

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО
ІНСТИТУТ ЕЛЕКТРОМЕХАНІКИ, ЕНЕРГОЗБЕРЕЖЕННЯ І СИСТЕМ
УПРАВЛІННЯ



МЕТОДИЧНІ ВКАЗІВКИ
ЩОДО ПРАКТИЧНИХ ЗАНЯТЬ
З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«МІКРОПРОЦЕСОРНІ ПРИСТРОЇ»
ДЛЯ СТУДЕНТІВ ДЕННОЇ ТА ЗАОЧНОЇ ФОРМ НАВЧАННЯ
ЗІ СПЕЦІАЛЬНОСТІ
141 – «ЕЛЕКТРОЕНЕРГЕТИКА, ЕЛЕКТРОТЕХНІКА ТА
ЕЛЕКТРОМЕХАНІКА»
(У ТОМУ ЧИСЛІ СКОРОЧЕНИЙ ТЕРМІН НАВЧАННЯ)
ОСВІТНЬОГО СТУПЕНЯ «БАКАЛАВР»

КРЕМЕНЧУК 2017

Методичні вказівки щодо практичних занять з навчальної дисципліни «Мікропроцесорні пристрої» для студентів денної та заочної форм навчання зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» (у тому числі скорочений термін навчання) освітнього ступеня «Бакалавр»

Укладач старш. викл. М. Ю. Юхименко

Рецензент к. т. н., доц. А. П. Калінов

Кафедра систем автоматичного управління та електроприводу

Затверджено методичною радою Кременчуцького національного університету імені Михайла Остроградського

Протокол «___» від «___» _____ 2017 р.

Голова методичної ради _____ проф. В. В. Костін

ЗМІСТ

Вступ	4
1 Теми та погодинний розклад практичних занять	6
2 Перелік практичних занять.....	7
Практичне заняття № 1 Особливості подання інформації в мікропроцесорних пристроях і машинна арифметика	7
Практичне заняття № 2 Арифметико-логічні обчислення на мікроконтролері	12
Практичне заняття № 3 Програмна реалізація опитування датчиків аналогових величин	19
Практичне заняття № 4 Програмне визначення тривалості часових інтервалів	26
Практичне заняття № 5 Програмне виведення інформації з мікропроцесорної системи	33
Практичне заняття № 6 Розробка прикладного програмного забезпечення мікропроцесорних пристроїв для керування електромеханічними системами	40
3 Критерії оцінювання знань студентів.....	48
Список літератури.....	49

ВСТУП

Вивчення навчальної дисципліни «Мікропроцесорні пристрої» зумовлене метою навчання, що, відповідно до кваліфікаційної характеристики спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка», полягає у підготовці фахівців, здатних самостійно і творчо розв'язувати задачі проектування, раціонального використання мікропроцесорних засобів керування електромеханічним обладнанням, яке широко використовують у різних галузях вітчизняної промисловості.

Метою навчальної дисципліни «Мікропроцесорні пристрої» є поглиблення спеціальної підготовки фахівця широкого профілю, набуття студентами цілісного підходу під час аналізу режимів функціонування мікропроцесорних пристроїв керування електромеханічними системами, сприяння закріпленню та поглибленню теоретичних знань, одержання практичних навичок з даного питання.

Навчальна дисципліна «Мікропроцесорні пристрої» базується на знанні предметів «Обчислювальна техніка та програмування», «Мікроелектронні пристрої електромеханічних систем», «Теорія автоматичного керування» та ін.

Після вивчення навчальної дисципліни студент повинен

знати:

– сучасний стан та можливості елементної бази мікропроцесорної техніки;

– типові структури, склад та основні характеристики мікропроцесорів та однокристальних мікро-ЕОМ;

– особливості організації розробки програмного забезпечення мікропроцесорних систем;

– організацію апаратних та програмних засобів мікропроцесорних систем модульної структури;

– методи використання засобів мікропроцесорної техніки для проектування засобів і систем керування технологічними процесами та об'єктами;

уміти:

– розробляти алгоритми і програмні додатки для реалізації типових обчислювальних та керуючих операцій;

– обґрунтовано вибирати елементну базу для реалізації сучасних цифрових пристроїв керування електромеханічними системами, розробляти структуру та принципи функціонування мікропроцесорних пристроїв;

– проектувати системи збору, обробки та передачі технологічної інформації з використанням мікропроцесорних засобів;

– розробляти засоби узгодження мікропроцесорних систем та однокристальних мікро-ЕОМ з технологічним обладнанням.

1 ТЕМИ ТА ПОГОДИННИЙ РОЗКЛАД ПРАКТИЧНИХ ЗАНЯТЬ

№ пор.	Тема	Денна форма навчання		Заочна форма навчання	
		Кільк.	Кільк.	Кільк.	Кільк.
		год (ауд)	год СРС	год (ауд)	год СРС
Змістовний модуль 1					
1	Особливості подання інформації в мікропроцесорних пристроях і машинна арифметика	2	3	1	4
2	Арифметико-логічні обчислення на мікроконтролері	2	3	1	4
3	Програмна реалізація опитування датчиків аналогових величин	2	3	1	4
Змістовний модуль 2					
4	Програмне визначення тривалості часових інтервалів	2	3	1	4
5	Програмне виведення інформації з мікропроцесорної системи	2	3	–	–
6	Розробка прикладного програмного забезпечення мікропроцесорних пристроїв для керування електромеханічними системами	2	3	–	–
Усього за семестр		12	18	4	16

2 ПЕРЕЛІК ПРАКТИЧНИХ ЗАНЯТЬ

Практичне заняття № 1

Тема. Особливості подання інформації в мікропроцесорних пристроях і машинна арифметика

Мета: набуття практичних навичок застосування позиційних систем числення, які використовуються в мікропроцесорних пристроях, вивчення операцій алгебри логіки.

Короткі теоретичні відомості

В ЕОМ і цифрових пристроях арифметичні та логічні операції проводяться над числами, поданими у вигляді спеціальних (машинних) кодів у прийнятій для даної машини системі числення (СЧ).

У цифровій техніці набула поширення двійкова система числення з основою $q = 2$. У двійковій системі для запису чисел використовують дві цифри: 0 і 1. Для запису числа вказують лише розрядні коефіцієнти. Запис має вигляд 1100. У процесі перетворення інформації в ЕОМ виникає задача переведення чисел з однієї СЧ в іншу. Наприклад, основа першої СЧ дорівнює q_1 , а другої – q_2 . Відповідно до рівняння (1.1) числа в обох СЧ можна записати у вигляді

$$\sum_{i=-m}^n a_i q_1^i = \sum_{j=-s}^k b_j q_2^j. \quad (1.1)$$

У загальному вигляді задачу перерахунку числа із системи числення з основою q_1 у систему числення з основою q_2 можна подати як задачу визначення коефіцієнтів b_j нового ряду. Розв'язати цю задачу можна, наприклад, підбором коефіцієнтів b_j нового ряду. Після знаходження максимального ступеня q_2 знаходять усі ступені нової основи, менші за максимальну.

Найбільшого поширення набули алгоритми переведення чисел з однієї системи числення в іншу:

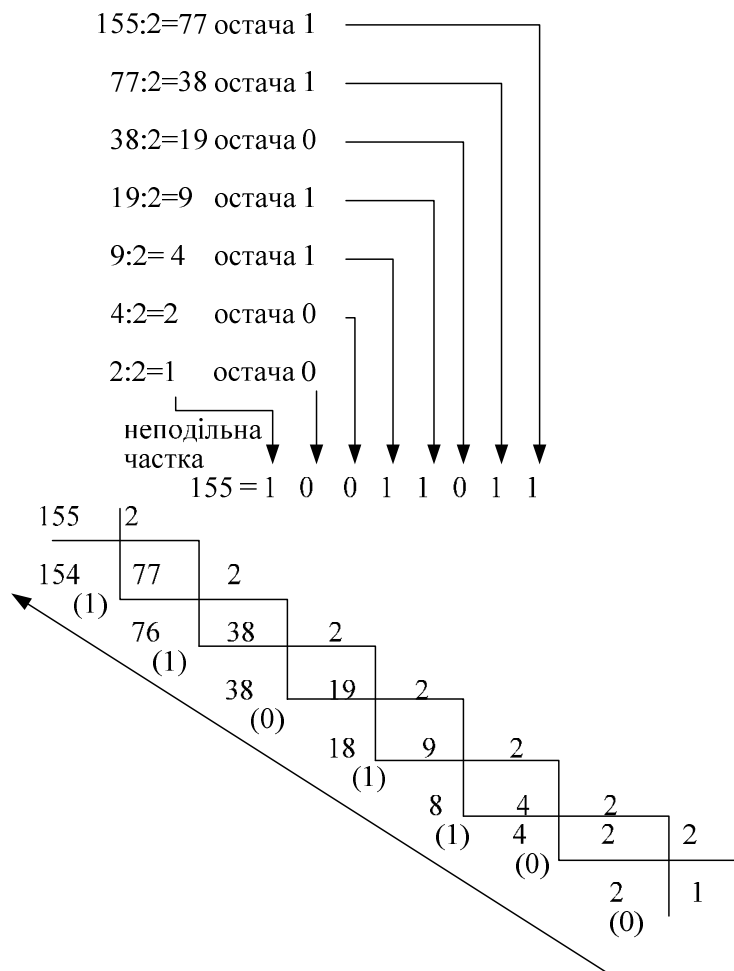
1. переведення цілих чисел поділом на основу нової системи;
2. табличний метод;
3. метод з використанням проміжної СЧ.

