

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Запорізький національний технічний університет

МЕТОДИЧНІ ВКАЗІВКИ

до виконання лабораторних робіт студентами всіх форм навчання при вивченні дисципліни «Основи мікропроцесорної техніки» для підготовки бакалаврів за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка» з подальшим навчанням за освітніми програмами «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв».
Частина 1

2019

Методичні вказівки до виконання лабораторних робіт студентами всіх форм навчання при вивченні дисципліни «Основи мікропроцесорної техніки» для підготовки бакалаврів за спеціальністю 141 «Електроенергетика, електротехніка та електромеханіка» з подальшим навчанням за освітніми програмами «Електричні та електронні апарати» та «Електромеханічне обладнання енергоємних виробництв». Частина 1 / Укл.: Л. Б. Жорняк, М. В. Антонова. - Запоріжжя: ЗНТУ, 2019. – 72 с.

Укладачі: Л.Б. Жорняк, доцент, к.т.н.
М.В. Антонова, ст. викладач

Рецензент: О.В. Близняков, доцент, к.т.н.

Відповідальний за випуск: П.Д. Андрієнко, професор, д.т.н.

Затверджено
на засіданні НМК ЕТФ
протокол № 6
від 24. 01. 2019

Затверджено
на засіданні кафедри
"ЕЕА", протокол №8
від 26. 12. 2018

ЗМІСТ

ВИМОГИ ЩОДО ВИКОНАННЯ, ОФОРМЛЕННЯ ТА ЗАХИСТУ ЗВІТІВ ПРО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ	5
1 ЛАБОРАТОРНА РОБОТА №1 НАВЧАЛЬНИЙ МІКРОПРОЦЕСОРНИЙ КОМПЛЕКТ.....	6
1.1 Мета роботи	6
1.2 Загальні відомості.....	6
1.3 Самостійне виконання роботи	15
1.4 Зміст звіту	16
1.5 Контрольні запитання	16
2 ЛАБОРАТОРНА РОБОТА № 2 КОМАНДИ СИСТЕМНОГО МОНИТОРА НМК.....	16
2.1 Мета роботи	16
2.2 Робота з емулятором К580.....	16
2.3 Зміст звіту.....	23
2.4 Контрольні запитання.....	23
3 ЛАБОРАТОРНА РОБОТА № 3 АРХІТЕКТУРА МІКРОПРОЦЕСОРА КР580ВМ80А. КОМАНДИ ЗАВАНТАЖЕННЯ ТА ПЕРЕСИЛАННЯ.....	23
3.1 Мета роботи.....	23
3.2 Загальні відомості.....	23
3.3 Самостійне виконання роботи.....	35
3.4 Зміст звіту.....	36
3.5 Контрольні запитання.....	36
4 ЛАБОРАТОРНА РОБОТА № 4 ВИВЧЕННЯ ПОКРОКОВИХ РЕЖИМІВ РОБОТИ НМК.....	36
4.1 Мета роботи.....	36
4.2 Загальні відомості.....	36
4.3 Самостійне виконання роботи.....	39
4.4 Зміст звіту.....	43
4.5 Контрольні запитання.....	43
5 ЛАБОРАТОРНА РОБОТА № 5 АДРЕСАЦІЯ ПАМ'ЯТІ. МЕТОДИ І КОМАНДИ	

РОБОТИ З ПАМ'ЯТТЮ.....	46
5.1 Мета роботи.....	46
5.2 Загальні відомості.....	46
5.3 Самостійне виконання роботи.....	56
5.4 Зміст звіту.....	58
5.5 Контрольні запитання.....	58
6 ДВІЙКОВА АРИФМЕТИКА МІКРОПРОЦЕСОРА. ОПЕРАЦІЇ І КОМАНДИ.....	59
6.1 Мета роботи.....	59
6.2 Загальні відомості.....	59
6.3 Самостійне виконання роботи.....	69
6.4 Зміст звіту.....	70
6.5 Контрольні запитання.....	70
ПЕРЕЛІК ПОСИЛАНЬ.....	71

ВИМОГИ ЩОДО ВИКОНАННЯ, ОФОРМЛЕННЯ ТА ЗАХИСТУ ЗВІТІВ ПРО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Виконання лабораторної роботи складається з роботи студента у лабораторії, а також із самостійної роботи. Активна самостійна робота є визначальним чинником ефективного сприйняття змісту лабораторних занять.

Самостійна робота студента передбачає:

- використання на практиці теорії для вирішення дискретних задач в електроніці;

- підготовку відповідей на контрольні запитання, що наведені в кінці кожної лабораторної роботи;

- оформлення звітів про виконання лабораторних робіт.

Теоретична підготовка до чергової роботи передбачає:

- опрацювання теоретичних розділів, що визначають зміст та мету лабораторної роботи, та методичних вказівок до лабораторних робіт з використанням рекомендованих підручників та посібників;

- програмування МП у шістнадцятирічних кодах і складати прості програми на мові Асемблер.

Робота студента у лабораторії передбачає такі етапи:

- отримання допуску до лабораторної роботи;

- написання програм за допомогою системи команд МП КР580, тобто мовою Асемблера;

- формулювання висновків, розробка та налагодження розробленої програми, доводка її до рішення та отримання чисельних результатів;

- захист звітів про виконання лабораторних робіт.

Примітки:

1) Виконання лабораторних робіт у лабораторії здійснюється за розкладом, затвердженим проректором з навчальної роботи.

2) Відпрацьовування пропущених робіт у лабораторії здійснюється перед модульним тижнем за окремим розкладом, затвердженим завідувачем кафедри.

1 ЛАБОРАТОРНА РОБОТА №1

НАВЧАЛЬНИЙ МІКРОПРОЦЕСОРНИЙ КОМПЛЕКТ

1.1 Мета роботи

Вивчення будови навчального мікропроцесорного комплекту НМК-1 і органів його керування, введення, індикації.

1.2 Загальні відомості

1.2.1 Призначення і характеристики НМК

НМК являє собою закінчену мікроЕОМ і призначений для підготовки фахівців в галузі мікропроцесорної техніки шляхом вивчення структур й основ програмування мікропроцесора КР580ВМ80А, НМК може бути використаний у якості керуючої ЕОМ при створенні і дослідженні роботи систем керування електричними апаратами і процесами. Він є легко освоюваним і зручним засобом для налагодження невеликих (до 2 Кбайт) програм користувача. Засоби індикації на чільній панелі дозволяють спостерігати процеси перетворення і передачі інформації під час роботи НМК. Технічні характеристики НМК-1 представлені в таблиці 1.1

Таблиця 1.1–Технічні характеристики НМК-1

Характеристика	Параметр
Тип застосовуваного мікропроцесора	КР580ВМ80А
Об'єм оперативного запам'ятовуючого пристрою	2Кбайта
Об'єм постійного запам'ятовуючого пристрою	2Кбайта
Можливість переривання	1 вектор
Програмне забезпечення	Системна програма "Монітор"
Напруга живлення	220 В \pm 22 В частотою 50Гц \pm 1 Гц
Рівні вхідних і вихідних сигналів сумісні з рівнями ТТЛ ІС	

1.2.2 Будова НМК

НМК складається з таких складових частин (рисунок.1.1):

- мікроЕОМ;
- пульт оператора;
- блок живлення.

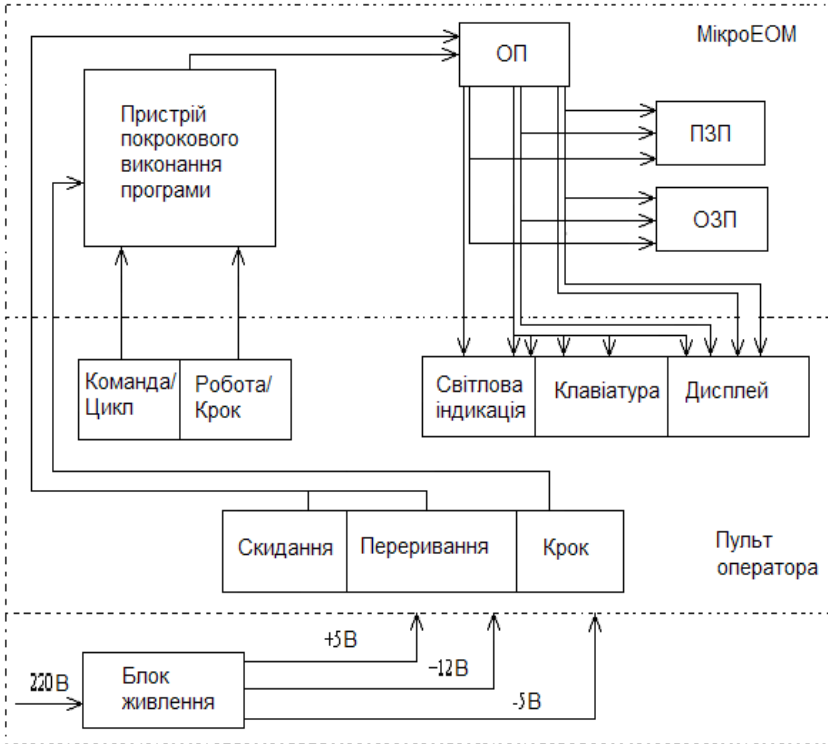


Рисунок 1.1 – Структура НМК

МікроЕОМ є основною частиною і керує роботою усього НМК, здійснює звертання до пам'яті, уведення, виведення і індикацію інформації. Основою мікроЕОМ є операційний пристрій ОП, що складається з мікропроцесора (МП) і генератора тактової частоти (ГТЧ).

Мікропроцесор КР580ВМ80А призначений для виконання визначеного набору команд і реалізований на одній ВІС, що включає акумулятор і кілька регістрів загального призначення (РЗП).

Регістром будемо називати мінімальний адресуючий запам'ятовуючий елемент, що складається з 8 розрядів (бітів). Інформація, яка зберігається в такому регістрі, називається словом. Регістри оперативного запам'ятовуючого пристрою (ОЗП) звичайно називають комірками, кожному з яких приписаний деякий номер-адреса. Ці регістри використовуються для збереження даних і команд користувача і виконані у виді окремих ВІС. Мікропроцесор може звертатися до них за допомогою спеціальних команд. Акумулятор (А) - найважливіший регістр МП. Більшість команд використовує акумулятор для збереження вихідних даних і результатів операції. Робота НМК полягає в обміні даними між регістрами й описується системою команд, кожна з яких визначає конкретний вид обміну і перетворення даних. Тому для розробки програми мовою машинних команд необхідно знати склад і призначення регістрів НМК і систему команд МП КР580.

Розглянемо структуру доступних користувачу регістрів НМК на базі МП КР580ВМ80А (рис. 1.2). Адресація всіх регістрів усередині НМК здійснюється в двійковій системі числення, у той час як користувачу зручно для скорочення запису користатися шістнадцятковою системою. У подальшому для позначення адрес і вмісту регістрів будемо використовувати шістнадцяткову систему. Розмір слова МП КР580 складає 8 біт (1 байт), тому розрядність усіх наведених регістрів, крім покажчика стеку SP (Stack Pointer) і лічильника команд PC (Program Counter) також складає 1 байт. МП КР580 має один акумулятор А і шість РЗП: В,С,D,E,H,L. РЗП можуть поєднуватися в двобайтові регістри ВС,DE,HL. У цьому випадку в регістрах В,D,H знаходиться старший, а в регістрах С,E,L - молодший байт шістнадцятирозрядного слова. МП містить також регістр прапорів умов F, у якому використовуються 5 розрядів з 8.

Виконання програм вимагає, як правило, звертання до основної пам'яті і буферних регістрів пристрою вводу-виводу (ПВВ). Призначенням буферних регістрів є узгодження характеристик МП і зовнішніх пристроїв користувача (принтер, дисплей, датчики, вимірювальні прилади та ін.).

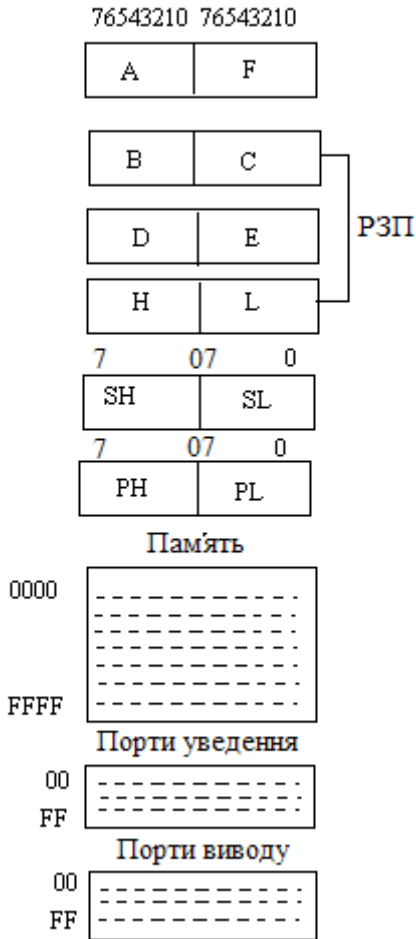


Рисунок 1.2 – Структура регістрів мікропроцесора

Звертання до усіх цих регістрів викликано тим, що велика кількість даних не вміщується в РЗП і програма повинна мати методи зв'язку з "зовнішнім світом" для одержання й видачі інформації. Тому необхідно уявляти собі доступний об'єм пам'яті й регістрів ПУВ.

Адресний простір (поле) пам'яті - це сукупність комірок пам'яті, до яких може адресуватися МП. Буферний регістр адреси МП КР580 - шістнадцятирозрядний, тому поле пам'яті містить $2^{16} = 65536$ комірок (64 Кбайт, тому що 1 Кбайт = 1024 байт) ємністю 1 байт кожний з відповідною адресою від 0000 до FFFF. Реальна мікроЕОМ може використовувати не усе адресне поле, а тільки його частину. Сукупність комірок пам'яті, що фактично існують в даній мікроЕОМ, утворюють робочий (фізичний) простір (поле) пам'яті. В НМК робочий простір пам'яті складає 4 Кбайти, з яких 2 Кбайти займає ПЗП і 2 Кбайти - ОЗП. З ПЗП інформація може тільки зчитуватися, а ОЗП дозволяє як зчитувати зміст його комірок, так і записувати в них нові дані.

