

О.Є. Рубаненко, К.І. Кравцов, В.О. Комар

## **МІКРОПРОЦЕСОРНА ТЕХНІКА**

Міністерство освіти і науки України  
Вінницький національний технічний університет

О.С. Рубаненко, К.І. Кравцов, В.О. Комар

### **Мікропроцесорна техніка**

Затверджено Вченою радою Вінницького національного технічного університету як лабораторний практикум для студентів напряму підготовки - "Електричні станції" та "Електричні системи і мережі".  
Протокол № 4 від 25 листопада 2004 р.

УДК 621.316  
Р 49

*Рецензенти:*

*В.М. Кутін*, доктор технічних наук, професор  
*Ю.В. Демет'єв*, кандидат технічних наук, професор  
*В.І. Нагул*, кандидат технічних наук, доцент

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України

**Рубаненко О.Є., Кравцов К.І., Комар В.О.**  
Р 49 **Мікропроцесорна техніка.** Лабораторний практикум. - Вінниця: ВНТУ, 2005. - 83 с.

Лабораторний практикум "Мікропроцесорна техніка" є основою практичної підготовки зі спеціальностей - "Електричні станції" та "Електричні системи і мережі" факультету електроенергетики з дисципліни "Мікропроцесорна техніка". В лабораторному практикумі розглянуто приклади складання програм на машинній мові для мікропроцесора КР580ВМ80А та на мові Ассемблер, приклади використання сучасних персональних комп'ютерів для керування силовим енергетичним обладнанням та його діагностики.

УДК 621.316

## ЗМІСТ

Лабораторна робота №1 .....	4
Лабораторна робота №2 .....	9
Лабораторна робота №3 .....	15
Лабораторна робота №4 .....	23
Лабораторна робота №5 .....	35
Лабораторна робота №6 .....	41
Лабораторна робота №7 .....	47
Лабораторна робота №8 .....	68
Лабораторна робота №9 .....	73
Лабораторна робота №10 .....	78
Лабораторна робота №11 .....	80
Література .....	83

# ЛАБОРАТОРНА РОБОТА № 1

## ОЗНАЙОМЛЕННЯ З РОБОТОЮ НА НАВЧАЛЬНІЙ МІКРО-ЕОМ

**Мета роботи:** ознайомлення зі структурою навчальної мікро-ЕОМ (електронною обчислювальною машиною), картою пам'яті, органами керування та режимами роботи.

### Короткі відомості з теорії

Після вклучення навчальної мікро-ЕОМ або натискання клавіші початкового встановлення "Сброс" зміст програмного лічильника великої інтегральної схеми мікропроцесора (ВІС МП) буде дорівнювати нулю і виконання програми починається з нульової комірки пам'яті, де записані програми тестів вузлів мікро-ЕОМ [1,2].

Після виконання тестів відбувається початкове встановлення всіх внутрішніх програмно-доступних регістрів МП ВІС. Початкове встановлення полягає в тому, що в програмний лічильник записується перша адреса оперативного запам'ятовуючого пристрою (ОЗП), у покажчик стека - певне число, а решта регістрів МП ВІС обнулюється.

### Завдання для домашньої підготовки

1. Ознайомтесь з описом навчальної мікро-ЕОМ.

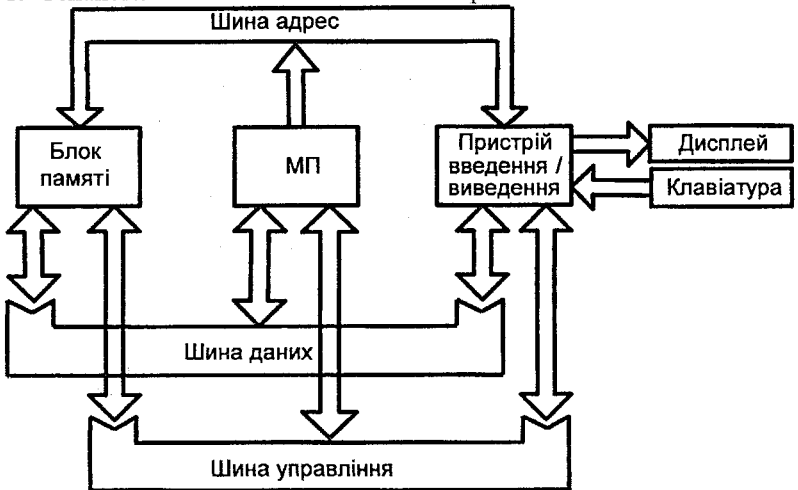


Рисунок 1.1 – Схема мікропроцесорної системи

2. Ознайомтесь з типовою мінімальною структурою мікро-ЕОМ, методами організації магістралей, під'єднання пам'яті і зовнішніх пристроїв до магістралей.

3. Вивчіть алгоритм роботи керуючої програми і можливі режими роботи за рис. 1.1.

4. Розгляньте роботу МП ВІС в режимі очікування і стану магістралей мікро-ЕОМ при його виконанні.

5. Вивчіть внутрішні регістри МП ВІС КР580ВМ80А і часові діаграми виконання команд.

### **Завдання до лабораторної роботи**

**Завдання 1.** Дослідити порядок включення мікро-ЕОМ.

#### **Порядок виконання завдання**

1. Під'єднати шнур живлення до мережі.

2. Ввімкнути тумблер. В результаті виконання тестових програм світлодіоди вихідного регістра та сегменти дисплея будуть включені на час, що дорівнює  $\sim 2$  с. [3], через певний час мікро-ЕОМ буде готова реагувати на команди. Перебуваючи в цьому режимі, мікро-ЕОМ реагує на натискання клавіші керування. Виклик можливих режимів роботи з цього стану мікро-ЕОМ визначається алгоритмом, наведеним на рис. 1.2. З будь-якого місця керуючої програми можна повернути мікро-ЕОМ в початковий стан натисканням на клавішу "Сброс". В цьому випадку мікро-ЕОМ починає виконувати керуючу програму з нульової комірки пам'яті. В результаті її виконання всі комірки ОЗП будуть занулені і, отже, буде стерта вся програма користувача, записана раніше в ОЗП.

3. Натиснути на клавішу "Сброс" і впевнитись в тому, що тестові програми проходять заново.

**Завдання 2.** Дослідити зміст пам'яті.

#### **Порядок виконання завдання**

1. Натиснути на клавішу "Сброс". При цьому на дисплеї в усіх розрядах будуть нулі, а 8 світлодіодів будуть світитися.

2. Послідовно натиснути на клавіші 8, 0, 0, 0. Переконатися при цьому, що кожна цифра буде записана в молодший розряд адресного дисплея і відбудеться одночасний зсув всіх знаків на адресному дисплеї на один розряд вліво [2]. При введенні четвертої цифри з клавіатури та натисканні клавіші "Уст.ад." мікро-ЕОМ виведе на дисплей число, що записане за цією адресою. На семисегментних індикаторах дисплея з'явиться певне число. Запишіть це число у звіт. У випадку помилки при введенні коду адреси слід натиснути на клавішу "Сброс" і повторити введення.

3. Натиснути на клавішу "Ад+". В цьому режимі мікро-ЕОМ збільшить на одиницю адресу комірки пам'яті на адресному дисплеї і виведе вміст комірки пам'яті на дисплей. Запишіть це число. Послідовно натискаючи на клавішу "Ад+", перевірити і записати вміст інших п'яти комірок пам'яті і їхні адреси в ОЗП.

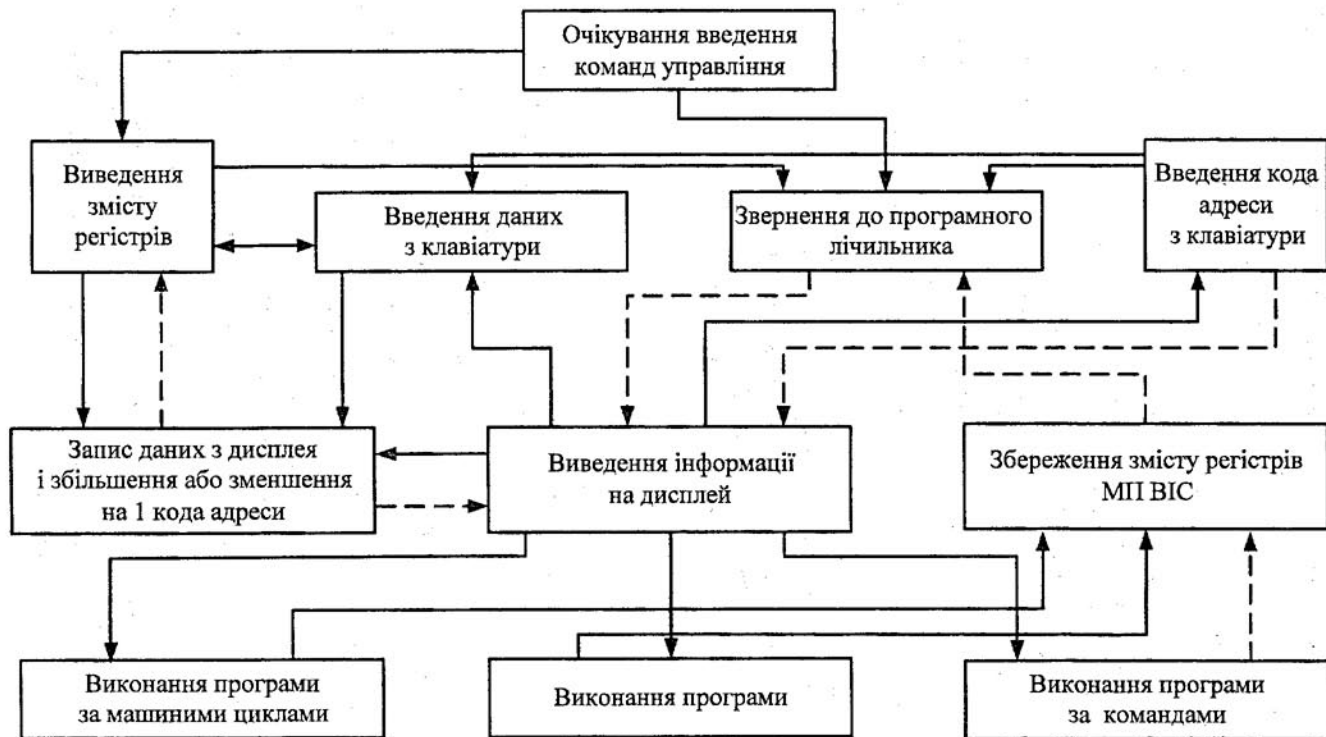


Рисунок 1.2 - Схема переходу до різних режимів роботи в мікро-ЕОМ

4. Натиснути на клавішу "Сброс" і ввести код 0000 (перша адреса ПЗП). На дисплеї даних з'явиться певний код. Послідовно натискаючи на клавішу "Ад+" продивитись вміст декількох комірок ПЗП. Таким чином, використання цього режиму дозволяє перевірити вміст всіх комірок пам'яті мікро-ЕОМ.

**Завдання 3.** Записати числа в пам'ять мікро-ЕОМ.

**Порядок виконання завдання**

1. Ввімкнути тумблер МЕРЕЖА (або при ввімкненій раніше мікро-ЕОМ натиснути на клавішу "Сброс").

2. Набрати адресу 8000 і натиснути на клавішу "Уст.ад." на дисплеї даних висвітиться вміст комірки за адресою 8000.

3. Натиснути на клавішу 1. Мікро-ЕОМ вводить значення цифри натиснутої клавіші в молодший розряд дисплея.

4. Натиснути на клавішу 2. Впевнитись при цьому, що цифра 1 молодшого розряду дисплея перемістилася на старший розряд дисплея, а цифра 2 - на її місце. Перевірити, що наступні натискання на цифрові клавіші призводять до введення цифри натиснутої клавіші в молодший розряд дисплея і переміщення в старший розряд попередньої натиснутої клавіші.

5. Записати натисканням на клавішу "Запись" число, представлене на дисплеї в комірку пам'яті за адресою 8000. При цьому на дисплеї з'явиться ця адреса і вміст комірки пам'яті 8001.

6. Записати будь-яке число на дисплеї.

7. Натиснути на клавішу "Ад-". На адресному дисплеї з'явиться адреса 8000 з її вмістом. Впевнитись при цьому, що число, яке було записано за цією адресою те саме.

8. Натиснути на клавішу "Ад+" і перевірити вміст адреси 8001. Впевнитись в тому, що число, яке висвітилось в дисплеї даних не було записане в пам'ять при натисканні на клавішу "Ад-".

9. Здійснити виведення на дисплей вмісту адреси 0000 (ПЗП). На дисплеї буде певне число. Записати за цією адресою на дисплеї будь-яке інше число. Впевнитись, що при натисканні на клавішу "Запись" запис даних в ПЗП неможливий.

**Завдання 4.** Записати числа в програмно-доступні регістри МП ВІС.

**Порядок виконання завдання**

1. Ввімкнути мікро-ЕОМ. Після цього натиснути на клавішу "Сброс". Введіть адресу 83ЕВ. Натисніть клавішу "Уст. адр.". На дисплей засвітиться вміст регістра акумулятора (А) МП ВІС. Багатократно натискаючи на клавішу "Ад-", перевірити послідовність виведення вмісту внутрішніх регістрів МП ВІС (регістра ознак, регістрів В, С, D, Е, Н, L) на дисплей.

2. Змінити число, записане в реєстрі за адресою 8000 при наявності на дисплеї інформації про його вміст, за допомогою цифрових клавіш клавіатури. Натиснути на клавішу "Запись".

3. Ввести адресу 8000 і натиснути на клавішу "Уст.адр.". Впевнитись при цьому в правильності запису числа в реєстр.

**Завдання 5. Здійснити пуск програми.**

**Порядок виконання завдання**

1. Для здійснення пуску програми з будь-якої адреси пам'яті вивести цю адресу на дисплей (0300, "Уст.адр."). Пуск програми відбувається натисканням на клавішу "Пуск", при цьому в програмний лічильник МП ВІС записується адреса, вказана на дисплеї, і виконання програми починається з цієї адреси. Записати на дисплеї адресу 0300, яка є початком музичної програми, що записана в ПЗП.

2. Натиснути на клавішу "Пуск", при цьому мікро-ЕОМ виконає мелодію по нотах, коди яких записані в ПЗП.

3. Переконайтесь, що виконання програми може бути зупинено натисканням на клавішу "Сброс". Для цього повторити пуск музичної програми заново і при виконанні програми натиснути на клавішу "Сброс".

**Зміст звіту**

Звіт повинен містити:

1. Схему структури навчальної мікро-ЕОМ.

2. Карту пам'яті.

3. Інформацію про вміст внутрішніх програмно-доступних реєстрів МП ВІС після програми початкового встановлення мікро-ЕОМ.

**Завдання для самоперевірки**

1. Нарисуйте структуру навчальної мікро-ЕОМ.

2. Вкажіть функціональні частини на електричній схемі навчальної мікро-ЕОМ.

3. З якою метою в мікро-ЕОМ використовується робота МП ВІС в режимі очікування?

4. Що таке карта пам'яті мікро-ЕОМ?

5. Які адреси пам'яті мікро-ЕОМ відносяться до ОЗП і ПЗП?

6. Розкажіть про можливі режими роботи мікро-ЕОМ.

7. Як записати числа в програмно-доступні реєстри МП ВІС або пам'ять мікро-ЕОМ?

8. Що відбувається в мікро-ЕОМ при спробі запису даних в ПЗП?

9. Які вузли перевіряються в мікро-ЕОМ в процесі виконання тестів при її ввімкненні або натисканні на клавішу "Сброс"?

10. Вкажіть можливі зміни режимів роботи мікро-ЕОМ, які здійснюються при виконанні програми за командами.

11. Опишіть функції кожної клавіші керування на клавіатурі.
12. Вкажіть, які значення записуються в регістри МП ВІС і ОЗП в процесі програми початкового завантаження мікро-ЕОМ.

## ЛАБОРАТОРНА РОБОТА № 2 ЗАПИС ТА ВИКОНАННЯ ПРОСТИХ ПРОГРАМ

**Мета роботи:** дослідження виконання окремих команд і простих програм; використання різних методів адресації в програмах; запис програм.

### Короткі відомості з теорії

Мікропроцесор МП ВІС КР580ВМ80А має фіксований набір команд. Час виконання команди визначається процесом отримання, декодування і виконання команди. Цей час можна показати таким, що складається з низки часових інтервалів. Найбільш короткий часовий інтервал, що дорівнює періоду синхросигналів МП ВІС, називається **машинним тактом**. Час, необхідний для добування 1 байту інформації з пам'яті або зовнішнього пристрою або виконання команди, яка визначається одним машинним словом, називається **машинним циклом**. Машинний цикл для МП ВІС може включати в себе 3-5 машинних тактів. В залежності від виду команди час виконання може складатись з 1-5 машинних циклів. Для МП ВІС є 10 різних типів машинних циклів:

- добування коду команди (цикл  $M_1$ ),
- читання даних з пам'яті,
- запис даних в пам'ять,
- добування даних зі стека,
- запис даних в стек,
- введення даних з зовнішнього пристрою,
- запис даних в зовнішній пристрій,
- цикл обслуговування переривання,
- зупинення,
- обслуговування переривання в режимі зупинення.

Першим машинним циклом при добуванні будь-якої команди є цикл  $M_1$ .

На кожному машинному такті МП ВІС перевіряє стан сигналу "Готовий" на своєму вході. Нульовий сигнал при цьому вході тимчасово зупиняє нормальну роботу МП ВІС, при цьому на магістралях мікро-ЕОМ присутня вся інформація, що передається на розглядуваному машинному циклі. В навчальній мікро-ЕОМ це використовується для дослідження виконання команд за машинними тактами. В цьому режимі інформація на магістралях мікро-ЕОМ відображається світлодіодами стану.

Програма записується в мікро-ЕОМ в послідовних комітках пам'яті.

Розглянемо найпростішу програму (програма 2.1), яка добуває число з адреси пам'яті 8020, інвертує його і записує результат в комірку пам'яті за адресою 8021 [10,11].

Таблиця 2.1 - Текст програми 2.1 (в мнемосодах)

Мнемосокод	Коментар
LDA 8020	Отримати число з адреси 8020
CMA	інвертувати число
STA 8021	Записати результат за адресою 8021
RST1	перервати виконання програми

При записі програми всі числа подаються в шістнадцятковій системі числення.

Для запису програми в пам'ять мікро-ЕОМ необхідно перевести мнемосоди команд в машинні коди. Команди в програмі можуть бути одно-, дво- або трибайтові і повинні в пам'яті займати відповідно одну, дві або три адреси.

