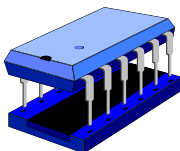


Міністерство освіти і науки України  
Національний університет «Запорізька політехніка»

**АРХІТЕКТУРА КОМП'ЮТЕРА  
ТА НИЗЬКОРІВНЕВЕ ПРОГРАМУВАННЯ**

**програма, методичні вказівки та  
контрольні завдання**

**для студентів спеціальності  
122 «Комп'ютерні науки»  
заочної форми навчання**



Архітектура комп'ютера та низькорівневе програмування.  
Програма, методичні вказівки та контрольні завдання для студентів спеціальності 122 «Комп'ютерні науки» заочної форми навчання / Уклад. С.К. Корнієнко, Запоріжжя: НУ «Запорізька політехніка», 2021. – 30 с.

Укладач: С.К. Корнієнко, доцент, к.т.н.

Рецензент: В.Ф. Оніщенко, доцент, к. ф.-м.н.

Відповідальний  
за випуск: С.О. Субботін, професор, д.т.н.

Затверджено  
на засіданні кафедри  
програмних засобів

Протокол № 1  
від 18.08.2020 р.

## ЗМІСТ

1	Мета й задачі дисципліни .....	4
2	Загальні вказівки .....	5
3	Програма дисципліни й методичні вказівки по її вивченню .....	6
3.1	Загальна характеристика мікропроцесорних засобів .....	6
3.2	Представлення інформації в мікропроцесорних системах .....	8
3.3	Архітектура однокришталю мікропроцесора i8080 .....	12
3.4	Загальні питання побудови мікропроцесорних систем .....	14
4	Контрольна робота .....	19
4.1	Загальні вказівки .....	19
4.2	Завдання на контрольну роботу та методичні вказівки по її виконанню .....	19
4.3	Методичні вказівки по оформленню контрольних робіт .....	20
	Література .....	22
	Додаток А Система команд мікропроцесора i8080 .....	23
	Додаток Б Словник термінів .....	29

# 1 МЕТА Й ЗАДАЧІ ДИСЦИПЛІНИ

## 2. Мета та задачі навчальної дисципліни

Мета: вивчення загальних принципів побудови архітектури сучасних комп'ютерів.

Основними задачами вивчення дисципліни є:

- формування у студентів знань, достатніх для самостійного освоєння обчислювальних систем з новими архітектурами;
- ознайомлення з технічними (апаратними), програмними й технологічними рішеннями, що використовуються для опису й розробки комп'ютерів;
- надбання практичних навичок написання низькорівневих програм мовою асемблера, у тому числі для програмування апаратних ресурсів комп'ютерів.

У результаті вивчення навчальної дисципліни студент повинен отримати **загальні компетентності**: ЗК2. Здатність застосовувати знання у практичних ситуаціях.

**фахові компетентності**:

- СК8. Здатність проєктувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування;
- СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника;
- СК12. Здатність забезпечити організацію обчислювальних процесів в інформаційних системах різного призначення з урахуванням архітектури, конфігурування, показників результативності функціонування операційних систем і системного програмного забезпечення.

**Очікувані програмні результати навчання:**

ПР13. Володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем.

## 2 ЗАГАЛЬНІ ВКАЗІВКИ

Працювати над дисципліною необхідно систематично, вивчаючи теми в зазначеному порядку. При цьому необхідно користуватися не тільки основною, але й додатковою літературою.

Виклад матеріалу в навчальній літературі по темах зазвичай не завжди точно відповідає програмі дисципліни. Тому рекомендується вивчати матеріал вибірково, керуючись програмою й не обмежуючись одним джерелом.

При самостійному проробленні матеріалу варто також враховувати, що контрольні завдання складені так, щоб результат їхнього виконання був покладений в основу виконання лабораторного практикуму.

Підведення підсумків вивчення дисципліни здійснюється у вигляді іспиту. До здачі іспиту допускаються студенти, які успішно виконали всі контрольні завдання та лабораторний практикум.

## 3 ПРОГРАМА ДИСЦИПЛІНИ Й МЕТОДИЧНІ ВКАЗІВКИ ПО ЇЇ ВИВЧЕННЮ

### 3.1 Загальна характеристика мікропроцесорних засобів

*Класифікація мікропроцесорних засобів. Принципи побудови МПС. Поняття архітектури. Фоннейманівська та гарвардська архітектура.*

Мікропроцесор (МП) уявляє собою функціонально завершений програмно-керований пристрій цифрової обробки інформації, виконаний у вигляді однієї чи декількох мікропроцесорних ВІС (Великих Інтегральних Схем).

Програмно-апаратний принцип побудови МПС є одним з основних принципів їхньої організації й полягає в тім, що реалізація цільового призначення МПС досягається не тільки апаратними засобами, але й за допомогою програмного забезпечення - організованого набору програм і даних.

На сучасному етапі розвитку мікропроцесорної техніки практично будь-яку МПС можна віднести до одного з трьох класів:

- системи на базі секційних мікропроцесорних ВІС із мікропрограмним керуванням;
- системи на основі однокристальних МП із програмним керуванням;
- системи зі скороченим набором команд.

Основними принципами, які використовуються при побудові МПС, є модульність, магістральність і мікропрограмуємість.

Модульна організація систем. Модульний принцип передбачає побудову МПС з окремих функціонально, електрично та конструктивно закінчених пристроїв, які називаються *модулями*.

Магістральний спосіб обміну інформацією. Суть його полягає в передачі інформації за допомогою шин (або магістралей), що представляють собою сукупність ліній передачі, які об'єднані загальним функціональним призначенням. У залежності від виду переданої інформації шини підрозділяються на шини даних, адреси й керування.

