

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО



МЕТОДИЧНІ ВКАЗІВКИ  
ЩОДО ВИКОНАННЯ КУРСОВОГО ПРОЕКТУ  
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
**«АРХІТЕКТУРА КОМП'ЮТЕРІВ»**  
ДЛЯ СТУДЕНТІВ ДЕННОЇ ТА ЗАОЧНОЇ ФОРМ НАВЧАННЯ  
ЗІ СПЕЦІАЛЬНОСТІ 123 – «КОМП'ЮТЕРНА ІНЖЕНЕРІЯ»  
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

КРЕМЕНЧУК 2018

Методичні вказівки щодо виконання курсового проекту з навчальної дисципліни «Архітектура комп'ютерів» для студентів денної та заочної форм навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітнього ступеня «Бакалавр»

Укладачі: старш. викл. Ю. В. Зілінський,  
старш. викл. А. Л. Юдіна

Рецензент к.т.н., доц. В. М. Сидоренко

Кафедра комп'ютерних та інформаційних систем

Затверджено методичною радою Кременчуцького національного університету імені Михайла Остроградського

Протокол \_\_\_\_ від « \_\_\_\_ » \_\_\_\_\_ 2018 р.

Голова методичної ради

проф. В. В. Костін

## ЗМІСТ

Вступ.....	4
1 Мета і задачі проекту .....	5
2 Рекомендації щодо написання та оформлення курсового проекту .....	5
3 Завдання на курсовий проект .....	7
4 Рекомендації щодо виконання проекту.....	12
5 Захист курсових проєктів.....	26
6 Критерії оцінювання курсового проекту .....	26
Список літератури.....	28
Додаток.....	29

## ВСТУП

Дисципліна «Архітектура комп'ютерів» є складовою кваліфікаційних вимог для бакалаврів спеціальності 123 – «Комп'ютерна інженерія», узагальненим об'єктом діяльності яких є технічні засоби та системне програмне забезпечення комп'ютерних систем і мереж універсального та спеціального призначення і їх компонент.

Однією з типових задач подальшої професійної діяльності бакалаврів зі спеціальності «Комп'ютерна інженерія» є розробка проектів (на рівні окремих блоків і вузлів) універсальних і спеціалізованих ЕОМ, мереж ЕОМ, обчислювальних комплексів, систем, спеціалізованих мікропроцесорних пристроїв, контролерів та адаптерів.

Запропонована в методичних вказівках тематика проекту безпосередньо пов'язана з розробкою елементів спеціалізованих мікропроцесорних пристроїв. Розглянуті при цьому питання охоплюють визначення конфігурації ПК, вибір структурної організації арифметично-логічного пристрою спеціалізованого мікропроцесора, складання мікроалгоритмів виконання операцій в арифметично-логічному пристрої, розробку мікропрограм виконання команд, складання функціональних і принципівих схем окремих пристроїв процесора, розробку системи команд процесора.

## **1 МЕТА І ЗАДАЧІ ПРОЕКТУ**

Курсовий проект з навчальної дисципліни «Архітектура комп'ютерів» повинен закріпити основні теоретичні положення, одержувані студентами з лекційного матеріалу та зі спеціальної літератури. Метою такої роботи є набуття практичних навичок у використанні отриманих знань для розробки структури комп'ютера, навичок проектування функціональних і принципових схем пристроїв комп'ютера, розробки системи команд, структур даних, способів адресації, мікроалгоритмів і мікропрограм реалізації різних операцій, а також ознайомлення із засобами автоматизації проектування обчислювальних засобів.

У результаті виконання проекту студенти повинні одержати чітке уявлення про конфігурацію та склад основних блоків ПК, про взаємодію основних вузлів і блоків комп'ютера під час обробки інформації та навчитися використовувати апарат, методи й засоби проектування елементів цифрової обчислювальної техніки. Цій меті якнайкраще відповідає самостійне виконання студентом розробки структури, алгоритмічного опису, схем і конструкції елементів спеціалізованого процесора комп'ютера.

## **2 РЕКОМЕНДАЦІЇ ЩОДО НАПИСАННЯ ТА ОФОРМЛЕННЯ КУРСОВОГО ПРОЕКТУ**

Під час виконання курсового проекту відповідно до варіанта завдання студент повинен:

- 1) визначити конфігурацію власного ПК, характеристики та параметри основних блоків;
- 2) визначити характеристики комп'ютера із заданими параметрами;
- 3) розробити систему команд процесора із заданими параметрами;
- 4) розробити структурну схему арифметично-логічного блока (АЛБ) арифметично-логічного пристрою (АЛП) на різних рівнях опису;

- 5) розробити мікроалгоритми виконання в АЛП заданих операцій на різних рівнях опису;
- 6) розробити мікропрограми для реалізації мікроалгоритмів;
- 7) розробити принципову схему АЛБ.

Курсовий проект оформляється у вигляді пояснювальної записки та графічної частини. Пояснювальна записка повинна містити такі елементи:

1. Титульна сторінка, на якій вказуються назва роботи, прізвище та номер групи студента.
2. Завдання на курсовий проект відповідно з індивідуальним варіантом.
3. Розділ 1 «Конфігурація ПК», у якому необхідно подати перелік основного обладнання ПК, на якому виконується курсова робота, із зазначенням основних технічних характеристик.
4. Розділ 2 «Процесор», у якому мають бути подані розрахунки показників комп'ютера згідно з третім завданням роботи.
5. Розділ 3 «Система команд процесора», повинен містити опис структури і форматів команд згідно з другим завданням роботи.
6. Розділ 4 «Арифметично-логічний пристрій» може складатися з двох підрозділів «Арифметично-логічний блок» і «Блок керування», у яких повинні бути подані результати виконання четвертого завдання роботи, а також опис загальної структури АЛП, опис призначення й особливості організації вузлів вибраного варіанта структури АЛБ, список реалізованих в АЛБ мікрооперацій, опис структури блока мікропрограмного керування та призначення його вузлів, опис формату мікрокоманд і коди мікрооперацій, таблиця розміщення мікрокоманд у мікропрограмній пам'яті (таблиця мікропрограмування);

Графічна частина проекту повинна містити:

- 1) загальну структурну схему АЛП;
- 2) структурну схему блока мікропрограмного керування (БМУ) АЛП;
- 3) структурну схему АЛБ на всіх рівнях опису із зазначенням на ній усіх керуючих сигналів та інформаційних зв'язків;

4) принципову схему АЛБ.

