

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
КАФЕДРА ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

МЕТОДИЧНІ ВКАЗІВКИ

до виконання розрахунково-графічної роботи
з дисципліни «Електроніка та мікропроцесорна техніка»
для здобувачів першого (бакалаврського) рівня вищої освіти
спеціальності 141 Електроенергетика, електротехніка та електромеханіка
освітньо-професійної програми
Електроенергетика, електротехніка та електромеханіка
всіх форм навчання

Кривий Ріг

2024 р.

Укладачі: Федотов В. О., канд. техн. наук, доцент;
Сьомочкин А. Б., канд. техн. наук, доцент;
Осадчук Ю. Г., канд. техн. наук, доцент.

Відповідальний за випуск: Федотов В. О., канд. техн. наук, доцент.

Рецензент: Сінчук О. М., д-р. техн. наук, професор.

Методичні вказівки містять короткі теоретичні відомості, варіанти завдань розрахунково-графічної роботи, приклад рішення завдання з необхідними поясненнями, алгоритмами та програмою для мікроконтролера, список рекомендованої літератури.

ЗМІСТ

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНИХ РОБІТ	4
ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАВДАННЯ.....	8
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	20

ЗАВДАННЯ ДЛЯ РОЗРАХУНКОВО-ГРАФІЧНИХ РОБІТ

Масив цілих чисел X_i (максимальна довжина масиву складає 20 елементів, довжина елемента масиву складає 8 біт), від периферійного пристрою надходить до мікроконтролера через порт PORT1, використовується програмно-керований обмін з квітуванням (рис. 1.1) з лінією P3.x для приймання сигналу готовності до обміну периферійного пристрою RDY та з лінією P3.y для передачі сигналу підтвердження АСК від мікроконтролера, масив має довжину L або кінець масиву позначений міткою M, потрібно видалити елементи масиву, які дорівнюють числу N, а також виконати операцію, яка зазначена в таблиці 1.

Таблиця 1

Варіант	Операція
1	поміняти місцями 1 та 2 біти непарних елементів масиву
2	інвертувати старшу тетраду парних елементів масиву
3	інвертувати 6 та 7 біти парних елементів масиву
4	поміняти місцями тетради непарних елементів масиву
5	інвертувати парні біти елементів масиву
6	записати значення бітів старшої тетради до парних бітів елементів масиву
7	записати одиниці в непарні біти елементів масиву
8	записати нулі в 0 та 1 біти парних елементів масиву
9	поміняти місцями тетради від'ємних елементів масиву
10	інвертувати молодшу тетраду парних елементів масиву
11	записати значення бітів молодшої тетради до непарних бітів елементів масиву
12	інвертувати непарні біти елементів масиву
13	записати значення непарних бітів елементів масиву в старшу тетраду
14	записати одиниці в парні біти елементів масиву
15	інвертувати молодшу тетраду непарних елементів масиву
16	записати одиниці в молодшу тетраду елементів масиву
17	записати нулі в молодшу тетраду елементів масиву
18	записати значення парних бітів елементів масиву в молодшу тетраду
19	інвертувати старшу тетраду непарних елементів масиву

20	записати нулі в 4 та 7 біти непарних елементів масиву
21	поміняти місцями 3 та 7 біти парних елементів масиву
22	записати одиниці в старшу тетраду елементів масиву
23	поміняти місцями тетради парних елементів масиву
24	записати нулі в старшу тетраду елементів масиву
25	записати одиниці в 6 та 7 біти непарних елементів масиву
26	записати нулі в парні біти елементів масиву
27	записати нулі в непарні біти елементів масиву
28	інвертувати 0 та 7 біти непарних елементів масиву
29	записати одиниці в 0 та 3 біти парних елементів масиву
30	представити від'ємні елементи масиву додатковим кодом

Після перетворення отримані елементи масиву розташувати у комірках пам'яті починаючи з адреси ADDR.

Організувати передачу отриманого масиву однобайтових чисел на зовнішній пристрій, який постійно готовий до обміну, через порт PORT2 з частотою f за допомогою таймера мікроконтролера (тактова частота мікроконтролера 12МГц).

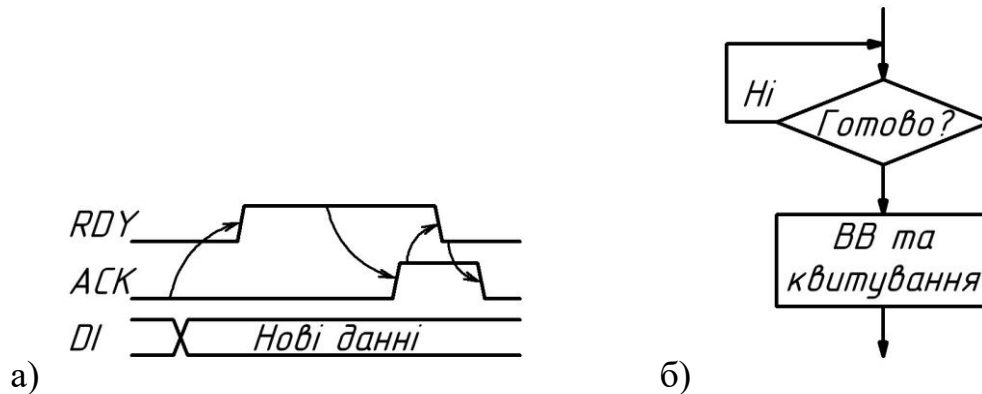


Рисунок 1.1 – Часова діаграма умовного обміну (а) та алгоритм програмно-керованого обміну умовного із заняттям циклу (б)

