

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Кафедра електропостачання та енергетичного менеджменту

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІНИ
«ІНФОРМАЦІЙНІ І КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ
В ЕЛЕКТРОІНЖЕНЕРІЇ»
для студентів спеціальності
141 «Електроенергетика, електротехніка та електромеханіка»

Одеса -2024

Міністерство освіти і науки України
Національний університет «Одеська політехніка»
Кафедра електропостачання та енергетичного менеджменту

МЕТОДИЧНІ ВКАЗІВКИ
ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ З ДИСЦИПЛІНИ
«ІНФОРМАЦІЙНІ І КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ
В ЕЛЕКТРОІНЖЕНЕРІЇ»
для студентів спеціальності
141. «Електроенергетика, електротехніка та електромеханіка»

Затверджено
на засіданні кафедри
електропостачання та
енергетичного менеджменту
протокол №2
від 18.09.2024

Одеса - 2024

Методичні вказівки до виконання лабораторних робіт з дисципліни «Інформаційні і комунікаційні технології в електроінженерії» для студентів спеціальності 141. «Електроенергетика, електротехніка та електромеханіка», Укл.: В.Л.Біляєв.- Одеса: Національний університет «Одеська політехніка», 2024 – 28 с.

Укладач: В.Л. Біляєв, канд. техн. наук, доцент.

ВСТУП

Сучасні системи електропостачання промислових підприємств є складними системами, які містять велику кількість елементів, пов'язаних між собою як детермінованими, так і імовірнісними зв'язками. Ефективне управління режимами таких електроенергетичних систем неможливе без використання сучасних засобів виміру, контролю, і управління. Основними вимогами, що пред'являються до таких засобів, є: низька питома вартість на операцію в секунду, надійність, широкі функціональні можливості, здатність роботи в умовах промислових цехів.

Мікропроцесорна елементна база дозволяє будувати промислові контролери з дуже високою надійністю, які розміщуються на одній або декількох платах невеликого розміру і легко вбудовуються в технологічне устаткування.

Винахід мікропроцесора є революційною подією в технології виробництва великих інтегральних схем (ВІС), в обчислювальній техніці і схемотехніці. Мікропроцесор вирішив протиріччя між високою мірою інтеграції ВІС і великою кількістю різних типів ВІС, яке потрібне для створення широкого класу обчислювальних пристроїв.

Перераховані чинники вказують на необхідність оволодіння основами теорії і навичками програмування і експлуатації мікропроцесорних систем.

Мікропроцесор - побудоване на технології великих інтегральних схем ВІС функціонально закінчене облаштування обробки даних, здатне виконувати команди і програми. Він включає арифметико-логічний пристрій(АЛУ), внутрішню надоперативну пам'ять (НОЗУ) для зберігання команд і операндів, устрій управління виконанням команд (УУВК), шини зв'язку(внутрішні і зовнішні). На рисунку 1.1 представлена структурна схема мікропроцесора типу K580ИК80.

На цій схемі показані основні функціональні вузли МП :

АЛУ - арифметико-логічний пристрій, що є 8, - розрядний паралельний пристрій, що забезпечує виконання усіх основних операцій по зберіганню і обробці даних. За допомогою облаштування десяткової корекції АЛУ може обробляти двійкові дані (у двійковій системі числення);

АСС - 8 - розрядний акумулятор або накопичуючий регістр, де зберігається результат виконання команди;

АСТ - 8 - розрядний допоміжний акумулятор;

ТРМ- 8 - розрядний регістр тимчасового зберігання даних;

F - регістр ознак умов;

B, C, D, E, H, L - 8 - розрядні регістри загального призначення - призначені для зберігання даних, або для зберігання адреси зовнішньої пам'яті - в цьому випадку вони використовуються парами BC, DE, HL.

Дані передаються на внутрішню шину даних МП через мультиплексор(МПЛ), а 16 - розрядні адреси обмінюються між регістрами і БУФЕРОМ АДРЕСИ через адресну клямку і схему інкремента - декремента (РА).

Система команд МП цього типу містить команди довжиною 1, 2 або 3 байти (1 байт = 8 бітам). Другий і третій байти багатобайтових команд зазвичай розміщуються в допоміжних регістрах W і Z.

Перший байт команди завжди містить код операції, який розшифровується

ДЕШИФРАТОРОМ КОМАНД, який генерує відповідні мікропрограми формування машинних циклів.

Усі подальші дії виконуються відповідно до мікропрограми через СХЕМУ УПРАВЛІННЯ, яка видає не лише сигнали управління внутрішніми вузлами МП, але і зовнішні сигнали, що управляють взаємодією МП із зовнішньою пам'яттю і облаштуваннями введення, - виведення.

БУФЕР ДАНИХ - це двонаправлена схема з трьома логічними станами, призначеними для обміну інформацією МП із зовнішньою пам'яттю і облаштуваннями введення, - виведення. Він ізолює МП від зовнішньої магістралі даних, забезпечуючи прямий доступ до пам'яті з боку зовнішніх пристроїв.

На відміну від шини даних, 16 - розрядна шина адреси - однонапрямлена і завжди працює тільки в режимі видачі. БУФЕР АДРЕСИ також має три логічні стани, забезпечуючи відключення МП від шини адреси в режимі захоплення цієї шини іншими МП або зовнішніми пристроями.

РЕГІСТР ОЗНАК УМОВ - 8 - розрядний регістр, що служить для індикації стану результату операції і включає 5 розрядів умов :

- 3 - ознака перенесення із старшого розряду(1 - є, 0 - ні);
- 1 – значення розряду незмінне;
- Р - ознака парності результату(1 - чіт, 0 - непарне число);
- 0 – значення розряду незмінне;
- НС - ознака допоміжного ознака перенесення з третього розряду;
- 0– значення розряду незмінне;
- Z - ознака нульового результату(якщо результат виконання операції дорівнює нулю, то $Z=1$, якщо результат не дорівнює нулю те $Z= 0$);
- S - ознака знаку(якщо результат <0 , то $S=1$, якщо результат >0 те $S=0$).