Приблизний обсяг пояснювальної записки – 30–40 сторінок. Пояснювальну записку виконують на аркушах форматом А4. З усіх чотирьох боків аркуша повинні бути залишені поля розміром 20 мм, наповненість сторінки має становити 75% від її загального обсягу. Текст виконують комп'ютерним способом, шрифтом гарнітурою Times New Roman, розміром 14 пунктів через півтора міжрядкових інтервали. Заголовок розділу, підрозділу від попереднього тексту відокремлюють трьома інтервалами. Відступ від назви підрозділу (пункту, підпункту) до наступного тексту повинен складати півтора інтервали.

Рисунки, таблиці, графи-схеми мікроалгоритмів та інший пояснювальний матеріал виконується в основному тексті пояснювальної записки. Принципова схема АЛБ виконується згідно з вимогами ГОСТ 2.109-73. Цифрові логічні елементи позначають згідно з ГОСТ 2.743-91. У разі неможливості відповідати вимогам ГОСТ 2.743-91 у зв'язку з великою кількістю елементів, принципова схема АЛБ може виконуватися на аркуші форматом А3 і наводитися не в основному тексті пояснювальної записки, а в додатку. Для розробки принципової схеми АЛБ припустимо також використання можливостей спеціалізованого програмного забезпечення для моделювання та проектування цифрових електронних схем (PSpice, P-CAD, OrCAD, Design Center, ACCEL, Micro Cap V, Electronics Workbench та ін.).

### 3 ЗАВДАННЯ НА КУРСОВИЙ ПРОЕКТ

Варіант завдання визначається чотирма молодшими бітами номера залікової книжки у двійковій системі числення за наведеними нижче таблицями 1.1 та 2.1. Номер залікової книжки у десятковій системі числення потрібно перевести в двійкову систему, тобто подати його у вигляді  $b_{n-1} \dots b_2 b_1 b_0$ , де  $b_i$  – значення відповідних двійкових розрядів і підставити до таблиць отримані значення розрядів  $b_3 b_2 b_1 b_0$ . Наприклад, якщо номер залікової книжки  $(978498)_{10} = (11101110111001000010)_2$  то  $b_0=0$ ,  $b_1=1$ ,  $b_2=0$ ,  $b_3=0$ ,  $b_4=0$ ,  $b_5=0$ ,  $b_6=1$ .

### **Завдання 1.**

Надати перелік основного обладнання ПК, на якому виконується курсова робота, із зазначенням виробника, основних технічних характеристик та описом призначення пристрою.

### **Завдання 2.**

Комп'ютер з архітектурою фон Неймана побудований на базі процесора із зовнішньою тактовою частотою  $f_{ext}$ , розрядністю внутрішніх реєстрів  $R$ , розрядністю шини даних  $D$ , розрядністю шини адреси  $A$ , величиною циклу обміну по шині даних  $\tau$ . Кількість операцій у системі команд процесора  $K$ . Операнди команд адресуються без використання РЗП, один з операндів можна задати безпосереднім значенням, але при цьому його розрядність збігається з внутрішньою розрядністю процесора. Адресація команд примусова, усі команди основної групи трьохоперандні. Мінімальною структурною одиницею пам'яті, що адресується, є байт. Виконання операцій з плаваючою комою підтримується арифметичним співпроцесором, система команд якого налічує FP команд. Адресні простори процесора і співпроцесора збігаються і використовуються для обміну даними між ними. Для запису числа у формі з плаваючою комою в розрядній сітці використовується єдиний формат. Числа зберігаються в нормалізованому вигляді. Формат передбачає наявність розряду знака мантиси, поля цифрових розрядів мантиси завдовжки  $M$ , розряду знака порядку, поля цифрових розрядів порядку завдовжки  $S$ . Співпроцесор використовує два формати команд: команди обробки, адресація операндів яких побудована на стековій структурі реєстрів співпроцесора; однооперандні команди обміну між вершиною стека співпроцесора і пам'яттю. Адресація команд співпроцесора автоінкрементна. Система команд співпроцесора не має команд передавання керування. Визначити:

1. Адресний простір такого комп'ютера.
2. Діапазон подання чисел з фіксованою комою.
3. Діапазон подання чисел з плаваючою комою.
4. Швидкість передачі даних або смугу пропускання шини даних.



5. Довжину машинного коду команди основної групи у разі розміщенні всіх операндів у пам'яті.

6. Довжину машинного коду команди основної групи, якщо один з операндів задано безпосереднім значенням.

7. Довжину машинного коду команд обробки співпроцесора у разі адресації операндів з використанням стекової пам'яті.

8. Довжину машинного коду команд співпроцесора для обміну з пам'яттю.

Таблиця 1.1 – Початкові дані до завдання 2

$b_3b_2b_1b_0$	$f_{ext}$ , МГц	R/D/A	Reg	$\tau$ , такт	K	Mem, Мб	FP	M, біт	S, біт
1	2	3	4	5	6	7	8	9	10
0000	66	32/16/22	8	5	98	1	25	15	15
0001	100	32/32/24	12	3	127	4	34	30	16
0010	133	32/16/32	16	2	164	16	42	12	18
0011	166	32/32/32	32	1	198	32	51	40	24
0100	200	32/64/36	48	1	230	64	59	64	30
0101	33	24/16/20	64	7	100	1	67	41	23
0110	66	32/32/28	32	3	147	16	69	48	14
0111	100	32/64/32	48	2	286	32	81	20	10
1000	133	64/64/48	64	1	312	128	121	24	36
1001	166	64/32/64	20	1	287	256	97	23	23
1010	200	64/32/32	128	2	268	196	65	10	20
1011	66	24/16/24	12	4	154	4	49	18	12
1100	100	32/32/32	32	5	232	32	32	14	48
1101	166	32/64/32	48	6	293	128	83	36	24
1110	200	32/64/32	64	2	326	256	100	64	62
1111	333	64/64/64	128	1	348	512	161	62	64

### Завдання 3

Відмовившись в умовах попереднього завдання від примусової адресації команд, розробити структуру машинних команд основної групи та команд обміну між верхівкою стека співпроцесора і пам'яттю при використанні в них також базової та базово-індексної адресації операндів. Визначити довжину машинного коду для кожної з можливих структур.

### Завдання 4

Для реалізації операції Op над двійковими числами з фіксованою комою, що зберігаються в пам'яті в С коді виконати такі дії:

1. Розробити структурну схему операційного блока АЛП на Ф-рівні.
2. Розробити структурну схему операційного блока АЛП на ФС-рівні.
3. Згідно з розробленими структурними схемами розробити:

- а) змістовну граф-схему алгоритму операції на Ф-рівні;
- б) закодовану граф-схему алгоритму операції на ФС-рівні;
- в) логічну схему алгоритму операції;

г) структуру мікрокоманди та мікропрограму для БМУ, використовуючи МР метод мікропрограмування.