Переведення цілих чисел виконується поділом на основу нової системи.

Приклади виконання завдання

Приклад 1.1. Перевести число 155 з десятичної системи числення у двійкову, вісімкову та шіснадцяткову.

Розв'язання



Завдання до теми

Завдання 1.1. Перевести числа з десяткової системи числення відповідно у двійкову, вісімкову та шіснадцяткову.

Таблиця 1.1 – Варіанти для самостійного розв'язання до завдання 1.1.

№ варіанту	Завдання				
1	23	50	89	123	221
2	24	51	6	124	223
3	25	52	70	125	224
4	26	53	71	126	225
5	27	54	7	127	226
6	28	55	72	129	227
7	29	56	73	156	229
8	30	57	8	166	256
9	31	58	74	167	266
10	23	1	99	118	240
11	32	33	98	129	218
12	20	34	97	132	229
13	39	35	96	141	232
14	50	36	95	153	241
15	30	37	94	164	253
16	5	38	93	175	264
17	11	32	57	175	291
18	20	48	86	100	298
19	21	49	87	108	200
20	22	3	88	121	208
21	32	59	9	168	267
23	14	42	80	138	228

Продовження таблиці 1.1

24	15	43	81	148	238
25	16	44	82	158	248
26	17	45	83	168	258
27	18	46	84	178	268
28	19	47	85	198	278
29	32	59	97	132	222
30	13	41	96	141	258

Завдання № 2

Виконати арифметичні дії з двійковими числами, що задані в прямому коді:

Таблиця 1.2 – Варіанти завдань для самостійного розв'язання

№ варіанту	Завдання	
1	1000+10000001	10001101-10000001
2	0100+11000000	11001110-11000000
3	0010+1100000	11101110-1100000
4	0001+110000	11111111-110000
5	1100+1011000	10111111-1011000
6	0110+1001100	11111111-1001100
7	0011+1000110	10000001-1000110
8	1101+1000011	11000000-1000011
9	1011+101010	1100000-10101010
10	1110+10101	110000-1010101
11	111+11011011	1011000-11011011
12	1001+1101101	101100-1101101
13	1100+11101110	1000110-11101110

Продовження таблиці 1.3

14	11+10111011	10001101–10111011
15	110+10001111	11001110–10001111
16	1111+10001110	11101110–10001110
17	111+10001100	11111111–10001100
18	101+10001101	10111111–10001101
19	110+11001110	1111111–11001110
20	1000+11101110	10000001–11101110
21	1100+11111111	11000000–11111111
22	1001+10111111	1100000–10111111
23	101+11111111	110000–111111
24	100+10000001	1011000–10000001
25	1+11000000	1001100–11000000
26	1100+1100000	1000110–110000
27	1011+110000	10001101–110000
28	1101+1011000	11001110–1011000
29	1110+1001100	11101110–101100
30	1111+1000110	11111111–1000110

Контрольні питання

1. Дайте визначення системи числення.
2. Які системи числення використовуються в мікропроцесорних пристроях і чому саме вони?
3. Поясніть особливості подання чисел у двійковій системі числення?
4. Поясніть особливості перетворення чисел з десяткової системи у двійкову, вісімкову та шіснадцяткову.
5. Які особливості складання і віднімання двійкових чисел? Поясніть поняття «перенесення» і «позики».

7. Поясніть, які символи використовуються в шістнадцятковій системі числення для запису чисел?

8. Поясніть, як саме реалізується двійково-десятькове кодування чисел у мікропроцесорних пристроях?

10. Якій арифметичній операції еквівалентне зрушення двійкового числа вліво на два розряди?

11. Якій арифметичній операції відповідає зсув двійкового числа праворуч на один розряд?

Література: [1, с. 17–46; 2, с. 8–33; 3, с. 44–56].

Практичне заняття № 2

Тема. Арифметико-логічні обчислення на мікроконтролері

Мета: практичне освоєння методів реалізації програмних додатків мовою асемблера для мікроконтролера K1816BE51, які виконують обчислення арифметичних і логічних виразів.

Короткі теоретичні відомості

Система команд мікроконтролера K1816BE51 складається з таких груп команд: команди пересилання даних; команди арифметичних операцій; команди логічних операцій; команди передачі керування; команди бітового (бульового) процесора.

Для безпосередніх операндів буде вказуватися тільки один байт даних із загальним позначенням data 8/16 (data 8 – один байт зі значенням індикатора $w = 0$, data 16 – два байти зі значенням індикатора $w = 1$, і спочатку йде молодший, а потім старший байт даних).

Максимальна довжина команд дорівнює трьом байтам. Далі для опису команд буде використовуватися не більш трьох байт, але для будь-якої команди зі скороченою довжиною легко встановити її повний формат, оскільки виключаються тільки старший байт у двобайтового безпосереднього операнда

(якщо він є) і один чи два байти зсувів `disp8` і `disp16` у команд, що мають поле `r/m`.

Приклади виконання завдання

Як приклад розглянемо підпрограму, що реалізує сортування вмісту масиву даних (наприклад, результатів серії вимірювань) за зростанням значень. Блок–схема алгоритму програми наведена на рис. 2.1.

Програма виконує визначення максимального елемента масиву й розміщує його в останню позицію, міняючи місцями з елементом, який був останнім.

Після цього проводиться зсув межі масиву так, щоб в аналізованій частині залишилися ще не відсортовані елементи, і процедура повторюється до вичерпання масиву.

Розв'язання

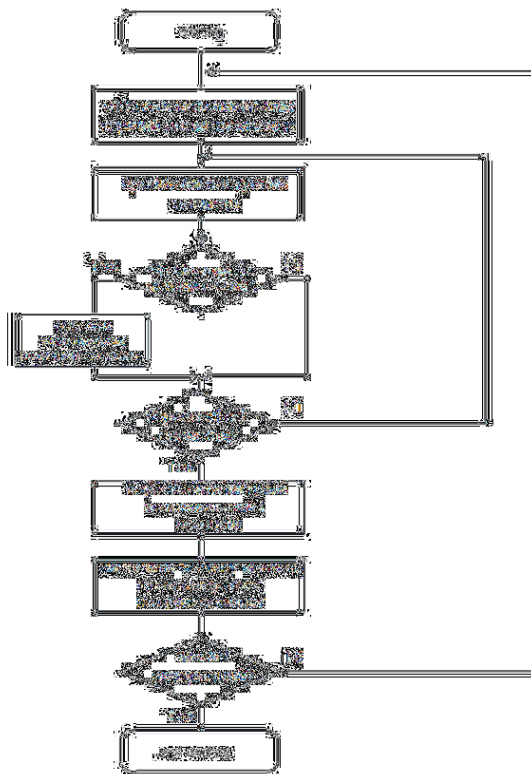


Рисунок 2.1 – Блок–схема програми сортування масиву

Текст програми на мові асемблера для мікроконтролера МК-51 наведено в лістингу.

```
ORG 8100h

START_ADR: EQU 8200h      ; адреса початку розміщення масиву
NUM_DATA:  EQU 30h       ; кількість елементів масиву
TMP_MAX:   EQU 31h       ; локальний максимум
ADR_HI:    EQU 32h       ; адреса максимального: старший байт
ADR_LO:    EQU 33h       ; молодший байт
MAIN:      MOV NUM_DATA,#15 ; виконання сортування
           LCALL SORTING ; п'ятнадцяти елементів

RET

SORTING:   PUSH PSW      ; збереження проміжних результатів у стек
           PUSH A        ; ресурсів МК
           PUSH 0
           PUSH 1
           PUSH DPH
           PUSH DPL
           MOV R0,NUM_DATA ; спочатку несортований увесь масив
$STEP:    CJNE R0,#2,$CHCK_LEN ; у масиві має бути
$CHCK_LEN: JNC $OK_LEN   ; більше одного елемента
           LJMP END_SORT
$OK_LEN:   MOV R1,0       ; визначення кількості
           ; несортованих елементів
           MOV DPTR,#START_ADR ; читання першого з них
           MOVX A,@DPTR
           MOV TMP_MAX,A ; об'явлення першого
           MOV ADR_HI,DPH ; елемента максимальним
           MOV ADR_LO,DPL
           DEC R1
```

```

$SEARCH:      INC DPTR          ; пошук максимального
              MOVX A,@DPTR    ; серед тих, що залишилися
              CJNE A,TMP_MAX,$CMP
$CMP:         JC $NO_MAX
$MAX:         MOV TMP_MAX,A   ; оновлення максимуму
              MOV ADR_HI,DPH
              MOV ADR_LO,DPL
$NO_MAX:      DJNZ R1,$SEARCH
              MOVX A,@DPTR    ; перестановка останнього
              XCH A,TMP_MAX   ; елемента в ділянці масиву,
              MOVX @DPTR,A;   що аналізується
              MOV DPH,ADR_HI  ; і максимального елемента
              MOV DPL,ADR_LO  ; в цій ділянці
              MOV A,TMP_MAX
              MOVX @DPTR,A
              DJNZ R0,$STEP   ; зсув межі
                              ; невідсортованої частини масиву
END_SORT:     POP DPL         ; відновлення
              POP DPH         ; використуваних ресурсів МК
              POP 1           ; зі стеку
              POP 0
              POP A
              POP PSW
              RET
ORG 8200H     ; дані, що підлягають сортуванню
DATA:         DB 12h,0,33h,41h,42h,43h,0FFh,0,0,1,2,0ABh,5,0C5h,87h

```

З основної програми MAIN здійснюється виклик підпрограми SORTING, у функції параметрів для якої сформовані початкова адреса масиву і кількість елементів у ньому. Масив визначено в області зовнішньої пам'яті.

