

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

МЕТОДИЧНІ ВКАЗІВКИ

з виконання самостійних робіт дисципліни

«ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ»

для студентів спеціальності
141 – Електроенергетика, електротехніка та
електромеханіка
освітньої програми «Електротехнічні системи
електроспоживання»
усіх форм навчання

Методичні вказівки з виконання самостійних робіт дисципліни «Основи мікропроцесорної техніки» для студентів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка освітньої програми «Електротехнічні системи електроспоживання» усіх форм навчання. /Укл: О.С. Назарова - Запоріжжя: НУ «Запорізька політехніка», 2020. – 26 с.

Укладач:

О.С. Назарова, к.т.н., доцент

Рецензент:

А.В. Пирожок, к.т.н., доцент

Відповідальний за випуск: П.Г. Засипко, завідувач лабораторії

Затверджено
на засіданні кафедри
Електропривода і автоматизації
промислових установок
протокол № 12 від 09.07.2019 р.

Рекомендовано
до видання НМК ЕТФ
протокол № 04 від 21.11.2019 р.

ЗМІСТ

Передмова	4
1. Основні питання з дисципліни «Основи мікропроцесорної техніки».....	5
2. Завдання №1	
Вивчення арифметичних команд та команд пересилання даних	6
3. Завдання №2	
Розроблення програм з підпрограмами	9
4. Завдання №3	
Розроблення програм з розгалудженими структурами.....	11
5. Завдання №4	
Перелік посилань.....	13
Додаток А Перелік команд мікроконтролера Intel 8051.....	15
Додаток Б Представлення даних у двійковій, шістнадцятковій та десятковій системі числення	26

ПЕРЕДМОВА

Методичні вказівки містять основні питання та опис трьох практичних завдань для самостійного виконання з дисципліни «Основи мікропроцесорної техніки» у відповідності до навчальних планів ОКР бакалаврів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка, рекомендації до їх виконання і мають два додатки.

У додатках подано перелік команд мікроконтролера Intel 8051, який включає назву, мнемокод, опис дій, що виконуються, та таблиця представлення даних у двійковій, шістнадцятковій та десятковій системі числення.

Для студентів спеціальності 141 – Електроенергетика, електротехніка та електромеханіка освітньої програми «Електротехнічні системи електроспоживання» денної форми навчання.

1. ОСНОВНІ ПИТАННЯ З ДИСЦИПЛІНИ «ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ»

1. Загальні поняття і структура мікроконтролерних систем керування електромеханічними системами.
2. Дискретні або цифрові системи керування.
3. Особливості представлення даних у двійковій, шістнадцятковій та десятковій системі числення.
4. Основні поняття булевої алгебри.
5. Структура і основні функціональні можливості базової моделі MCS-51 (МК-51).
 6. Методи адресації в МК-51.
 - 6.1 Пряма адресація.
 - 6.2 Безпосередня адресація.
 - 6.3 Регістрова адресація.
 - 6.4 Побітова адресація.
 - 6.5 Непряма адресація.
 7. Система команд МК-51.
 - 7.1 Арифметичні команди.
 - 7.2 Команди пересилання даних.
 - 7.3 Логічні команди.
 - 7.3.1 Логічне «І»: умовні позначення, таблиця істинності, особливості синтаксису команд, апаратна реалізація.
 - 7.3.2 Логічне «АБО»: умовні позначення, таблиця істинності, особливості синтаксису команд, апаратна реалізація.
 - 7.3.3 Логічне «НЕ»: умовні позначення, таблиця істинності, особливості синтаксису команд, апаратна реалізація.
 - 7.3.4 Логічне «Виключне АБО»: умовні позначення, таблиця істинності, особливості синтаксису команд, апаратна реалізація.
 - 7.4 Команди роботи з бітами.
 - 7.5 Команди передачі керування.
8. Розроблення програм розгалужених структур.
9. Організація циклів.
10. Робота з масивами даних.
11. Використання підпрограм.

2. ЗАВДАННЯ №1

Вивчення арифметичних команд та команд пересилання даних.

Мета: здобути навички використання арифметичних команд та команд пересилання даних.

Короткі теоретичні відомості.

Команди пересилання даних (таблиця А.1 додатку А) забезпечують копіювання інформації з операнду-джерела в операнд-приймач. Джерелом може бути акумулятор, регістр, комірка внутрішньої пам'яті даних або константа, що безпосередньо знаходиться у команді, приймачем - акумулятор, регістр та комірка внутрішньої пам'яті даних. Допустимі всі можливі комбінації вказаних операндів, за винятком регістр-регістр.

Арифметичні команди (таблиця А.2 додатку А) забезпечують виконання операцій додавання, віднімання (у т.ч. з урахуванням переносу) над цілочисельними 8-розрядними операндами, інкрементування (збільшення на один) й декрементування (зменшення на один) вмісту регістрів і комірок внутрішньої пам'яті даних, множення й ділення 8-розрядних цілочисельних операндів.

Завдання.