У таблиці 2 представлені варіанти завдання, ADDR – адреси комірок пам'яті ОЗП, PORT – порт введення-виведення мікроконтролера, який використовується для вводу даних, ADDR – адреса комірки пам'яті, починаючи з якою розташовуються елементи масиву, P3.x – лінія порту P3 для передачі сигналу RDY, P3.y – лінія порту P3 для передачі сигналу ACK, f – частота виводу елементів масиву на зовнішній пристрій, L – довжина масиву, M –

кінець масиву, N – елементи масиву, які потрібно видалити, h – числа представлені шістнадцятковою системою числення, d – числа представлені десятковою системою числення, b – числа представлені двійковою системою числення.

Таблиця 2

Варіант	PORT1	PORT2	ADDR	P3.x	P3.y	f, кГц	L	M	N
0	P2	P1	34h	P3.0	P3.1	0,1	17d	–	15d
1	P1	P2	35h	P3.3	P3.7	125	8d	–	90h
2	P2	P1	48h	P3.1	P3.2	0,2	12h	–	80h
3	P0	P0	55h	P3.3	P3.4	4	–	90h	20h
4	P1	P2	5Bh	P3.5	P3.6	100	18d	–	35d
5	P2	P1	38h	P3.7	P3.0	1	20d	–	45d
6	P0	P2	30h	P3.2	P3.3	10	10h	–	2d
7	P1	P0	40h	P3.4	P3.5	5	12d	–	5d
8	P2	P0	50h	P3.6	P3.7	0,4	–	00h	10h
9	P0	P1	5Ah	P3.1	P3.3	8	14d	–	88d
10	P1	P2	58h	P3.5	P3.7	40	–	71d	23h
11	P2	P1	40h	P3.0	P3.2	2	–	99h	99d
12	P0	P2	32h	P3.4	P3.6	0,5	15d	–	30h
13	P1	P0	44h	P3.0	P3.3	25	16d	–	15h
14	P2	P0	56h	P3.1	P3.5	50	–	FFh	7d
15	P0	P1	4Ah	P3.2	P3.6	1	–	52d	150d
16	P0	P2	48h	P3.5	P3.7	5	12h	–	15H
17	P1	P0	38h	P3.7	P3.1	0,4	–	12h	95d
18	P2	P1	46h	P3.4	P3.3	100	–	BBh	180d
19	P0	P1	4Bh	P3.5	P3.1	1	19d	–	120d
20	P1	P2	53h	P3.6	P3.7	50	–	50h	250d
21	P0	P2	46h	P3.4	P3.0	2	–	10d	12d
22	P1	P2	30h	P3.0	P3.1	5	–	12h	25d
23	P2	P1	40h	P3.2	P3.3	0,5	18d	–	10h
24	P0	P1	50h	P3.0	P3.5	10	–	00h	FFd
25	P2	P0	33h	P3.7	P3.2	30	–	25h	48h
26	P1	P2	52h	P3.1	P3.2	20	12d	–	150d
27	P2	P0	3Ah	P3.2	P3.0	0,4	8d	–	90h
28	P0	P2	36h	P3.3	P3.1	4	–	16h	130d
29	P1	P0	42h	P3.1	P3.7	0,8	–	AAh	5Ah

30	P2	P1	3Fh	P3.2	P3.4	25	20d	–	85d
----	----	----	-----	------	------	----	-----	---	-----

Програма повинна бути представлена за форматом таблиці 2.

Поле «Адреса команди» заповнюється під час занесення програми в пам'ять мікроконтролера.

Таблиця 2

№ блоку	Адреса команди	Мітка	Мнемокод і операнди	Виконувана операція	Коментар

ПРИКЛАДИ РОЗВ'ЯЗАННЯ ЗАВДАННЯ

Варіант 30

Перетворюємо умови завдання відповідно до варіанту 0.

Варіант	PORT1	PORT2	ADDR	P3.x	P3.y	f, кГц	L	M	N
30	P2	P1	34h	P3.0	P3.1	0,1	17d	–	15d

Масив цілих чисел X_i (максимальна довжина масиву складає 20 елементів, довжина елемента масиву складає 8 біт), від периферійного пристрою надходить до мікроконтролера через порт P2, використовується програмно-керований обмін з квитуванням (рис. 1.1) з лінією P3.0 для приймання сигналу готовності до обміну від периферійного пристрою RDY та з лінією P3.1 для передачі сигналу підтвердження АСК від мікроконтролера, масив має довжину 17d (11h), потрібно видалити елементи масиву, які дорівнюють числу 15d (0Fh), а також виконати наступну операцію:

30) представити від'ємні елементи масиву додатковим кодом.

Після перетворення отримані елементи масиву розташувати у комірках пам'яті починаючи з адреси 34h.

Організувати передачу отриманого масиву однобайтових чисел на зовнішній пристрій, який постійно готовий до обміну, через порт P1 з частотою 0,1 кГц за допомогою таймера мікроконтролера.

