

Міністерство освіти і науки України

ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ  
УНІВЕРСИТЕТ

Кафедра “Автоматизації виробничих процесів”

## **“Мікропроцесори та мікро-ЕОМ”**

### **Методичні вказівки**

для виконання лабораторних робіт  
для студентів спеціальності: 141- Електроенергетика,  
електротехніка та електромеханіка

Затверджено на засіданні кафедри  
«Автоматизація виробничих процесів»  
Протокол №1 від 28 серпня 2020 р.

Кропивницький 2020 р.

Мікропроцесори та мікро-ЕОМ/ Методичні вказівки для виконання лабораторних робіт для студ. спец.: 141- Електроенергетика, електротехніка та електромеханіка, /[Уклад. І.І. Єніна, І.В. Волков]; М-во освіти і науки України, Центральноукраїн. нац. техн. ун-т, каф. автоматизації вироб. процесів. – Кропивницький: - 2020. – 63с.

## ***Зміст***

Вступ

Робота із стендом. ....	10
Приклад завантаження програми в стенд .....	11
Список умовних скорочень .....	12
Лабораторна робота №1. Вивчення стенду, команд однокристальної ЕОМ сімейства MCS-51.....	14
Лабораторна робота №2. Способи побудови схем відображення інформації .....	17
Лабораторна робота №3. Система переривань. Опит дискретних датчиків .....	26
Лабораторна робота №4. Цифр–аналогове перетворення .....	34
Лабораторна робота №5. Аналого-цифрове перетворення .....	40
Лабораторна робота №6. Обробка частотних і часових сигналів .....	46
Лабораторна робота №7 Вивчення інтегральних мікросхем пам'яті з послідовним інтерфейсом .....	54
Лабораторна робота №8. Програмування послідовного порту ОЕОМ .....	59
Перелік літератури .....	60

\

### **Вступ.**

Стенд являє собою мікропроцесорний контролер, оснащений пам'яттю програм, пам'яттю даних і різноманітними периферійними пристроями. Він дозволяє налагоджувати програми, написані на мовах Сі й Асемблер.

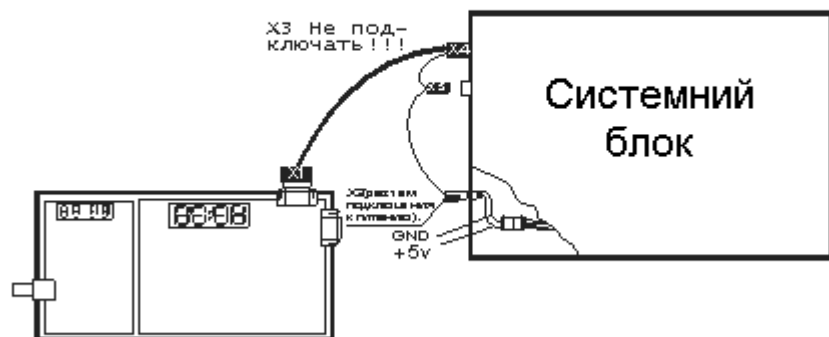
Завантаження програми відбувається з персонального комп'ютера по послідовному порту RS-232C. Підключення напруги живлення стенда (+5 вольт) і зв'язок з персональним комп'ютером здійснюється за допомогою одного універсального кабелю зв'язку з комп'ютером (DB9+DB25+живлення).

### **Схема підключення УНС “EV8031” до ПК. Живлення стенда.**

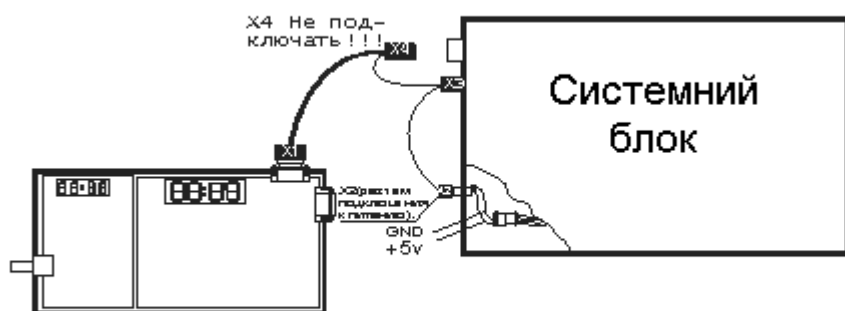
Зв'язок учбово-налагоджувального стенда “EV8031” з ПК здійснюється через COM-порт. У комплекті зі стендом ПНЗтавляється універсальний з'єднувальний кабель із двома роз'ємами (25 й 9 pin) для з'єднання з одним із COM -портів. Крім того, є роз'єма підключення до комп'ютерного блоку живлення й роз'єм для підключення стенда. Роз'єм підключення живлення стенда зручно кріпитися на задній панелі системного блоку комп'ютера й тому не виникає необхідності постійно знімати кришку системного блоку.