Таблиця 2.2 - Текст програми 2.1 (розміщення за адресами пам'яті)

Адреса	Число	Коментар
8000	3A	код команди LDA
8001	20	молодший байт адреси
8002	80	старший байт адреси
8003	2F	код команди CMA
8004	32	код команди STA
8005	21	молодший байт адреси
8006	80	старший байт адреси
8007	CF	код команди RST1

Попередній запис програми зручно проводити в більш компактній формі. В програмі вказується початкова адреса кожної команди і при цьому розуміється, що в залежності від довжини (одно-, дво- або трибайтова) команди в пам'яті будуть займати від однієї до трьох послідовних комірок. При такому записі в лівому стовпці вказується лише адреса команд в програмі. Це дозволяє скоротити об'єм при описі програм і зробити більш простим їх аналіз.

Таблиця 2.3 - Текст програми 2.1 (загальний вигляд запису)

Адреса	Машинний код	Мітка	Мнемосокод	Коментар
1	2	3	4	5
8000	3A 8020		LDA,	отримати число
8003	2F		CMA	інвертувати число

Продовження таблиці 2.3

1	2	3	4	5
8004	32 8021		STA, 8021	записати за адресою 8021
8007	CF		RST1	перервати виконання програми

Тут використовується прямий спосіб адресації.

Розглянемо програму, аналогічну програмі 2.1, з використанням непрямого способу адресації (програма 2.2).

Таблиця 2.4 - Текст програми 2.2

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	21 8020		LXI H, 8020	записати в регістри H,L число 8020
8003	7E		MOV A, M	отримати число з адреси, вказаної в регістрі H,L
8004	2F		CMA	інвертувати число в акумуляторі
8005	23		INX M	збільшити та 1 число в регістрах H,L
8006	77		MOV M,A	записати число з акумулятора за адресою, вказаною в H,L
8007	CF		RST 1	перервати виконання програми

**Завдання для домашньої підготовки**

1. Ознайомтесь з мовою програмування і структурою команд МП ВІС КР580ВМ80А.

2. Вивчіть режими роботи і часові діаграми процесу виконання команд МП ВІС КР580ВМ80А.

3. Вивчіть методи програмування на мові асемблера і в машинних кодах для МП ВІС КР580ВМ80А.

4. Розгляньте правила виконання команд INR A (3C), DCR A (3D), ADD A (87), ANA A (A7), ORA A (B7), CMP A (BF), DAA (27).

5. Розгляньте результат виконання програми 2.1 при записі за адресою 8003 команд, наведених в п. 4 завдання. Результати виконання програми при різних командах, записаних в програмі 2.1 за адресою 8003, занести в табл. 2.5.

6. Змініть вигляд і запишіть програму 2.2 так, щоб при її виконанні дослідне число першопочатково записувалось за адресою 8021.

7. Розробіть програми:

- а) збільшення на 5 числа, записаного за адресою 8020, і запис результату за адресою 8021 (програма 2.3);
- б) додавання чисел, записаних за адресами 8020 і 8021, і запис результату за адресою 8022 (програма 2.4);
- в) порівняння чисел в адресах 8020 і 8021 і запис більшого з них в регістр В (програма 2.5).

Таблиця 2.5 - Результати виконання програми при різних командах

Число, записане за адресою 8020	Команда, записана за адресою 8003	Число записане за адресою 8021

### **Завдання до лабораторної роботи**

#### **Завдання 1. Дослідити програму 2.1.**

##### **Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.1.
2. Записати за адресою 8020 дослідне число.
3. Здійснити пуск програми 2.1 з адреси 8000. Перевірити результат виконання програми шляхом дослідження числа, записаного за адресою 8021.
4. Дослідити процес виконання програми за командами. Після виконання кожної команди проаналізувати зміст всіх програмно-доступних регістрів МП ВІС.
5. Дослідити процес виконання команд в програмі 2.1 за машинними циклами. Звернути увагу на послідовність передачі і перетворення інформації в мікро-ЕОМ при виконанні кожної команди. Показати часові діаграми процесу виконання будь-якої одно- та трибайтової команди в програмі.
6. Замінюючи в програмі 2.1 команду СМА на команди INR A (3C), DCR A (3D), ADD A (87), ANA A (A7), ORA A (B7), CMP A (BF), DAA (27), дослідити результати виконання вказаних команд за числом, записаним за адресою 8021. Перевірити табл. 2.4, заповнену при виконанні домашнього завдання.

#### **Завдання 2. Дослідити програму 2.2.**

##### **Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.2.
2. Записати за адресою 8020 дослідне число.
3. Здійснити пуск програми з адреси 8000. Перевірити результат виконання програми за числом, записаним за адресою 8021.
4. Дослідити процес виконання команди MOV A, M за машинними циклами.

5. Ввести змінену програму 2.2, яка дозволяє з початку записувати дослідне число за адресою 8021, в мікро-ЕОМ і дослідити її виконання.

**Завдання 3.** Дослідити програму 2.3.

**Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.3, розроблену при домашній підготовці.

2. Здійснити пуск програми 2.3 і перевірити результат її виконання за числом, записаним за адресою 8021 (при числах 05, FE, записаних за адресою 8020).

**Завдання 4.** Дослідити програму 2.4.

**Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.4, розроблену при домашній підготовці.

2. Перевірити результат виконання програми за числом, записаним за адресою 8022, послідовно записуючи за адресами 8020 та 8021 відповідно числа 0B та B0, FE та B5 і здійснюючи пуск програми. Змінити вигляд і дослідити програму 2.4 для випадку, коли сума двох чисел буде перевищувати восьмирозрядне число.

**Завдання 5.** Дослідити програму 2.5.

**Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.5, розроблену при домашній підготовці.

2. Записати за адресами 8020 та 8021 дослідні числа.

3. Здійснити пуск програми 2.5. Перевірити результат її виконання за числом, записаним в регістрі В.

**Завдання 6.** Дослідити програму 2.6.

**Порядок виконання завдання**

1. Ввести в мікро-ЕОМ програму 2.6.

2. Здійснити пуск програми 2.6. Перевірити результат її виконання за числом, записаним в регістрі 8021.

3. Дослідити результат виконання програми.

Таблиця 2.6 - Текст програми 2.6 (загальний вигляд запису)

Адреса	Машинний код	Мітка	Мнемокод	Коментар
1	2	3	4	5
8000	3E 01		MVI A, 01H	записати число 01 в акумулятор
8001	2F		CMA	інвертувати число 01
8002	32		STA, 8021	записати результат за

Продовження таблиці 2.6

1	2	3	4	5
8003	21			адресою 8021
8004	80			
8005	CF		RST1	перервати виконання програми
8006				

### Робота на навчальній ЕОМ

При виконанні лабораторної роботи на навчальній мікро-ЕОМ слід мати на увазі, що:

1. Введення досліджуваної програми здійснюється шляхом її послідовного запису в комірки пам'яті. Для запису числа за адресою використовують клавіші "Уст.адр.", "Запись" і цифрові клавіші на клавіатурі.

2. Пуск програми здійснюється натисканням на клавішу "Пуск".

3. Виконання програми за командами здійснюється натисканням на клавішу "Шаг". Після виконання кожної команди мікро-ЕОМ дозволяє виводити на дисплей вміст програмно-доступних регістрів МП ВІС за допомогою клавіш "Уст.адр." і "Ад+".

#### Зміст звіту

Звіт повинен містити:

1. Заповнену табл. 2.5 для різних варіантів виконання програми 2.1 при використанні команд, представлених в п. 4 завдання для домашньої підготовки.

2. Часові діаграми виконання будь-якої довільно вибраної одно- та трибайтової команди в програмі 2.1.

3. Часову діаграму виконання мікро-ЕОМ команди MOV A, M в програмі 2.2.

4. Вигляд зміненої програми 2.2, записаної в машинних кодах, яка дозволяє першочергово записувати дослідне число за адресою 8021.

5. Розроблені в процесі домашньої підготовки програми 2.3, 2.4 і 2.5, що записані в машинних кодах; результати досліджень роботи програми за п. 3, 4, 5.

#### Завдання для самоперевірки

1. За скільки машинних тактів виконується кожна команда в програмі 2.1 та 2.2?

2. Вкажіть розбіжності в способах адресації, використаних в мікро-ЕОМ, побудованих на основі мікропроцесорного комплекту (МПК) серії K580.

3. При виконанні яких команд, приведених в п. 4 завдання для домашньої підготовки, використовуються розряди регістра стану МП ВІС?

4. Сформулюйте правила виконання МП ВІС команд, приведених в п. 4 завдання для домашньої підготовки.

5. Нарисуйте часові діаграми процесу виконання мікро-ЕОМ таких команд: LDA<A<sub>2</sub>> <A<sub>1</sub>>, СМА, STA<A<sub>2</sub>> <A<sub>1</sub>>, MOV M, A, MOV A, M, INX M.

### ЛАБОРАТОРНА РОБОТА № 3 ВВЕДЕННЯ-ВИВЕДЕННЯ, МАСКУВАННЯ ДАНИХ ТА ОРГАНІЗАЦІЯ УМОВНИХ ПЕРЕХОДІВ

**Мета роботи:** дослідження методів під'єднання та організації обміну інформацією з найпростішими пристроями введення-виведення. Вивчення програмних засобів маскування даних та організації умовних переходів в мікро-ЕОМ.

#### Короткі відомості з теорії

До команд введення-виведення МП ВІС КР580ВМ80А відносяться команди IN<A<sub>1</sub>> та OUT<A<sub>1</sub>>. При виконанні команди IN<A<sub>1</sub>> мікро-ЕОМ зчитує число з вхідного пристрою за адресою (A<sub>1</sub>) (A<sub>1</sub>) і записує його в акумулятор. При виконанні команди OUT<A<sub>2</sub>> МП ВІС записує число з акумулятора у вихідний пристрій за адресою (A<sub>1</sub>) (A<sub>1</sub>). Оскільки адреса пристрою вказується в одному байті, то за допомогою цих команд мікро-ЕОМ може обмінюватися інформацією не більше ніж з 256 зовнішніми пристроями [3,4].

Як найпростіший пристрій введення-виведення можуть використовуватись 8-розрядні регістри (наприклад, багаторежимний буферний регістр (ББР) К589ИР12). Обмін даними між мікро-ЕОМ і зовнішніми пристроями може викликатися як у певних місцях в програмі, так і за сигналами переривання. В останньому випадку підпрограма обміну даними із зовнішнім пристроєм буде викликатися за рахунок переведення мікро-ЕОМ в режим обслуговування переривання [1,2].

Викликана підпрограма обслуговування переривання змушує мікро-ЕОМ звертатися до вхідного пристрою для отримання даних. При надходженні сигналу запиту від зовнішнього пристрою (ЗП) на вхід STB (вхід С) багаторежимний буферний регістр виробляє сигнал низького рівня на виході  $\overline{INT}$ , який може подаватися на вхід переривання МП ВІС [4]. Підпрограма обслуговування цього переривання дозволяє записати дані в ББР за сигналами вибірки. Цими ж сигналами скидається внутрішній тригер запиту переривання ББР, що призводить до появи сигналу одиничного рівня на виході  $\overline{INT}$ , який сповіщає зовнішній пристрій про прийом даних від мікро-ЕОМ в ББР.

Як пристрій введення-виведення можуть застосовуватись і більш складні схеми, наприклад програмований пристрій введення-виведення

інформації в паралельному коді (KP580BB55A). При замкнутому перемикачеві на вхід регістра подається "0", а при розімкненому – "1". Перемикачі використовуються для імітації передачі даних від зовнішнього пристрою. До регістра можна під'єднати світлодіоди ( $HL_1 - HL_8$ ) для індикації чисел, записаних в ньому. Світлодіоди  $HL_1 - HL_8$  вказують число, записане у вихідному пристрої.

Найпростіша програма (програма 3.1) перезапису числа з вхідного пристрою (за адресою FA) у вихідний пристрій (за адресою F9) має такий вигляд [10,11]:

Таблиця 3.1 - Текст програми 3.1

Адреса	Машинний код	Команда	Коментар
8000	3E	MVI	Запис в акумулятор даних для програмування інтерфейсу. Програмування інтерфейсу ВІС KP580BB55 ( $251_{10}=FBH$ )
8001	81	A,81H	
8002	D3	OUT 251	
8003	FB		
8004	DB	IN 250	Записати число з комірки вхідного пристрою, яка має адресу $250_{10}=FAH$ в акумулятор (ця комірка з'єднана з зовнішніми перемикачами)
8005	FA		
8006	D3	OUT 249	Записати число з акумулятора у пристрій виведення на світлодіодні індикатори за адресою $249_{10}=F9H$
8007	F9		
8008	C3	JMP	Зацикловання програми
8009	04		
8010	80		

Організація умовних переходів в мікро-ЕОМ здійснюється за допомогою команд керування переходами, які керуються станом відповідних прапорців та регістра ознак.

### Прапорці

В МП KP580BM80A є п'ять умовних прапорців, пов'язаних з виконанням команд: нуля – Z, знаку – S, парності – P, перенесення – CY, допоміжного перенесення – AC. Кожний прапорець це однобітовий регістр. Прапорець встановлений, якщо біт дорівнює логічній 1, прапорець скинутий, якщо біт дорівнює логічному 0.

Прапорець нуля (Z) – встановлюється, якщо результат операції дорівнює 0 та скидається, якщо результат не дорівнює 0.

Прапорець знаку (S) – встановлюється, якщо старший біт результату операції дорівнює 1, а в протилежному разі прапорець скидається.

Прапорець парності (P) – якщо після виконання операції байт результату містить парну кількість одиниць, то прапорець встановлюється

в 1. Якщо результат містить непарну кількість одиниць, прапорець скидається.

Прапорець перенесення (CY) – якщо при виконанні операції відбувається перенесення (при додаванні) або позичання у старшого біта (при відніманні і порівнянні), то прапорець встановлюється, в іншому випадку - прапорець скидається.

Допоміжний перенос (AC) – якщо при виконанні операції відбувається перенос з 3-го в 4-ий біт результату, то прапорець встановлюється. Якщо операція не виконується, то прапорець скидається. Цей прапорець встановлюється при додаванні, відніманні, зменшенні, збільшенні, порівнянні і логічних операціях, але звичайно він використовується при додаванні і порівнянні, що передують команді DAA (десятькове доповнення акумулятора).

Регістр ознак має п'ять розрядів, кожний з яких встановлюється за певним правилом у відповідності з виконанням МП ВІС останньої команди. Цими розрядами є:

1. Розряд переповнення C-CARRY. В нього записується 1, якщо при виконанні арифметичної команди або команди зсуву було переповнення акумулятора, в протилежному випадку в розряд записується 0.

2. Розряд знака S-SING. В нього записується 1, якщо при виконанні арифметичної або логічної команди в старшому, сьомому, розряді акумулятора записана 1, в протилежному випадку записується 0.

3. Розряд нульового результату Z-ZERO. В нього записується 1, якщо при виконанні арифметичної або логічної команди у всіх розрядах числа в акумуляторі є 0, в протилежному випадку в розряд записується 0.

4. Додатковий розряд переповнення AC-AUX CARRY. В нього записується 1, якщо при виконанні команд в акумуляторі з'являється одиниця перенесення з третього розряду числа.

5. Розряд парності P-PARITY. В нього записується 1, якщо при виконанні команди кількість одиниць в розрядах акумулятора буде парною.

В багатьох випадках при виконанні програм необхідно перевірити або змінювати (маскувати) стан одного або декількох розрядів числа в акумуляторі. Це можна здійснити за допомогою таких операцій [9]:

1) логічне множення числа в акумуляторі і масці, яке очищує розряд числа, якщо у відповідному розряді маски буде записаний 0, і не змінює його, якщо в розряді маски буде записана 1;

2) логічне додавання чисел в акумуляторі і масці, яке встановлює розряд числа в 1, якщо в такому ж розряді маски буде записана 1, і не змінює його, якщо в цьому розряді записаний 0;

3) логічного "виключне АБО" числа в акумуляторі і масці, яке інвертує вміст розряду числа, якщо у відповідному розряді маски записана 1, і не змінює його, якщо в цьому розряді записаний 0.

Приклади використання цих команд наведені в табл. 3.2.

Таблиця 3.2 - Приклади використання логічних команд

Мнемокод	Машинний код	Число в акумуляторі	Маска	Коментар	Результат в акумуляторі
ANI<D <sub>1</sub> >	E6<D <sub>1</sub> >	00111010 11111111 00000000 10101010 11110000 00001111 00100010	10101100 00100010 00100010 00100010 11111111 11111111 00000000	Логічне множення вмісту акумулятора з байтом D <sub>1</sub>	00101000 00100010 00000000 00100010 11110000 00001111 00000000
ORI<D <sub>1</sub> >	F6<D <sub>1</sub> >	00111010 00001111 11110000	10101100 00001111 00001111	Логічне додавання вмісту акумулятора з байтом D <sub>1</sub>	10111110 00001111 11111111
XRI<D <sub>1</sub> >	EF<D <sub>1</sub> >	00111010 00001111 11110000	10101100 00001111 00001111	Логічне "виключне АБО" вмісту акумулятора з байтом D <sub>1</sub>	10010110 00000000 11111111

Проведення логічних операцій можливе також з вмістом акумулятора і внутрішніми регістрами МП ВІС. В цьому випадку команди – однобайтові. При виконанні всіх логічних команд задіяні розряди Z, S, P, AC регістра ознак (в розряд C записується 0). Це дозволяє перевіряти стан будь-якого розряду числа і виконувати умовні переходи в програмах. Програма маскування окремих розрядів числа (програма 3.2), записаного у вхідному пристрої, наведена нижче. Програма розміщує результат маскування у вихідному пристрої.

Умовні переходи організують в програмах за допомогою команд умовних переходів. При виконанні цих команд ВІС МП перевіряє стан відповідного розряду регістра стану. Якщо при перевірці стану розряду цього регістра умова не підтверджується, то виконується наступна за порядком команда програми. Всі команди умовних переходів – трибайтові: перший байт містить код команди, другий і третій байти – адресу передачі керування. Таким чином, команди умовних переходів дозволяють будувати розгалужені алгоритми і в залежності від поточного значення результату виконання програми переходить на різні ділянки програми.

Таблиця 3.3 - Текст програми 3.2

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	3E		MVIA,	Програмування інтерфейсу KP580BB55
8001	81		81H	
8002	D3		OUT 251	
8003	FB			
8004	DB	CNT	IN 250	Отримання числа з вхідного порту в акумулятор (з вхідних перемикачів)
8005	FA		ANI FA	Маска FA (250 <sub>10</sub> ) =FAH
8006	E6			
8007	FA		OUT 249	Записування результату з акумулятора у порт виведення на світлодіоди
8008	D3			
8009	F9			
800A	C3		JMP CNT	Зацикловання програми
800B	05			
800C	80			

Таблиця 3.4 - Текст програми 3.3

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	3E		MVI A,81	Програмування інтерфейсу K580BB55
8001	81			
8002	D3		OUT 251	
8003	FB			
8004	DB	WAIT	IN 250	Читання даних з вхідного порту в акумулятор
8005	FA			
8006	E6		ANI FA	Перевірка стану п'ятого розряду числа в акумуляторі
8007	FA			
8008	CA		JZ WAIT	Перехід на мітку WAIT (адреса 8004H), якщо число яке надійшло з вхідних перемикачів співпало з FAH (індикатор z = 0)
8009	04			
800A	80			
800B	CF		RST 1	закінчити виконання програми

Вище наведена програма (програма 3.3) для визначення 1 в п'ятому розряді числа, записаного у вхідному пристрої. Програма використовує маскування числа і умовний перехід.

