер із стану Q , якщо визначені керуючі сигнали на його входах.

2. Регістри. Призначені для зберігання, запису та видачі інформації у вигляді двійкового коду. Крім цих функцій регістр може реалізувати зсув коду, порозрядні логічні операції. Для зберігання кожного двійкового розряду в регістрі використовують один тригер. Прикладом може бути ІМС КР 1533ИР 29 [3].

По способу прийому та видачі слів регістри поділяють на паралельні, зсувні та паралельно-зсувні.

В паралельному регістрі прийом та видача слів виконується по усім розрядам одночасно. Тригерні розряди не мають зв'язків між

собою. Загальними для розрядів є ланцюги тактування, скиду/встановлення дозволу виходу або прийому. На рис. 7 зображений регістр на D- тригерах з прямими входами, входом скиду \overline{R} та виходами з трьома станами, що керується сигналом EN.

Зсувний регістр – це група тригерів, які з'єднані ланцюгами переносів. Код в регістрі може бути зсунутий вліво або вправо. Такий регістр на D- тригерах показаний на рис. 8. Зсув виконується по передньому фронту синхросигналу С в одному напрямку (вправо або вліво). Запис коду виконується по входу «X» послідовно біт за бітом. Регістр, в якому реалізуються обидва типи зсуву називають реверсивним.

Універсальний багаторежимний регістр. Це регістр, в якому можливі такі режими: паралельний синхронний запис; паралельно-послідовний синхронний запис; реверсивний. Прикладом багаторежимного регістру є ІМС КР 1533ІР 24 (рис. 9). Це 8-розрядний регістр, що має паралельні сумісні входи та виходи, вхід асинхронного скиду \overline{R} та входи вибору режиму S_0 і S_1 , що задають чотири режими: паралельне завантаження, два зсуви та зберігання, а також бітові входи SR (зсув вправо), SL (зсув вліво).

3. Лічильники. Це пристрої, які під дією вхідних імпульсів переходять із одного стану в інший, фіксуючи при цьому число імпульсів. Основною операцією лічильника є зміна їх стану на одиницю. Лічильники виконують і інші операції – скид, встановлення, паралельне завантаження та інші.

Лічильник характеризується модулем лічби М, що визначає число можливих станів. Напрямок лічби може бути прямий (+1), обернений (-1) та реверсивний. Двійковий лічильник має модуль $M= 2^n$. Лічильник може будуватись на тригерах будь-якого типу. Найбільш використовуваним (вживаним) є синхронний Т-тригер, який отримують із D- або JK тригерів (рис. 10). Каскадне включення Т-тригерів (рис. 11) реалізує послідовний сумуючий лічильник з коефіцієнтом лічби $2^2=4$. Після подання на вхід чотирьох імпульсів лічильник повертається в початковий стан. На рис. 12 показаний двохрозрядний послідовний віднімаючий

лічильник, в якого входи тригерів з'єднані з інверсними виходами попередніх тригерів. На часовій діаграмі початковому стану лічильника відповідає двійкове число 11. Послідовність імпульсів зменшує вміст лічильника до 00.

При об'єднанні схем керування входами сумуючого та віднімаючого лічильників отримуємо реверсивний лічильник. Прикладом двійкового реверсивного лічильника є ІМС КР 1533ІЕ13 [3].

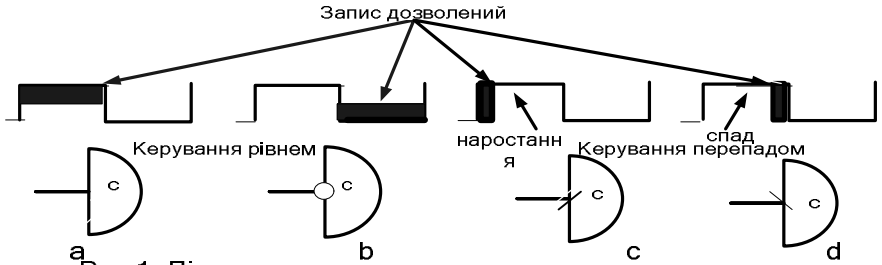


Рис.1. Діаграми запису синхронного тригера та умовне позначення синхровходів "С": а - прямий рівень; б - інверсний рівень; с- керування фронтом; d - керування спадом

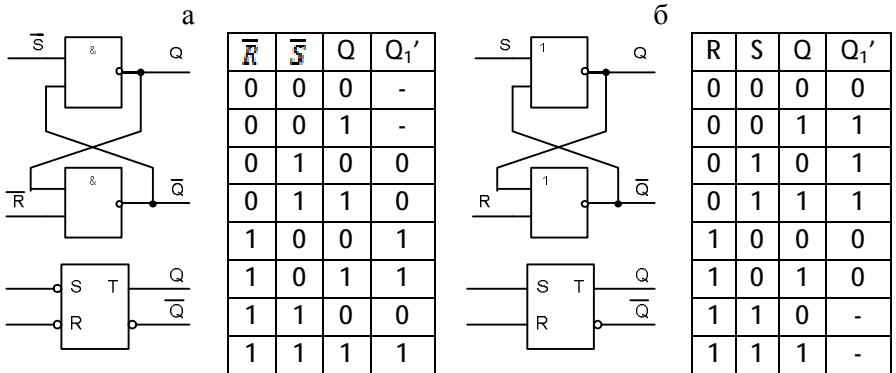
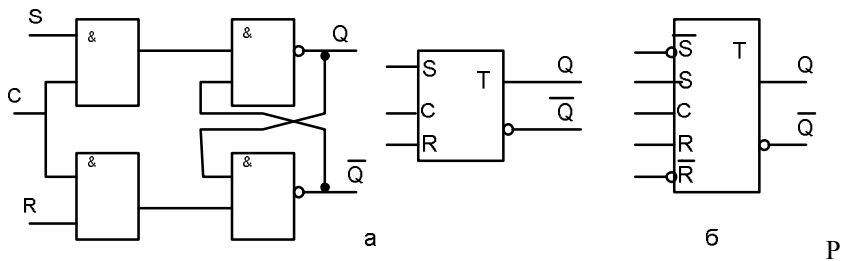


Рис. 2. Асинхронний RS-тригер: а – на ЛЕ "І-НІ" ; б – на ЛЕ "АБО-НІ"



ис. 3. Синхроний RS-тригер: а – на ЛЕ “І-НІ” та умовне позначення ; б – з асинхронними входами \overline{S} , \overline{R}

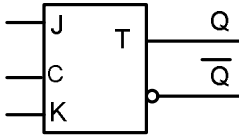


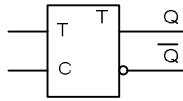
Рис. 4. JK-триггер

J	K	Q	Q'
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	∅

$$Q' = QK \cup \bar{Q}J$$

T	Q	Q'
0	0	0
0	1	1
1	0	1
1	1	0

Рис. 5. Т-тригер



D	Q	Q'
0	0	0
0	1	0
1	0	1
1	1	1

Рис. 6. D-тригер

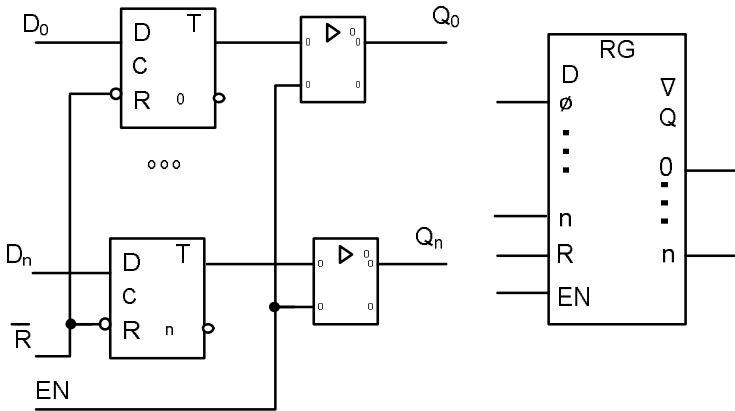
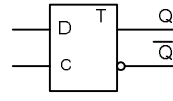
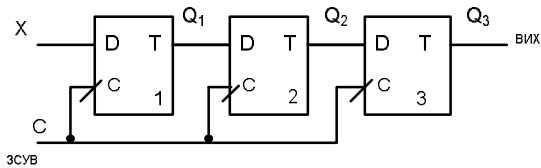
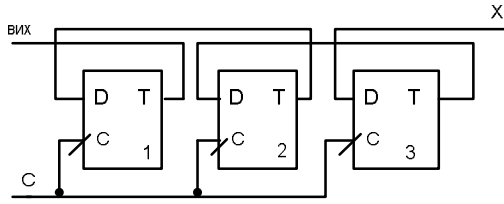


Рис. 7. Паралельний регістр та його умовне позначення



а



б

Рис. 8. Зсувний регістр: а – зсув вліво; б – зсув вправо

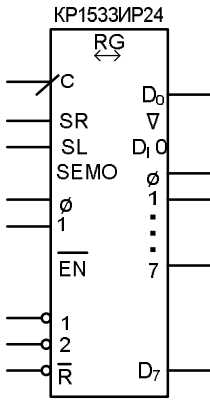


Рис. 9. Універсальний регістр ИР

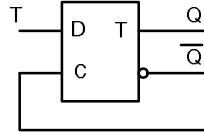
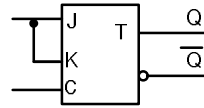


Рис. 10. Т-тригери на основі JK- та D-тригерів

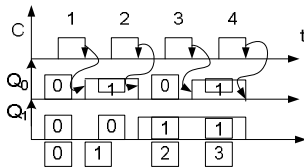
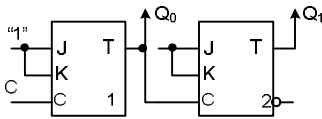


Рис. 11. Послідовний лічильник

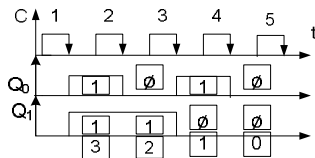
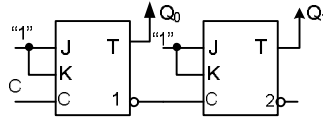


Рис. 12. Послідовний віднімаючий лічильник

Обладнання, прилади та матеріали

В ЛР розв'язуються задачі аналізу та розробки цифрових пристроїв з пам'яттю. Для виконання роботи використовується комп'ютерний клас.

Заходи безпеки при виконанні ЛР

Заходи безпеки визначають перелік вимог, які необхідно виконувати при роботі в комп'ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання, основні теоретичні положення та розв'язки задач аналізу та розробки цифрових пристроїв з пам'яттю, що виникають при проектуванні вузлів ЕОМ. Ця робота є домашнім завданням.

2. Об'єктами задач є: тригери, регістри, лічильники.

3. Результати домашнього завдання представляються викладачу і захищаються.

4. При наявності помилок необхідно їх ідентифікувати і виправити.

5. Елементна база – ІМС серії КР1533 [3].

6. Варіанти завдань представлені в табл.1.

Таблиця 1

Варіанти завдань

№ варіанта	Тригер	Регістр	Лічильник
1	1.1	2.1	3.1
2	1.2	2.2	3.2
3	1.3	2.3	3.3
4	1.4	2.4	3.4
5	1.5	2.1	3.5
6	1.6	2.2	3.1
7	1.2	2.3	3.2
8	1.1	2.4	3.3

9	1.3	2.1	3.4
10	1.4	2.2	3.5
11	1.5	2.3	3.2
12	1.6	2.4	3.3
13	1.2	2.1	3.1

В табл. 1 рядки визначають варіанти завдань, а стовпці – групи задач: тригер; регістр; лічильник.

1. Задачі групи «Тригер»

1.1. На рис.13 показані схема асинхронного RS – тригера на ЛЕ «I – НІ » та його умовне позначення:

- поясніть принцип його роботи;
- скласти таблицю функціонування тригера;
- записати власну функцію тригера;
- побудувати часову діаграму роботи тригера з урахуванням затримок його елементів;

• на вхід тригера подані сигнали $S=R=0$ – заборонена комбінація. Потім ці сигнали змінюються на $S=R=1$. Яким буде стан тригера?

1.2. На рис. 14 показана схема асинхронного RS – тригера на ЛЕ «АБО – НІ » та його умовне позначення:

- скласти таблицю функціонування тригера;
- записати власну функцію тригера;
- на вхід тригера подані сигнали $S=R=1$. Потім вони змінюються на $S=R=0$. Яким буде стан тригера?