**Увага!!!** Для справної й правильної роботи стенда, його необхідно підключати **тільки з одним** com-портом ПК. Не рекомендується від'єднувати стенд від ПК при включеному живленні. Те ж саме стосується від'єднання плати розширення від стенда.

# Схема підключення стенда EV8031/AVR до ПК+живлення



Вариант включения 1



Вариант включения 2

Структурна схема стенда представлена на рис. 1.

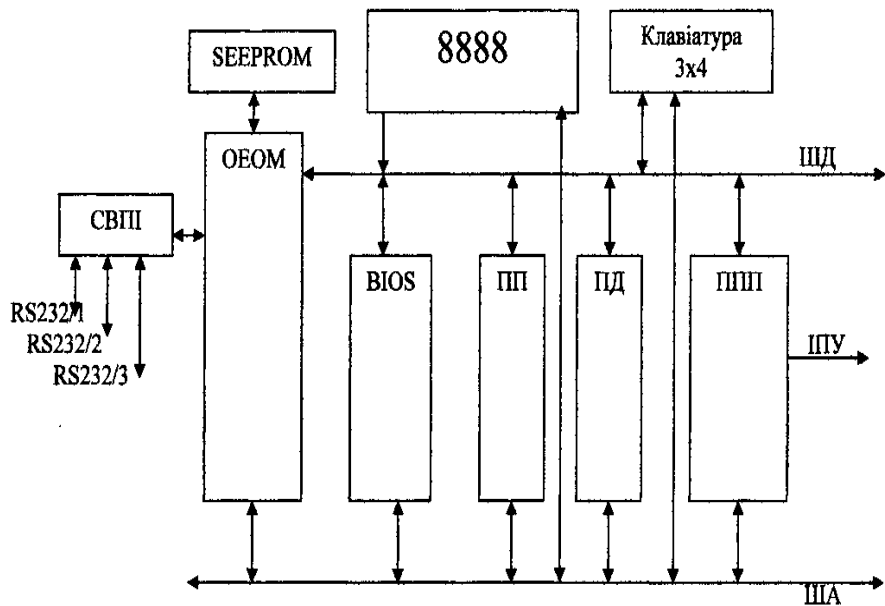


Рис.1.Структурна схема стенда

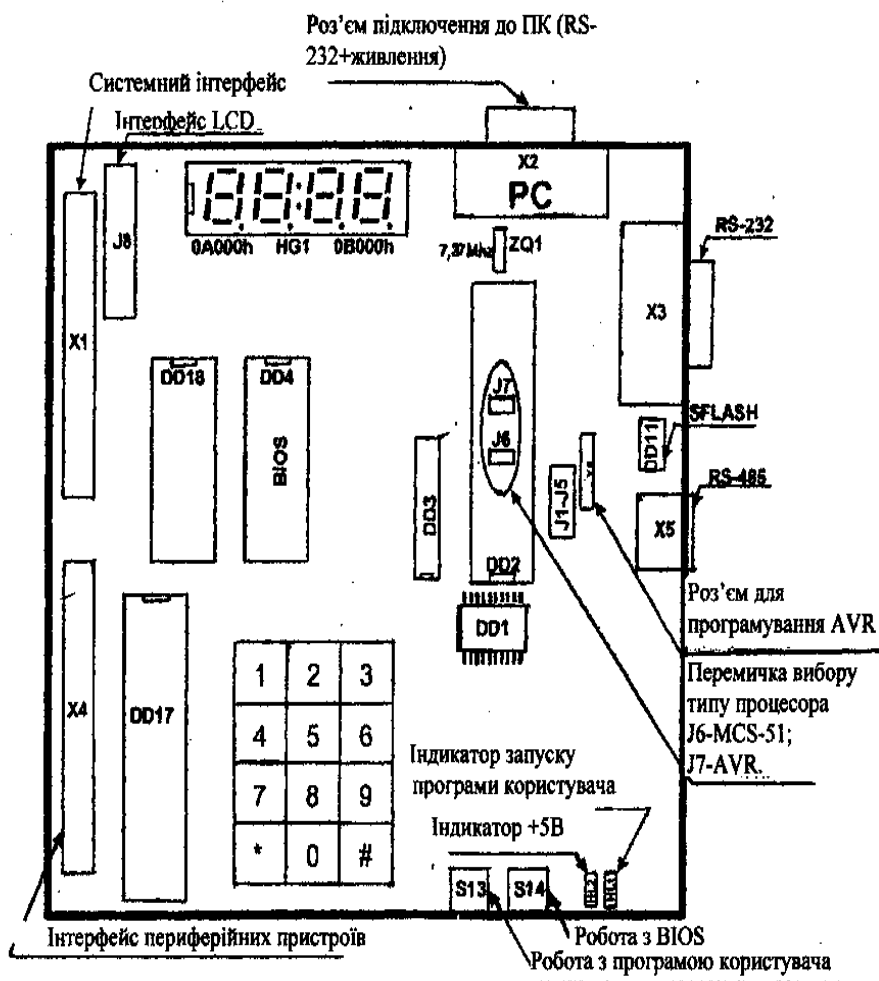


Рис. 2.Схема розташування елементів стенда

## Роз'єм для програмування AVR



Рис. 3Схема роз'єму для програмування AVR.

На рис. 3 представлена схема для програмування AVR контролера. Номерами позначені контакти роз'єму:

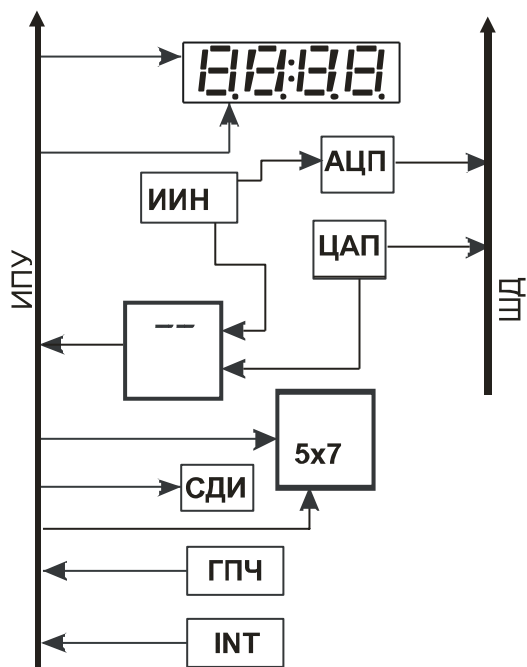
1. MOSI.
2. MISO.
3. CLK.
4. RESET.
5. GND.
6. VCC.

На кабелі програмування AVR контролера також позначені відповідні контакти.

Структурна схема плати розширення представлена на рис.4. Плата містить набір пристроїв, які дозволяють освоїти основні прийоми програмування. Курс лабораторних робіт складається з дев'яти робіт, до кожної роботи додається спрощена функціональна схема стенда. Спрощена схема містить тільки ті елементи, які необхідні для проведення даної лабораторної роботи (рекомендується також дивитися загальну принципову схему).

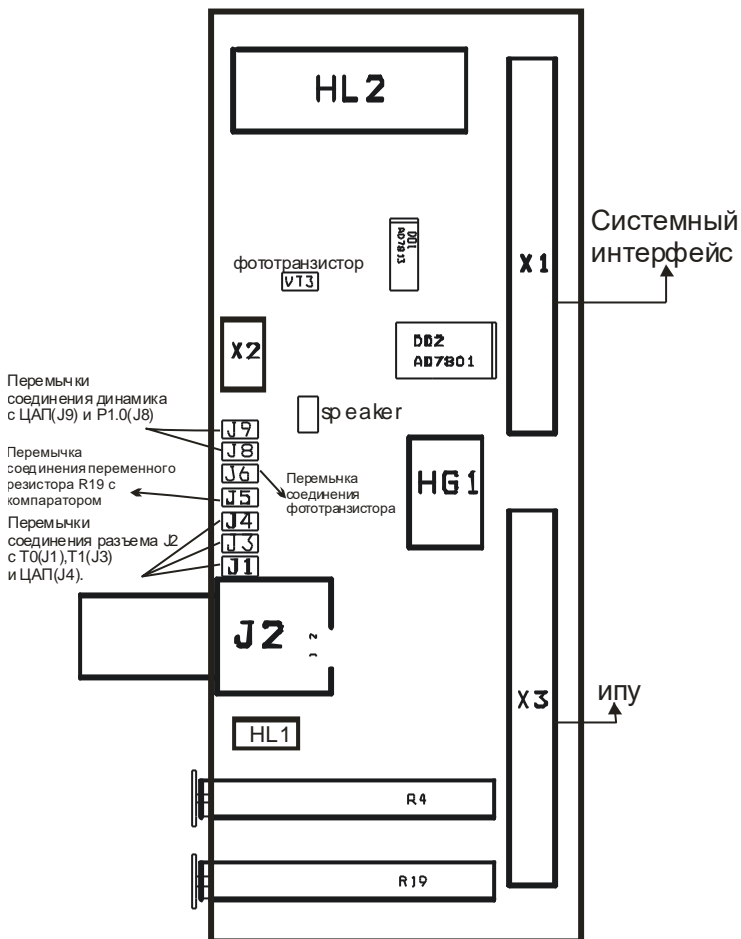
Розташування основних органів керування, і мікросхем на платі наведені на рис. 2 та рис. 4.