Слід зазначити, що такий спосіб виявлення закінчення чергового проходу сортування має обмеження щодо обсягу масиву, яке визначається розрядністю змінної NUM_DATA, а також способом організації циклів на основі команди DJNZ – у масиві може бути не більше 255 елементів.

Зняти це обмеження можливо, наприклад, уведенням двобайтових лічильників номерів елементів масиву і використанням команд CJNE при виявленні ситуації виходу за межі аналізованої області масиву.

Завдання до теми

Завдання № 1

Написати програму мовою асемблера для МК1816BE51 з коментарями відповідно до варіанту (табл. 2.1)

Таблиця 2.1 – Варіанти завдань для самостійного розв'язання

№ варіанту	Завдання
1	Дана матриця розміром 5 на 10. Визначити максимальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 1000h, другий – 1001h і т. д. Результат занести в стек за адресою 5000h
2	Дана матриця розміром 4 на 11. Визначити мінімальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 1000h, другий – 1001h і т. д.
3	Дана матриця розміром 5 на 5. Підрахувати кількість елементів матриці, що дорівнюють нулю, якщо перший елемент матриці розташований у пам'яті за адресою 1000h, другий – 1001h і т. д. Результат занести в стек за адресою 3000h
4	Дана матриця розміром 4 на 4. Підрахувати кількість елементів матриці, що дорівнюють нулю, перший елемент матриці розташований у пам'яті за адресою 2000h, другий – 2001h і т. д.

Продовження таблиці 2.1

5	Дана матриця розміром 5 на 5. Підрахувати кількість елементів матриці, що дорівнюють нулю, перший елемент матриці розташований у пам'яті за адресою 1000h, другий – 1001h і т. д.
6	Дана матриця розміром 4 на 4. Підрахувати кількість елементів матриці, що дорівнюють нулю, якщо перший елемент матриці розташований у пам'яті за адресою 2000h, другий – 2001h і т. д. Результат занести в стек за адресою 5000h
7	Підрахувати кількість символів у рядку, що розташований у пам'яті, починаючи з адреси 1000h та закінчуючи адресою 2000h, без урахування пропусків. Відомо, що кожен символ займає один елемент пам'яті, а пропуск кодується як 01h
8	Дана матриця розміром 4 на 4. Підрахувати кількість елементів матриці, що дорівнюють нулю, якщо перший елемент матриці розташований у пам'яті за адресою 2000h, другий – 2001h і т. д. Результат занести в стек за адресою 5000h
9	Розрахувати кількість рядків, якщо відомо, що в одному рядку вміщується 72 символи. Один символ займає один елемент пам'яті. Адресація починається з 0000h, а весь текст займає 8 кБайт пам'яті
10	Дана матриця розміром 7 на 12. Визначити максимальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 2000h, другий – 2001h і т. д. Результат помістити в стек за адресою 4000h
11	Дана матриця розміром 8 на 10. Визначити мінімальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 3000h, другий – 3001h і т. д. Результат занести в стек за адресою 4000h

Продовження таблиці 2.1

12	Дана матриця розміром 4 на 4. Підрахувати кількість елементів матриці, що дорівнюють нулю, якщо перший елемент матриці розташований у пам'яті за адресою 2000h, другий – 2001h і т. д. Результат занести в стек за адресою 5000h
13	Розрахувати кількість рядків, якщо відомо, що в одному рядку вміщується 74 символи. Один символ займає один елемент пам'яті. Адресація починається з адреси 0000h, а весь текст займає 6 кБайт пам'яті
14	Дана матриця розміром 4 на 4. Підрахувати кількість елементів матриці, що дорівнюють нулю, якщо перший елемент матриці розташований у пам'яті за адресою 2000h, другий – 2001h і т. д. Результат занести в стек за адресою 5000h
15	Дана матриця розміром 5 на 10. Визначити максимальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 1000h, другий – 1001h і т. д. Результат занести в стек за адресою 5000h
16	Дана матриця розміром 5 на 6. Визначити мінімальний елемент матриці, якщо перший елемент розташований у пам'яті програм за адресою 1000h, другий – 1001h і т. д. Результат занести в стек за адресою 5000h

Контрольні питання

1. Проаналізуйте наявні формати даних і системи команд для МК K1816BE51.
2. Система команд МК K1810BE51. Група команд пересилання даних.
3. Система команд МК K1816BE51. Група команд арифметичних і логічних операцій.
4. Система команд МК K1816BE51. Група команд передачі керування.

5. Пояснити призначення ознак, що формовані МК К1816ВЕ51.
6. Порівняйте довжину команд, що використовують пряму адресацію і непряму адресацію.
7. Опишіть склад команди асемблера для МК К1816ВЕ51. Призначення кожного з полів команди.
8. Які види адресації операндів для МК К1816ВЕ51 ви знаєте, в чому полягає їх відмінність один від одного?
9. Які способи адресації використовуються в МК К1816ВЕ51?

Практичне заняття № 3

Тема. Програмна реалізація опитування датчиків аналогових величин

Мета: набуття практичних навичок програмного введення до мікропроцесорної системи інформації з датчиків технологічних параметрів, в яких вихідне значення надано аналоговим сигналом.

Короткі теоретичні відомості

У системах керування електромеханічним обладнанням одним з типових завдань є завдання визначення значень технологічних величин об'єкта керування, що надходять з датчиків і є інформацією, поданою у вигляді аналогового сигналу – рівня напруги.

До спеціалізованих пристроїв інтерфейсу мікропроцесорних систем відносяться АЦП, ЦАП та перетворювачі кодів. Спеціалізовані пристрої перетворюють інформацію до вигляду, зручного для сприйняття мікропроцесорною системою (це виконують АЦП і перетворювачі кодів). ЦАП перетворюють інформацію до вигляду, зручного для сприйняття периферійними пристроями. Перетворення аналогової інформації в цифрову базується на теоремі Котельникова: будь-який аналоговий сигнал може бути записаний у дискретній формі й інформація не буде втрачена, якщо крок квантування Δt буде вибраний за умови

$$\Delta t = \frac{1}{2F_{\max}}, \quad (3.1)$$

де F_{\max} – максимальна частота спектру передавального сигналу.

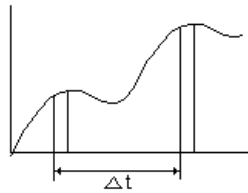


Рисунок 3.1 – Вибір кроку квантування для перетворення аналогового сигналу в цифровий

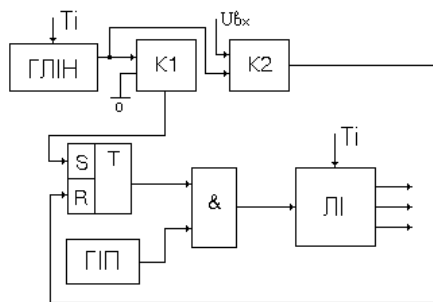


Рисунок 3.2 – Структурна схема аналогово-цифрового перетворювача

В розглянутій схемі відбувається проміжне перетворення напруги $U_{вх}$ у тривалість імпульсу тригера, яка у свою чергу визначає кількість імпульсів генератора імпульсної послідовності (ГП), підрахованих лічильником. Лічильник видає результат у двійковому коді, отже показання лічильника пропорційні $U_{вх}$. Тактовий імпульс запускає генератор напруги, яка лінійно змінюється (ГЛЗН), і обнуляє лічильник. У двох компараторах К1 і К2 відбувається порівняння напруги ГЛЗН з «0» і з $U_{вх}$.

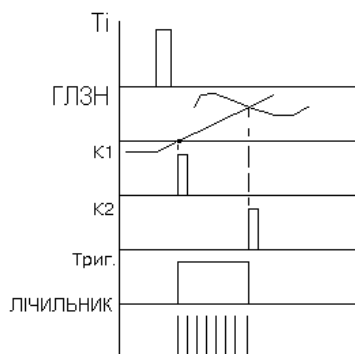


Рисунок 3.3 – Часові діаграми роботи аналогово-цифрового перетворювача

Коли рівень напруги ГЛЗН дорівнює «0», на виході К1 виробляється сигнал, який устанавлює тригер в одиничний стан. На схему «І» подається одиничний сигнал з виходу тригера і послідовність імпульсів з виходу ГП. Імпульси ГП подаються на лічильник, який їх рахує. Коли напруга пілкоподібних імпульсів буде дорівнювати $U_{вх}$, сигнал з виходу К2 перемкне тригер у нульовий стан, імпульси через схему «І» не проходять, рахунок припиняється. Кількість минулих імпульсів пропорційна $U_{вх}$. Оскільки напруга «пилки» не є лінійною, особливо на початковій ділянці, то точність перетворення невелика. Більшу точність забезпечують АЦП із зворотним зв'язком (ЗЗ) і АЦП стежувального типу.