1.3. Накресліть схему синхронного одноступеневого RS – тригера на ЛЕ «I – НІ ». Скласти таблицю переключень .

1.4. Розробіть схему двохступеневого RS – тригера, використовуючи тригер (Задача 1.3) .

1.5. Побудувати одноступеневий D – тригер на базі синхронного RS – тригера.

1.6. На основі JK – тригера (рис. 15) отримати T та D – тригери.

2. Задачі групи «Регістр»

2.1. Побудувати 4 – розрядний паралельний регістр на тригерах (рис.16) :

- чи потрібно перед записом нової інформації стерти стару?
- що виникне, якщо зміниться інформація на вході регістра під час дії CLK C ?

2.2. На рис. 17 показана схема 3 – розрядного зсувного регістра на D – тригерах та його умовне позначення:

- поясніть принцип роботи регістра на прикладі просування однієї «1»;
- чи можливе використання в регістрі JK – тригерів?

2.3. Як використати зсувний регістр (рис. 17) в якості розподільника імпульсів (сигнал «1» з'являється почергово на кожному виході).

2.4. У цифровому пристрої необхідно передавати інформацію у паралельному коді від декількох регістрів одному приймачу (регістру) по загальній шині:

- чи можливо до цієї шини підключити регістр (рис. 17)?
- як можна підключити названі регістри до загальної шини?

3. Задачі групи «Лічильник»

3.1 На рис. 18 показана схема 3 – розрядного двійкового лічильника на D – тригерах:

- принцип роботи лічильника?
- скласти таблицю переключень, визначити коефіцієнт перерахування «К»;
- побудувати часові діаграми сигналів на виходах Q_1 , Q_2 , Q_3 .
- визначити максимальну затримку, яка створюється лічильником. Затримка одного тригера $t_{зтр} = 50$ нс.

3.2 Лічильник (рис. 18) необхідно перетворити на віднімаючий?

3.3 Сумований лічильник (рис. 18) знаходиться в нульовому стані. В якому він буде стані після подання 64 та 67 імпульсів ?

3.4 На рис. 19 показаний розподільник імпульсів:

- пояснити принцип роботи;
- визначити число виходів;

- визначити часову діаграму.

3.5 Лічильники з паралельним переносом (рис. 20) мають затримки $\tau_3=50$ нс, ЛЕ «І» - 10 нс. Визначити затримку всього лічильника та його коефіцієнт переліку (модуль лічби).

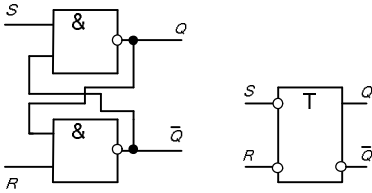


Рис.13. Асинхронний RS-тригер на ЛЕ „І-НІ”

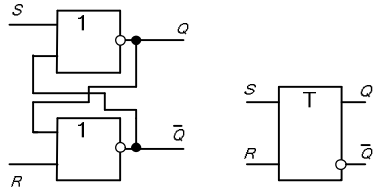


Рис.14. Асинхронний RS-тригер на ЛЕ „АБО-НІ”

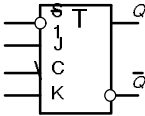


Рис.15. JK-тригер КР 1533 ТВ 10

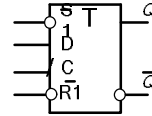


Рис.16. Тригер КР 1533 ТМ 2

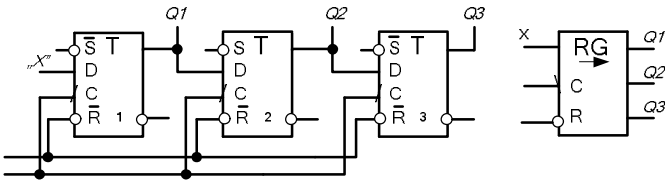


Рис.17. 3-розрядний зсувний регістр

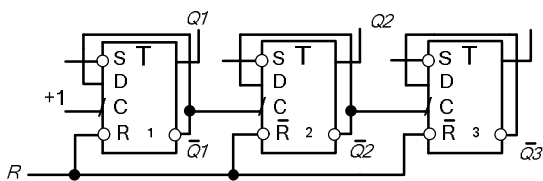


Рис. 18. Двійковий лічильник на D-тригерах

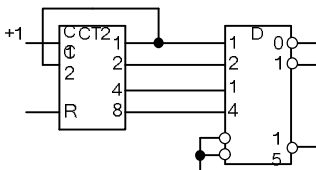


Рис.19. Розподільник імпульсів (задача 3.4)

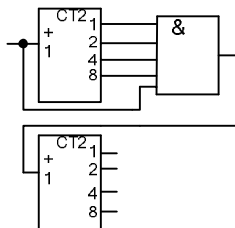


Рис.20. Схема лічильника (задача 3.5)

Оформлення результатів експерименту

Результатами експерименту є розв'язки задач в групах. Це можуть бути розроблені або спрощені функціональні схеми, аналітичні вирази.

Обробка експериментальних даних

Аналізуються розв'язки задач: аналітичний вираз; складність функціональної схеми, число корпусів ІМС.

Аналіз одержаних результатів

Формуються висновки по результатам виконання лабораторної роботи.

Запитання для самоперевірки

1. Назвіть типи тригерів, їх особливості, області використання.
2. Чим відрізняється RS – тригер від D – тригера?
3. Що означає поняття реверсний регістр?
4. Які необхідно виконати схемо технічні рішення для реалізації реверсної функції в регістрі?

5. Як визначається коефіцієнт переліку (пересчета)?
6. В якому реєстрі відсутня операція встановлення у початковий стан?
7. Який тип тригера використовується у лічильнику?
8. Яке призначення розподільника сигналів (імпульсів)?

Джерела
[1, 4].

ТЕМА 3. ІНТЕГРОВАНІ СИСТЕМИ ЕЛЕМЕНТІВ

Лабораторна робота

КЕРУЮЧІ АВТОМАТИ

Мета роботи. Дослідження та закріплення теоретичних знань по синтезу керуючих автоматів (КА) з довільними переходами.

Основні завдання роботи

1. Отримати варіант домашнього завдання.
2. Вивчити теоретичну частину роботи.
3. Виконати синтез керуючого автомату по графу мікропрограми (ГМП).

Основні теоретичні відомості

Математичною моделлю КА є абстрактний автомат, який визначається вектором

$$S = (A, X, Y, \delta, \lambda, a_1),$$

де $A=(a_1, \dots, a_n)$ - множина станів, $X=(x_1, \dots, x_F)$ - множина вхідних сигналів, $Y=(y_1, \dots, y_G)$ - множина вихідних сигналів, δ і λ - відповідно функції переходів і виходів, a_1 - початковий стан автомата.

Найбільше поширення на практиці одержали автомати Мілі й Мура, закон функціонування яких заданий відповідно рівняннями (1) і (2) [5]:

$$\begin{aligned} a(t+1) &= d(a(t), x(t)), & (1) \quad a(t+1) &= d(a(t), x(t)), & (2) \\ y(t) &= I(a(t), x(t)), & & & & y(t) &= I(a(t)). \end{aligned}$$

Розрізняють абстрактний і структурний синтез автомата.

На етапі абстрактного синтезу визначається закон функціонування й мінімізуються стани автомата.

Метою структурного синтезу є побудова схеми автомата з елементів заданого типу. Структурний синтез базується на використанні елементарних автоматів спеціального виду: з пам'яттю, що мають більше одного стану, й без пам'яті – з одним станом.

Структурний метод синтезу припускає представлення схеми автомата у вигляді двох частин: запам'ятовуючої (ЗЧ) і комбінаційної (КЧ) (рис. 1).

Комбінаційна частина реалізує залежність вихідних сигналів Y та сигналів збудження V від вхідних сигналів автомату X та сигналів стани U .

Запам'ятовуюча частина складається з попередньо вибраних елементарних автоматів пам'яті P_1, \dots, P_k . Таким чином, синтез структурного автомата зводиться до вибору елементів пам'яті, кодування станів і синтезу КЧ.

При синтезі КА по ГМП закон функціонування визначається так. Спочатку виконується відмітка ГМП станами автомата, потім відшукуються всілякі переходи в ГМП між сусідніми станами a_1, a_2, \dots .

Для автомата Мілі символом (станом) a_1 відзначаються вихід початкової та вхід кінцевої вершин. Виходи всіх операторних вершин послідовно відзначаються станами a_2, a_3 і т.д. Приклад відмітки ГМП станами автомата Мілі наведений на рис. 2, *a*.

У випадку автомата Мура кожній операторній вершині ГМП ставиться у відповідність деякий стан автомата. На ГМП (рис. 2, *a*) стани позначені поруч із операторними вершинами.

Граф автомата по відзначеному ГМП визначається виявленням усіляких шляхів переходу між сусідніми станами a_1, a_2, \dots . На кожному переході зі стану a_m у стан $a_s(a_m, a_s)$ повинна зустрічатися тільки одна операторна вершина. На переході виду (a_m, a_1) операторна вершина може бути відсутня. ГМП (рис. 2, *a*) відповідає графу автоматів Мілі (рис. 2, *б*) і Мура (рис. 2, *в*).

Автомат Мура дає рівну або більшу кількість станів, чим автомат Мілі.

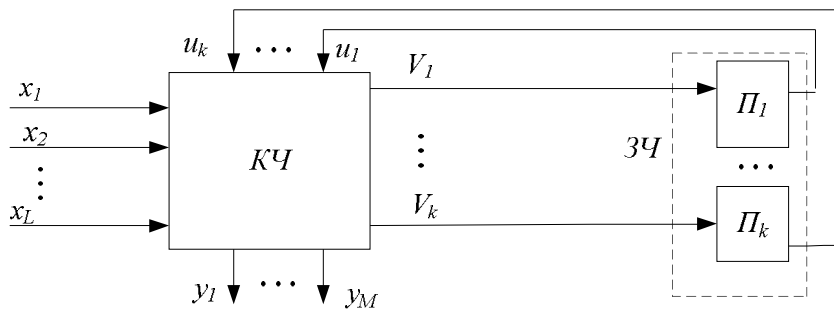
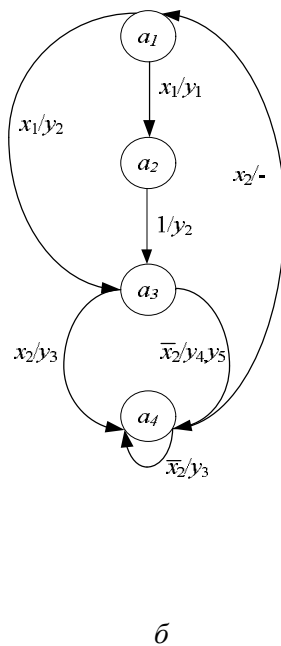
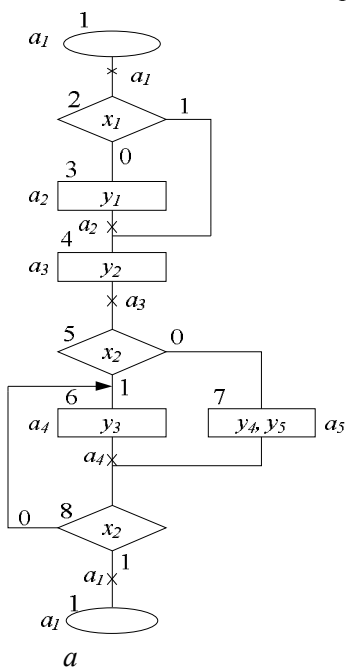
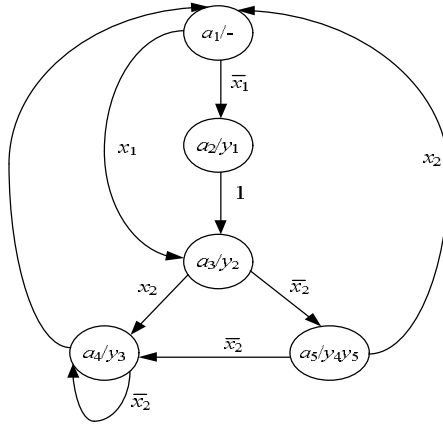


Рис. 1. Структурна схема КА





6

Рис. 2. ГМП та графи автоматів Мілі та Мура

При великій кількості станів закон функціонування автомата (граф автомата) зручніше представляти у вигляді структурної таблиці переходів (табл. 1).