Адресний простір ПБВ дозволяє мікроЕОМ мати до $2^8 = 256$ пристроїв (портів) уведення і виводу, адресованих роздільно. Адреси портів можуть знаходитися в межах 00 - FF.

В НМК 1 Кбайт ПЗП (адреси 0000 - 03FF) займає програма "Монітор" і 1 Кбайт ПЗП (адреси 0400 - 07FF) наданий у розпорядження користувача. ОЗП використовується для збереження програм, що змінюються, і даних користувача.

Останні 54 елементи ОЗП (адреси FFCA - FFFF) зайняті програмою "Монітор" для збереження оперативних даних і не повинні використовуватися оператором. Адреса 0800 є початковою, з якої починаються всі програми користувача.

Операційний пристрій проводить всі операції по переробці інформації. Його вихідним станом є читання інформації з нульової адреси ПЗП кожен раз після натискання керуючої кнопки <СБРОС> на пульті. При цьому викликається програма "Монітор", що забезпечує уведення інформації з клавіатури і виведення її на індикатор. Інформація про стан ОП фіксується в регістрі стану на початку кожного машинного циклу. У табл. 1.2 приведені можливі стани ОП.

Стану 0 у таблиці відповідає низький рівень потенціалу, а стану 1 - високий. У залежності від стану цього регістра формуються сигнали, що керують роботою усієї мікроЕОМ.

У табл. 1.3 надане визначення кожного біта регістра стану. Риска над керуючим сигналом WO у табл.1.2 і 1.3 вказує, що активним станом сигналу є логічний 0.

Пристрій покрокового виконання програми переводить ОП в стан "чекання" після виконання чергового кроку.

Пульт оператора призначений для взаємодії оператора з мікро-ЕОМ. Органи керування, уведення й індикації НМК розташовані на чільній панелі й описані нижче. Блок живлення забезпечує постійними стабілізованими напругами мікроЕОМ і пульт оператора.

Таблиця 1.2 – Можливі стани ОП

Стан ОП	Розряди регістра стану ОП							
	D0 INTA	D1 WO	D2 STACK	D3 HALT	D4 OUT	D5 M1	D6 INP	D7 MEMR
Вибір команди	0	1	0	0	0	1	0	1
Читання пам'яті	0	1	0	0	0	0	0	1
Запис до пам'яті	0	0	0	0	0	0	0	0
Читання стеку	0	1	1	0	0	0	0	1
Запис до стеку	0	0	1	0	0	0	0	0
Ввод	0	1	0	0	0	0	1	0
Вивод	0	0	0	0	1	0	0	0
Переривання	1	1	0	0	0	1	0	0
Зупин	1	0	0	0	1	0	1	0
Переривання при зупину	0	0	1	0	1	0	1	1

Таблиця 1.3 – Біти регістру стану ОП

Значення сигналу	Розряд регістра стану	Пояснення
INTA	D0	Сигнал підтвердження запиту переривання. Використовується для уведення на шину даних команди RST.
WO	D1	Вказує на те, що в даному циклі виконується запис до пам'яті або операція виводу.
STACK	D2	Означає наявність на шині адреси вмісту покажчика стеку.
HLTA	D3	Сигнал підтвердження команди HLT
OUT	D4	Вказує на те, що в даному циклі виконується операція виводу.
M1	D5	Вказує на те, що даний цикл використовується для вибірки першого байту команди.
INP	D6	Вказує на те, що в даному циклі виконується команда уведення.
MEMR	D7	Вказує на те, що в даному циклі буде проведене читання пам'яті.

1.2.3 Органи керування, уведення й індикації НМК.

На передній панелі НМК розташовано:

- кнопка вмикання/вимикання;
- кнопки скидання та переривання;
- кнопки керування покроковим режимом роботи;
- функціональна клавіатура;
- клавіатура вводу даних;
- шестисегментний дисплей;
- світлодіоди індикації стану шини даних, шини адреси, керуючих сигналів МП;
- роз'єм для приєднання макетного ТЕЗ.

Кнопка вмикання/вимикання НМК (символ "~") розташована в лівій нижній частині чільної панелі. Натиснута кнопка відповідає увімкненому стану НМК, віджата - вимкненому. Над кнопкою розміщені три світлодіоди: + 5 В; - 5 В; + 12 В. При перевантаженнях спрацьовує захист блоку живлення і загоряється відповідний світлодіод. У цьому випадку необхідно вимкнути НМК. Кнопка скидання (<СБ>) служить для ініціалізації системної програми "Монітор". Після натискання кнопки <СБ> здійснюється запуск цієї програми й у лівій позиції дисплея з'являється символ " - " це означає, що НМК готовий до прийому команд.

Кнопка переривання (<ПР>) розташована під кнопкою <СБ>. При натисканні цієї кнопки виробляється сигнал **ЗАПИТ НА ПЕРЕРИВАННЯ** сьомого рівня і, якщо переривання дозволені (виконана команда EI - Enabled Interrupt),буде закінчене виконання поточної команди, після чого керування передається на адресу 38H (відповідає сьомому рівню переривань). Починаючи з цієї адреси розміщується програма обробки переривань. Якщо кнопка <ПР> натиснута під час роботи програми "Монітор", то на дисплеї виводиться символ " ? ". В іншому випадку на дисплей виводиться адреса точки переривання й керування передається програмі "Монітор".

Керування покроковим режимом роботи НМК виробляється за допомогою кнопок РАБОТА / ШАГ (<РБ/ШГ>), КОМАНДА / ЦИКЛ (<КМ / ЦК>), ШАГ (<ШГ>). Цими кнопками встановлюється один із двох режимів покрокової роботи. Перший режим - командний. Для встановлення цього режиму слід натиснути кнопку <РБ / ШГ>. Кожне натискання кнопки <ШГ> викликає виконання поточної команди. При цьому на світлодіодах індикації стану шин даних, адреси й керуючих сигналів будуть висвітлюватися в двійковому коді, відповідно, адреса і код виконуваної команди, а також керуючі сигнали МП. Другий режим покрокової роботи - робота за командними циклами. Для встановлення цього режиму слід натиснути кнопки <РБ/ШГ> і <КМ/ЦК>. У цьому випадку можна простежити хід виконання команди. При кожному натисканні кнопки <ШГ> буде виконаний наступний машинний цикл. На світлодіодах індикації буде відбиватися інформація, що відповідає кожному машинному циклу. Для роботи в автоматичному режимі обидві клавіші <РБ/ШГ> і <КМ/ЦК> повинні бути віджаті.

Клавіатура НМК розділена на дві частини. У лівій частині розташовані функціональні клавіші. За кожною клавішею закріплена визначена функція програми "Монітор".

Призначення функціональних клавіш:

- <П> – перегляд і модифікація вмісту комірок пам'яті;
- <РГ> – перегляд і модифікація вмісту регістрів МП;
- <СТ> – старт програми;
- <КС> – підрахунок контрольної суми;
- <ЗК> – заповнення масиву пам'яті константою;
- <ПМ> – переміщення масиву пам'яті;
- <-> – роздільник;
- <ВП> – виконати.

Права частина клавіш призначена для уведення параметрів у шістнадцятковій системі. На ці клавіші нанесені символи: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. На клавіші 4,5,6,7,8,9 нанесені позначення ідентифікаторів регістрів МП:

- <PH> – старший байт лічильника команд;
- <PL> – молодший байт лічильника команд;
- <SH> – старший байт покажчика стеку;
- <SL> – молодший байт покажчика стеку.

Для ідентифікації інших регістрів МП використовуються клавіші:

- <A> – регістр А;
- – регістр В;
- <C> – регістр С;
- <D> – регістр D;
- <E> – регістр Е;
- <F> – регістр F.

Шестисегментний дисплей призначений для відображення даних у шістнадцятковій системі. Ліві чотири сегменти використовуються для відображення адреси й ідентифікаторів регістрів МП, а праві два для відображення даних. Індикація стану шин адреси, даних і керуючих сигналів МП здійснюється за допомогою світлодіодів. Шина адреси ідентифікується шістнадцятьма світлодіодами, шина даних - за допомогою восьми світлодіодів (відображення в двійковій системі). Ці світлодіоди позначені, відповідно, <АДРЕСА> і <ДАНИ>. Під кожним світлодіодом зазначені номери розрядів, яким вони відпо-

відають. Світлодіоди керуючих сигналів МП позначені написом <СО-СТОЯНИЕ> і кожен розряд має своє призначення.

1.2.4 Підготовка НМК до роботи.

1.2.4.1 Установіть кнопку вмикання НМК у віджатий стан.

1.2.4.2 Підключить НМК до мережі змінного струму напругою 220 В, частотою 50 Гц.

1.2.4.3 Кнопки <РБ / ШГ> і <КМ / ЦК> установіть у натиснутий стан.

1.2.4.4 Включить НМК кнопкою вмикання.

При цьому не повинні загорятися світлодіоди захисту + 5В, - 5 В, + 12 В.

1.2.4.5 Натисніть кнопку <СБ> і в крайньому лівому розряді дисплея повинний з'явитися знак " - ".

Після цього НМК готовий до прийому команд.

Повторне вмикання НМК слід робити не раніше, ніж через 20 с після вимикання. У протилежному випадку спрацює захист блоку живлення і загоряться відповідно світлодіоди захисту. У цьому випадку необхідно вимкнути НМК і дочекатися, коли світлодіоди захисту згаснуть, після чого повторно вимкнути НМК.

1.3 Самостійне виконання роботи

1.3.1 Визначити місце основних органів керування на НМК, їхнє призначення.

1.3.2 З'ясувати місце і призначення індикаторів, а також засобів приєднання зовнішніх пристроїв.

1.3.3 Вивчити призначення основних вузлів НМК, та їх характеристики.

1.3.4 Освоїти практично порядок вмикання та вимикання НМК з обов'язковою витримкою часу до 20 секунд (момент згасання всіх трьох світлодіодів зліва на чільній панелі НМК) перед наступним вмиканням.

1.3.5 Освоїти порядок виклику вмісту комірок пам'яті, регістрів, перегляду вмісту пам'яті, а також регістрів.

1.3.6 Освоїти правильні прийоми роботи з клавіатурою. Натискання окремих клавіш робити вертикально, з мінімальним зусиллям кисті (не руки), як це прийнято на електричних і електронних друкарських машинках.

1.3.7 З'ясувати можливості встановлення покрокових режимів у виконанні програм, роль лінійних індикаторів.

4 Зміст звіту

Звіт повинен містити:

- назву та мету роботи;
- стислі відповіді на контрольні запитання;
- висновки.

1.5 Контрольні запитання

1.5.1 Основні характеристики НМК, його функціональні можливості.

1.5.2 Структура НМК, основні вузли, їхні характеристики.

1.5.3 Характеристика й організація пам'яті.

1.5.4 Регістрова структура мікропроцесора.

1.5.5 Характеристика можливих станів операційного пристрою.

1.5.6 Характеристика органів керування та індикації НМК.

1.5.7 Основні вимоги до роботи з НМК.

2 ЛАБОРАТОРНА РОБОТА № 2 КОМАНДИ СИСТЕМНОГО МОНІТОРА НМК

2.1 Мета роботи

Вивчення можливостей емулятора системної програми "Монітор" і одержання навичок роботи з основними командами МП на базі К580.

2.2 Робота з емулятором К580

Для виконання запропонованих програм можна використовувати емулятор К580, що знаходиться в ауд. 226 а у системного адміністратора на сервері, або в мережі Internet.

Це є програмна модель (емулятор) лабораторного стенда навчального мікропроцесорного комплексу НМПК-80 (мікроЕОМ із базо-

вим мікропроцесором типу K580, U880, Intel 8080, 8085, Z80 і вбудованим дисплеєм. У комплекті НМК програма вбудована в постійний запам'ятовуючий пристрій (ПЗП), котру можна використовувати в навчальному процесі для вивчення різноманітних мікро-ЕОМ разом з НМК, що дозволяє студентам підготувати лабораторні роботи вдома за допомогою обчислювальних пристроїв.

Ця модель була розроблена для скорочення студентами часу на рутинні операції (асемблювання, дізасемблювання і т.і.). Вона дає більш широкі та зручні можливості для набору та відпрацювання програм (наприклад, може бути можливий одночасна перевірка всіх регістрів, комірок пам'яті, ввід команд в мнемонічних позначках (операторах) та в шістнадцятирічних кодах, асемблювання та дізасемблювання команд і т.і.). Для вивчення емулятора необхідно спробувати роботу на прикладі. Тож виконайте таку послідовність для програмування в емуляторі.

2.2.1 Запишіть до пам'яті програму згідно з таблицею 2.1, що реалізує рівняння : $HL = (B \text{ OR } C) \text{ AND } DE \text{ AND } F0FF \text{ H}$.

Таблиця 2.1

Адреса	Команда	Код	Коментар
800	MOV A, C	79	Запис $A < - C$ (вмісту)
801	ORA B	B0	Порівняння $A = A \text{ OR } B$
802	ANA E	A3	Порівняння $A = \text{AND } E$
803	ANI FFH	E6 FF	Порівняння $A = A \text{ AND } FFH$
805	MOV L, A	6F	Пересилання $L < - A$, молодшого байту результату
806	MOV A, D	7A	Пересилання $A < - D$
807	ANI F0H	E6 F0	$A = A \text{ AND } F0H$
809	MOV H, A	67	Пересилання $H < - A$, старшого байту результату

Спочатку напишіть програму, для чого свій зріст у шістнадцяти-річному коді візьміть як вихідні. Для запуску емулятора необхідно активізувати безпосередньо програму K580. На екрані можна побачити головне вікно разом з контекстним меню (рис. 2.1).