АЦП із зворотним зв'язком

ЦАП виробляє напругу ЗЗ. Ця напруга має ступінчасту форму. З приходом тактового імпульсу лічильник починає зчитувати від генератора імпульсної послідовності (ГП). Рахунок припиняється в разі подачі нульового сигналу з вихідного компаратора. Цей сигнал виробляється в тому випадку, якщо напруга ЗЗ більша або дорівнює $U_{вх}$. Кількість зчитаних імпульсів пропорційна $U_{вх}$ у двійковому коді.

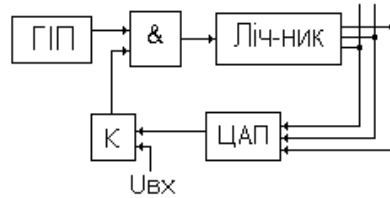


Рисунок 3.4 – Структурна схема аналого-цифрового перетворювача зі зворотним зв'язком

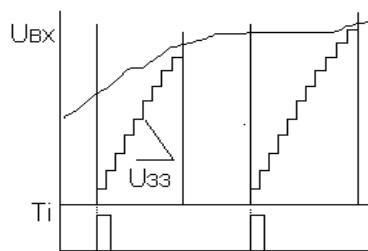


Рисунок 3.5 – Часові діаграми аналого-цифрового перетворювача зі зворотним зв'язком

АЦП стежувального типу

АЦП зі зворотним зв'язком перетворюють у двійковий код тільки напругу $U_{вх}$, що збільшується. АЦП стежувального типу перетворюють у цифровий сигнал як збільшувану, так і напругу, що зменшується. У цій схемі застосовується реверсивний лічильник. Коли на виході компаратора (К) одиничний сигнал ($U_{вх} > U_{зз}$), лічильник знаходиться в режимі прямого рахунку. Коли $U_{вх} = U_{зз}$, на виході К – «0» і лічильник переходить у режим зворотного рахунку.

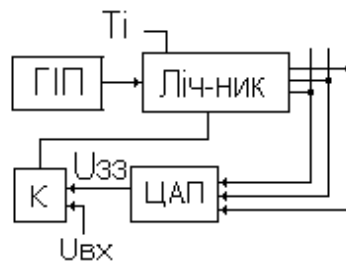


Рисунок 3.6 – Структурна схема аналого-цифрового перетворювача стежувального типу

Розглянемо підсистему вимірювання рівня освітленості, що використовує як датчик фотодіод. Принципову схему пристрою сполучення з МК 80C515 показано на рис. 3.7.

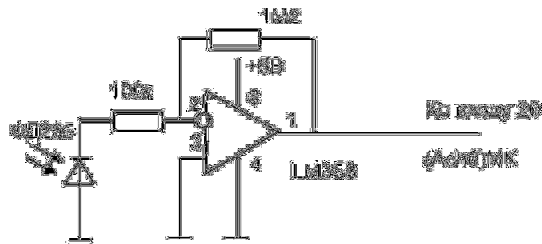


Рисунок 3.7 – Сполучення датчика з МК

Приклади виконання завдання

Як приклад розглянемо реалізацію опитування датчика, підключеного до нульового каналу АЦП. Діапазон масштабування операційним підсилювачем вхідних сигналів становить 0 – 5 В.

Розв’язання

Блок–схему алгоритму програми наведено на рис. 3.8, а її текст – у лістингу.

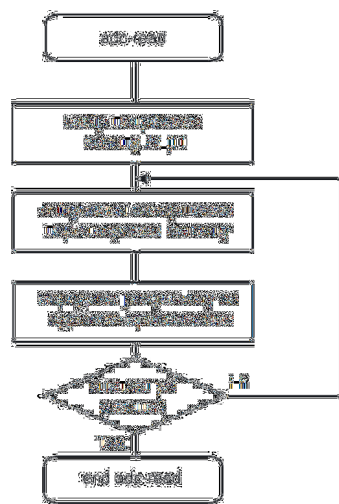


Рисунок 3.8 – Блок–схема алгоритму програми, що реалізує перетворення аналогового сигналу

Лістинг програми:

```
U1:      EQU 40h          ; напруга з датчика
U1_READ: PUSH 0          ; процедура опитування датчиків
          MOV R0,#0       ; вибір нульового
          LCALL ADC_SWITCH ; каналу
          MOV 0DAH,#0h    ; завдання діапазону
          LCALL ADC_DELAY ; затримка
          MOV U1,0D9h     ; збереження в змінну U1
          POP 0
          RET

ADC_SWITCH: MOV A,0D8h    ; налаштування АЦП:
          ANL A,#11111000b ; безперервне
          ORL A,#00001000b ; перетворення
          ORL A,R0        ; вибір каналу
          MOV 0D8h,a
          RET

ADC_DELAY: PUSH 0        ; витримка паузи
          MOV R0,#20     ; для перетворення
          DJNZ R0,$ ;в АЦП
          POP 0
          RET
```

У підпрограмі застосовується витримка часу, свідомо більшого за необхідний для виконання аналого-цифрового перетворення на розглянутому мікроконтролері.

Іншими варіантами взаємодії з АЦП є опитування прапорця готовності й застосування переривань.

Завдання до теми

Написати програму мовою асемблера для МК1816BE51 з коментарями відповідно до варіанту (табл. 3.1).

Таблиця 3.1 – Варіанти індивідуальних завдань для самостійного виконання

№ пор.	Зміст індивідуального завдання
1	Реалізувати програмно АЦП методом порозрядного врівноваження
2	Реалізувати програмно метод половинних наближень, значення відображати на динамічному індикаторі
3	Реалізувати програмно слідкувальний метод аналого-цифрового перетворення з первісним порозрядним врівноваженням
4	Реалізувати програмно слідкувальний метод аналого-цифрового перетворення з первісним половинним наближенням
5	Реалізувати програмно АЦП методом половинних наближень
6	Реалізувати програмно АЦП порозрядного врівноваження
7	Реалізувати програмно АЦП з первісним порозрядним урівноваженням
8	Реалізувати програмно АЦП слідкувального типу з первісним половинним наближенням
9	Реалізувати програмно АЦП половинних наближень

Контрольні питання

1. Назвіть відомі вам методи аналого-цифрового перетворення та основні типи АЦП.
2. Назвіть основні характеристики аналогових датчиків електричних і неелектричних величин.
3. Надайте характеристику процесу узгодження аналогових датчиків з мікропроцесорною системою.
4. Як визначаються параметри сигналу для нормалізації перед введенням в АЦП?
5. Дайте характеристику динамічних параметрів АЦП.

6. Поясніть фізичний зміст таких характеристик АЦП: час перетворення, час затримки запуску, час циклу перетворення, максимальна частота перетворення.

7. Які чинники впливають на значення похибки АЦП?

8. Наведіть приклади практичного застосування АЦП в електромеханічних системах.

9. Дайте характеристику побудови пристроїв АЦП з використанням мікросхем ЦАП.

Література: [1, с. 17–46; 2, с. 8–33; 3, с. 44–56, 7 с. 5–40].

Практичне заняття № 4

Тема. Програмне визначення тривалості часових інтервалів

Мета: набуття практичних навичок програмної реалізації вимірювання часових параметрів інформаційних сигналів.

Короткі теоретичні відомості

У системах автоматичного керування досить часто доводиться вимірювати такі величини, як частота f , період T , тривалість t , зсув фаз j різноманітних імпульсних сигналів. Для цього зазвичай застосовується перетворення «частота–код». Залежно від того, який саме параметр потрібно вимірювати, застосовують різні підходи. Під час вимірювання частоти f_x проводиться підрахунок імпульсів вхідного сигналу протягом фіксованого відрізка часу (рис. 4.1).

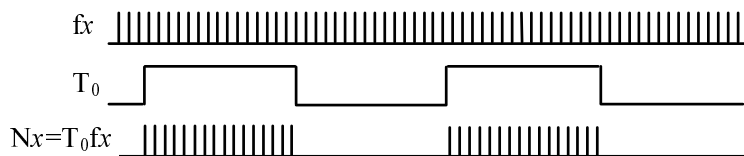


Рисунок 4.1 – Пояснення вимірювання частоти досліджуваного сигналу

Такий метод прийнятний для вимірювання частот, більших за 100 Гц.

Верхня межа обмежується швидкістю елементів схеми та розрядністю лічильників.

Вимірювання частот, нижчих за 100 Гц, замінюється вимірюванням періоду T_x . При цьому проводиться підрахунок імпульсів фіксованої частоти f_0 на інтервалі, що дорівнює або кратному вимірюваному періоду (рис. 4.2).