4. Розробити принципову схему та креслення операційного блока АЛП.

Для виконання цього завдання передбачити формування значень прапорців регістра ознаки результату та за необхідності їх мікропрограмну обробку.

Таблиця 2.1 – Початкові дані до завдання 4

b <sub>3</sub> b <sub>2</sub> b <sub>1</sub> b <sub>0</sub>	Op	C	MP
1	2	3	5
0000	$U = \frac{X^2 + 3Y}{Z}$	прямому	горизонтальний
0001	$U = \frac{2X^2 + 3Y}{Z}$	зворотному	комбінований

## Продовження таблиці 2.1

0010	$U = \frac{3X^2 - 2Y}{Z}$	додатковому	горизонтальний
0011	$U = \frac{X^2 + 3Y}{Z}$	зворотному	комбінований
0100	$U = \frac{2X^2 + 3Y}{Z}$	прямому	горизонтальний
0101	$U = \frac{3X^2 - 2Y}{Z}$	додатковому	комбінований
0110	$U = \frac{X^2 + 3Y}{Z}$	додатковому	горизонтальний
0111	$U = \frac{2X^2 + 3Y}{Z}$	прямому	комбінований
1000	$U = \frac{3X^2 - 2Y}{Z}$	зворотному	горизонтальний
1001	$U = \frac{X^2 + 3Y}{Z}$	прямому	комбінований
1010	$U = \frac{2X^2 + 3Y}{Z}$	додатковому	горизонтальний
1011	$U = \frac{3X^2 - 2Y}{Z}$	прямому	комбінований
1100	$U = \frac{X^2 + 3Y}{Z}$	зворотному	горизонтальний
1101	$U = \frac{2X^2 + 3Y}{Z}$	додатковому	комбінований
1110	$U = \frac{3X^2 - 2Y}{Z}$	прямому	горизонтальний
1111	$U = \frac{X^2 + 3Y}{Z}$	додатковому	комбінований

## 4 РЕКОМЕНДАЦІЇ ЩОДО ВИКОНАННЯ ПРОЕКТУ

### Завдання 1

Для визначення конфігурації ПК необхідно скористатися відомостями, наданими операційною системою. Перелік встановленого обладнання можна отримати за допомогою спеціального виклику «Панель керування» → «Система» → «Обладнання» → «Диспетчер пристроїв».

### Завдання 2

Сукупність усіх фізичних адрес, які можуть бути сформовані на зовнішній шині адреси процесора, утворюють його адресний простір. Розмір адресного простору процесора (тобто кількість машинних слів, до яких він може звернутися) залежить від його розрядності.

Розрядністю процесора називають максимальну кількість розрядів двійкового коду (бітів), які можуть оброблятися або передаватися одночасно. Поняття «розрядність» охоплює розрядність внутрішніх регістрів (R), розрядність шини даних (D), розрядність шини адреси (A). Тому зазвичай розрядність процесора позначають R/D/A (таблиця 3.1).

Адресний простір процесора визначається розрядністю шини адреси. Якщо ширина шини адреси дорівнює A бітів, а розмір машинного слова пам'яті дорівнює одному байту, то адресний простір процесора складає  $2^A$  байти. Наприклад, для процесорів з 20-розрядною шиною адреси адресний простір становить  $2^{20}$  байти ( $2^{20}/1024/1024 = 1$  Мбайт).

Розрядність внутрішніх регістрів R (внутрішня довжина машинного слова) визначає діапазони подання чисел у розрядній сітці. Діапазон подання чисел визначається відношенням  $|X_{\max}|$  до  $|X_{\min}|$  – відповідно максимального та мінімального за абсолютним значенням машинного слова, що може бути розміщено в розрядній сітці комп'ютера довжиною n бітів.

З фіксацією коми після старшого розряду  $|X_{\max}|=1-2^{-n}$ ,  $|X_{\min}|=2^{-n}$ . З фіксацією коми після молодшого розряду  $|X_{\max}|=2^n-1$ ,  $|X_{\min}|=1$ . І в тому, і в іншому випадку діапазон подання чисел складає:

$$D = \frac{|X_{max}|}{|X_{min}|} = \frac{1 - 2^{-n}}{2^{-n}} = \frac{2^n - 1}{1} = 2^n - 1 \approx 2^n.$$

Кількість розрядів сітки  $n$ , яка використовується для розміщення числа в пам'яті зазвичай збігається з розрядністю внутрішніх регістрів процесора.

Діапазон подання чисел з плаваючою комою  $X = \pm M \cdot p^{\pm S}$  у нормалізованому вигляді ( $2^{-1} \leq |M| \leq 1 - 2^{-n}$ ) визначається довжиною  $n$  поля мантиси  $M$  і довжиною  $k$  поля порядку  $S$ , але вирішальне значення має довжина  $k$  поля порядку.

$$|X_{min}| = |M_{min}| \cdot 2^{-|S_{max}|} = 2^{-1} \cdot 2^{-(2^k - 1)} = 2^{-2^k},$$

$$|X_{max}| = |M_{max}| \cdot 2^{|S_{max}|} = (1 - 2^{-n}) \cdot 2^{2^k - 1} \approx 2^{2^k - 1},$$

$$D = \frac{|X_{max}|}{|X_{min}|} = \frac{2^{2^k - 1}}{2^{-2^k}} = 2^{2^k} \cdot 2^{2^k - 1} = 2^{2^k - 1 + 2^k} = 2^{2^{k+1} - 1}.$$

Від розрядності шини даних (зовнішньої довжини машинного слова) залежить швидкість передачі інформації між процесором, основною пам'яттю і зовнішніми пристроями. Швидкість передачі даних або смуга пропускання шини даних визначається як:

$$P = \frac{f_{ext} \cdot D}{\tau},$$

де  $\tau$  – величина циклу обміну, такт;  $D$  – розрядність шини, біт;  $f_{ext}$  – зовнішня частота процесора, МГц.