Мікропрограмна організація керування. Для сучасних систем обробки інформації характерна багаторівнева організація програмного керування. Принцип мікропрограмного керування забезпечує найбільшу гнучкість при організації багатофункціональних мікропроцесорних модулів.

*Архітектура МПС* – це функціональні можливості апаратних засобів МПС, які використовуються для представлення програм і даних, а також для керування процесом обчислень.

Поняття архітектури МП відбиває його структуру, способи звертання до усіх доступних для користувача елементів структури, а також систему команд, режими адресації, формати та довжину даних.

У МПС основними типами архітектури є *фоннейманівська* та *гарвардська*. У першому типі архітектури для збереження даних і команд використовується єдиний простір пам'яті, й немає ніяких ознак, що вказують на тип збереженої інформації. Зміст чарунки пам'яті інтерпретується самим мікропроцесором.

У гарвардській архітектурі для збереження програм і даних виділяються окремі сегменти (області) пам'яті.

### **Контрольні питання**

- Що розуміється під мікропроцесорною системою?
- Які особливості однокришталного мікропроцесора?
- Які особливості секційних мікропроцесорів?
- Що таке однокриштална мікро-ЕОМ?
- Що розуміється під архітектурою обчислювальної системи?
- Які відмінності гарвардської архітектури від фоннейманівської?
- Що називається магістраллю МПС?

### 3.2 Представлення інформації в мікропроцесорних системах

#### *Кодування інформації. Формати даних і команд. Методи адресації.*

##### **Типи та формати даних**

Розглянемо дані питання на прикладі мікропроцесора i8080. Типи та формати даних, які обробляються мікропроцесором, показані на рисунку 3.1.

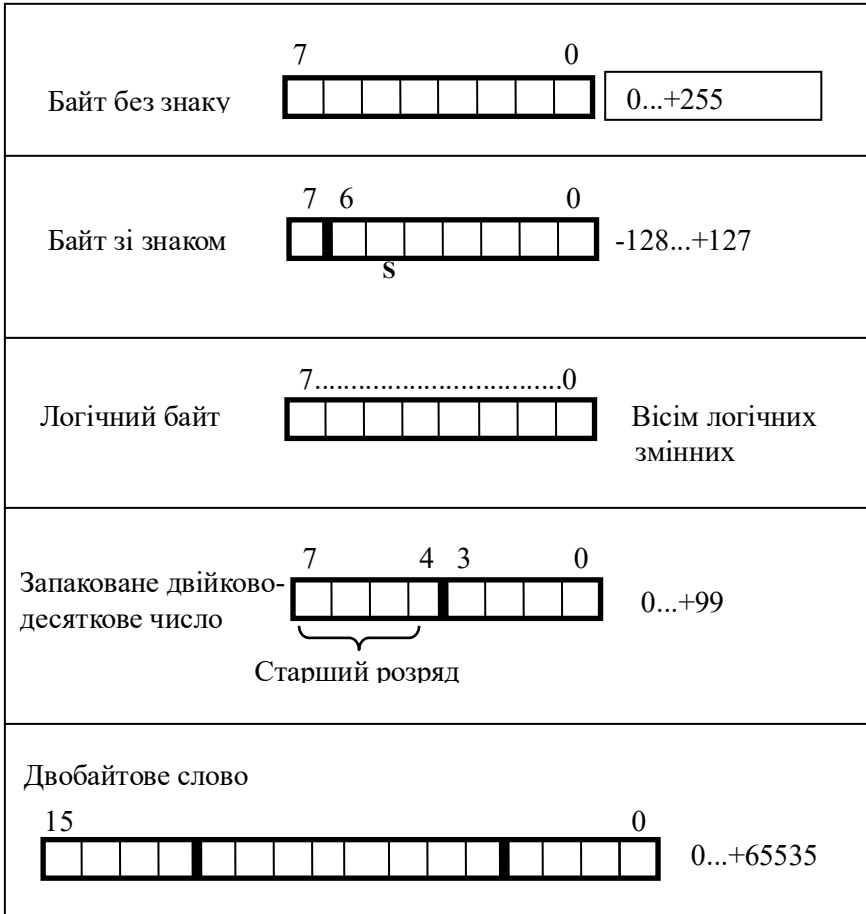
Над двійковими байтами виконуються операції додавання, вирахування, збільшення, зменшення на одиницю; над двійково-десятьковими - операції додавання, збільшення на одиницю. Над логічними даними виконуються логічні операції **І**, **АБО**, **ВИКЛЮЧАЮЧЕ АБО**, **НЕ**, циклічні зсуви. Над двобайтовими словами виконуються арифметичні операції додавання, збільшення та зменшення на одиницю.

Арифметичні операції виконуються за правилами двійкової арифметики над числами в додатковому коді. Двійково-десятькові числа складаються в двійковому коді, а потім операцією корекції код результату з двійкового перетворюється в двійково-десятьковий. Логічні операції виконуються за правилами двійкової логіки.

##### **Формат команд**

Мікропроцесор i8080 має одноадресний формат команд. Команди представляються одним, двома чи трьома байтами (рисунок 3.2). Однобайтовий формат використовується для кодування команд звертання до регістрів надоперативного запам'ятовуючого пристрою (НОЗП), непрямої адресації ЗП, при роботі зі стеком. Двох- і трибайтовий формати використовуються для кодування команд із безпосередньою та прямою адресацією.





**Рисунок 3.1 – Типи та формати команд мікропроцесора I8080**

В другому байті двобайтового формату вказується байт даних або 8-розрядна адреса зовнішнього пристрою, а в трибайтовому форматі в другому й третьому байтах вказується двобайтове слово даних або 16-розрядна адреса ЗП.