#### Завдання до домашньої підготовки

1. Ознайомтеся зі схемою багаторежимного буферного регістра та схемою програмного пристрою введення-виведення інформації в паралельному коді KP580BB55.

2. Вивчіть способи організації обміну інформацією між мікро-ЕОМ і зовнішніми пристроями. Розгляньте схеми під'єднання пристрою введення-виведення даних при різних способах обміну.

3. Ознайомтеся з командами введення-виведення ВІС МП КР580ВМ80А.

4. Зобразіть схему під'єднання до мікро-ЕОМ пристрою введення-виведення, який використовується при проведенні лабораторної роботи.

5. Вивчіть групу логічних команд і команд умовної передачі керування.

6. Ознайомтеся з розрядами регістра ознак ВІС МП та правилами запису в них 1.

7. Ознайомтеся з програмами 3.1, 3.2, 3.3.

8. Самостійно розробіть програми: а) увімкнення світлодіодів вихідного пристрою, якщо число, що записане у вхідному пристрої, більше 3; б) увімкнення світлодіодів вихідного пристрою, якщо число, записане у вхідному пристрої, більше 3, але менше 8.

9. Змініть вигляд програми 3.3 так, щоб мікро-ЕОМ реагувала на 0 в п'ятому розряді при записаних 1 у всіх інших розрядах.

### **Завдання до лабораторної роботи**

**Завдання 1.** Дослідити програму 3.1.

#### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 3.1. Здійснити пуск програми.

2. Впевнитись, що при виконанні програми мікро-ЕОМ постійно переписує дані з вхідного пристрою у вихідний. Для цього за допомогою перемикачів вхідного пристрою змінити числа, записані в ньому. Інформація про числа в пристроях введення-виведення може відобразитися світлодіодами.

**Завдання 2.** Дослідити програму 3.2.

#### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 3.2. Здійснити пуск програми і дослідити результат її виконання за числом, записаним у вхідний пристрій.

2. Замінюючи в програмі 3.2 двобайтову команду ANI<D> на однобайтові ANA A, XRA A, ORA A, дослідити результати їх виконання за числом, записаним у вихідному пристрої.

**Завдання 3.** Дослідити програму 3.3.

#### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 3.3. Здійснити пуск програми і впевнитись, що при її виконанні мікро-ЕОМ реагує лише на ті числа у вхідному пристрої, які містять 1 в п'ятому розряді. Після закінчення вико-

нання програми (виконання в програмі команди RST1) в розряді Z регістра стану записана 1.

2. Дослідити програму 3.3, яка змінила вигляд і дозволяє мікро-ЕОМ реагувати на 0 в п'ятому розряді при записаних 1 у всіх інших розрядах.

**Завдання 4.** Дослідити програму 3.4.

**Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 3.4. Здійснити пуск програми.
2. Дослідити програму 3.4.

Таблиця 3.5 - Текст програми 3.4

8000	3E		A,81H	
8001	81		MVI	
8002	D3		OUT 251	
8003	FB			
8004	DB	CNT	IN 250	
8005	FA			
8006	E6		ANI 246	
8007	F6			
8008	D3		OUT 249	
8009	F9			
800A	C3		JMP CNT	
800B	04			
800C	80			

**Завдання 5.** Дослідити програми 3.5, 3.6, 3.7.

**Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 3.5. Здійснити пуск програми.
2. Дослідити програму 3.5.

Текст програми 3.5.

```
.org 8000h
mvi A,81H
out 251
cnt: in 250
mvi b,0ah
ana b
out 249
jmp cnt
.end
```

Програма 3.6.

```
.org 8000h
; засвічування всіх індикаторів
mvi A,81H
```

```

cnt:   out 251
       in 250
       mvi b,04h
       ana b
       jz off
       mvi a,255
       out 249
       jmp cnt
off:   mvi a,00h
       out 249
       jmp cnt
       .end

```

### Текст програми 3.7.

.org 8000h  
; програма послідовного засвічування світлодіодів

```

mvi A,81H
out 251
seq:   mvi a, 92h
       out 249
       mvi d,1
       call delay
       mvi a,49h
       out 249
       mvi d,1
       call delay
       mvi a,24h
       out 249
       mvi d,1
       call delay
       jmp seq
delay: lxi b,2500h
loop:  dcx b
       mov a,b
       ora c
       jnz loop
       dcr d
       jnz delay
       ret
       .end

```

### **Робота на навчальній ЕОМ**

Адреса вхідного пристрою – FA, а вихідного – F9. Таким чином, всі програми, приведені в лабораторній роботі, завдання і порядок їх

виконання можуть бути без будь-яких змін виконані на навчальній мікро-ЕОМ.

### Зміст звіту

Звіт повинен містити:

1. Схеми під'єднання зовнішніх пристроїв до мікро-ЕОМ.
2. Самостійно розроблені і дослідженні в процесі виконання лабораторної роботи програми, вказані в п. 8 завдання для домашньої підготовки.
3. Повний перелік команд передачі керування за умовою для ВІС МП КР580.
4. Повний перелік команд логічних операцій для ВІС МП КР580ВМ80А.
5. Опишіть роботу програм 3.1-3.7.

### ЛАБОРАТОРНА РОБОТА №4 ПІДПРОГРАМА І СТЕК

**Мета роботи:** дослідження особливостей запису і звертання до підпрограм; вивчення методів використання стеку при створенні програм.

#### Короткі відомості з теорії

Частина програми, яка неодноразово повторюється, або програма, яка часто використовується, можуть бути оформлені у вигляді підпрограм. **Підпрограма** – послідовність команд, виконання яких може бути викликано багаторазово з будь-якого місця програми. Це зумовлено тим, що пам'ять мікро-ЕОМ, яка побудована на основі МПК серії К580, може мати не більше 65536 одnobайтних комірок. Враховуючи обмежені можливості пам'яті при розробці програм, потрібно намагатись зробити їх як можна коротшими. Процес передачі керування до підпрограми називається її **викликом**. Дані і адреси, потрібні для роботи підпрограми, називаються **вхідними параметрами**. Результати роботи підпрограми, що передаються після закінчення її виконання в основну програму, називаються **вихідними параметрами**.

Для виклику підпрограми і повернення з неї використовуються команди  $CALL\langle A_{12} \rangle \langle A_{11} \rangle$  і  $RET$ .

Команда  $CALL \langle A_{12} \rangle \langle A_{11} \rangle$  завантажує в програмний лічильник МП ВІС вміст байтів  $\langle A_{12} \rangle \langle A_{11} \rangle$ , записаних в наступних двох комірках пам'яті після комірки пам'яті, де записаний код команди виклику підпрограми  $CALL (CD)$ . Байти  $\langle A_{12} \rangle \langle A_{11} \rangle$  вказують на комірку пам'яті з якої починається підпрограма. Команда  $CALL \langle A_{12} \rangle \langle A_{11} \rangle$  є трибайтовою командою. Першим байтом є код команди. Вміст другого байта  $\langle A_{12} \rangle$  записується в молодший байт PCL програмного лічильника, а третій байт

<A<sub>11</sub>> команди - в старший байт РСН програмного лічильника, при цьому МП ВІС автоматично зберігає в стеку адресу основної програми, до якої вона буде звертатися після виконання підпрограми.

**Стек** – спеціально організована область ОЗП, задіяна в мікро-ЕОМ для тимчасового зберігання даних або адрес. Число, записане в стек останнім, дістається з нього першим.

Команда RET (C9) записує в програмний лічильник останнє число, яке на даний момент знаходиться в стеку. Після цього виконання програми буде здійснюватись з цієї адреси. Будь-яка підпрограма повинна закінчуватись командою RET.

Автоматичне зберігання і відновлення адреси основної програми при виконанні підпрограм дозволяє зробити підпрограми вкладеними, тобто здійснити виклик однієї підпрограми з іншої. Рівень вкладеності підпрограм (для даної мікро-ЕОМ) визначається лише розміром стека.

Є також команди умовного виклику підпрограм і повернення з них. Вони дозволяють викликати підпрограму і повернутися з неї за певним станом заданих розрядів регістра ознак (аналогічно командам умовних переходів). Всі команди умовного виклику підпрограми – трибайтові. В другому і третьому байтах записується початкова адреса підпрограми. Команди виклику підпрограм і повернення з них використовують стек і внутрішній регістр МП ВІС SP (STACK POINTER) для адресації до стека.

Крім команд виклику підпрограм і повернення з них зі стеком можна обмінюватись інформацією за допомогою команд PUSH<R> (записати в стек вміст позначеного регістра МП ВІС) і POP<R> (записати дані зі стека в позначений регістр МП ВІС). Ці команди є однобайтовими, і в них міститься вказівка на пару регістрів МП ВІС.

При запису в стек вмісту пари регістрів або програмного лічильника за адресою SP-1 записується вміст старшого регістра з вказаної пари або старший байт РСН програмного лічильника, а за адресою SP-2 в стек записується вміст молодшого регістра з вказаної пари молодшого байта PCL програмного лічильника.

При запису зі стека даних (в пару регістрів або в програмний лічильник) в молодший регістр пари (або в PCL) записується число з адреси, вказаної в покажчику стека SP, а в старший регістр пари або РСН – число, записане за адресою SP+1. В результаті виконання команди вміст покажчика стека SP збільшується на 2. Дані в пам'яті не змінюються, а лише відбувається їх читання і збільшення вмісту SP.

Таким чином, при запису даних адреси стека зростають від більших до менших, а показник стека SP завжди містить останню адресу стека, в якій записано число.

При розробці програм необхідно спочатку призначати область стека, записуючи в SP адресу за допомогою команди LXI SP<A<sub>2</sub>> <A<sub>1</sub>> або команди SPHL.

Всі операції зі стеком повинні бути збалансовані, тобто кожна підпрограма повинна містити однакову кількість команд PUSH<R> і POP<R> та закінчуватись командою RET. В протилежному випадку виконання команди RET в кінці підпрограми призводить до запису в програмний лічильник випадкового числа зі стека. Адреса повернення в основну програму буде втрачена і зміниться послідовність її виконання.

Як правило, на початку кожної програми зберігають в стеку вміст всіх задіяних при її виконанні регістрів за допомогою команд PUSH<P>. В кінці підпрограми відновлення вмісту регістрів здійснюється за допомогою команд POP<R> і в зворотній послідовності по відношенню до їх запису в стек.

Звичайно у вигляді підпрограм записуються багатократно використані фрагменти програм, наприклад підпрограма видачі звукового сигналу, підпрограма обслуговування клавіатури і дисплея тощо.

Алгоритм роботи часової затримки наведений на рис. 4.1. Загальний час затримки обчислюється за формулою

$$T_D = t_1 + (t_2 + t_3 + t_4)N_i + t_5,$$

де  $N_i$  - число, спочатку записане в лічильнику.

Як лічильник вибраний регістр В, в який записується число  $N_i$  з регістра С.

Команда NOP потрібна для збільшення часу виконання циклу, а таким чином, і загальної затримки. Замість команди NOP може бути записана будь-яка послідовність команд, виконання яких не змінює вміст регістрів мікропроцесора. Час запису числа  $N_i$  в регістр В і повернення з підпрограми  $t_1 + t_5$  фіксований і в цикл не входить. Мінімальна затримка для наведеної підпрограми визначається при  $N_i = 0,1$  і дорівнює

$$T_{DMN} = t_1 + t_2 + t_3 + t_4 + t_5.$$

Максимальна затримка має місце при  $N_i = 00$  і обчислюється за формулою

$$T_D = t_1 + (t_2 + t_3 + t_4)256 + t_5.$$

Підпрограма DLY1 (підпрограма 4.1) є підпрограмою часової затримки, яка записана у відповідності з алгоритмом, показаним на рис. 4.1. Розглянемо приклад використання підпрограми часової затримки при організації звукових сигналів в мікро-ЕОМ.

В мікро-ЕОМ звукові сигнали можуть формуватися найпростішою схемою, на вхід якої зі звуковою частотою записуються по черзі "0" і "1". Схема алгоритму роботи підпрограми генерації звукових коливань (підпрограма 4.2) наведена на рис. 4.2.

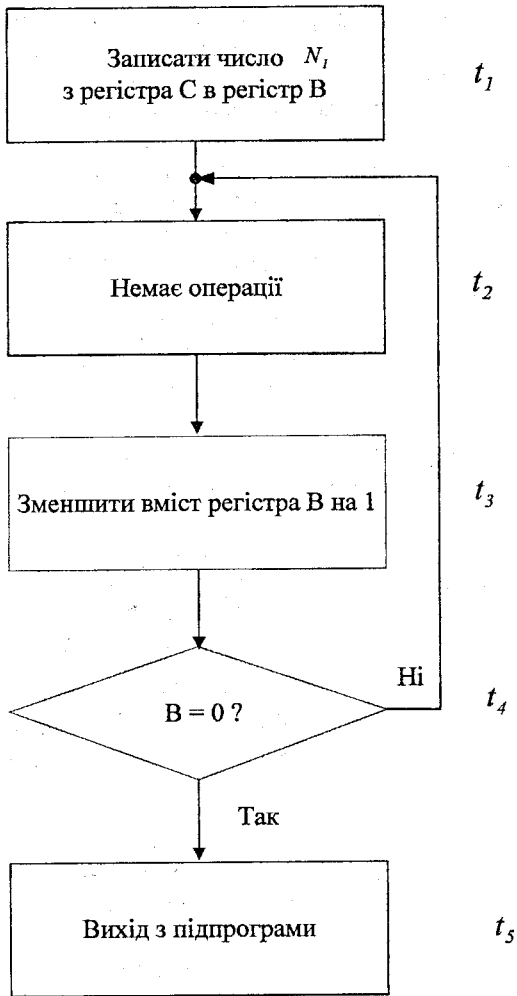


Рисунок 4.1 - Алгоритм простої підпрограми часової затримки

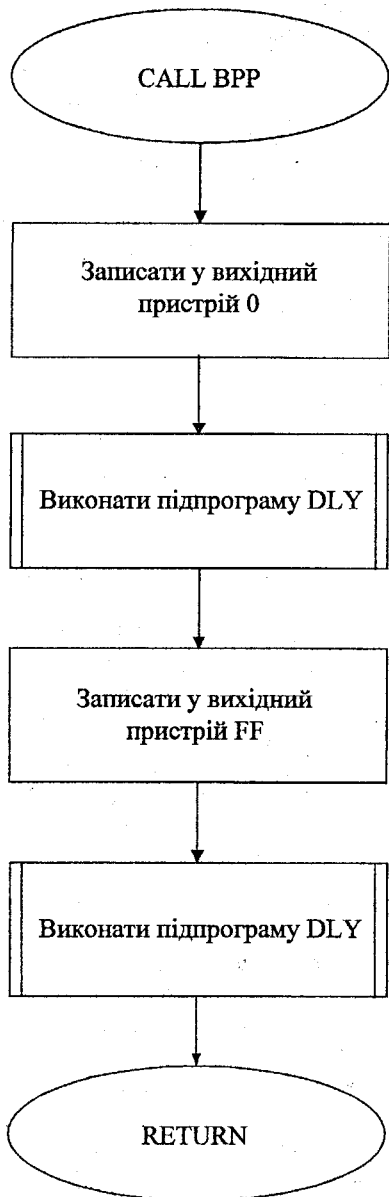


Рисунок 4.2 – Алгоритм підпрограми генерації звукових коливань

Таблиця 4.1 - Текст підпрограми 4.1 "DLY1"

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8100	41		MOV B,C	Записати число з регістра С в регістр В
8101	00	DLY	NOP	Немає операції
8102	05		DCR B	Зменшити число в регістрі В на 1
8103	C2		JNZ DLY	Якщо число, записане в регістрі В, не дорівнює нулю, то йти на мітку DLY
8104	00			
8105	81			
8106	C9		RET	Вихід з підпрограми

Таблиця 4.2 - Текст підпрограми 4.2 "BPP"

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8010	AF	BPP	XRA A	Очистити акумулятор
8011	D3		OUT F9	Записати 00 у вихідний пристрій
8012	F9			
8013	CD		CALL 8100	Викликати підпрограму затримки DLY1
8014	00			
8015	81			
8026	2F		CMA	Записати код FF в акумулятор
8027	D3		OUT F9	Записати код FF у вихідний пристрій
8028	F9			
8029	CD		CALL 8100	Викликати підпрограму затримки DLY1
802A	00			
802B	81			
802C	C9		RET	Вихід з підпрограми

Програма MAIN (програма 4.3) є програмою генерації звукового сигналу.

При використанні підпрограм за рахунок зміни вхідних параметрів можна впливати на кінцевий результат виконання підпрограм. Розглянемо це на прикладі підпрограми регульованої затримки часу (підпрограма 4.4).

Підпрограма складається з двох підпрограм: DELB (адреса 8130) і DELA (адреса 8129). Підпрограма DELB здійснює регульовану часову затримку. Вхідним параметром її є двобайтове число, записане в парі регістрів (В, С). Це число і визначає тривалість затримки в мілісекундах. Підпрограма DELA схожа на підпрограму DELB. Вона здійснює фіксовану затримку в 1 мс і не має вхідних параметрів. Якщо необхідно мати в програмі фіксовану затримку в 1 мс, то можна звертатися за допомогою команди CALL до підпрограми DELA. При необхідності отримання певної

часової затримки в програмі необхідно записати відповідне число в регістри В, С, а потім викликати підпрограму DELB.

Таблиця 4.3 - Текст програми 4.3

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	3E		MVI A,81	
8001	81			
8002	D3		OUT FB	
8003	FB			
8004	DB	MAIN	IN FA	Записати число з вхідного регістра в акумулятор.
8005	FA			
8006	4F		MOV C, A	Записати число з акумулятора в регістр С.
8007	CD		CALL BPP	Викликати підпрограму BPP.
8008	10			
8009	80			
800A	C3		JMP MAIN	Зацикловання.
800B	04			
800C	80			

#### Завдання для домашньої підготовки

1. Вивчіть часові діаграми виконання мікро-ЕОМ команд CALL<A<sub>2</sub>> <A<sub>1</sub>> і RET.

2. Ознайомтесь з командами виклику і повернення з підпрограми за умовою для МП ВІС.

3. Визначте, при яких числах, записаних в регістрі В, підпрограма 4.1 буде здійснювати мінімальний і максимальний час затримки. Визначте цей час, якщо машинний такт T = 1 мкс.

4. Розробіть підпрограму 5- і 10-секундної часової затримки.