Наприклад, при  $D = 64$  біта,  $f_{ext} = 200$  МГц,  $\tau = 1$  такт

$$P = \frac{200 \text{ МГц} \cdot 64 \text{ біта}}{1} = \frac{200 \cdot 10^6 \text{ Гц} \cdot 64 \text{ біта}}{1} = 12800 \cdot 10^6 \frac{\text{біт}}{\text{с}} = 12800 \frac{\text{Мбіт}}{\text{с}}$$

Машинна команда являє собою закодоване за певними правилами машинне слово, що містить узагальнені вказівки процесора про вид виконуваної операції, місця розташування операндів і результату й про адресу

наступної команди. Спрощено структура машинної команди може бути подана так:

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

Операційна частина містить код операції (КОП). КОП – закодоване машинне слово, що визначає вид елементарних машинних операцій (додавання, пересилання й т. д), що виконує процесор. Адресна частина команди містить операнди, або, у загальному випадку, інформацію про їх місце розташування й адреси, місце розташування й адресу результату. У разі використання так званої примусової адресації машинний код команди також містить і адресу наступної команди, інакше адреса наступної команди отримується процесором збільшенням адреси попередньої команди на довжину її коду.

До команд основної групи належать команди бінарних (тобто тих, що мають два операнди) операцій, які виконуються АЛП процесора (арифметичні й порозрядні логічні команди, команди двійково-десятькової арифметики тощо).

За умови примусової адресації, для того щоб команда містила наявно всю необхідну інформацію про виконувану операцію, вона повинна складатися з поля коду операції (КОП) і чотирьох адресних полів для вказівки місця розташування операндів (A1 й A2), результату (A3) й адреси наступної виконуваної команди (A4).

КОП	Перша адреса (A1)	Друга адреса (A2)	Третя адреса (A3)	Четверта адреса (A4)
-----	----------------------	----------------------	----------------------	-------------------------

Число двійкових розрядів, необхідних для кодування виду операції (довжина КОП), визначається як  $N_{КОП} = \lceil \log_2 K \rceil$ , де  $K$  – кількість команд у системі процесора, а символи  $\lceil \rceil$  позначають операцію округлення у бік більшого цілого. Довжина кожного з адресних полів визначається як  $N_{адреси} = \lceil \log_2 Mem \rceil$ , де  $Mem$  – ємність основної пам'яті в мінімальних

структурних одиницях, які може адресувати процесор, що для більшості комп'ютерів становить один байт. Наприклад, при  $K=200$  і  $Mem=16$  Мб одержимо:

$$N_{КОП} = \lceil \log_2 200 \rceil = 7.64 \approx 8;$$

$$N_{адреси} = \lceil \log_2 (16 \cdot 1024 \cdot 1024) \rceil = 24;$$

$$N_{команди} = 24 \cdot 4 + 8 = 104 \text{ біти} = 13 \text{ байт.}$$

Якщо один з операндів задається безпосереднім значенням, то при цьому зазвичай його розрядність збігається з внутрішньою розрядністю процесора. У цьому випадку  $N_{команди} = N_{КОП} + 3 \cdot N_{адреси} + R = \lceil \log_2 K \rceil + 3 \cdot \lceil \log_2 Mem \rceil + R$ .

Адресація з використанням стекової пам'яті дозволяє проводити обробку машинних слів лише з використанням операцій читання/запису стека й повністю безадресними командами, що вказують спеціальним способом тільки код операції.

Довжина машинного коду команд обробки співпроцесора під час адресації операндів з використанням стекової пам'яті визначається довжиною КОП системи команд співпроцесора (тобто кількістю команд у системі команд співпроцесора) і складає  $N_{команди} = N_{КОП} \lceil \log_2 FP \rceil$ . Довжина машинного коду команд співпроцесора для обміну з пам'яттю залежить від кількості команд у системі команд співпроцесора та способів адресації пам'яті. Якщо операнди команд адресуються без використання регістрів загального призначення і адресний простір співпроцесора перетинається з адресним простором процесора, то довжина машинного коду таких команд співпроцесора буде  $N_{команди} = N_{КОП} + N_{адреси} \lceil \log_2 FP \rceil + \lceil \log_2 Mem \rceil$ .

У розглянутому вище прикладі при  $K=200$  і  $Mem=16$  Мб довжина машинного коду команди виявилася рівною 13 байтам, тобто кратною мінімальній структурній одиниці пам'яті, яку може адресувати процесор. Загалом у разі невиконання умови кратності довжина машинного коду команди повинна бути штучно вирівняна на межу байта. Таке вирівнювання може

виконуватися додаванням зайвих бітів у поле КОП команди та/або введенням у машинний код команди ще одного поля – ознаки адресації.

### Завдання 3

Якщо кількість операцій у системі команд процесора така, що для КОП не вистачає 256 різних кодів, які можна закодувати одним байтом, то двох байтів, у яких вже можна закодувати 65635 кодів операцій, виявляється занадто багато. У такому разі команди поєднуються в групи, відповідно до їх функціональних ознак і їм привласнюється той самий байтовий КОП, а в другому байті цей КОП уточнюється указанням типу адресації операндів.

КОП	Ознака адресації	Адресна частина
-----	------------------	-----------------

Способи адресації визначають техніку обчислення (трансляції) адрес комірок пам'яті, у яких зберігаються операнди й виконання операцій над адресними регістрами. Різні способи адресації розроблені й використовуються для забезпечення компактних адресних посилань для випадків, коли машинна адреса має занадто велику довжину і її незручно в такому вигляді записувати в команду.

Найпоширенішими способами адресації операндів машинних команд є адресація з використанням регістрів загального призначення (РЗП). Адресація з використанням РЗП дозволяє зменшити довжину команди, тому що для завдання коду регістра в кодї команди потрібна невелика кількість розрядів. Використання РЗП також позитивне щодо часу виконання команд, оскільки звертання до регістрів для формування адреси операнда виконується дуже швидко. Серед усіх різновидів способів адресації операндів з використанням РЗП дуже широко використовуються базова та базово-індексна адресація.

У разі базової адресації адресна частина команди містить адресу (код) регістра, у якому перебуває адреса операнда. У разі базової індексної регістрової адресації в адресній частині команди розташовуються коди двох



регістрів – базового й індексного. Адреса комірки пам'яті, у якій знаходиться операнд, отримується підсумовуванням значень базового й індексного регістра.

Для розробки формату машинних команд основної групи необхідно визначити правила кодування поля ознаки адресації з урахуванням усіх можливих способів адресації та комбінацій операндів, а також правила кодування адресної частини команди.

Найпростішим рішенням буде виділення в межах поля ознаки адресації окремих полів для зазначення способів адресації кожного з трьох операндів команди.