Таблиця 1

Загальний вид структурної таблиці переходів автомата Мілі

Початковий стан	Код початкового стану	Стан переходу	Код стану переходу	Вхідний сигнал	Вихідний сигнал	Сигнал збудження
a_m	$K(a_m)$	a_s	$K(a_s)$	X	Y	V

Таблиця переходів автомата Мура має на один стовбець менше, тому що вихідний сигнал у ній записується поруч із початковим станом a_m .

Обладнання, прилади та матеріали

В ЛР розв'язується задача синтезу КА по ГМП. Для виконання роботи використовується клас ПК.

Заходи безпеки при виконанні ЛР

Заходи безпеки відповідають вимогам, при роботі в комп'ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання, та синтез КА по ГМП. При цьому визначаються закон функціонування й кодування станів автомата.

2. Результати домашнього завдання необхідно представити викладачу та захистити їх.

3. Розв'язок задачі синтезу визначається за допомогою програми «РІСТПІК», яка виконує розмітку ГМП, формує структурну таблицю переходів та кодування станів [4].

4. Машинна операція задається у вигляді ГМП, наприклад (рис. 3, а) або в табличному вигляді (рис. 3, г).

№ вершини ГМП	М		Зміст
	1	2	3
1	0	2	-
2	3	4	x_1
3	0	4	y_1
4	0	5	y_2
5	7	6	x_2
6	0	8	y_3
7	0	8	y_4, y_5
8	6	1	x_2

Умовні вершини	
1	2
2	1
5	2
8	2

б

а

Рис. 3. Початкові дані: а – масив М; б – масив УВ

5. Початковими даними для програми «РІСТПІК» є ГМП (рис. 2, а), який завантажується масивами М та УВ. Масив М (рис. 3, а) описує зв'язки між вершинами ГМП, а масив УВ (рис. 3, б) – умовні вершини.

6. Варіанти завдань наведені в табл.2. Таблиці, що описують ГМП, одержати у викладача.

Оформлення результатів експерименту

Результатом ЛР є вектор розмітки ГМП операції, структурна таблиця переходів та закодовані стани КА.

Обробка експериментальних даних

Порівнюються результати отримані в домашньому завданні та комп'ютерною програмою «РІСТПШК».

Аналіз одержаних результатів

Формуються висновки по результатам виконання ЛР.

Запитання для самоперевірки

1. Яке призначення КА?
2. У чому складається відмінність автоматів Мілі й Мура?
3. У чому полягає структурний метод синтезу КА?
4. Назвіть основні етапи структурного синтезу КА по ГМП.
5. Намалюйте структурну схему КА.
6. Для чого необхідно оптимальне кодування станів автомата?
7. Як виконується розмітка станами автоматів Мілі, Мура?

Джерела

[4, 5]

Таблиця 2

Варіанти домашнього завдання

Варіант	Автомат	ГМП № (табл.)	ЕП	ЛЕ
1	2	3	4	5
1	1	1	<i>T</i>	I, АБО, НЕ
2	2	2	<i>T</i>	I-НЕ

Продовження таблиці 2

1	2	3	4	5
3	1	3	<i>RS</i>	АБО-НЕ
4	2	4	<i>D</i>	I-НЕ
5	1	5	<i>T</i>	АБО-НЕ
6	2	6	<i>RS</i>	I-НЕ
7	1	7	<i>T</i>	I-НЕ
8	2	8	<i>D</i>	I-НЕ
9	1	9	<i>RS</i>	АБО-НЕ
10	2	10	<i>D</i>	I-НЕ
11	1	11	<i>RS</i>	АБО-НЕ
12	2	12	<i>T</i>	I-НЕ
13	1	13	<i>D</i>	АБО-НЕ

ЕП - елемент пам'яті

ЛЕ – логічний елемент

ТЕМА 4. ЦИФРОВІ КОМП'ЮТЕРИ

Лабораторна робота

ПРИСТРІЙ КЕРУВАННЯ З ПРОГРАМОВАНОЮ ЛОГІКОЮ

Мета роботи. Дослідження та закріплення теоретичних знань по синтезу керуючих автоматів (КА) з програмованою логікою.

Основні завдання роботи

1. Отримати варіант домашнього завдання.
2. Вивчити теоретичну частину роботи.
3. Виконати синтез КА з програмованою логікою:
 - побудувати закодовану мікропрограму операції;
 - визначити формат мікрокоманди;
 - розробити структуру КА
 - підготувати початкові дані для ПП «Мікрокод».

Основні теоретичні відомості

При синтезі КА з програмованою логікою основними задачами є вибір способу кодування мікрооперацій (МО) і формування адреси наступної мікрокоманди (МК). За аналогією з кодом команди в МК виділяють операційну та адресну частини.

Для кодування операційної частини (ОЧ) застосовують горизонтальний, вертикальний та змішаний способи.

Горизонтальний і вертикальний способи реалізують мінімальну і максимальну ступені кодування. У першому випадку розрядність ОЧ максимальна і дорівнює $n_r = n_{\max} = M$, де M - загальне число МО. У другому випадку довжина ОЧ - мінімальна і дорівнює:

$$n_B = n_{\min} = \lceil \log_2 M \rceil.$$

Змішане кодування поєднує горизонтальний і вертикальний способи. Множина МО Y розбивається на " K " підмножин (полів) Y_i . На практиці знайшов спосіб кодування ОЧ МК, коли підмножини Y_i кодуються горизонтально, а МО всередині підмножин - вертикально. Довжина ОЧ МК дорівнює

$$n = \sum_{i=1}^k \lceil \log_2 M_i \rceil,$$

де M_i - число МО, що представляються у полі i .

Проблема полягає в пошуку способу розбиття множини Y на підмножини Y_1, Y_2, \dots, Y_k , який мінімізує обсяг пам'яті мікрокоманд. З цього випливає необхідність, щоб число полів було мінімальним і кожна МО представлялася лише в одному полі, тобто:

$$Y_i \cap Y_j = \emptyset. \quad (1)$$

Окрім того, підмножини Y_1, \dots, Y_k повинні відповідати вимозі повноти

$$\bigcup_{p=1}^k Y_p = Y. \quad (2)$$

Умови (1), (2) висувають умову несумісності МО одного поля. Дві МО називаються несумісними, якщо виключена принципова можливість виконання їх в одному машинному такті. Відношення

МО задаються матрицею сумісності S , яка є булевою і симетричною:

$$S = \begin{bmatrix} 0 & s_{12} & s_{1M} \\ s_{21} & 0 & s_{2M} \\ \cdot & \cdot & \cdot \\ s_{M1} & & 0 \end{bmatrix}, \quad s_{ij} = \begin{cases} 1, & \text{коли МО } - y_i \text{ і } y_j \text{ - сумісні,} \\ 0 & \text{то } - y_i \text{ і } y_j \text{ - не сумісні.} \end{cases}$$

Існує ряд способів рішення задачі змішаного кодування МК.

Одним із них є метод прямого включення. Сутність його полягає в тому, що процес розподілу МО $Y=(y_1, \dots, y_M)$ по полях Y_1, Y_2, \dots, Y_k мікрокоманди розділяється на M кроків. На кожному кроці для чергової МО y_i відшукується таке поле Y_p , вона несумісна з жодною МО цього поля. Якщо серед полів Y_1, Y_2, \dots, Y_t такого поля не існує, то для цієї МО вводиться нове поле Y_{t+1} .

Стан процесу включення на кожному кроці характеризується матрицею включення:

$$R = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1M} \\ & \cdot & \cdot & \cdot \\ & & & \\ r_{k1} & r_{k2} & \dots & r_{kM} \end{bmatrix}, \quad r_{ij} = \begin{cases} 1, & \text{при } y_i \in Y_p, \\ 0, & \text{при } y_i \notin Y_p. \end{cases}$$

Таким чином, умова включення МО y_i в поле Y_p формулюється так. МО y_i включається в поле Y_p , якщо i -й рядок S_i матриці S не перетинається з p -м рядком R_p матриці R , тобто:

$$S_i \mathbf{I} R_p = \emptyset.$$

Приклад. Виконати розподіл МО $Y=(y_1, y_2, \dots, y_6)$, відношення сумісності яких задано матрицею:

1 2 3 4 5 6

$$S = \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix}.$$

1. Для МО y_1 призначимо поле Y_1

$$R = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 0 & 0 & 0 & 0 \end{matrix}.$$

2. Визначаємо поле для МО y_2 . Оскільки $S_2 \mathbf{I} R_1 = \emptyset$, то для y_2 призначаємо нове поле Y_2

$$R = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \end{matrix}.$$

3. Визначаємо поле для МО y_3 . Оскільки $S_3 \mathbf{I} R_1 = \emptyset$, то МО y_3 включається в поле Y_1

$$R = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \end{matrix}.$$

4. Визначаємо поле для МО y_4 . Оскільки $S_4 \mathbf{I} R_1 = \emptyset$, то МО y_4 включається в поле Y_1

$$R = \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 & 0 \end{matrix}.$$

5. Призначимо поле для МО y_5 . Оскільки $S_5 \mathbf{I} R_1 = \emptyset$, то МО y_5 включається в поле Y_1

$$1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6$$

$$R = \frac{1}{2} \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

6. Призначимо поле для МО y_6 . Оскільки $S_6 \mathbf{I} R_1 = 0$ і $S_6 \mathbf{I} R_2 = 0$, то МО y_6 включається в поле Y_3

$$R = \frac{1}{3} \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Таким чином, одержимо такий розподіл МО

$$Y_1 = (y_1, y_3, y_4, y_5), Y_2 = (y_2), Y_3 = (y_6).$$

Визначимо довжину ОЧ МК і закодуємо МО в підмножинах Y_i (табл. 1)

$$n = \lceil \log_2(M_1 + 1) \rceil + \lceil \log_2(M_2 + 1) \rceil + \lceil \log_2(M_3 + 1) \rceil = 3 + 1 + 1 = 5.$$

Таблиця 1

Підмножини МО

Y_1		Y_2		Y_3		Примітки
y_i	$K(Y_1)$	y_i	$K(Y_2)$	y_i	$K(Y_3)$	
-	000	-	0	-	0	Немає МО
y_1	001	y_2	1	y_6	1	
y_3	010					
y_4	011					
y_5	100					

2. Для реалізації змішаного кодування вихідна інформація задається матрицею сумісності S , яка може формуватися двома способами.

2.1. Матриця складається з урахуванням структурних обмежень, що накладаються ОА. Ознаки сумісності встановлюються між кожною парою МО y_i, y_j . Нехай в ОА із закріпленням МО реалізується такий список МО:

$$y1) CT = 0, \quad y3) SM = RGA + RGB,$$

$$y4) RGA = L1(RGA), \quad y5) RGB = R1(RGB).$$

З урахуванням сказаного матриця S має вигляд

$$S = \begin{matrix} & \begin{matrix} 1 & 2 & 3 & 4 & 5 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{matrix} & \begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{bmatrix} \end{matrix}$$

2.2. Формування матриці S ґрунтується на аналізі МП, у МК яких визначаються МО, що використовуються спільно. У матриці S елемент $S_{ij}=1$, якщо МО y_i і y_j використовуються в одній МК. Сумісні МО y_p і y_t , але такі, що не використовуються спільно в МК помічаються ознакою несумісності $S_{pt}=0$.

Такий підхід призводить до зменшення числа “1” у матриці S , в результаті значно більше число МО розглядається як несумісні й об’єднуються в одну підмножину (поле). Це дозволяє скоротити довжину ОЧ МК, але при цьому губиться універсальність МК, оскільки зміни, що вносяться в МК, можуть породити нові функціональні оператори, що реалізуються за декілька тактів.

3. У відносно простих КА застосовується природна адресація МК. Адреса наступної МК визначається шляхом збільшення на 1 адрес поточної МК. Природна адресація припускає використання МК двох типів – операційної й керуючої (рис. 1). Операційна МК визначає набір МО й неявно задає адресу наступної МК на 1 більше поточної. Керуюча МК використовується для зміни природного порядку виконання МК.