8888 - 4 розрядна динамічна індикація;  
 ІПП- Інтерфейс периферійних пристроїв;  
 ЦАП – Цифро-аналоговий перетворювач;  
 СДИ - Світлодіодні індикатори;  
 ЗСИ - Знакосинтезуючий індикатор 5x7;  
 ГЗЧ - Генератор змінної частоти генерації;  
 ІНТ - Кнопки запиту переривання;  
 ДВН - джерело вимірюваної напруги;  
 ШД - шина даних.

Рис. 4. Структурна схема плати розширення



HG1 - знакосинтезуючий індикатор 5x7;  
 HL2 - 4-х розрядна динамічна індикація;  
 J2 - роз'єм підключення зовнішніх контрольно-вимірювальних приладів.  
 R4 - змінний резистор, джерело вхідного сигналу для АЦП;  
 R19 - змінний резистор, змінює частоту генерації змінного генератора;  
 HL1 - індикатор стану виходу компаратора.

Рис.5.Схема розташування елементів плати розширення

### **Робота зі стендом**

У цьому розділі розглянутий один з варіантів завантаження програми в стенд (див. додаток №1).

1. На персональному комп'ютері завантажити текстовий редактор.
2. У текстовому редакторі набрати текст програми в мнемокодах мови Асемблер (мови Сі) для 8051.
3. Зберегти набраний файл із розширенням \*.ASM (\*.C).
4. Відкопіювати набрану програму відповідними засобами.
5. Можливі помилки в програмі можна переглянути в однойменному файлі з розширенням \*.LST
6. Після усунення всіх помилок, дані файлу з розширенням \*.HEX програмою EVAL32.EXE необхідно перенести в стенд. Вивід на екран підказки про параметри програми EVAL32.EXE, здійснюється запуском EVAL32.EXE.
7. При передачі даних з персонального комп'ютера в стенд на екрані монітора відображаються передані дані. Ці ж дані відображаються на індикаторі стенда. Горить світлодіод HL9.
8. Зупинка, завантаженої програми й перехід у режим очікування на прийом даних з персонального комп'ютера можливо натисканням кнопки SW2. При цьому гасне світлодіод HL9.
9. Запис нової програми можливий в будь-який момент часу роботи завантаженої програми.

### Приклад завантаження програми в стенд

1. На персональному комп'ютері завантажити текстовий редактор.
2. Уміст CD переписати на диск ІЗ
3. У текстовому редакторі набрати текст програми в мнемокодах мови Асемблер для 8051:

```
ORG 0
Start:
mov DPTR, #0A000h      ; занести в регістр DPTR
                        ; адреса індикації
mov A, #28h            ; занести в регістр A дані для відображення
movx @DPTR, A          ; винести на індикацію, адресу якої
                        ; перебуває в регістрі DPTR число 28
jmp Start              ; перехід на початок
END
```

4. Зберегти набраний файл із розширенням \*.ASM.  
(Наприклад IND.ASM)
5. Відкопіювати набрану програму можливими засобами.  
(...\ASM51...EXEIND.ASM)
6. Дані файлу з розширенням \*.HEX перенести в стенд.  
(...\EVAL32...EXE-hs-com29600IND.HEX).