5. Складіть програму, яка послідовно вмикає світлодіоди вихідного пристрою на час відповідно 10 і 5 с. При розробці програми скористаємося підпрограмами, розробленими в п. 4 завдання. Ознайомтесь з текстом програми 4.7.

6. Напишіть текст програми 4.5, що визначає, який з трьох перемикачів вхідного пристрою мікро-ЕОМ встановлений в положення "0":

а) складіть алгоритм роботи програми ;

в) встановіть, в якому регістрі МП ВІС міститься інформація про номер перемикача вхідного пристрою, який встановлений в "0";

г) визначте, як буде працювати програма 4.5, якщо на вхідному пристрої будуть встановлені в стан "0" не один, а два перемикачі.

Таблиця 4.4 – Текст підпрограми 4.4

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8129	C5	DELA	PUSH B	Зберегти вміст регістрів B і C в стеку.
812A	01		LXI B, 0100	Встановити тривалість 1 с.
812B	01			
812C	00			
812D	C3			
812E	31		JMP DEL1	
812F	81			
8130	C5	DELB	PUSH B	Зберегти вміст регістрів B і C в стеку.
8131	F5	DEL1	PUSH PSW	Зберегти вміст PSW в стеку.
8132	AF		XRA A	Очистити акумулятор.
8133	D5		PUSH D	Зберегти вміст регістрів D і E в стеку.
8134	16	DEL 2	MVI D,	Завантажити лічильник.
8135	67		TIME	
8136	15	DEL 3	DCR D	1 с затримки, зменшити вміст лічильника на 1.
8137	C2		JNZ DEL 3	Якщо не 0, то перейти на DEL 3, інакше продовжувати.
8138	36			
8139	81			
813A	0B		DCX B	Зменшити вміст лічильника на 1.
813B	B8		CMP B	Якщо не 0, то перейти на DEL 2, інакше продовжувати.
813C	C2		JNZ DEL2	
813D	34			
813E	81			
813F	B9		CMP C	Якщо не 0, то перейти на DEL 2, інакше продовжувати.
8140	C2		JNZ DEL2	
8141	34			
8142	81			
8143	D1		POP D	Відновити вміст регістрів D, E.
8144	F1		POP PSW	Відновити вміст регістрів PCW.
8145	C1		POP B	Відновити вміст регістрів B, C.
8146	C9		RET	Повернутися з підпрограми.

### **Завдання до лабораторної роботи**

**Завдання 1.** Дослідити процес виконання команд виклику і повернення з підпрограми, а також команд роботи зі стеком.

#### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 4.6.
2. Виконати програму 4.6 за командами, використовуючи покроковий режим виконання програми. Для цього перемикач режиму встановити в положення "Шаг". Після кожної команди перевірити вміст всіх регістрів МП ВІС.
3. Виконати команди CALL STDY, PUSH PSW, POP, RET за машинними циклами і побудувати часові діаграми їх виконання мікро-ЕОМ.
4. Замінити в програмі 4.6 команду POP PSW на NOP (00) і простежити, як буде виконуватись програма 4.6. Пояснити зміни, що відбулися.

**Завдання 2.** Дослідити програму часової затримки на прикладі роботи програми генерації звукових коливань (програма 4.3).

#### **Порядок виконання завдання:**

1. Записати повний текст програми 4.3 для генерації звукових сигналів з урахуванням підпрограм 4.1 і 4.2. Ввести в мікро-ЕОМ програму.
2. Здійснити пуск програми з адреси 8000.
3. Простежити за виконанням програми.

**Завдання 3.** Дослідити програму, що регулюється часовою затримкою на прикладі програми, яка послідовно вмикає і вимикає світлодіоди вихідного пристрою на час відповідно 10 і 5 с.

#### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму, розроблену в п.5 завдання для домашньої підготовки.
2. Здійснити пуск програми і перевірити її виконання в мікро-ЕОМ.
3. Змінити числа, які регулюють часову затримку. Перевірити можливості зміни тривалості затримки.

**Завдання 4.** Дослідити програму 4.5.

#### **Порядок виконання завдання:**

1. Ввести програму 4.5 в мікро-ЕОМ.
2. Встановити перемикачі в таке положення, щоб мікро-ЕОМ знаходилась в режимі очікування появи 1 у відповідному розряді вхідного пристрою.

Таблиця 4.5 – Текст програми 4.6.

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	31		LXI SP, 8000	Записати в покажчик стека SP адресу 8000.
8001	80			
8002	00			
8003	CD		CALL STDY	Викликати підпрограму STDY.
8004	07			
8005	80			
8006	CF		RST 1	Зупинити виконання програми.
8007	F5	STDY	PUSH PSW	Записати слово стан МП ВІС в стек.
8008	C5		PUSH B	Записати вміст регістрів BC в стек.
8009	D5		PUSH D	Записати вміст регістрів D, E в стек.
800A	E5		PUSH H	Записати вміст регістрів H, L в стек.
800B	3E		MVI A 05	Записати в регістр А число 05.
800C	05			
800D	47		MOV B, A	Записати число з регістра А в регістр В.
800E	87		ADD A	Подвоїти вміст акумулятора.
800F	5F		MOV E, A	Записати вміст регістра А в регістр Е.
8010	67		MOV H, A	Записати вміст регістра А в регістр H.
8011	E1		POP H	Записати числа зі стека в регістри H, L.
8012	D1		POP D	Записати числа зі стека в регістри D, E.
8013	C1		POP B	Записати числа зі стека в регістри B, C.
8014	F1		POP PSW	Записати числа зі стека в регістри МП ВІС.
8015	C9		RET	Повернутися з підпрограми.

3. Встановити 0 за допомогою перемикачів у відповідному розряді вхідного пристрою. Перевірити вміст регістрів МП ВІС після закінчення виконання програми 4.5.

4. Здійснити повторний пуск програми при наявності нулів в двох розрядах вхідного пристрою. Яке число буде записано в регістрі В МП ВІС після закінчення виконання програми?

### Робота на навчальній ЕОМ

Всі наведені в лабораторній роботі програми і порядок їх виконання можуть бути без будь-яких змін виконані на навчальній мікро-ЕОМ.

### Зміст звіту

Звіт повинен містити:

1. Часові діаграми виконання команд  $CALL\langle A_2 \rangle$   $\langle A_1 \rangle$  і  $RET$ .
2. Відповіді на запитання п. 4-6 завдання для домашньої підготовки.
3. Повний перелік команд виклику і повернення з підпрограми для МП ВІС КР580ВМ80А.
4. Розроблені в процесі домашньої підготовки програми 5- та 10-секундної затримки.
5. Перелік команд роботи зі стеком для МП ВІС КР580ВМ80А.
6. Відповіді на запитання, поставлені в п. 7 завдання для домашньої підготовки.

### Завдання для самоперевірки

1. Вкажіть кількість машинних тактів виконання команди  $CALL\langle A_2 \rangle$   $\langle A_1 \rangle$ .
2. В якій послідовності записується і зчитується зі стека вміст акумулятора і регістра ознак МП ВІС при виконанні команд  $PUSH\ PSW$  і  $POP\ PSW$ ?
3. За допомогою яких команд можна задати або перепозначити область пам'яті, що відведена під стек?
4. Вкажіть порядок виконання мікро-ЕОМ команди  $RET$ .
5. Порівняйте процес виконання мікро-ЕОМ команд  $CALL$  і  $RST$ .
6. В якій послідовності зберігається і дістається вміст регістрів МП ВІС в підпрограмах?
7. Якою мінімальною адресою буде записаний в покажчик стека  $SP$  МП ВІС при виконанні підпрограми 4.6?
8. Як буде виконуватись підпрограма 4.6, якщо замість команди  $POP\ V$  в ній буде записана команда  $NOP$ ?
9. Який мінімальний час затримки може забезпечувати підпрограма 4.1, якщо тривалість машинного такту  $T = 1\ \mu\text{с}$ ?
10. Який максимальний і мінімальний час затримки може забезпечити підпрограма 4.4, якщо тривалість машинного такта  $T = 1\ \mu\text{с}$ ?
11. Як збільшити час затримки з використанням підпрограми 4.4?

Таблица 4.7 – Текст программы 4.7

Адреса	Машинный код	Метка	Мнемокод	Комментар
8000	3E		mvi a,81h	
8001	81			
8002	D3		out 251	
8003	FB			
8004	AF	bpp:	xra a	
8005	D3		out 249	
8006	F9			
8007	16		mvi d,1	
8008	01		call delay	
8009	CD			
800A	17			
800B	80			
800C	2F		cma	
800D	D3		out 249	
800E	F9			
800F	16		mvi d,1	
8010	01			
8011	CD		call delay	
8012	17			
8013	80			
8014	C3		jmp bpp	
8015	04			
8016	80			
8017	01	delay:	lxi b,4fffh	
8018	4F			
8019	FF			
801A	0B	loop1:	dcx b	
801B	78		mov a,b	
801C	B1		ora c	
801D	C2		jnz loop1	
801E	1A			
801F	80			
8020	15		dcr d	
8021	C2		jnz delay	
8022	17			
8023	80			
8024	C9		ret	
8025	79		hlt	

## ЛАБОРАТОРНА РОБОТА №5 ВИКОНАННЯ АРИФМЕТИЧНИХ ОПЕРАЦІЙ ДОДАВАННЯ ТА ВІДНІМАННЯ

**Мета роботи:** вивчення способів організації і дослідження програм виконання арифметичних операцій додавання та віднімання.

### Короткі відомості з теорії

З двох способів визначення чисел з фіксованою і плаваючою точкою, перший отримав найбільше розповсюдження при програмуванні мікро-ЕОМ на МП КР580ВМ80А. Це пов'язано з відсутністю спеціальних команд, що дозволяють працювати МП з числами з плаваючою точкою. В свою чергу, двійкове восьмирозрядне число з фіксованою точкою можна подати як двійкове число зі знаком, що має значення від  $-128_{10}$  до  $+127_{10}$ . При цьому від'ємні числа подаються в додатковому коді, а старший, сьомий розряд числа використовується як знаковий. Таке подання чисел не дозволяє виконувати арифметичні операції з використанням перенесення при складанні і позичання при відніманні.

Число з фіксованою точкою можна подати також двійковими числами без знака, що має значення від 0 до  $255_{10}$ .

Для МП КР580ВМ80А можна подати такі числа у вигляді двійково-десятькового числа – BINARY - CODED - DECIMAL (BCD), при якому кожний байт розглядається як два півбайта – дві тетради, кожна з яких кодує десяткову цифру. Таке подання дозволяє закодувати в 1 байт числа від 0 до  $99_{10}$ .

Таблиця 5.1 - Текст програми 5.1

Адреса	Машин-ний код	Мітка	Мнемокод	Коментар
1	2	3	4	5
8000	21	MAIN	LXI H,8020	Записати в реєстри H, L адресу першого доданку
8001	20			
8002	80			
8003	06		MVI B, 05	Завантажити в реєстр B кількість доданків
8004	05		CALL SADD	Викликати підпрограму додавання
8005	CD			
8006	09			
8007	80			
8008	CF		RST 1	Перервати виконання програми
8009	AF	SADD	XRA A	Очистити акумулятор
800A	4F		MOV C, A	Очистити лічильник перенесення
800B	86	CNT	ADD M	Додати до вмісту акумулятора число з масиву доданків

Продовження таблиці 5.1

1	2	3	4	5
800C	D2		JNC TRM	Якщо перенесення немає, то перейти на TRM
800D	10			
800E	80			
800F	0C		INR C	Збільшити вміст регістра C на 1
8010	23	TRM	INX H	Вказати на наступну адресу доданка
8011	05		DCR B	Зменшити вміст лічильника доданків
8012	C2		JNZ CNT	Якщо не всі доданки, то перейти на CNT
8013	0B			
8014	80			
8015	C9		RET	Повернутися з підпрограми

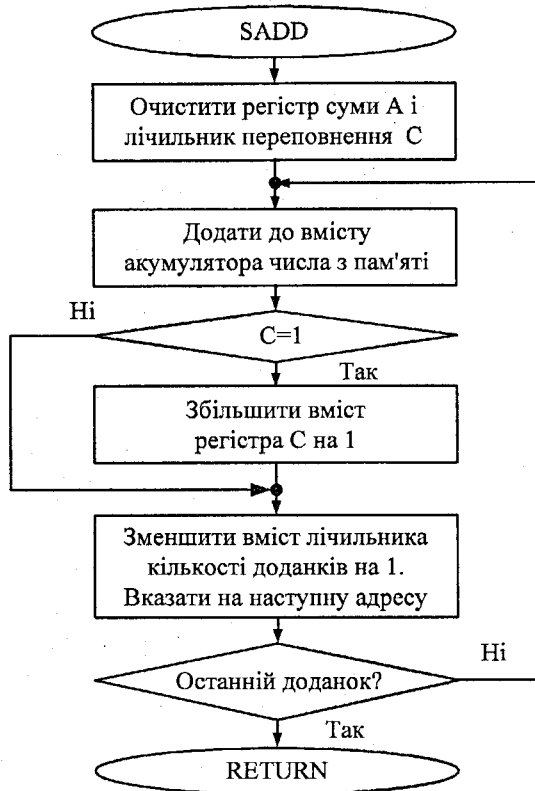


Рисунок 5.1 - Алгоритм підпрограми додавання

Програма додавання масиву одnobайтових чисел з отриманням двобайтового результату – програма MAIN (програма 5.1). Доданки повинні бути розташовані в послідовних адресах пам'яті. Вхідними параметрами з підпрограми SADD є адреса першого доданка, записана в регістрах H, L, і число доданків, записаних в регістрі B. Вихідним параметром програми MAIN є сума, старший байт якої записаний в регістрі C, а молодший – в акумуляторі A. Алгоритм програми додавання полягає в тому, що після додавання кожного елемента масиву визначається переповнення акумулятора (розряд C-1), і якщо це має місце, то вміст регістра C збільшується на 1. Таким чином, за рахунок додавання одиниць перенесення отримується старший байт суми. Мікро-ЕОМ може проводити арифметичні операції з числами з подвійною або більшою довжиною машинного слова. Оскільки МП КР580ВМ80А має 8-розрядний АЛП, то операції з такими числами повинні проводитись побайтово, починаючи з молодших байтів.

Так, операція додавання чисел  $17F5_{16} + 3411_{16}$  буде проводитись у відповідності до таблиці 5.2.

Таблиця 5.2 - Додавання чисел  $17F5_{16} + 3411_{16}$

Старший байт	Прапорець C	Молодший байт	Числа
00010111		11110101	17F5
+		+	+
00110100		00010001	3411
+1	←1		
01001100		00000110	4C06

Операція віднімання чисел  $6F5C_{16} - 13C5_{16}$  буде здійснюватись у відповідності до таблиці 5.3.

З наведених прикладів видно, що при додаванні (відніманні) молодших байтів чисел необхідно застосовувати команду ADD (SUB), а для додавання (віднімання) інших – команду ADC (SBB), яка буде враховувати стан розряду регістра C ознак МП.

Таблиця 5.3 - Віднімання чисел  $6F5C_{16} - 13C5_{16}$

Старший байт	Прапорець C	Молодший байт	Числа
00110111		01011100	6F5C
-		-	-
00010011		11000101	13C5
-1	←1		
01011011		10010111	5B97

Програма знаходження різниці чисел, що мають однакову довжину, - програма 5.2. Вхідні параметри: регістр C – довжина чисел (в байтах),

реєстри H, L – адреса молодшого байта від'ємника, реєстри D, E – адреса молодшого байта зменшуваного. Кожне з чисел записується в послідовних адресах пам'яті, починаючи з молодших байтів. Результат заноситься в область пам'яті, що відводиться під від'ємник. В тому випадку, якщо зменшуване менше від'ємника, буде подаватися звуковий сигнал.

Таблиця 5.4 - Текст програми 5.2

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	AF	SBN	XRA A	Очистити акумулятор і прапорець C
8001	1A	CNT	LDAX D	Записати в акумулятор зменшуване
8002	9E		SBB M	Відняти з вмісту акумулятора від'ємник
8003	77		MOV M, A	Записати різницю на місце від'ємника
8004	23		INX H	Вказати на наступний байт від'ємника
8005	13		INX D	Вказати на наступний байт зменшуваного
8006	0D		DCR C	Зменшити вміст лічильника довжини числа
8007	C2		JNZ CNT	Якщо не останній (старший) байт числа, то перейти на CNT
8008	01			
8009	80			
800A	D2		JNC BRK	Якщо байт старший і результат без позичання (C = 0), то перейти на BRK
800B	10			
800C	80			
800D	CD		CALL BEEP	Якщо було позичання (C = 1), то подати сигнал
800E	50			
800F	03			
8010	CF	BRK	RST 1	Перервати виконання програми

Виконання MOV M, A; INX D не впливає на розряд C. В програмі SBN підпрограма звукового сигналу (BEEP) починається з адреси 0350<sub>16</sub>. При дослідженні програми SBN необхідно перед початком виконання її здійснити початкове встановлення усіх вхідних параметрів.

Таблиця 5.5 - Текст програми 5.3 на додавання двох чисел, розміщених у оперативній пам'яті.

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	21		LXI H,8021	Завантажити у HL-пару адресу 8021
8001	21			
8002	80			
8003	3A		LDA 8020	Завантажити у акумулятор число з комірки за адресою 8020
8004	20			
8005	80			
8006	86		ADD M	Додати вміст акумулятора до вмісту комірки пам'яті, на яку вказує HL-пара
8007	32		STA 8022	Розмістити результат в комірці 8022
8008	22			
8009	80			
800A	CF		RST 1	Закінчити роботу програми

Всі наведені в лабораторній роботі програми, а також порядок їх виконання можуть бути досліджені на навчальній мікро-ЕОМ.

Таблиця 5.6 - Текст програми 5.4 на віднімання двох чисел, розміщених у оперативній пам'яті

Адреса	Машинний код	Мітка	Мнемокод	Коментар
1	2	3	4	5
8000	21		LXI H, 8021	Завантажити у H-L-пару адресу 8021
8001	21			
8002	80			
8003	3A		LDA, 8020	Завантажити у акумулятор число з комірки за адресою 8020
8004	20			
8005	80			
8006	96		SUB M	Відняти від вмісту акумулятора вміст комірки пам'яті, на яку вказує H-L-пара

Продовження таблиці 5.6

1	2	3	4	5
8007	32		STA, 8022	Розмістити результат в комірці 8022
8008	22			
8009	80			
800A	CF		RST 1	Закінчити роботу програми

**Завдання до лабораторної роботи**

**Завдання 1.** Дослідити програму додавання одnobайтових чисел з використанням двобайтового результату.

**Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 5.1.
2. Записати в мікро-ЕОМ послідовність з п'яти чисел, починаючи з адреси 8020<sub>16</sub>.
3. Здійснити пуск програми і перевірити її виконання за даними, записаними в регістрі С і акумуляторі МП.
4. Змінити програму 5.1 так, щоб результат виконання записувався за адресою 8050<sub>16</sub> і 8060<sub>16</sub>. Перевірити результат її виконання.
5. Замінити в програмі покажчик кількості доданків в сумі і виконати програму знову.

