

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**ОДЕСЬКА ДЕРЖАВНА
АКАДЕМІЯ БУДІВНИЦТВА ТА
АРХІТЕКТУРИ**

**Кафедра інформаційних технологій
та прикладної математики**

Методичні рекомендації

**з навчальної дисципліни Комп'ютерна схемотехніка
та архітектура комп'ютерів**

до індивідуальних завдань
для студентів освітньо-професійної програми
Інформаційні системи та технології
за спеціальністю 126. Інформаційні системи та технології
Освітній рівень - перший (бакалаврський)

ОДЕСА - 2023

«РЕКОМЕНДОВАНО»
науково-методичною комісією ННІ БІТ
Протокол № 7 від 31.03.23 р.

Укладач: Пилявський В.В., кандидат технічних наук, доцент
Патлаєнко М.О., кандидат технічних наук, старший викладач

Рецензенти: к.ф.-м.н., доцент Козаченко Т.О., кафедра теоретичної механіки
ОДАБА,
д.т.н., професор Савельєва О.С., кафедра інтегрованих технологій
управління Національного університету «Одеська політехніка»

Методичні рекомендації до виконання індивідуальних завдань для студентів спеціальності 126 Інформаційні системи та технології. Розділи присвячені вивченню основних складових комп'ютерної схемотехніки та архітектурі комп'ютерів. Представлено різноманітні архітектури побудови схем.

Відповідальна за випуск: Лазарева Д.В. - завідувач кафедри інформаційних систем та прикладної математики, кандидат технічних наук, доцент

ВСТУП.....	4
РОЗДІЛ 1. Мета і задачі індивідуального завдання.....	5
РОЗДІЛ 2. Рекомендації щодо написання та оформлення індивідуального завдання	6
РОЗДІЛ 3. Рекомендації щодо виконання завдання	9
РОЗДІЛ 4. Захист індивідуального завдання	24
РОЗДІЛ 5. Індивідуальне завдання для студентів	25
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	26

ВСТУП

Метою індивідуального завдання з дисципліни "Комп'ютерна схемотехніка та архітектура комп'ютерів" є закріплення основних теоретичних положень, які студенти засвоюють з лекційного матеріалу та спеціальної літератури. Головним завданням цієї роботи є набуття студентами практичних навичок у використанні знань, щоб розробляти структуру комп'ютера, проектувати функціональні та принципові схеми пристроїв, розробляти систему команд, структури даних, способи адресації, мікроалгоритми та мікропрограми для різних операцій, а також ознайомитися з засобами автоматизації проектування обчислювальних засобів.

Після виконання роботи студенти повинні мати чітке уявлення про конфігурацію та склад основних блоків ПК, про взаємодію між основними вузлами та блоками комп'ютера під час обробки інформації та вміти використовувати апарат, методи та засоби проектування елементів цифрової обчислювальної техніки. Цій меті найбільше відповідає самостійна розробка структури, алгоритмічного опису, схем та конструкції елементів спеціалізованого процесора комп'ютера.

РОЗДІЛ 1

Мета і задачі індивідуального завдання

Мета індивідуального завдання з дисципліни "Комп'ютерна схемотехніка та архітектура комп'ютерів" полягає в закріпленні основних теоретичних знань, які студенти здобувають з лекцій та спеціальної літератури. Основна задача цього проекту - навчити студентів використовувати набуті знання для розробки структури комп'ютера, проектування функціональних та принципових схем пристроїв, створення системи команд, розробки структур даних, способів адресації, мікроалгоритмів та мікропрограм для різних операцій, а також ознайомлення з інструментами автоматизації проектування обчислювальних засобів. Після виконання проекту студенти повинні мати чітке розуміння конфігурації та складу основних блоків ПК, взаємодії між основними вузлами та блоками комп'ютера під час обробки інформації та вміти використовувати апарат, методи та засоби проектування елементів цифрової обчислювальної техніки. Для досягнення цієї мети найкраще підходить самостійна розробка структури, алгоритмічного опису, схем та конструкції елементів спеціалізованого процесора комп'ютера.