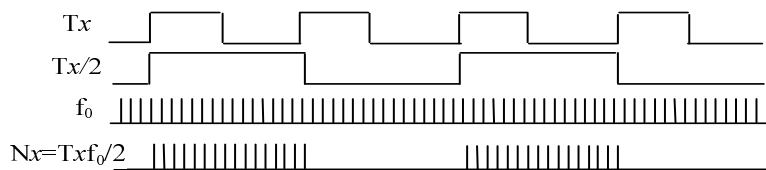


Рисунок 4.2 – Вимірювання періоду методом підрахунку кількості імпульсів відомої частоти

Так само вимірюється тривалість імпульсу τ_x , відмінність полягає лише в тому, що під час вимірювання тривалості імпульсу не потрібно ділити частоту на 2. У разі вимірювання зсуву фаз часовий інтервал τ_ϕ формують шляхом кон'юнкції сигналів f_{1x} і f_{2x} , отриманий інтервал часу вимірюють методом, описаним вище.

Розглянемо завдання визначення часу тривалості події, початок якого характеризується переходом зі стану логічної одиниці в стан логічного нуля двійкового сигналу У1, а кінець – переходом зі стану логічної одиниці в стан логічного нуля двійкового сигналу У2.

Для зручності виконання завдання подамо сигнал У1 на вхід першого запиту переривання МК, а сигнал У2 – на вхід нульового запиту переривання (рис. 4.3).

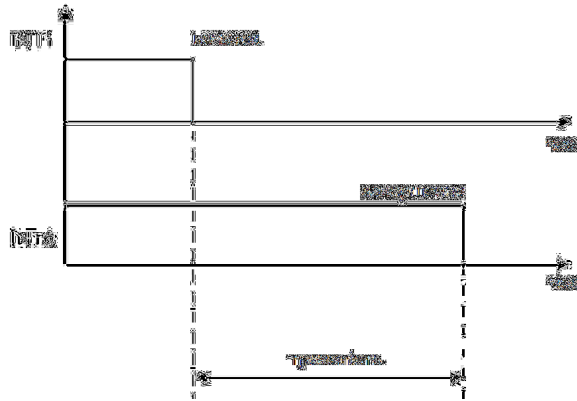


Рисунок 4.3 – Сигнали початку і закінчення події

Приклади виконання завдання

Реалізувати мовою асемблера для МК K1816BE51 програму, що вимірює тривалість імпульсу, який надходить на лінії введення/виведення порта P3.

Для зниження впливу підпрограм визначення часу події на інші процеси, що реалізуються в системі, слід вирішувати завдання з використанням системи переривань. За зрізом сигналу на вході INT1 виникає переривання, обробник якого запускає таймер, що відраховує кванти часу тривалості події, за зрізом сигналу на вході INTO виникає переривання, обробник якого припиняє рахунок, блокуючи таймер. Для підвищення точності визначення часу в ситуації можливих багатократних змін станів входів наступні переривання від джерел стартового і стопового імпульсів блокуються до завершення поточного циклу підрахунку часу.

Блок-схему алгоритму програмного комплексу у вигляді сукупності трьох підпрограм обробників переривань наведено на рис. 4.4, а їх текст у лістингу.

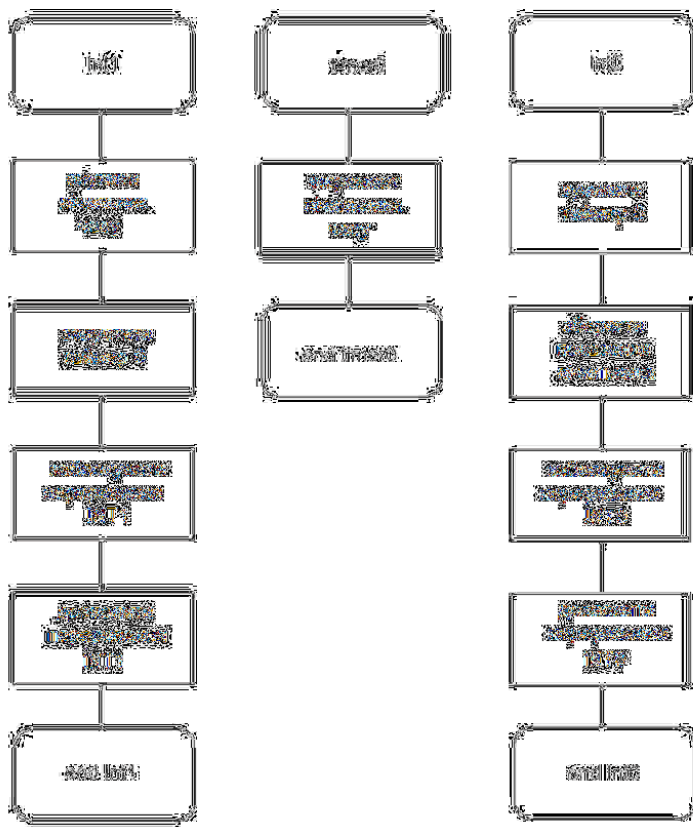


Рисунок 4.4 – Блок-схема програми визначення тривалості події

```

CNT_HI:      EQU 30h    ; лічильник поточного
CNT_LO:      EQU 31h    ; часу
LAST_TIM_H:  EQU 40h      ; тривалість
LAST_TIM_L:  EQU 41h      ; попереднього циклу
MAIN:        LCALL INIT   ; основна програма
LOOP: ...

                SJMP LOOP

INIT:          ; ініціалізація змінних і пристроїв:
                MOV CNT_HI,#0 ; онулення
                MOV CNT_LO,#0; лічильників
                MOV LAST_TIM_H,#0

```

```

MOV LAST_TIM_L,#0
MOV A,TMOD ; таймер T0 в режимі
ANL A,#11110000b ; перезавантаження
ORL A,#00000010b ; будемо вимірювати
MOV TMOD,A ; інтервали часу
MOV TH0,#155 ; в 100 мкс
MOV TL0,#155
SETB ET0 ; дозвіл роботи обробників
SETB EX1 ; необхідних
SETB EA ; типів переривань
RET

```

;-----

```

INT1: ; обробка події старт-імпульсу
MOV CNT_HI,#0 ; початок нового циклу
MOV CNT_LO,#0 ; вимірювань
MOV TL0,#155
SETB TR0 ; запуск таймера
SETB EX0 ; підготовка до розпізнання
; стоп-імпульсу
CLR EX1 ; блокування переривання цього типу
RETI

```

;-----

```

INT0: ; обробка події стоп-імпульсу:
CLR TR0 ; заборона подальшого рахунку
CLR EX0 ; блокування переривання цього типу
MOV LAST_TIM_H,CNT_HI ; збереження виміряного
MOV LAST_TIM_L,CNT_LO ; часу
SETB EX1 ; підготовка до розпізнавання
; чергового старт-імпульсу

```

```

INT0_EX:      RETI
              ;-----
TIM0:         PUSH PSW
              PUSH A
              INC CNT_LO ; збільшення значення
              MOV A,CNT_LO ; довжини імпульсу
              JNZ TIM0_EX
              INC CNT_HI
TIM0_EX:      POP A
              POP PSW
              RETI
              ;-----
; переходи до обробників
; необхідних типів переривань
ORG 0003h
              LJMP INTO
ORG 000Bh
              LJMP TIM0
ORG 0013h
              LJMP INT1

```

У змінних LAST_TIM_H і LAST_TIM_L зберігаються відповідно старший і молодший байти тривалості завершеної події у квантах часу по 100 мкс.

Завдання до теми

Написати програму мовою асемблера для МК1816BE51 з коментарями відповідно до варіанту (табл. 4.1).

Таблиця 4.1 – Варіанти індивідуальних завдань для самостійного виконання

№ пор.	Зміст індивідуального завдання
1	Реалізувати вимірювання частоти сигналів, що надходять за лінією T0
2	Реалізувати програмне вимірювання тривалості імпульса сигналів, що надходять за лінією T1
3	Реалізувати програмне вимірювання шпаруватості імпульсів, що надходять за лінією T1
4	Реалізувати програмне вимірювання коефіцієнта заповнення імпульсів, що надходять за лінією T0
5	Реалізувати програмне визначення моментів початку і закінчення імпульсів, що надходять за лінією INT0
6	Реалізувати програмне визначення моментів початку і тривалості імпульсів, що надходять за лінією INT1
7	Реалізувати програмне визначення моменту переходу сигналу з «1» в «0» на лінії INT0 і підрахувати кількість імпульсів, що надійшли
8	Реалізувати програмне визначення кількості імпульсів і безімпульсної паузи для сигналів, що надходять за лінією T0
9	Реалізувати програмне вимірювання тривалості імпульса, які надходить за лінією T1 і сформулювати інверсне значення цього імпульса на лінії P2.3
10	Реалізувати програмне вимірювання періоду сигналів, які надходять за лінією T0

Контрольні питання

1. Які методи частотного й часового перетворення сигналів ви знаєте?
2. Надайте характеристику параметрів тривалості сигналу.
3. Які чинники впливають на величину похибки вимірювання тривалості вимірюваного сигналу?

4. Якими способами можна в мікроконтролері реалізувати фіксацію початку і закінчення вимірюваного сигналу?

5. Як програмно реалізувати вимірювання періоду досліджуваного сигналу?

6. Дайте характеристику підсистеми реального часу мікроконтролера КР1816ВЕ51. Як здійснюється налаштування параметрів таймерів-лічильників для підрахунку інтервалів часу або кількості зовнішніх подій?