1. Написати програму у відповідності з індивідуальним завданням та отримати результат у вказаній комірці.
2. Підставити у використовувані комірки задані числа.
3. Перевірити правильність написаної програми шляхом проведення розрахунків з використанням інженерного калькулятора, розписавши отримані результати по кожній дії та кінцевий результат.

Таблиця 2.1 – Дані для виконання індивідуального завдання

№	Завдання	Дані для підстановки
1.	$((20H) + (21H)) * 03H - 24H \rightarrow (24H)$	05H; 10H
2.	$0FFH - ((22H) / (23H) + 1AH) * 02H \rightarrow (21H)$	40H; 04H
3.	$((23H) - (22H)) * ((21H) + 12H) \rightarrow (20H)$	0AH; 09H; 02H
4.	$((20H) + 35H - 10H / 02H) + (21H) \rightarrow (22H)$	0ACH; 0A2H
5.	$0AFH + (20H) / 03H - (21H) * 02H \rightarrow (22H)$	09H; 0AH

Продовження таблиці 2.1

6.	$((21H)-(22H)+(23H)/02H)*02H \rightarrow (24H)$	FFH; BCH; 22H
7.	$(10H-05H)*(20H)+(21H)/02H \rightarrow (23H)$	12H; 60H
8.	$(20H)*02H+(21H)/02H-10H \rightarrow (22H)$	3FH; 0AH
9.	$1FH-(20H)*02H+(21H)/03H \rightarrow (22H)$	03H; 12H
10.	$10H+(21H)/(22H)-05H*02H \rightarrow (23H)$	50H; 02H
11.	$(21H)*05H/02H+(22H)-01H \rightarrow (23H)$	10H; 0BDH
12.	$0FFH-((21H)*04H+(22H)/05H) \rightarrow (23H)$	02H; 25H
13.	$(21H)+(0ABH-2AH*02H)/(22H) \rightarrow (23H)$	0FFH; 02H
14.	$(23H-22H)*(20H)+(21H)/(22H) \rightarrow (24H)$	1AH; 08H; 02H
15.	$01H+((21H)/(20H)-(22H)*02H) \rightarrow (23H)$	CBH; 02H; 10H
16.	$(21H)-((18H/(22H)*(23H))+0FH) \rightarrow (24H)$	03H; 02H; 5AH
17.	$20H-05H+(24H)*02H/(25H) \rightarrow (26H)$	04H; 08H
18.	$(23H)-(24H)*02H+(25H)/05H \rightarrow (27H)$	DDH; 13H; 0FH
19.	$0DDH+(25H)*10H-(26H)/03H \rightarrow (27H)$	01H; 03H
20.	$((23H)*(24H)-10H+(25H))/02H \rightarrow (26H)$	02H; 11H; 0BH
21.	$FFH-(25H)*02H/05H+1BH+(22H) \rightarrow (23H)$	0BH; 1CH
22.	$(45H-42H)/(22H)+(23H)*08H-05H \rightarrow (24H)$	03H; 01H
23.	$(21H)-((22H)*10H+1AH/02H) \rightarrow (23H)$	0FFH; 01H
24.	$((25H)-(26H))*02H-09H/03H+1EH \rightarrow (27H)$	50H; 40H
25.	$(0CCH-(21H))*03H+(22H)/02H \rightarrow (23H)$	0B2H; 04H
26.	$11H+(21H)*(22H)/02H-08H \rightarrow (23H)$	0AH; 03H
27.	$(21H)+(22H)/02H*10H-(23H) \rightarrow (24H)$	04H; CDH; 22H
28.	$(28H-(26H))*03H-(23H)/02H+01H \rightarrow (25H)$	26H; 08H
29.	$(21H)-(22H)+10H*02H/04H \rightarrow (23H)$	7BH; 0BH
30.	$(10H-08H)/02H+(21H)*(22H) \rightarrow (23H)$	07H; 03H

Дані для перевірки підставляються послідовно у відповідні комірки внутрішньої пам'яті даних згідно індивідуального завдання.

Приклад:

$(20H) + 12H - (21H) / 02H \rightarrow (22H)$

ORG 0H

MOV A, 21H ;(21H)→(A)

MOV B, #02H ;02H→(B)

```

DIV AB      ;(A)/(B) →(A).(B)
MOV R1,A    ;(A) →(R1)
MOV A, #12H ;12H→(A)
CLR C       ;(C)=0
SUBB A, R1   ;(A)-(R1)-(C) →(A)
ADD A, 20H   ;(A)+(20H) →(A)
MOV 22H, A   ;(A) →(22H)
L1: JMP L1
END

```

Дані для перевірки слід підставляти у комірки (20H), (21H) і виконувати дії на калькуляторі з урахування пріоритетів виконання дій.

Зміст звіту з самостійної роботи.

Звіт має містити титульний лист, тему, мету лабораторної роботи, умову індивідуального завдання, текст програми з коментарями до кожної команди (групи команд), які реалізують виконання однієї з умовних частин всієї програми, висновки.

Контрольні запитання.

1. Поняття операнда. Види операндів.
2. Реєстровий метод адресації. Навести приклади команд.
3. Прямий метод адресації. Навести приклади команд.
4. Безпосередній метод адресації. Навести приклади команд.
5. Назвати та пояснити команди арифметичних операцій.
6. Назвати та пояснити команди пересилання даних.