РОЗДІЛ 2

Рекомендації щодо написання та оформлення індивідуального завдання

Під час виконання роботи згідно з індивідуальним варіантом студент повинен виконати наступні завдання:

Визначити конфігурацію власного ПК, основні характеристики та параметри блоків.

Визначити характеристики комп'ютера з вказаними параметрами.

Розробити систему команд процесора з вказаними параметрами.

Розробити структурну схему арифметично-логічного блоку (АЛБ) та арифметично-логічного пристрою (АЛП) на різних рівнях опису.

Розробити мікроалгоритми виконання в АЛП заданих операцій на різних рівнях опису.

Розробити мікропрограми для реалізації мікроалгоритмів.

Розробити принципову схему АЛБ.

Виконана робота має бути оформлена у вигляді пояснювальної записки та графічної частини. Пояснювальна записка має включати такі розділи:

Титульна сторінка з назвою роботи, прізвищем та номером групи студента.

Завдання за індивідуальним варіантом.

Розділ 1 "Конфігурація ПК", що містить перелік обладнання ПК з основними технічними характеристиками.

Розділ 2 "Процесор", що містить розрахунки показників комп'ютера згідно з третім завданням.

Розділ 3 "Система команд процесора", що містить опис структури та форматів команд згідно з другим завданням.

Розділ 4 про "Арифметично-логічний пристрій" може бути розділений на два підрозділи: "Арифметично-логічний блок" та "Блок керування". У цих підрозділах повинні бути описані результати виконання четвертого завдання роботи, загальна структура АЛП, призначення та особливості організації

вузлів вибраного варіанта структури АЛБ, список мікрооперацій, які реалізовані в АЛБ, опис структури блока мікропрограмного керування та призначення його вузлів, опис формату мікрокоманд та кодів мікрооперацій, а також таблиця розміщення мікрокоманд у мікропрограмній пам'яті (таблиця мікропрограмування). Графічна частина роботи повинна включати загальну структурну схему АЛП, структурну схему блока мікропрограмного керування (БМУ) АЛП, та структурну схему АЛБ на всіх рівнях опису з усіма керуючими сигналами та інформаційними зв'язками.

Для написання пояснювальної записки рекомендується використовувати сторінки формату А4 з полями розміром 20 мм на кожній стороні та наповненістю сторінки 75%. Для тексту слід використовувати шрифт Times New Roman розміру 14 пунктів з півтора міжрядковим інтервалом. Рисунки, таблиці, та інший пояснювальний матеріал слід включати в основний текст. Якщо принципова схема АЛБ не може вміститися на стандартних аркушах, то її можна виконати на аркуші форматом А3 та включити в додаток. Цифрові логічні елементи слід позначати згідно з ГОСТ 2.743-91, а принципову схему АЛБ - згідно з вимогами ГОСТ 2.109-73. Також можна використовувати спеціалізоване програмне забезпечення для моделювання та проектування цифрових електронних схем, таке як, P-CAD, Simulink, Electronics Workbench та інше.

За виконання індивідуального завдання студент може отримати загальну оцінку до 100 балів, яка складається з чотирьох основних частин: змісту та оформлення пояснювальної записки, поданого до захисту ілюстративного матеріалу, захисту роботи та відповідей на додаткові запитання під час захисту.

Пояснювальна записка повинна містити детальний опис виконання всіх пунктів, зазначених у методичних вказівках з дотриманням чинних норм та стандартів на оформлення технічної документації.

Ілюстративний матеріал, який буде поданий до захисту, повинен достатньою мірою відображати виконання роботи та її результати.

Під час захисту студент повинен продемонструвати теоретичні та практичні знання в галузі розробки цифрових пристроїв.