ЛІЧИЛЬНИК КОМАНД (програмний лічильник) - вказує на адресу, де знаходиться в пам'яті черговий байт команди.

ПОКАЖЧИК СТЕКА - вказує на адресу, де зберігається початкова адреса підпрограми і адреса повернення до основної програми.

2.1 Команди МП K580

2.2 : Усі команди МП можна розділити на 5 груп:

- пересилки -выполняют операції обміну даними між регістрами або між регістрами і пам'яттю;
- арифметичних операцій -сложения, віднімання, інкрементирования і декрементирования даних в регістрах або елементах пам'яті;
- логічних операцій - I, АБО, що виключає АБО, порівняння, зсуву або доповнення даних в регістрах або елементах пам'яті;
- розгалуження- безумовний і умовний перехід в програмі
- виклик підпрограми і повернення з підпрограми;
- роботи із стеком, введенням-виведенням, управлінням.

Мова Асемблера є набором команд машинного коду, представленим в доступному для читання форматі.

Вивчаючи вміст байта пам'яті, не можна розрізнити, що знаходиться в цьому байті -код команди або дані, щоб їх розрізнити, потрібно усвідомити логіку програми.

Асемблерна програма складається з послідовності пропозицій. Кожна пропозиція належить одній з трьох груп :

- команд;
- псевдокоманд;
- наказів асемблера.

Кожна команда складається з чотирьох окремих і таких, що відрізняються один від одного полів, розташованих в наступному порядку :

- поле імені;
- поле коду;
- поле операндів;
- поле приміток

2.3 Способи адресації команд МП K580

2.3.1 Пряма адресація

У цьому способі адресації передбачається переміщення даних в акумулятор з елементу зовнішньої пам'яті, адреса якої слідує в другому і третьому байтах команди пересилки LDA *.

Зворотну операцію виконує команда пересилки STA *- переміщення даних з акумулятора в елемент зовнішньої пам'яті, адреса якої слідує в другому і третьому байтах команди.

2.3.2 Непряма адресація

При необхідності багатократного звернення до якої - або елементу пам'яті її адреса заноситься в регістрову пару загального призначення. Тоді в команді необхідно вказати лише ім'я першого регістра пари, де знаходиться адреса осередку.

Наприклад, по команді STAX B, слово даних з осередку, адреса якого зберігається в регістровій парі BC переноситься в акумулятор. Команда LDAX B виконує зворотну дію слово даних з акумулятора переноситься в осередок, адреса якого зберігається в регістровій парі BC.

2.2.3. Безпосередня адресація

У командах з безпосередньою адресацією слово даних або адреса слідує безпосередньо в другому або другому і третьому байтах команди услід за кодом операції в першому байті.

Наприклад, по команді MVI D 05 в регістр D поміщається двійкове слово даних 05, а по команді ADI 08 станеться складання утримуваного акумулятора з двійковим словом даних 08 і результат залишиться в акумуляторі.

2.2.4 Регістрова адресація

Якщо оброблювані дані розміщуються у внутрішніх регістрах МП, то звернутися до них можна вказавши код команди, що оперує з відповідними регістрами.

Наприклад, команда пересилки MOV CB здійснює пересилку слова даних з регістра У в регістр З, а команда ADD H виконує складання утримуваного акумуля-

тора з двійковим словом даних в регістрі Н і результат залишиться в акумуляторі.

2.2.5 Побічно - регістрова адресація

Якщо адресу зовнішнього елементу пам'яті розмістити в регістровій парі HL, то до слова даних, яке міститься в цьому осередку можна звернутися за допомогою однієї з команд побічно, - регістрової адресації. Ознакою цих команд є наявність в мнемокоді літери М.

Наприклад, команда ADD М проводить складання утримуваного акумулятора із словом даних : що знаходиться в елементі зовнішньої пам'яті, адреса якої зберігається в регістровій парі HL.

2.3 Команди логічних операцій

При виконанні всіх логічних команд слід задіяти розряди **Z, S, P** і **AC** регістру ознак (у розряд **C** записується **0**). Це використовується для перевірки стану будь- якого розряду числа і виконання умовних переходів у програмі, тобто залежно від отриманого в результаті виконання програми поточного значення переходити на різні ділянки програми (будувати алгоритми, що розгалужуються).

У багатьох випадках при виконанні програм необхідно перевіряти або змінювати (маскувати) стан одного або декількох розрядів числа в акумуляторі за допомогою команд логічного множення, додавання чи АБО, що виключає. Приклади використання цих команд наведені нижче в табл. 2.1, де акумулятор позначений **<А>**; коди команд **ANA R, ORA R** і **XRA R** визначаються як кодом операції, так і регістром загального призначення, що приймає в ній участь, (**B, C, D, E, H, L**).

Таблиця 2.1 - Приклади виконання логічних команд

Дія	Логічне множення	Логічне додавання	АБО, що виключає
Мнемокод	ANI <D8>, ANA M	ORI <D8>, ORA M	XRI <D8>, XRA M
Машинний код	E6 <D8>, A6	F6 <D8>, B6	EF <D8>, AE
Число в <А>	00111010	00111010	00111010
Маска	10101100	10101100	10101100

Результат	00101000	10111110	10010110
Число в <A>	00001111	00001111	00001111
Маска	11001100	11001100	11001100
Результат	00001100	11001111	11000011

Продовження таблиці 2.1

Число в <A>	00001111	00001111	00001111
Маска	00001111	00001111	00001111
Результат	00001111	00001111	00000000

При виконанні всіх логічних команд слід задіяти розряди **Z**, **S**, **P** і **AC** регістра ознак (у розряд **C** записується 0). Це використовується для перевірки стану будь-якого розряду числа і виконання умовних переходів у програмі, тобто в залежно від отриманого в результаті виконання програми поточного значення переходити на різноманітні ділянки програми (будувати алгоритми, що розгалужуються).