Рис.1

Показник S визначає тип МК:

$$S = \begin{cases} 0 - \text{операційна,} \\ 1 - \text{керуюча.} \end{cases}$$

У полі X кодуються сигнали логічних умов (ЛУ), поле B вміщує адресу наступної МК:

$$A = \begin{cases} B, \text{ при } X = 0, \\ B, \text{ при } X \neq 0 \text{ і } x_x = 0, \\ A + 1, \text{ при } X \neq 0 \text{ і } x_x = 1. \end{cases}$$

Розрядність полів X і B визначається так:

$$n_x = \lceil \log_2(N + 1) \rceil, \quad n_B = \lceil \log_2 P \rceil,$$

де N - число оригінальних сигналів ЛУ, P - число МК в МП.

Приклад. Визначити довжину керуючої МК і закодувати сигнали ЛУ (x_1, \dots, x_4). Число МК $P=14$.

$$n_x = \lceil \log_2(4 + 1) \rceil = 3,$$

$$n_B = \lceil \log_2 14 \rceil = 4,$$

$$n_Y = n_x + n_B + n_S = 3 + 4 + 1 = 8.$$

Закодовані сигнали ЛУ наведені в табл. 2.

Таблиця 2
Сигнали логічних умов

x_i	-	x_1	x_2	x_3	x_4
X	000	001	010	011	100

Обладнання, прилади та матеріали

ЛР пов'язана з розв'язанням задачі синтезу КА з програмованою логікою. Для проведення ЛР використовується комп'ютерний клас.

Заходи безпеки при виконанні ЛР

Заходи безпеки відповідають вимогам, яким необхідно дотримуватись при роботі в комп'ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання та синтез КА з програмованою логікою.

2. Результати домашнього завдання необхідно представити викладачу.

3. При синтезі структури мікрокоманди використовується ПП «Мікрокод»[4]. Початковими даними для програми є ГМП і список мікрооперацій однойменного варіанта лабораторної роботи №1, тема 3.

4. У домашньому завданні синтезуються два варіанти операційних МК, обумовлених способами формування матриці S . У якості робочої вибирається МК, що має меншу розрядність. Розбиття мікрокоманди Y на підмножини (поля) Y_i порівнюється з результатами ПП “Мікрокод”.

5. Розрядність операційної і керуючої МК вибирається однаковою. Більш коротка МК доповнюється нульовими розрядами.

6. При природній адресації число МК «Р» визначається як сума операторних і умовних вершин ГМП. Операторним вершинам відповідають операційні МК, умовним вершинам – керуючі МК.

7. Для побудови закодованої МП вершини ГМП відмічаються адресами. На переході, де неможлива природна адресація, вводять додаткову керуючу МК, що забезпечує безумовний перехід до потрібної МК. На ГМП (рис.2) знаками 0) ,..., 6) позначені адреси МК. Адресою 5) відмічена керуюча МК, що реалізує перехід від МК з адресою 4) до МК з адресою 1). Адресу 6) має МК, що визначає кінець МП. Це може бути операційна МК, у якій у i -у полі закодована ознака кінця МП. Закодована МП має заголовок (рис. 3).

8. Структурна схема КА повинна містити пам'ять МК, регістр МК, дешифратори полів Y_1, \dots, Y_k, X , схему, що виробляє ознаку МП=0, лічильник адреси МК.

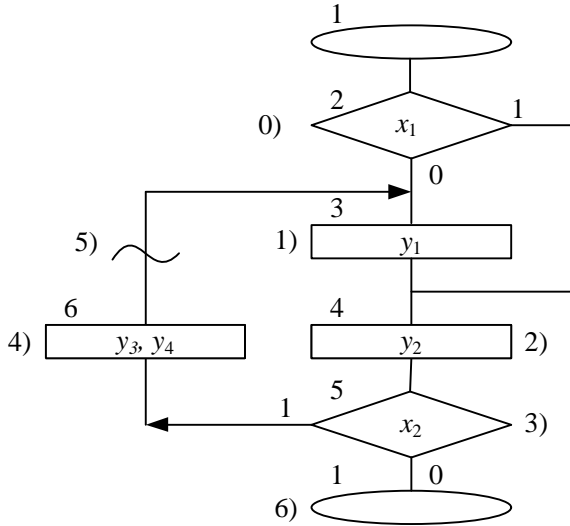
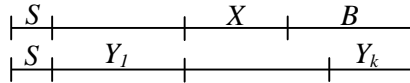


Рис. 2. ГМП



Адреса МК	МК						
	0	1	2	3	4	...	n
0							
1							
2							
3							

Рис. 3. Закодована МП

Оформлення результатів експерименту

Результатом ЛР є формат мікрокоманди та структура КА з програмованою логікою.

Обробка експериментальних даних

Порівнюються результати, які отримані в домашньому завданні та ПП «Мікрокод».

Аналіз одержаних результатів

Формулюються висновки по результатам виконання ЛР.

Запитання для самоперевірки

1. Поясніть принцип роботи КА з програмованою логікою.
2. Які способи кодування операційної частини МК?
3. Які мікрооперації називаються сумісними?
4. Як формується матриця сумісності S ?
5. У чому полягає сутність методу прямого включення?
6. Як сформувати ознаку кінця МП?
7. З якою метою в МП вводять додаткові керуючі МК?
8. Назвіть типи МК, які застосовуються при природній адресації.
9. Як визначити довжину поля X ?
10. Від чого залежить розрядність лічильника адреси МК?

Джерела

[3].

ТЕМА 5. МІКРОПРОЦЕСОРНІ СИСТЕМИ

Лабораторна робота

БЛОК ОБРОБКИ ДАНИХ

Мета роботи. Вивчення схемотехніки процесорного елемента (ПЕ) K1804 BC1, побудови блоку обробки даних (БОД) на його основі та розробка мікропрограм (МП) обчислення функцій.

Основні завдання роботи

1. Отримати варіант домашнього завдання.
2. Вивчити теоретичну частину роботи.
3. Розробити принципову схему блоку обробки даних.
4. Розробити мікроалгоритми та МП обчислення функцій, представити результати їх виконання (стани регістрів RAM, шини DO, ознаки).
5. Підготувати дані для ПП «Емулятор K1804BC1».
6. Відповісти на контрольні запитання.

Основні теоретичні відомості

Мікропроцесорний комплект (МПК) серії K1804, використовують для побудови контролерів різного призначення та ЕОМ з довільною системою команд.

1. Процесорний елемент K1804BC1

Процесорний елемент BC1 має чотирирозрядну організацію, використовують його для побудови операційних блоків процесорів з довільною розрядністю. Структурна схема ПЕ зображена на рис. 1 і містить такі вузли.

Блок внутрішньої пам'яті. Це двопортовий регістровий запам'ятовуючий пристрій з організацією RAM ємністю 16 чотирирозрядних слів та змішувач даних SHL1. Вибір довільного регістра виконують шляхом завдання його адреси на входах A(3:0) (порт A) або B(3:0) (порт B). Із RAM є можливість одночасного зчитування двох слів. Зчитані дані заносять в регістри RGA та RGB. Запис даних в RAM можливий тільки по входу B(3:0). Інформацію в RAM можна записувати як без зсуву, так і зі зсувом вправо або вліво на 1 розряд.

Сигнали SB3, SB0 виникають на двонапрямлених виводах ПЕ при виконанні операції зсуву даних перед їх записом в RAM. Якщо зсуви на змішувачі SHL1 не виконуються, то вказані виводи перебувають у третьому стані.

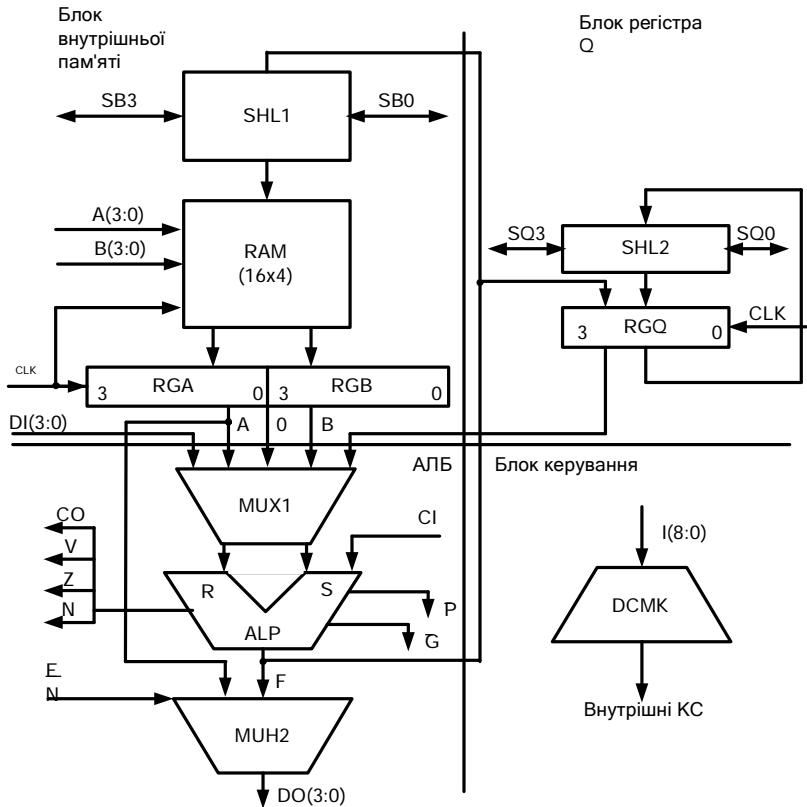


Рис. 1. Структура ПЕ K1804BC1

Арифметично-логічний блок (АЛБ). До його складу входять мультиплексор даних MUX1, арифметико-логічний пристрій (АЛП) та мультиплексор вихідних даних MUX2. Функції, які реалізує АЛП визначаються мікрокодом I (8:0).

Мультиплексор MUX1 забезпечує вибір пар операндів на входах R та S АЛП. Джерелами операндів АЛП є: шина даних DI(3:0); константа "0"; реєстр RGQ; реєстри RAM, які зчитані з каналів A та B. Керування вибором пар операндів забезпечується полем I (2:0) (табл. 1).

АЛП виконує арифметичні та логічні операції. Тип операцій в АЛП – задається полем мікрокоманди I (5:3) (табл. 2). Арифметичні операції виконують із врахуванням значення вхідного переносу CI за правилами доповняльного коду. При цьому формуються чотири ознаки результату: CO - перенос зі старшого розряду; V - переповнення; N - знак або вміст старшого розряду на виході АЛЛ, Z - ознака нульового значення F на виході АЛЛ.

Таблиця 1

Джерела операндів			Входи АЛП	
I (2:0)			R	S
12	11	10	8-а с. о.	
0	0	0	C	Q
0	0	1	1	B
0	1	0	2	Q
0	1	1	3	B
1	0	0	4	A
1	0	1	5	A
1	1	0	6	Q
1	1	1	7	O

Таблиця 2

Операції в АЛП				
I (5:3)			Операція	
15	14	13		8-а о. о.
0	0	0	0	R+S+CI
0	0	1	1	S-R-1+CI
0	1	0	2	R-S-1+CI
0	1	1	3	R ∇ S
1	0	0	4	R ∇ S
1	0	1	5	R ∇ S
1	1	0	6	R ∇ S
1	1	1	7	R ∇ S

Результат операції на виході АЛП задається словом F та видається в шину DO, в RAM або регістр RGQ відповідно до розрядів мікрокоду I (8:6) (табл. 3).

Запис слова F в регістри RAM здійснюється прямо (RAM[B]=F), зсувом праворуч (RAM[B]=R1(F)) або зсувом вліво (RAM[B]=L1(F)). В останніх двох випадках одночасно може зсуватися і вміст регістра RGQ.

Таблиця 3

Приймачі результату

I (8:0)				RAM [B]	RGQ	DO
i8	i7	I6	8-а с. о.			
0	0	0	0		F	F
0	0	1	1			F
0	1	0	2	F		A
0	1	1	3	F		F
1	0	0	4	R1(F)	RI(RGQ)	F
1	0	1	5	R1(F)		F
1	1	0	6	L1 (F)	LI (RGQ)	F
1	1	1	7	L1(F)		F

Мультиплексор MUX2 забезпечує передавання слова F з виходу АЛП або виходу A RAM на шину DO (3:0).