Студент може отримати додаткові бали під час захисту, якщо успішно відповість на додаткові питання, що виявлять його поглиблені знання щодо проектування цифрових автоматів. Розподіл балів за відповідними складовими частинами наведено в таблиці 2.1.

Таблиця 2.1

Нарахування балів для оцінювання індивідуального завдання

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

РОЗДІЛ 3

Рекомендації щодо виконання завдання

Завдання 1

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

Завдання 2

Адресний простір такого комп'ютера складається з розрядності шини адреси A , що означає, що можливо адресувати до 2^A окремих байтів пам'яті.

Діапазон подання чисел з фіксованою комою залежить від розрядності регістрів і виконуваних операцій, і не вказано у тексті.

Діапазон подання чисел з плаваючою комою залежить від формату, який передбачає розряд знака мантиси, поля цифрових розрядів мантиси завдовжки M , розряд знака порядку, поля цифрових розрядів порядку завдовжки S . Загальна довжина числа складається з $M+S+1$ розрядів. Числа зберігаються в нормалізованому вигляді, що дозволяє отримати максимальну точність представлення.

Швидкість передачі даних або смугу пропускання шини даних не вказано у тексті.

Довжина машинного коду команди основної групи залежить від кількості операцій у системі команд процесора K , розмірів операндів та виконуваних операцій. У разі розміщення всіх операндів у пам'яті, довжина машинного коду команди складається з $K+2$ байтів.

Довжина машинного коду команди основної групи залежить від розміру операндів та виконуваних операцій. Якщо один з операндів задано безпосереднім значенням, то довжина машинного коду команди збільшується на розмір заданого операнду, який повинен збігатися з внутрішньою розрядністю процесора.

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.

У системі співпроцесора немає команд передачі керування. Нижче наведені відповіді на деякі запитання, що стосуються цієї системи:

- Адресний простір комп'ютера.
- Діапазон значень чисел з фіксованою комою.
- Діапазон значень чисел з плаваючою комою.
- Швидкість передачі даних або пропускну здатність шини даних.
- Довжина машинного коду команди основної групи в разі, якщо всі операнди розміщено в пам'яті.
- Довжина машинного коду команди основної групи, якщо один з операндів задано безпосереднім значенням.
- Довжина машинного коду команд обробки співпроцесора в разі адресації операндів з використанням стекової пам'яті.
- Довжина машинного коду команд співпроцесора для обміну з пам'яттю.

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 X.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 ($2^{-1} \leq |M| \leq 1 - 2^{-n}$) визначається довжиною n поля мантиси M і довжиною k поля порядку S , але вирішальне значення має довжина k поля порядку.

$$\begin{aligned} |X_{\min}| &= |M_{\min}| \cdot 2^{-|S_{\max}|} = 2^{-1} \cdot 2^{-(2^k - 1)} = 2^{-2^k}, \\ |X_{\max}| &= |M_{\max}| \cdot 2^{+|S_{\max}|} = (1 - 2^{-n}) \cdot 2^{2^k - 1} \approx 2^{2^k - 1}, \\ D &= \frac{|X_{\max}|}{|X_{\min}|} = \frac{2^{2^k - 1}}{2^{-2^k}} = 2^{2^k} \cdot 2^{2^k - 1} = 2^{2^k - 1 + 2^k} = 2^{2^{k+1} - 1}. \end{aligned}$$

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:

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

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

Наприклад, при $D = 64$ біта, $f_{\text{ext}} = 200$ 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 $N_{\text{КОП}} = \lceil \log_2 K \rceil$, 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 $N_{\text{адреси}} = \lceil \log_2 \text{Mem} \rceil$, 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:

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

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

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

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

У цьому завданні потрібно розробити структуру машинних команд основної групи та команд для обміну даними між верхівкою стека співпроцесора та пам'яттю з використанням базової та базово-індексної адресації операндів замість примусової адресації команд в попередньому завданні. Потрібно визначити довжину машинного коду для кожної можливої структури.

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 уточнюються тип 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 коді 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.