Команди зсуву, так само як і команди **ANA**, **ANI**, **ORA**, **ORI**, **XRA**, **XRI** працюють над кодом, тобто над кожним бітом числа, що знаходиться в акумуляторі, окремо. Довжина цих команд - один байт, мнемонічні позначення **RLC** (зсув уліво), **RRC** (зсув управо), **RAL** (зсув уліво через розряд ознак **C**) і **RAR** (зсув управо через розряд ознак **C**). При виконанні команди виконується зсув на один розряд. Зсув полягає в одночасному переміщенні вмісту всіх розрядів акумулятора в сусідні ліві при зсуві вліво або праві при зсуві вправо розряди. Зсув виконується циклічно: так, при зсуві вліво по команді **RLC** вміст старшого розряду акумулятора записується в його молодший розряд і одночасно потрапляє в розряд **C** регістра ознак. Наприклад, нехай до роботи команди **RLC** в акумуляторі знаходилося число **10101010** і в розряді **C** регістру ознак був **0**. Після виконання цієї команди в акумуляторі виявиться число **01010101**, а в розряді **C** буде **1**. Таке ж число виявиться в акумуляторі, якщо замість **RLC** виконати команду **RRC**, але в розряді **C** збережеться **0**. Команди **RAL** і **RAR** працюють аналогічно командам **RLC** і **RRC** із тою відмінністю, що розряд ознак **C**

розглядається як додатковий дев'ятий розряд.

2.4 Команди безумовних та умовних переходів

У будь-якій задачі керування необхідно вибрати один з альтернативних варіантів. Це виконують командами умовної і безумовної передачі керування. У процесорі K580ИК80 усі команди цієї групи, крім команди PCHL, мають пряму адресацію і відповідно трибайтовий формат. Вони використовують раніше вироблені ознаки, самі їх не змінюючи. Якщо задана в операції її кодом умова виконується, тобто розряд необхідної ознаки знаходиться в потрібному стані, то передача керування відбувається, у противному випадку команда пропускається і процесор виконує наступну команду програми. Процес передачі керування полягає в занесенні адреси, що утримується в команді (або на регістрах загального призначення H,L для команди PCHL) у лічильник команд. У випадку команд умовної передачі керування це відбувається лише при виконанні заданої умови.

Часто використовувану програму або її частину доцільно виконати у виді підпрограми - послідовності команд, виконання яких можна викликати з будь-якого місця програми довільне число разів. Передачу керування підпрограмі називають її викликом, необхідні для її роботи дані й адреси - вхідними параметрами, а результати, передані по закінченні її роботи в основну програму - вихідними параметрами.

Виклик і повернення з підпрограми можуть бути безумовними й умовними, що подано в табл. 2.2.

Для безумовного виклику використовують команду **CALL <A2><A1>;** при цьому в програмний лічильник завантажуються старший <A1> і молодший <A2> байти адреси підпрограми, а у стеку зберігається адреса основної програми, із якої вона буде виконуватися після завершення підпрограми по команді **RET**.

Таблиця 2.2. - Команди умовного переходу, виклику і повернення

Умова		Переклад	JUMP	CALL	RETURN
	No Zero	не дорівнює 0	JNZ {C2}	CNZ {C4}	RNZ {C0}
	Zero	дорівнює 0	JZ {CA}	CZ {CC}	RZ {C8}

Результат	Parity Even	парний	JPE {EA}	CPE {EC}	RPE {E8}
	Parity Odd	непарний	JPO {E2}	CPO {E4}	RPO {E0}
	Plus	позитивний	JP {F2}	CP {F4}	RP {F0}
	Minus	негативний	JM {FA}	CM {FC}	RM {F8}
Перенос	Carry	є	JC {DA}	CC {DC}	RC {D8}
	No Carry	немає	JNC {D2}	CNC {D4}	RNC {C0}