Багатобайтові команди зберігаються в сусідніх чарунках пам'яті й адресуються по першому байту, причому двобайтові слова даних і адрес розташовуються в порядку зростання адрес ЗП - спочатку молодший, а потім старший байт.

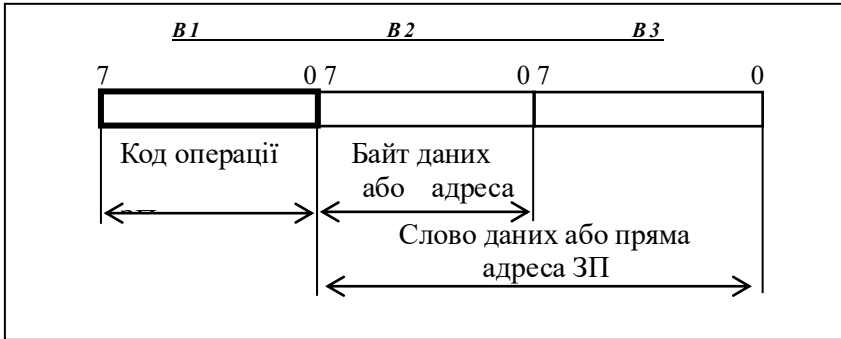


Рисунок 3. 2 - Формат команд МП І8080

### Методи адресації

У МП і8080 використовуються наступні методи адресації:

- неявна ;
- пряма;
- безпосередня;
- непряма.

### Неявна адресація

При цьому методі адресації адреса модуля МПС, із яким працює дана команда, не згадується ніде й визначається кодом операції.

Неявна адресація використовується:

- для звертання до акумулятора. Наприклад, команда **СМА** (інвертувати вміст акумулятора);
- для звертання до прапорців регістра ознак, наприклад, **STC** (установити біт переносу);
- для адресації тригера переривань (команди заборони та дозволу переривань).

### Пряма адресація

При цьому методі адресації в команді задається адреса елемента МПС, який використовується під час виконання операції. Пряма адресація використовується:

- для адресації пам'яті. У цьому випадку виконавча адреса записується в другому та третьому байтах команди, наприклад, **STA 1000H** - завантажити чарунку пам'яті з адресою **1000H** вмістом акумулятора;

- для адресації регістрів зовнішніх пристроїв (ЗвП). У цьому випадку виконавча адреса записується в другому байті команди, наприклад, **OUT 05H** - завантажити регістр зовнішнього пристрою під номером **5** вмістом акумулятора;

- для адресації регістрів і регістрових пар МП. У цьому випадку виконавча адреса записується в поле коду операції, наприклад, **MOV A,B** - передати вміст регістра **B** до регістра **A** (акумулятора).

### Безпосередня адресація

При цьому методі адресації в команді безпосередньо вказується потрібна константа. Безпосередня адресація використовується в операціях завантаження регістрів і регістрових пар, таких як

**MVI B,05H** - завантажити до регістра **B** число **05H**;

**LXI H,FD16H** - завантажити до регістрової пари **HL** число **FD16H**.

### Опосередкована адресація

При цьому методі адресації виконавча адреса попередньо формується в одній з регістрових пар **B, D, H**. Наприклад, **STAX D** - записати вміст акумулятора за адресою, яка зберігається в регістровій парі **D**.

Різновидом опосередкованої адресації є адресація з автоіндексуванням, при якій значення адреси, що зберігається в покажчику стека **SP**, автоматично збільшується або зменшується на 2 після звертання до стека, наприклад, **PUSH PSW**.

### Контрольні питання

- Які операції виконуються над одно- та двобайтовими даними?
- Які типи даних використовуються в МП i8080?
- Які формати команд використовуються в МП i8080?
- Яким чином розташовуються в пам'яті багатобайтові команди?
- Дайте порівняльну характеристику методів адресації.

### 3.3 Архітектура однокришталного мікропроцесора i8080

*Блок реєстрів. Організація стека. Блок АЛП. Блок керування. Слово стану. Системний контролер. Машинні цикли й такти. Виконання команд.*

При розгляді структури мікропроцесора треба особливу увагу приділити програмно-доступним реєстрам, оскільки вони використовуються програмістом при написанні програм керування МПС.

Ці реєстри включають шість 16-розрядних реєстрів (рис. 3.3).

Три реєстри загального призначення ( **BC**, **DE**, **HL** ) можна адресувати побайтово або як реєстрові 16-розрядні пари послівно. У першому випадку реєстри використовуються для збереження байтових даних, у другому - для збереження адрес або двобайтових слів даних.

Старший байт реєстра слова стану програми **PSW (Program Status Word)** використовується як акумулятор, молодший байт - для збереження ознак.

16-розрядний покажчик стека **SP (Stack Pointer)** призначений для збереження адреси вершини стека.

16-розрядний програмний лічильник **PC (Program Counter)** використовується для природної адресації команд. Для примусової адресації команд використовується безпосередня адресація з 16-розрядною адресою, що дозволяє виконувати переходи в межах всього адресного простору ЗП.



Рисунок 3.3 - Регістри мікропроцесора

Усі дії мікропроцесора синхронізуються вкладеними друг у друга циклами трьох рівнів: командним, машинним і мікротактом.

Командний цикл складається з фази вибірки й фази виконання.

Машинний цикл - це період часу, необхідний для звернення до пам'яті для читання або запису одного байта інформації.

Машинний такт (мікротакт) дорівнює періоду проходження синхроімпульсів фази C1.