Далі 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 передається через пряму адресацію, то довжина адресної частини повинна бути не меншою за $\lceil \log_2 Mem \rceil$. Якщо операнд передається як безпосереднє значення, то довжина відповідної адресної частини повинна бути не меншою за R. Якщо для передачі адреси використовуються базова та базово-

індексна адресація, то довжина адресної частини визначається кількістю регістрів і пар регістрів, які використовуються для задання 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

Для виконання операції Ор над двійковими числами з фіксованою комою, які зберігаються в пам'яті в С коді, потрібно виконати наступні дії:

- Створити структурну схему операційного блока АЛП на Ф-рівні.
- Створити структурну схему операційного блока АЛП на ФС-рівні.

- Створити наступні елементи відповідно до розроблених структурних схем:
 - 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.

- Створити принципову схему та креслення операційного блока АЛП. Передбачити формування значень прапорців регістра ознаки результату та їх мікропрограму обробку, якщо це необхідно.

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.

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

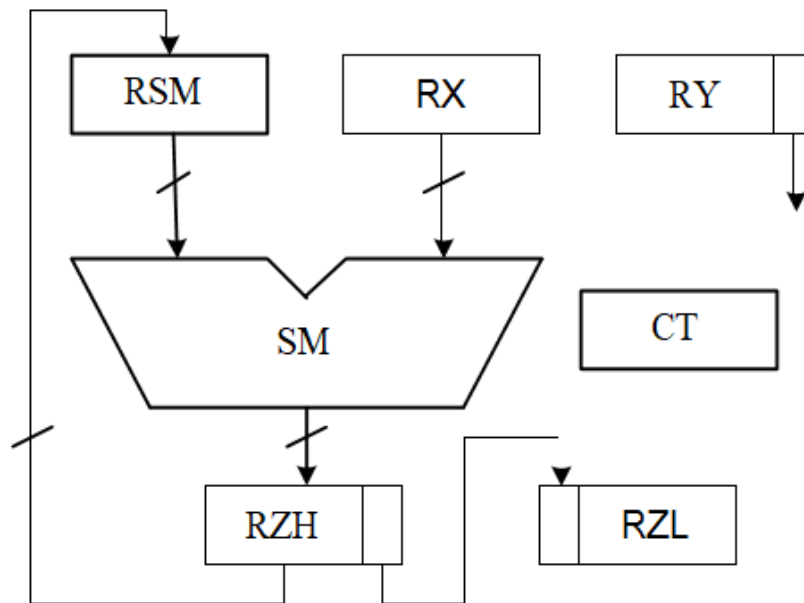


Рис. 4.1 – Структурна схема операційного блока на Φ -рівні

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 Φ -рівні в змістовній формі зображено на Рис. 4.2.