7. Як реалізуються апаратні засоби частотного й часового перетворення сигналів?

8. Наведіть приклади практичного застосування частотного й часового перетворення сигналів.

Література: [1, с. 17–46; 2, с. 8–33; 3, с. 44–56].

Практичне заняття № 5

Тема. Програмне виведення інформації з мікропроцесорної системи

Мета: набуття практичних навичок програмної реалізації виведення інформації на пристрої відображення в мікропроцесорних пристроях на базі МК К1816ВЕ51.

Короткі теоретичні відомості

У номенклатуру завдань виведення інформації в електромеханічних системах входять як завдання виведення керувальної інформації, що надходить у вигляді різних сигналів (амплітудних, частотних та ін.), на об'єкт управління, так і організація виведення інформації на пульт оператора.

У більшості вбудованих систем управління електроприводами необхідно організувати зв'язок з оператором для виведення інформації про параметри роботи електромеханічної системи.

Розглянемо завдання виведення алфавітно-цифрової інформації на рідкокристалічний алфавітно-цифровий індикатор (РКІ) моделі DM2021 фірми Sanyo. Ця модель РКІ має поле виведення інформації розміром два рядки по

двадцять символів у кожному. РКІ містить відеопам'ять, в якій зберігаються відображувані символи, а також має власну систему управління рідкокристалічною панеллю. Наявність зазначених вузлів істотно спрощує роботу з РКІ, оскільки власне управління відображенням точок – елементів зображення – здійснюється автоматично відповідно до поданої на РКІ команди.

РКІ містить восьмирозрядну шину команд-даних і шину управління, до складу якої входять однорозрядні лінії дозволу програмування (E), вибору типу посилки «команда-дані» (RS) і вибору напряму передачі даних «читання-запис» (RW). Підключення РКІ до МК 80C515 показано на рис. 6.8. Порт P4 призначений для організації шини команд-даних, а старші три біта порта P1 призначені для реалізації шини управління. Змінний резистор необхідний для регулювання контрастності зображення.

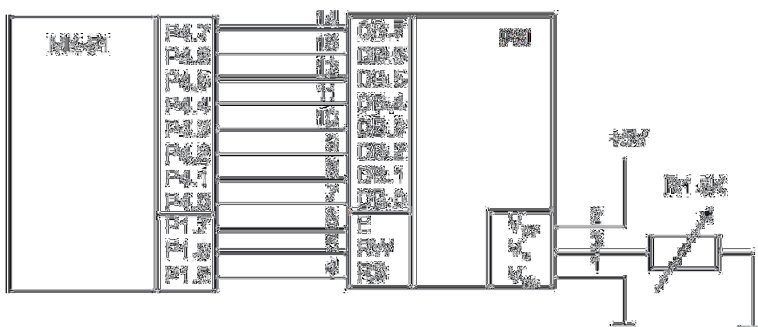


Рисунок 5.1 – Схема сполучення РКІ з МК-51

Процедура запису в РКІ виконується в три етапи: на шині DB встановлюється інформація, яка містить команди-дані, що подаються на РКІ, а потім встановлюються необхідні значення на лініях RW і RS, і, нарешті, на вході E формується перехід від високого логічного рівня до низького.

Для повернення системи управління РКІ в початковий стан слід перевести вхід E в стан логічної одиниці.

Без наведення всієї системи команд РКІ зупинимося на таких: 38h – встановлення 8-бітного режиму обміну з РКІ, використання для виведення

обох рядків з розміром символу 5×7 точок; 0Ch – активізація всіх знакомісць РКІ в режимі погашеного курсора; 80h – установлення адреси, починаючи з якої записуються до РКІ дані, які будуть послідовно розташовуватися у відеопам'яті.

Блок–схема алгоритму і текст програми, що здійснює виведення на РКІ сорока символів, коди яких розташовані в зовнішній пам'яті даних мікроконтролера, починаючи з адреси 0FFD0h, наведено відповідно на рис. 5.2 і в лістингу.

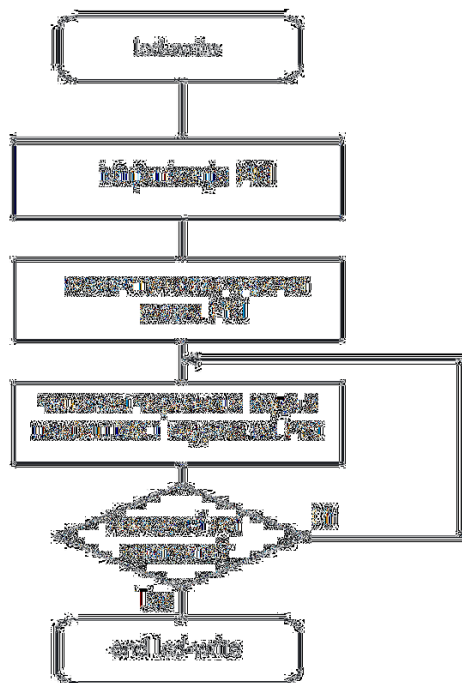


Рисунок 5.2 – Схема програми виведення інформації на РКІ

INDIC:

```

INDIC_INIT:  MOV INDIC_W1,#0    ; ініціалізація РКІ
              MOV INDIC_W0,#38h
              LCALL INDIC_WR
              MOV INDIC_W0,#0Ch
  
```

```

        LCALL INDIC_WR
        MOV INDIC_W0,#80h ; установлення адреси першого
        LCALL INDIC_WR ; символу першого рядка
        MOV INDIC_W1,#1
        MOV DPTR,#0FFD0h
INDIC_DATA_WR_1: MOVX A,@dptr ; виведення символів першого
        MOV INDIC_W0,A ; рядка
        LCALL INDIC_WR
        INC DPTR
        MOV A, DPL
        CJNE A,#0E4H, INDIC_DATA_WR_1
        MOV INDIC_W1,#0
        MOV INDIC_W0,#0C0h ; установлення адреси першого
        LCALL INDIC_WR ; символу другого рядка
        MOV INDIC_W1,#1
INDIC_DATA_WR_2: MOVX A,@DPTR ; виведення символів
        MOV INDIC_W0,A ; другого рядка
        LCALL INDIC_WR
        INC DPTR
        MOV A, DPL
        CJNE A,#0F8H, INDIC_DATA_WR_2
INDIC_EXIT: RET
INDIC_WR: MOV 0E8H, INDIC_W0 ; процедура запису в ПКІ
        SETB P1.7
        CLR P1.6
        MOV A,INDIC_W1
        MOV C,ACC.0
        MOV P1.5,C
        LCALL INDIC_DELAY

```

```

CLR P1.7
LCALL INDIC_DELAY
SETB P1.7
RET
INDIC_DELAY: NOP      ;процедура програмної затримки
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
RET

```

```
ORG 0FFD0h
```

```

DATA:   DB 'IT IS TEST EXAMLPE. ' ; приклад організації вивідних
        DB '0123456789ABCDEF!@#$' ; даних

```

Визначений виробником час витримки сигналів RW і RS щодо E складає долі мікросекунди, у зв'язку з чим затримка в процедурі запису в РКІ не обов'язкова. Проте час відпрацювання кожної з використовуваних у програмі команд в РКІ становить 40 мкс, у зв'язку з чим між поданими командами має бути витримана відповідна пауза. У цій програмі в процедуру видачі команд INDIC_WR включені затримки, що забезпечують сумарний час виконання процедури 36 мкс, а з урахуванням часу підготовки даних і часу виклику – необхідні 40 мкс.

Завдання до теми

Написати програму мовою асемблера для МК1816BE51 з коментарями відповідно до варіанту (табл. 5.1).

Таблиця 5.1 – Варіанти індивідуальних завдань для самостійного виконання

№.	Зміст індивідуального завдання
1	Занести до регістра R1 двійково-десятькове число XXh, віднімаючи від числа по 1, відображувати результат на рідкокристалічному індикаторі, реалізувати рядок, що «біжить» на нижньому рядку
2	Занести до регістра B двійково-десятькове число X0, до регістра R1 – XX, число з регістра B відобразити на рідкокристалічному індикаторі
3	Занести до R6 двійково-десятькове число XXh, до R5 – двійково-десятькове число XX, до R0 двійково-десятькове число XX, відобразити числа з R5, R6 на рідкокристалічному індикаторі
4	Почергово відобразити на РКІ числа від 0 до 9
5	Занести до A двійково-десятькове число 0X, до регістра R2 – X0, число з A відобразити на рідкокристалічному індикаторі, число з регістра R2 – у другому рядку РКІ з частотою 0,6 Гц
6	Занести до акумулятора двійково-десятькове число XX, до регістра R1 – XX, молодші два розряди суми чисел відобразити на рідкокристалічному індикаторі
7	Занести в регістр R6 число XXh, перетворити його на двійково-десятькове, відобразити його на динамічному індикаторі, відобразити значення R6 на світлодіодах HL1-HL8 і його інверсний стан із частотою 1 Гц
8	Занести до регістра B двійково-десятькове число XX, до регістра R3 – XX, різницю чисел відобразити на рідкокристалічному індикаторі
9	Занести до акумулятора число XX, до регістра R5 – X0, число з акумулятора відобразити на рідкокристалічному індикаторі
10	Занести до регістра R0 двійково-десятькове число XX, поперемінно відобразити молодший і старший розряди на рідкокристалічному індикаторі з частотою 0,5 Гц