### Контрольні питання

- Технічні характеристики мікропроцесора i8080?
- Що таке регістри загального призначення?
- Які функції програмного лічильника?
- Яке призначення регістра ознак?
- Що таке "слово стану програми", його призначення?

- Призначення АЛП?
- Призначення слова стану процесора?
- Які функції системного контролера?
- Що називається командним і машинним циклом, мікротактом?
- Формати команд мікропроцесора i8080?
- Методи адресації?
- Призначення основних команд мікропроцесора?
- Як організований стек у мікропроцесорній системі на базі мікропроцесора i8080?

### 3.4 Загальні питання побудови мікропроцесорних систем

**3.4.1 Система пам'яті МПС. Основні характеристики запам'ятовуючих пристроїв (ЗП). Статичні та динамічні ОЗП. Постійні ЗП. Організація кеи-пам'яті. Зовнішні ЗП на магнітних носіях. Фізична й логічна структура магнітного диску.**

Система пам'яті МПС являє собою багаторівневу ієрархічну структуру, що включає надоперативний ЗП (НОЗП), основну пам'ять (ОП), зовнішні запам'ятовуючі пристрої та буферну пам'ять.

НОЗП знаходиться всередині мікропроцесора й складається з регістрів загального призначення (РЗП). Для МП i8080 - це регістри **В, С, D, E, H, L**.

Основна пам'ять складається з оперативного запам'ятовуючого пристрою (ОЗП) і постійного запам'ятовуючого пристрою (ПЗП).

ОЗП являє собою енергозалежну пам'ять з довільним доступом, тобто при відключенні живлення інформація в ОЗП руйнується, але час звертання до пам'яті не залежить від адреси чарунки пам'яті.

У свою чергу, ОЗП підрозділяються на статичні та динамічні. Чарунка *статичного* ОЗП являє собою тригер (пристрій із двома стійкими станами). Інформація, збережена в чарунці пам'яті, визначається станом тригера ( 0 або 1 ).

Динамічна пам'ять будується на основі конденсаторів (вхідні ємності МОН-транзисторів). Стан чарунки визначається наявністю або відсутністю заряду на ємності. Через наявність струмів витоку записану інформацію необхідно періодично оновлювати (регенерація

пам'яті), що вимагає додаткового часу. Однак, з огляду на більш низьку вартість динамічних ОЗП, саме на них реалізується оперативна пам'ять комп'ютерів .

Постійні ЗП являють собою енергонезалежні запам'ятовуючі пристрої, призначені для тривалого збереження інформації, що не руйнується після відключення живлення.

ПЗП підрозділяються на:

- матричні, які програмуються на заводі-виробнику;
- програмовані користувачем;
- репрограмовані, що допускають багаторазовий перезапис інформації на спеціальному пристрої (програмагорі).

До зовнішніх запам'ятовуючих пристроїв (ЗЗП) відносяться накопичувачі на магнітних стрічках, магнітних і оптичних дисках. Вони характеризуються низькою вартістю біта збереженої інформації та меншою швидкістю в порівнянні з основною пам'яттю.

Незважаючи на відносно високу швидкість ОЗП, мікропроцесор при звертанні до пам'яті змушений велику частку часу працювати в режимі чекання, що знижує загальну продуктивність роботи комп'ютера.

Тому між процесором і ОЗП включають кеш-пам'ять, яка представляє собою швидкісною статичну пам'ять, що дозволяє скоротити час звертання до основної пам'яті.

Інформація на магнітних дисках записується на концентричних доріжках, розділених на сектори. Оскільки жорсткі магнітні диски складаються з декількох дисків, вводиться поняття "циліндра", що означає сукупність доріжок усіх поверхонь, що мають однаковий номер. Адреса інформації визначається номерами доріжки (циліндра), сектора й поверхні.

Жорсткі магнітні диски умовно розбиваються на логічні диски, з якими користувач працює, як з окремими фізичними дисками.

### **Контрольні питання**

- З чого складається НОЗП?
- Чим відрізняються ОЗП від ПЗП?
- Які типи ОЗП Ви знаєте?
- Які типи ПЗП Ви знаєте?

- На якому принципі заснована робота кеш-пам'яті?
- Яка фізична структура диска?
- Що таке логічний диск?

***3.4.2 Інтерфейси МПС. Класифікація. Основні принципи організації. Паралельний та послідовний інтерфейси. Інтерфейс із загальною й ізольованою шиною. Апаратні засоби інтерфейса. Порти вводу/виводу.***

Під інтерфейсом розуміється сукупність апаратних і програмних засобів, за допомогою яких відбувається об'єднання компонентів МПС, забезпечуючи їхню функціональну, електричну і конструктивну сумісність.

По функціональному призначенню інтерфейси підрозділяються на системні інтерфейси, інтерфейси периферійних пристроїв, інтерфейси мультимікропроцесорних систем, інтерфейси локальних обчислювальних систем і інтерфейси інтелектуальної периферії.

За способом передачі інформації інтерфейси підрозділяються на *паралельні* (обмін інформацією ведеться словами) та *послідовні* (обмін інформації ведеться побітно).

За способом використання адресного простору виділяються інтерфейси з ізольованою та загальною шиною. У першому випадку адресні простори пам'яті та вводу/виводу розділяються, й звертання до них здійснюється за допомогою різних груп команд. У другому випадку адресні простори пам'яті та вводу/виводу організуються в єдиній зоні адрес, обумовленій розрядністю адресної шини. При цьому команди вводу/виводу замінюються командами звертання до пам'яті.

Для підключення периферійних пристроїв використовуються так звані порти вводу/виводу, що представляють собою, у загальному випадку, комбінацію дешифратора та шинного формувача.