**Завдання 2.** Дослідити програму віднімання двох чисел, що мають однакову довжину.

**Порядок виконання завдання:**

1. Ввести програму 5.2 в мікро-ЕОМ.
2. Записати в регістри D, E і H, L відповідно початкові адреси молодших байтів зменшуваного і від'ємника. Записати в регістр С довжину числа в байтах. Для першого випадку записати в регістр С число 01 (розглядається віднімання двох 8-розрядних чисел).
3. Записати за адресами, що вказані в регістрах H, L і D, E, зменшуване і від'ємник. Вибрати при цьому зменшуване більшим за від'ємник.
4. Здійснити пуск програми 5.2 і дослідити результат її виконання за числом, що записане за адресою, де записаний від'ємник.
5. Змінити програму 5.2 так, щоб результат обчислення різниці двох 8-розрядних чисел записувався за адресою 8055<sub>16</sub>.
6. Змінити числа, що записані за адресами, вказаними в регістрах H, L і D, E, так, щоб зменшуване було меншим за від'ємник.
7. Здійснити пуск програми і простежити, що при цьому отримаємо.
8. Дослідити в аналогічній послідовності результат віднімання двобайтових чисел.

Програма 5.2 побудована так, що в тому випадку, коли зменшуване менше за від'ємник, мікро-ЕОМ буде подавати звуковий сигнал.

### Зміст звіту

Звіт повинен містити:

1. Повний перелік арифметичних команд МП КР580ВМ80А.
2. Програму додавання двох 8-розрядних чисел з отриманням результату в двійково-десятковому коді, яку необхідно написати самостійно під час підготовки до лабораторної роботи.
3. Змінений варіант програми 5.1, яка досліджувалась в завданні 1.
4. Змінений варіант програми 5.2, яка досліджувалась в завданні 2.
5. Результати виконання арифметичних операцій за всіма завданнями.

### ЛАБОРАТОРНА РОБОТА №6

### ВИКОНАННЯ АРИФМЕТИЧНИХ ОПЕРАЦІЙ МНОЖЕННЯ ТА ДІЛЕННЯ

**Мета роботи:** вивчення способів організації і дослідження програм виконання арифметичних операцій множення та ділення.

### Короткі відомості з теорії

**Множення чисел.** Існує декілька алгоритмів множення чисел. При першому алгоритмі множення можна замінити багатократним додаванням, наприклад  $14 \cdot 3 = 14 + 14 + 14$ . Існує один недолік цього способу – значна тривалість процесу обчислення. При другому алгоритмі множення здійснюється в стовпець. Цей алгоритм застосовується і для множення двійкових чисел, наприклад

$$0110 = 6_{10}$$

$$0011 = 3_{10}$$

$$\begin{array}{r} 0110 \\ 0110 \\ 0000 \\ 0000 \\ \hline \end{array}$$

$$00010010 = 18_{10}.$$

При обчисленні результату за другим алгоритмом необхідно здійснити багатократне додавання зі зсувом наліво множеного при одночасній перевірці вмісту розрядів множника починаючи зі сторони його розряду. При цьому якщо в черговому розряді множника записана 1, то множене додається до суми і посувається наліво на один розряд, а якщо

в розряді записаний 0, то відбудеться тільки зсув множеного. Зсув множеного наліво можна замінити зсувом суми направо.

Таблиця 6.1 - Текст підпрограми 6.1

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8001	01	MPL	LXI B, 0000	Очистити вміст регістрів B, C
8002	00			
8003	00			
8004	3E		MVI A, 01	Завантажити в акумулятор показчик розряда
8005	01			
8006	A7		ANA A	Очистити прапорець C
8007	F5	MPL1	PUSH PSW	Зберегти показчик розряду в стеку
8008	A3		ANA E	Перевірити вміст чергового розряду множника
8009	78		MOV A, B	Завантажити в акумулятор старший байт суми
800A	CA		JZ MPL2	Якщо в черговому розряді записаний 0, то перейти на MPL 2
800B	0E			
800C	80			
800D	82		ADD D	Додати множене до суми
800E	1F	MPL2	RAR	Зсунути суму вправо (молодший біт → C)
800F	47		MOV B, A	Зберегти вміст акумулятора в регістрі B
8010	79		MOV A, C	Завантажити в акумулятор молодший байт суми
8011	1F		RAR	Зсунути число в акумуляторі направо (C → старший біт)
8012	47		MOV C, A	Зберегти вміст акумулятора в регістрі C
8013	F1		POP PSW	Отримати зі стека показчик розряда
8014	17		RAL	Змістити показчик на наступний розряд
8015	D2		JNC MPL1	Якщо розряд не останній, то перейти на MPL 1
8016	07			
8017	80			
8018	C9		RET	Якщо розряд останній, то повернення

За цим алгоритмом працює програма множення двох одnobайтових чисел з отриманням двобайтового результату (програма 6.1). Початкова адреса програми – 8001; вхідні параметри: регістр D – множене, регістр E – множник. Результат множення записується в регістри B, C.

Множення чисел також реалізується у мікропроцесорі за допомогою команд додавання і зсуву. Для отримання добутку двох операндів, до суми двох часткових добутків додається перший операнд, якщо черговий розряд другого операнда дорівнює одиниці, після чого здійснюється зсув часткової суми; вказана процедура повторюється для кожного розряду другого операнда. При цьому аналіз його бітів можна починати від старших розрядів і відповідний зсув виконувати вліво або діяти навпаки.

Таблиця 6.2 - Текст програми 6.2 на множення двох восьмирозрядних чисел

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	21	ML88		
8001	00		LXI H, 0000	Скинути суму регістрів часткових добутків
8002	00			
8003	06		MVI B, 08	Встановити початкове значення лічильника циклів
8004	08			
8005	29	LAB1	DAD H	Зсунути вміст регістрової пари (HL) вліво
8006	07		RLC	Ввести черговий розряд другого числа в біт переповнення
8007	D2			
8008	0B		JNC LAB2	Якщо черговий біт другого числа дорівнює нулю, то перейти на LAB1
8009	80			
800A	19		DAD D	Додати перше число
800B	05	LAB2	DCR B	Підрахувати кількість циклів
800C	C2			
800D	05		JNZ LAB1	Повторити дії, якщо обчислення не закінчені
800E	80			
800F	E7		RST 4	Зупинити

У розглянутій нижче програмі використовується такий алгоритм. Знак добутку формують незалежно від наведеної процедури на основі аналізу співмножників. При здійсненні множення використовуються прямі коди операндів без знака. Необхідно врахувати, що розряди часткових добутків, що звільняються при зсувах, повинні заповнюватися нулями. Оскільки для

подання суми часткових добутків і самого результату потрібно шістнадцять розрядів, виникає необхідність у зсуві двобайтових слів. Для цього зручно використовувати команду DAD H, яка здійснює подвоєння значення пари регістрів (HL), або логічний зсув змісту цієї пари на один розряд вліво, тобто в сторону старших розрядів.

Друге число варто розмістити в акумуляторі, з тим щоб здійснювати аналіз його розрядів за допомогою команд циклічного зсуву вліво і умовного переходу за бітом переповнення. Операцію додавання числа до часткових добутків можна виконувати за допомогою команди DAD D, якщо перше число розмістити у регістрі E, а у регістр D занести нуль.

**Ділення чисел.** Ділення двійкових чисел, як і чисел, що подані в будь-якій іншій системі обчислення, ґрунтується на послідовному відніманні дільника від діленого і залишків від ділення. Однак двійкове ділення реалізується простіше, оскільки використання тільки двох цифр (0 і 1) усуває в кожному циклі ділення необхідність визначення числа дільників, що містяться в поточному значенні діленого або остачі (достатньо тільки порівняти їх).

Програма DIVB побудована за цим алгоритмом (програма 6.3). Вхідними параметрами цієї програми є ділене (в регістрі E) і дільник (в регістрі D); вихідними параметрами – частка (в регістрі H) і залишок (в регістрі C).

#### **Завдання для домашньої підготовки**

1. Вивчити групу арифметичних команд МП КР580ВМ80А.
2. Ознайомитись з правилами використання команди DAA – десятиковою корекцією акумулятора.
3. Розглянути програми 6.1 – 6.3, наведені вище та попередньо оцінити результат виконання кожної програми на конкретних числових прикладах.
4. Провести оцінювання часу виконання програм множення і ділення 8-розрядних чисел.
5. Скласти програму для дослідження результату множення двох чисел на підставі підпрограми 6.1.
6. Розробити програму множення двох 8-розрядних чисел з отриманням результату в двійково-десятковому коді.
7. Оцінити час виконання програми 6.3.

#### **Завдання до лабораторної роботи**

**Завдання 1.** Дослідити програму множення двох 8-розрядних чисел з отриманням 16-розрядного результату.

### Порядок виконання завдання:

1. Ввести в мікро-ЕОМ програму для дослідження результату множення двох чисел, що розроблена в п. 5 завдання для домашньої підготовки.
2. Здійснити пуск програми і перевірити результат двох чисел за адресою, записаною в регістрах В, С.

**Завдання 2.** Дослідити програму ділення двох 8-розрядних чисел.

### Порядок виконання завдання:

1. Ввести в мікро-ЕОМ програму 6.3.
2. Записати в регістри E, D відповідно ділене і дільник.
3. Здійснити пуск програми і перевірити результат ділення двох чисел за вмістом регістрів H, C.

### Зміст звіту

Звіт повинен містити:

1. Повний перелік арифметичних команд МП КР580ВМ80А.
2. Програму для дослідження результату множення двох 8-розрядних чисел, розроблену в п. 5 завдання для домашньої підготовки.
3. Підпрограму 6.1, яка досліджувалась в завданні 1.
4. Програму 6.3, яка досліджувалась в завданні 2.
5. Результати виконання арифметичних операцій за всіма завданнями.

Таблиця 6.3 - Текст програми 6.3

Адреса	Машинний код	Мітка	Мнемокод	Коментар
1	2	3	4	5
8000	21	DIVB	LXI H, 0008	Завантажити лічильник бітів (регістр L) і очистити регістр частки H
8001	00			
8002	08			
8003	0E		MVI C, 00	Очистити регістр проміжного діленого
8004	00			
8005	7B	MXTB	MOV A, E	Завантажити ділене в акумулятор
8006	17		RAL	Зсунути старший байт в розряд C
8007	5F		MOV E, A	Повернути ділене в регістр E
8008	79		MOV A, C	Завантажити в акумулятор проміжне ділене з регістра C
8009	17		RAL	Зсунути розряд C в молодший байт

Продовження таблиці 6.3

1	2	3	4	5
800A	92		SUB D	Відняти від вмісту акумулятора дільник
800B	D2		JNC NADD	Якщо C = 1, то відновити вміст акумулятора
800C	0F			
800B	80			
800E	B2		ADD D	Додати до акумулятора вміст регістра D
800F	4F	NADD	MOV C, A	Повернути проміжні дані із акумулятора в регістр C
8010	3F		CMC	Інвертувати розряд C
8011	7C		MOV A, H	Зсунути розряд C в молодший байт регістра частки H
8012	17		RAL	
8013	67		MOV H, A	
8014	2D		DCRL	Чи перевірені всі вісім розрядів?
8015	C2		JNZ MXTB	Якщо ні, продовжити
8016	05			
8017	80			
8018	CF		RST 1	Якщо да, перервати виконання програми

**Завдання для самоперевірки**

1. Які команди арифметичних операцій може виконувати МП КР580ВМ80А?
2. Сформулюйте правило виконання МП команди DAA.
3. Після яких команд можна здійснювати десяткову корекцію числа акумулятора (виконувати команду DAA)?
4. Як оцінити максимальний час виконання підпрограми множення двох чисел (підпрограма 6.1)?
5. Чи можна безпосередньо дослідити множення двох чисел за підпрограмою 6.1?
6. На чому ґрунтуються алгоритми роботи підпрограм множення і ділення чисел?
7. Подайте числа від 0 до 20 в двійково-десятковому коді.
9. Вкажіть можливі способи подання чисел для МП КР580ВМ80А.
10. Оцініть максимальний час виконання програми ділення двох чисел, якщо час машинного такта для МП:  $T = 1 \text{ мкс}$ .

## ЛАБОРАТОРНА РОБОТА №7 ПІД'ЄДНАННЯ ДИСПЛЕЯ І КЛАВІАТУРИ ДО МІКРО-ЕОМ

**Мета роботи:** вивчення програмно-апаратних засобів під'єднання дисплея і клавіатури до мікро-ЕОМ.

### Короткі відомості з теорії

Як пристрій виведення інформації, зручний для сприйняття, часто використовується дисплей. Розглянемо метод під'єднання дисплея, який складається з восьми сегментних індикаторів, які мають вісім світлодіодів із загальним анодом в одному корпусі. Кожний індикатор має сім світлодіодів для відображення сегментів цифр, а восьмий світлодіод відображає десяткову точку. Індикатор може відображати цифри від 0 до 9, а також деякі літери.

Для зменшення схематехнічного забезпечення, необхідного для під'єднання дисплея до мікро-ЕОМ, часто застосовують мультиплексний режим роботи індикаторів. При цьому для виведення на дисплей інформації використовують два вихідних регістри: PrCg для запису семисегментного коду і PrCk для запису номера індикатора.

Однакові сегменти кожної комірки індикатора пов'язані загальною шиною, яка з'єднана з одним із транзисторних ключів на виході регістра PrCg. Загальні аноди індикатора під'єднані до одного із транзисторних ключів на виході регістра сканування PrCk.

Сигнали заведені на регістри PrC і PrCk (K589IP12) так, що при надходженні на вхід BK1 сигналу вибірки від дешифратора адреси, дані з шини даних ШД, яка під'єднана до виходів D<sub>0</sub>-D<sub>7</sub>, записуються в регістр і з'являються на його входах Q<sub>0</sub>-Q<sub>7</sub>. Таким чином, наприклад, при запису в регістр сегментів числа 0000 0110 відкриваються одні транзисторні ключі, а при запису в регістр сканування PrCk числа 0010 0000 відкривається інший транзисторний ключ. При цьому на лівому індикаторі висвітлиться 1. Якщо тепер за допомогою програми висвічувати по черзі всі комірки, записуючи їх код в регістр і включаючи необхідну цифру за допомогою регістра PrCk, то при достатньо високій частоті перемикань можна отримати стійке зображення інформації на дисплеї.

### Під'єднання клавіатури до мікро-ЕОМ

Клавіатура є одним з широко розповсюджених пристроїв введення даних і керованих впливів в мікро-ЕОМ. За допомогою клавіатури можна вводити програму в оперативний запам'ятовуючий пристрій (ОЗП), ініціювати різні режими роботи мікро-ЕОМ (пуск програми за машинними циклами, виведення вмісту регістрів ВІС МП на дисплей тощо).

У всіх випадках при організації введення інформації з клавіатури в мікро-ЕОМ перед розробником ставиться низка задач, до основних з яких можна віднести:

- 1) визначення факту натискання клавіші на клавіатурі;
- 2) знаходження номера натиснутої клавіші;
- 3) здійснення передачі керування на відповідну підпрограму.

Перші дві задачі є специфічними при організації введення інформації з клавіатури, і саме їм буде приділена увага в цій лабораторній роботі. Остання, як правило, вирішується програмними методами.

Найбільш зручно організувати клавіатуру у вигляді матриці розміром  $n \times m$ , де  $n$  і  $m$  – відповідна кількість рядків і стовпців. При такому способі організації до мікро-ЕОМ можна під'єднати  $m \times n$  клавіш. З'єднання клавіатури з мікро-ЕОМ виконують за допомогою пристроїв введення-виведення даних.

При підпрограмному способі дешифрації натиснутої клавіші визначення факту натискання на клавішу може бути здійснено за допомогою такої послідовності операцій:

1. Записати нуль в розряди вихідного пристрою.
2. Підрахувати вміст розрядів вхідного пристрою.
3. Повторити знову, якщо у всіх розрядах вхідного пристрою записані одиниці.

Визначити номер натиснутої клавіші можна за допомогою певного алгоритму. Алгоритм базується на послідовному записі нуля в кожний з рядків матриці клавіатури. При наявності нуля в кожному рядку мікро-ЕОМ визначає факт натискання на клавіші, які знаходяться в стовпцях клавіатури, що належать аналізованому рядку. Якщо будь-яка клавіша натиснута, то визначається її номер за номером розряду, в якому записаний нуль.

Часто при роботі мікро-ЕОМ її операційна система будується таким чином, що інформація, яка вводиться з клавіатури, показується на дисплеї.

Клавіатура є набором ключів (кнопок), організованих у вигляді матриці  $8 \times 3$ . Кожний ряд із 8 кнопок опитується окремо. Дані, які перетворюються в код, що відповідає натиснутій кнопці, за допомогою підпрограми монітора KEYIN (адреса 0216H). Ця програма викликається за допомогою команди CALL KEYIN. Після повернення з програми KEYIN в акумуляторі буде знаходитись код натиснутої кнопки (табл.7.1). Ця моніторна програма може використовуватися для читання клавіатури.

У табл.7.2 наведено текст програми, яка викликає звуковий сигнал, якщо натиснута кнопка 5. Клавіатура читається за допомогою вищеописаної програми KEYIN. Вона заносить код натиснутої клавіші в акумулятор.

Команда CPI 05 встановлює нульовий прапорець процесора, якщо вміст акумулятора дорівнює 05. Команда JNZ READ викликає повернення

програми на початок, якщо нульовий прапорець не встановлений. Якщо нульовий прапорець встановлений, то буде викликатися підпрограма ВЕЕР, Потім процес повторюється. Таким чином, звуковий сигнал буде генеруватись при натисканні кнопки 5.

Таблиця 7.1 - Коды кнопок мікролабораторії

Кнопка	Код (H)	Кнопка	Код (H)
0	00	C	0C
1	01	D	0D
2	02	E	0E
3	03	F	0F
4	04	ПУСК	10
5	05	ВОЗВР.	11
6	06	УСТ. АД.	12
7	07	АД.-	13
8	08	АД.+	14
9	09	ЗП	15
A	0A	ВЫВОД	16
B	0B	ВВОД	17

Таблиця 7.2 - Програма 7.1 виклику звукового сигналу

Адреса	Зміст	Мітка	Команда	Коментар
8000	CD	READ	CALL KEYIN	Виклик підпрограми читання з клавіатури
8001	16			
8002	02			
8003	FE		CPI 05	Порівняння кода кнопки з 05
8004	07			
8005	C2		JNZ READ	Повернення, якщо кнопка 5 не натиснута
8006	00			
8007	80			
8008	CD		CALL BEEP	Звуковий сигнал, якщо кнопка 5 натиснута
8009	50			
800A	03			
800B	C3		JMP READ	Повторення програми
800C	00			
800D	80			

Введіть текст програми в мікро-ЕОМ і перевірте правильність введення. Запустіть програму.