КОП	Ознака адресації першого операнда	Ознака адресації другого операнда	Ознака адресації третього операнда	Адресна частина
-----	-----------------------------------	-----------------------------------	------------------------------------	-----------------

Потім потрібно визначити кодування поля ознаки адресації кожного операнда з урахуванням усіх можливих її способів. Оскільки в умовах завдання є чотири способи адресації, то зробити це можна, наприклад, так: 00 – безпосередній операнд, 01 – базова адресація, 10 – базово-індексна адресація, 11 – пряма адресація.

Потім необхідно визначити правила кодування адресної частини команди. Для трьохоперандного формату команд основної групи адресна частина складатиметься з трьох полів.

КОП	Ознака адресації			Адресна частина першого операнда	Адресна частина другого операнда	Адресна частина третього операнда
	ОА1	ОА2	ОА3			

Адресна частина кожного операнда повинна містити адресний код відповідно до способу адресації, зазначеного у відповідному полі ознаки адресації. Кодування адресних частин залежить від розміру пам'яті комп'ютера, внутрішньої розрядності та кількості регістрів процесора.

Якщо операнд задається прямою адресою, то довжина відповідної адресної частини повинна бути не менш ніж  $\lceil \log_2 \text{Mem} \rceil$ . Якщо операнд задається безпосереднім значенням, то довжина відповідної адресної частини повинна бути не менш ніж  $R$ . Якщо для адресації використовується базова та базово-індексна адресація, то довжина відповідної адресної частини визначається кількістю регістрів і пар регістрів, що використовуються для зазначення адреси.

Вибір кодування полів адресних частин операндів повинен як мінімум забезпечити кратність байта загальної довжини машинного коду команди і можливість його однозначної інтерпретації. Загальні рекомендації такі:

1. Якщо довжина КОП менша, ніж байт, то КОП можна вирівнювати до байта розміщенням у ньому бітів поля ознаки результату.

2. Якщо довжина КОП більша, ніж байт, то КОП можна вирівнювати до двох байтів розміщенням у ньому бітів поля ознаки результату.

3. Якщо довжина поля ознаки результату менша, ніж байт, то поле ознаки можна вирівнювати до байта розміщенням у ньому частини бітів поля будь-якої з адресних частин.

4. Можна фіксувати положення певної адресної частини відповідно до способу адресації, а не до положення операнда.

5. Для базової адресації можна використовувати не всі наявні регістри процесора, а лише певну їх частину.

6. Для базово-індексної адресації можна комбінувати в пари не всі можливі регістри, а певну їх частину.

7. Для кожного з трьох операндів можна застосовувати індивідуальні набори регістрів у разі базової і пари регістрів у разі базово-індексної адресації.

8. За певних комбінацій способів адресації у полі ознаки адресації можна використовувати деяку кількість бітів однієї адресної частини для розміщення в ній бітів іншої адресної частини.

9. Можна відмовитись від певної частини можливих комбінацій ознак адресації.

#### **Завдання 4**

Операційний блок для реалізації операції цього завдання може реалізовуватися як для функціонального, так і для універсального АЛП. У функціональних АЛП різні операції виконуються в окремих блоках, що підвищує швидкість роботи, тому оскільки блоки можуть паралельно виконувати відповідні операції, але при цьому значно збільшуються витрати устаткування. В універсальних АЛП усі операції виконуються тими самими схемами, які комутуються потрібним чином, залежно від необхідного режиму роботи.

Використовуючи метод декомпозиції для реалізації операційного блока функціонального АЛП, можна зазначити, що обчислення значення будь-якого виразу цього завдання вимагає виконання операцій множення та зсуву результату вліво чи вправо. Піднесення до квадрата реалізується множенням операнда на самого себе, а операції додавання та віднімання можуть бути реалізовані на основі суматорів блоків множення з використанням відповідного кодування. Операції множення на 2 і 3 – це операції спеціальної арифметики, які легко реалізуються виконанням операцій зрушення вліво на регістрах і додавання вихідного та зрушеного значення операнда.

Структура блоків для виконання операцій множення та ділення залежить від вибраного алгоритму виконання операції, але завдання не обумовлює використання тих або інших методів, їх може вибирати розробник на власний розсуд. Наприклад, для множення можна скористатися першим методом – множенням, починаючи з молодших розрядів множника, зі зрушенням суми часткових добутоків управо з непорушним множенням, а для ділення – методом без відновлення негативного залишку з непорушним дільником і діленням, що зрушується вліво.

Структурну схему операційного блока на Ф-рівні для виконання множення за першим методом для чисел з фіксованою комою в прямому коді зображено на рисунку 4.1.

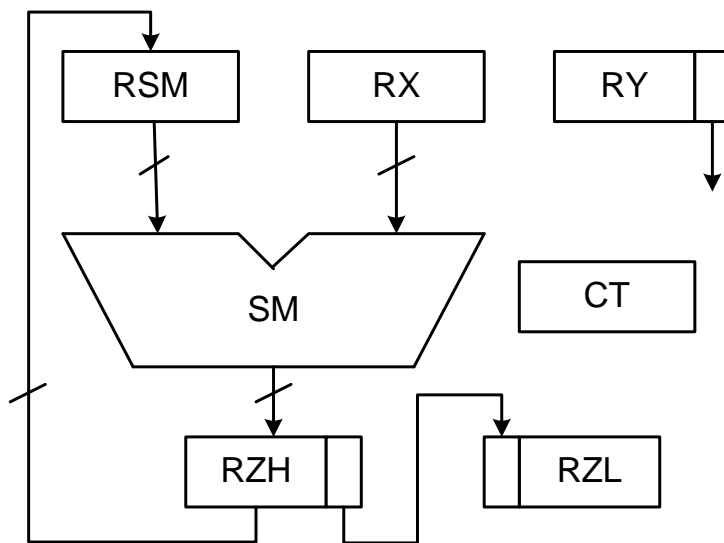


Рисунок 4.1 – Структурна схема операційного блока на Ф-рівні

RX, RY, RSM, RZH, RZL – n-розрядні регістри з набором мікрооперацій встановлення в 0, приймання і видачі машинного слова, лінійного зрушення на один розряд управо. Для RX, RY, RZH і RZL встановлення в 0 не обов’язкове.

Для RX і RSM лінійне зрушення на один розряд вправо не обов’язкове. CT – n-розрядний лічильник, що рахує в природному порядку з мікроопераціями декремента та встановлення заданого початкового значення.

Вважаємо, що початкові значення операндів X і Y завантажені у вихідному стані в регістри RX та RY і мікрооперації запису в регістри для реалізації мікроалгоритму не потрібні.