Рішення

Для організації обміну з периферійним пристроєм використовується програмно-керований обмін з лініями P3.0 для передачі сигналу готовності до обміну периферійного пристрою RDY та з лінією P3.1 для передачі сигналу підтвердження АСК. Потрібно налаштувати лінію P3.0 на вхід сигналу, для чого можна використати команду SETB P3.0. Для встановлення та зняття сигналу підтвердження АСК – команди SETB P3.1 та CLR P3.1. Для перевірки зміни рівня сигналу RDY по лінії P3.0 можливо використання команди JB P3.0, Lx (Lx – мітка переходу).

Масив, який приймає мікроконтролер, має певну довжину. Тому логічно

використовувати програмний лічильник, який можна створити, наприклад, на регістрі R2 банку пам'яті RB0, початково завантаживши в нього кількість елементів `MOV R2,#11h` та зменшувати його вміст після кожної обробки елемента масиву командою `DJNZ R2, Lx` з переходом до мітки Lx, якщо вміст не дорівнює 0.

Довжину отриманого після обробки масиву можна визначати за допомогою програмного лічильника на регістрі R4 банку пам'яті RB0.

Розташування елементів масиву в комірках пам'яті мікроконтролера, а також видачі отриманих чисел в порт на зовнішній пристрій, зручно виконувати, використовуючи непряму адресацію через регістр R0 банку пам'яті RB0. Такий спосіб дозволить легко змінювати покажчик на комірку пам'яті використавши команду збільшення на одиницю вмісту регістру.

Для запису числа, яке потрібно вилучити з масиву, використаємо регістр R3 банку пам'яті RB0.

Для перетворення від'ємних елементів масиву, який надходить від периферійного пристрою, в додатковий код необхідно всі розряди числа (крім знакового) інвертувати, а до результату додати 1, в отриманому числі встановлюється старший (знаковий) розряд в 1. Це можна виконати записавши число до акумулятора – `MOV A,ad`, інвертувати вміст акумулятора – `CPL A`, збільшити на одиницю вміст акумулятора – `INC A` та встановити старший розряд в 1 вмісту акумулятора – `ORL A,#80h`.

Для того, щоб передавати отримані елементи масиву на зовнішній пристрій з заданою частотою за допомогою таймеру контролера потрібно налаштувати таймер/лічильник. Для цього обираємо таймер/лічильник T/C0, налаштовуємо його на роботу таймером в режимі 2. Це дозволить перезавантажуватись таймеру самостійно при переповненні, але розрядність таймера в такому режимі буде дорівнювати 8. Підрахунок імпульсів буде виконуватись в молодшій частині регістру таймера TL0, а вміст старшої частини регістру таймера TH0 буде автоматично перезавантажуватись до молодшої частини TL0 при кожному переповненні таймера.

Через те, що частота обміну з зовнішнім пристроєм повинна дорівнювати 0,1 кГц, то потрібно підрахувати $(12 \text{ МГц}/12)/0,1 \text{ кГц} = 10000$ імпульсів. Але максимальне число, яке вміщує у собі восьмирозрядний регістр TL0 – 255. Обираємо число 250, тому для підрахунку 10000 імпульсів потрібно зафіксувати $10000/250 = 40$ (28h) переповнень таймера для передачі одного числа до зовнішнього пристрою. Для організації лічильника переповнень таймера T0 використаємо регістр R5.

Для підрахунку таймером 250 імпульсів потрібно, щоб таймер починав працювати не з нульового значення. Визначимо число, яке потрібно записати в таймер. Для цього береться максимальне значення регістру TL0, яке дорівнює 255, додається до нього 1 та віднімається кількість імпульсів, яку треба підрахувати: $(255 + 1) - 250 = 6$. Тому в регістри таймера TL0 та TH0 записується число 6.

Для дозволу роботи таймера T0 потрібно до біту керування TR0 таймером/лічильником T/C0 записати 1.

Кожне переповнення таймера викликає переривання мікроконтролера. Тому при організації періодичної видачі чисел на зовнішній пристрій доцільно використовувати це переривання, в підпрограмі обслуговування якого буде виконуватись: підрахунок потрібної кількості переповнень – DJNZ R5,Mx, передача потрібного числа на зовнішній пристрій через порт P1 – MOV P1,@R0, збільшення вмісту R0 для адресації комірки пам'яті з наступним числом INC R0, поновлення вмісту регістру R5 – MOV R5,#28h, контролю кількості переданих чисел – DJNZ R4,Mx, забороні роботи таймера T0 після закінчення передачі всіх чисел CLR TR0, та повернення із підпрограми обслуговування переривання RETI.

Схеми алгоритмів та програм для вирішення завдання зображені на рис. 1.1 та рис. 1.2. У схемі прийняті наступні позначення: (Ri) означає вміст регістра Ri, [Ri] – вміст комірки пам'яті, адреса якої розташована у регістрі Ri (наприклад, якщо в регістрі R1 записано число 20h, то позначення [R1] – це вміст комірки пам'яті з адресою 20h).

Визначаємо наступний вміст регістрів:

Регістр	R0	R2	R3	R4	R5
Вміст	ADDR	L	N	Лічильник кількості записаних елементів масиву	Лічильник кількості переповнень таймера T0

Алгоритм роботи основної програми мікроконтролера.