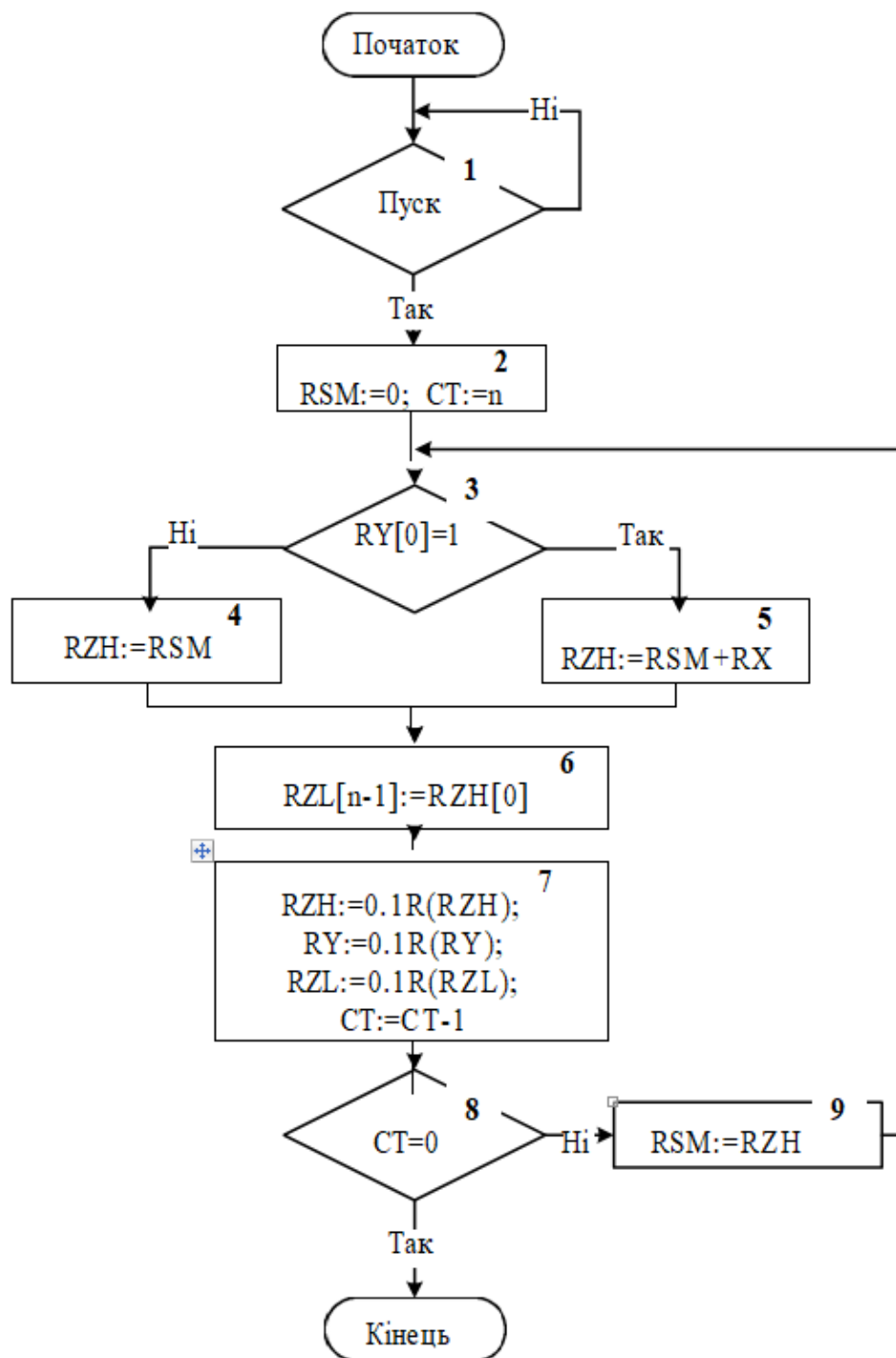


Рис. 4.2 – Граф-схема алгоритму (ГСА) операції на Φ -рівні
в змістовній формі

Для переходу до опису мікроалгоритму на Φ -рівні виконаємо деталізацію схеми (Рис. 4.3) і складемо таблицю мікрооперацій мікроалгоритму.



Така схема дозволяє спростити реалізацію операції зрушення суми часткових добутків і поєднати вершини 6 і 7 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 вправо;
- y_6 – запис у RZH;
- y_7 – видача з RZH у прямому коді;
- y_8 – установлення початкового значення CT;
- y_9 – декремент CT;
- y_{10} – запис у RZL

Таблиця 4.1

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 Рис. 4.4, а логічна схема алгоритму має вигляд:

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

де y_i – сигнали мікрооперацій (Таблиця 4.1), X_1 – логічна умова $CT=0$

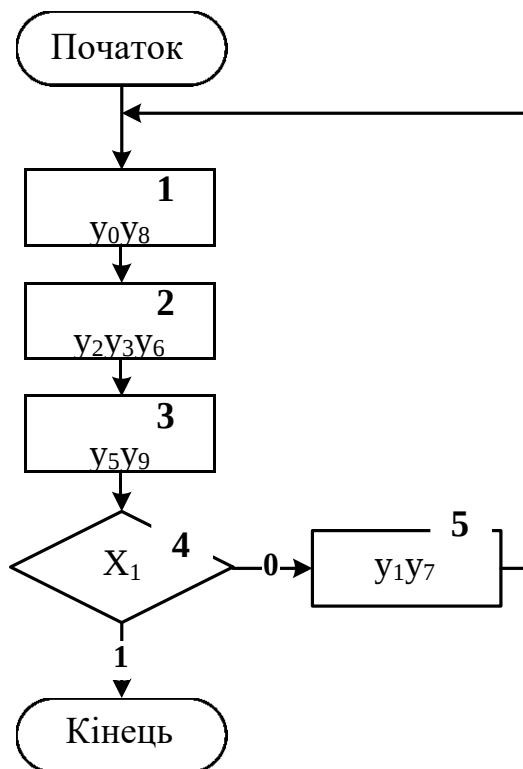


Рис. 4.4 – Граф-схема алгоритму операції на ФС-рівні в закодованій формі

Таблиця 4.2

Карта мікропрограмування для БМУ з примусовою адресацією мікрокоманд з використанням горизонтального методу мікропрограмування

№	Адреса	Мікрокоманда										4		
		1		2										
	М	К	y0	y1	y2y3	y4y5	y6	y7	y8y9					
Н	0 0 0 0	1 1	0 0 0	0	0	0	0	0	0	0	0	0		
1	0 0 0 1	0 0	0 0 1	1	0	0	0	0	0	0	0	1	0	
2	0 0 1 0	1 1	0 0 1	0	0	1	1	0	0	1	0	0	0	
3	0 0 1 1	0 0	0 1 0	0	0	0	0	0	1	0	0	0	1	
4	0 1 0 0	0 1	0 1 1	0	0	0	0	0	0	0	0	0	0	
5	0 1 1 0	0 0	0 0 0	0	1	0	0	0	0	0	0	1	0	0
К	0 1 1 1	0 0	0 0 0	0	0	0	0	0	1	0	0	0	0	1

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'єднаного блока можна додати спеціальний окремий блок, або реалізувати їх на рівні апаратних можливостей будь-якої частини об'єднаного 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.

РОЗДІЛ 4

Захист індивідуального завдання

Захист роботи є особливою формою перевірки якості її виконання та знань у цій галузі.

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 роботи будуть виявлені грубі порушення ЄСКД, або виявиться, що спроектований пристрій принципово непрацездатний, робота оцінюється незадовільно і повертається на доопрацювання.

Студент повинен здати роботу в установлені строки та захистити її.

РОЗДІЛ 5

Індивідуальне завдання для студентів

Для визначення варіанту завдання необхідно скористатися номером залікової книжки. Цей номер потрібно перевести з десяткової системи числення в двійкову систему. Це можна зробити, подаючи номер у вигляді послідовності двійкових розрядів ($b_{n-1} \dots b_2 b_1 b_0$), де b_i - це значення відповідного розряду. Якщо номер залікової книжки дорівнює 978498 в десятковій системі числення, то його можна перевести в двійкову систему числення за допомогою бітових операцій. Наприклад, номер залікової книжки у двійковій системі числення виглядатиме наступним чином: $(11101110111001000010)_2$.

Таблиця 5.1 призначена для визначення варіанту. Необхідно використовувати номери залікової книжки у двійковій системі числення. Для визначення варіанту необхідно скопіювати значення останніх восьми біт номера залікової книжки, що відповідають бітам $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$, відповідно до таблиці 5.1. Наприклад, якщо номер залікової книжки дорівнює $(978498)_{10}$, а його двійкове представлення - $(11101110111001000010)_2$, то для визначення варіанту потрібно скопіювати значення бітів $b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0$, тобто 00101110. Варіант, що відповідає цим бітам, буде використовуватися для вирішення завдань.