Граф-схему алгоритму (ГСА) операції на Ф-рівні в змістовній формі зображено на

Рисунок 0.2. Для переходу до опису мікроалгоритму на ФС-рівні виконаємо деталізацію схеми (Рисунок 0.3) і складемо таблицю мікрооперацій мікроалгоритму.

Зрушення вправо суми часткових добутків з непорушним множенням виконується передаванням навкіс її старших бітів із суматора SM і записом їх

до регістра RZH. При цьому для збереження молодшого біта використовують рушійний регістр RZL. Рушійний регістр являє собою групу тригерів, з'єднаних так, що інформація з кожного тригера передається до наступного, зрушуючи код, що записаний у регістрі. Запис до регістра RZL при цьому виконується одночасно із записом до регістра RZH і  $y_{10} = y_6$ . Така схема дозволяє спростити реалізацію операції зрушення суми часткових добутоків і поєднати вершини 6 і 7 ГСА.

Керувальний сигнал мікрооперації видачі значення з регістра RX для підсумовування із сумою часткових добутоків стробується значенням молодшого біта регістра RY, що за нульового значення цього біта забезпечує зрушення суми часткових добутоків без додавання множеного. Така схема з лінійним зрушенням вправо на один біт вмісту регістра з кожним тактом дозволяє позбутися вершини 3 ГСА.

За умови, що лічильник СТ виконує декремент початкового значення, комбінаційна схема КС для формування логічної умови  $X_1$  – нульове значення лічильника, являтиме собою один n-входовий логічний елемент АБО-НІ.

Деталізована схема пристрою для опиму мікроалгоритму на ФС-рівні містить позначення таких мікрооперацій:

- $y_1$  – запис у RSM;
- $y_2$  – видача з RSM у прямому коді;
- $y_3$  – видача з RX у прямому коді;
- $y_4$  – видача з RY у прямому коді;
- $y_5$  – лінійне RY зрушення на один розряд управо;
- $y_6$  – запис у RZH;
- $y_7$  – видача з RZH у прямому коді;
- $y_8$  – установлення початкового значення СТ;
- $y_9$  – декремент СТ;
- $y_{10}$  – запис у RZL.

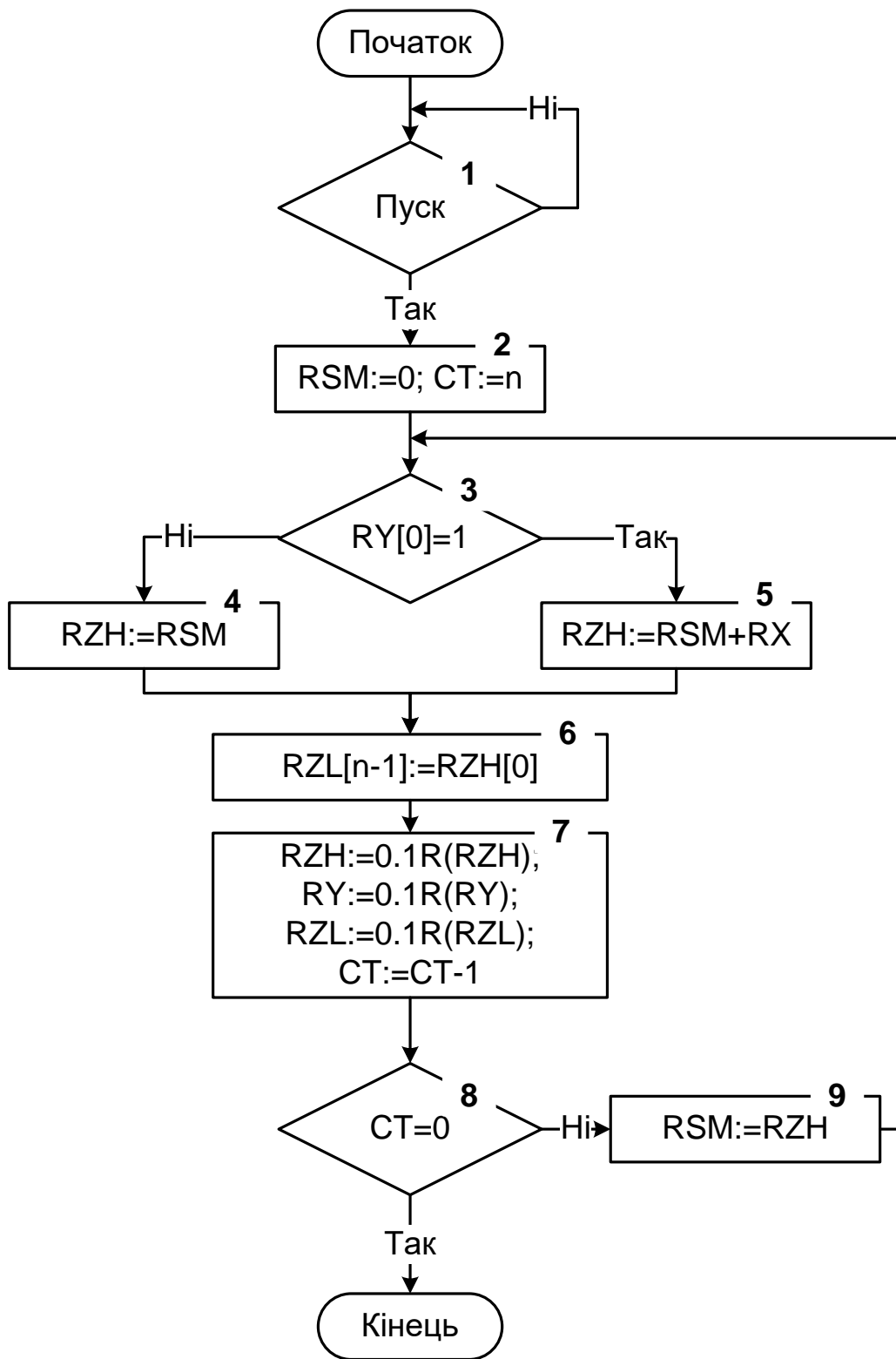


Рисунок 0.2 – Граф-схема алгоритму (ГСА) операції на  $\Phi$ -рівні в змістовній формі