3. ЗАВДАННЯ №2

Розроблення програм з підпрограмами.

Мета. Здобуття навичок використання команд роботи з підпрограмами.

Короткі теоретичні відомості.

Підпрограма (з англ. subroutine) – названа або іншим способом ідентифікована частина програми, яка містить опис визначеного набору дій. Підпрограма може бути багаторазово викликана з різних частин програми. Підпрограми дозволяють структурувати програму, полегшуючи тим самим її написання та відлагодження.

Завдання.

Написати програму для формування послідовності коротких та довгих імпульсів на вказаному виході порта 0 при низькому рівні сигналу вказаного біта порта 2 згідно індивідуального завдання лабораторної роботи №4.

При розробці програми необхідно використати команди роботи з підпрограмами.

Таблиця 3.1 - Дані для виконання індивідуального завдання

№	Біт P2	Біт P0	Умовне позначення послідовності імпульсів
1.	P2.0	P0.5	* *_ _* *_ _ _ _
2.	P2.1	P0.4	* *_ _ _* *_ _ _
3.	P2.2	P0.3	* *_ _ _ _* *_ _ _ _
4.	P2.3	P0.2	* *_ *_ _* *_ _ _ _
5.	P2.4	P0.1	* *_ *_ _ _* *_ _ _ _
6.	P2.5	P0.0	* *_ *_ _ _ _* *_ _ _ _
7.	P2.6	P0.7	* *_ *_ *_ _* *_ *_ _ _
8.	P2.7	P0.6	* *_ *_ *_ _ _* *_ _ _ _ _
9.	P2.0	P0.5	* *_ *_ *_ _ _ _* *_ _ _
10.	P2.1	P0.4	_ _* *_ *_ _ _ _* *_ *_
11.	P2.2	P0.3	_ _ _* *_ *_ _ _ _* *_
12.	P2.3	P0.2	_ _ _ _* *_ *_ _ _* *_ *_

Продовження таблиці 3.1

13.	P2.4	P0.1	-- *** _ _ _ _ **
14.	P2.5	P0.0	_ _ _ *** _ _ _ _ **
15.	P2.6	P0.7	_ _ _ _ * * * _ _ * *
16.	P2.7	P0.6	_ _ * * * * _ _ _ * * *
17.	P2.0	P0.5	_ _ _ * * * * _ _ * * *
18.	P2.1	P0.4	_ _ _ _ * * * * _ _ * *
19.	P2.2	P0.3	* * _ _ * * * _ _
20.	P2.3	P0.2	* * _ _ _ * * * _ _ _
21.	P2.4	P0.1	* * _ _ _ _ * * * _ _
22.	P2.5	P0.0	* * * _ _ * * * * _ _ _
23.	P2.6	P0.7	* * * _ _ _ * * _ _
24.	P2.7	P0.6	* * * _ _ _ _ * * _ _ _
25.	P2.0	P0.5	_ _ * * _ _ _ _ * * * *
26.	P2.1	P0.4	_ _ _ * * _ _ _ _ * * *
27.	P2.2	P0.3	_ _ _ _ * * _ _ _ * * *
28.	P2.3	P0.2	_ _ * * * _ _ _ _ * * * *
29.	P2.4	P0.1	_ _ _ * * * _ _ _ _ * * * *
30.	P2.5	P0.0	_ _ _ _ * * * _ _ _ * *

Зміст звіту з самостійної роботи.

Звіт має містити титульний лист, тему, мету роботи, умову індивідуального завдання, текст програми з коментарями до кожної команди (групи команд), які реалізують виконання однієї з умовних частин всієї програми, висновки.

Контрольні запитання.

1. Дати визначення поняттю «підпрограма».
2. Порівняти використання підпрограм та команд умовного переходу.
3. Пояснити застосування команд роботи з підпрограмами.

4. ЗАВДАННЯ №3

Розроблення програм з розгалуженими структурами.

Мета. Розширення навичок застосування команд розгалужених структур.

Завдання.

Написати програму для пошуку кількості чисел, що знаходяться в межах: $05 < X < 10H$ з масиву цілих чисел у комірках пам'яті (20H), (21H), ..., (29H). Результат помістити в комірку (2AH).

Приклад лістингу програми з коментарями наводиться нижче.

```

ORG 0
MOV R0 , #20H      ;20H → (R0) – базова адреса масиву;
MOV R1 , #0AH      ;0AH→(R1) – лічильник циклів;
MOV R2 , #0        ;0H→(R2) – обнулення лічильника
                   ; чисел;

L0: MOV A , @R0     ;((R0)) → (A)
CLR C              ;C=0
SUBB A , #05H      ;(A) – 05H – C → (A)
JZ L1              ;якщо (A)=0 (X=05), то перехід мітку L1
JC L1              ;якщо C=1 (X<05), то перехід мітку L1
MOV A , @R0        ;((R0)) → A
CLR C              ; C=0
SUBB A , #10H      ;(A) – 10H – C → (A)
JZ L1              ;якщо A=0 (X=10H), то перехід на L1
JNC L1             ;якщо C=0 (X>10H) , то перехід на L1
INC R2             ;(R2)+1→(R2) збільшити лічильник
                   ; чисел на 1

L1: INC R0          ;(R0)+1→(R0) збільшити адресу на 1
DJNZ R1, L0        ;(R1)-1→(R1); якщо (R1)≠0, то перехід
                   ; на мітку L0

MOV 2AH, R2        ;(R2)→ (2AH)
L3: JMP L3          ; перехід на L3
END