Таблиця 5.1

Індивідуальні завдання

$b_3 b_2 b_1 b_0$	f_{ext} , Гц	R/D/A	Reg	τ , такт	K
1	2	3	4	5	6
0000	66	32/16/22	8	5	98
0001	100	32/32/24	12	3	127
0010	133	32/16/32	16	2	164
0011	166	32/32/32	32	1	198
0100	200	32/64/36	48	1	230
0101	33	24/16/20	64	7	100
0110	66	32/32/28	32	3	147
0111	100	32/64/32	48	2	286
1000	133	64/64/48	64	1	312
1001	166	64/32/64	20	1	287
1010	200	64/32/32	128	2	268
1011	66	24/16/24	12	4	154
1100	100	32/32/32	32	5	232
1101	166	32/64/32	48	6	293
1110	200	32/64/32	64	2	326
1111	333	64/64/64	128	1	348

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Матвієнко М.П. Архітектура комп'ютерів: Навчальний посібник / Матвієнко М.П., Розен В.П., Закладний О.М. – К.: Ліра-К, 2019. – 264 с.

Допоміжна література

1. Архітектура обчислювальних машин. Методичні рекомендації до домашньої роботи / Уклад.: В.М. Єфимець, Є.В. Красовська. – К.: НАУ, 2011. – 36с.
2. Кондратенко Ю.П., Сидоренко С.А., Підпригора Д.М. “Поведінковий синтез цифрових пристроїв в середовищі Active-HDL” - Миколаїв, Миколаївський державний гуманітарний університет імені Петра Могили, 2002. – 116 с.
3. Методичні вказівки до використання мови VHDL для опису і моделювання цифрових електронних схем при виконанні лабораторних і курсових робіт з дисциплін "Схемотехніка ЕОМ та елементи і схеми комп'ютерних систем", "ЕОМ і мікропроцесорні системи". Для студентів спеціальностей 6.0804.00 "Інформаційні управляючі системи та технології", "Інформаційні технології проектування" /Укл. О.А.Щербина, Київ: КНУБА, 1999. – 35 с.
4. Болдаков О.І., Красовська Г.В., Голенков В.Г., Красовський К.М. Операційні системи та системне програмування: Мова програмування Асемблер: Конспект лекцій. – К.:КНУБА, 2003. – 80 с.
5. Комп'ютерна схемотехніка та архітектура комп'ютерів. Лабораторний практикум/ Укладачі В.М. Єфимець, Є.В. Красовська та ін. – К: НАУ, 2013. - 64с.
6. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. – С. 228 – 275.
7. Воробйова О.М. Цифрові пристрої: навч. посіб. – Ч. 1. / О.М. Воробйова, М.П. Савицька., Ю.В. Флейта – Одеса: ОНАЗ ім. О.С. Попова, 2016. – Ч. 1. – С. 112.
8. Advanced Micro Devices AMD64 Architecture Programmer's Manual Vol. 1: Application Programming. URL: http://developer.amd.com/wordpress/media/2012/10/24592_APM_v11.pdf (дата звернення: 17.03.2018)
9. Advanced Micro Devices AMD64 Architecture. Programmer's Manual. Vol. 2: System Programming. URL:

- http://developer.amd.com/wordpress/media/2012/10/24593_APM_v21.pdf (дата звернення: 17.03.2018). 156
10. Intel® 64 and IA-32 Architectures Software Developer's Manual. URL: <https://software.intel.com/sites/default/files/managed/39/c5/325462-sdm-vol-1-2abcd-3abcd.pdf> (дата звернення: 17.03.2018)
11. Advanced Micro Devices AMD64 Architecture. Programmer's Manual. Vol. 3: General-Purpose Instruction Reference URL: <http://www.scs.stanford.edu/05aucs240c/lab/amd64/AMD64-3.pdf> 16. Introduction to x64 Assembly. URL: <https://software.intel.com/en-us/articles/introduction-to-x64-assembly>