Блок регістра Q. Складається з регістра RGQ та змішувача SHL2. Регістр RGQ забезпечує прийом слова F ($RGQ=F$) та його зсув свого вмісту на один розряд вліво або вправо. Сигнали SQ3, SQ0 виникають на однойменних двонапрямлених виводах ПЕ при зсуві інформації в регістрі RGQ. Якщо зсув не виконується, то вказані виводи знаходяться в третьому стані.

Блок керування. Формує внутрішні КС шляхом декодування поля мікрокода I (8:0)

Вихідна шина DO (3:0) має три стани. Стан високого опору шини встановлюється сигналом $\overline{EN} = 1$.

Вихід "Z" АЛП виконаний за схемою з відкритим колектором.

Такт роботи ПЕ полягає в тому, що по позитивному перепаду тактового сигналу CLK на керуючі входи ПЕ поступає код мікрокоманди I (8:0), у відповідності з яким виконується вибір операндів, перетворення їх в АЛП та видача результату на шину DO, якщо MUX2 відкритий ($\overline{EN} = 0$).

Умовно-графічне позначення (УГП) ПЕ ВСІ та призначення виводів зображені на рис. 2.

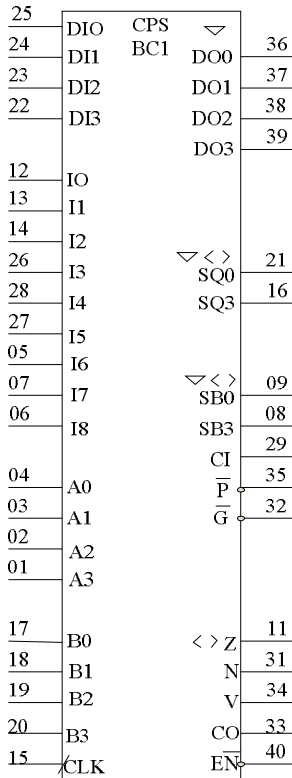


Рис. 2. Умовно-графічне позначення ПЕ K1804 BC

2. Побудова блоку оброблення даних

Необхідна довжина розрядної сітки процесора забезпечується шляхом з'єднання кількох ПЕ BC1, як показано на рис. 3.

Виходи ознак CO, F15, V, Z (ПЕ DD4) підключаються до регістра стану RGC (на рис. 3. відсутній).

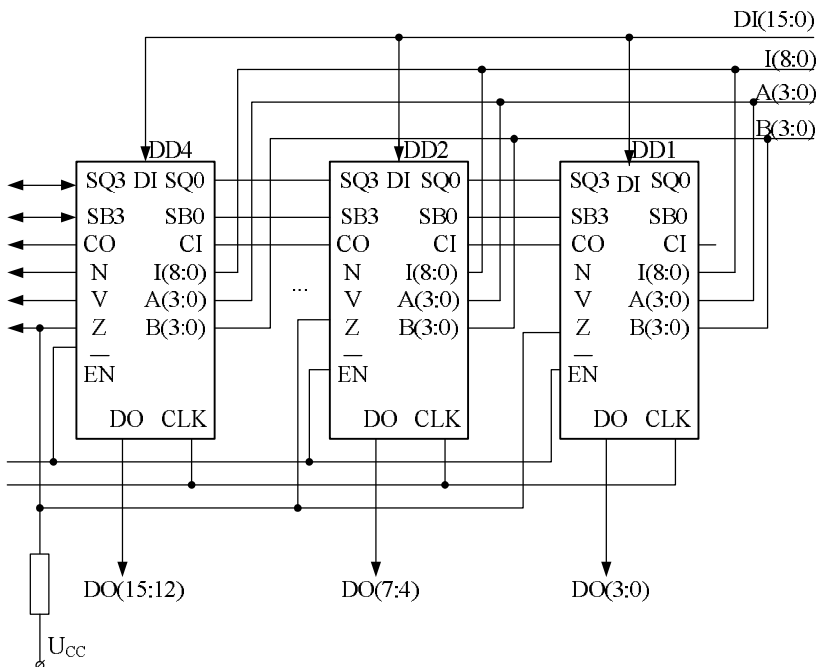


Рис. 3. Об'єднання ПЕ VC1 при послідовному поширенні переносу

Дані в блок поступають із шини DI (15:0).

Результат перетворення видається на шину даних DO (15:0).

Керування операційним блоком процесора відбувається за допомогою мікрокоманди:

I(8:0)	B(3:0)	A(3:0)	\overline{EN}	SQ15	SQ0	SB15	SBO	CI
--------	--------	--------	-----------------	------	-----	------	-----	----

3. Використання БОД. Розробити мікропрограму обчислення виразу $f=a/2+(b-c)/4$ для 16-розрядного процесора. Дані a , b та c подані в доповнювальному коді. Варіант мікроалгоритму показаний на рис. 4.

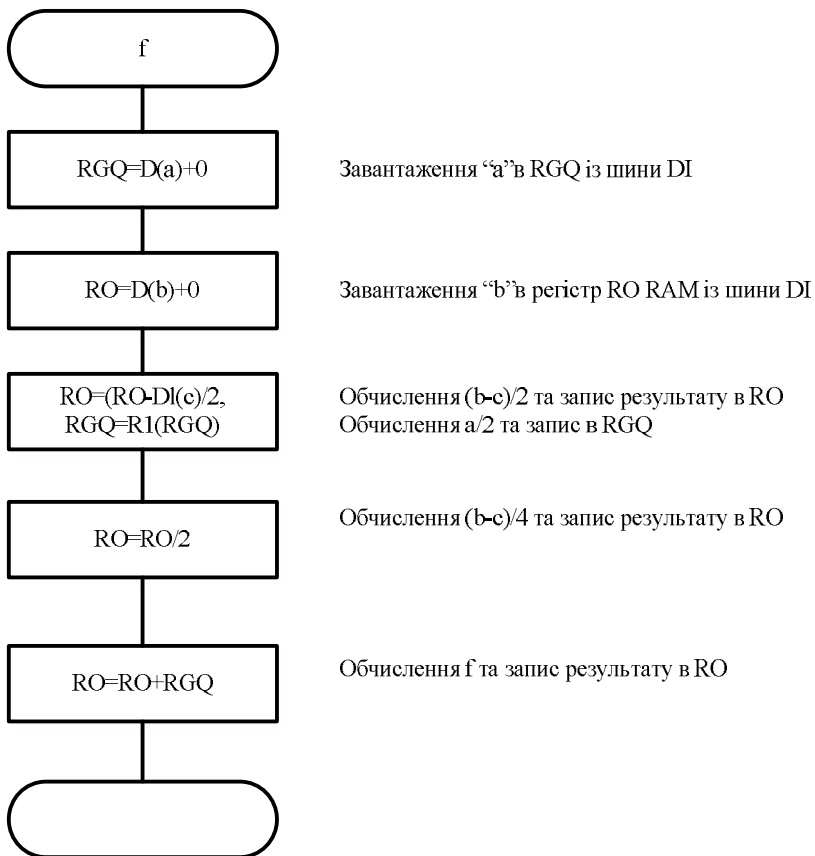


Рис. 4. Мікроалгоритм обчислення $f = a/2 + (b-c)/4$

Розглянемо реалізацію мікрокоманди (блок 3).

Для виразу $R0 = (R0 - DI(c))/2$ джерелами даних на входах S та R АЛП (див. рис. 1) є регістр R0 RAM та вхідна шина DI з операндом c (DI(c)). Відповідно до табл. 1 визначимо значення мікрокоду $1(2:0) = 101 = 5$. Читання з RAM робиться через порт В. В АЛП повинна виконуватись операція S-R, якій відповідає мікрокод $I(6:3) = 001 = 1$ (табл. 2). При цьому $CI = 1$. Результат з виходу F АЛП записується в RAM за адресою $V = 00$ зі зсувом на один розряд

вправо. Одночасно зсувається вправо на один розряд також вміст регістра RGQ. Цим діям відповідає мікрокод $I(8:6)=100=4$ (див. табл. 3). При арифметичному зсуві вправо старший розряд, який звільняється, заповнюється знаком числа F(15). Для реалізації блоку 3 обов'язковими є такі поля МК: $I=415$, $B=00$, $SQ15=0$, $SB15=F(15)$, $CI=1$.

Послідовність МК, яка відповідає мікропроалгоритму (рис. 4), наведена в табл. 4. Знаком "x" позначені довільні значення коду. Мікрокоманди 3 та 5 відповідно реалізують операції віднімання та додавання.

Таблиця 4

Мікропрограма обчислення $f=a/2+(b-c)/4$

МК	Оператори	Мікрокоманда								
		I(8:0)	B(3:0)	A(3:0)	\overline{EN}	SQ15	SQ0	SB 15	SB0	CI
1	RGQ=DI(a)+0	037	XX	XX	1	X	X	X	X	X
2	RO=DI(b)+()	337	00	XX	1	X	X	X	X	X
3	RO=(RO-DI(c))/2, RGQ =R1 (RGQ)	415	00	00	1	0	X	F(15)	-	1
4	R0=(R0+0)/2	533	00	XX	1	X	X	F(15)	X	X
5	RO=RO+RGQ	300	00	00	0	X	X	X	X	0

Обладнання, прилади та матеріали

В ЛР моделюється ВІС ВС1 з використанням ПП «Емулятор К1804 ВС1» []. Для проведення ЛР необхідний клас ПК.

Заходи безпеки при виконанні ЛР

Заходи безпеки відповідають вимогам, яким необхідно дотримуватись при роботі в комп'ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання, основні теоретичні положення та розв'язки завдання.

2. Результати домашнього завдання необхідно представити викладачу.

3. Варіанти завдань наведені в табл. 5, кожне з яких складається з двох частин:

– в першій частині розробляють принципову схему БОД розрядності n . Для запам'ятовування ознак Z , V , CO , N . В використовується регістр стану RGC .

– в другій частині розробляють алгоритми та мікропрограми функцій (табл. 6). Якщо в табл. 5 джерелами операндів виступають регістри RAM та Q , то спочатку їх необхідно завантажити. Для цього використовують мікрокоманди $RAM[i]=DI(x)$ та $Q=DI(X)$. В колонці 3 запис $5(x3)$ означає: 5 - адреса регістра RAM ; $x3$ - змінна, яка розміщена в $RAM[5]$. В колонці 4 символами 0,1 позначені відповідні константи. В 6-й колонці вказані адреси регістрів RAM . Знаком "+" в колонках 7, 8 відмічені приймачі результату обчислення функції. Знаком Z позначений 3- й стан шини DO .

2. Дані для ПП "Емулятор K1804BC1" готують у вигляді МП відповідно до (табл. 7).

Таблиця 7

Вихідні дані для ПП «Емулятор К 1804BC1»

Номер МК	МК									Дані
	instr	RB	RA	Ш	SQ15	SQ0	SB15	SBO	CI	
1	037	XX	XX	0	X	X	X	X	X	000004 ₍₈₎
2	337	00 ₍₈₎	XX	0	X	X	X	X	X	000002 ₍₈₎
3	034	01 ₍₈₎	00 ₍₈₎	0	-	0	-	0	0	xxxxxx
4	301	00 ₍₈₎	01 ₍₈₎	0	X	X	X	X	0	xxxxxx
5	010	XX	00 ₍₈₎	0	X	X	X	X	1	xxxxxx
6	537	00 ₍₈₎	XX	0	X	X	0	-	X	000010 ₍₈₎
7	300	00 ₍₈₎	00 ₍₈₎	0	X	X	X	X	0	xxxxxx

Таблиця 5

Варіанти завдань

Номер варіанта	Функція	Джерела операндів			Приймачі результату			Значення операндів	БО Д n
		RAM [i]	DI	RQ	RAM(i)	RQ	DO		
1	2	3	4	5	6	7	8	9	10
1	1	5(x3),0(x2)	1		15	+	+	x3=1, x2=5	8
2	3	1(x1), 5(x2)			15		+	x1=1, x2=6	12
3	8	0(x2),1(x3)	x1				+	x2=6, x3=0, x1=2	12
4	3	6(x2)	x1	x1	7		+	x2=4, x=3	16
5	4	2(x2)		x1		+	+	i =10 • x2=6, x1=1	16
6	1	5(x2)		x1	6		+	x2=2, x1=3	16
7	3	4(x2)	x1	x1	12	+	Z	x2=2, x1=6	12
8	7	5(x2)		x1	6		+	x2=2, x1=3	12
9	5	2(x2)	0	x1	7	+	+	x2=1, x3=5, x1=4	8
10	8	3(x2),0(x1)		x3	7	+		x1=2, x3=4, x2=1	8
11	9	7(x3),1(x1)	x2		12	+	+	x3=3, x1=5, x2=1	16
12	7	2(x2),7(x1)		x3	10		+	x2=5, x1=-7, x3=9	16

Таблиця 9

Множина функцій

Номер функції	Функція
1	$f=x3 \vee x2+1$
2	$f= x2 \vee x1+1$
3	$f=x2/2+2x1$
4	Читання регістрів RGQ, RAM [i]
5	Запис "0" в RGQ
6	$f=(x2+1)/2$
7	$f=(x2 \wedge x1)+x3$
8	
9	$f=(x3-x1+x2)*2$

3. ПП "Емулятор K1804BC1" призначена для контролю та налагодження МП (мікрокоманд) керування БОД.