Продовження таблиці 5.1

11	Занести до регістра R2 двійково-десятькове число XX, до регістра R5 – XX, їх суму відобразити на рідкокристалічному індикаторі
12	Занести до регістра B двійково-десятькове число, з частотою 2 Гц вивести його на рідкокристалічний індикатор
13	Занести до комірки з адресою 0010h число 0X, до регістра R3 – XX, суму чисел відобразити на рідкокристалічному індикаторі
14	Занести до регістра R1 двійково-десятькове число 0X, до регістра R3 – XX, суму відобразити на рідкокристалічному індикаторі
15	На рідкокристалічному індикаторі паралельно висвічувати кількість увімкнених світлодіодів. Інтервал між засвічуваннями – 1 с

Контрольні питання

1. Які способи управління пристроями відображення інформації в мікропроцесорних пристроях ви знаєте?
2. Поясніть особливості розрахунку часу регенерації для динамічного методу відображення інформації.
3. Дайте характеристику застосуванню різних методів відображення інформації.
4. Які схемотехнічні рішення використовуються для побудови схем відображення інформації?
5. Наведіть схеми підключення до ліній введення/виведення мікропроцесорної системи дискретних індикаторів.
6. Як здійснюється включення рідкокристалічних індикаторів?
7. Як здійснюється включення газорозрядних індикаторів?

Література: [1, с. 55–64; 2, с. 112–137; 3, с. 74–77].

Практичне заняття № 6

Тема. Розробка прикладного програмного забезпечення мікропроцесорних пристроїв керування електромеханічним обладнанням

Мета: набуття практичних навичок створення програмного забезпечення мікропроцесорних систем керування електромеханічним обладнанням.

Короткі теоретичні відомості

Система команд МК K1816BE51 складається зі 111 базових команд, більшість з яких (94) мають формат 1 або 2 байти і виконуються за 1 або 2 машинних цикли (99 команд). МК обробляє операнди 4-х типів: біти, 4-бітові цифри, байти і 16-бітові слова. Для доступу до даних використовується 4 способи адресації: пряма, безпосередня, непряма і регістрова.

Оригінальний текст програми на мові асемблера має певний формат. Кожний рядок команди мовою асемблера може містити до чотирьох полів (необов'язкові елементи рядка взяті у квадратні дужки []):

[МІТКА:] ОПЕРАЦІЯ [ОПЕРАНД(И)] [;КОМЕНТАР].

Поля можуть відділятися один від одного довільним числом прогалін. У полі мітки розміщується символічне ім'я комірки пам'яті, в якій зберігається зазначена команда або операнд.

Приклади розв'язання завдань

У ході технологічного процесу на конвеєрі виробу автоматично упаковуються в коробки. Одна коробка має містити 500 виробів. Потрібно:

- 1) кожні 120 мкс подавати сигнали на завантаження в коробку одного виробу (момент відвантаження перехід 1→0);
- 2) по закінченні завантаження коробки здійснити подачу сигналу на упаковування коробки та переміщення її далі по конвеєру (перехід 1→0);
- 3) зробити паузу на час подачі наступної коробки (не менше 2 с), після чого продовжити завантаження.

Розв'язання

Постановку задачі програмування можливо сформулювати так: для підрахунку числа виробів потрібний програмний лічильник розрядністю не менше, ніж 9 біт (2^9 дає 512 значень). Для формування часових інтервалів використовуємо таймер/лічильник 0. Дані часового інтервалу дозволяють використовувати таймер/лічильник у режимі 2 як таймер. Після установки прапорця переповнення таймера зовнішній вихід повинен перейти зі стану «1» у стан «0». Для керування процесом відвантаження виробу використовуємо лінію P1.0. Коли лічильник дорахує до 500, то рахунок за таймером 0 припиняється, подається сигнал на переміщення по конвеєру (P1.1). За допомогою лічильника/таймера 1 реалізується затримка 2 с. Оскільки максимальний коефіцієнт рахунку в режимі 1 дорівнює 0FFFFh, то час переповнення дорівнює 0,065536 с, тому слід виконати повний цикл прорахунку 31 раз, що складає 2,032 с. Після цього продовжити видачу імпульсів на вході P0.0.

Лістинг програми:

```
NAME Real_pprocess
LOAD EQU 90H                ;Опис символічних імен, що підставляються
NEXT_BOX EQU 91H
PROG SEGMENT CODE          ;Опис сегментів
STACK SEGMENT IDATA
FLAGS SEGMENT BIT
RSEG STACK                  ;Сегмент стека
    DS 10H
RSEG FLAGS                  ;Сегмент прапорців
    COUNT_H: DBIT1          ;Старший розряд лічильника циклів
    END_LOAD: DBIT1         ;Прапорець закінчення циклу завантаження
    CSEG AT 0                ;Код початку програми
    USING    0                ;Банк 0
```

```

JMP START           ;Початок програми
CSEG      AT 0BH    ;Код оброблювача переривання таймера 0
USING 1           ;Використовуємо банк регістрів 1
SETB RS0         ;Переключення банку
CLR LOAD         ;Подаємо імпульс на завантаження виробу
MOV R1,A         ;Зберігаємо вміст А
MOV A,R0         ;У R0 молодший байт лічильника циклів
JB COUNT_H,LAB1  ;Перехід, якщо в старшому розряді
                  ;лічильника 1
ADD A,#1         ;Інкремент акумулятора
JNC LAB2         ;Перехід , якщо немає переповнення
SETB      COUNT_H ;Установка старшого розряду лічильника
SJMP LAB2       ;Перехід на завершення оброблювача
LAB1:
INC A           ;Інкремент А
CJNE A,#0F4H   ;Перехід, якщо молодший байт лічильника не
LAB2:          ;дорівнює 0F4h (500 – це 1 F4h)
CLR COUNT_H   ;Очищення старшого розряду лічильника
CLR A         ;Очищення акумулятора
SETB END_LOAD ;Установка прапорця кінця циклу завантаження
LAB2:
MOV R0,A      ;Збереження нового значення лічильника
MOV A,R1      ;Відновлення А після переривання
CLR RS0       ;Переключення на банк регістрів 0
RETI          ;Повернення з обробки переривання
*****Основна програма *****
RSEG PROG
START:  MOV SP,#STACK-1
        CLR COUNT_H      ;Початкова ініціалізація змінних

```

```

CLR END_LOAD
MOV 08H,A           ;Занесення в регістр R0 банка 1 нуля
MOV TMOD,#12H      ;Програмування режимів таймерів
                   ; «0» – режим 2 , «1» – режим 1
MOV IE,#82H        ;Дозвіл переривань за таймером T/C0
LAB3:
MOV TH0,#135       ;Ініціалізація рахункових регістрів таймера 0
MOV TL0,#135
MOV TCON,#10H      ;Запуск таймера T/C0
LAB4:
SETB LOAD          ;Установка виходу P1.0 у стан «1»
JNB END_LOAD,LAB4  ;Якщо END_LOAD дорівнює «1», то перехід за
                   ;міткою LAB4
CLR END_LOAD       ;Очищення прапорця закінчення циклу
MOV TCON,#0        ;Відключення таймерів
CLR NEXT_BOX       ;Подача сигналу на переміщення коробки
MOV TH1,#0         ;Ініціалізація рахункових регістрів таймера 1
MOV TL1,#0
MOV TCON,#40H      ;Запуск таймера T/C1
CLR A              ;Очистити в акумуляторі лічильник числа
                   ;переповнень таймера T/C1
LAB5:
JNB TF1,LAB5       ;Очікування встановлення прапорця
                   ;таймера T/C1
CLR TF1           ;Очищення прапорця таймера T/C1
INC A             ;Інкремент А
CJNE A,#31,LAB5   ;Для досягнення часу 2 с число проходів
                   ;дорівнює 31
SETB NEXT_BOX     ;Повернення виходу переміщення коробки

```

```

;у стан «1»
MOV TCON,#0      ;Вимикання таймера
SJMP LAB3        ;Повторення технологічного циклу
END

```

Завдання до теми

Скласти докладний алгоритм та написати керувальну програму мовою асемблера для мікроконтролерів K1816BE51 для заданого опису технологічного процесу (табл. 6.1). Усі програмні конструкції повинні бути докладно прокоментовані.