```

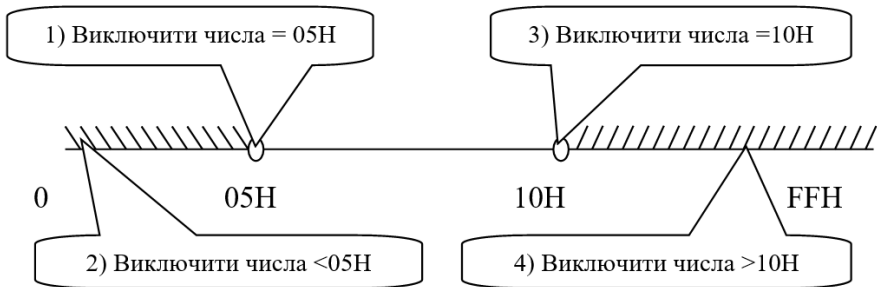


Рисунок 4.1 – Ілюстрація розв'язання прикладу завдання 1.

Для порівняння чисел X та Y треба відняти $X-Y$ і проаналізувати результати (табл. 4.1)

Таблиця 4.1 – Аналіз результату порівняння двох чисел

Умови	Результат	
$X > Y$	$A \neq 0$, $H_i - 0$	Позики немає, $C=0$
$X = Y$	$A=0$, Так -1	Позики немає, $C=0$
$X < Y$	$A \neq 0$, $H_i - 0$	Позики є, $C=1$

Самостійно написати програму для розв'язання математичного завдання.

Знайти мінімум з масиву цілих чисел, що розташовані у комірках з адресами (20H) ... (29H). Результат помістити у комірку (2AH).

ПЕРЕЛІК ПОСИЛАНЬ

1. Мікропроцесорна техніка: навч. посібник / В.В. Ткачов, Г.Грулер, Н. Нойбергер та ін. – Д.: Національний гірничий університет, 2012. – 188 с.
2. Белов, А. В. Самоучитель по микропроцессорной технике. – СПб.: Наука и техника, 2003. – 224 с.
3. Сташин В.В. Проектирование цифровых устройств на одно кристалльных микроконтроллерах / В.В. Сташин, А.В. Урусов, О.Ф. Молногонцева – М.: Энергоатомиздат, 1990. – 224 с.
4. Карташов, Б. А. Системы автоматического регулирования с микроЭВМ / Б.А. Карташов, Е.А. Шабаев – Зерноград: АЧГАА, 2008. – 26 с.
5. Балашов Е. П., Пузанков Д. В. Микропроцессоры и микропроцессорные системы / Под ред. В.Б. Смолова – М.: Радио и связь, 1981. – 328 с.
6. Липовецкий Г.П. и др. Однокристалльные микроЭВМ семейств МК48, МК51. – М., 1992. – 344 с.
7. Ишматов, З. Ш. Микропроцессорное управление электроприводами и технологическими объектами. Полиномиальные методы. – Екатеринбург, УГТУ-УПИ, 2007. – 278 с.
8. Магда, Ю. С. Микроконтроллеры серии 8051: практический поход. – М. ДМК Пресс, 2008. – 228 с.
9. Горюнов А. Г., Ливенцов С.Н. Архитектура микроконтроллера INTEL 8051: Учебное пособие. – Томск, Изд-во ТПУ, 2005. – 86 с.
10. Веприк, В.Н. и др. Микроконтроллеры семейства MCS-51: Учебное пособие / В.Н. Веприк, В.А. Афанасьев, А.И. Дружинин, А.А. Земсков, А.Р. Исаев, О.В. – Новосибирск, 1997. – 62 с.
11. Осадчий, В. В. Лабораторный стенд для исследования алгоритмов микропроцессорных систем управления шаговыми двигателями / В. В. Осадчий, Е. С. Назарова, С. Ю. Тоболкин // Електромеханічні і енергозберігаючі системи. – 2014. – Вип. 2/(26). – С.102-108.
12. Коваленко, М. А. Автономний експериментальний стенд для випробування уніполярного крокового двигуна на базі мікроконтролера / М. А. Коваленко, Д. С. Мацюк // Електротехніка та електроенергетика. – 2015. – № 2. – С. 15-20.

13. Осадчий В. В. Дослідження позиційного електропривода двомасової системи з внутрішнім слідкуючим контуром / В.В.Осадчий, О. С. Назарова, М. О. Олейніков // Вісник НТУ «ХП» - Харків, 2019. – С.47-54.