Максимальний розмір ПП - 99 МК (рядків).

Інтерфейс ПП виконаний у вигляді меню режимів та сукупності підменю кожного із режимів.

Вибір режиму в меню здійснюють натисненням клавіш "Влево", "Вправо", вхід в режим - за допомогою клавіші "Ввод".

Вибір режиму в підменю виконують за допомогою клавіш "Вниз", "Вверх", а вхід в режим - клавішею "Ввод".

3.1. Режим "Редактор". Забезпечує створення та редагування МК і має такі підрежими:

"Створення" - МП завантажується в ПК;

"Редагування" - редагування мікрокоманд МП.

При завантаженні МП необхідно враховувати: стани "0" та "1" вказуються відповідно цифрами 0 та 1; довільний стан позначається знаком "x"; вихід позначається знаком, числові дані та адреси А та В представляють у 8-й системі обчислення.

В цьому режимі використовують такі клавіші:

"Влево" - курсор переміщується вліво;

"Вправо" - курсор переміщується вправо;

"Вверх" - курсор переміщується вгору;

"Вниз" - курсор переміщується донизу;

"Home" - курсор переміщується до початку рядка;

"End" - курсор переміщується до кінця рядка;

"Pg/Down" - курсор переміщується на сторінку донизу;

"Pg/Up" - курсор переміщується на сторінку вгору;

"0,...,7,x" - набір коду адреси та значень логічних сигналів;

- знищення рядка мікрокоманди;

"Space" - переміщення курсору вправо;

"F4" - контроль помилок, що обов'язковий для режиму

"Обчислення": "F1" - довідкова інформація.

3.2. Режим "Обчислення". Забезпечує автоматичне або покрокове виконання МП і має підрежими:

"Результат" - автоматичне виконання МП;

"Крок" - покрокове виконання МП.

Числова інформація представлена у 8-й системі обчислення, номери регістрів RAM - в 10-й системі обчислення.

3.3. Режим "Допомога" видає довідкову інформацію.

3.4. Режим "Вихід" забезпечує вихід в середовище ОС.

1. Приклад автоматичної обробки МП (режим "Обчислення" та підрежими "Результат", "Друк") приведений на рис. 5.

СОЗП	R0 000008	R4 000000	R8 000000	R12 000000
	R1 000004	R5 000000	R9 000000	R13 000000
	R2 000000	R6 000000	R10 000000	R14 000000
	R3 000000	R7 000000	R11 000000	R15 000000
	DO 000004 CO 0 V 0 SQ15 0 SQO 0	DO,		
RO 000004 F15 0 Z 0 SB15 0 SBO 0	ознаки			

Рис.5. Стан 4 регістрів RAM, шини DO після виконання мікропрограми.

Оформлення результатів експерименту

Результатами виконання ЛР є принципіальна схема БОД, мікропрограми задач завдання.

Обробка експериментальних даних

Результати одержані в ЛР порівнюються із результатами ПП «Емулятор ВС1».

Аналіз одержаних результатів

Формулюються висновки по результатам виконання ЛР.

Запитання для самоперевірки

1. Назвіть основні блоки ПЕ К1804ВС1.
2. В якому коді представляються дані?
3. Назвіть джерела операндів та приймачів результату.
4. Назвіть операції, які виконуються за один такт.
5. Які ознаки формуються в ПЕ ВС1?
6. Назвіть можливі призначення регістра Q.
7. Яка чисельність регістрів RAM?
8. Призначення вузла ShL1?

Джерела

[2].

ТЕМА 6. ПАРАЛЕЛЬНІ ОБЧИСЛЮВАЛЬНІ СИСТЕМИ

Лабораторна робота

КЛАСТЕРНА ОБЧИСЛЮВАЛЬНА СИСТЕМА

Мета роботи. Закріплення теоретичних знань по синтезу структур кластерних обчислювальних систем (ОС).

Основні завдання роботи

- 1.Отримати варіант домашнього завдання.
- 2.Вивчити теоретичну частину роботи.
- 3.Представити в лабораторному зошиті результати виконання домашнього завдання.

Основні теоретичні відомості

1. Архітектури ОС. Для паралельних обчислювальних систем (ОС) важливим є спосіб керування процесорами та способи їх взаємодії.

По способу керування процесорами ОС бувають типів SIMD (один потік команд, багато потоків даних) та MIMD (багато потоків команд, багато потоків даних) (рис. 1).

В SIMD ОС (рис. 1, а.) процесори P_i знаходяться під керуванням HOST-процесора (контролера) і в кожний момент часу виконують одну команду або знаходяться в стані очікування.

В MIMD ОС (рис 1, б.) процесори функціонують під керуванням власних програм. В таких ОС виникає проблема синхронізації виконуваних процесорами завдань у відповідному порядку та з правильними даними.

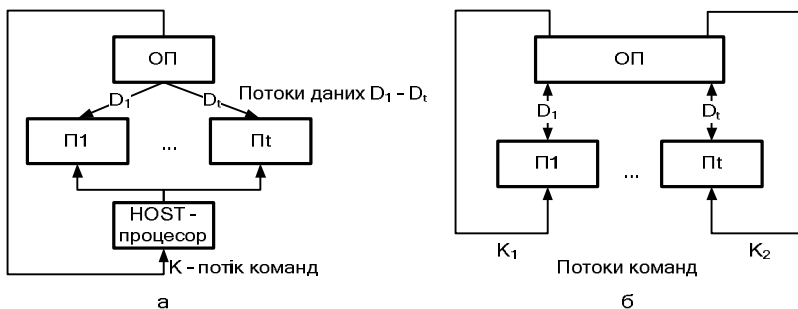


Рис. 1. Архітектури ОС: а – SIMD, б – MIMD

2. Топології ОС. В паралельних ОС важливим є спосіб з'єднань окремих процесорів між собою.

2.1. Повнозв'язана ОС. Процесори Π_i мають безпосередні зв'язки між собою (рис. 2).

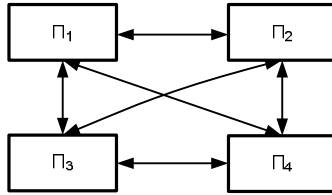


Рис. 2. Повнозв'язана ОС

2.2. ОС з комутатором. Доступ процесорів Π_i до модулів пам'яті M_j забезпечують перемикачі, що входять до складу комутатора (рис. 3.).

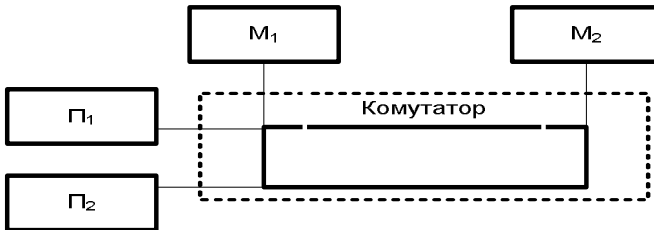


Рис. 3. ОС з комутатором

2.3. ОС із загальною шиною. Усі процесори з'єднані за допомогою швидкісної загальної шини (ЗШ) (рис. 4.)

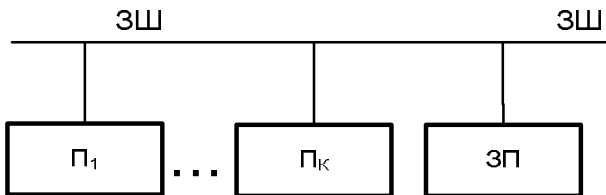


Рис. 4. ОС із ЗШ

2.5. Решітка процесорів. В цій топології кожен процесор з'єднується з декількома сусідніми (рис. 5).

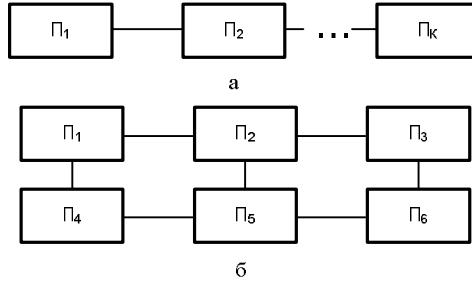


Рис. 5. Решітка процесорів: а – лінійний масив;
б – площинний масив (решітка)

2.5. Гіперкуб. На рис. 6 зображений тривимірний куб, у вершинах якого розміщені процесори Π_i , а ребра виконують функцію локальних зв'язків між ними.

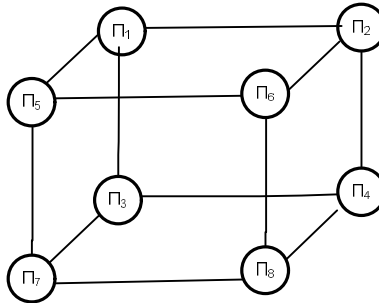


Рис. 6. Гіперкуб

3.6. Кластери. Кластерна ОС (рис. 7) складається із «р» кластерів (пучків процесорів), які об'єднані шиною. Кластер, в свою чергу, об'єднує «т» процесорів. З'єднання в кластері називаються локальними, а між кластерами – глобальними. Функціонування ОС базується на поєднанні цих двох типів взаємозв'язків. Для кожного кластера домінуючими є обміни в межах відповідного кластера. Обміни між кластерами виникають значно рідше.

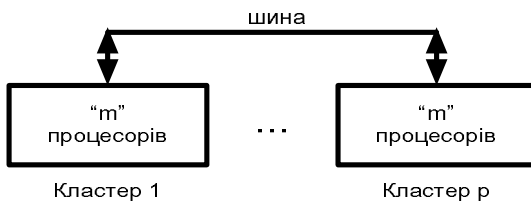


Рис. 7. Кластерна ОС

Кластерна ОС може бути реалізована різними способами. В межах кластера зв'язки процесорів можна організувати за допомогою шини (швидкісна мережа), решітки, гіперкуба і т. д. Названі схеми можуть бути використані і для з'єднання кластерів.

3.Мінімізація міжкластерних обмінів. Для кластерної ОС актуальним є мінімізація кластерних обмінів. Розв'язання цієї проблеми можна звести до пошуку мінімального розрізу графа задачі. Граф задачі G розрізається на підграфи (шматки) G_i , що відповідають підзадачам, їх обчислення виконують кластери. Для пошуку мінімального розрізу графа використовуємо один із послідовних алгоритмів [2]. Зробимо деякі визначення.

Граф G (без петель), у якого існує мінімум одна пара вершин, з'єднаних m ребрами називають мультиграфом, а максимальне m – мультичислом графа G .

Ребра, що з'єднують одну і ту ж пару вершин називають кратними. Дві вершини графа називають суміжними, якщо існує ребро, що їх з'єднує. Якщо вершина є кінцем ребра (дуги), то вони - інцидентні.

Число ребер, що інцидентні вершині a_i називають локальною ступінню $d(a_i)$ цієї вершини.

Кожному графу G відповідає квадратна матриця суміжності D , елементи якої визначаються так:

$$d_{ij} = \begin{cases} m, \text{ якщо вершини } a_j, a_i - \text{ суміжні,} \\ 0, \text{ якщо вершини } a_j, a_i - \text{ несуміжні} \end{cases}$$

Критерій розрізу графа G є мінімальна кількість ребер між шматками G_i . Підграф у шматку має максимальну кількість ребер.

Деякі вершини графа G можуть попередньо бути закріпленими за шматками G_i . Тому маємо два варіанти пошуку мінімального розрізу графа: закріплені вершини $\epsilon (Q \neq \emptyset)$; закріплені вершини відсутні ($Q = \emptyset$).