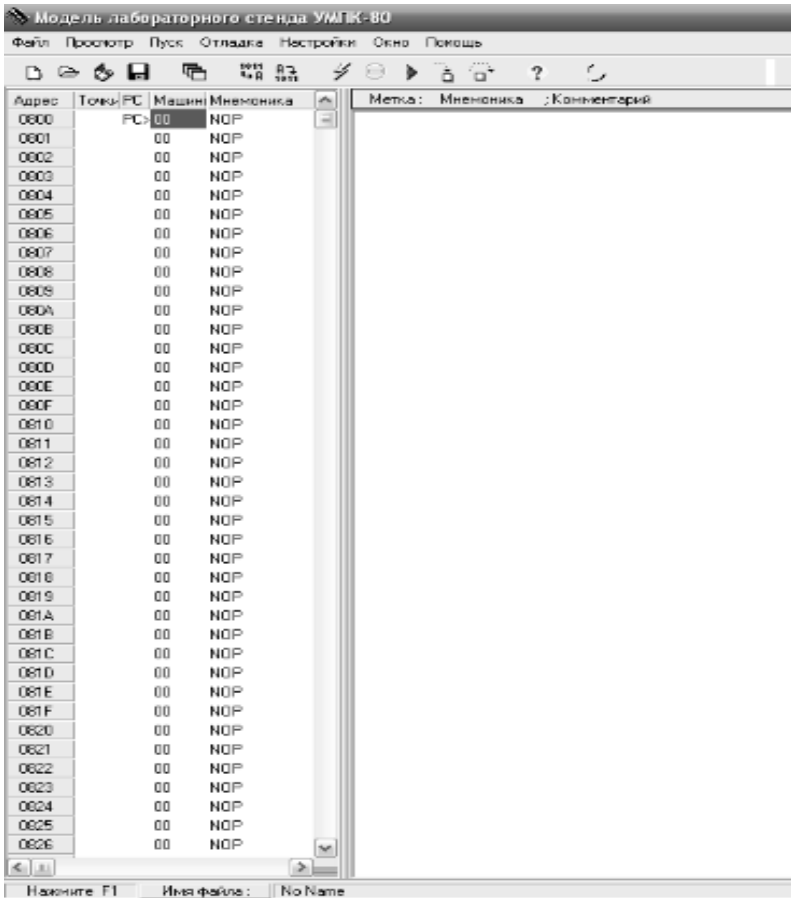


Рисунок 2.1 – Головне вікно емулятора K580

2.2.2 Наступним кроком є внесення машинних кодів до пам'яті. Для цього необхідно курсор поставити на «Машинний код», як це зображено на рис. 2.2. Емулятор автоматично буде заносити мнемокоди (або команди Асемблера), які Ви завантажили.

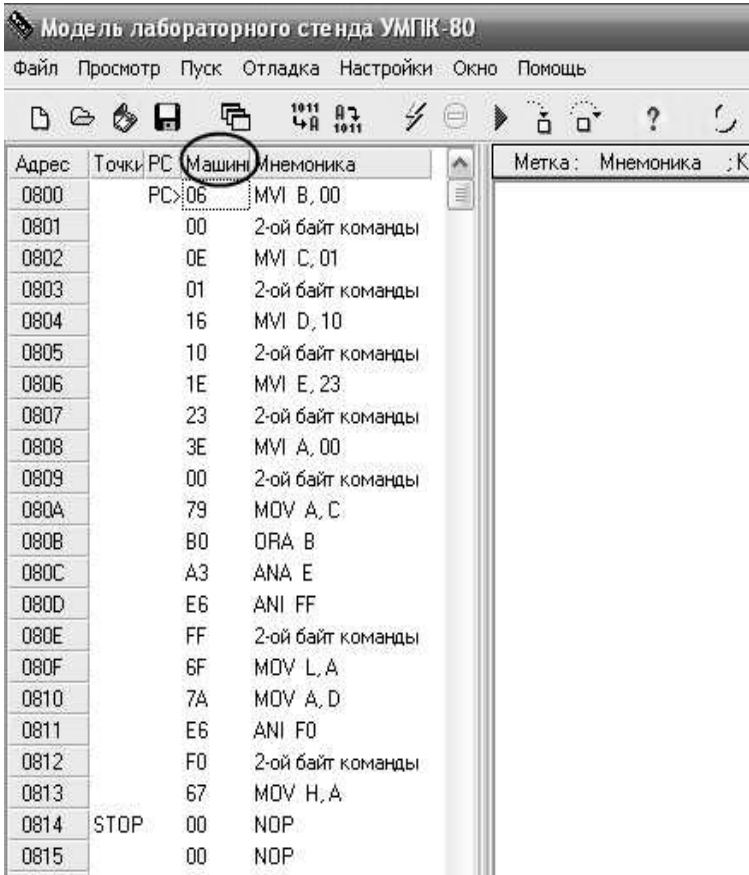


Рисунок 2.2 – Запис кодів команд до пам'яті емулятора

2.2.3 Якщо необхідно записати код в останню комірку пам'яті, треба клікнути лівою клавішею біля наступної адреси для появи слова «STOP». Якщо необхідно встановити початок програми, тобто почат-

кову адресу пам'яті, треба двічі клікнути лівою клавішею біля відповідної адреси до появи знаку «PC», як це зображено на рис. 2.3.

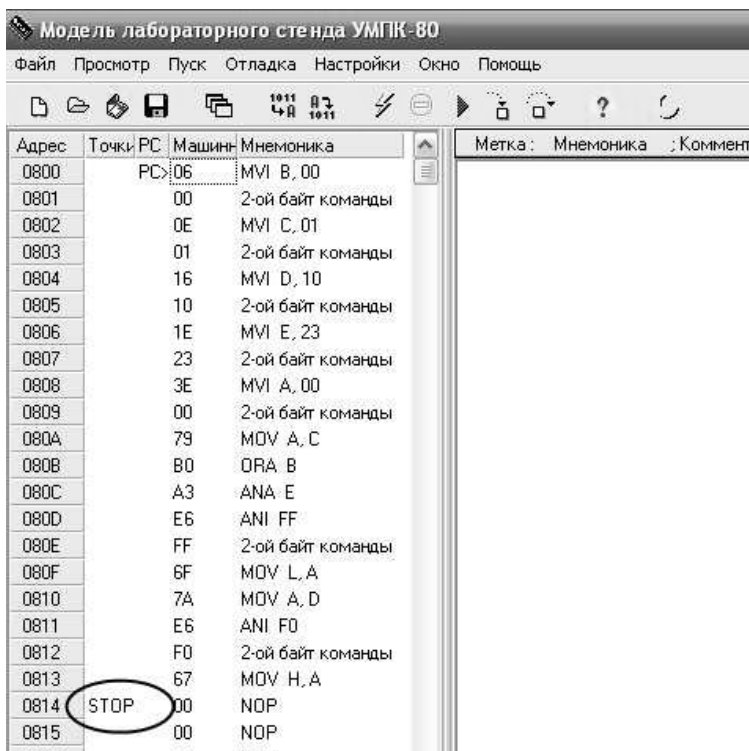


Рисунок 2.3 – Встановлення останньої точки зупину програми

За необхідністю набрану програму можна зберегти в особистих папках «Студент» відповідно до адреси кожного студента.

2.2.4 Для виконання безпосередньо самої програми Ви маєте залишити курсор на початку програми (тобто на першій адресі), зайти в «Пуск» контекстного меню і натиснути на «Выполнить», як зображено на рис. 2.4. при цьому емулятор має зупинитися саме на останній адресі виконуваної програми.

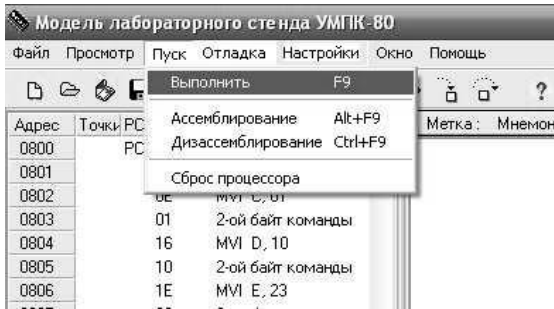


Рисунок 2.4 – Запуск виконання програми

2.2.5 Для перевірки отриманого результату необхідно зайти в «Просмотр» і обрати «Регистры и флаги» у вікні (див. рис. 2.5).

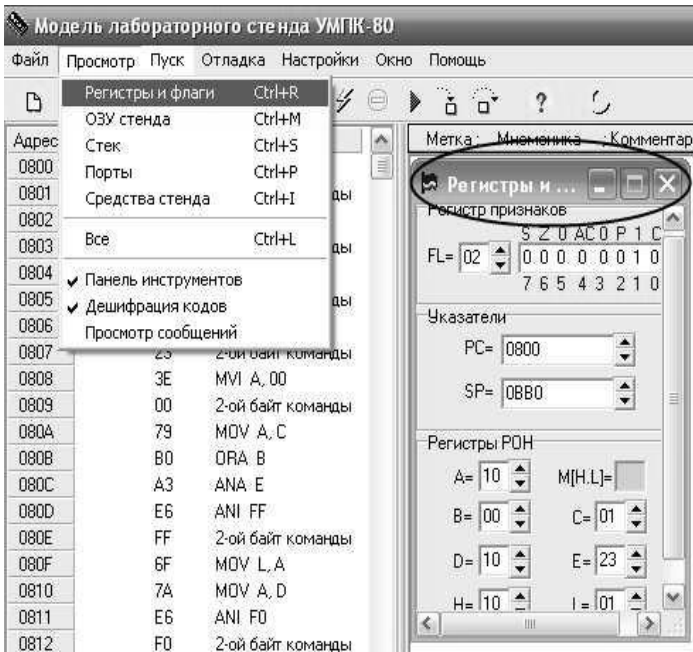


Рисунок 2.5 – Перевірка вмісту регістрів

2.2.6 Для перевірки записаної програми або отриманого результату, що знаходиться в комірці пам'яті, необхідно зайти в «Просмотр» та «ОЗУ станда» (рис. 2.6).

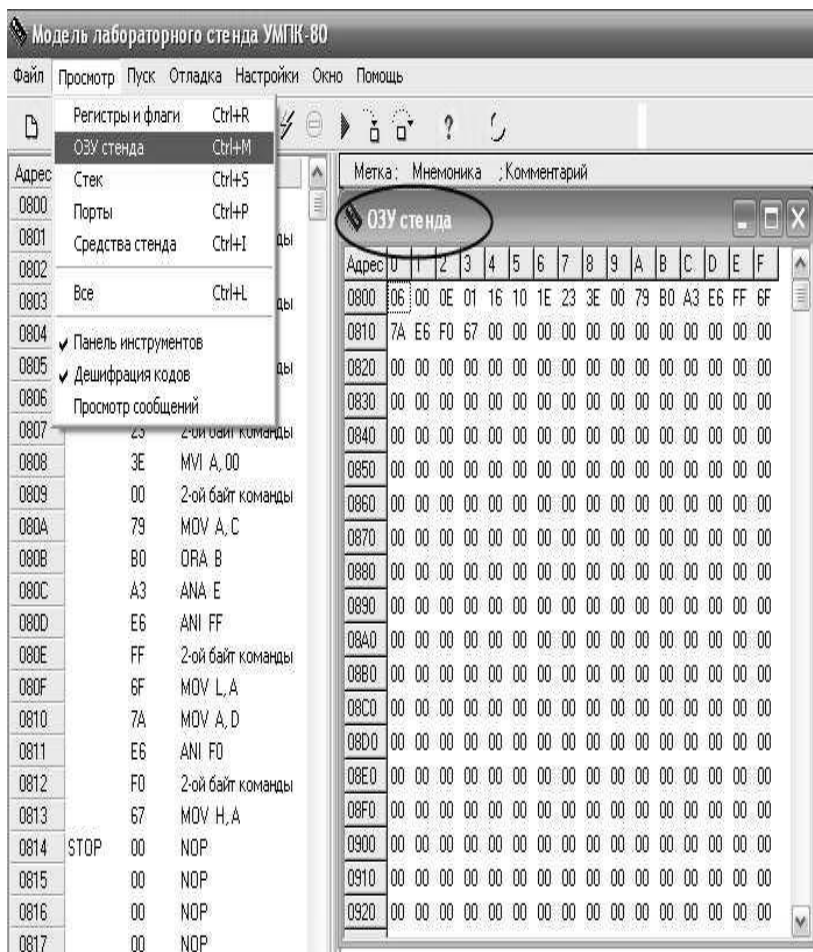


Рисунок 2.6 – Перевірка вмісту комірок пам'яті

2.3 Зміст звіту

Звіт повинний містити:

- назву та мету роботи;
- алгоритми виконання кожної команди відповідно до завдань в підрозділі 2.3;
- висновки.

2.4 Контрольні запитання

2.4.1 Призначення й можливості емулятора програми “Монітор”. Характеристика її системних ознак.

2.4.2 Характеристика основних команд і операцій, їхній зв'язок з функціональною клавіатурою й індикацією.

2.4.3 Основні операції з масивами пам'яті.

2.4.4 Поняття буфера, особливості його використання.

2.4.5 Порядок і особливості роботи з областями пам'яті, що перекриваються.

3 ЛАБОРАТОРНА РОБОТА № 3 АРХІТЕКТУРА МІКРОПРОЦЕСОРА КР580ВМ80А. КОМАНДИ ЗАВАНТАЖЕННЯ ТА ПЕРЕСИЛАННЯ

3.1 Мета роботи

Вивчення структури МП КР580ВМ80А і команд роботи з програмно доступними регістрами цього мікропроцесора.

3.2 Загальні відомості

3.2.1 Організація мікропроцесора КР580ВМ80А.

Структурна схема досліджуваного МП представлена на рисунку 3.1.

До складу МП КР580ВМ80А входять такі вузли:

- шість восьмирозрядних регістрів загального призначення (РЗП) – В, С, D, E, H, L;
- восьмирозрядний акумулятор (А);

- чотири восьмирозрядних реєстри для тимчасового збереження інформації ВР1, ВР2, W і Z;
- восьмирозрядний реєстр ознак (прапорів) PO (F);
- восьмирозрядний арифметико-логічний пристрій (АЛП) паралельної дії;
- схема десяткової корекції (СДК);
- восьмирозрядний реєстр команд (РК);
- шістнадцятирозрядний лічильник команд (ЛК або РС);
- шістнадцятирозрядний показчик стека (ПС або SP);
- шістнадцятирозрядний реєстр адреси (РА);
- дешифратор команд і схема керування машинним циклом (ДШК і СКМЦ);
- мультиплексор (М);
- схема вибору реєстра (СВР);
- пристрій керування (ПК);
- буфер даних (БД);
- буфер адреси (БА).