14. Назарова О. С. Ідентифікація кутової швидкості при завадах в оптичній системі енкодера / О.С. Назарова, В. В. Осадчий, І. А. Мелешко, М. О. Олейніков // Вісник НТУ «ХП» - Харків, 2019. – С.65-69.

15. Осадчий, В.В. Система автоматичного керування двомасовим позиційним електроприводом / В.В. Осадчий, О.С. Назарова, М.О. Олейніков // 7-а міжнар. наук.-практ. інтернет-конф. молодих учених і студентів «Актуальні проблеми автоматизації та управління», 30 лист. 2018 р. : мат. конф. - Луцьк, 2019. - С. 59-65.

16. Назарова, О.С. Визначення точності метода ідентифікації ступеня завантаження двошвидкісного ліфта / О.С. Назарова, В.В.Осадчий , С.С. Шульженко // 7-а наук.-практ. конференції «Методи та засоби неруйнівного контролю промислового обладнання», 19-20 лист. 2019 р. : мат. конф. - Івано-Франківськ, 2019. - С.85-88.

17. Nazarova O., Osadchyy V., Shulzhenko S. Accuracy improving of the two-speed elevator positioning by the identification of loading degree // International Conference «Modern Electrical and Energy Systems» (MEES-2019), September 23-25, 2019 Kremenchuk Mykhailo Ostrohradskyi National University, Ukraine, pp. 50 - 53 DOI: 10.1109/MEES.2019.8896414

Додаток А

Перелік команд мікроконтролера Intel 8051

Таблиця А.1 - Група команд пересилання даних

Назва команди	Мнемокод	Т	Б	Ц	Операція
Пересилання в акумулятор з регістра (n=0..7)	MOV A,Rn	1	1	1	(A)←(Rn)
Пересилання в акумулятор вмісту комірки ВПД	MOV A,dir	3	2	1	(A)←(dir)
Пересилання в акумулятор байта з РПД (i=0,1)	MOV A,@Ri	1	1	1	(A)←((Ri))
Завантаження в акумулятор константи	MOV A,#d	2	2	1	(A)←#d
Пересилання в регістр з акумулятора	MOV Rn,A	1	1	1	(Rn)←(A)
Пересилання в регістр вмісту комірки ВПД	MOV Rn,dir	3	2	2	(Rn)←(dir)
Завантаження в регістр константи	MOV Rn,#d	2	2	1	(Rn)←#d
Пересилання у комірку ВПД вмісту акумулятора	MOV dir,A	3	2	1	(dir)←(A)
Пересилання у комірку ВПД вмісту регістра	MOV dir,Rn	3	2	2	(dir)←(Rn)
Пересилання у комірку ВПД з комірки ВПД	MOV dir,dir	9	3	2	(dir)←(dir)
Пересилка байта з РПД у комірку ВПД	MOV dir,@Ri	3	2	2	(dir)←((Ri))
Пересилання константи у комірку ВПД	MOV dir,#d	7	3	2	(dir)←#d
Пересилання в РПД вмісту акумулятора	MOV @Ri,A	1	1	1	((Ri))←(A)
Пересилання в РПД вмісту комірки ВПД	MOV @Ri,dir	3	2	2	((Ri))←(dir)

Продовження таблиці А.1

Пересилання в РПД константи	MOV @Ri,#d	2	2	1	$((Ri)) \leftarrow \#d$
Завантаження вказівника даних	MOV DPTR,#d16	13	3	2	$(DPTR) \leftarrow \#d16$
Пересилання байта коду, пов'язаного з DPTR в акумулятор	MOVC A,@A+DPTR	1	1	2	$(A) \leftarrow ((A) + (DPTR))$
Пересилання байта коду пов'язаного з PC в акумулятор	MOVC A,@A+PC	1	1	2	$(PC) \leftarrow (PC) + 1$ $(A) \leftarrow ((A) + (PC))$
Пересилання байта із ЗПД в акумулятор	MOVX A,@Ri	1	1	2	$(A) \leftarrow ((Ri))$
Пересилання байта із ЗПД в акумулятор	MOVX A,@DPTR	1	1	2	$(A) \leftarrow ((DPTR))$
Пересилання з акумулятора в комірку ВПД	MOVX @Ri,A	1	1	2	$((Ri)) \leftarrow (A)$
Пересилання з акумулятора комірку ЗПД	MOVX @DPTR,A	1	1	2	$((DPTR)) \leftarrow (A)$
Завантаження комірки ВПД у стек	PUSH dir	3	2	2	$(SP) \leftarrow (SP) + 1$ $((SP)) \leftarrow (dir)$
Вивантаження зі стека в комірку ВПД	POP dir	3	2	2	$(dir) \leftarrow (SP)$ $(SP) \leftarrow (SP) - 1$
Обмін акумулятора з регістром	XCH A,Rn	1	1	1	$(A) \leftrightarrow (Rn)$
Обмін акумулятора з коміркою ВПД	XCH A, dir	3	2	1	$(A) \leftrightarrow (dir)$
Обмін акумулятора з непрямоадресованою коміркою ВПД	XCH A,@Ri	1	1	1	$(A) \leftrightarrow ((Ri))$
Обмін молодшими тетрадами між непрямоадресованою коміркою ВПД і акумулятором	XCHD A,@Ri	1	1	1	$(A_{0..3}) \leftrightarrow ((Ri)_{0..3})$