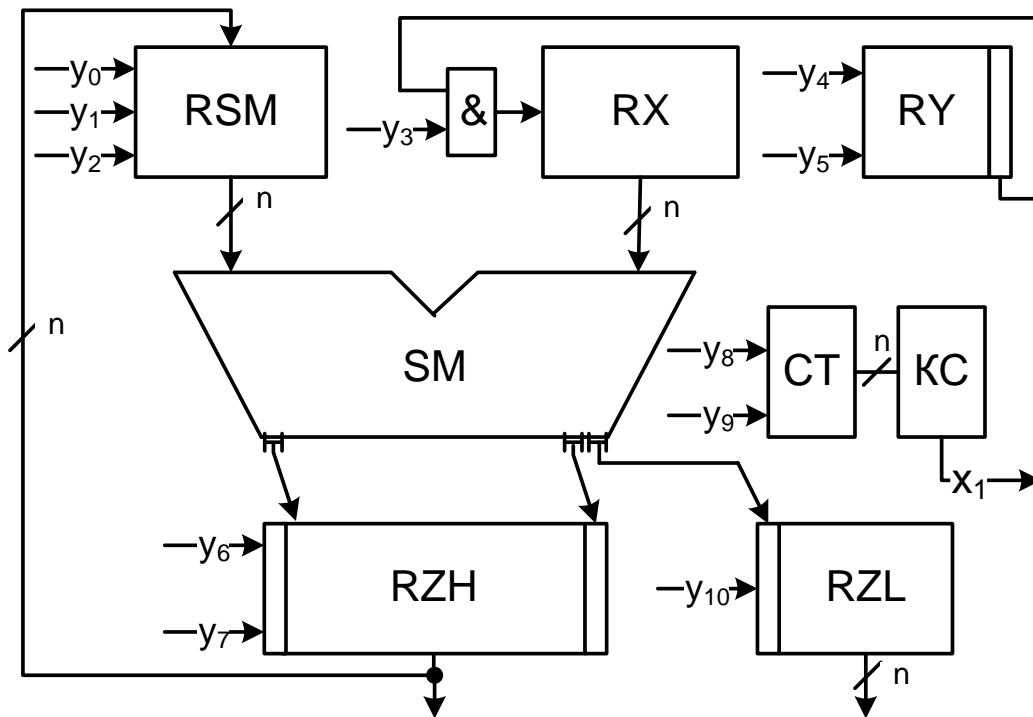


Рисунок 0.3 – Деталізована схема пристрою

Таблиця 0.1 – Таблиця мікрооперацій мікроалгоритму

МО $Y_i$	Опис мовою мікрооперацій	КС МО
1	2	3
$Y_0$	RSM:=0	$y_0$
$Y_1$	CT:=n	$y_8$
$Y_2$	RZH:=RSM	$y_2 y_3 y_6$
$Y_3$	RZH:=RSM+RX	$y_2 y_3 y_6$
$Y_4$	RZL[n-1]:=RZH[0]	$y_2 y_3 y_6$
$Y_5$	RZH:=0.1R(RZH)	$y_2 y_3 y_6$
$Y_6$	RY:=0.1R(RY)	$y_5$
$Y_7$	RZL:=0.1R(RZL)	$y_2 y_3 y_6$
$Y_8$	CT:=CT-1	$y_9$
$Y_9$	RSM:=RZH	$y_7 y_1$

Граф-схему алгоритму операції на ФС-рівні в закодованій формі зображено на Рисунок 0.4, а логічна схема алгоритму має вигляд:

$$\overset{1}{\Pi} \downarrow (y_0, y_8) (y_2 y_3 y_6) (y_5 y_9) \overset{2}{\uparrow} x_1 (y_1 y_7) \overset{12}{\uparrow} \downarrow x_1 K,$$

де  $y_i$  – сигнали мікрооперацій (Таблиця 0.1),  $X_1$  – логічна умова  $CT=0$

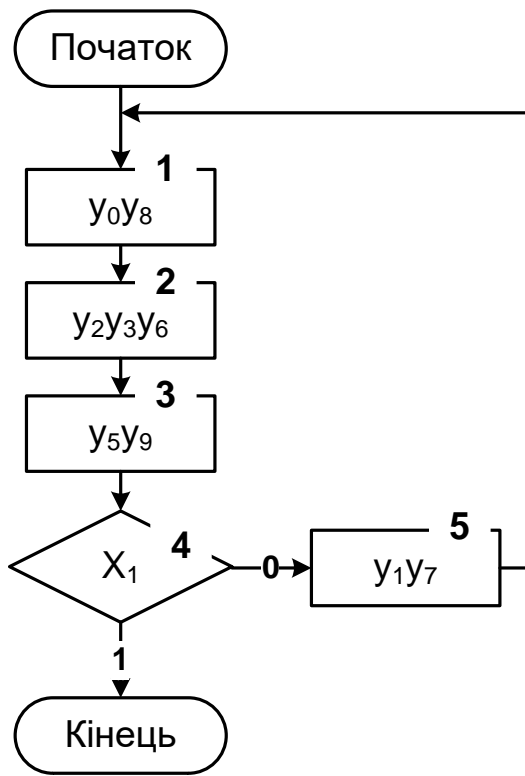


Рисунок 0.4 – Граф-схема алгоритму операції на ФС-рівні в закодованій формі

Таблиця 4.2. – Карта мікропрограмування для БМУ з примусовою адресацією мікрокоманд з використанням горизонтального методу мікропрограмування

№ МК	Адреса	Мікрокоманда															
		$\beta_1$		$\beta_2$				$\beta_4$									
		М	К	$y_0$	$y_1$	$y_2 y_3$	$y_4 y_5 y_6$		$y_7$	$y_8 y_9$							
Н	0000	11	000	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0001	00	001	1	0	0	0	0	0	0	0	0	0	1	0		
2	0010	11	001	0	0	1	1	0	0	1	0	0	0				
3	0011	00	010	0	0	0	0	0	0	1	0	0	0	0	1		



Продовження таблиці 4.2

4	0 1 0 0	0 1	0 1 1	0 0 0 0 0 0 0 0 0 0 0	1
5	0 1 1 0	0 0	0 0 0	0 1 0 0 0 0 0 1 0 0	0
К	0 1 1 1	0 0	0 0 0	0 0 0 0 0 0 0 0 0 0	0

Аналогічно вищерозглянутому виконується синтез операційного блока ділення. Наприклад, для реалізації алгоритму ділення можна скористатися методом без відновлення негативного залишку з непорушним дільником і діленим, що зрушується вліво, який розглядався в лабораторних роботах.

Операційний блок для виконання операції може реалізовуватися як для функціонального, так і для універсального АЛП. При реалізації функціонального АЛП розроблені окремі функціональні блоки виконання операцій множення та ділення поєднуються поміж собою з налагодженням потрібних інформаційних зв'язків. Під час реалізації операцій спеціальної арифметики до складу об'єданого блока можна додати спеціальний окремий блок, або реалізувати їх на рівні апаратних можливостей будь-якої частини об'єданого блока. Загальний мікроалгоритм виконання операції отримується об'єднанням мікроалгоритмів виконання операцій множення, ділення та спеціальної арифметики в послідовності, що визначається значенням виразу для обчислення.