Для програміста в МП КР580ВМ80А доступними є дев'ять реєстрів: шість РЗП, акумулятор, лічильник команд, показчик стеку, реєстр ознак. РЗП використовуються для збереження чисел, що беруть участь в операції, для показу адреси комірки пам'яті чи шістнадцятирозрядних даних. В останньому випадку восьмирозрядні реєстри В, С, D, Е, Н, L поєднуються в реєстрові пари ВС, DE, HL. У перших реєстрах пар - В, D, Н зберігаються старші байти, а в других - С, Е, L -молодші. Звертання до реєстрової пари здійснюється за іменами першого реєстра. Отже, до перерахованих пар можна звертатися по буквах В, D, Н відповідно. Крім імені кожен РЗП, а також реєстр А мають тризначний двійковий код:

АЛП призначений для виконання арифметичних і логічних операцій над восьмирозрядними даними й операндами.

Акумулятор А – головна частина арифметичного пристрою МП. Всі арифметичні і логічні операції виконуються в АЛП з використанням акумулятора. Для кожної з таких операцій передбачається розміщення одного операнда в акумуляторі, другого – в пам'яті або в РЗП. Результат операції розміщується в акумуляторі. Операції запису в стек і зчитування зі стека показані на рисунку.3.2.

М – це комірка пам'яті, адреса якої міститься в реєстровій парі HL.

Показчик стека SP (ПС). Стек – це область пам'яті, до якої дані записуються, і з якої зчитуються в строго визначеному порядку відповідно до алгоритму LIFO (Last-In, First-Out) – останній увійшов - перший вийшов. За допомогою шістнадцятирозрядного ПС програміст визначає яка область пам'яті виділяється під стек. ПС містить адресу комірки, що доступна для зчитування інформації. Після виконання чергового зчитування вміст ПС автоматично збільшується на 1. Перед записом до стеку вміст ПС зменшується на 1, після чого виконується запис.

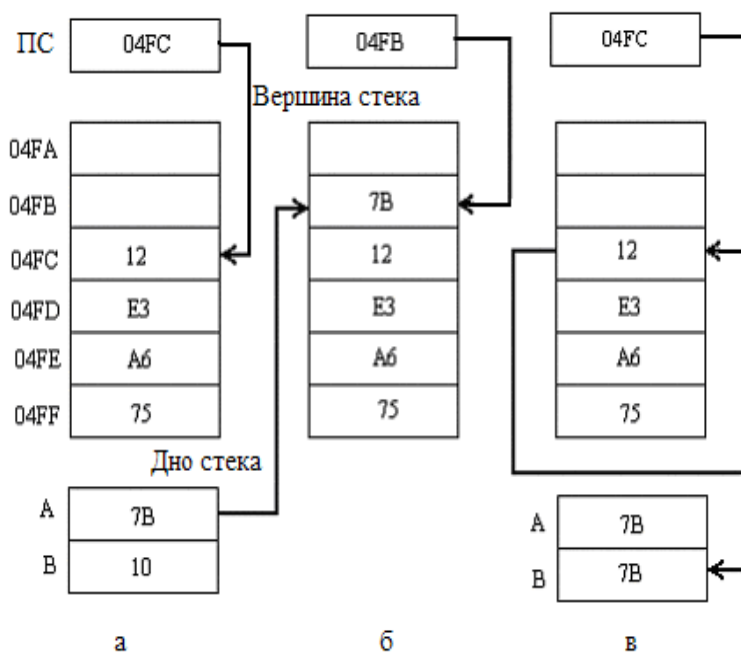


Рисунок 3.2 – Операції зі стеком

а) – початковий стан стека; б) – стек після виконання команди "записати в стек код з акумулятора А"; в) – стек після виконання команди "читати зі стека код у реєстр В"

Реєстр ознак F призначений для зберігання п'яти ознак (прапорів), які з'являються при виконанні деяких операцій. Ці ознаки (біти умов) зберігаються у відповідних розрядах PO (рис.3.3).

РЕГИСТР ОЗНАК F							
7	6	5	4	3	2	1	0
S	Z	O	AC	0	P	1	C

Рисунок 3.3 – Регістр ознак

S (Sign) – ознака знака результату, що зберігається в акумуляторі. При позитивному знаку $S=0$, при від'ємному $S=1$.

Z (Zero) – якщо вміст акумулятора $=0$, то ознака $Z=1$, інакше $Z=0$.

C (Carry) – ознака переносу. $C=1$, якщо при виконанні команд з'являється одиниця переносу зі старшого розряду (переповнення).

AC (Auxiliary Carry) – додаткова ознака переносу. Встановлюється в одиницю, якщо при виконанні команд виникає одиниця з третього розряду числа (з молодшої тетради в старшу).

P (Parity) – ознака парності. $P=1$, якщо кількість одиниць у розрядах акумулятора буде парною. Нульовий результат також відноситься до парного. У протилежному випадку $P=0$.

Розряди 1, 3 і 5 у регістрі ознак не використовуються як ознаки. Повний опис кожної команди МП повинен містити інформацію про те, на які ознаки в регістрі F ця команда впливає. Вміст пари регістрів A і F називають словом стану програми і позначають PSW.

3.2.2 Структура команд.

Двійкове число, обране з пам'яті і, що вказує на виконання означеної операції, називається командою. Двійкове число, яке підлягає обробці, називається операндом. Система команд, реалізована МП, містить 78 базових команд (загальне число команд - 244). Команда може містити від одного до трьох форматних полів. Звичайно команда МП КР580ВМ80А складається з двох форматних полів: поля коду операції (КОП) і поля адреси операнда (адресна частина). У МП КР580 використовуються кілька способів завдання адреси операндів - способів адресації: пряма регістрова адресація, непряма регістрова адресація, пряма адресація і безпосередня адресація. Для різних способів адресації потрібне різне число розрядів у форматі команди. Тому команди МП мають різну довжину - один, два чи три байти.

При прямій регістровій адресації операнд (один чи два) знаходяться в РЗП, адреса якого вказується в адресній частині команди (у двійковій системі адреси мають код 000 – 111, а в шістнадцятирічній - 0-7). Такі команди мають довжину один байт. На рис. 3.4,а приведене двійкове представлення команди пересилання вмісту акумулятора в регістр В. Праворуч зазначений шістнадцятирічний код команди.

При прямій адресації адреса комірки пам'яті вказується в полі адреси команди. Такі команди мають довжину в три байти: перший байт - код операції, другий байт - молодші розряди адреси, третій байт - старші розряди адреси. На рис. 3.4,б показана команда завантаження акумулятора вмістом комірки 7АСЗ. При безпосередній адресації операнд є частиною самої команди і у залежності від його розрядності займає другий чи другий і третій байти команди.

На рис. 3.4, в представлена команда пересилання коду 4С в акумулятор, а на рис. 3.4, г – завантаження в регістрову пару В коду 1F5А. Будь-яка команда має мнемонічний (символічний) опис, що полегшує написання програми для програміста. Наприклад, ADD (скласти), MOV (переслати), XCHG (обмін вмістом регістрових пар D і H) та ін. Поле мнемоніки повинне відокремлюватися від операндів хоча б одним пробілом. Операнди – це числа в шістнадцятирічній системі.

Якщо число починається з букви, то перед буквою ставиться незначущий нуль, інакше це число буде сприйняте як ім'я. Наприклад, 0FАН, 0В7Н, 0DЕН і т.д.

Операнди відокремлюються комою, якщо їх декілька. Пробіли між операндами неприпустимі. Слід пам'ятати, що:

а) максимальне восьмирозрядне число - 0FFH, а шістнадцятирозрядне - 0FFFFH;

б) необхідно по можливості використовувати більш короткі (менше число байтів) команди;

в) якщо команда впливає на означені біти регістра ознак, то решта бітів цього регістра не змінює свого стану.

КОП	Регістр приймач	Регістр джерело	В1	0	1	0	0	0	1	1	1	47
-----	--------------------	--------------------	----	---	---	---	---	---	---	---	---	----

а)

КОП	В1	0	0	1	1	1	0	1	0	3A
Молодший байт адреси	В2	1	1	0	0	0	0	1	1	C3
Старший байт адреси	В3	0	1	1	1	1	0	1	0	7A

б)

КОП	В1	0	0	1	1	1	1	1	0	3E
Операнд	В2	0	1	0	0	1	1	0	0	4C

в)

КОП		0	0	0	0	0	0	0	1	01
Операнд 1		0	1	0	1	1	0	1	0	5A
Операнд 2		0	0	0	1	1	1	1	1	1F

г)

а) – пряма регістрова адресація; б) – пряма адресація;
в), г) – безпосередня адресація.

Рисунок 3.4 – Способи адресації

Усі команди можна розділити на сім груп:

- 1) команди пересилання інформації;
- 2) команди передачі чи керування переходу;
- 3) команди арифметичних і логічних операцій;

- 4) команди організації підпрограм;
- 5) команди уведення й виведення;
- 6) команди роботи зі стеком;
- 7) команди керування станом МП.

У подальшому будемо користатися такими умовними позначеннями:

- R, R1, R2, ... – один з восьмирозрядних РЗП чи акумулятор;
- RP – регістрова пара В,D чи Н, а також покажчик стека;
- data – восьмирозрядний операнд чи його ім'я;
- data16 – шістнадцятирозрядний операнд чи його ім'я;
- addr – адреса комірки пам'яті;
- port – восьмирозрядна адреса ПУВ чи його ім'я.

3.2.3 Команди завантаження РЗП.

Команда **MVI (Move Immediate)** служить для завантаження в РЗП восьмирозрядного двійкового операнда. Довжина її два байти, виконується за сім тактів. Формат команди MVI R, data.

Наприклад: MVI A, 0FFH - завантажити акумулятор числом FFH; MVI C, FFH - завантажити регістр с вмістом імені FFH.

Команда завантаження кожного регістра має свій шістнадцятирічний код. У першому байті вказується КОП і регістр приймач, у другому – операнд. Якщо як регістр R у команді зазначений умовний регістр M, то другий байт команди (операнд) завантажуються в комірку пам'яті, адреса якої попередньо записана у регістрову пару HL.

Таблиця 3.1 - Коди першого байта команди MVI

Регістр - приймач	A	B	C	D	E	H	L	M
Код першого байта команди MVI	3E	06	0E	16	1E	26	2E	36

Як приклад, рекомендується виконати таке завдання: записати в пам'ять, починаючи з адреси 800H, послідовність команд.

Адреса	Команда	Машинний код	Коментар
800	MVI A, 00	3E 00	Завантажити регістр А кодом 00H
802	MVI B, 01	06 01	Завантажити регістр В кодом 01H
804	MVI C, 02	0E 02	Завантажити регістр С кодом 02H
806	MVI D, 03	16 03	Завантажити регістр D кодом 03H
808	MVI E, 04	1E 04	Завантажити регістр Е кодом 04H
80A	MVI H, 05	26 05	Завантажити регістр H кодом 05H
80C	MVI L, 06	2E 06	Завантажити регістр L кодом 06H

Варто пам'ятати, що команда MVI – двобайтна, тому для кожної команди приділяється дві комірки пам'яті. Машинний код команди вводиться побайтно. Так, в комірку 800 записується код 3E, а в комірку 801 – 00 і т.д. Виконайте цю послідовність за допомогою команди <СТ>800 <-> 80E<ВП>. На дисплеї з'явиться адреса зупину 80E.

Перевірити вміст регістрів, використовуючи команду **ПЕРЕГЛЯД І МОДИФІКАЦІЯ ВМІСТУ РЕГІСТРІВ**.

Команда **LXI (Load register pair Immediate)** служить для завантаження зазначеної в команді регістрової пари шістнадцятирозрядними даними. Довжина команди 3 байти, виконується за 10 тактів. У першому байті вказується код операції і регістр-приймач, у другому і третьому – шістнадцятирозрядний операнд. Причому третій байт команди завантажується в перший регістр пари, а другий байт – у другий регістр пари.

Формат команди LXI RP, data 16

У якості RP використовуються регістрові пари B, D, H, SP.

Наприклад: LXI D, 0FF00H - у регістр D записується старший байт – FF, у регістр E молодший – 00.

Коди команди LXI приведені в таблиці 3.2.

Таблиця 3.2 – Коди команди LXI

Адресуємі регістрові пари	B	D	H	SP
Код першого байта команди LXI	01	11	21	31

Як приклад, рекомендується виконати таке завдання:

- запишіть у пам'ять, починаючи з адреси 800H, послідовність команд

Адреса	Команда	Машинний код	Коментар
800	LXI B, 3132H	01 32 31	Завантажити RP BC кодом 3132H
803	LXI D, 3334H	11 34 33	Завантажити RP DE кодом 3334H
806	LXI H, 3536	21 36 35	Завантажити RP HL кодом 3536 H

– виконайте цю послідовність команд за допомогою команди **СТАРТ ПРОГРАМИ**;

– перевірте вміст усіх регістрів, що повинний збігатися з заданими значеннями.

Команди завантаження регістра покажчика стека

Команда безпосереднього завантаження регістра SP має вигляд:

LXI SP, data 16

Команда непрямого завантаження регістра SP має вигляд:

SPHL

Команда однобайтна. Код цієї команди F9. По команді SPHL у покажчик стека завантажуються вміст пари HL. Тому регістрова пара HL попередньо завантажуються потрібним числом.

Як приклад, рекомендується виконати таке завдання:

- запишіть у пам'ять за адресою 800H коди команди.

Адреса	Команда	Машинний код	Коментар
800	LXI SP, 0B10H	31 10 0B	Завантажити покажчик стека SP кодом 0B10H

- виконайте цю команду;
- перевірте вміст регістра SP;
- запишіть у пам'ять за адресою 800H такі команди

Адреса	Команда	Машинний код	Коментар
800	LXI H, 0B30H	21 30 0B	Завантажити RP HL кодом 0B30H
803	SPHL	F9	Завантажити регістр SP кодом з RP

- виконайте ці команди: <СТ> 800 <-> 804 <ВП>;
- перевірте вміст регістра SP побайтно.

Усі команди завантаження РЗП ознаки не формують, тобто стан регістра F не змінюють.

3.2.4 Команди пересилання

Дана команда призначена для передачі інформації з регістра в регістр, з регістра в пам'ять і з пам'яті в регістр. Загальний вид команди:

MOV RI, R2

R1- регістр – приймач;

R2- регістр - джерело.

Команда займає 1 байт і виконується за 5 тактів. Ознак не формує. Наприклад, MOV A, B - переслати вміст регістра B в акумулятор; MOV M, A - переслати вміст акумулятора в комірку пам'яті, адреса якої зберігається в парі HL. При виконанні всіх цих команд вміст регістра - джерела R2 зберігається.