Таблиця А.2 - Команди арифметичних операцій

Назва команди	Мнемокод	Т	Б	Ц	Операція
Додавання акумулятора і регістра (n=0..7)	ADD A,Rn	1	1	1	$(A) \leftarrow (A) + (Rn)$
Додавання акумулятора і комірки ВПД	ADD A, dir	3	2	1	$(A) \leftarrow (A) + (dir)$
Додавання акумулятора і непрямо адресованої комірки ВПД	ADD A,@Ri	1	1	1	$(A) \leftarrow (A) + ((Ri))$
Додавання акумулятора і константи	ADD A,#d	2	2	1	$(A) \leftarrow (A) + \#d$
Додавання акумулятора і регістра з урахуванням переносу	ADDC A,Rn	1	1	1	$(A) \leftarrow (A) + (Rn) + (C)$
Додавання акумулятора і коміркою ВПД з урахуванням переносу	ADDC A, dir	3	2	1	$(A) \leftarrow (A) + (dir) + (C)$
Додавання акумулятора і непрямо адресованої комірки ВПД з урахуванням переносу	ADDC A,@Ri	1	1	1	$(A) \leftarrow (A) + ((Ri)) + (C)$
Додавання акумулятора і константи з урахуванням переносу	ADDC A,#d	2	2	1	$(A) \leftarrow (A) + \#d + (C)$
Десяткова корекція акумулятора	DA A	1	1	1	Якщо $(A_{0..3}) > 9$ або $((AC)=1)$, то $(A_{0..3}) \leftarrow (A_{0..3}) + 6$, і якщо $(A_{4..7}) > 9$ або $((C)=1)$, то $(A_{4..7}) \leftarrow (A_{4..7}) + 6$

Продовження таблиці А.2

Віднімання від акумулятора регістра і позики	SUBB A,Rn	1	1	1	$(A) \leftarrow (A)-(C)-(Rn)$
Віднімання від акумулятора комірки ВПД і позики	SUBB A, dir	3	2	1	$(A) \leftarrow (A)-(C)-(dir)$
Віднімання від акумулятора непрямо адресованої комірки ВПД і позики	SUBB A,@Ri	1	1	1	$(A) \leftarrow (A)-(C)-((Ri))$
Віднімання від акумулятора константи і позики	SUBB A,#d	2	2	1	$(A) \leftarrow (A)-(C)-\#d$
Інкремент акумулятора	INC A	1	1	1	$(A) \leftarrow (A)+1$
Інкремент регістра	INC Rn	1	1	1	$(Rn) \leftarrow (Rn)+1$
Інкремент комірки ВПД	INC dir	3	2	1	$(dir) \leftarrow (dir)+1$
Інкремент непрямо-адресованої комірки ВПД	INC @Ri	1	1	1	$(Ri) \leftarrow (Ri)+1$
Інкремент покажчика даних	INC DPTR	1	1	2	$(DPTR) \leftarrow (DPTR)+1$
Декремент акумулятора	DEC A	1	1	1	$(A) \leftarrow (A)-1$
Декремент регістра	DEC Rn	1	1	1	$(Rn) \leftarrow (Rn)-1$
Декремент комірки ВПД	DEC dir	3	2	1	$(dir) \leftarrow (dir)-1$
Декремент непрямо-адресованої комірки ВПД	DEC @Ri	1	1	1	$(Ri) \leftarrow (Ri)-1$

Продовження таблиці А.2

Множення акумулятора на регістр В	MUL AB	1	1	4	$(B)(A) \leftarrow (A)*(B)$
Ділення акумулятора на регістр В	DIV AB	1	1	4	$(A).(B) \leftarrow (A)/(B)$

Таблиця А.3 - Команди логічних операцій

Назва команди	Мнемокод	Т	Б	Ц	Операція
Логічне І акумулятора і регістра	ANL A,Rn	1	1	1	$(A) \leftarrow (A) \wedge (Rn)$
Логічне І акумулятора і комірки ВПД	ANL A, dir	3	2	1	$(A) \leftarrow (A) \wedge (dir)$
Логічне І акумулятора і непрямо адресованої комірки ВПД	ANL A,@Ri	1	1	1	$(A) \leftarrow (A) \wedge ((Ri))$
Логічне І акумулятора і константи	ANL A,#d	2	2	1	$(A) \leftarrow (A) \wedge \#d$
Логічне І комірки ВПД і акумулятора	ANL dir,A	3	2	1	$(dir) \leftarrow (dir) \wedge (A)$
Логічне І комірки ВПД і константи	ANL dir,#d	7	3	2	$(dir) \leftarrow (dir) \wedge \#d$
Логічне АБО акумулятора і регістра	ORL A,Rn	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Логічне АБО акумулятора і комірки ВПД	ORL A, dir	3	2	1	$(A) \leftarrow (A) \vee (dir)$
Логічне АБО акумулятора і непрямоадресованої комірки ВПД	ORL A,@Ri	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$