3.1. Пошук мінімального розрізу графа при $Q \neq \emptyset$

Маємо граф $G=(A, U)$ (A - множина вершин, U – множина дуг) та множину закріплених за шматками графа вершин $Q=\{a_\epsilon, a_v, \dots\}$. Необхідно розрізати граф G на L шматків

1. Формування першого шматка $G_1=\{A_1, U_1\}$ починаємо з вершини a_ϵ , яку апріорно вважаємо елементом множини A_1 .

2. Для визначення другої вершини шматка G_1 будуємо множину вершин Γ_{a_ϵ} суміжних вершині a_ϵ .

3. Визначаємо відносні ваги $p(a_i)$, що входять у множину Γ_{a_ϵ}

$$p(a_i) = \delta(a_i) - d_i, \text{ де}$$

$\delta(a_i)$ локальний ступінь вершини a_i , d_i – число дуг, що з'єднують вершину a_i з вершинами графа G (в нашому випадку a_ϵ)

4. Відповідно критерію розрізу графа із множини Γ_{a_ϵ} вибираємо вершину a_i з мінімальним значенням $p(a_i)$. Це буде друга вершина шматка G_1 , тобто $A_1 = \{a_\epsilon, a_i\}$. Якщо в Γ_{a_i} є декілька вершин з однаковими вагами $h(a_i)$, то вибирається вершина з більшим $\delta(a_i)$.

5. Визначаємо множину вершин $\Gamma_{a_\epsilon} \cup \Gamma_{a_i}$. Для кожної вершини $a_k \in \Gamma_{a_\epsilon} \cup \Gamma_{a_i}$ визначаємо відносну вагу $h(a_k)$. Мінімальне значення ваги $h(a_k)$ матиме наступна вершина шматка G_1 $A_1 = \{a_\epsilon, a_i, a_k\}$.

Процес продовжується до тих пір поки в G_1 не буде визначено задане число вершин.

Отриманий шматок G_1 видаляється з граф G . Аналогічно формується шматок G_2 . Першим його елементом буде вершина $a_v \in Q$.

3.2. Пошук мінімального розрізу графа при $Q = \emptyset$

Першими вершинами шматків G_i будуть вершини із максимальним числом інцидентних їм кратних дуг, тобто $m(a_i)$.

Приклад 1. Граф задачі (рис. 11) необхідно розрізати на три шматки (підзадачі) G_1, G_2, G_3 по чотири вершини в кожному. Закріплені вершини $Q = \{a_1, a_{11}, a_{12}\}$.

1. Формування шматків G_i графа G починаємо з розподілу закріплених вершин

$$G_1 = \{a_1, \dots\}, G_2 = \{a_{11}, \dots\}, G_3 = \{a_{12}, \dots\}$$

2. Шматок $G_1 = \{a_1, \dots\}$.

2.1. По матриці D (рис. 2) визначимо множину вершин Γ_{a_1} суміжних a_1

$$\Gamma_{a_1} = \{a_2, a_4, a_6, a_7\}.$$

Вершина a_{12} не враховується, тому що вона входить до шматка G_3 .

2.2. Значення відносних ваг вершин множини Γ_{a_1}

$$h(a_2) = \delta(a_2) - d(a_1) = 3 - 1 = 2,$$

$$h(a_4) = \delta(a_4) - d(a_1) = 4 - 1 = 3,$$

$$h(a_6) = \delta(a_6) - d(a_1) = 3 - 2 = 1,$$

$$h(a_7) = \delta(a_7) - d(a_1) = 7 - 3 = 4,$$

$$h(a_i)_{\min} = h(a_6) = 1.$$

Вершина a_6 призначається шматку

$$G_1 = \{a_1, a_6\}.$$

2.3. Визначаємо множину вершин

$$\Gamma_{a_1} \cup \Gamma_{a_6} = \{a_2, a_4, a_6, a_7\} \cup \{a_1, a_7\} = \{a_2, a_4, a_7\}$$

2.4. Значення відносних ваг вершин $\Gamma_{a_1}, \Gamma_{a_6}$

$$h(a_2) = \delta(a_2) - d(a_1) - d(a_6) = 3 - 1 - 0 = 2,$$

$$h(a_4) = \delta(a_4) - d(a_1) - d(a_6) = 4 - 1 - 0 = 3,$$

$$h(a_7) = \delta(a_7) - d(a_1) - d(a_6) = 7 - 3 - 1 = 3;$$

$$h(a_i)_{\min} = h(a_2) = 2.$$

В шматок G_1 включається вершина a_2 . $G_1 = \{a_1, a_6, a_2\}$.

2.5. Визначаємо множину вершин

$$\Gamma_{a_1} \cup \Gamma_{a_6} \cup \Gamma_{a_2} = \{\cancel{a_2}, a_4, \cancel{a_6}, a_7\} \cup \{\cancel{a_4}, a_7\} \cup \{\cancel{a_7}, a_3, a_7\} \\ = \{a_4, a_7, a_3\}.$$

Закреслені вершини входять у G_1 .

2.6. Визначення відносних ваг вершин a_4, a_7, a_3

$$h(a_4) = \delta(a_4) - d(a_1) - d(a_6) - d(a_2) = 4 - 1 - 0 - 0 = 3,$$

$$h(a_7) = \delta(a_7) - d(a_1) - d(a_6) - d(a_2) = 7 - 3 - 1 - 1 = 2;$$

$$h(a_3) = \delta(a_3) - d(a_1) - d(a_6) - d(a_2) = 4 - 0 - 0 - 1 = 3,$$

$$h(a_i)_{\min} = h(a_7) = 2.$$

В шматок G_1 включається остання вершина a_7

$$G_1 = \{a_1, a_6, a_2, a_7\}.$$

3. Шматок $G_1 = \{a_{11}, \dots\}$

3.1. Із матриці суміжності D визначаємо множину вершин

$$\Gamma a_{11} = \{a_5, a_{10}\}.$$

3.2. Значення відносних ваг вершин Γa_{11}

$$h(a_5) = \delta(a_5) - d(a_{11}) = 6 - 1 = 5,$$

$$h(a_{10}) = \delta(a_{10}) - d(a_{11}) = 7 - 2 = 5.$$

В шматок G_2 включаємо вершину a_{10} , тому що $\delta(a_{10}) > \delta(a_5)$.

$$G_2 = \{a_{11}, a_{10}, \dots\}.$$

3.3. Визначаємо із матриці D множину вершин

$$\Gamma a_{11} \cup \Gamma a_{10} = \{a_5, a_7, a_{10}\} \cup \{a_3, a_5, a_9, a_{11}\} = \{a_5, a_3, a_9\}.$$

3.4. Значення відносних ваг вершин $\Gamma a_{11} \cup \Gamma a_{10}$

$$h(a_5) = \delta(a_5) - d(a_{11}) - d(a_{10}) = 6 - 1 - 3 = 2,$$

$$h(a_3) = \delta(a_3) - d(a_{11}) - d(a_{10}) = 4 - 0 - 1 = 3,$$

$$h(a_9) = \delta(a_9) - d(a_{11}) - d(a_{10}) = 5 - 0 - 1 = 4.$$

В шматок G_2 включаємо вершину a_5 .

$$G_2 = \{a_{11}, a_{10}, a_5, \dots\}.$$

3.5. Визначаємо із матриці D множину вершин

$$\Gamma a_{11} \cup \Gamma a_{10} \cup \Gamma a_5 = \{\cancel{a_5}, \cancel{a_7}, \cancel{a_{10}}\} \cup \{a_3, \cancel{a_5}, a_9, a_{11}\} \cup \{a_3, a_{10}, \cancel{a_{11}}\} = \{a_3, a_9\}.$$

3.6. Значення відносних ваг вершин a_3, a_9

$$h(a_3) = \delta(a_3) - d(a_{11}) - d(a_{10}) - d(a_5) = 4 - 0 - 1 - 2 = 1,$$

$$h(a_9) = \delta(a_9) - d(a_{11}) - d(a_{10}) - d(a_5) = 5 - 0 - 1 - 0 = 4.$$

В шматок G_2 включаємо останню вершину a_3

$$G_2 = \{a_{11}, a_{10}, a_5, a_3\}.$$

4. Вершини графа G , що залишились, складають шматок

$$G_3 = \{a_4, a_8, a_9, a_{12}\}.$$

Приклад 2. Знайти мінімальний розріз графа G (рис. 1) при умовах як і в прикладі 1, але $Q = \emptyset$.

1. Шматок G_1

Першою вершиною G_1 вибираємо вершину з найбільшим значенням $m(a_i)$ (рис.2), наприклад, a_i

$$G_1 = \{a_1, \dots\}.$$

1.2. Будуємо множину $\Gamma a_1 = \{ a_2, a_4, a_6, a_7, a_{12} \}$

та обчислюємо їх відносні ваги

$$h(a_2) = \delta(a_2) - d(a_1) = 3 - 0 = 3,$$

$$h(a_4) = \delta(a_4) - d(a_1) = 4 - 1 = 3,$$

$$h(a_6) = \delta(a_6) - d(a_1) = 3 - 2 = 1,$$

$$h(a_7) = \delta(a_7) - d(a_1) = 7 - 3 = 4,$$

$$h(a_{12}) = \delta(a_{12}) - d(a_1) = 6 - 1 = 5.$$

Вибираємо вершину $h(a_i)_{\min} = h(a_6) = 1$

$$G_1 = \{ a_1, a_6, \dots \}.$$

1.3. Будуємо множину $\Gamma a_1 \cup \Gamma a_6 = \{ a_2, a_4, a_7, a_{12} \} \cup \{ a_7 \} = \{ a_2, a_4, a_7, a_{12} \}$ та обчислюємо відносні ваги вершин

$$h(a_2) = \delta(a_2) - d(a_1) - d(a_6) = 3 - 1 - 0 = 2,$$

$$h(a_4) = \delta(a_4) - d(a_1) - d(a_6) = 4 - 1 - 0 = 3,$$

$$h(a_7) = \delta(a_7) - d(a_1) - d(a_6) = 7 - 3 - 1 = 3,$$

$$h(a_{12}) = \delta(a_{12}) - d(a_1) - d(a_6) = 6 - 1 - 0 = 5.$$

Вибираємо вершину $h(a_2) = 2$

$$G_1 = \{ a_1, a_6, a_2, \dots \}.$$

1.4. Будуємо множину $\Gamma a_1 \cup \Gamma a_6 \cup \Gamma a_2 = \{ a_2, a_4, a_6, a_7, a_{12} \} \cup \{ a_1, a_7 \} \cup \{ a_1, a_3, a_7 \} = \{ a_4, a_7, a_3, a_{12} \}$

та обчислюємо відносні ваги вершин

$$h(a_4) = \delta(a_4) - d(a_1) - d(a_6) - d(a_2) = 4 - 1 - 0 - 0 = 3,$$

$$h(a_7) = \delta(a_7) - d(a_1) - d(a_6) - d(a_2) = 7 - 3 - 1 - 1 = 2,$$

$$h(a_3) = \delta(a_3) - m(a_1) - d(a_6) - d(a_2) = 4 - 0 - 0 - 1 = 3,$$

$$h(a_{12}) = \delta(a_{12}) - m(a_1) - m(a_6) - m(a_2) = 6 - 1 - 0 = 5.$$

Вибираємо вершину з $n(a_i)_{\min} = n(a_7) = 2$. Це остання вершина шматка G_1

$$G_1 = \{ a_1, a_6, a_2, a_7 \}.$$

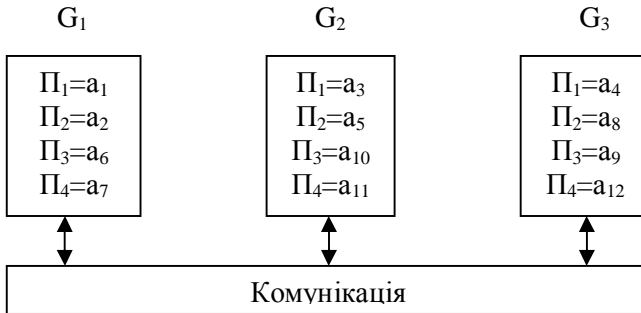
Першою вершиною шматка G_2 вибираємо вершину за максимальним $m(a_i)$ та $\delta(a_i)$. Це буде вершина a_{10} .

$$G_2 = \{ a_{10}, \dots \}.$$

Визначення шматків G_2, G_3 будуть аналогічними.