МІКРОПРОЦЕСОР ТА ЙОГО ОСНОВНІ КОМПОНЕНТИ

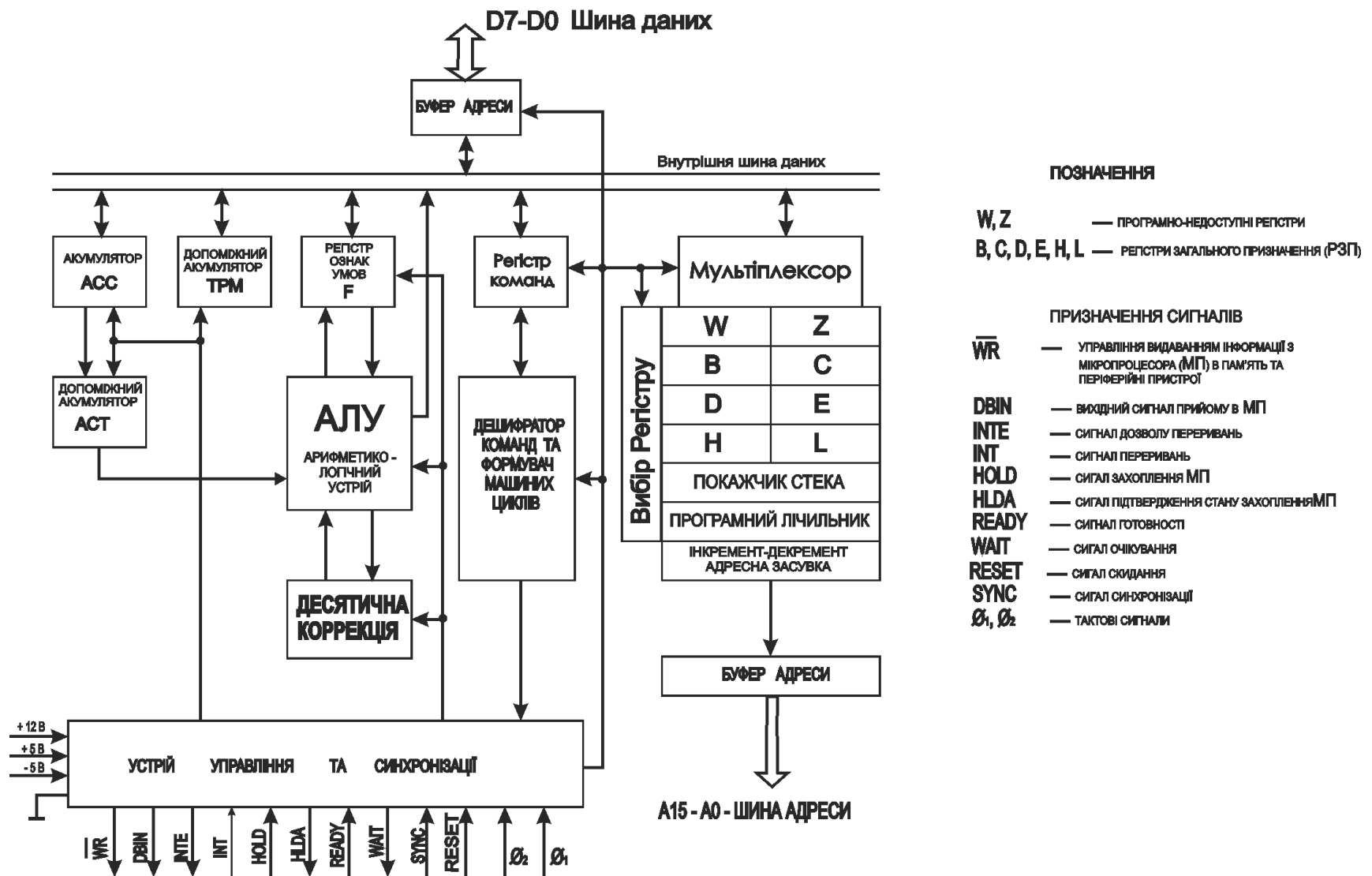


Рисунок 1.1 Структурна схема мікропроцесора типу K580ИК80.

3 ЛАБОРАТОРНА РОБОТА №1

ВИВЧЕННЯ КОМАНД МП K580 І МЕТОДІВ АДРЕСАЦІЇ; ЗАПИС І ВИКОНАННЯ ПРОГРАМ ІЗ ЛІНІЙНОЮ СТРУКТУРОЮ АЛГОРИТМУ

3.1. МЕТА РОБОТИ::

- вивчити систему команд мікропроцесора K580;
- ознайомитися з методами адресації та з записом програм;
- дослідити виконання окремих команд і простих програм.

У результаті виконання роботи студент повинен

знати:

- основні команди пересилки, арифметичних і логічних операцій;
- засоби адресації;

уміти:

- розробляти прості програми з лінійною структурою алгоритму;
- налагоджувати і виконувати прості програми
- набути навичок роботи з мікропроцесорною технікою.

3.2 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ №1

Завдання 1. Складіть програму додавання цілих чисел $D=A+B+C$, використовуючи команди для зазначених у таблиці 1 засобів адресації.

Завдання 2. Складіть програму обчислення цілих шістнадцятиричних чисел $D=A-B-C$.

Вивчіть виконання програми, записуючи зміст використовуваних регістрів на кожному кроці виконання програми.

3.3 ПОРЯДОК ВИКОНАННЯ РОБОТИ

3.3.1 Скласти програму.

3.3.2 Завантажити коди програми і вихідні дані в пам'ять емулятора роботи мікропроцесора K580

3.3.3 Виконати програму в покроковому режимі, записуючи після виконання кожної команди вміст регістрів, що беруть участь у виконанні даної програми і комі-

рок пам'яті.

3.3.4 Виконати програму в динамічному режимі, проаналізувати результат на предмет обсягу пам'яті, що займається програмою і часу її виконання.

3.4 ПРИКЛАД ВИКОНАННЯ ЗАВДАННЯ

Завдання. Скласти програму складання двох простих цілих чисел X і Y , які знаходяться в елементах зовнішньої пам'яті за адресами 0900 і 0901. Результат Z розмістити в осередку за адресою 0902.

Таблиця 3.1 - Приклад виконання завдання

Адреса	Мнемокод	Машинний код	Коментар
0800	LDA 0900	3A	Число X з осередку з адресою 0900 пересилається в акумулятор
0801		00	
0802		09	
0803 0804	MVI H 09	26 09	Перший напівбайт адреси числа Y пересилається у регістр H (09 в H)
0805 0806	MVI L 01	2E 01	Другий напівбайт адреси числа Y пересилається у регістр L (01 в L)
0807	ADD M	86	Складання X і Y , результат в акумуляторі
0808 0809 080A	LXI D 0902	11 02 09	У регістрову пару DE поміщається адреса елемента пам'яті, в який буде пересланий результат додавання
080B	STAX D	12	Результат додавання пересилається з акумулятора в осередок з адресою 0902
080C	HLT	76	Зупинка виконання програми

3.5 ЗМІСТ ЗВІТУ

3.5.1. Опис призначення, формату і засобів адресації команд пересилки й арифметичних операцій.