**Контрольні питання**

- Що таке інтерфейс?
- Що розуміється під функціональною, електричною та конструктивною сумісністю?



- За якими ознаками класифікуються інтерфейси?
- Характеристика інтерфейсу з ізольованою шиною.
- Характеристика інтерфейсу з загальною шиною.
- Яке призначення портів вводу/виводу ?

### ***3.4.3 Програмний обмін інформацією. Синхронний й асинхронний обмін. Обмін інформацією в режимі переривань. Прямий доступ до пам'яті.***

При організації взаємодії МПС із зовнішніми пристроями найбільш часто використовуються три способи обміну: програмний обмін, обмін із перериванням програми, обмін за допомогою прямого доступу до пам'яті.

Перший спосіб обміну ініціюється програмою, що виконується МП, для чого використовуються спеціальні команди вводу/виводу. Два інших способи обміну здійснюються з ініціативи зовнішнього пристрою.

При виконанні програмного обміну спочатку перевіряється готовність зовнішнього пристрою до обміну. При наявності готовності виконується операція вводу/виводу. В іншому випадку МП переходить у стан чекання, в якому буде знаходитися до появи готовності. Звідси випливає основний недолік програмного обміну, пов'язаний з непродуктивними втратами часу процесора на виконання циклів чекання.

Більш ефективним є обмін даними з перериванням програми. Цей вид обміну ініціюється зовнішнім пристроєм (ЗВП), що посилає в МПС спеціальний сигнал "Запит переривання". Після виконання поточної команди МП припиняє виконання основної програми і переходить до виконання підпрограми обміну, по закінченні якої відбувається повернення до основної програми.

У випадку одночасного надходження запитів на переривання від декількох зовнішніх пристроїв (ЗВП) керування передається пристрою, що має найвищий пріоритет.

У разі потреби передачі великих масивів інформації доцільно робити обмін даними між ОЗП і ЗВП безпосередньо, минаючи робочі регістри МП і, зокрема, акумулятор. Такий режим називається *прямим доступом до пам'яті*.

**Контрольні питання**

- Що називається синхронним й асинхронним обміном?
- Характеристика програмного обміну.
- У чому відмінність переривання за вектором від переривань з опитуванням (система полінга)?
  - Яку інформацію необхідно запам'ятовувати в стеці при обслуговуванні переривань?
  - У чому полягає маскування запитів на переривання?
  - Яким чином операції прямого доступу до пам'яті впливають на реальну продуктивність МПС?

## 4 КОНТРОЛЬНА РОБОТА

### 4.1 Загальні вказівки

У процесі вивчення дисципліни "ЕОМ і мікропроцесорні системи" студенти заочної форми навчання повинні виконати контрольну роботу. Основна мета її виконання - закріплення теоретичного матеріалу та розвиток навичок практичного програмування МПС.

Приступати до виконання контрольної роботи слід після вивчення матеріалу по літературі, що рекомендується, а також методичних указівок до контрольної роботи та практичних занять.

Виконана контрольна робота повинна бути представлена на перевірку в суворій відповідності з навчальним графіком. При наявності зауважень рецензента треба виконати всі виправлення та доповнення так, щоб при повторному рецензуванні можна було зіставляти колишнє та нове рішення.

### 4.2 Завдання на контрольну роботу та методичні вказівки по її виконанню

Призначення контрольної роботи міститься в перевірці знань студентів основ програмування мовою асемблера мікропроцесора і8080. При виконанні цієї роботи потрібно скласти програми, які реалізують наступні завдання :

4.2.1 У масиві чисел, розташованому з адреси 0150h до 0200h, встановити в "1" старший і в "0" молодший розряди чисел.

4.2.2 Підрахувати, скільки разів у масиві чисел з адресами 0200-02FF зустрічаються числа, менші, ніж перший елемент масиву.

4.2.3 У масиві чисел, які зберігаються в ОЗП у чарунках з адресами з 0500h до 05DEh, замінити від'ємні числа нулем.

4.2.4 Підрахувати число додатних елементів у масиві з адресами 0600h-06AEh.

4.2.5 Здійснити обмін інформації між масивами, розташованими з адрес 0400h і 0500у, які вміщують по 21 елементу.

4.2.6 Програма формування часової затримки тривалістю 0,3 с ( $f = 1,2 \text{ МГц}$ ).

4.2.7 Зсунути вміст чарунки пам'яті з адресою 040F на число бітів, яке визначається вмістом регістру D.

4.2.8 Скласти два числа, які знаходяться у чарунках з адресами 0100h та 0200h, і вивести молодший напівбайт у порт 01, а старший напівбайт - порт 02.

4.2.9 З масиву чисел із адресами з 0400h до 04CDh видавати вміст чарунок із непарними адресами до вихідного порту, помножуючи їх на 2.

4.2.10 Переписати 10 послідових байтів із одної області пам'яті (починаючи з адреси 2200h) до іншої (з адреси 2300h).

4.2.11 Замінити у масиві з адресами 0500h-05CCh числа, більші 9, числом 10.

4.2.12 Масив чисел з адресами з 0500h і 05ECh розсортувати, переписав від'ємні, починаючи з адреси 0600h, а решту - з адреси 0700h.

4.2.13 В масиві чисел, які зберігаються в ОЗП, в чарунках з адресами з 0500h до 05BEh, знайти число 25D і вказати його адресу на вихідних портах 01 і 02. Число в масиві є і воно єдине.

4.2.14 В масиві чисел із адресами з 0500h до 05FFh числа, які зберігаються у чарунках із парними адресами, замінити на 0.

4.2.15 В масиві чисел з адреси 0400h до 04FFh парні числа поділити на 2, непарні - помножити на 4.