4. Кластерні ОС

Загальна структура кластерної ОС представлена на рис. 8.



ОС складається з трьох кластерів $G_1 - G_3$, які об'єднані через комунікаційне середовище. Кожний кластер об'єднує чотири процесора P_1-P_4 , які ототожнюються з підзадачами (потокaми) a_i . Наприклад для G_1 це будуть (a_1, a_2, a_6, a_7) .

4.1. Структури кластерів

Взаємодії процесорів P_1-P_4 кластера G_1 можуть забезпечувати такі топології: повноз'язана; комутатор ЗП (загальна пам'ять); решітка (лінійка); загальна пам'ять; комутатор ЛП (локальна пам'ять). Структури кластерів представлені на рис. 9.

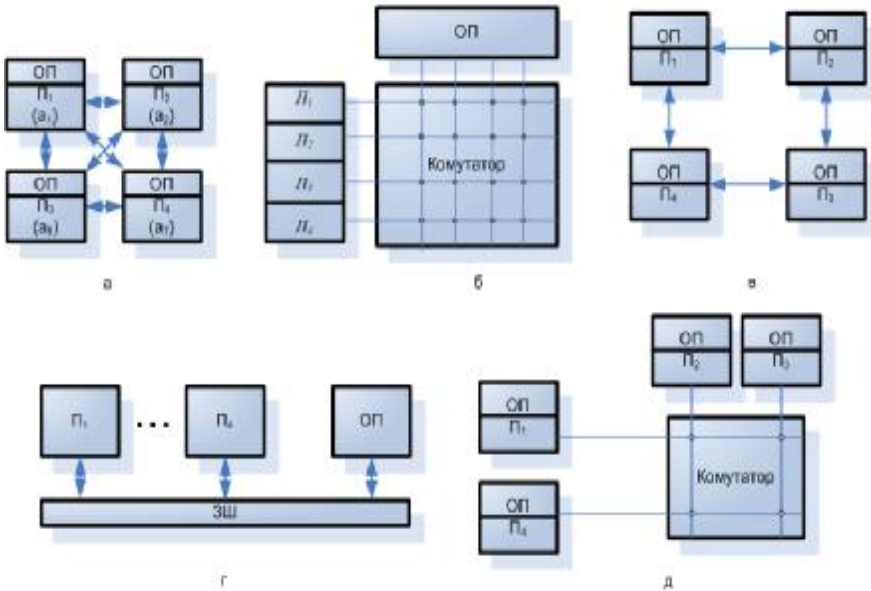


Рис. 9. Структури кластерів: а - повнозв'язана; б – комутатор ЗП; в – решітка (лінійка); г – ЗШ; д – комутатор ЛП.

4.2. Структури кластерних ОС

Взаємозв'язки кластерів в кластерній ОС (рис. 10) можуть бути подібними до зв'язків процесорів у кластері: повнозв'язними; через комутатор; решітка; швидкодіюча мережа (загальна шина).

В ОС (рис. 10.а) взаємодію кластерів забезпечує швидкодіюча мережа (Ffst Ethernet, Muginet та ін.) На (рис. 10.б) об'єднання кластерів виконує комутатор. ОС на (рис. 10.в) реалізує топологію решітка. В усіх ОС сервер забезпечує інтерфейс користувачів з системою через інтерфейсні адаптери А.

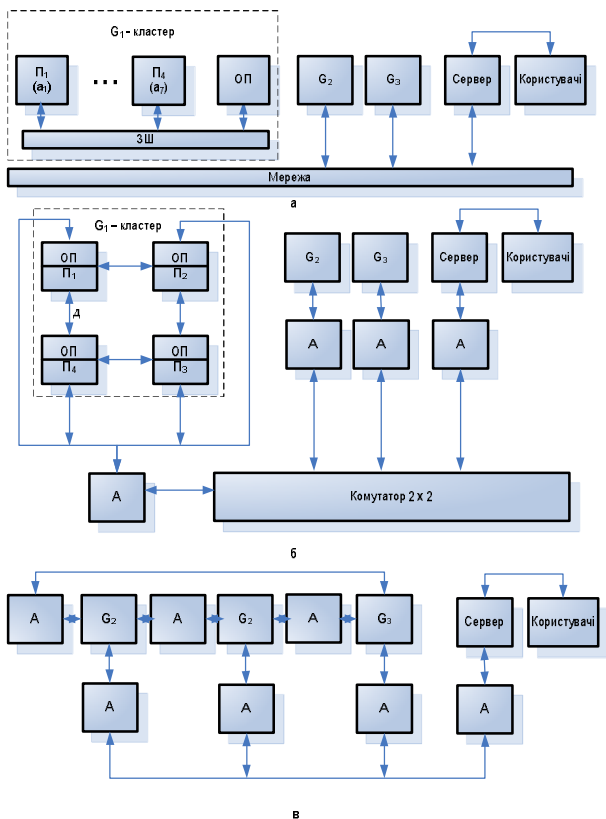


Рис. 10. Структури кластерних ОС: а – мережна; б- комутатор; в – решітка.

Обладнання, прилади та матеріали

В ЛР розв’язується задача синтезу кластера та кластерної ОС. Для виконання роботи використовується клас ПК.

Заходи безпеки при виконанні ЛР

Заходи безпеки відповідають вимогам, яким необхідно дотримуватись при роботі в комп’ютерному класі.

Порядок виконання роботи

1. В лабораторному зошиті наводяться мета, завдання та синтез кластерної ОС.

2. Результати домашнього завдання необхідно представити викладачу.

3. В ЛР використовується ПП «Розрізання графу» для контролю відповідності мінімального розрізу графа задачі отриманого у домашньому завданні

4. В ЛР визначаються архітектури та топології паралельних ОС у відповідності до класифікації Фліна, способи використання пам'яті.

5. Розробка структур кластера та кластерної ОС складається із послідовності кроків:

- На основі загального графа задачі G та матриці сумісності D формується граф G' та матриця D' варіанта завдання ЛР;
- визначаються шматки G_1, G_2, \dots та будується мінімальний розріз графа G' . При чому враховуються наявність або відсутність заборонених вершин;
- з урахуванням топології кластера, числа та розмірів шматків G_1, G_2, \dots розробляється структура кластера.

6. Варіанти завдань представлені в табл. 1. Вони містять кількісні параметри, що необхідні для побудови кластерної ОС.

Таблиця 1

Варіанти завдань

№ варіанта	P	M	N	Q	Топологія кластера	Топологія ОС
1	9	3	3,4,2	∅	Решітка	Комутатор
2	10	3	4,3,3	1,9	Загальна пам'ять (ЗП)	Мережа
3	11	3	3,3,5	10	Решітка	Решітка
4	12	4	3,3,4,2	5,12	Повнозв'язана	Комутатор
5	10	3	3,3,4	∅	Комутатор ЗП	Мережа
6	11	3	5,3,3	10,11	ЗП	Решітка
7	12	4	3,3,3,3	9,7,11	Решітка	Мережа
8	12	4	5,3,3,1	12,10	Повнозв'язана	Комутатор
9	10	3	3,3,3	∅	ЗП	Решітка
10	10	3	2,5,3	6,10	Комутатор ЛП	Мережа
11	11	3	4,3,4	∅	ЗП	Решітка
12	12	3	4,4,4	6,9,11	Повнозв'язана	Мережа
13	12	4	4,3,3,2	2,4,7,6	Комутатор ЗП	Комутатор

Вихідними даними для ЛР є: матриця суміжності D' , граф задачі G' , порядок матриці P ; число шматків (підграфів) M ; число вершин в шматках N ; закріплені (заборонені) вершини Q .

На рис. 11 та Рис. 12 показані загальні граф G та відповідна йому матриця суміжності D . Вершини графа a_i є підзадачі (поток), а ребра взаємозв'язки між ними. Порядок матриці D дорівнює 12. Для отримання графа G' та матриці D' необхідно в графі G викреслити вершини $a_i \dots$ та ребра зв'язків отримаємо граф задачі, що відповідає варіанту ЛР. Тотожність матриці D' графу G' досягається видаленням відповідних стовпців та рядків. При цьому корегуються також елементи матриці D' , стовпці $\delta(a_i)$ та $m(a_i)$.

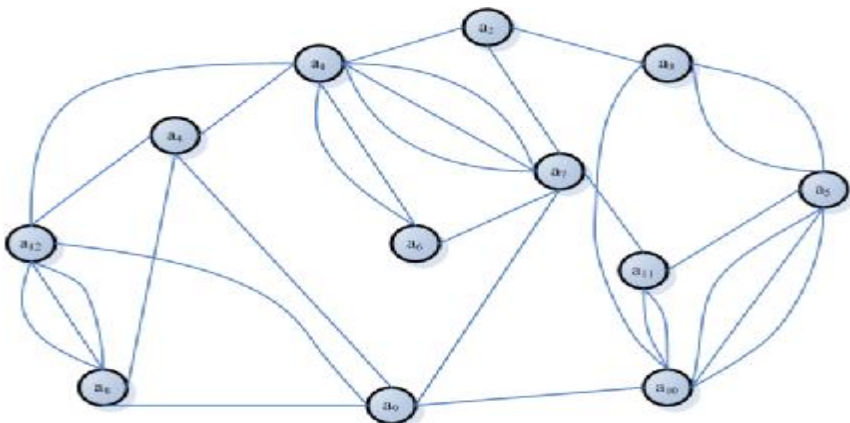


Рис. 11. Граф задачі G.

	1	2	3	4	5	6	7	8	9	10	11	12	$\delta(a_i)$	$m(a_i)$
1		1		1		2	3					1	8	3
2	1		1				1						3	1
3		1			2					1			4	2
4	1							1	1			1	4	1
5			2							3	1		6	3
6	2						1						3	2
7	3	1				1			1		1		7	3
8				1					1			3	5	3
9				1			1	1		1		1	5	1
10			1		3				1		2		7	3
11					1		1			2			4	2
12	1			1				3	1				6	3

Рис. 12. Матриця суміжності D, локальна ступінь - $\delta(a_i)$,
максимальне число кратних дуг - $m(a_i)$

Оформлення результатів експерименту

Результатом ЛР є структура кластера та кластерної ОС.

Обробка експериментальних даних

Наводяться та порівнюються результати отримані в домашньому завданні та ПП «Розрізання графу».

Аналіз одержаних результатів

Формулюються висновки по результатам виконання ЛР.

Запитання для самоперевірки

1. Як формується граф задачі в лабораторній роботі?
2. Як формується матриця суміжності?
3. Основні кроки послідовного алгоритму розрізання графа.
4. Типи архітектур паралельних ОС.
5. Способи організації структур кластерів та кластер них ОС.

Джерела

[3, 5].



СПИСОК ЛІТЕРАТУРИ

1. *Угрюмов Е. П.* Цифровая схемотехника / Е. П. Угрюмов – СПб. : БХВ – Петербург, 2001. – 528 с.
2. Микропроцессорные системы: Учебное пособие для вузов / Е. А. Александров, Р. И. Грушвицкий, М. С. Куприенов и др; Под. общ. ред. Пузанкова. – СПб. : Политехника, 2002. – 935 с.
3. *Цилькер Б. Я.* Организация ЭВМ и систем : учебник / Б. Я. Цилькер, С. А. Орлов. – СПб. : Питер, 2006. – 668 с.
4. *Єфимець В. М.* ЦЕОМ. Інтегральні мікросхеми серії КР1533. : Практичний порадник / В. М. Єфимець, ін. – К. : НАУ, 2003. – 36 с.
5. *Ортега Д.* Введение в параллельные и векторные методы решения линейных систем : пер. с англ. / Д. Ортега. – М. : МИР, 1991. – 367 с.

Навчальне видання

**КОМП'ЮТЕРНА СХЕМОТЕХНІКА
ТА АРХІТЕКТУРА КОМП'ЮТЕРІВ**

Лабораторний практикум
для студентів напряму підготовки 6.050101
«Комп'ютерні науки»

Укладачі:

ЄФИМЕЦЬ Валентин Микитович
ЖУКОВ Ігор Анатолійович
КРАСОВСЬКА Євгенія Вікторівна
КУДРЕНКО Станіслава Олексіївна
ЛУКАШЕНКО Вікторія Вікторівна