3.5.2. Схеми алгоритмів і тексти розроблених програм, оформлені відповідно до табл. 2.

3.5.3. Вміст регістрів і комірок пам'яті, отриманий при покроковому виконанні

програми.

3.5.4. Загальні висновки про пророблену роботу.

ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

Індивідуальні завдання наведені в табл. 2. Її перша колонка задає варіант за обліковим номером студента в журналі групи. Друга і третя колонки пропонують при розробці програми використовувати конкретні види адресації: П- пряма, Н- безпосередня, К- посередня, Р- регістрова, К-Р- посередньо-регістрова. Значення чисел А, В і С задає керівник; у команді з безпосередньою адресацією попередньо операнд записати як D8.

Таблиця 3.2- Індивідуальні завдання

№	Задача 1					Задача 2				
	П	Н	К	Р	КР	П	Н	К	Р	КР
1	X	X	X			X		X		
2		X	X	X		X		X	X	
3			X	X			X		X	
4				X	X		X	X	X	
5	X				X	X		X		
6	X			X			X			X
7	X		X	X				X		X
8	X	X			X	X			X	X
9	X	X					X		X	X
10	X	X		X				X	X	X
11	X		X			X	X	X		
12	X		X	X			X	X	X	
13		X		X				X	X	
14		X	X	X					X	X
15	X		X			X				X
16		X			X	X			X	
17			X		X	X		X	X	

18	X			X	X	X	X			X
----	---	--	--	---	---	---	---	--	--	---

Продовження таблиці 3.2

19		X		X	X	X	X			
20			X	X	X	X	X		X	
21			X	X			X		X	
22				X	X		X	X	X	
23	X				X	X		X		
24	X			X			X			X
25	X		X	X				X		X
26	X	X			X	X			X	X
27	X	X					X		X	X
28	X	X		X				X	X	X
29	X		X			X	X	X		

3.6 КОНТРОЛЬНІ ЗАПИТАННЯ

3.6.1. Як групують команди МП К580 за функціональним призначенням?

2.7.2. Який формат команд у МП К580 і як у кожному з них подані адреси, дані і код операції ?

2.7.3. Як організується пряма адресація? Наведіть приклади.

2.7.4. У чому складається посередньо-регістрова адресація? Наведіть приклади стосовних до неї команд.

2.7.5. У чому полягає регістрова адресація? Наведіть приклади команд та їх кодування.

2.7.6. Як зорганізується посереднє адресування? Наведіть приклади команд.

2.7.7. Якому виду адресування віддати перевагу?

2.7.8. У чому особливість стекового адресування? Наведіть приклади.

2.7.9. Як виконуються команди в алгоритмі лінійної структури?

4 ЛАБОРАТОРНА РОБОТА № 2

ВИВЧЕННЯ КОМАНД ЛОГІЧНИХ ОПЕРАЦІЙ І МАСКУВАННЯ ДАНИХ

4.1 МЕТА РОБОТИ:

- вивчити особливості виконання і застосування логічних команд;
- ознайомитися з програмними засобами маскування даних;

У результаті виконання роботи студент повинний

знати:

- порядок виконання різноманітних логічних команд і галузь їх застосування;
- засоби маскування даних;

уміти:

- складати алгоритми і програми з використанням логічних команд;
- виконувати операції по маскуванню даних;
- набути навички практичної роботи з упорядкування і налагодження програм із командами логічних операцій.

4.2 ПРИКЛАДИ ПРОГРАМ ІЗ МАСКУВАННЯМ ДАНИХ

4.2.1. Скласти алгоритм і програму маскування розрядів 0, 1, 2, 7 числа $X=AA$, поміщеного в регістр **B**, і розряди 3, 4, 5, 6 числа $Y = 55$, що знаходиться в регістрі **C**. З виділених розрядів чисел скласти число Z і помістити його в осередок пам'яті з адресою 2200.

Програма маскування чисел X і Y з одержанням числа Z наведена в табл.4.1

Таблиця 4.1 - Програма маскування чисел X і Y з одержанням числа Z

Адреса	Мнемокод	Машинний код	Коментар
0800 0801	MVI B, AA	06 AA	Помістити X у B
0802 0803	MVI C, 55	0E 55	Помістити Y у C
0804	MOV A, B	78	Передати B у A
0805 0806	ANI 87	E6 87	Маскування розрядів числа X , що не використовуються

0807	MOV B, A	47	Результат - у регістр B
0808	MOV A, C	78	Передати C у A

Продовження таблиці 4.1

0809 080A	ANI 78	E6 78	Маскування розрядів числа Y, що не використовуються
080B	ORA B	B0	Упорядкування числа Z
080C 080D 080E	LXI H, 2200	21 00 22	Завантажити в реєстрову пару HL адресу числа Z
080F	MOV M, A	77	Записати Z в осередок з адре- сою 2200
0810	HLT	76	Зупинка

4.3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ.

4.3.1. Вивчити виконання логічних команд И, АБО, АБО, що виключає НІ, Порівняння, Зсув.

4.3.2. Скласти програму маскування даних для завдань 1 і 2 із зазначеними у варіанті табл. 3 командами.

Завдання 1. Скласти алгоритм і програму маскування вказаних у варіанті завдання розрядів числа X і вказаних у варіанті завдання розрядів числа Y;. З виділе- них розрядів чисел X, Y, скласти число Z яке розмістити в елементі пам'яті.