Блок 1. Виконується початкове завантаження регістрів значеннями зазначеними в таблиці. Для можливості використання непрямої адресації через регістр R0 в нього завантажуються початкова адреса 34h (ADDR). Значення довжини масиву завантажуються в регістр R2 для можливості створення лічильника зчитаних елементів масиву Xi. Значення елемента масиву, який потрібно видалити завантажуються в регістр R3. Регістр R4 буде використовуватись для створення лічильника записаних до пам'яті елементів масиву. Регістр R5 міститиме кількість періодів переповнення таймера T0.

Блок 2. Виконується налаштування ліній портів введення-виведення мікроконтролера на введення інформацію. P3.0 – лінія для вводу сигналу RDY, а P2 – лінії вводу масиву чисел від периферійного пристрою.

Блок 3. Готовність периферійного пристрою до передавання визначається високим рівнем сигналу по лінії 0 порту P3, якщо 0, то перехід на блок 3, виконується очікування високого рівня сигналу.

Блок 4. Якщо на лінії 0 порту P3 встановлено високий рівень сигналу, то зчитується значення елемента масиву Xi та запис його до комірки пам'яті з адресою 20h для зберігання.

Блок 5. Відповідно до часової діаграми передачі умовного обміну встановлюється високий рівень на лінії 1 порту P3 для сигналізації периферійному пристрою про те, що дані зчитані.

Блок 6. Перевірка надходження низького рівня сигналу від периферійного пристрою по лінії 0 порту P3 для контролю того, що периферійний пристрій зафіксував, що дані були зчитані, якщо 1, то перехід на блок 6 та очікування.

Блок 7. Встановлення низького рівня на лінії 1 порту P3 відповідно до часової діаграми передачі умовного обміну.

Блок 8. Запис вмісту комірки пам'яті з адресою 20h в акумулятор для

можливості обробки елемента масиву.

Блок 9. Перевірка, чи дорівнює прийнятий елемент масиву значенню N для видалення такого елемента, якщо так, то перехід на блок 20, запис елемента в пам'ять не виконується.

Блок 10. Перевірка знаку прийнятого елемента, який записаний до комірки з адресою 20h, для чого перевіряється вміст старшого біту, якщо 1, то число додатне, перетворення в додатковий код непотрібно, виконується перехід на блок 16.

Блок 11. Запис вмісту комірки пам'яті з адресою 20h в акумулятор для можливості подальшої обробки елемента масиву.

Блок 12. Починається перетворення від'ємного елемента масиву в додатковий код, перша дія – інверсія вмісту акумулятора.

Блок 13. Збільшення вмісту акумулятора на 1.

Блок 14. Логічне «АБО» над вмістом акумулятора і числом 80h (10000000b), це необхідно для установки одиниці в старшому біті числа, яка показує, що число від'ємне.

Блок 15. Запис вмісту акумулятора в комірку пам'яті з адресою 20h для проміжного зберігання.

Блок 16. Запис вмісту комірки пам'яті з адресою 20h в акумулятор.

Блок 17. Запис вмісту акумулятора в комірку пам'яті, адреса якої знаходиться в регістрі R0. Зберігається оброблений елемент масиву в комірці пам'яті контролера за необхідною адресою.

Блок 18. Збільшення на 1 вмісту регістру R4 для підрахунку кількості записаних в пам'ять контролера елементів масиву.

Блок 19. Збільшення на 1 вмісту регістру R0 для можливості адресації комірки пам'яті для зберігання наступного елемента масиву.

Блок 20. Зменшення вмісту регістра R2 на 1, та перевірка отриманого значення на 0, якщо не дорівнює 0, то перехід на блок 3. Підрахунок кількості введених елементів масиву чисел, який надходить від периферійного пристрою.

Блок 21. Запис в регістр R0 значення 34h для можливості адресації

елементів при передачі їх на зовнішній пристрій.

Блок 22. Запис в регістр спеціальних функцій TMOD значення 02h – налагодження таймера/лічильника T/C0 на роботу таймером в режимі 2 з синхронізацією від внутрішнього генератора тактових імпульсів.

Блок 23. Запис в регістр TH0 значення 06h.

Блок 24. Запис в регістр TL0 значення 06h.

Блок 25. Запис 1 в біт ET0. Дозвіл переривання від таймера.

Блок 26. Запис 1 в біт EA. Дозвіл всіх переривань.

Блок 27. Запис 1 в біт TR0. Дозвіл роботи таймера.

Блок 28. Перевірка значення біту TR0, якщо дорівнює 1, то перехід на блок 28 – очікування кінця передачі чисел на зовнішній пристрій.

Алгоритм роботи підпрограми обслуговування переривання від таймера T0.

Блок 1. Для того, щоб організувати необхідну частоту передачі чисел на зовнішній пристрій потрібно, щоб тільки на кожному 40-му спрацюванні переривання від таймера виконувалась процедура передачі числа на зовнішній пристрій, для чого зменшується вміст регістра R5 на 1, перевіряється отримане значення на 0, якщо не дорівнює 0, то перехід на блок 7.