Натисніть кнопку 5 - динамік гудить. Натисніть будь-яку іншу кнопку - звуковий сигнал відсутній.

Таблиця 7.3 - Коди, які зчитуються з клавіатури

Кнопка	Код, який зчитується (двійковий)		Код, який зчитується (шістнадцатковий)
	1	2	
Немає натиснутих кнопок	1111	1111	FF
0	1111	1110	FE
1	1111	1101	FD
2	1111	1011	FB
3	1111	0111	F7
4	1110	1111	EF
5	1101	1111	DF
6	1011	1111	BF
7	0111	1111	7F

Таблиця 7.4 - Програма 7.2 визначення стану кнопок

Команда	Коментар
MVI A, 9F (OUT FA) IN F8	Вибір ряду кнопок 0...7 Зчитування стану кнопок

Програма KEYIN [10,11], описана вище, не тільки сканує клавіатуру, але і перетворює коди, які зчитуються з колонок (у залежності від вибраного ряду), у коди, наведені в табл.7.1.

У табл.7.5 наведена програма для перевірки стану кнопки 2. Вона використовує вищеописаний спосіб зчитування з клавіатури. Якщо кнопка 2 натиснута, то генерується звуковий сигнал. Наберіть програму на "Микролаб" і перевірте її роботу.

Таблиця 7.5 - Програма 7.3 для перевірки стану кнопки 2

Адреса	Вміст	Мітка	Команда	Коментар
8000	3E		MVI A,9F	Вибір ряду кнопок 0...7
8001	9F			
8002	D3			
8003	FA		OUT FA.	
8004	DB	READ	IN F8	Зчитування стану кнопок
8005	F8			
8006	FE		CPI FB	Порівняння акумулятора з кодом (кнопка 2)
8007	FB			
8008	C2		JNZ READ	Зчитати знову, якщо кнопка 2 не натиснута
8009	04			
800A	80			
800B	CD		CALL BEEP	Якщо натиснута кнопка 2, то з'являється звуковий сигнал
800C	50			
800D	03			
800E	C3			
800F	04		JMP READ	Повторення програми
8010	80			

### Програмування виведення даних на індикатори

У мікролабораторії для індикації використовуються вісім семи-сегментних цифрових індикаторів.

Кожному індикатору відповідає комірка пам'яті, де зберігається семисегментний код. Цей код управляє тим, як світяться сегменти індикатора. Інформація з цих комірок надходить на індикатори за допомогою спеціальної схеми, яка забезпечує динамічний режим індикації.

Задача матзабезпечення зводиться до підготування семисегментних кодів і засилення їх в комірки пам'яті 83F8...83FF. Комірка 83F8 відповідає лівому індикатору, 83FF - правому. Динамічна індикація здійснюється в мікропроцесорних системах не обов'язково за допомогою схеми. Сканування індикаторів можна здійснювати і за допомогою програми.

Монітор мікролабораторії містить підпрограму SEGCG (початкова адреса підпрограми - 01C0), яка перетворює шістнадцаткові коди 0, 1, 2...4, E, F у семисегментні і заносить їх в комірки пам'яті 83F8...83FF. Вихідні дані для цієї програми повинні знаходитися в комірках 83F4...83F7. Кожний байт із цих комірок відповідає парі індикаторів: вміст комірки 83F4 після перетворення висвітлиться на двох лівих індикаторах, вміст комірки 83F7 - на двох правих.

Напишіть програму, що виводить на два правих індикатори значення натиснутої кнопки, а на інші індикатори - нулі. Використовуйте при цьому моніторні підпрограми KEYIN і SEGCG.

Підпрограма SEGCG використовується для того, щоб одержати семисегментний код. Але ця підпрограма оперує тільки з шістнадцатковими символами від 0...F, тобто не повністю використовує можливості семисегментних індикаторів.

Таблиця 7.6 - Програма 7.4 показу на семисегментних індикаторах послідовності цифр 12345678

Адреса	Вміст	Мітка	Команда	Коментарі
8000	3E		MVI A, 12H	Завантажити в акумулятор число 12H
8001	12			
8002	32		STA 83F4H	Вивести 12H з акумулятора в комірку за адресою 83F4H
8003	F4			
8004	83			
8005	3E		MVI A, 34H	Завантажити в акумулятор число 34H
8006	34			
8007	32		STA 83F5H	Вивести 34H з акумулятора в комірку за адресою 83F5H
8008	F5			
8009	83			
800A	3E		MVI A, 56H	Завантажити в акумулятор число 56H
800B	56			
800C	32		STA 83F6H	Вивести 56H з акумулятора в комірку за адресою 83F6H
800D	F6			
800E	83			
800F	3E		MVI A, 78H	Завантажити в акумулятор число 78H
8010	78			
8011	32		STA 83F7H	Вивести 34H з акумулятора в комірку за адресою 83F7H
8012	F7			
8013	83			
8014	CD		CALL SEGCG	Звернення до підпрограми SEGCG
8015	C0			
8016	01			
8017	76		HLT	Зупинення роботи програми

Можна формувати семисегментний код і одержувати на індикаторах будь-які символи (символи, які дозволяють отримати використовувані в мікролабораторії індикатори). Як уже говорилося, семисегментні коди повинні надходити у певні комірки пам'яті. Кожний біт у цих комірках

відповідає певному сегменту (рис.7.1). Якщо біт дорівнює 1, то відповідний сегмент буде "світитися", і - навпаки.

БИТ	7	6	5	4	3	2	1	0
СЕГМЕНТ	Н	G	F	E	Л	C	B	A

Рисунок 7.1 – Відповідність розрядів та сегментів

У табл.7.6 наведена програма показу інформації, що використовує підпрограму SEGCG.

В табл.7.14 наведено текст програми 7.9, яка виводить на індикатори надпис "РОБ- -07-". Змінюючи семисегментні коди, ви маєте можливість вивести на індикацію будь-який можливий запис.

### Завдання для домашньої роботи

1. Вивчіть схеми під'єднання клавіатури і дисплея до мікро-ЕОМ.
2. Ознайомтесь з програмами 7.1 - 7.9.

Програма 7.5 призначена для перевірки виведення цифр 12345678. Під час виконання програми має скластись ілюзія переміщення цифр зліва направо. Це досягається короткочасним засвічуванням та стиранням відповідних цифр у відповідних розрядах.

Таблиця 7.8 – Текст програми 7.5 циклічного короткочасного засвічування цифр 12 34 56 78 (попарно) на семисегментних індикаторах  
.org 8000h

Адреса	Вміст	Мітка	Команда	Коментарі
1	2	3	4	5
8000			mvi a, 00h	
...			sta 83f7h	
		label1:	mvi a, 12h	
			sta 83f4h	
			call 01c0h	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83f4h	
			call 01c0h	
			mvi a, 34h	
			sta 83f5h	
			call 01c0h	
			mvi d,1	
			call delay	

Продовження таблиці 7.8

1	2	3	4	5
			mvi a, 00h	
			sta 83f5h	
			call 01c0h	
			mvi a, 56h	
			sta 83f6h	
			call 01c0h	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83f6h	
			call 01c0h	
			mvi a, 78h	
			sta 83f7h	
			call 01c0h	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83f7h	
			call 01c0h	
			jmp label1	
			hlt	
		delay:	lxi b,9fffh	
		loop1:	dcx b	
			mov a,b	
			ora c	
			jnz loop1	
			dcr d	
			jnz delay	
			ret	
			.end	

**Завдання до лабораторної роботи**

**Завдання 1.** Дослідити програми 7.4 - 7.9.

**Порядок виконання завдання:**

1. Введіть в мікро-ЕОМ програму 7.4 - 7.9.
2. Здійсніть пуск програм. Простежте за роботою програм.
3. Заповніть таблицю відповідності кодів числа, що записується в регістри, які обслуговуються підпрограмою виведення на індикатори дисплея - цифрам, що будуть висвічуватись.
4. Запишіть і перевірте коди літер E, H, P, Г, O.

5. На рис.7.2 показаний семисегментний однорозрядний індикатор. Знайдіть відповідність між сегментами (a, b, c, d, e, f, g) цього індикатора і кодами, які дають можливість засвітити ці сегменти (на крайньому правому індикаторі, на прикладі програми 7.8).

6. Знайдені коди запишіть в таблицю 7.12.

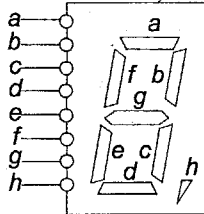


Рисунок 7.2 – Семисегментний однорозрядний індикатор

Програма 7.6 дозволяє бачити як на семисегментних індикаторах по чергово, короткочасно з'являються та зникають цифри 1 2 3 4 5 6 7 8 (по одній). При цьому складається ілюзія, що цифри рухаються зліва направо.

Таблиця 7.9 – Текст програми 7.6 циклічного короткочасного засвічування цифр 1 2 3 4 5 6 7 8 (по одній) на семисегментних індикаторах

.org 8000h

Адреса	Вміст	Мітка	Команда	Коментарі
1	2	3	4	5
			mvi a, 00h	
			sta 83feh	
			sta 83ffh	
		label1:	mvi a, 06h	
			sta 83f8h	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83f8h	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----2-----	
			mvi a, 5bh	
			sta 83f9h	

Продовження таблиці 7.9

1	2	3	4	5
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83f9h	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----3-----	
			mvi a, 4fh	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----4-----	
			mvi a, 66h	
			sta 83fbh	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83fbh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----5-----	
			mvi a, 6dh	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi d,1	

Продовження таблиці 7.9

1	2	3	4	5
			call delay	
			;-----6-----	
			mvi a, 7dh	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----7-----	
			mvi a, 07h	
			sta 83feh	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83feh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----8-----	
			mvi a, 7fh	
			sta 83ffh	
			mvi d,1	
			call delay	
			mvi a, 3fh	
			sta 83ffh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;--end--	
			jmp label1	
			Hlt	
		delay:	lxi b,9ffh	
		loop1:	dcx b	
			mov a,b	
			ora c	

Продовження таблиці 7.9

1	2	3	4	5
			jnz loop1	
			dcr d	
			jnz delay	
			ret	
			.end	

Програма 7.7 аналогічна програмі 7.6. В ній спочатку всі семисегментні індикатори не світяться, а потім є можливість бачити як на семисегментних індикаторах по чергово, короткочасно з'являються та зникають цифри 1 2 3 4 5 6 7 8 (по одній). При цьому складається ілюзія, що цифри рухаються зліва направо.

Таблиця 7.10 – Текст програми 7.7 циклічного короткочасного засвічування цифр 1 2 3 4 5 6 7 8 (по одній)

.org 8000h				
Адреса	Вміст	Мітка	Команда	Коментарі
1	2	3	4	5
			mvi a, 00h	
			sta 83f8h	
			sta 83f9h	
			sta 83fah	
			sta 83fbh	
			sta 83fch	
			sta 83fdh	
			sta 83feh	
			sta 83ffh	
		label1:	mvi a, 06h	
			sta 83f8h	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83f8h	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----2-----	
			mvi a, 5bh	
			sta 83f9h	
			mvi d,1	
			call delay	

## Продовження таблиці 7.10

1	2	3	4	5
			mvi a, 00h	
			sta 83f9h	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----3-----	
			mvi a, 4fh	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----4-----	
			mvi a, 66h	
			sta 83fbh	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fbh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----5-----	
			mvi a, 6dh	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----6-----	

Продовження таблиці 7.10

1	2	3	4	5
			mvi a, 7dh	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----7-----	
			mvi a, 07h	
			sta 83feh	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83feh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----8-----	
			mvi a, 7fh	
			sta 83ffh	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83ffh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			;-----	
			jmp label1	
			hlt	
		delay:	lxi b,9ffh	
		loop1:	dcx b	
			mov a,b	
			ora c	
			jnz loop1	
			dcr d	

Продовження таблиці 7.10

1	2	3	4	5
			jnz delay	
			ret	
			.end	

Таблиця 7.11 – Текст програми 7.8 роботи з клавіатурою .org 8000h

Адреса	Вміст	Мітка	Команда	Коментарі
1	2	3	4	5
			mvi a, 00h	
			sta 83f8h	
			sta 83f9h	
			sta 83fah	
			sta 83fbh	
			sta 83fch	
			sta 83fdh	
			sta 83feh	
			sta 83ffh	
		label1:	call 0216h	
			cpi 00h	
			jnz label2	
			;-----0-----	
			mvi a, 3fh	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi a, 00h	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi d, 1	
			call delay	
			call 0350h	
			jmp label1	
		label2:	cpi 01h	
			jnz label3	
			;-----1-----	
			mvi a, 06h	
			sta 83feh	
			mvi d, 1	
			call delay	
			mvi a, 00h	
			sta 83feh	

Продовження таблиці 7.11

1	2	3	4	5
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
			jmp label1	
		label3:	cpi 02h	
			jnz label4	
			;-----2-----	
			mvi a, 5bh	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fdh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
		label4:	cpi 03h	
			jnz label5	
			;-----3-----	
			mvi a, 4fh	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fch	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
			jmp label1	
		label5:	cpi 04h	
			jnz label6	
			;-----4-----	
			mvi a, 66h	
			sta 83fbh	
			mvi d,1	

Продовження таблиці 7.11

1	2	3	4	5
			call delay	
			mvi a, 00h	
			sta 83fbh	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
			jmp label1	
		label6:	cpi 05h	
			jnz label7	
			;-----5-----	
			mvi a, 6dh	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83fah	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
			jmp label1	
		label7:	cpi 06h	
			jnz label8	
			;-----6-----	
			mvi a, 7dh	
			sta 83f9h	
			mvi d,1	
			call delay	
			mvi a, 00h	
			sta 83f9h	
			mvi d,1	
			call delay	
			mvi d,1	
			call delay	
			call 0350h	
			jmp label1	
		Label8:	cpi 07h	
			jnz label9	

Продовження таблиці 7.11

1	2	3	4	5
			;-----7-----	
			mvi a, 07h	
			sta 83f8h	
			mvi d, 1	
			call delay	
			mvi a, 00h	
			sta 83f8h	
			mvi d, 1	
			call delay	
			mvi d, 1	
			call delay	
			call 0350h	
			jmp label1	
		label9:	cpi 08h	
			jnz labela	
			;-----8-----	
			mvi a, 7fh	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi a, 00h	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi d, 1	
			call delay	
			call 0350h	
			jmp label1	
		labela:	cpi 09h	
			jnz label1	
			;-----9-----	
			mvi a, 6fh	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi a, 00h	
			sta 83ffh	
			mvi d, 1	
			call delay	
			mvi d, 1	
			call delay	

Продовження таблиці 7.11

1	2	3	4	5
			call 0350h	
			jmp label1	
			Hlt	
		delay:	lxi b,9ffh	
		loop1:	dcx b	
			mov a,b	
			ora c	
			jnz loop1	
			dcr d	
			jnz delay	
			ret	
			.end	

Наступні програми для перевірки роботи з клавіатурою. В таблиці 7.11 наведено текст програми 7.8. Ця програма дозволяє бачити на семисегментних індикаторах у відповідних розрядах цифри від 0 до 9 лише після натискання відповідних клавіш. Кожне натискання супроводжується звуковим сигналом.

**Завдання 2.**

Дослідження програми виведення інформації на дисплей.

**Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ програму 7.7.

Таблиця 7.12 - Відповідність кодів символів - символам, що світяться на семисегментних індикаторах

Код символу	Символ
	0
	1
	2
	...
	9
	Е
	Н
	Р
	Г
	О

2. Дослідіть роботу програми. З'ясуйте за якими адресами необхідно записати символи Р, О, Б, -,0,7.

3. Записати за адресами 83F8-83FF коди букв для виведення на дисплей повідомлення "--ЕНЕРГО--".

4. Здійснити пуск програми і перевірити правильність виведення повідомлення на дисплей.

5. Ввести в мікро-ЕОМ програму 7.8.

6. Здійснити пуск програми і перевірити, що на дисплей виводиться лише одна цифра.

Встановити в програмі час увімкненого стану цифри на кожній ко-мірці дисплея, що дорівнює 1 с (час задається підпрограмою затримки; простежити зміни в інформації, що виводиться на дисплей).

Таблиця 7.13 - Відповідність кодів сегментів семисегментного індикатора - сегментам, які світяться на семисегментному індикаторі

Код символу	Сегмент
	a
	b
	c
	d
	e
	f
	g

### Завдання 3.

Дослідження програми обслуговування клавіатури.

#### Порядок виконання завдання:

1. Ввести в мікро-ЕОМ програму 7.8.
2. Здійснити пуск програми і в покроковому режимі перевірити вміст регістра А після кожного натискання на клавішу.
3. Дослідіть роботу програми 7.8.

### Робота на навчальній ЕОМ

В навчальній мікро-ЕОМ дисплей і клавіатура під'єднані до шин даних, керування, адрес. Регістр сканування RгСк використовується як для сканування дисплея, так і для сканування клавіатури. Програми 7.1, 7.2 можуть бути виконані без будь-яких змін на навчальній мікро-ЕОМ без під'єднання додаткового дисплею.

В ПЗП мікро-ЕОМ записані програми, що дозволяють визначити код натиснутої клавіші і вивести на дисплей повідомлення. Дешифрування в семисегментний код здійснюється за допомогою спеціальної табл. 7.3 шляхом додавання відносної адреси символу до початкової адреси таблиці.

Табл. 7.3 побудована так, що при дешифруванні шістнадцять цифр ніяких додаткових перетворень не потребують, тому що значення цифри є її відносна адреса. Для перезапису повідомлення в область 83F0-

83F5 застосовується підпрограма, яка викликається командою RST 3. Вхідними параметрами її є адреса першого байта повідомлення, розташованого в парі регістрів D, E, і повідомлення, записане з цієї адреси.

Таблиця 7.14 - Текст програми 7.9, яка виводить на індикатори надпис "РОБ- -07-"

Адреса	Вміст	Мітка	Команда	Коментар
			.org 8000h	
			mvi a, 73h	
			sta 83f8h	
			mvi a, 3fh	
			sta 83f9h	
			mvi a, 7dh	
			sta 83fah	
			mvi a, 40h	
			sta 83fbh	
			mvi a, 40h	
			sta 83fch	
			mvi a, 3fh	
			sta 83fdh	
			mvi a, 07h	
			sta 83feh	
			mvi a, 40h	
			sta 83ffh	
			hlt	
			.end	

### Вимоги до оформлення звіту

Звіт повинен містити:

1. Схему під'єднання досліджуваної клавіатури до мікро-ЕОМ.

2. Програми, розроблені в п. 5, 6, 7 завдання для домашньої підготовки.

3. Таблицю відповідності кодів чисел, записаних в реєстр сегментів дисплея, сегментам, які світяться (таблиця заповнюється при виконанні завдання 1).

4. Програми, розроблені при виконання завдань 5 і 6 в розділі "Робота на навчальній мікро-ЕОМ". Блок схеми алгоритмів програм. Тексти програм на асемблері та в кодах мікролабораторії з поясненнями виконуваних дій.