При виконанні цієї частини контрольної роботи для всіх програм необхідно складати схему алгоритму відповідно до вимог стандартів, а також виконувати опис програми з урахуванням вимог формату командного рядка Асемблера, тобто правильно заповнювати поле мітки, коду операції, операнда та коментарів.

### **4.3 Методичні вказівки по оформленню контрольних робіт**

Результати виконання кожної контрольної роботи оформлюються у вигляді пояснювальної записки на стандартних аркушах папера формату А4. Набір тексту та його друк повинні бути комп'ютерними. Оформлення роботи повинно бути у відповідності до вимог СТП 15-96 "Стандарт підприємства. Пояснювальна записка до курсових і дипломних проєктів".

Структура пояснювальної записки: титульний лист, основна частина, список посилань.

На титульному листі повинні бути зазначені: повні найменування міністерства, університета та кафедри, прізвище студента, номер групи, шифр (числа: А - рік народження, В - номер залікової книжки) та домашня адреса.

Основна частина включає результати виконання всіх етапів роботи відповідно до її змісту. Кожен етап виконання повинний мати як текстові, так і ілюстративні пояснення. Текст, рисунки й схеми повинні бути оформлені акуратно, з урахуванням вимог діючих стандартів.

Робота повинна бути датована та підписана виконавцем.

Разом із пояснювальної запискою треба додавати дискету з текстами відлагоджених програм.

## ЛІТЕРАТУРА

1. Корнієнко С.К. Архітектура комп'ютера та низькорівневе програмування.: Конспект лекцій для студентів спеціальності 122 «Комп'ютерні науки» / С.К. Корнієнко. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 98 с.

2. Методичні вказівки до лабораторних робіт з дисципліни «Архітектура комп'ютера та низькорівневе програмування» для студентів спеціальності 122 «Комп'ютерні науки» / Укл.: С.К. Корнієнко. – Запоріжжя: НУ «Запорізька політехніка», 2021. – 46 с.

3. Методичні вказівки до виконання самостійної роботи з дисципліни «Архітектура комп'ютера та низькорівневе програмування» для студентів спеціальності 122 «Комп'ютерні науки» усіх форм навчання /Уклад.: Корнієнко С.К. – Запоріжжя: НУ «Запорізька політехніка», 2021. –14 с.



## Додаток А

### Система команд мікропроцесора i8080

Система команд однокришталного мікропроцесора I8080 має команди трьох форматів. Перший байт команди містить інформацію про формат команди, код операції, вид адресації та про реєстри або реєстрові пари, якщо вони приймають участь у виконанні операцій.

У двобайтових командах другий байт (**B2**) містить 8-розрядний операнд або 8-розрядну адресу пристрою вводу чи виводу.

У трибайтових командах другий та третій байти (**B2, B3**) містять 16-розрядні адреси (у командах із прямою адресацією пам'яті) або 16-розрядні операнди (у командах завантаження реєстрових пар або покажчика стека). Другий байт трибайтової команди містить молодший байт числа, третій – старший.

У таблиці А.1 наведені мнемонічні позначення та опис команд мікропроцесора I8080. При цьому використовуються наступні умовні позначення :

- r** - реєстр загального призначення;
- rp** - пара реєстрів загального призначення;
- PC** - програмний лічильник (лічильник команд);
- SP** - покажчик стека;
- M** - чарунка пам'яті;
- B2,B3** - другий та третій байти команд

Таблиця А.1 - Система команд мікропроцесора І8080

Команда	Довжина команди, байт	Число тактів	Опис команди	Ознаки
1	2	3	4	5
<i>1 Команди пересилання даних</i>				
MOV r1,r2	1	5	Пересилання даних із регістра r2 до регістра r1	-
MOV M,r (MOV r,M)	1	7	Пересилання даних із регістра r до пам'яті за адресою, що зберігається у регістровій парі H-L (із пам'яті до регістру r)	-
XCHG	1	4	Обмін даними між парами регістрів H-L і D-E	-
MVI r,<B2> (MVI M,<B2>)	2	7 (10)	Занесення байта даних до регістру r (до пам'яті)	-
LXI rp <2байта>	3	10	Занесення двох байтів даних у пару регістрів (B-C, D-E, H-L, SP). Третій байт команди заноситься в старший регістр, а другий - у молодший	-
LDAX rp	1	7	Завантаження в накопичувач вмісту чарунки, яка опосередковано адресується парою регістрів rp (B-C, D-E)	-
LDA <адреса>	3	13	Завантаження накопичувача вмістом чарунки за вказаною адресою. 2-й байт команди - молодший байт адреси, 3-й байт - старший	-
STAX rp	1	7	Занесення вмісту накопичувача до чарунки, яка опосередковано адресується парою rp	-
STA, <адреса>	3	13	Занесення вмісту накопичувача до чарунки за вказаною адресою	-