Блок 2. Запис вмісту комірки пам'яті, яка адресована регістром R0 в порт P1 – запис в порт P1 обробленого елемента масиву Xi із комірки пам'яті.

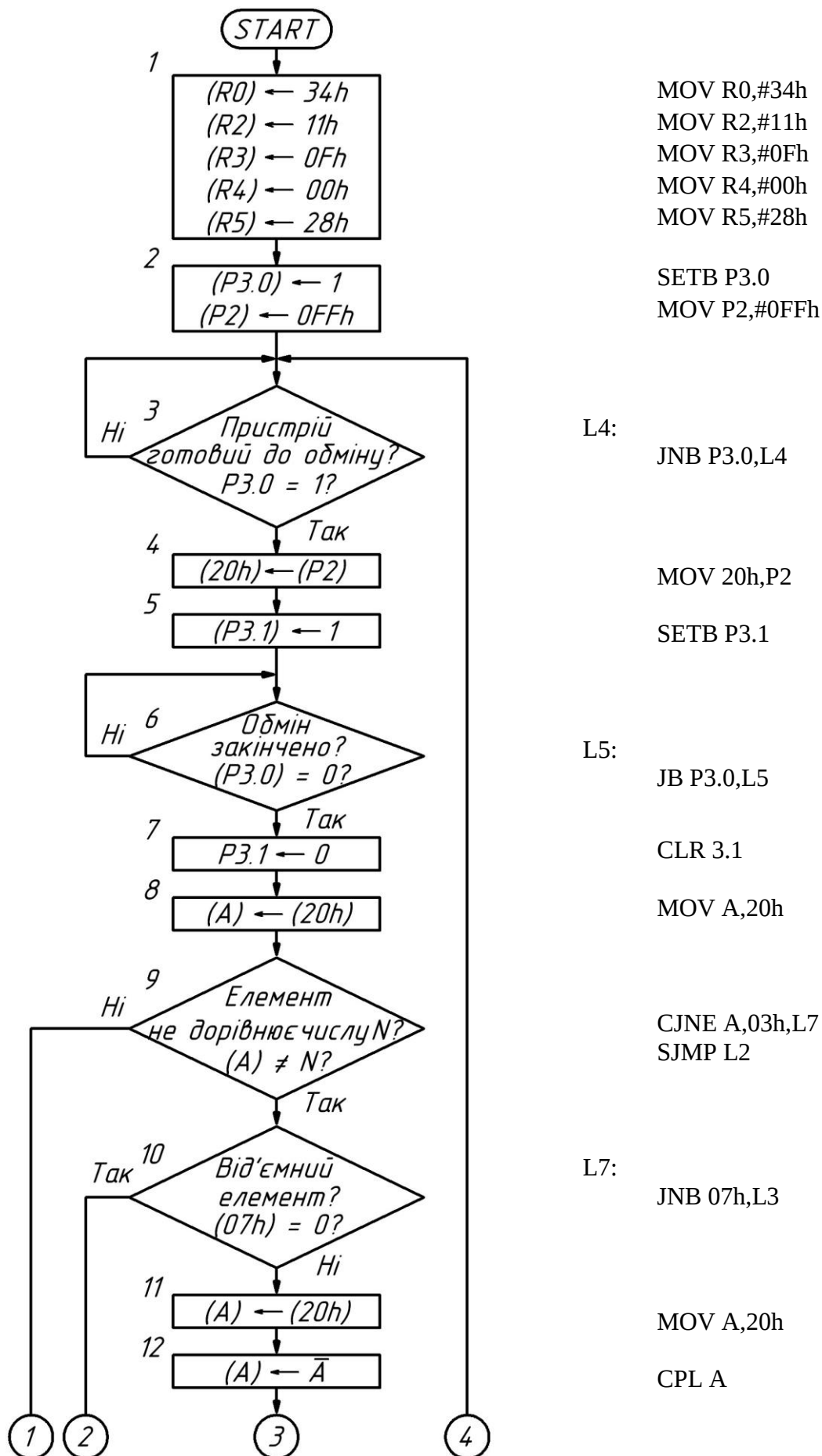
Блок 3. Збільшення на 1 вмісту регістру R0 для можливості адресації наступного числа.

Блок 4. Для завдання кількості переповнень таймера T0 в регістр R5 записується число 40d (28h).

Блок 5. Для підрахунку кількості переданих елементів зменшується вміст регістра R4 на 1 та перевіряється отримане значення на 0, якщо не дорівнює 0, то не всі елементи передані та виконується перехід на блок 7.

Блок 6. Якщо всі елементи передані, то записується 0 в біт TR0 для зупинки таймера T0.

Блок 7. Виконується повернення з підпрограми обслуговування.