### 5. Завдання для самоперевірки

1. Які коди необхідно записати за адресами 83f8-83ff для виведення на дисплей чисел 1, 2, 3, 4, 5, 6, 7, 8?

2. Як потрібно змінити програму 7.4, щоб зображення на дисплеї почало рівномірно пересуватися?

3. Як потрібно змінити програму 7.5 для зміни напрямку сканування індикатора дисплея?

4. Яка з клавіш буде визначатися натиснутою за допомогою програми 7.8, якщо на клавіатурі будуть одночасно натиснуті: а) клавіші 3 і 5; б) клавіші 2 і 8?

5. Змінити вигляд програми 7.4 так, щоб мікро-ЕОМ визначала номер натиснутої клавіші в клавіатурі, організованої у вигляді матриці 3×3.

6. Що вам відомо про індикаторні пристрої виведення?

7. Що вам відомо про під'єднання клавіатури до мікро-ЕОМ?

8. Наведіть спрощену схему клавіатури і опишіть її роботу.

9. Наведіть приклади програмування виведення даних на індикатори.

10. Опишіть периферійні пристрої мікро-ЕОМ.

11. Що вам відомо про програмований периферійний інтерфейс?

## ЛАБОРАТОРНА РОБОТА 8

### ДОСЛІДЖУВАННЯ ОСЦИЛОГРАМ СИГНАЛІВ В МІКРО-ЕОМ

**Мета роботи:** дослідження часових діаграм процесів передачі інформації в мікро-ЕОМ.

#### Короткі відомості з теорії

Процеси отримання, перетворення і передачі інформації в мікро-ЕОМ в часі тактуються синхросигналами  $\Phi_1$  і  $\Phi_2$ , які надходять на входи МП ВІС [5]. Період синхросигналів  $\Phi_1$  і  $\Phi_2$  називається *машинним тактом*. При аналізі часових співвідношень (при обміні інформацією в мікро-ЕОМ) використовують також поняття *машинний цикл* і *час виконання команди*.

При виконанні програми на всіх магістралях мікро-ЕОМ процеси, як правило, не є періодичними, що ускладнює їх дослідження за допомогою простих технічних засобів. Найбільш простим і зручним технічним засобом є осцилограф [8]. Специфіка вивчення часових діаграм передачі інформації в мікро-ЕОМ полягає в необхідності отримання стійкої картини на екрані осцилографа, що можливо лише для періодичних сигналів. Всі експерименти, наведені в завданнях лабораторної роботи, дозволяють проводити дослідження часових діаграм роботи як МП ВІС, так і сигналів на всіх магістралях мікро-ЕОМ. Приведена діаграма в завданні 3, по суті, дозволяє дослідити процес виконання мікро-ЕОМ будь-якої команди. Для дослідження процесів необхідний лише двопроменевий осцилограф. В подальшому входи осцилографа будуть називатися входи А і В. Отримання стійкої картини досліджуваних процесів на екрані осцилографа досягається раціональним вибором сигналу, який синхронізує запуск розгортки осцилографа [6,7]. У всіх експериментах цей сигнал – періодичний.

#### **Завдання для домашньої підготовки**

1. Ознайомтеся з часовими діаграмами команд JMP<A<sub>2</sub>> <A<sub>1</sub>>; IN<A<sub>1</sub>>; OUT <A<sub>1</sub>>; MOV M,A; PUSH B, POP D [8].
2. Розгляньте типи машинних циклів для МП ВІС KP580BM80A.
3. Вивчіть схему запису слова стану МП ВІС. Визначіть, в якій момент часу слово стану МП ВІС записується в регістр слова стану МП ВІС записується в регістр слова стану.
4. Ознайомтеся з вимогами до параметрів синхросигналів  $\Phi_1$  і  $\Phi_2$  МП ВІС.
5. Розгляньте стан магістралей мікро-ЕОМ при роботі МП ВІС в режимах ОЧКУВАННЯ, ЗУПИНКА, ЗАХОПЛЕННЯ.
6. Ознайомтеся з вмістом розрядів слова стану МП ВІС при виконанні різних машинних циклів.

#### **Завдання до лабораторної роботи**

**Завдання 1.** Дослідити параметри  $\Phi_1$  і  $\Phi_2$ .

##### **Порядок виконання завдання:**

1. Подати на входи осцилографа А і В синхросигнали  $\Phi_1$  і  $\Phi_2$ .
2. Виміряти для кожного імпульсу такі параметри: тривалість сигналу, тривалості фронтів, відстань між сигналами, рівні сигналів логічних "1" і "0".

**Завдання 2.** Дослідити часові діаграми виконання команди JMP<A<sub>2</sub>> <A<sub>1</sub>>.

##### **Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ просту програму 8.1.

Таблиця 8.1 – Текст програми 8.1

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	C3 0080	HERE	JPM HERE	Перейти на себе

2. Здійснити пуск програми.

3. Подати на вхід А осцилографа сигнал з розряду  $D_5$  регістра слова стану МП ВІС. Одиничний сигнал на цьому виході буде з'являтися на початку кожного машинного такту отримання коду команди з пам'яті (такт  $\Phi_1$ ).

4. Здійснити синхронізацію осцилографа від цього сигналу. Визначити кількість тактів і час, необхідний для виконання команди JPM HERE. Нарисувати осцилограму.

5. Під'єднати вхід В осцилографа до вихода "Синхр" МП ВІС. Визначити кількість і тривалість сигналів на цьому виході. Нарисувати осцилограму.

6. Під'єднати вхід В осцилографа до виходу "Прийм" МП ВІС. Визначити кількість і тривалість сигналів на цьому виході МП ВІС при виконанні команди JPM HERE. Нарисувати осцилограму.

7. Визначити за допомогою осцилографа стан розрядів регістра слова стану МП ВІС при виконанні машинного циклу отримання команди (цикл  $M_1$ ) і циклу читання даних з пам'яті. Результати занесіть в таблицю 8.2. Знайдіть відмінності у визначенні МП ВІС цих двох машинних циклів.

Таблиця 8.2 – Результати дослідження стану розрядів регістра слова

Цикли	Входи тактових сигналів		Вихід синхронізація	Магістраль даних			Регістр слова стану
	$\Phi_1$	$\Phi_2$	C	$D_0$	...	$D_7$	PrCC
$M_1$							
...							

8. Під'єднати вхід В осцилографа до однієї з підмагістралей даних мікро-ЕОМ. Визначити, скільки раз змінюється інформація на ній при виконанні команди JPM HERE. Пояснити, чим викликана кожна зміна даних на цій підмагістралі. Нарисувати осцилограму.

9. Під'єднати вхід В осцилографа до підмагістралі  $A_0$  (магістралі адреси мікро-ЕОМ). Визначити, скільки раз змінюється інформація на ній при виконанні команди JPM HERE. Нарисувати осцилограму.

10. Під'єднати вхід В осцилографа до виходу  $\overline{ZP}$  МП ВІС. Нарисувати осцилограму сигналу на цьому виході при виконанні команди JPM HERE.

**Завдання 3.** Дослідити осцилограми процесів в мікро-ЕОМ при виконанні різних команд.

**Порядок виконання завдання:**

1. Ввести в мікро-ЕОМ підпрограму 8.3.
2. Здійснити пуск програми.
3. Під'єднати вхід А осцилографа до виходу D<sub>6</sub> регістра слова стану МП ВІС. Задати синхронізацію осцилографа від сигналу, що надходить на вхід А. Під'єднати вхід В осцилографа до виходу D<sub>5</sub> регістра слова стану МП ВІС. На цьому виході МП ВІС записує одиничний сигнал на початку кожного машинного такту Ф<sub>1</sub>. Впевнитись, що при виконанні програми є два цикла M<sub>1</sub>. Визначити тривалість виконання команди IN 250. Визначити, скільки машинних тактів займає виконання всієї програми. Нарисувати осцилограму сигналу на виході D<sub>5</sub> регістра слова стану МП ВІС.

Таблиця 8.2 – Текст підпрограми 8.3.

Адреса	Машинний код	Мітка	Мнемокод	Коментар
8000	E1		MVI A,81	Програмування інтерфейса
8001	81			
8002	D3		OUT 251	KP580BB55A
8003	FB			
8004	DB	STRT	IN 250	Записати число з вхідного пристрою (з перемикачів)
8005	FA			
8006	C3		JMP STRT	Перейти на STRT
8007	04			
8008	80			

4. Визначити за допомогою осцилографа стан розрядів регістра слова стану МП ВІС при виконанні машинного циклу читання даних з зовнішнього пристрою. Результати занести в таблицю, складену при виконанні п. 7 завдання 2.

5. Ввести в мікро-ЕОМ програму 8.3.

Таблиця 8.3 – Текст програми 8.3

Адреса	Машинний код	Мітка	Мнемокод	Коментар
1	2	3	4	5
8000	E1		MVI A,81	Програмування інтерфейса
8001	81			
8002	D3		OUT 251	KP580BB55A
8003	FB			
8004	DB	STRT	IN 250	Записати число з вхідного пристрою (з перемикачів)
8005	FA			
8006	D3		OUT F9	Запис даних в порт виведення
8007	F9			

Продовження таблиці 8.3

1	2	3	4	5
8008	C3		JMP STRT	Перейти на STRT
8009	04			
800A	80			

6. Здійснити пуск програми. Залишаючи синхронізацію осцилографа по входу А від сигналу з виходу D<sub>6</sub> регістра слова стану МП ВІС, дослідити часові діаграми виконання мікро-ЕОМ команди OUT F9. Звернути увагу на момент появи і тривалість сигналу на виході  $\overline{CI}$  МП ВІС. Дослідити за допомогою осцилографа стан розрядів регістра слова стану МП ВІС при виконанні команди запису числа у зовнішній пристрій. Результати занести в таблицю, складену при виконанні п. 7 завдання 2. Нарисувати осцилограми процесу виконання мікро-ЕОМ команди OUT F9.

7. Змінити в програмі п. 5 команду запису числа у вихідний пристрій мікро-ЕОМ на будь-яку команду запису числа в пам'ять (наприклад, MOV M, A; MOV M, B тощо). Ввести програму в мікро-ЕОМ і дослідити осцилограми процесу виконання запису числа в пам'ять. Нарисувати осцилограми. Дослідити за допомогою осцилографа стан розрядів регістра слова стану МП ВІС. Результати занести в таблицю, складену при виконанні п. 7 завдання 2.

8. Змінити в програмі п. 5 команду OUT F9 на одну з команд роботи зі стеком (наприклад, PUSH B, POP B тощо). Дослідити за допомогою осцилографа часові діаграми виконання цих команд. Нарисувати осцилограми. Дослідити за допомогою осцилографа стан розрядів регістра слова стану МП ВІС. Результати занести в таблицю, складену при виконанні п. 7 завдання 2.

### Робота на навчальній ЕОМ

Всі завдання, наведені в лабораторній роботі, можуть бути виконані на навчальній мікро-ЕОМ.

#### Зміст звіту

Звіт повинен містити: осцилограми, зняті при виконання пунктів завдань 1, 2, 3, 2, таблицю вмісту розрядів регістра стану МП ВІС при виконанні різних машинних циклів.

#### Завдання для самоперевірки

1. На якому такті слова стану МП ВІС записується в регістр слова стану МП ВІС?
2. Скільки можливих варіантів слова стану існує для МП ВІС KP580BM80A?
3. Наведіть схему запису слова стану МП ВІС в регістр слова стану.
4. Які сигнали формуються на магистралі керування мікро-ЕОМ?

5. В чому відмінність в слові стану, що видає МП ВІС на машинних циклах зчитування даних із зовнішнього пристрою і пам'яті, запису даних у зовнішній пристрій і пам'ять?

6. Нарисуйте часові діаграми виконання мікро-ЕОМ таких команд:  $IN <A_1>$ ,  $OUT <A_1>$ ,  $JMP <A_2> <A_1>$ ,  $MOV M, A$ ,  $POSH PSW$ ,  $POP PSW$ ,  $HLT$ ,  $NOP$ .

7. Нарисуйте часові діаграми для синхросигналів  $\Phi_1$  і  $\Phi_2$  МП ВІС.

8. Вкажіть стан магістралей мікро-ЕОМ при роботі МП ВІС в режимах ОЧІКУВАННЯ, ЗУПИНКА, ЗАХОПЛЕННЯ.

9. Вкажіть, яким сигналом, що формує МП ВІС, здійснюється в мікро-ЕОМ керування двонаправленим шинним формувачем.

10. Вкажіть, для чого в мікро-ЕОМ застосовуються режими ОЧІКУВАННЯ, ЗУПИНКА, ЗАХОПЛЕННЯ.

### ЛАБОРАТОРНА РОБОТА №9

### ВИКОРИСТАННЯ АНАЛОГО-ЦИФРОВИХ ПЕРЕТВОРЮВАЧІВ ДЛЯ ВИМІРЮВАННЯ АНАЛОГОВИХ СИГНАЛІВ ТА ОБРОБКИ ЇХ ЗА ДОПОМОГОЮ ЕОМ

**Мета роботи:** Ознайомитись із видами і основними характеристиками аналого-цифрових перетворювачів та навчитись використовувати їх для вимірювання аналогових сигналів.

#### Короткі відомості з теорії

Структурна схема мікропроцесорного вольтметра показана на рисунку 9.1. Використання МП дозволило сумістити в одному пристрої функції деяких вимірювальних приладів: цифрового вольтметра постійного струму; автоматичного потенціометра; диференціального компаратора напруг; вимірювача відношень напруг; вимірювача потужності; вимірювача нестабільності. Поєднання автоматичної корекції похибок з високोलінійним цифроаналоговим перетворювачем, який виконаний на основі індуктивних дільників напруги, дозволило досягти високих метрологічних характеристик. В приладі реалізований компенсаційний метод перетворення. Мікропроцесор використовується як для виконання функції керування і обробки вимірювальної інформації, так і для сервісної функції.

Режими роботи приладу задаються або оператором, або дистанційно від зовнішніх пристроїв через інтерфейс. Мікропроцесор переводить прилад у відповідний режим роботи, розшифрувавши стан органів керування. Органи керування мікропроцесорного вольтметра виконані у вигляді кнопочких перемикачів без фіксації стану. Натискання кнопки фіксується програмним способом. Під час розробки приладу було розглянуто декілька можливих варіантів розшифровки стану органів керування програмним шляхом. В першому варіанті органи керування були

пов'язані з додатковим регістром, в якому фіксувалось натискання будь-якої кнопки. На початку кожного такту вимірювання МП зчитував стан регістра і переходив в подальшому до виконання відповідної підпрограми. В другому варіанті МП, працюючи в циклічному режимі періодично "опитує" стан кнопок керування, при виявленні натиснутої кнопки розшифровує даний стан і переходить на відповідну підпрограму. Такий варіант прийнятний при швидкодії приладу більше десятків вимірювань за одну секунду. При малій швидкодії довелось би по декілька разів, протягом одного такту вимірювання, аналізувати стан органів керування. В третьому варіанті мікропроцесорний блок організований таким чином, що будь-яке натискання кнопки діє як затримка. При цьому МП затримує виконання основної програми і переходить до підпрограми аналізу стану органів керування. Прилад, який розглядається, може функціонувати з різною швидкодією. Для підвищення стійкості до помилки, використовується перший варіант розшифровки станів органів керування.

В залежності від вибраної швидкодії (час вимірювання встановлюється 0,1; 1 і 10 с) в приладі передбачено забезпечення чутливості, що відповідає 4,5; 6,5 і 7,5 десятковим розрядом на границях 100 мВ; 1 В; 10 В; 100 В, при цьому вимірювання здійснюється або в два такти, або в три такти, або в три такти з усередненням.

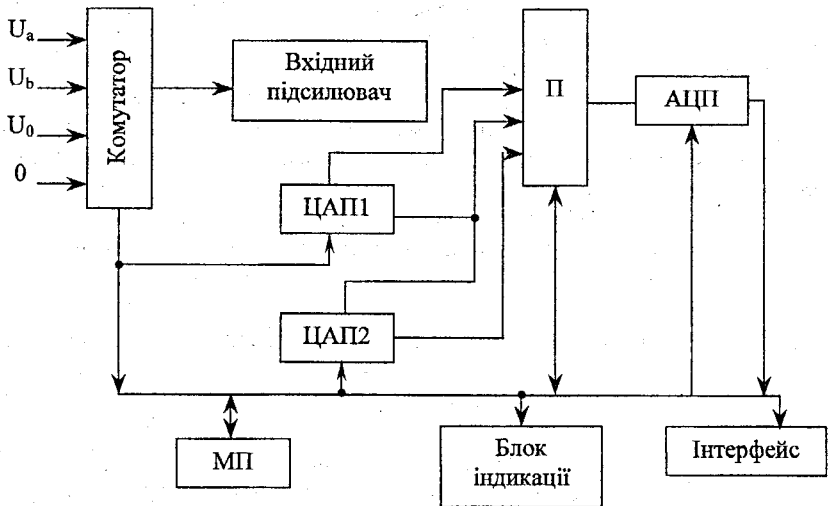


Рисунок 9.1 – Схема мікропроцесорного вольтметра

Існує багато різних способів перетворення аналогового сигналу (електричної напруги або струму), який змінюється неперервно, у потік цифрових даних, придатних для подальшої їх обробки мікропроцесором.

Пристрої, що перетворюють миттєве значення аналогового сигналу у цифровий код, називають аналого-цифровими перетворювачами (АЦП). Якщо для обробки на ЕОМ необхідно отримати не миттєве, а неперервне значення вхідного сигналу, то здійснюють серію вимірювань миттєвих значень із наперед заданою частотою вимірювання, а потім згладжують отримані результати за допомогою математичних методів. Нижче описані основні характеристики АЦП, за якими здійснюють їх вибір у відповідності до задачі, що розв'язується.

1. Опорна напруга.

2. Розрядність. Під розрядністю АЦП розуміють кількість бітів, за допомогою яких представляється цифрове значення вимірюючого сигналу. Чим більша розрядність АЦП, тим точніше результуючий цифровий код буде відповідати реальному миттєвому значенню вхідного сигналу. Розрядність АЦП показує, на скільки ступенів від 0 до опорної напруги буде розбито вхідний сигнал. Наприклад, якщо АЦП має 10-ти бітову розрядність, то це означає, що вхідна напруга в результаті буде представлена одним із  $2^{10}=1024$  ступенів, на які розбивається опорна напруга. Якщо опорна напруга складає 2,56 В, то мінімальна зміна вхідного сигналу, яку відобразить АЦП на своїх виходах складе  $2,56/1024=0,0025$  В.