Завдання 2. Скласти алгоритм і програму маскування вказаних у варіанті завдання розрядів числа X і і вказаних у варіанті завдання розрядів числа Y. Виділені розряди числа X перемістити в старші розряди регістра D. Виділені розряди числа Y перемістити в молодші розряди регістра D

4.4 ПОРЯДОК ВИКОНАННЯ РОБОТИ

4.4.1. Скласти програму.

4.4.2. Завантажити коди програми і вихідні дані в пам'ять емулятора роботи мікропроцесора K580 за адресами, поданими у табл. 3.

4.4.3. Запустити програму з трасуванням (по кроках). На кожному кроці перевіряти зміст внутрішніх регістрів мікропроцесора, результати записати в таблицю.

4.4.4. Запустити програму в автоматичному режимі, перевірити результати маскування даних.

4.5 ЗМІСТ ЗВІТУ.

4.5.1. Завдання, схеми алгоритмів і тексти розроблених програм.

4.5.2. Висновки про пророблену роботу.

4.6. ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

Таблиця 4.2 - Варіанти індивідуальних завдань

№ вар	Завдання 1										Завдання 2									
	Розряди, що виділяються								Числа		Розряди, що виділяються								Числа	
	0	1	2	3	4	5	6	7	X	Y	0	1	2	3	4	5	6	7	X	Y
1	X	Y	X	Y	X	Y	X	Y	BD	5F	Y	X	Y	X	Y	X	X	X	EE	87
2	Y	X	Y	X	Y	X	Y	Y	E4	2C	X	Y	X	Y	X	Y	X	X	D9	CB
3	Y	Y	X	Y	X	Y	X	Y	98	6A	Y	X	X	X	X	Y	Y	X	2B	69
4	Y	Y	Y	X	Y	X	Y	X	78	A9	X	Y	X	X	X	X	Y	X	58	49
5	Y	Y	Y	X	Y	X	Y	X	F1	C7	Y	X	Y	X	Y	X	X	Y	1D	3C
6	X	Y	X	Y	X	Y	X	X	2D	81	Y	X	Y	Y	Y	Y	X	Y	3C	D1
7	X	X	Y	X	Y	X	Y	X	3E	F4	X	Y	X	X	Y	X	X	Y	5F	72
8	X	X	X	Y	X	Y	X	Y	38	9A	Y	Y	Y	Y	X	Y	Y	X	2C	DE
9	X	Y	X	X	X	Y	X	Y	4C	E5	X	Y	Y	Y	X	Y	X	Y	6A	BC
10	Y	Y	Y	X	Y	X	Y	X	45	92	X	Y	X	X	X	X	Y	X	A9	98
11	X	Y	X	Y	X	Y	Y	Y	28	A8	Y	X	Y	Y	X	Y	X	X	C7	BF
12	Y	X	Y	X	Y	X	Y	Y	B9	DF	X	X	X	X	Y	X	X	Y	81	9A
13	Y	X	Y	X	Y	X	Y	X	71	39	Y	X	X	X	Y	X	Y	X	F4	41
14	X	X	X	Y	X	Y	X	Y	48	7D	Y	Y	X	Y	X	Y	X	Y	9A	EF

4.7 КОНТРОЛЬНІ ЗАПИТАННЯ

4.7.1. Охарактеризуйте призначення і порядок виконання таких команд логічних операцій: додавання, множення, АБО, що виключає.

4.7.2. Як перевірити (маскувати) стан одного або декількох розрядів двійкового числа?

5 ЛАБОРАТОРНА РОБОТА № 3

ВИВЧЕННЯ КОМАНД ЛОГІЧНИХ ОПЕРАЦІЙ, БЕЗУМОВНОЇ Й УМОВНОЇ ПЕРЕДАЧІ КЕРУВАННЯ

5.1 МЕТА РОБОТИ:

- вивчити особливості виконання і застосування логічних команд;
 - вивчити організацію безумовних і умовних переходів
- У результаті виконання роботи студент повинен

знати :

- порядок виконання різноманітних логічних команд і область їх застосування;
- засоби виконання умовних і безумовних переходів

уміти:

- складати алгоритми і програми з використанням логічних команд;
- складати програми з командами передачі керування;
- організовувати в програмі циклічні обчислення;
- набути навички практичної роботи з упорядкування і налагодження програм із командами логічних операцій, умовних і безумовних переходів.

5.2 ПРИКЛАДИ ПРОГРАМ ІЗ КОМАНДАМИ ПЕРЕДАЧІ КЕРУВАННЯ

5.2.1 Визначити значення нульового і сьомого розрядів числа, що знаходиться в осередку пам'яті 2220; якщо в цих розрядах знаходяться одиниці, то число проінвертувати і записати у той же осередок 2220, якщо ж хоч в одному із цих розрядів нуль, то число лишити без зміни. Програму наведено у табл. 5.1

Таблиця 5.1

Адреса	Мнемокод	Машинний код	Коментар
0800	LXI 0920	21	09 поміщається в Н, 20 поміщається в L
0801		20	
0802		09	
0803	MOV A, M	7E	X поміщається в A
0804	MOV B, A	47	X поміщається в B
0805	RLC	07	Зсув уліво

Продовження таблиці 5.1

0806	JNC 0811	D2	Якщо не 1 - ПУ
0807		11	
0808		08	
0809	RRC	0F	Подвійний зсув управо
080A	RRC	0F	
080B	JNC 0811	D2	Якщо не 1 - ПУ
080C		11	
080D		08	
080E	MOV A, B	78	X поміщається в A
080F	CMA	2F	Інвертування X
0810	MOV M, A	77	Результат поміщається в 0920
0811	HLT	76	Зупин

5.2.2. У пам'яті є два числа: в осередку 0830 - X, в комірці 0831 - Y. Визначити, яке з цих чисел більше алгебраїчно і помістити його в комірку 0820. Програму цієї задачі наведено у табл. 5.2.