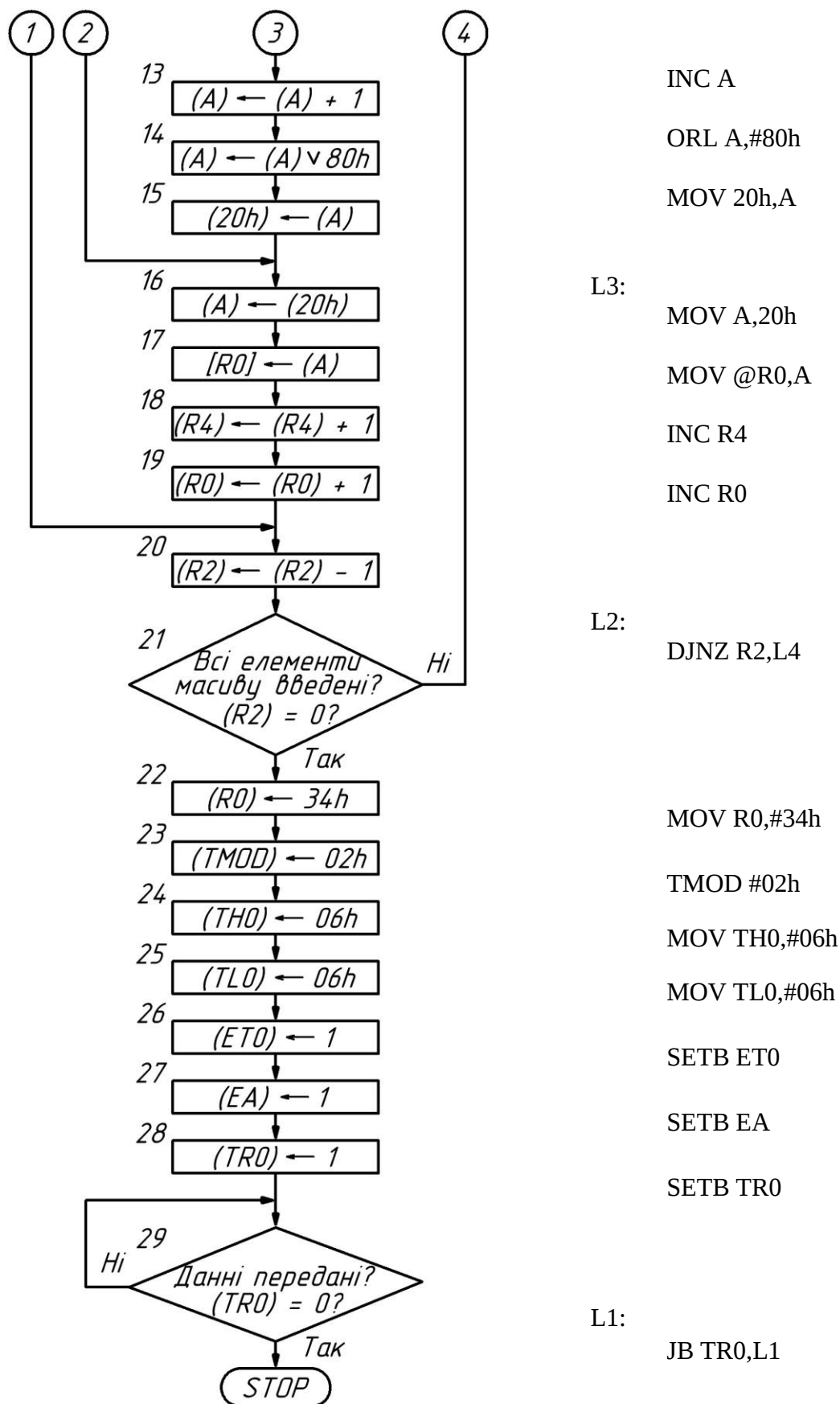


Рисунок 1.1 – Алгоритм та програма роботи мікроконтролера

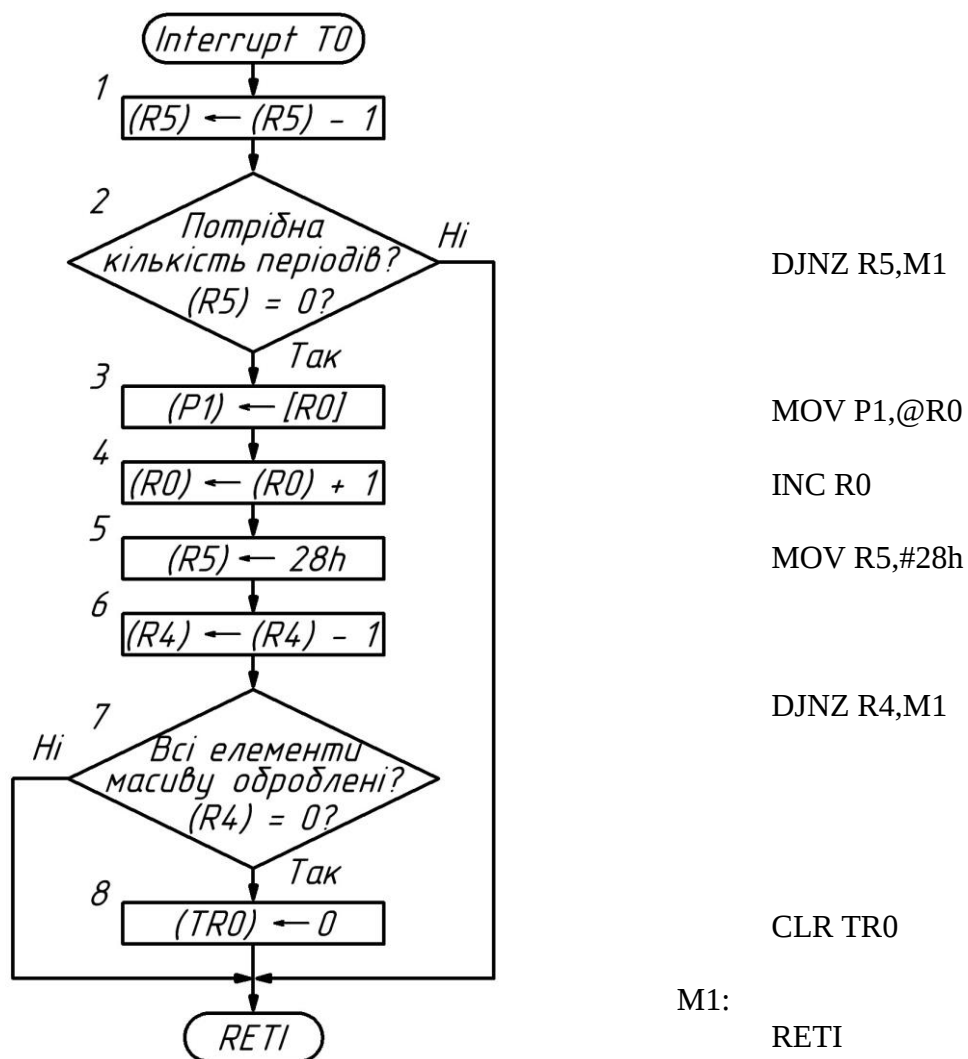


Рисунок 1.2 – Алгоритм та підпрограма обслуговування переривання від таймера T0

Основна програма роботи мікроконтролера

№ блоку	Адреса команди	Мітка	Мнемокод і операнди	Виконувана операція	Коментар
1	0030 0032 0034 0036 0038		MOV R0,#34h MOV R2,#05h MOV R3,#0Fh MOV R4,#00h MOV R5,#28h	(R0) ← 34h (R2) ← 05h (R3) ← 0Fh (R4) ← 00h (R5) ← 28h	Виконується початкове завантаження регістрів значеннями зазначеними в таблиці.
2	003A 003C		SETB P3.0 MOV P2,#0FFh	(P3.0) ← 1 (P2) ← 0FFh	Налаштування ліній портів введення-виведення мікроконтролера на введення.

3	003F	L4	JNB P3.0,L4	Якщо (P3.0) = 0, то (PC) ← 003Fh	Перевірка надходження високого рівня сигналу від периферійного пристрою на лінію 0 порту P3, якщо 0, то перехід на мітку L4 та очікування.
4	0042		MOV 20h,P2	(20h) ← (P2)	Запис елементу масиву до комірки пам'яті з адресою 20h.
5	0045		SETB P3.1	(P3.1) ← 1	Встановлення високого рівня на лінії 1 порту P3.