Як приклад, рекомендується виконати таке завдання: запишіть у пам'ять, починаючи з адреси 800H, послідовність команд

Таблиця 3.3 – Коди команди MOV

Адреса	Команда	Машинний код	Коментар
800	MVI A, 00H	3E 00	Обнулити регістр А
802	MOV B, A	47	Переслати вміст А в В
803	MOV C, B	48	Переслати вміст В в С
804	MOV D, C	51	Переслати вміст С у D
805	MOV E, D	5A	Переслати вміст D у E
806	MOV H, E	63	Переслати вміст E в H
807	MOV L, H	6C	Переслати вміст H в L

– виконайте цю послідовність команд;

– перевірте вміст регістрів.

Порівняйте об'єм пам'яті, займаний цією програмою, і тривалість її виконання з об'ємом і тривалістю аналогічної програми обнуління тих же регістрів, виконуваної за допомогою команд MVI для кожного регістра (див. завдання до 3.2.3).

3.2.5 Команда завантаження лічильника команд

По цій команді в лічильник команд записується вміст регістрової пари HL.

Команда має вид: PCHL. Код цієї команди E9.

Наприклад, щоб завантажити в лічильник команд адреса 900 потрібно попередньо завантажити його в RP HL, а потім виконати команду PCHL. Ця команда не має операндов і виконує безумовний перехід з непрямою адресацією. Довжина команди 1байт, тривалість 4 такти. Команда PCHL ознак не формує.

Як приклад, рекомендується виконати таке завдання:

– запишіть у пам'ять за адресою 800H послідовність команд

Адреса	Команда	Машинний код	Коментар
800	LXI H, 0900H	21 00 09	Завантажити RP HL кодом 0900H
803	PCHL	E9	Завантажити PC кодом RP HL. Перейти на адресу 0900H

– виконайте ці команди: <СТ> 800 <-> 804<ВП>.

На дисплеї з'явиться адреса 0900. Це буде означати, що в лічильник команд завантажена адреса 0900H і здійснений перехід на цю адресу.

3.3 Самостійне виконання роботи

3.3.1 Згідно з рис. 3.1 детально вивчити структуру КР580ВМ80А, призначення основних вузлів і сигналів.

3.3.2 Виділити програмно доступні елементи, їхнє призначення і застосування.

3.3.3 З'ясувати основи побудови системи команд, її характеристики.

3.3.4 З'ясувати принципи поділу команд по групах, види адресації.

3.3.5 Визначити структуру команди, її опис.

3.3.6 Написати і виконати програму завантаження РЗП кодами, що відповідають вашому зросту в сантиметрах в шістнадцятирічному коді. Перевірити правильність виконання програми.

3.3.7 Написати і виконати програму завантаження регістрових пар кодами, що відповідають вашому зросту. Перевірити правильність виконання програми.

3.3.8 Написати і виконати програму завантаження регістра SP кодами, що відповідають вашому зросту в сантиметрах в шістнадцятирічному коді:

а) командою безпосереднього завантаження:

SP: = 0820H; SP: = 0840H; SP: = 0AFFH;

б) командою непрямого завантаження:

SP: = 0825H ;SP: = 0EAFH; SP: = 05FCH.

Перевірити правильність виконання програми.

3.3.9 Написати і виконати програму пересилання для всіх РЗП, попередньо завантаживши один з них кодами, що відповідають вашому зросту в сантиметрах в шістнадцятирічному коді.

3.4 Зміст звіту

- назву та мету роботи;
- написані програми відповідно до завдань в підрозділі 3.3 в мнемокодах та машинних кодах з коментарями;
- висновки.

3.5 Контрольні запитання

3.5.1 Розрядність основних шин.

3.5.2 Програмно доступні вузли мікропроцесора, їхня коротка характеристика.

3.5.3 Система команд, група команд, опис команди.

3.5.4 Особливості адресації.

3.5.5 Команди завантаження регістрів і пари регістрів.

3.5.6 Команди пересилання.

4 ЛАБОРАТОРНА РОБОТА № 4 ВИВЧЕННЯ ПОКРОКОВИХ РЕЖИМІВ РОБОТИ НМК

4.1 Мета роботи

Вивчення процесу виконання команд і налагодження програм.

4.2 Загальні відомості

Покрокові режими надають користувачу можливість виконувати програму, одну команду за іншою, з паузами чи зупинками на кожному кроці. Під час зупинки на індикаторі фіксуються стан операційного пристрою (ОП), адреси й коди команд, призначених для вибірки (виконання).

Для НМК існує два режими роботи за кроками:

- режим роботи за машинними циклами;
- режим роботи за командами.

Режим роботи НМК за машинними циклами використовується для вивчення процесу виконання команди мікропроцесором, а режим роботи НМК за командами використовується для вивчення команд і налагодження програм. Розглянемо режим роботи НМК за машинними циклами. Для цього встановлюється даний режим роботи і виконується задана послідовність команд. Розглянемо приклад, приведений у таблиці 4.1.

Таблиця 4.1 – Приклад програми

Адреса	Команда	Код	Коментар
800	NOP	00	
801	MOV A, B	78	$B \leftarrow A$
802	ADD	81	$A = A + C$
803	RRC	0F	Цикл зміщується на один вправо
804	ANI 0FH	E6 0F	$A = AND\ 0FH$
806	PUSH H	E5	Запис стека = HL
807	LXI H, 900H	21 00 09	Завантаження HL=900H, адреса M
80A	MOV M, A	77	Запис M = A на адресу HL = 900H
80B	MVI A, OFFH	3E FF	Завантаження A = OFFH
80D	SUB M	96	$A = A$ на адресу HL
80E	POP H	E1	$HL \leftarrow$ стек
80F	JMP 800H	C3 00 08	Перехід на адресу 800

Режим роботи НМК за командами відрізняється від розглянутого вище тим, що крокам відповідають закінчені операції з командами. Для цього задається установка зазначеного режиму і виконується задана послідовність команд. Розглянемо покомандний режим роботи НМК на прикладі послідовності команд, приведених у таблиці 4.2.

Як видно наведені команди для виконання вимагають різне число кроків, що відповідають числу машинних циклів. Перша команда NOP складається з одного машинного циклу і для її виконання досить виконати один крок. Остання виконана команда наведеної послідовності виконується протягом трьох машинних циклів і передає керування на початок послідовності команд.

Таблиця 4.2 – Приклад програми

Адреса	Команда	Код	Коментар
1	2	3	4
800	LXI H, 900H	21 00 09	Адреса М, завантаження HL = 900H
803	MVI M, 0AAH	36 AA	Запис M=0AAH до адреси HL
805	MVI A, 55H	3E 55	Завантаження A = 55H
807	ADD M	86	A=A+M до адреси HL
808	STA 901H	32 01 09	Запис M=A до адреси 901H
80B	LXI D, 902H	11 02 09	Завантаження DE=902H
80E	LDAX D	1A	Завантаження A=M до адреси DE
80F	ORA M	B6	A=A OR M до адреси HL
810	LXI B, 903H	01 03 09	Завантаження BC=903H
813	STAX B	02	Запис M=A до адреси BC
814	JMP 800H	C3 00 08	Перехід до адреси 800H

4.3 Самостійне виконання роботи

4.3.1 Освоїти операції встановлення режимів, виконуючи програми, приведені в таблиці 4.1 та таблиці 4.2.

4.3.2 Для встановлення режиму роботи НМК за машинними циклами необхідно виконати такі дії:

- увімкнути НМК і привести в робочий стан;
- ввести програму з таблиці 4.3 до пам'яті НМК;
- натиснути до фіксації клавішу <РБ/ШГ>;
- натиснути до фіксації клавішу <КМ/ЦК>.

4.3.3 Покроковий режим роботи НМК реалізується натисканням клавіші <ШГ>.

4.3.4 Записати результати, що висвітлюються на шиних даних, адреси та стану ОП в таблицю 4.3 у відповідних кодах.

4.3.5 Продовжити виконання програми до моменту, коли не з'явиться перший шаг програми на світлодіодах. Після чого відключити клавішу <РБ/ШГ> та клавішу <КМ/ЦК>. Натиснути кнопку <СБРОС>.

4.3.6 Для встановлення режиму роботи НМК за командами необхідно виконати такі дії:

- ввести програму з таблиці 4.4 до пам'яті НМК;
- натиснути до фіксації клавішу <РБ/ШГ>.

4.3.7 Виконати програму, наведену в таблиці 4.4 в режимі роботи за командами.

4.3.8 Решта маніпуляцій така сама, як в п.п.4.3.4 та 4.3.5.

Кожній таблиці завдання відповідає своя таблиця результатів, у яку записують двійкові коди адреси, даних і коди регістра стану по кожному кроці. По закінченню завдання коди двійкові переводять у шістнадцятиричну і перевіряють відповідність до вихідних даних.

В покроковому режимі роботи НМК передбачена світлова індикація стану шин: шини адреси (ША), шини даних (ШД) та шини регістра стану (ШРС) – відображена в двійковому коді. При цьому стану 1 відповідає стан, що світиться, індикатора розряду, а стану 0 – що не світиться. Розряди складають шини. Індикатор шин наведений на рис. 4.1. Розряди в ША та ШД згруповані у виді тетрад, що полегшує переведення двійкових чисел в шістнадцятиричні. Двійкова тетрада відповідає шістнадцятиричному числу (таблиця 4.6). Таким чином ШД відображає два розряди шістнадцятиричного числа, ША – чотири. ШРС

має вісім розрядів, кожен з яких самостійний та являє собою слово стану МП. Можливі стани ОП приведені в таблиці 1.2, а визначення кожного біта регістра стану в таблиці 1.3. Команди МП КР580ВМ80А наведені в таблиці 4.7.

Таблиця 4.3 – Програма для виконання за машинними циклами

Адреса	Команда	Код	Коментар
800	LXI B, 940H	01 40 09	Завантаження BC=940H
803	LXI D, 960H	11 60 09	Завантаження DE=960H
806	MVI L, 10H	2E 10	Завантаження L=10H
808	LDAX B	0A	Завантаження A=M до адреси BC
809	STAX D	12	Завантаження M=A до адреси DE
80A	INX B	03	BC=BC+1
80B	INX D	13	DE=DE+1
80C	DCR L	2D	L=L - 1
80D	JMP 800H	C3 00 08	Перехід до адреси 800H

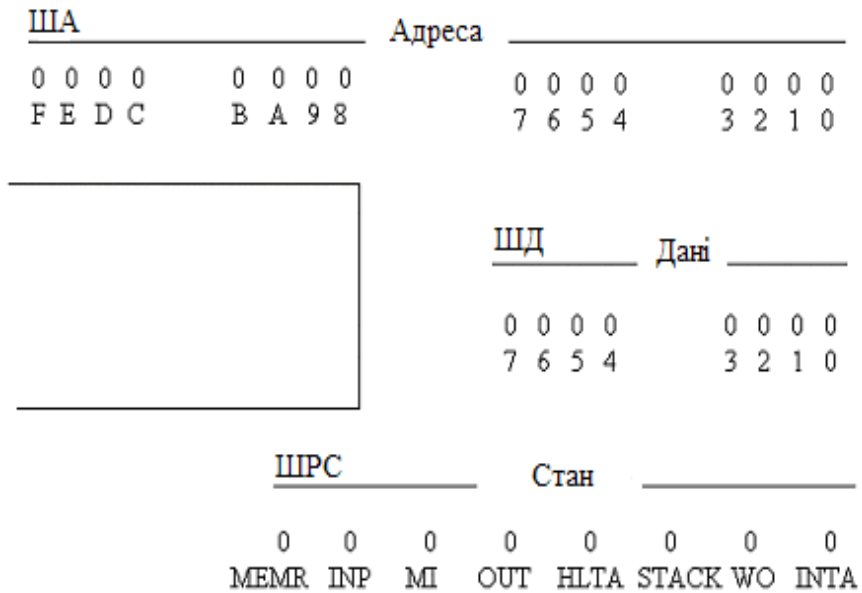


Рисунок 4.1 – Індикатор стану шин

Таблиця 4.4 – Програма для виконання за командами

Адреса	Команда	Код	Коментар
1	2	3	4
800	LXI H, 900H	21 00 09	Завантаження адреси HL=900H
803	LXI B, 920H	01 20 09	Завантаження адреси BC=920H
806	LXI D, 940H	11 40 09	Завантаження адреси DE=940H
809	LDAX B	0A	Завантаження A=M до адреси BC
80A	SUB M	96	A=A2-M до адреси H
80B	STA 900H	32 50 09	Запис M=A до адреси 950H
80E	LDAX D	1A	Завантаження A=M до адреси DE
80F	AND M	A6	A=A AND M до адреси HL
810	RAL	17	Зсув регістра A вліво
811	STA 951H	32 51 09	Запис M=A до адреси 951H
814	LDA 950H	3A 50 09	Запис A=M до адреси 950H
817	MOV L, A	6F	Перехід L ← A
818	LDA 951H	3A 51 09	A=M до адреси 951H
81B	MOV H, A	67	Перенос вмісту H ← A
81C	SHLD 952H	22 52 09	Запис HL до адреси 952H-953H
81F	JMP 800H	C3 00 08	Перехід до адреси 800H

Таблиця 4.5 – Таблиця результатів

Шаг	Код шини адреси		Код шини даних		Шина стану	
	В	Н	В	Н	Код	Операція
1						
2						
..						

4.4 Зміст звіту

Звіт повинен містити:

- назву та мету роботи;
- вихідні послідовності команд, результати, оброблені дані;
- висновки по роботі.

4.5 Контрольні запитання

4.5.1 Призначення й особливості покрокового режиму роботи МП системи.

4.5.2 Види кроків, їхня характеристика.

4.5.3 Цикли в роботі МП. Їхня характеристика.

4.5.4 Відображення поточної інформації в покрокових режимах.

4.5.5 Перевід двійкової цифри в шістнадцятирічний код.

4.5.6 Відображення стану операційного пристрою.