Під час синтезу універсального АЛП для вибору методів виконання операцій множення та ділення для реалізації мікроалгоритмів слід вибирати ті, які найкраще суміщуються (передбачають використання суматорів і регістрів однакової розрядності, однакового напрямку зрушення на регістрах). Проаналізувавши розроблені окремі функціональні блоки виконання операцій множення та ділення, можна визначити елементи, які можуть спільно використовуватися при виконанні різних операцій. Це дозволить скоротити апаратні витрати і кількість сигналів мікрооперацій мікроалгоритмів.

## **ЗАХИСТ КУРСОВИХ ПРОЕКТІВ**

Захист курсових проектів є особливою формою перевірки якості виконання роботи і знань у цій галузі.

Захист проводиться перед спеціальною комісією за безпосередньої участі керівника проекту.

Захист полягає в короткій доповіді (5–8 хв) студента з виконаної роботи і відповідях на питання. Студент повинен дати всі пояснення по суті роботи.

Якщо в конструкторській документації проекту будуть виявлені грубі порушення ЄСКД, або виявиться, що спроектований пристрій принципово непрацездатний, роботу оцінюють незадовільно і повертають на доопрацювання.

Студент, який не подав в установлені строки курсовий проект та не захистив його з неповажної причини, вважається таким, що має академічну заборгованість.

## **КРИТЕРІЇ ОЦІНЮВАННЯ КУРСОВОГО ПРОЕКТУ**

Загальна оцінка, яку може отримати студент, за виконання курсового проекту становить 100 балів та складається із чотирьох основних частин:

- зміст та оформлення пояснювальної записки,
- поданий до захисту ілюстративний матеріал,
- захист проекту,
- відповіді на додаткові запитання під час захисту.

Пояснювальна записка повинна містити детальний опис виконання всіх пунктів, зазначених у п. 3 даних методичних вказівок, з дотриманням чинних норм і стандартів на оформлення технічної документації.

Поданий до захисту ілюстративний матеріал повинен достатньою мірою відображати виконання курсового проекту та результати проектування.

У доповіді під час захисту студент повинен продемонструвати теоретичні та практичні знання в галузі розробки цифрових пристроїв.

У разі успішної відповіді студента на додаткові питання, що виявлять його поглиблені знання щодо проектування цифрових автоматів, студент може отримати додаткові бали під час захисту. Розподіл балів за відповідними складовими частинам наведено у табл. 6.1.

Таблиця 6.1 – Нарахування балів для оцінювання курсового проекту

Пояснювальна записка	Ілюстративна частина	Захист роботи	Відповіді на додаткові питання	Сума
до 40	до 20	до 30	до 10	100

## СПИСОК ЛІТЕРАТУРИ

1. Матвієнко М. П. Архітектура комп'ютера : навчальний посібник / М. П. Матвієнко, В. П. Розен, О. М. Закладний. – К : Видавництво Ліра, 2016. – 264 с.
2. Мельник А. Архітектура комп'ютера: наукове видання / А. Мельник. – Луцьк : Волинська обласна друкарня, 2008. – 470 с.
3. Жабін В. І. Прикладна теорія цифрових автоматів : навч. посібник / В. І Жабін, І. А. Жуков, І. А. Клименко, В. В. Ткаченко. – К. : Книжкове видавництво НАУ, 2007. – 364 с.
4. Бабич Н. П. Компьютерная схемотехника. Методы построения и проектирования: учебное пособие / Н. П. Бабич, И. А. Жуков. – К. : МК-Пресс, 2004 – 576 с.
5. Нешумова К. А. Электронные вычислительные машины и системы / К. А. Нешумова. – М. : Высш. шк., 1989. – 366 с.
6. Каган Б. М. Электронные вычислительные машины и системы / Б. М. Каган. – М. : Энергоатомиздат, 1985. – 522 с.
7. Баранов С. И. Синтез микропрограммных автоматов / С. И. Баранов. – [2-е изд., перераб. и доп.] – Л. : Энергия, 1979. – 232 с.
8. Шауман А. М. Основы машинной арифметики./ А. М. Шауман. – Л.: Изд-во Ленингр. ун-та, 1979. – 312 с.
9. Байков В. Д. Аппаратурная реализация элементарных функций в ЭВМ / В. Д. Байков, В. Д. Смоллов. – Л. : Изд-во Ленингр. ун-та, 1975. – 96 с.
10. Карцев М. А. Арифметика цифровых машин / М. А. Карцев. – М. : Наука, 1969. – 576 с.

Зразок оформлення титульної сторінки пояснювальної записки

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

КАФЕДРА КОМП'ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ СИСТЕМ

### КУРСОВИЙ ПРОЕКТ

з навчальної дисципліни «Архітектура комп'ютерів»  
на тему «Визначення конфігурації ПК. Розробка арифметичного пристрою»

Студента 2 курсу КІ-14-1 групи  
зі спеціальності 123  
«Комп'ютерна інженерія»  
Іванова І. І.  
Керівник старш. викладач каф. КІС  
Петров П. П.

Національна шкала \_\_\_\_\_  
Кількість балів: \_\_\_\_ Оцінка: ECTS \_\_\_\_

Члени комісії \_\_\_\_\_  
(підпис) (прізвище та ініціали)  
\_\_\_\_\_  
(підпис) (прізвище та ініціали)  
\_\_\_\_\_  
(підпис) (прізвище та ініціали)

Кременчук – 2018

Методичні вказівки щодо виконання курсового проекту з навчальної дисципліни «Архітектура комп'ютерів» для студентів денної та заочної форм навчання зі спеціальності 123 – «Комп'ютерна інженерія» освітнього ступеня «Бакалавр»

Укладачі: старш. викл. Ю. В. Зілінський,  
старш. викл. А. Л. Юдіна

Відповідальний за випуск доц. В. М. Сидоренко

Підп. до др. \_\_\_\_\_. Формат 60x84 1/16. Папір тип. Друк ризографія.  
Ум. друк. арк. \_\_\_\_\_. Наклад \_\_\_\_\_ прим. Зам. № \_\_\_\_\_. Безкоштовно.

Видавничий відділ  
Кременчуцького національного університету  
імені Михайла Остроградського  
вул. Першотравнева, 20, м. Кременчук, 39600