6	0047	L5	JB P3.0,L5	Якщо (P3.0) = 1, то (PC) ← 0047h	Перевірка надходження низького рівня сигналу від периферійного пристрою на лінію 0 порту P3, якщо 1, то перехід на мітку L5 та очікування.
7	004A		CLR P3.1	(P3.1) ← 0	Встановлення низького рівня на лінії 1 порту P3.
8	004C		MOV A,20h	(A) ← (20h)	Запис вмісту комірки пам'яті з адресою 20h в акумулятор.
9	004E		CJNE A,03h, L7	Якщо (A) <>(03h), то (PC) ← 0053h	Перевірка, чи не дорівнює прийнятий елемент масиву значенню N (N знаходиться в регістрі R3, комірка пам'яті якого має адресу 03h), якщо так, то перехід на мітку L7. Якщо елемент масиву дорівнює значенню N, то перехід на мітку L2.
	0051		SJMP L2	(PC) ← 0063h	
10	0053	L7	JNB 07h,L3	Якщо (07h) = 0, то (PC) ← 005Ch	Перевірка знаку прийнятого елемента, якщо число додатне, то перехід на мітку L3.
11	0056		MOV A,20h	(A) ← (20h)	Запис вмісту комірки пам'яті з адресою 20h в акумулятор.
12	0058		CPL A	(A) ← ($\bar{A}$)	Інверсія вмісту акумулятора для подальшого переведення числа в додатковий код.
13	0059		INC A	(A) ← (A) + 1	Збільшення вмісту акумулятора на 1.
14	005A		ORL A,#80h	(A) ← (A) ∨ 80h	Логічне «АБО» над вмістом акумулятора і числом 80h, це необхідно для установки одиниці в старшому біті числа, яка показує, що число від'ємне.