Продовження таблиці А.3

Логічне АБО акумулятора і константи	ORL A,#d	2	2	1	$(A) \leftarrow (A) \vee \#d$
Логічне АБО комірки ВПД і акумулятора	ORL dir,A	3	2	1	$(dir) \leftarrow (dir) \vee (A)$
Логічне АБО комірки ВПД і константи	ORL dir,#d	7	3	2	$(dir) \leftarrow (dir) \vee \#d$
Що виключає АБО акумулятора і регістра	XRL A,Rn	1	1	1	$(A) \leftarrow (A) \vee (Rn)$
Що виключає АБО акумулятора і комірки ВПД	XRL A, dir	3	2	1	$(A) \leftarrow (A) \vee (dir)$
Що виключає АБО акумулятора і непрямоадресованої комірки ВПД	XRL A,@Ri	1	1	1	$(A) \leftarrow (A) \vee ((Ri))$
Що виключає АБО акумулятора і константи	XRL A,#d	2	2	1	$(A) \leftarrow (A) \vee \#d$
Що виключає АБО комірки ВПД і акумулятора	XRL dir,A	3	2	1	$(dir) \leftarrow (dir) \vee (A)$
Що виключає АБО комірки ВПД і константи	XRL dir,#d	7	3	2	$(dir) \leftarrow (dir) \vee \#d$
Очищення акумулятора	CLR A	1	1	1	$(A) \leftarrow 0$
Інверсія акумулятора	CPL A	1	1	1	$(A) \leftarrow \overline{(A)}$
Зрушення акумулятора вліво	RL A	1	1	1	$(A_{n+1}) \leftarrow (A_n),$ $n=0..6, (A_0) \leftarrow (A_7)$

Продовження таблиці А.3

Зрушення акумулятора вліво через прапор переносу	RLC A	1	1	1	$(A_{n+1}) \leftarrow (A_n),$ $n=0..6, (A_0) \leftarrow (C),$ $(C) \leftarrow (A_7)$
Зрушення акумулятора вправо	RR A	1	1	1	$(A_{n+1}) \leftarrow (A_n),$ $n=0..6, (A_7) \leftarrow (A_0)$
Зрушення акумулятора вправо через прапор переносу	RRC A	1	1	1	$(A_{n+1}) \leftarrow (A_n),$ $n=0..6, (A_7) \leftarrow (C),$ $(C) \leftarrow (A_0)$
Обмін місцями тетрад в Акумуляторі	SWAP A	1	1	1	$(A_{0-3}) \leftarrow (A_{4-7})$

Таблиця А.4 - Команди роботи з бітами

Назва команди	Мнемокод	Т	Б	Ц	Операція
Очищення переносу	CLR C	1	1	1	$(C) \leftarrow 0$
Очищення біта	CLR bit	4	2	1	$(\text{bit}) \leftarrow 0$
Установка переносу	SETB C	1	1	1	$(C) \leftarrow 1$
Установка біта	SETB bit	4	2	1	$(\text{bit}) \leftarrow 1$
Інверсія переносу	CPL C	1	1	1	$(C) \leftarrow (\bar{C})$
Інверсія біта	CPL bit	4	2	1	$(\text{bit}) \leftarrow (\bar{\text{bit}})$
Логічне І біта і прапору переносу	ANL C, bit	4	2	2	$(C) \leftarrow (C) \wedge (\text{bit})$
Логічне І інверсії біта і переносу	ANL C, /bit	4	2	2	$(C) \leftarrow (C) \wedge (\bar{\text{bit}})$

Продовження таблиці А.4

Логічне АБО біта і прапора переносу	ORL C,bit	4	2	2	$(C) \leftarrow (C) \vee (\text{bit})$
Логічне АБО інверсії біта і прапора переносу	ORL C,/bit	4	2	2	$(C) \leftarrow (C) \vee (\overline{\text{bit}})$
Пересилання біта в прапор переносу	MOV C,bit	4	2	1	$(C) \leftarrow (\text{bit})$
Пересилання прапора переносу в біт	MOV bit,C	4	2	2	$(\text{bit}) \leftarrow (C)$

Таблиця А.5 - Команди передачі керування

Назва команди	Мнемокод	Т	Б	Ц	Операція
Довгий перехід	LJMP .. 16	12	3	2	$(PC) \leftarrow \text{dir } 16$
Абсолютний перехід всередині сторінки у 2 Кбайта	AJMP dir11	6	2	2	$(PC) \leftarrow (PC) + 2$ $(PC_{0..10}) \leftarrow \text{dir } 11$
Короткий відносний перехід всередині сторінки у 256 байт	SJMP rel	5	2	2	$(PC) \leftarrow (PC) + 2$ $(PC) \leftarrow (PC) + \text{rel}$
Непрямий відносний перехід	JMP @A+DPTR	1	1	2	$(PC) \leftarrow (A) + (DPTR)$
Перехід, якщо акумулятор дорівнює нулю	JZ rel	5	2	2	$(PC) \leftarrow (PC) + 2$, якщо $(A) = 0$, то $(PC) \leftarrow (PC) + \text{rel}$