Таблиця 5.2 - Програма на Асемблері

Адреса	Мнемокод	Машинний код	Коментар
--------	----------	--------------	----------

0800 0801 0802	LDA 0830	3A 30 08	X з осередку 0830 копіюється в А
0803	MOV B, A	47	X з А переміщується в В
0804 0805 0806	LDA 0831	3A 31 08	Y з осередку 0831 копіюється в А
0807	CMP B	B8	Порівняння X і Y
0808 0809 080A	JM 0811	FA 11 08	Якщо X більше Y, ПУ 0811

Продовження таблиці 5.2

080B 080C 080D	STA 0820	32 20 08	Y поміщається в 0820
080E 080F 0810	JMP 0815	C3 15 08	ПУ 0815
0811	MOV A, B	78	X поміщається в А
0812 0813 0814	STA 0820	32 20 08	X поміщається в 0820
0815	HLT	76	Зупин

5.3. ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

5.3.1 **Завдання 1.** Складіть програму порівняння п'ятих розрядів двох цілих чисел X і Y; якщо розряди однакові, записати 1 в осередок M, якщо немає, записати 0.

5.3.2 **Завдання 2.** Складіть програму, що записувала б в осередок M те з чисел, у якого молодша половина виявилася б більшою.

5.3.3. Виконуйте програму задачі 1 для випадків збігу і розбіжності розрядів.

5.3.4. Вивчіть виконання програми, записуючи зміст використовуваних регістрів на кожному кроці виконання програми.

5.4 ПОРЯДОК ВИКОНАННЯ РОБОТИ

5.4.1 Скласти програму.

5.4.2. Завантажити коди програми і вихідні дані в пам'ять емулятора K580.

5.4.3. Виконати програму в покроковому режимі, записуючи після виконання кожної команди вміст регістрів і осередків пам'яті, що беруть участь у виконанні даної програми.

5.4.4. Виконати програму в динамічному режимі, проаналізувати результат на предмет обсягу пам'яті, що займається програмою, і часу її виконання.

5.5 ЗМІСТ ЗВІТУ

5.5.1. Опис призначення, формату і засобів адресації команд пересилки й арифметичних операцій.

5.5.2. Схеми алгоритмів і тексти розроблених програм, оформлені відповідно до табл.5.2. та 5.3

5.5.3. Вміст регістрів і осередків пам'яті, отриманий при покроковому виконанні програми.

5.5.4. Загальні висновки про пророблену роботу.

5.6 ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

Індивідуальні завдання наведені в табл. 5.3. Її перша колонка задає варіант за обліковим номером студента в журналі групи. Друга і третя колонки пропонують при розробці програми використовувати конкретні види адресації: П- пряма, Н- безпосередня, К- посередня, Р- регістрова, С - стекова, К-Р- посередньо-регістрова. Четверта і п'ята колонки вказують зсув від початкової адреси. Для програм прийняти початкову адресу в контролері - 0800Н.; початкова адреса для вихідних даних і результатів - 0900Н.

Значення чисел А,В і С задає керівник; у команді з безпосередньою адресацією попередньо операнд записати як D8.

Таблиця 5.3 - Варіанти індивідуальних завдань

№	Задача 1						Задача 2						Зсув адрес	
	П	Н	К	Р	С	К-Р	П	Н	К	Р	З	К-Р	програми	вихідних даних і результатів
1	X	X	X				X		X		X		0F	2A
2		X	X	X			X		X			X	0A	1B
3			X	X	X		X			X		X	3C	0D
4				X	X	X		X		X	X		1E	31
5	X				X	X		X			X	X	21	45
6	X			X	X				X		X	X	17	2E
7	X	X			X				X	X		X	0B	19

Продовження таблиці 5.3

8	X	X		X				X	X		X		55	1C
9	X		X	X				X	X			X	15	3B
10	X	X				X		X		X		X	1F	0C
11	X		X		X		X	X	X				3D	0A
12	X		X			X		X	X	X			29	0B
13	X			X		X			X	X	X		0D	18
14		X		X	X					X	X	X	45	0E
15		X			X	X	X				X	X	2B	1E
16			X		X	X	X			X	X		0C	0F
17			X	X		X	X	X			X		33	2D
18		X	X		X		X	X		X			26	08
19		X	X			X	X		X	X			2F	33
20		X		X		X	X	X				X	30	4F

5.7. КОНТРОЛЬНІ ЗАПИТАННЯ

5.7.1. Як групують команди МП К580 за функціональним призначенням?

5.7.2. Який формат команд у МП К580 і як у кожному з них подані адреси, дані і код операції ?

5.7.3. Як організується пряма адресація? Наведіть приклади .

5.7.4. Опишіть адресний простір МП К580, контролера МС2702 і УМК VEF.

5.7.5. У чому полягає посередньо-реєстрова адресація? Наведіть приклади стосовних до неї команд.

5.7.6. У чому полягає реєстрова адресація? Наведіть приклади команд і їх кодування.

5.7.7. Як організується посередня адресація? Наведіть приклади команд.

6. ЛАБОРАТОРНА РОБОТА №4

ВИВЧЕННЯ ПРОГРАМНОГО КОМПЛЕКСУ «АЛЬТАЇР» У СКЛАДІ АСКОВЕ ОНПУ

6.1 МЕТА РОБОТИ:

- ознайомитися з основними складовими та режимами роботи ПК «АЛЬТАЇР»;
- ознайомитися з способами відображення інформації, накопиченої у лічильнику;
- дослідити процес пересилання інформації, накопленої у електrolічильнику у персональний комп'ютер .