Таблиця 4.6 – Таблиця кодів систем числення

Число D	B	Q	H
0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F

Таблица 4.7 – Команды МП

		SECOND TETRAD																
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
FIRST TETRAD	0	NOP	LXI BC	STAX BC	INX BC	INR B	DCR B	MVI B	RLC	--	DAD BC	LDAX BC	DCX BC	INR C	DCR C	MVI C	RRC	0
	1	--	LXI DE	STAX DE	INX DE	INR D	DCR D	MVI D	RAL	--	DAD DE	LDAX DE	DCX DE	INR E	DCR E	MVI E	RAR	1
	2	--	LXI HL	SHLD HL	INX HL	INR H	DCR H	MVI H	DAA	--	DAD HL	LHLD HL	DCX HL	INR L	DCR L	MVI L	CMA	2
	3	--	LXI SP	STA adr	INX SP	INR M	DCR M	MVI M	STC	--	DAD SP	LDA adr	DCX SP	INR M	DCR M	MVI A	CMC	3
	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	MOV C, M	MOV C, A	4
	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	MOV E, M	MOV E, A	5
	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	MOV L, M	MOV L, A	6
	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	MOV A, M	MOV A, A	7
	8	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC	ADC E	ADC H	ADC L	ADC M	ADC A	8
	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	9
	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	A
	B	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	B
	C	RNZ	POP BC	JNZ adr	JMP adr	CNZadr	PUSH BC	ADI data	RST0	RZ	RET	JZ adr	--	CZ adr	CALL adr	ACI data	RST1	C
	D	RNC	POP DE	JNC adr	OUT port n	CNC adr	PUSH DE	SUI data	RST2	RC	--	JC adr	IN port n	CC adr	--	SBI data	RST3	D
	E	RPO	POP HL	JPO adr	XTHL	CPO adr	PUSH HL	ANI data	RST4	RPE	PCHL	JPE adr	XCHG	CPE adr	--	XRI data	RST5	E
	F	RP	POP psw	JP adr	DI	CP adr	PUSH psw	ORI data	RST6	RM	SPHL	JM adr	EI	CM adr	--	CPI data	RST7	F

5 ЛАБОРАТОРНА РОБОТА № 5

АДРЕСАЦІЯ ПАМ'ЯТІ. МЕТОДИ І КОМАНДИ

РОБОТИ З ПАМ'ЯТТЮ

5.1 Мета роботи

Вивчення методів адресації і команд роботи з пам'яттю.

5.2 Загальні відомості

Пам'яттю називають пристрій, що забезпечує збереження команд і даних. Розрізняють пам'ять постійну (довгострокову) і оперативну. Відповідно і пристрої, що реалізують кожний вид пам'яті, скорочено називають ПЗП і ОЗП. Доступним користувачу практично у всіх основних режимах роботи НМК є ОЗП, його роботу надалі і слід розуміти при використанні терміна пам'ять. Пристрій пам'яті складається з блоків однакового розміру - комірок пам'яті, розміром один байт кожна, призначених для збереження одного слова інформації в двійковій коді. У свою чергу кожна комірка складається з елементів пам'яті, стан кожного з яких відповідає одній двійковій цифрі - біту інформації. Комірки пам'яті нумеруються числами від 0 до 65535, що називають адресами. Часто для зручності використовують шістнадцятирічні значення адрес, тоді діапазон адресації НМК складає 0000H-FFFFH. Щоб записати в пам'ять слово, належить подати на шину адреси пам'яті сигнали, що відповідають адресі комірки, у яку необхідно помістити записуване слово, і подати необхідні сигнали, що відповідають значенню слова, на шину запису.

Пам'ять організована таким чином, що задане слово буде передано в комірку із зазначеною адресою і може зберігатися там як завгодно довго. В ОЗП слово зберігається доти, поки подана живляча напруга. У будь-який момент, звернувшись до пам'яті, можна одержати вміст збереженого слова (копію). Для цього вказується необхідна адреса. При зчитуванні вміст комірки пам'яті не змінюється. Час доступу не залежить від адреси комірки пам'яті. В НМК доступними користувачу є комірки з адресами з 0800 до FFC9. Адреси портів уведення-виведення можуть знаходитися в межах 00H-FFH, програма «Монітор» займає простір FFC0H - FFFFH. Для проміжного оперативного

збереження слів застосовуються реєстри і реєстрові пари, що складаються з набору запам'ятовуючих елементів, пронумерованих за розрядами.

Розрізняють такі засоби адресації:

1) пряму, 2) безпосередню, 3) непряму, 4) пряму реєстрову, 5) непряму реєстрову, 6) індексну, 7) відносну, 8) базову або індексну-базову, 9) неявну і 10) стекову.

1) Пряма адресація забезпечує доступ до обмеженої частини адресного простору, оскільки в коді команди записується лише невелике число біт адреси (коротка пряма адреса), що доповнюється нулями в інших розрядах. Іноді для розміщення виконавчої адреси використовується збільшення команди до двох або трьох машинних слів. Молодший байт адреси міститься в другому байті команди, а старший – у третьому.

2) Безпосередня адресація (рис. 5.1, б) забезпечує швидке виконання команди, оскільки в полі адреси записана не адреса операнда, а власне операнд. Після вибірки такі команди виконуються без обертання до пам'яті, реалізуючи, наприклад, занесення коефіцієнта в РЗП або інші дії з константами.

3) Непряма (косвенная) адресація (рис. 5.1, в) передбачає вибірку адреси операнда з комірки пам'яті, адреса якого вказується в команді. Цей засіб можна назвати «адресацією адреси». Він дозволяє виконувати різноманітні дії з адресами, що знаходяться в пам'яті, не змінюючи команд програми. Це спрощує адресацію елементів масивів, реалізацію розгалуження (ветвления) і т.д.

Реєстрова адресація. Вибірка операндів пришвидшується, якщо вони розміщені в реєстрах РЗП процесора. Для їхньої адресації (вказівки номера) потрібно усього 3 - 4 біта в коді команди. При такій 4) прямій реєстровій адресації (рис. 5.1, г) скорочується час виконання команд, але використовані дані попередньо повинні бути сформовані в процесорі або занесені ззовні. Для роботи з масивами, а також у багатьох інших випадках зручна 5) непряма реєстрова адресація (рис. 5.1, д), коли адреса операнду знаходиться в одному із реєстрів процесора і для обертання до пам'яті його вміст пересилається, як виконавча адреса. При цьому з'являється проста можливість зміни адресів даних, наприклад, додавання (або віднімання) по одиниці до адреси в реєстрі, що організує лічильник

адреси. Цей засіб одержав назву автоінкрементної (+1) і автодекрементної (-1) адресації (рис.5.1, е).

б) Індексна адресація. Одним із засобів зміни виконавчих адрес без модифікації команд є використання спеціального зсуву (індексу), який записується в окремому індексному реєстрі процесора і додається до адреси команди, забезпечуючи, наприклад, вибірку елементів масивів по їхніх індексах у циклах. Спеціальні засоби адресації ОП показані на рис.5.1. Індексна адресація (рис. 5.1 ,а) призводить до деяких затримок витягу операнда, тому що необхідно обчислити його адресу, що звичайно виконується АЛП або спеціальним суматором у процесорі. Варіант цього засобу адресації можна розглядати як непряму регистрову адресацію на основі індексного реєстра, до вмісту якого додається зсув із поля адреси команди.

7) Відносна адресація. Одним із засобів формування складової адреси є відносна адресація або базова (рис. 5.1, б), що дозволяє робити налаштування програми, і зокрема, команд організації циклів, переходів та інших на адреси, що змінюються. Адреса операнду, що виконується, обчислюється шляхом додавання базової адреси (бази) і позитивного або негативного зсуву від поля адреси команди. У якості базового реєстра використовується лічильник команд, до номерів яких задаються як адреси даних, так і адреси переходів. Відносна адресація забезпечує можливість переміщення програм і даних у пам'яті ЕОМ без зміни команд.

8) Базова або індексна-базова адресація. Для обертання до пам'яті великого об'єму використовуються складові адреси, коли в коді команди записується лише зсув, а виконувана адреса утворюється додатком до зсуву розрядів базової перемінної (покажчика), що зберігається в спеціальному базовому реєстрі, а в ряді випадків і індексу, номер реєстра якого визначається в коді команди. Механізм індексної-базової адресації із зсувом показаний на рис. 5.1, в. Він дозволяє істотно розширити область оперативної пам'яті (наприклад, до 1 Мбайта) при широких можливостях зміни (переміщення) виконавчих адрес без модифікації програм.

9) Неявна адресація. Ряд команд в ЕОМ виконується без спеціальної вказівки місця положення операнду, оскільки по характеру самої операції це місце однозначно припускається,

наприклад, команди зсувів або пересилок у РЗП. Це неявний засіб адресації (рис. 5.1, г), який забезпечує швидке виконання команди.

10) Стекова адресація. Роботу з виділеним в ОП стеком забезпечує спеціальний регістр - покажчик стеку (ПС), у котрому постійно зберігається адреса «верхівки стеку». Для занесення в стек або вибірки зі стеку використовується варіант непрямой регістрової адресації з автозменшенням (-1) і автопідвищенням (+1), що базується на ПС (рис. 5.1, д). Команди зі стековою адресацією дозволяють, ефективно реалізувати процедури переривання програм і переходів на підпрограми, чітко зберігаючи послідовність їхнього виконання.

У кожній ЕОМ використовуються ті або інші засоби адресації в залежності від її організації і призначення.

У мікропроцесорній системі застосовуються два методи адресації до пам'яті: безпосередня і непряма. Розрізняють такі види адресації: регістрову, непряму - регістрову, безпосередню і пряму. При безпосередній адресації адреса комірки пам'яті вказується в другому і третьому байтах команди. Формат команди в загальному виді: КОП АНАL,

де КОП - код операції,

АНАL - адреса комірки пам'яті (АН - старший байт адреси, АL - молодший байт адреси).

Через те, що зазначена команда трибайтна, вона буде розміщена в трьох комірках пам'яті. Після байта коду операції розташовуються молодший байт адреси, а потім – старший:

n КОП;

n + 1 АL;

n + 2 АН.

5.2.2 Існує дві команди безпосереднього запису в пам'ять.

STA АНАL - запис у пам'ять по безпосередній адресі АНАL вмісту регістра А.

SHLD АНАL - запис у пам'ять вмісту регістрової пари HL.

По даній команді за адресою АНАL записується вміст регістра L, а за адресою АНАL + 1 - вміст регістра H.

Для прикладу виконайте таке завдання:

– запишіть у пам'ять команди з таблиці 5.1;

Таблиця 5.1

Адреса	Команда	Код	Коментар
800	STA 880H	32 80 08	Запис у пам'ять з регістра А за адресою 880H
803	SHLD 890H	22 90 08	Запис у пам'ять: з регістра L за адресою 890H; з регістра H за адресою 891H.

– уведіть значення регістрів: А = 37H, H = 55H, L = ААН;

– виконайте команду: <СТ> 800 <-> 806 <ВП>;

– перевірте результат за значеннями вмісту комірок 880H, 890H, 891H.

5.2.3 Непряма адресація припускає, що адреса комірки пам'яті розташована у регістрових парах HL, BC, DE. Для кожної конкретної команди роботи з пам'яттю закріплена своя регістрова пара. Тому перш, ніж задати команду необхідно вказати адресу у відповідній регістровій парі.

Наприклад:

1) LXI H, 800 H – запис адреси в регістр HL;

MOV M, A – запис у пам'ять з регістра А за адресою HL.

2) LXI D 900H – запис адреси в регістр DE;

STAX D – запис у пам'ять з регістра А за адресою DE.

Команди читання пам'яті так само розрізняють за типами адресації на безпосередню і непряму.

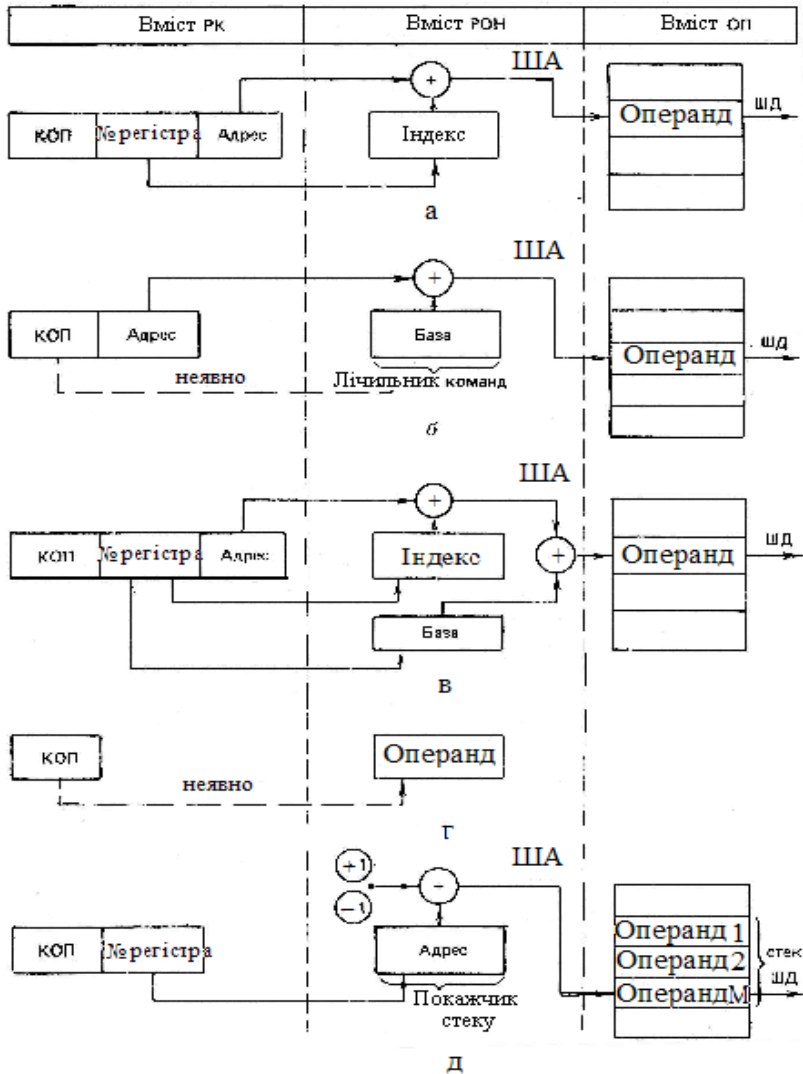
5.2.4 Команди безпосереднього читання пам'яті:

LDA ANAL - читання пам'яті по безпосередній адресі ANAL у регістр А;

LHLD ANAL - читання пам'яті по безпосередній адресі ANAL у регістрову пару HL.

При цьому в регістр L записується вміст комірки ANAL, а в регістр H - вміст комірки з адресою ANAL + 1.