Продовження таблиці А.5

Перехід, якщо акумулятор не дорівнює нулю	JNZ rel	5	2	2	$(PC) \leftarrow (PC) + 2$, якщо $(A) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Перехід, якщо прапор переносу дорівнює одиниці	JC rel	5	2	2	$(PC) \leftarrow (PC) + 2$, якщо $(C) = 1$, то $(PC) \leftarrow (PC) + rel$
Перехід, якщо прапор переносу дорівнює нулю	JNC rel	5	2	2	$(PC) \leftarrow (PC) + 2$, якщо $(C) = 0$, то $(PC) \leftarrow (PC) + rel$
Перехід, якщо біт дорівнює одиниці	JB bit,rel	11	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(bit) = 1$, то $(PC) \leftarrow (PC) + rel$
Перехід, якщо біт дорівнює нулю	JNB bit,rel	11	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(bit) = 0$, то $(PC) \leftarrow (PC) + rel$
Перехід, якщо біт встановлено, з наступним скиданням біта	JBC bit,rel	11	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(bit) = 1$, то $(bit) \leftarrow 0$ і $(PC) \leftarrow (PC) + rel$
Декремент регістра і перехід, якщо він не дорівнює нулю	DJNZ Rn,rel	5	2	2	$(PC) \leftarrow (PC) + 2$, $(Rn) \leftarrow (Rn) - 1$, якщо $(Rn) \neq 0$, то $(PC) \leftarrow (PC) + rel$
Декремент комірки ВПД і перехід, якщо її вміст не дорівнює нулю	DJNZ dir,rel	8	3	2	$(PC) \leftarrow (PC) + 2$, $(dir) \leftarrow (dir) - 1$, якщо $(dir) \neq 0$, то $(PC) \leftarrow (PC) + rel$

Продовження таблиці А.5

Порівняння акумулятора з коміркою ВПД і перехід, якщо вони не дорівнюють одне одному	CJNE A, dir, rel	8	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(A) \neq (\text{dir})$, то $(PC) \leftarrow (PC) + \text{rel}$, якщо $(A) < (\text{dir})$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
Порівняння акумулятора з константою і перехід, якщо вони не дорівнюють одне одному	CJNE A, #d, rel	10	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(A) \neq \#d$, то $(PC) \leftarrow (PC) + \text{rel}$, якщо $(A) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
Порівняння регістра з константою і перехід, якщо вони не дорівнюють одне одному	CJNE Rn, #d, rel	10	3	2	$(PC) \leftarrow (PC) + 3$, якщо $(Rn) \neq \#d$, то $(PC) \leftarrow (PC) + \text{rel}$, якщо $(Rn) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
Порівняння непрямоадресованої комірки ВПД з константою і перехід, якщо вони не дорівнюють одне одному	CJNE @Ri, #d, rel	10	3	2	$(PC) \leftarrow (PC) + 3$, якщо $((Ri)) \neq \#d$, то $(PC) \leftarrow (PC) + \text{rel}$, якщо $((Ri)) < \#d$, то $(C) \leftarrow 1$, інакше $(C) \leftarrow 0$
Довгий виклик підпрограми	LCALL dir16	12	3	2	$(PC) \leftarrow (PC) + 3$, $(SP) \leftarrow (SP) + 1$, $((SP)) \leftarrow (PC_{0..7})$, $(SP) \leftarrow (SP) + 1$, $((SP)) \leftarrow (PC_{8..15})$, $(PC) \leftarrow \text{dir } 16$

Продовження таблиці А.5

Абсолютний виклик підпрограми в межах сторінки в 2 Кбайта	ACALL dir11	6	2	2	$(PC) \leftarrow (PC) + 2,$ $(SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{0..7}),$ $(SP) \leftarrow (SP) + 1,$ $((SP)) \leftarrow (PC_{8..15}),$ $(PC_{0..10}) \leftarrow \text{dir } 11$
Повернення з підпрограми	RET	1	1	2	$(PC_{8..15}) \leftarrow ((SP)),$ $(SP) \leftarrow (SP) - 1,$ $(PC_{0..7}) \leftarrow ((SP)),$ $(SP) \leftarrow (SP) - 1$
Повернення з підпрограми оброблення переривання	RETI	1	1	2	$(PC_{8..15}) \leftarrow ((SP)),$ $(SP) \leftarrow (SP) - 1,$ $(PC_{0..7}) \leftarrow ((SP)),$ $(SP) \leftarrow (SP) - 1$
Порожня команда	NOP	1	1	1	$(PC) \leftarrow (PC) + 1$

Додаток Б

Представлення даних у двійковій, шістнадцятковій та десятковій системі числення

Таблиця Б.1 – Відповідність представлення даних у різних системах числення

BIN (двійкова)				HEX (шістнадцяткова)	DEC (десятькова)
$2^3 = 8$ ст.біт	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$ мол.біт		
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	A	10
1	0	1	1	B	11
1	1	0	0	C	12
1	1	0	1	D	13
1	1	1	0	E	14
1	1	1	1	F	15