## Продовження таблиці А.1

1	2	3	4	5
<b>LHLD &lt;адреса&gt;</b>	3	16	Завантаження регістра <b>L</b> вмістом чарунки за вказаною адресою, а регістр <b>H</b> - чарунки з адресою на одиницю більше	-
<b>SHLD &lt;адреса&gt;</b>	3	16	Занесення вмісту регістрів <b>H</b> і <b>L</b> до пам'яті (аналогічно команді <b>LHLD</b> )	-
<b>2 Арифметичні команди</b>				
<b>ADD r (ADD M)</b>	1	4 (7)	Складання змісту регістра <b>r</b> (чарунки пам'яті) і накопичувача	Z,S,P,C
<b>ADC r (ADC M)</b>	1	4 (7)	Складання змісту регістра <b>r</b> (чарунки пам'яті) і накопичувача з бігом перенесення	Z,S,P, C,AC
<b>SUB r (SUB M)</b>	1	4 (7)	Віднімання змісту регістра <b>r</b> (чарунки пам'яті) від змісту накопичувача	Z,S,P, C <sup>1</sup> ,AC <sup>2</sup>
<b>SBB r (SBB M)</b>	1	4 (7)	Віднімання змісту регістра <b>r</b> (чарунки пам'яті) та біта перенесення від змісту накопичувача	Z,S,P, C <sup>1</sup> ,AC <sup>2</sup>
<b>ADI ,&lt;байт&gt;</b>	2	7	Складання байта зі змістом накопичувача	Z,S,P,C, AC
<b>ACI ,&lt;байт&gt;</b>	2	7	Складання байта зі змістом накопичувача та бігом перенесення	Z,S,P,C, AC
<b>SUI,&lt;байт&gt;</b>	2	7	Віднімання байта із змісту накопичувача	Z,S,P, C <sup>1</sup> ,AC <sup>2</sup>
<b>SBI,&lt;байт&gt;</b>	2	7	Віднімання байта команди і біта перенесення від змісту накопичувача	Z,S,P, C <sup>1</sup> ,AC <sup>2</sup>
<b>DAD rp</b>	1	10	Складання змісту пари регістрів <b>rp</b> ( <b>B-C,D-E,H-L,SP</b> ) зі змістом пари регістрів <b>H-L</b>	C
<b>INR r (INR M)</b>	1	5 (10)	Збільшення змісту регістра <b>r</b> (чарунки пам'яті) на одиницю	Z,S,P, AC
<b>DCR r (DCR M)</b>	1	5 (10)	Зменшення змісту регістра <b>r</b> (чарунки пам'яті) на одиницю	Z,S,P, AC <sup>2</sup>

Продовження таблиці А.1

1	2	3	4	5
<b>INX r</b> <b>(DCX r)</b>	1	5	Збільшення (зменшення) змісту пари регістрів <b>rp</b> ( <b>B-C,D-E,H-L,SP</b> ) на одиницю	-
<b>DAA</b>	1	4	Перетворення змісту накопичувача в двійково-десятичний код	Z,S,P,C, AC
<b>3 Логічні команди</b>				
<b>ANA r</b> <b>(ANA M)</b>	1	4 (7)	Порозрядне 'Г' над змістом регістра <b>r</b> (чарунки пам'яті) і накопичувача	Z,S,P, C=0, AC=0
<b>XRA r</b> <b>(XRA M)</b>	1	4 (7)	Порозрядне виключаюче 'АБО' над змістом регістра <b>r</b> (чарунки пам'яті) і накопичувача	Z,S,P, C=0, AC=0
<b>ORA r</b> <b>(ORA M)</b>	1	4 (7)	Порозрядне 'АБО' над змістом регістра <b>r</b> (чарунки пам'яті) і накопичувача	Z,S,P, C=0, AC=0
<b>CMP r</b> <b>(CMP M)</b>	1	4 (7)	Порівняння змісту регістра <b>r</b> (чарунки пам'яті) і накопичувача	(Z,S,P, C,AC) <sup>3</sup>
<b>ANI,&lt;байт&gt;</b>	2	7	Порозрядне 'Г' над змістом накопичувача і байтом	Z,S,P, C=0, AC=0
<b>XRI,&lt;байт&gt;</b>	2	7	Порозрядне виключаюче 'АБО' над змістом накопичувача і байтом	Z,S,P, C=0, AC=0
<b>ORI,&lt;байт&gt;</b>	2	7	Порозрядне 'АБО' над змістом накопичувача і байтом	Z,S,P, C=0, AC=0
<b>CPI,&lt;байт&gt;</b>	2	7	Порівняння байта зі змістом накопичувача	(Z,S,P, C,AC) <sup>3</sup>
<b>RLC</b> <b>(RRC)</b>	1	4	Циклічний зсув змісту накопичувача вліво (вправо)	C <sup>4</sup>
<b>RAL</b> <b>(RAR)</b>	1	4	Циклічний зсув змісту накопичувача вліво (вправо) через перенесення	C <sup>4</sup>
<b>CMA</b>	1	4	Порозрядне інвертування накопичувача	-
<b>STC</b>	1	4	Встановлення ознаки перенесення в одиницю	C=1
<b>CMC</b>	1	4	Інвертування ознаки перенесення C	C= $\overline{C}$

Продовження таблиці А.1

1	2	3	4	5
<b><i>4 Команди переходів</i></b>				
<b>PCHL</b>	1	5	Занесення змісту регістрів <b>H, L</b> в лічильник команд (зміст <b>H</b> – в старший байт, <b>L</b> – в молодший)	-
<b>JMP,&lt;адреса&gt;</b>	3	10	Безумовний перехід по вказаній адресі	-
<b>JC/(JNC),&lt;адреса&gt;</b>	3	10	Перехід при наявності (відсутності) перенесення	-
<b>JZ/(JNZ),&lt;адреса&gt;</b>	3	10	Перехід при наявності (відсутності) нуля	-
<b>JP/(JM),&lt;адреса&gt;</b>	3	10	Перехід при плюсі (мінусі)	-
<b>JPE/(JPO),&lt;адреса&gt;</b>	3	10	Перехід при парності (непарності)	-
<b>CALL,&lt;адреса&gt;</b>	3	17	Виклик підпрограми	-
<b>CC/(CNC),&lt;адреса&gt;</b>	3	11 (17)	Виклик підпрограми при наявності (відсутності) перенесення	-
<b>CZ/(CNZ),&lt;адреса&gt;</b>	3	11 (17)	Виклик підпрограми при наявності (відсутності) нуля	-
<b>CP/(CM),&lt;адреса&gt;</b>	3	11 (17)	Виклик підпрограми при плюсі (мінусі)	-
<b>CPE/(CPO),&lt;адреса&gt;</b>	3	11 (17)	Виклик підпрограми при парності (непарності)	-
<b>RET</b>	1	10	Повернення з підпрограми	-
<b>RC/(RNC)</b>	1	5 (11)	Повернення при наявності (відсутності) перенесення	-
<b>RZ/(RNZ)</b>	1	5 (11)	Повернення при наявності (відсутності) нуля	-
<b>RP/(RM)</b>	1	5 (11)	Повернення при плюсі (мінусі)	-
<b>RPE/(RPO)</b>	1	5 (11)	Повернення при парності (непарності)	-
<b>RST,&lt;номер&gt;</b>	1	11	Повторне запуснення з адреси 8 x номер (0,8,...,56)	-
<b>5 Команди вводу-виводу та управління</b>				
<b>IN,&lt;порт&gt;</b>	2	10	Ввод даних із вказаного порта до накопичувача	-
<b>OUT,&lt;порт&gt;</b>	2	10	Вивід даних із накопичувача до вказаного порта	-