Як приклад, запишіть у пам'ять команди з табл. 5.2:



а) – індексна, б) – відносна, в) – індексно-базова,
г) – неявна, д) – стекова

Рисунок 5.1 – Спеціальні засоби адресації.

Таблиця 5.2

Адреса	Команда	Код	Коментар
800	LDA 880H	3A 80 08	Читання в регістр А з комірки 880H
803	LHLD 890H	2A 90 08	Читання в регістр L з комірки 890H;

- виконайте зазначені команди: <СТ> 800 <-> 806 <ВП>;
- перегляньте вміст регістрів: А = 37 H, H = 55 H, L = AA H.

5.2.5 Команди читання - запису пам'яті при адресації через регістрову пару HL мають вид:

MOV M, R – запис у пам'ять вмісту регістра;

MOV R, M – завантаження регістра з пам'яті,

де R – регістр загального призначення А, В, С, D, E, H, L.

Як приклад, запишіть в пам'ять програму з табл. 5.3:

- виконайте команду: <СТ> 800 <-> 826 <ВП>;
- перевірте вміст комірок пам'яті: 900 H = AAH, 901 = CCH, 902 = BBH, 903 = EEH, 904 H = DDH, 905 H = 09H, 906 H = 06H.

Таблиця 5.3

Адреса	Команда	Код	Коментар
1	2	3	4
800	MVI A, AAH	3E AA	Завантаження регістра А
802	MVI B, BBH	06 BB	Завантаження регістра В
804	MVI C, CCH	0CC	Завантаження регістра С
806	MVI D, DDH	16 DD	Завантаження регістра D
808	MVI E, EEH	1E EE	Завантаження регістра E

Продовження таблиці 5.3

1	2	3	4
80A	LXI H, 900H	21 00 09	Завантаження HL = 900H, адреса M
80D	MOV M, A	77	Запис M=A, за адресою HL
80E	LXI H, 901H	21 01 09	----- ‘ ’ -----
811	MOV M, C	71	----- ‘ ’ -----
812	LXI H, 902H	21 02 09	----- ‘ ’ -----
815	MOV M, B	70	----- ‘ ’ -----
816	LXI H, 903H	21 03 09	----- ‘ ’ -----
819	MOV M, E	73	----- ‘ ’ -----
81A	LXI H, 904H	21 04 09	----- ‘ ’ -----
81D	MOV M, D	72	----- ‘ ’ -----
81E	LXI H, 905 H	21 05 09	----- ‘ ’ -----
821	MOV M, H	74	----- ‘ ’ -----
822	LXI H, 906 H	21 06 09	----- ‘ ’ -----
825	MOV M, L	75	----- ‘ ’ -----

Як приклад, запишіть в пам'ять програму з таблиці 5.4:

- виконайте команду: <СТ> 800 <-> 81С <ВП>;
- перевірте вміст регістрів:
A = DDH, B = EEH, C = BBH, D = CCH,
E = AAH, H = 09H, L = 06H.

Таблиця 5.4

Адреса	Команда	Код	Коментар
800	LXI H, 900 H	21 00 09	Завантаження HL = 900 адресою M
803	MOV E, M	5E	Читання E=M, за адресою HL
804	LXI H, 901H	21 01 09	і т.д.
807	MOV D, M	56	----- ‘ ’ -----
808	LXI H, 902H	21 02 09	----- ‘ ’ -----
80B	MOV C, M	4E	----- ‘ ’ -----
80C	LXI H, 903H	21 03 09	----- ‘ ’ -----
80F	MOV B, M	46	----- ‘ ’ -----
810	LXI H, 904H	21 04 09	----- ‘ ’ -----
813	MOV A, M	7E	----- ‘ ’ -----
814	LXI H, 905H	21 05 09	----- ‘ ’ -----
817	MOV H, M	66	----- ‘ ’ -----
818	LXI H, 906 H	21 06 09	----- ‘ ’ -----
81B	MOV L, M	6E	----- ‘ ’ -----

5.2.6 Команди читання – запису при адресації з використанням регістрових пар BC і DE мають вид:

STAX B – запис вмісту регістра A в пам'ять за адресою зазначеною в регістровій парі BC;

STAX D – запис з регістра A в пам'ять за адресою з регістрової пари DE;

LDAX D – читання вмісту пам'яті в регістр A за адресою зазначеному в регістровій парі DE;

LDAX B – читання з пам'яті в регістр A за адресою з регістрової пари BC;

Як приклад, виконайте таке завдання:

– - записати в пам'ять програму з таблиці 5.5;

Таблиця 5.5

Адреса	Команда	Код	Коментар
800	LXI B, 900H	01 00 09	Завантаження BC = 900H
803	MVI A, 0FH	3E 0F	Завантаження A = 0FH
805	STAX B	02	Запис M = A за адресою BC
806	LXI D, 910H	11 10 09	Завантаження DE = 910H
809	MVI A, 0F0H	3E F0	Завантаження A = F0H
80B	STAX D	12	Запис M = A за адресою DE

- виконайте зазначену послідовність команд: <СТ> 800 <-> 80C <ВП>;
- перевірте вміст комірок 900H і 910H; 900H = 0FH, 910H = F0H.

Як приклад, запишіть в пам'ять програму з таблиці 5.6;

Таблиця 5.6

Адреса	Команда	Код	Коментар
800	LXI D, 900H	11 00 09	Завантаження DE = 900H
803	LDAX D	1A	Читання A = M за адресою DE
804	MOV L, A	6F	Пересилання L ← A
805	LXI B, 910H	01 10 09	Завантаження BC = 910H
808	LDAX B	0A	Читання A = H за адресою BC
809	MOV H, A	67	Пересилання H ← A

- виконайте зазначену послідовність команд: <СТ> 800 <-> 80A <ВП>; - перевірте вміст регістрів H, L: H = F0H, L = 0FH.

5.3 Самостійне виконання роботи

5.3.1 Вивчити структуру пам'яті, її функціональні особливості, способи запису і відтворення вмісту, основні команди, їхні формати, способи опису. Виконати умови прикладів.

5.3.2 Написати і виконати програми запису даних у пам'ять, тобто записати дані з регістра А, використовуючи команду завантаження регістра А і команду запису в пам'ять регістра А за безпосередньою адресою. Вихідні дані відповідають Вашому зросту. Відповідно до таблиці 5.7 згідно до номерів варіанту записати дані з регістрової пари HL. Використовувати команди завантаження HL і запису в пам'ять з регістрової пари за безпосередньою адресою.

Таблиця 5. 7

Варіант	1		2		3	
Адреса	920	921	925	926	92C	92D
Дані	31	32	4F	50	FD	01

Перевірте правильність роботи програм, результати показати в звіті з текстами вихідних програм.

5.3.3 Написати і виконати програми завантаження регістрів, використовуючи команди читання з пам'яті в регістр за безпосередньою адресою і команди пересилання. Дані взяті з таблиці 5.8.

Таблиця 5.8

Варіант	1	2	3
Адреса	900	905	90C
Дані	зріст	зріст	зріст
Регістр	B	C	D

Виконати аналогічне завдання по таблиці 5.9, використовуючи регістрову пару HL.

Таблиця 5.9

Варіант	1		2		3	
Адреса	920	921	925	926	92C	92D
Дані	зріст	зріст	зріст	зріст	зріст	зріст
Регістр	B	C	D	E	H	L

Переписати дані з однієї ділянки пам'яті в іншу відповідно до таблиці 5.10

Таблиця 5.10

Варіант	1	2	3
Адреса 1	940	941	943
Адреса 2	960	961	963
Дані	00	01	03

5.3.4 Написати і виконати програми читання-запису при адресації через регістрову пару HL відповідно до таблиці 5.11, записати в пам'ять вміст регістрів.

Таблиця 5.11

Варіант	1		2		3	
Регістр	B	C	D	E	H	L
Вміст	зріст	зріст	зріст	зріст	зріст	зріст
Адреса	921	922	923	924	925	926

Згідно з таблиці 5.11 прочитати дані з пам'яті.

5.3.5. Написати і виконати програми читання-запису при адресації через регістрові пари BC, DE:

а) відповідно до таблиці 5.12 записати дані з двох областей пам'яті;

Таблиця 5.12

Варіант	1	2	3
Адреса 1	910	912	915
Адреса 2	930	932	935
Дані	зріст	зріст	зріст
Адреса 3	960	961	962

б) перезаписати дані з однієї області пам'яті, адресуючи через BC в іншу область (DE), як в таблиці 5.12.

5.5 Зміст звіту

Звіт повинен містити:

- назву та мету роботи;
- програми, написані на мові Асемблера;
- результати виконання програм, висновки по роботі.

5.6 Контрольні запитання

5.6.1 Фізичні основи пам'яті, її технічна реалізація. Види пам'яті.

5.6.2 Засоби і види адресації.

5.6.3 Особливості і види регістрової адресації.

5.6.4 Особливості звертання до комірок пам'яті.

5.6.5 Відмінність між прямою і безпосередньою адресаціями.

5.6.6 Відмінність регістрової адресації від непрямой регістрової.

6 ДВІЙКОВА АРИФМЕТИКА МІКРОПРОЦЕСОРА. ОПЕРАЦІЇ І КОМАНДИ.

6.1 Мета роботи

Вивчення арифметичних операцій і команд мікропроцесора КР580ВМ80А.

6.2 Загальні відомості

6.2.1 Незалежно від форми подання чисел у вихідних програмах МП оперує тільки з двійковими числами, що має свою специфіку. Поняттям двійкова арифметика відображується специфіка арифметичних операцій із двійковими числами. Базовими арифметичними операціями МП є додавання і віднімання двійкових чисел, що впливає з принципів організації обчислювальних процесів.

Додавання двійкових чисел, як і десяткових, виконується за розрядами від молодшого до старшого, але значно простіше в реалізації, тому що числа мають тільки два значення. Найбільшу відповідність природному представленню додавання дає таблична форма, у якій додатки утворюють рядки і стовпці, а результат додавання цифр одноіменних розрядів знаходиться на перетинанні відповідних рядка і стовпця. Стосовно до додавання двійкових цифр зазначена таблиця має вид таблиці 6.1.

Таблиця 6.1 – Додавання двійкових чисел

Додаток	Додаток		
	+	0	1
0	0	0	1*
1	1	1	0

* має місце перенос одиниці в старший розряд.

Додавання двійкових чисел виконується за наступними правилами:

– додавання двох одиниць дає нуль у молодшому розряді результату і перенос одиниці в старший розряд ($1+1=0^*$);

- додавання одиниці до нуля молодшого розряду дасть у результаті одиницю без переносу в старший розряд ($1+0=1$);
- нулі, що складаються, результатом дають нуль ($0+0=0$).

При всій простоті двійкового додавання як процедури, громіздкість запису великих чисел у двійковій формі викликає необхідність безлічі переносів з одного розряду в інший, що створює певні незручності.

Двійкове віднімання також багато в чому подібне десятковому. Таблична форма запису в даному випадку має вид таблиці 6.2.

Таблиця 6.2 – Віднімання двійкових чисел

Зменьшувач	Від'ємне		
		0	1
	0	0	1*
	1	1	0

* означає позику одиниці із сусіднього старшого розряду.

Основні правила віднімання:

- віднімання з одиниці саме одиниці і з нуля саме нуль дає нуль ($1 - 1 = 0$, $0 - 0 = 0$);
- віднімання з одиниці нуля дає одиницю ($1 - 0 = 1$);
- віднімання з нуля одиниці вимагає позики одиниці зі старшого розряду і дає в результаті одиницю ($0 - 1 = 1$).

До числа елементарних арифметичних операцій відносять зсув двійкового числа на один розряд уліво чи управо, порівняння двох двійкових чисел, а також збільшення чи зменшення числа на 1.

До числа складних арифметичних операцій відносять такі, які можна представити у виді комбінації декількох простих. Так множення можна представити у виді комбінації додавання і зсуву, а ділення у виді послідовності віднімання і зсуву.

У мікропроцесорній системі КР580, варіантом якої є НМК, передбачена реалізація таких команд двійкової арифметики:

- а) додавання восьмирозрядних чисел;
- б) додавання шістнадцятирозрядних чисел;
- в) віднімання восьмирозрядних чисел;
- г) інкремент;

д) декремент.

Всі арифметичні операції з восьмирозрядними операндами в МП КР580 основані на уявленні про те, що один з операндів розміщується в акумуляторі (регістр А), а інший - у регістрі (РЗП) чи в пам'яті (М). Адреса комірки пам'яті (М) вказується в регістровій парі HL. Другий операнд може бути також числом, заданим безпосередньо в самій команді. Результат арифметичної операції записується в акумулятор. Операція віднімання має ту особливість, що робиться завжди із вмісту акумулятора, тобто зменшуване повинне бути операндом, розміщеним в акумуляторі. По результаті арифметичних операцій додавання і віднімання встановлюються такі типи ознак:

C – переносу, Z – нуля, S – знаку, P – парності, AC – допоміжного переносу.

Додавання шістнадцятирозрядних двійкових чисел у зазначеній системі називають подвійним додаванням. Воно засновано на тому, що один з операндів знаходиться в регістровій парі HL, а другий - або в DE, або у BC. Результат записується в HL. По результату операції встановлюється чи скидається біт переносу – C.

Операція інкремента полягає в збільшенні вмісту регістрів, регістрових пар, комірки пам'яті за адресою в HL на одиницю. Інкремент регістра і пам'яті змінює біти ознак: Z, S, P, AC. Інкремент регістрової пари не торкається бітів ознак.

Операція декремента полягає в зменшенні вмісту регістрів, регістрових пар, комірки пам'яті за адресою в HL на одиницю. Змінювані біти ознак аналогічні команді інкремента.

6.2.2 Команди додавання восьмирозрядних чисел:

- ADD R - додавання регістра: A, B, C, D, E, H, L;
- ADD M - додавання комірки пам'яті (адреса в HL);
- ADI B - додавання безпосереднього числа B;
- ADC R - додавання регістра: A, B, C, D, E, H, L плюс біт переносу C;
- ADC M – додавання комірки пам'яті (адреса в HL) плюс біт переносу C;
- ACI B – додавання безпосереднього числа B плюс біт переносу C.