15	005C		MOV 20h,A	$(20h) \leftarrow (A)$	Запис вмісту акумулятора в комірку пам'яті з адресою 20h.
16	005E	L3	MOV A,20h	$(A) \leftarrow (20h)$	Запис вмісту комірки пам'яті з адресою 20h в акумулятор.
17	0060		MOV @R0,A	$[R0] \leftarrow (A)$	Запис вмісту акумулятора в комірку пам'яті, адреса якої знаходиться в регістрі R0.
18	0061		INC R4	$(R4) \leftarrow (R4) + 1$	Збільшення на 1 вмісту регістру R4.
19	0062		INC R0	$(R0) \leftarrow (R0) + 1$	Збільшення на 1 вмісту регістру R0.
20	0063	L2	DJNZ R2,L4	$(R2) \leftarrow (R2) - 1$	Зменшення вмісту регістра R2 на 1, та перевірка отриманого значення на 0, якщо не дорівнює 0, то перехід на мітку L4.
21				Якщо $(R2) < 0$, то $(PC) \leftarrow 003Fh$	
22	0065		MOV R0,#34h	$(R0) \leftarrow 34h$	Запис в регістр R0 значення 34h.
23	0067		MOV TMOD,#02h	$(TMOD) \leftarrow 02h$	Запис в регістр спеціальних функцій TMOD значення 02h.
24	006A		MOV TH0,#06h	$(TH0) \leftarrow 06h$	Запис в регістр TH0 значення 06h.
25	006D		MOV TL0,#06h	$(TL0) \leftarrow 06h$	Запис в регістр TL0 значення 06h.
26	0070		SETB ET0	$(ET0) \leftarrow 1$	Запис 1 в біт ET0. Дозвіл переривання від таймеру.
27	0072		SETB EA	$(EA) \leftarrow 1$	Запис 1 в біт EA. Дозвіл всіх переривань.
28	0074		SETB TR0	$(TR0) \leftarrow 1$	Запис 1 в біт TR0. Дозвіл роботи таймеру.
29	0076	L1	JB TR0,L1	Якщо $(TR0) = 1$, то $(PC) \leftarrow 0075h$	Перевірка значення біту TR0, якщо дорівнює 1, то перехід на мітку L1.

Підпрограма обслуговування переривання від таймера T0

№ блоку	Адреса команди	Мітка	Мнемокод і операнди	Виконувана операція	Коментар
1	000B		DJNZ R5,M1	$(R5) \leftarrow (R5) - 1$	Зменшення вмісту регістра R5 на 1, та перевірка отриманого значення на 0, якщо не дорівнює 0, то перехід на мітку M1.
2				Якщо $(R5) < 0$, то $(PC) \leftarrow 0016h$	
3	000D		MOV P1,@R0	$(P1) \leftarrow [R0]$	Запис вмісту комірки пам'яті, яка адресована регістром R0 в порт P1.

4	000F		INC R0	$(R0) \leftarrow (R0) + 1$	Збільшення на 1 вмісту регістру R0.
5	0010		MOV R5,#28h	$(R5) \leftarrow 28h$	Запис числа 28h в регістр R5.
6	0012		DJNZ R4,M1	$(R4) \leftarrow (R4) - 1$	Зменшення вмісту регістра R4 на 1, та перевірка отриманого значення на 0, якщо не дорівнює 0, то перехід на мітку M1.
7				якщо $(R4) \neq 0$, то $(PC) \leftarrow 0016h$	
8	0014		CLR TR0	$(TR0) \leftarrow 0$	Запис 0 в біт TR0 для зупинки таймера T0.
	0016	M1	RETI	$(PC_{8_15}) \leftarrow [SP];$ $(SP) \leftarrow (SP) - 1;$ $(PC_{0_7}) \leftarrow [SP];$ $(SP) \leftarrow (SP) - 1$	Повернення з підпрограми обслуговування переривання від таймера T0.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Сучасні мікроконтролери в електронній та інформаційно-вимірювальній техніці : навч. посіб. для студентів ВНЗ / О. В. Вовна та ін. Покровськ : ДВНЗ «ДонНТУ», 2020. 311 с.
2. Локазюк В. М. Мікропроцесори та мікроЕОМ у виробничих системах : посіб. Київ : Видавничий центр «Академія», 2002. 368 с.
3. Грицук Ю. С. Мікропроцесорні пристрої : навч. посіб. Харків : НТУ «ХП», 2008. 348 с.
4. Матвієнко М. П. Проектування цифрових пристроїв : підручник. Київ : Ліра-К, 2024. 362 с.
5. Мікропроцесорна техніка : підручник / Ю. І. Якименко та ін. ; ред. Т. О. Терещенко. 2-ге вид. перероб. та допов. Київ : Політехніка, Кондор, 2018. 440 с.
6. Огородник К. В., Книш Б. П. Мікропроцесорна техніка : навч. посіб. Вінниця : ВНТУ, 2018. 106 с.
7. Мікропроцесорна техніка : навч. посіб. / В. В. Ткачов та ін. Дніпропетровськ : Нац. гірн. ун-т, 2012. 188 с.

МЕТОДИЧНІ ВКАЗІВКИ
до виконання розрахунково-графічної роботи
з дисципліни «Електроніка та мікропроцесорна техніка»
для здобувачів першого (бакалаврського) рівня вищої освіти
спеціальності 141 Електроенергетика, електротехніка та електромеханіка
освітньо-професійної програми
Електроенергетика, електротехніка та електромеханіка
всіх форм навчання

Укладачі: Федотов В. О., канд. техн. наук, доцент;
Сьомочкин А. Б., канд. техн. наук, доцент;
Осадчук Ю. Г., канд. техн. наук, доцент.

Реєстраційний № _____

Підписано до друку «___» _____ 2024 р.

Формат	А5	
Обсяг	21	стор.
Тираж	10	прим.

Видавничий центр КНУ, вул. Віталія Матусевича, 11,
м. Кривий Ріг