Продовження таблиці А.1

1	2	3	4	5
<b>PUSH rp</b>	1	11	Занесення змісту пари регістрів <b>rp (B-C,D-E,H-L, PSW)</b> до стека	-
<b>POP rp</b>	1	10	Видання даних зі стека в пару регістрів <b>rp (B-C,D-E, H-L,PSW)</b>	(Z,S,P,C AC) <sup>6</sup>
<b>XTHL</b>	1	18	Обмін даними між верхівкою стека та парою регістрів <b>H-L</b>	-
<b>SPHL</b>	1	5	Занесення в показчик стека змісту регістрів <b>H-L</b>	-
<b>DI/EI</b>	1	5	Заборонити/дозволити переривання	-
<b>NOP</b>	1	4	Порожня операція	-
<b>HLT</b>	1	7	Зупин	-

**Примітки:**

1. Встановлюється при наявності займу до старшого розряду, в протилежному випадку скидається.

2. Встановлюється при наявності займу зі старших чотирьох розрядів в молодші, в протилежному випадку скидається.

3. **Z** встановлюється, якщо зміст регістра та байта даних дорівнює змісту накопичувача ;

**S,C**, якщо зміст регістра або байта даних більше змісту накопичувача;

**AC**, якщо зміст молодших чотирьох розрядів регістра і байта даних більше змісту молодших чотирьох розрядів накопичувача;

**P**, якщо байт різниці між змістом накопичувача та змістом регістра або байта даних містить парне число одиниць;

4. Стан ознаки дорівнює значенню висунутого з накопичувача двійкового розряду.

5. В знаменнику дробу вказано кількість тактів при виконанні умов, в чисельнику – при невиконанні.

6. За командою **POP PSW** ознаки встановлюються відповідно до значення розрядів слова, яке занесене до стека, при інших значеннях гр ознаки не змінюються.

## Додаток Б

### Словник термінів

**Алгоритм** - упорядкований набір дій для рішення задач, або план рішення конкретної задачі.

**Арифметико-логічний пристрій (АЛП)** - основний пристрій мікропроцесора для виконання арифметичних і логічних операцій.

**Асемблер** - програма для перетворення тексту прикладної програми, що містить математичні забезпечення команд ЕОМ і операндів, у машинні коди.

**Інтерфейс** - сукупність технічних засобів і правил для взаємодії пристроїв обчислювальної техніки.

**Код операції** - комбінація двійкових знаків, яка сприймається мікропроцесором при виконанні визначеної операції. Зазвичай команда складається або з коду операції, або з коду операції плюс дані, або з коду операції плюс адреса даних.

**Лічильник команд** - регістр, що містить адресу чергової частини команди, що підлягає виконанню.

**Машинний такт** - часовий інтервал, протягом якого мікропроцесор виконує елементарну дію.

**Машинний цикл** - часовий інтервал, протягом якого мікропроцесор здійснює одне звертання до пам'яті або порту вводу/виводу.

**Мікропроцесор** - програмно-керуємий пристрій, який призначений для обробки інформації та виконаний на одній або декількох інтегральних схемах.

**Мультиплексна передача** - система передачі інформації від декількох пристроїв по одному каналу.

**Операнд** - дані, над якими виконується математична чи логічна операція.

**Оперативний запам'ятовуючий пристрій (ОЗП)** - запам'ятовуючий пристрій ЕОМ, що забезпечує можливість швидкої зміни збереженої інформації та служить для збереження результатів проміжних обчислень, а також програм, які виконуються в поточний час.

**Показчик стека** - реєстр, який керує зверненнями до стека.

**Порт** - функціональний пристрій для організації вводу або виводу даних із зовнішніх пристроїв (у зовнішні пристрої).

**Постійний запам'ятовуючий пристрій (ПЗП)** - запам'ятовуючий пристрій з незмінним умістом.

**Прапорець** - тригер, що може встановлюватися чи скидатися в залежності від результату виконання операції.

**Прямий доступ до пам'яті (ПДП)** - спосіб доступу, при якому здійснюється взаємний обмін даними між зовнішніми пристроями й без участі мікропроцесора.

**Реєстр** – електроний вузол, призначений для зберігання одного слова інформації.

**Слово** - сукупність двійкових розрядів, яка сприймається при обробці як єдине ціле.

**Стек** - область пам'яті, яка адресується за допомогою реєстра - показчика стека. Звертання до стека виконується за принципом "останній, що надійшов, обслуговується першим".