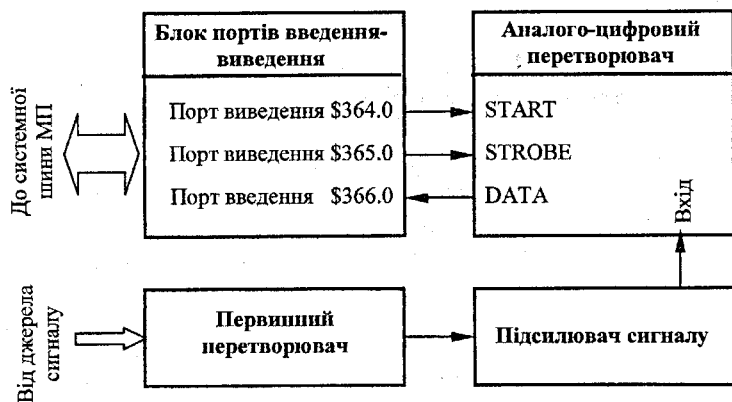


Рисунок 9.2 – Структурна схема лабораторного стенду для вимірювання аналогових сигналів за допомогою АЦП

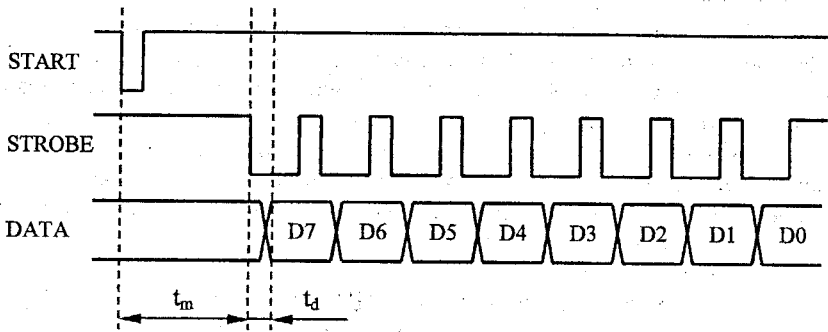


Рисунок 9.3 – Часова діаграма роботи АЦП

У наш час широко використовуються аналого-цифрові перетворювачі (АЦП) послідовного наближення.

**Принцип дії АЦП послідовного наближення.** Напруга, яка вимірюється, надходить на один із входів компаратора. На інший вхід компаратора надходить напруга від цифро-аналогового перетворювача, яким керує лічильник послідовних наближень. Ця напруга змінює своє значення.

#### Завдання для домашньої підготовки

1. Ознайомитись з теоретичними відомостями даної лабораторної роботи.
2. Використовуючи мову програмування Паскаль скласти програму, яка у циклі опитує аналого-цифровий перетворювач та відображає вимірне значення постійної напруги джерела блока керування на екрані навчальної ЕОМ. Вихід з програми організувати натисканням клавіші "ESC".
4. Використовуючи мову програмування Паскаль скласти програму, яка протягом 3 секунд опитує аналого-цифровий перетворювач та виміряні значення змінної напруги джерела блока керування записує у масив. Після закінчення часу вимірювання дані з масиву записати у файл. Для визначення тривалості вимірювання використати системний годинник ЕОМ.

#### Завдання до лабораторної роботи

**Завдання.** Навчитись використовувати аналого-цифрові перетворювачі та ПЕОМ для вимірювання аналогових сигналів.

#### Порядок виконання завдання:

1. Ознайомитись з навчальною ПЕОМ та лабораторним стендом.

2. Увімкнути живлення навчальної ПЕОМ та дочекатись завантаження операційної системи.

3. Завантажити програмну оболонку компілятора мови програмування Паскаль.

4. Ввести програму, яка вимірює значення постійної напруги джерела блока керування.

5. Перевести перемикач "Напруга" блока керування у положення "Постійна".

6. Запустити програму на виконання і впевнитись у її працездатності, регулюючи напругу за допомогою регулятора, розташованого на блоці керування. У крайньому лівому положенні має бути найменше значення напруги, а у крайньому правому – найбільше.

7. Змінюючи напругу від найменшої до найбільшої з кроком 0,2 В скласти таблицю відповідності між ступенями аналого-цифрового перетворювача, що відображаються на екрані ЕОМ та показами вольтметра, розташованого на блоці керування.

8. Використовуючи отриману таблицю, внести корекцію до програми таким чином, щоб на екрані ЕОМ відображалось значення напруги, аналогічне до значення вольтметра.

9. Запустити програму на виконання та при різних положеннях регулятора порівняти значення, виміряні за допомогою ЕОМ з аналого-цифровим перетворювачем і за допомогою вольтметра.

10. Ввести програму, яка протягом 3 секунд вимірює значення змінної напруги джерела блока керування.

11. Перевести перемикач "Напруга" блока керування у положення "Змінна".

12. Запустити програму на виконання і, після закінчення її роботи, проаналізувати отриманий файл із значеннями змінної напруги.

13. Прийнявши частоту змінної напруги рівною 50 Гц, визначити середню кількість вимірювань за один період змінної напруги. За отриманою кількістю вимірювань визначити фактичну швидкість отриманої вимірювальної системи.

14. Вимкнути живлення навчальної ЕОМ.

15. Зробити висновки з лабораторної роботи та оформити звіт.

### **Зміст звіту**

1. Короткий опис аналого-цифрових перетворювачів та структурну схему їх під'єднання до ЕОМ або мікропроцесора.

2. Часову діаграму обміну інформацією між аналого-цифровим перетворювачем та навчальною ЕОМ.

3. Робочі програми.

### **Завдання для самоперевірки**

1. Розкажіть про призначення аналого-цифрових перетворювачів.
2. Опишіть основні види аналого-цифрових перетворювачів та коротко охарактеризуйте переваги та недоліки кожного виду.
3. Перерахуйте та поясніть основні параметри, за якими виконують вибір аналого-цифрового перетворювача.
4. Нарисуйте та поясніть часову діаграму обміну інформацією між аналого-цифровим перетворювачем та навчальною ЕОМ.
5. Розкажіть, у чому полягає відмінність вимірювання постійної та змінної напруги за допомогою аналого-цифрових перетворювачів.
6. Поясніть призначення пристрою вибірки-зберігання.
7. Охарактеризуйте види похибок, які виникають при використанні аналого-цифрових перетворювачів.
8. Опишіть способи приведення вхідної напруги або струму первинного кола до рівня, що входить у діапазон вимірювання аналого-цифрового перетворювача.

### **ЛАБОРАТОРНА РОБОТА №10**

#### **ВИКОРИСТАННЯ ПОРТІВ ВВЕДЕННЯ-ВИВЕДЕННЯ ПЕРСОНАЛЬНОЇ ЕОМ ДЛЯ КОНТРОЛЮ ДИСКРЕТНИХ СИГНАЛІВ ЗОВНІШНІХ ПРИСТРОЇВ ТА КЕРУВАННЯ ЗОВНІШНІМИ ПРИСТРОЯМИ**

**Мета роботи:** ознайомитись з апаратними та програмними засобами ЕОМ для керування зовнішніми виконуваними пристроями та для вимірювання і обробки дискретних сигналів.

### **Завдання для домашньої підготовки**

1. Ознайомитись з теоретичними відомостями даної лабораторної роботи.
2. Використовуючи мову програмування Паскаль скласти програму, яка робить запит про стан дискретного порту введення, та виводить на екран стан контактів, під'єднаних до дискретних входів у вигляді: "Контакт X замкнутий", або "Контакт X розімкнутий" відповідно до виміряного значення.
3. Використовуючи мову програмування Паскаль скласти програму, яка звертається до клавіатури ЕОМ і в залежності від натиснутих клавіш вмикає або вимикає світлодіоди, під'єднані до дискретних портів виведення. Клавіша "1" повинна керувати червоним світлодіодом, а клавіша "2" – зеленим.
4. Використовуючи дві попередні програми скласти програму, яка у циклі робить запит про стан дискретного порту введення, до якого під'єднані контакти високовольтного вимикача, відображає на екрані стан

контактів, і за допомогою клавіатури ЕОМ вмикає та вимикає вимикач. Клавіша "1" повинна вмикати вимикач, а клавіша "0" – вимикати. Тривалості керуючих імпульсів вмикання та вимикання прийняти рівними 50 мс.

### **Завдання до лабораторної роботи**

Навчитись використовувати порти введення-виведення ПЕОМ для контролю дискретних сигналів зовнішніх пристроїв та для керування зовнішніми пристроями.

#### **Порядок виконання завдання:**

1. Ознайомитись з навчальною ЕОМ та лабораторним стендом.
2. Увімкнути живлення навчальної ЕОМ та дочекатись завантаження операційної системи.
3. Завантажити програмну оболонку компілятора мови програмування Паскаль.
4. Ввести програму, яка визначає стан контактів вимикача.
5. Перевести перемикач лабораторного стенда у положення "масляний".
6. Увімкнути живлення лабораторного стенда.
7. Запустити програму на виконання і перевірити її працездатність, виконуючи операції ввімкнення та вимкнення вимикача за допомогою кнопок керування лабораторного стенда.
8. Вимкнути живлення лабораторного стенда.
9. Ввести програму, яка керує світлодіодами блока керування.
10. Запустити програму на виконання і перевірити її працездатність, контролюючи стап світлодіодів блока керування.
11. Ввести програму контролю та керування вимикача.
12. Запустити програму на виконання і, контролюючи світлодіоди блока керування, перевірити правильність виведення керуючих імпульсів. А саме: світлодіоди керуючих портів ввімкнення та вимкнення повинні засвілюватись лише короткочасно, в момент натискання на відповідну клавішу клавіатури ЕОМ; світлодіоди керуючих портів ввімкнення та вимкнення не повинні світитись одночасно.
13. Увімкнути живлення лабораторного стенда.
14. Перевірити працездатність програми, виконуючи операції ввімкнення і вимкнення високовольтного вимикача за допомогою клавіатури ЕОМ. Правильність виконання відповідної операції контролювати за зміною стану контактів вимикача на екрані ЕОМ та за реальним станом контактів на досліджуваному вимикачі.
15. Вимкнути живлення лабораторного стенда.
16. Вимкнути живлення навчальної ЕОМ.
17. Зробити висновки з лабораторної роботи та оформити звіт.

### **Зміст звіту**

1. Навести структурну схему лабораторного стенда.
2. Часові діаграми запису у порт виведення та зчитування із порту введення по системній шині PC-104 (ISA).
3. Робочі програми.

### **Завдання для самоперевірки**

1. Назвіть системні шини EOM з архітектурою IBM та коротко охарактеризуйте їх призначення.
2. Опишіть методи під'єднання зовнішніх пристроїв до EOM.
3. Побудуйте спрощену часову діаграму доступу процесора до порту введення або виведення через немультіплексовану шину (на прикладі системної шини ISA).
4. Назвіть стандартні порти введення та виведення, що входять до типової конфігурації сучасної персональної EOM.
5. Опишіть способи під'єднання силового електроенергетичного обладнання до портів персональної EOM.

### **ЛАБОРАТОРНА РОБОТА №11**

#### **ВИКОРИСТАННЯ ПРОГРАМОВАНОГО ТАЙМЕРА-ЛІЧИЛЬНИКА ТА СХЕМИ ПЕРЕРИВАНЬ РОБОТИ ПРОЦЕСОРА ДЛЯ ВИКОНАННЯ ОПЕРАЦІЙ, ЯКІ ЗАЛЕЖАТЬ ВІД ЧАСУ**

**Мета роботи:** Ознайомитись із програмними та апаратними засобами персональної EOM для виконання вимірювань і керуючих дій із заданими інтервалами часу.

### **Завдання для домашньої підготовки**

1. Ознайомитись з теоретичними відомостями даної лабораторної роботи.
2. Використовуючи мову програмування Паскаль скласти програму, яка вмикає зелений світлодіод блока керування з періодом 1 секунда. Для цього: скласти головну частину програми, яка перепрограмує системний таймер-лічильник навчальної персональної EOM на інтервал часу 1 мілісекунда, та скласти підпрограму переривання, яка один раз за кожні 500 викликів змінює значення однібітового порту виведення, до якого під'єднаний зелений світлодіод, на протилежне. Тобто, якщо на виході порту було встановлене значення "0", то встановити значення "1", а якщо було встановлене значення "1", то встановити значення "0". Вихід з програми організувати після натискання клавіші "ESC". Перед виходом із програми відновити параметри таймера-лічильника та адресу підпрограми переривання.

3. Використовуючи мову програмування Паскаль скласти програму, яка здійснює вимірювання часових характеристик високовольтного вимикача у одному із типових циклів його роботи (за вказівкою викладача). Для цього: скласти головну частину програми, яка перепрограмує системний таймер-лічильник навчальної персональної ЕОМ на інтервал часу 1 мілісекунда, та скласти підпрограму переривання, яка у кожному циклі виклику здійснює опитування дискретного порту введення, до якого під'єднані контакти вимикача, та здійснює опитування аналого-цифрового перетворювача, під'єданого до вимірювального шунта, для визначення струму, що протікає через котушки керування вимикача. Виміряні значення записувати у масив. Після виконання 1000 вимірювань необхідно закінчити вимірювання та записати виміряні значення із масивів у файл. Після цього необхідно закінчити роботу програми, попередньо відновивши параметри таймера-лічильника та адресу підпрограми переривання.

### **Завдання до лабораторної роботи**

Навчитись використовувати програмований таймер-лічильник та схему переривань роботи мікропроцесора для виконання операцій, які залежать від часу.

#### **Порядок виконання завдання:**

1. Ознайомитись з навчальною ПЕОМ та лабораторним стендом.
2. Увімкнути живлення навчальної ПЕОМ та дочекатись завантаження операційної системи.
3. Завантажити програмну оболонку компілятора мови програмування Паскаль.
4. Ввести програму з п. 2 завдання для домашньої підготовки.
5. Запустити програму на виконання, та перевірити правильність її роботи, використовуючи візуальний контроль за станом світлодіода, протягом 1 хвилини.
6. Закінчити роботу програми та встановити коректні дату та час системного годинника навчальної ПЕОМ.
7. Ввести програму, яка вимірює часові характеристики високовольтного вимикача.
8. Переконавшись у тому, що живлення стенда для дослідження вимикачів вимкнуте.
9. Запустити програму на виконання і, контролюючи світлодіоди блока керування, перевірити правильність виведення імпульсів керування. А саме: світлодіоди керуючих портів увімкнення та вимкнення повинні засвітитись лише короткочасно і не повинні світитись одночасно.
10. Увімкнути живлення стенда для дослідження вимикачів.
11. Запустити програму на виконання і, контролюючи зміну стану контактів вимикача, впевнитись у правильності виконання керуючих імпульсів.

12. Проаналізувати файл з результатами вимірювань та порівняти їх із реальними часовими діаграмами спрацювання досліджуваного вимикача.
13. Вимкнути живлення стенда для дослідження вимикачів.
14. Вимкнути живлення навчальної ПЕОМ.
15. Зробити висновки з лабораторної роботи та оформити звіт.

### **Зміст звіту**

1. Короткі теоретичні відомості щодо використання таймерів-лічильників та схеми переривань роботи процесора.
2. Часові діаграми досліджуваного вимикача.
3. Робочі програми.

### **Завдання для самоперевірки**

1. Розкажіть про призначення таймерів-лічильників.
2. Яким чином змінити часовий інтервал таймера-лічильника?
3. Розкажіть про призначення схем переривання роботи процесора.
4. Назвіть декілька пристроїв персональної ЕОМ, які викликають переривання роботи процесора.
5. Розкажіть про особливості підпрограм переривання роботи процесора.
6. Що таке вектор переривання?
7. Яким чином встановлюється вектор переривання підпрограми користувача?
8. Поясніть, яким чином виконується вимірювання точних коротко-термінових часових інтервалів за допомогою персональної ЕОМ.

## Література

1. Балашов Е.П., Пузанков Д.В. Микропроцессоры и микропроцессорные системы. – М.: Радио и связь, 1981.
2. Соучек Б. Микропроцессоры и микро-ЭВМ: Пер. с англ. – М.: Советское радио, 1979.
3. Горбунов В.Л., Панфилов Д.И., Преснухин Д.Л. Микропроцессоры. Основы построения микро-ЭОМ. – М.: Высшая школа, 1984.
4. Учебная микро-ЭОМ на базе микропроцессорного комплекта серии К580 (Панфилов Д.И., Красавин В.И., Романенко О.А. – Электронная промышленность, 1983, № 9.
5. Костинок Л.Д., Паранчук Я.С., Щур І.З. Мікропроцесорні засоби та системи: Навчальний посібник - 2-ге вид., перероб., доп. - Львів: Видавництво Національного університету "Львівська політехніка", 2002. - 200 с.
6. Микропроцессоры. В 3-х кн. Кн. 1. Архитектура и проектирование микро-ЭВМ. Организация вычислительных процессов: Учеб. для вузов/ П.В.Нестеров, В.Ф. Шаньгин, В.Л. Горбунов и др.; Под редакцией Л.Н. Преснухина. - М.: Высш. шк., 1986.- 495 с.
7. Микропроцессоры. В 3-х кн. Кн. 2. Средства сопряжения. Контролирующие и информационно-управляющие системы: Учеб. для вузов/ В.Д.Вернер, Н.В. Воробьев, А.В. Горячев и др.; Под редакцией Л.Н. Преснухина. - М.: Высш. шк., 1986.- 383 с.
8. Микропроцессоры. В 3-х кн. Кн. 3. Средства отладки, лабораторный практикум и задачник: Учеб. для вузов/ Н.В. Воробьев, В.Л. Горбунов, А.В. Горячев и др.; Под редакцией Л.Н. Преснухина. - М.: Высш. шк., 1986.- 351 с.
9. Токхайм Р. Микропроцессоры: Курс и упражнения/Пер. с англ., под ред. В.Н. Грасевича. - М.: Энергоатомиздат, 1987. - 336 с.
10. Микролаб КР580ИК80 907. Техническое описание и инструкция по эксплуатации 3.031.068 ТО. - М.: Внешторгиздат, 1988. – 44 с.
11. Микропроцессорная лаборатория "Микролаб КР580ИК80". - М.: Внешторгиздат, 1988. – 127 с.

*Навчальне видання*

Олександр Євгенійович Рубаненко, Константин Іванович Кравцов,  
В'ячеслав Олександрович Комар

## **МІКРОПРОЦЕСОРНИЙ ЗАХИСТ**

Лабораторний практикум

Оригінал-макет підготовлено Рубаненком О.Є.

Редактор В.О. Дружиніна  
Коректор З.В. Поліщук

Навчально-методичний відділ ВНТУ  
Свідоцтво Держкомінформу України  
серія ДК № 746 від 25.12.2001  
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ

Підписано до друку 8.11.2005 р.

Формат 29,7x42  $\frac{1}{2}$

Друк різнографічний

Наклад 75 прим.

Зам № 2005-186

Гарнітура Times New Roman

Папір офсетний

Ум. друк. арк. 4.62

Віддруковано в комп'ютерному інформаційно-видавничому центрі  
Вінницького національного технічного університету  
Свідоцтво Держкомінформу України  
серія ДК № 746 від 25.12.2001 р.  
21021, м. Вінниця, Хмельницьке шосе, 95, ВНТУ