У результаті виконання роботи студент повинен

знати:

- схеми включення електронних лічильників «Енергія» типу СТКЗ.
- основні режими роботи ПК «АЛЬТАЇР» ;
- засоби відображення інформації; ПК «АЛЬТАЇР»;

уміти:

- користуватись ПК «Альтаір» у режимі диспетчера.
- одержувати інформацію з лічильників у режимі адміністратора
- набути навичок роботи з мікропроцесорною технікою.

6.2 СТИСЛІ ТЕОРЕТИЧНІ ВІДОМОСТІ

З основними характеристиками ПК «Альтаір» та лічильників «Енергія» СТКЗ ознайомитись за допомогою «Керівництва користувача»

6.3 ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

6.3.1 За допомогою ПК «Альтаір» зняти графіки навантаження за період вказаний керівником у вигляді Excell файла, за даними якого зробити наступні розрахунки.

6.3.2 Розрахувати добове та місячне споживання електроенергії.

6.3.3 Розрахувати добове та місячне споживання реактивної енергії

6.3.4 Розрахувати та вибрати потужність необхідної конденсаторної установки

6.3.5 Розрахувати добове та місячне споживання реактивної енергії з урахуванням роботи обраної конденсаторної установки.

6.3.6 Розрахувати величину 30 хвилинного середнього навантаження ймовірність перевищення якого становить 5%. (Викорастати надбудови Excell).

6.3.7 Визначити розрахункове значення активної потужності (максимальне з середніх 30- хвилинних навантажень)

6.3.8 Розрахувати середньозважений коефіцієнт потужності до та після компенсації реактивної потужності

6.4 ЗМІСТ ЗВІТУ

6.4.1 Графікі навантаження активної та реактивної потужності.

6.4.2 Розрахунки та висновки.

6.4.3 Схеми включення лічильників «Енергія» СТКЗ.

6.5. ВАРІАНТИ ІНДИВІДУАЛЬНИХ ЗАВДАНЬ

Надаються керівником.

Додаток А Таблица кодів команд асемлера мікропроцесора К 580

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	LXI B,&	STAX B	INX B	INR B	DCR B	MVI B,#	RLC	-	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C,#	RRC
1	-	LXI D,&	STAX D	INX D	INR D	DCR D	MVI D,#	RAL	-	DAD D	LDA X D	DCX D	INR E	DCR E	MVI E,#	RAR
2	-	LXI H,&	SHLD *	INX H	INR H	DCR H	MVI H,#	DAA	-	DAD H	LHLD *	DCX H	INR L	DCR L	MVI L,#	CMA
3	-	LXI SP,&	STA *	INX SP	INR M	DCR M	MVI M,#	STC	-	DAD M	LDA *	DCX SP	INR A	DCR A	MVI A,#	CMC
4	MOV B,B	MOV B,C	MOV B,D	MOV B,E	MOV B,H	MOV B,L	MOV B,M	MOV B,A	MOV C,B	MOV C,C	MOV C,D	MOV C,E	MOV C,H	MOV C,L	MO V C,M	MO V C,A
5	MOV D,B	MOV D,C	MOV D,D	MOV D,E	MOV D,H	MOV D,L	MOV D,M	MOV D,A	MOV E,B	MOV E,C	MOV E,D	MOV E,E	MOV E,H	MOV E,L	MO V E,M	MO V E,A
6	MOV H,B	MOV H,C	MOV H,D	MOV H,E	MOV H,H	MOV H,L	MOV H,M	MOV H,A	MOV L,B	MOV L,C	MOV L,D	MOV L,E	MOV L,H	MOV L,L	MO V L,M	MO V L,A
7	MOV M,B	MOV M,C	MOV M,D	MOV M,E	MOV M,H	MOV M,L	HLT	MOV M,A	MOV A,B	MOV A,C	MOV A,D	MOV A,E	MOV A,H	MOV A,L	MO V A,M	MO V A,A
8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
9	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
A	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
D	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMP E	CMP H	CMP L	CMP M	CMP A
C	RNZ	POP B	JNZ *	JMP *	CNZ *	PUSH B	ADI #	RST 0	RZ	RET	JZ *	-	CZ *	CALL *	ACI #	RST 1
D	RNC	POP D	JNC *	OUT №	CNC *	PUSH D	SUI #	RST 2	RC	-	JC *	IN №	CC *		SBI #	RST 3
E	RPO	POP H	JPO *	XTHL	CPO *	PUSH H	ANI #	RST 4	RPE	PCHL	JPE *	XCHG	CPE *		XRI #	RST 5
F	RP	POP PSW	JP *	DI	CP *	PUSH PSW	ORI #	RST 6	RM	SPHL	JM *	EI	CM *		SPI #	RST 7

№ - номер порту вводу-виводу
 & - двохбайтовий операнд - D16
 * - двлхбайтовий операнд - ADR
 # - однобайтовий операнд – D*

Приклад: Команда STAX D має код операції 12;
 Код операції CA відповідає команді JZ ADR

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Ліхтциндер Б.Я., Кузнецов В.Н. Мікропроцесорні та обчислювальні пристрої у радіотехніці: Учб. посібник. - К.:Вища шк. Головне вид-во, 1991.-272с.
2. Танненбаум Э. Современные операционные системы. - 2е изд.-СПБ.: Питер, 2002 - 1040 с.: ил.
3. Организация ЭВМ. 5-е изд./ К. Хамахер, З. Варншич, С. Заки. - СПб.: Питер; Киев: Издательская группа BHV, 2003.-848 с.:ил.
4. Таннебаум Э. Архитектура компьютера. - СПб.: Питер, 2002 - 704 с.:ил.
5. Столингс, Вильям. Операционные системы, 4-е издание.: Пер.с англ - М.: Издательский дом "Вильямс", 2002 - 848с.:л - Парал. тит. англ.