Таблиця 6.1 – Вихідні дані для самостійного виконання

№ варіанту	Зміст завдання
1	Температура води в робочій машині регулюється шляхом релейної зміни витрати через теплообмінник. Зі зростанням температури вище +90 °С витрата збільшується шляхом відкриття додаткового клапана. У разі зменшення температури нижче +90 °С цей клапан закривається через хвилину. Увімкнення і вимикання насоса виконується кнопками
2	У термопечі ТВЧ обробляють 3 види деталей. Режим обробки вибирається перемикачем. Увімкнення нагрівання виконується кнопкою, вимикання – за часом, залежно від типу деталі. Після вимикання нагрівання на 30 секунд вмикається інтенсивне повітряне охолодження, потім протягом 3 секунд звучить сигнал, що дозволяє відкриття печі
4	Ліфт обслуговує 2 поверхи. Керування ручне. У середині кабіни – кнопки з номерами поверхів, на поверхах – кнопки виклику. У разі перевантаження ліфт не рушає з місця і загорається сигнальна лампа

Продовження таблиці 6.1

5	Робоча машина має 3 приводи. Другий привод не можна увімкнути, поки не увімкнений перший, третій не можна увімкнути, поки не увімкнений другий. Вимикання виконується від однієї кнопки «СТОП», спочатку вимикається перший привод, через 30 секунд – другий, ще через 30 секунд – третій
6	Температура повітря в приміщенні регулюється шляхом увімкнення й вимикання додаткового вентилятора. Вентилятор повинен вмикатися, якщо температура перевищує +45°C. Вимикання вентилятора відбувається при температурі +30 °C. після витримки 1 хв
7	Рольганг з електроприводом подає сталевий лист до гільйотинних ножиців. За ножицями установлений упор із кінцевим вимикачем, що подає сигнал на зупинку рольганга й відрізання листа. Далі подача листа повторюється. Увімкнення привода рольганга відбувається у разі виявлення листа на першому ролику, вимикання – від кнопки чи за відсутності листа перед гільйотиною протягом 30 секунд
8	Рольганг з електроприводом подає деталі двох типів на кантувач, який деталь першого типу перевертає ліворуч, а деталь другого типу – праворуч. Деталі завжди подаються по чергово. Увімкнення привода рольганга відбувається у разі попадання чергової деталі на перший ролик, вимикання – при заході деталі в кантувач. Схема повинна керувати рольгангом і кантувачем із дотриманням необхідних блокувань
9	Подача свердлильної головки вниз відбувається на двох швидкостях – спочатку на прискореній подачі, потім за сигналом шляхового датчика – на робочій подачі. Після закінчення свердління відбувається підйом нагору на прискореній подачі.

Продовження таблиці 6.1

10	Ворота цеху відкриваються при натисканні на кнопку і закриваються після витримки 50 секунд. За наявності будь-якого предмета у створі воріт закриття припиняється, ворота повертаються у вихідне положення, лунає дзвоник (5 секунд) і ворота залишаються відкритими до натискання кнопки
11	Рольганг з електроприводом подає деталі двох типів на кантувач, який деталь першого типу перевертає ліворуч, а деталь другого – праворуч. Деталі завжди йдуть у такій послідовності: ААБААБ. Увімкнення привода рольганга відбувається у разі потрапляння чергової деталі на перший ролик, вимикання – у разі заходу деталі в кантувач. Схема повинна керувати рольгангом і кантувачем з дотриманням необхідних блокувань
12	Вантажна платформа перевозить вантаж із точки А в точку В. Керування платформою – ручне, з пульта, встановленого в точці А. Якщо за 5 хвилин після прибуття вантаж не знятий із платформи, відбувається автоматичне повернення в точку А. Якщо за цей час вантаж зняти, то відправлення платформи в точку А відбуваються через хвилину після зняття вантажу. Перед відправленням платформи 5 секунд лунає дзвоник
13	У термопечі ТВЧ обробляються 3 види деталей. Режим обробки вибирається перемикачем. Увімкнення нагрівання виконується кнопкою, вимикання – за часом, залежно від типу деталі. Після вимикання нагрівання на 30 секунд вмикається інтенсивне охолоджувальний вентилятор, потім протягом 3 секунд звучить сигнал, що дозволяє відкриття печі

Продовження таблиці 6.1

14	Вантажна платформа перевозить вантаж із точки А в точку В чи в точку С. Керування платформою ручне, з пульта, встановленого в точці А. Вибір кінцевого пункту за допомогою перемикача. Якщо за 5 хвилин після прибуття вантаж не зняти з платформи, відбувається автоматичне повернення в точку А. Якщо вантаж зняти у зазначений час, то відправлення платформи в точку А відбуваються через хвилину після зняття вантажу. У будь-якому випадку перед відправленням 5 секунд лунає дзвоник
15	Ліфт обслуговує 2 поверхи. Керування ручне. У середині кабіни – кнопки з номерами поверхів, на поверхах – кнопки виклику. У разі перевантаження ліфт не рушає з місця і загоряється сигнальна лампа
16	Забезпечити на виводах 0–3 порта виведення чотирифазну систему напруги для керування кроковим двигуном. Імпульси тривалістю 10 мс, зсунуті один відносно одного на 5 мс. Тактова частота 12 МГц

Контрольні питання

~~1~~.1. Назвіть основні етапи розробки програмного забезпечення мікропроцесорної системи.

~~2~~.2. Прокоментуйте кожний етап розробки програмного забезпечення. Які етапи і в яких ситуаціях можуть бути пропущені?

~~3~~.3. Що називається алгоритмом роботи керуючої програми?

~~4~~.4. У чому полягає алгоритмізація вирішення задач керування промисловими об'єктами?

~~5~~.5. Перерахуйте основні способи запису алгоритмів роботи програми.

~~6~~.6. Приведіть графічні елементи блок-схем алгоритмів.

Література: [1, с. 17–46; 2, с. 8–33; 3, с. 44–56].

Формат: Список

3 КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

Оцінювання роботи студентів на практичних заняттях бере до уваги якість теоретичної підготовки, оформлення результатів практичних занять, якість відповідей на контрольні запитання, правильність і повноту відповідей на запитання, поставлені викладачем під час захисту практичних робіт.

Бали за виконання практичних робіт у цілому можуть бути одержані лише за умови, що виконано всі пункти, передбачені завданням до практичної роботи та складений звіт.

Кількість балів визначається глибиною наведених пояснень, чіткістю формулювань, якістю викладання матеріалу та його оформлення. Ураховуються правильність оформлення звіту, володіння державною мовою, відповідність розрахункового матеріалу та іншого оформлення діючим стандартам. Розподіл балів по практичних роботах з навчальної дисципліни «Мікропроцесорні пристрої» наведено нижче.

Критерії оцінювання роботи студентів на практичних заняттях:

- відвідування практичного заняття – 0,5 бала за заняття;
- підготовка до практичних занять – 0,1 бала за семестр;
- виконання практичних робіт – 0,1 бала за семестр;
- підготовка звітів з практичних робіт – 0,1 бала за семестр;
- захист практичних робіт – 1 бал за роботу;
- своєчасність захисту практичних робіт – 0,1 бала за семестр;
- активність студента під час виконання практичних занять – 0,1 бала

за семестр.

Формат: Список

СПИСОК ЛИТЕРАТУРИ

1. Болл С. Р. Аналоговые интерфейсы микроконтроллеров / С. Р. Болл. – М. : Издательский дом «Додэка–XXI», 2007. – 360 с.
2. Баранов В. Н. Применение микроконтроллеров AVR: схемы, алгоритмы, программы / В. Н. Баранов. – М. : Издательский дом «Додека–XXI», 2004.– 288 с.
3. Белов А. В. Конструирование устройств на микроконтроллерах / А. В. Белов. – СПб. : Наука и техника, 2005. – 256 с.
4. Белов А. В. Микроконтроллеры AVR в радиолюбительской практике / А. В. Белов. – СПб. : Наука и техника, 2007. – 352 с.
5. Белов А. В. Самоучитель разработчика устройств на микроконтроллерах AVR / А. В. Белов. – СПб. : Наука и Техника, 2008. – 544 с.
6. Белов А. В. Создаем устройства на микроконтроллерах / А. В. Белов. – СПб. : Наука и Техника, 2007. – 304 с.
7. Вальпа О. Д. Полезные схемы с применением микроконтроллеров и ПЛИС / О. Д. Вальпа – М. : Издательский дом «Додэка–XXI», 2006. – 416 с.
8. Волович Г. И. Схемотехника аналоговых и аналого-цифровых электронных устройств / Г. И. Волович. – М. : Издательский дом «Додэка–XXI», 2005. – 528 с.
9. Евстифеев А. В. Микроконтроллеры AVR семейств Tiny и Mega фирмы «ATMEL» / А. В. Евстифеев. – М. : Издательский дом «Додэка–XXI», 2004. – 560 с.
10. Рюмик С. М. Интерфейс I2C. Технические подробности / С. М. Рюмик // Радиоаматор. – 2004. – № 1, 2. – С. 35–39, 29–31.
11. Точки Р. Д. Цифровые системы. Теория и практика / Р. Д. Точки, Н. С. Уидмер; пер. с англ. – М. : Издательский дом «Вильямс», 2004. – 1024 с.

Методичні вказівки щодо практичних занять з навчальної дисципліни «Мікропроцесорні пристрої» для студентів денної та заочної форм навчання зі спеціальності 141 – «Електроенергетика, електротехніка та електромеханіка» (у тому числі скорочений термін навчання) освітнього ступеня «Бакалавр»

Укладач старш. викл. М. Ю. Юхименко

Відповідальний за випуск зав. кафедри САУЕ проф. Д. Й. Родькін

Підп. до др. _____. Формат 60×84 1/16. Папір тип. Друк ризографія.

Ум. друк. арк. _____. Наклад _____ прим. Зам. № _____. Безкоштовно.

Видавничий відділ
Кременчуцького національного університету
імені Михайла Остроградського
вул. Першотравнева, 20, м. Кременчук, 39600