Як приклад, виконайте таке завдання:

- уведіть програму, приведену в таблиці 61.3., що реалізує операцію $A = A + B + M + 1$;

Таблиця 6.3

Адреса	Команда	Код	Коментар
800	ADD B	80	$A=A+B$
801	LXI H, 900H	21 00 09	Завантаження адреси $HL=900H$
804	ADD M	86	$A=A+M$
805	ADI 01	C6 01	$A=A+1$

– виконайте програму, попередньо задавши вихідні значення відповідно до таблиці 6.4 ($M=900H$) за допомогою команди: <СТ> 800 <-> 807 <ВП>;

– перевірте отримані результати (результат операції в регістрі А та біти умов у регістрі F).

Таблиця 6.4

Регістр	Початкові дані	A	Ваш зріст
		B	Ваш зріст
		M (900H)	Ваш зріст
Результат		A	
		F регістр ознак	

Як приклад, виконайте таке завдання:

– уведіть в пам'ять програми додавання шістнадцятирозрядних чисел на основі команд восьмирозрядного додавання представлену в таблиці 6.5 для виразу: $HL=DE+BC$;

– виконайте програму, попередньо задавши вихідні значення відповідно до Вашого зросту, за допомогою такої дії: <СТ > 800 <-> 806 <ВП>;

– перевірте отримані результати в регістрах H, L та F .

Таблиця 6.5

Адреса	Команда	Код	Коментар
800	MOV A, C	79	$A \leftarrow C$
801	ADD E	83	$A + E$
802	MOV L, A	6F	$L, \leftarrow A$
803	MOV A, B	78	$A \leftarrow B$
804	ADC D	8A	$D + B$, з урахуванням переносу
805	MOV H, A	67	$H \leftarrow A$

6.2.3 Команди віднімання восьмирозрядних чисел мають вид:

- SUB R - віднімання регістра: A, B, C, D, E, H, L;
- SUB M- віднімання комірки пам'яті (адреса в HL);
- SUI B - віднімання безпосереднього числа B;
- SBB R - віднімання регістра: A,B,C,D,E,H,L мінус біт переносу C;
- SBB M - віднімання комірки пам'яті(адреса в HL), мінус біт переносу C;
- SBI B - віднімання безпосереднього числа мінус біт переносу.

Як приклад, виконайте таке завдання:

- уведіть в пам'ять програму, представлену в таблиці 6.6, що реалізує вираз: $A = A - B - M - 1$;

Таблиця 6.6

Адреса	Команда	Код	Коментар
800	SUB B	90	$A = A - B$
801	LXI H, 900H	21 00 09	Завантаження HL =900H, адреса M
804	SUB M	96	$A=A - M$
805	SBI 1	DE 01	$A=A - 1$

- виконайте програму, попередньо задавши вихідні дані відповідно до Вашого зросту командою: <СТ> 800 <-> 807 <ВП>;
 - перевірте отримані результати.
- Як приклад, виконайте таке завдання:
- уведіть в пам'ять програму, наведену в таблиці 6.7, віднімання шістнадцятирозрядних чисел: HL=DE-BC;

Таблиця 6.7

Адреса	Команда	Код	Коментар
800	MOV A, E	7B	$E \leftarrow A$
801	SUB C	91	$E - C$
802	MOV L, A	6F	$L \leftarrow A$
803	MOV A, D	7A	$A \leftarrow D$
804	SBB B	98	$D - B$ з урахуванням переносу
805	MOV H, A	67	$H \leftarrow A$

- виконайте програму, попередньо задаючи вихідні значення відповідно до Вашого зросту за допомогою команди: <СТ> 800 <-> 806 <ВП>;
- перевірте отримані результати.

6.2.4 Команди подвійного додавання мають вид:

- DAD H – додавання $HL=HL+HL$;
- DAD B – додавання $HL=HL+BC$;
- DAD D – додавання $HL=HL+DE$.

Як приклад, виконайте таке завдання:

– уведіть згідно з таблицею 6.8 програму, що реалізує операцію:
 $HL=BC+DE$;

Таблиця 6.8

Адреса	Команда	Код	Коментар
800	MOV H, B	60	$H \leftarrow B$
801	MOV L, C	69	$L \leftarrow C$
802	DAD D	19	$HL = HL + DE$

- виконайте програму, задавши вихідні значення відповідно до Вашого зросту за допомогою команди: $\langle CT \rangle 800 \leftrightarrow 803 \langle BP \rangle$;
- перевірте результати.

6.2.5 Команди інкремента мають вид:

- INR R - збільшення на одиницю вмісту регістрів: A, B, C, D, E, H, L;
- INR M - збільшення на одиницю вмісту комірки пам'яті, адреса M в HL;
- INX RP - збільшення на одиницю вмісту пари регістрів: BC, DE, SP. У даній команді вказується ідентифікатор старшого регістра (INX B).

Як приклад, виконайте таке завдання:

- запишіть в пам'ять команду INR E ($E=E+1$) за адресою 800H: 800 1C;
- виконайте зазначену команду для заданих значень вмісту регістра E згідно з таблицею 6.9 за допомогою команди: $\langle CT \rangle 800 \leftrightarrow 801 \langle BP \rangle$;

Таблиця 6.9

Регістр	Е початкове значення	Ваш зріст
	Е результат	
	Ф регістр ознак	

– перевірте отриманий результат.

Завдання:

– запишіть в пам'ять коди команд:

800 21 00 09 LXI H, 900H Завантаження. HL = 900H, адреса М

803 34 INR M M = M + 1

– виконайте зазначену послідовність для заданих в таблиці 6.10 значень вмісту комірки пам'яті М за допомогою команди: <СТ> 800 <-> 804 <ВП>;

Таблиця 6.10

Регістр	М початкове значення	Ваш зріст
	М результат	
	Ф регістр ознак	

– перевірте отримані результати.

Завдання:

– запишіть в пам'ять код команди:

800 13 INX D DE = DE + 1;

– виконайте приведену команду для заданих у таблиці 6.11 значень вмісту регістрової пари DE за допомогою команди: : <СТ> 800 <-> 801 <ВП>;

Таблиця 6.11

DE початкове значення	Ваш зріст (два байти)
DE результат	
F реєстр ознак	

– перевірте отримані результати.

Примітка: Значення реєстра ознак F залишається рівним останньому значенню попереднього завдання через те, що інкремент пари реєстрів не змінює ознак.

6.2.6 Команди декремента мають вид:

– DCR R - зменшення на одиницю вмісту реєстра: A, B, C, D, E, H, L;

– DCR M - зменшення на одиницю вмісту комірки пам'яті - адреса M у HL;

– DCX RP зменшення на одиницю вмісту пари реєстрів: BC, DE, HL, SP.

У команді вказують ідентифікатор старшого реєстра, наприклад, DCX B.

Як приклад, виконайте таке завдання:

– запишіть в пам'ять таку команду:

800 0D DCR C C=C - 1.

– виконайте зазначену команду для заданих у таблиці 6.12 значень реєстра C: <СТ> 800 <-> 801 <ВП>;

Таблиця 6.12

С початкове значення	Ваш зріст
С результат	
F реєстр ознак	

– перевірте отримані результати.

Завдання:

– запишіть в пам'ять коди команд:

800 21 00 09 LXI H, 900H Завантаження HL = 900H,

адреса M;

803 35 DCR M: M = M - 1.

– виконайте зазначені команди для початкових значень вмісту комірки пам'яті M, заданих у таблиці 6.13;

– виконайте команду за допомогою команди: <СТ> 800 <-> 804 <ВП>;

– перевірте отримані результати.

Таблиця 6.13

M початкове значення	Ваш зріст
M результат	
F реєстр ознак	

Завдання:

– запишіть в пам'ять код команди:

800 2B DCX H HL = HL - 1

– виконайте команду для значень вмісту реєстрової пари H, заданих у таблиці 6.14.

Таблиця 6.14

Н поч.	0000	1000	FFFF	0001
Н рез.	FFFF	0FFF	FFFE	0000
F	13	13	13	13

- команда запуску: <СТ> 800 <-> 801 <ВП>;
- перевірте отримані результати.

Примітка: Команда декремент регістрової пари не стосується вмісту бітів ознак.

6.3 Самостійне виконання роботи

6.3.1 Написати і виконати програму, що реалізує операцію: $C = D + E$, задавши вихідні значення відповідно до Вашого зросту.

6.3.2 Написати і виконати програму додавання вмісту двох комірок пам'яті: $M1 = M2 + M3$ з використанням адрес $M1 = 900H$, $M2 = 901H$, $M3 = 902H$.

Вміст комірок пам'яті (вихідні значення) маєте задати відповідно до Вашого зросту.

6.3.3 Написати і виконати програму додавання:

$$HL = BC + DE + 4000H.$$

Задати вихідні значення відповідно до Вашого зросту.

6.3.4 Написати програму віднімання: $C = D - E - 10H$.

6.3.5 Написати програму віднімання вмісту двох комірок пам'яті: $M1 = M2 - M3$.

Адреси комірок пам'яті $M1 = 900H$, $M2 = 901H$, $M3 = 902H$. Вміст комірок пам'яті (вихідні значення) маєте задати відповідно до Вашого зросту.

6.3.6 Написати і виконати програму віднімання вмісту регістрових пар та числа: $HL = BC - DE - 0FFFFH$ для довільних початкових даних. Задати вихідні значення відповідно до Вашого зросту.

6.3.7 Написати і виконати програму заповнення масиву по заданому індексу елемента масиву відповідно до таблиці 6.15.

Таблиця 6.15

Базова адреса масиву	900
Номер елемента	05
Адреса елемента	
Вміст	Ваш зріст

Перевірте правильність роботи програми.

6.3.8 Написати і виконати програму заповнення масиву пам'яті (900H - 904H) даними (00 - 04), використовуючи команди інкремента регістра і пари регістрів.

1.3.10 Написати і виконати програму заповнення масиву пам'яті (90FH - 90AH) даними (0FH - 0AH), використовуючи команди декремента регістра і пари регістрів.

6.4 Зміст звіту

Звіт повинен містити:

- назву та мету роботи;
- написані програми відповідно до завдань 1.3;
- результати виконання програм;
- висновки.

6.5 Контрольні запитання

6.5.1 Характеристика елементарних операцій двійкової арифметики.

6.5.2 Особливості і форми представлення операцій додавання і віднімання двійкових чисел.

6.5.3 Сутність і призначення операцій подвійного додавання, віднімання.

6.5.4 Операції зсуву, їхнє практичне значення.

6.5.5 Принципи заповнення і переносу масивів.

ПЕРЕЛІК ПОСИЛАНЬ

1. Бойко, В.І. Основи схемотехніки електронних систем підручник [Текст]:/ Бойко В.І., Гуржій А.М., Жуйков В. Я. та ін. - К.: Вища шк., 2004. – 527 с.: іл.
2. Бабич, Н.П. Компьютерная схемотехника[Текст]:/ Бабич Н.П., Жуков И.А.– К: МК-Пресс, 2004. – 576 с.
3. Алексенко, А.Г. Микросхемотехника[Текст]:/ Алексенко А.Г., Шагурин И.И.- М.: Радио и связь, 1990. - 496 с.
4. Прянишников, В.А. Электроника курс лекций[Текст]:/ В.А. Прянишников– Санкт Петербург.: КОРОНА принт, 1998. – 401 с.
5. Бойт, К. Цифроваясхемотехника[Текст]:/ Клаус Бойт. – Москва: Техносфера, 2007. – 472 с.
6. Богданович, М.Й. Цифровые интегральные микросхемы: Справочник [Текст]:/ М.Й. Богданович. – Минск: Беларусь, 1991.- 606с.
7. Гилмор, Ч. Введение в микропроцессорную технику [Текст]:/ Ч. Гилмор. – М.: Мир, 1984. – 334с.
8. Каган, Б.М. Микропроцессоры в цифровых системах[Текст]:/ Б.М. Каган, , В.В. Сташин. -М.: Энергия, 1979.
9. Преснухин, Л.Н.Микропроцессоры. В 3-х книгах. Учебник для ВТУЗов[Текст]:/ Л.Н.Преснухин, П.В.Нестеров, В.Ф.Шаньгин,В.Д.Горбунов. - М.: Высш. шк., 1986.
- 10.Морозевич, А.Н. МикроЭВМ, микропроцессоры и основы программирования [Текст]:/ А.Н. Морозевич, А.Н. Дмитриев, В.Н. Мухаметов. – М.: Высш. шк., 1990. – 352с.
- 11.Старыгин, Б.В.. Основи вычислительной, микропроцессорной техники и программирования [Текст]:/ Б.В.Старыгин, Л.С.Щарев.- М.: Высш. шк., 1989 . – 479с.
- 12.Файнштейн, В.Г.Микропроцессорные системы управления электроприводами [Текст]:/ Файнштейн В.Г., Файнштейн Э.Г. – М.: Энергоатомиздат, 1986. – 240с.
- 13.Хвощ, С.Т. Микропроцессоры и микроЭВМ в системах автоматического управления: Справочник [Текст]:/ С.Т. Хвощ. – Л.: Машиностроение, 1987. – 640с.
14. Майоров, В.Г. Практический курс программирования мик-

ропроцессорных систем [Текст]:/ В.Г. Майоров, А.И. Гаврилов. - М.: Машиностроение, 1989. – 272 с.

15.Погорелый, С.Д. Программное обеспечение микропроцессорных систем: Справочник[Текст]:/ С.Д. Погорелый, Т.Ф. Слободянюк. - К.: Технжа, 1989. - 301 с.

16.Молчанов, А.А. Справочник по микропроцессорным устройствам [Текст]:/ А.А.Молчанов, В.И.Корнейчук, В.П.Тарасенко, Д.А.Россошинский. - К.: Техніка, 1987. – 288 с.

17.Токхайм, Р. Микропроцессоры: курс и упражнения [Текст]:/ Р. Токхайм. Пер. с англ., под ред. В.Н. Грасевича. - М.: Энергоатомиздат, - 1987. - 336 с.

18.Методичні вказівки для самостійної роботи студентами всіх форм навчання при вивченні дисципліни «Основи мікропроцесорної техніки» для підготовки бакалаврів напряму 050702 «Електромеханіка» спеціальностей 8.05070201 «Електричні машини та апарати» зі спеціалізацією «Електричні апарати» та 8.05070207 «Електромеханічне обладнання енергоємних виробництв» / Укл.: Л.Б. Жорняк, В.І. Осинська. – Запоріжжя: ЗНТУ, 2016. – 46 с.