✓ *Довідку про параметри програми завантаження даних у стенд із ПК можна переглянути, запустивши програму EVAL32.EXE без параметрів.*

## Перелік умовних скорочень

Скор назва.	Опис	Познач.
ОЕОМ	Однокристална ЕОМ	
Т/Л	Таймер лічильник ОЕОМ	
ПНС	Програмно-налагоджувані засоби	
ІМС	Інтегральна мікросхема	
ШД	Шина даних	
ША	Шина адреси	
ДД	Двійково-десятькове число	
ПК	Персональний комп'ютер	
R0-R7,A,B	Регістри загального призначення ОЕОМ	
XXH	Шістнадцяткове число	
0X	Число в якому старша тетрада –"0", молодша будь-яке число від нуля до шістнадцяткового F	
X0	Число в якому старша тетрада будь-яке число від нуля до шістнадцяткового F, молодша –"0"	
HL1-HL8	Індикатори	HL1,HL2...HL8
C_Інд.	Елемент статичної індикації	HG1.0 HG1.1 HG1.2 HG1.3
D_Інд.	Елемент динамічної індикації	HL2
ЗС_Інд.	Знакосинтезуюча індикація	HG1
ГФЧ	Генератор фіксованої частоти	DD18-1,18-2,18-3
ГЗЧ	Генератор змінної частоти	DD1.1
КЛ	Клавіатура 3*4	SW3-SW14
КН1, КН2	Кнопка 1, 2	SW15, SW16
485	485 інтерфейс	DD11

## Карта портів вводу/виводу стенда

Адреса	Тип циклу	B7	B6	B5	B4	B3	B2	B1	B0	Ім'я
Порти периферійних пристроїв										
8xx0	Запис	[Порт А]								PA_REG
8xx1	Запис	[Порт В]								PB_REG
8xx2	Запис	[Порт С]								PC_REG
8xx3	Запис	x	x	x	x	x	TRISC	x	x	TRIS
PKI										
8xx4	Запис	Регістр команд РК індикатора								LCD_CMD
8xx5	Запис	Регістр даних РК індикатора								LCD_DATA
ПНЗлідовний порт										
9xxx	Читання	CTS	DSR	DCD	RI	KL3	KL2	KL1	KL0	US_REG
Cxx0	Запис	x	x	X	x	DTR	RTS	CFG1	CFG0	UC_REG
Індикатор і світлодіоди										
Axx0	Запис	[Регістр індикатора 0]								DISPLAY[0]
Axx1	Запис	[Регістр індикатора 1]								DISPLAY[1]
Axx2	Запис	<зарезервовано>								DISPLAY[2]
Axx3	Запис	<зарезервовано>								DISPLAY[3]
Axx4	Запис	DP3	DP2	DP1	DP0	BL3	BL2	BL1	BL0	DC_REG
Axx5	Запис	<зарезервовано>								EDC_REG
Axx6	Запис	LED7	LED6	LED5	LED4	LED3	LED2	LED1	LED0	LED_REG
Управління роботою										
Axx7	Запис	x	x	X	x	x	x	x	RUN	SYS_CTL
Сумісні регістри										
Bxx0	Запис	[Регістр індикатора 1]								DISPLAYB

## Лабораторні роботи №1-№8

### Лабораторна робота №1. Вивчення стенда, команд однокристалльної ЕОМ сімейства MCS-51

**Тема:** Вивчення стенда, команд однокристалльної ЕОМ сімейства MCS-51.

**Мета роботи:** Вивчення функціональних можливостей учбово-налагоджувального стенда, внутрішньої структури й системи команд ОЕОМ сімейства MCS-51.

**Навчальне завдання:** Вивчення команд передачі даних , арифметичних, логічних команд, команд передачі управління. Взаємодія внутрішніх вузлів ОЕОМ.

#### *Порядок виконання лабораторної роботи:*

- 1) Вивчити структурну схему стенда, розподіл пам'яті, призначення вузлів. Вивчити структуру ОЕОМ сімейства MCS-51. Вивчити синтаксис команд пересилання, арифметичних команд, команд переходів.
- 2) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття (таблиця завдання №1).
- 3) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття.
- 4) Ввести програму індивідуального завдання в ПК.
- 5) Вивчити програмно налагоджувальні засоби (ПНЗ) для MCS-51.
- 6) За допомогою ПНЗ проаналізувати виконання індивідуальної програми.
- 7) Завантажити програму в стенд ОЕОМ. Переконавшись в правильному виконанні індивідуального завдання, при негативному результаті здійснити зміну алгоритму або програми. Повторити завантаження програми в стенд ОЕОМ
- 8) Роздрукувати листинг правильно працюючої програми.
- 9) Відповісти на контрольні питання викладача

#### *Контрольні питання:*

1. Час виконання команд (поняття такту, машинного циклу)
2. Типи команд. Формат команд. Команди прямої й непрямої адресації.
3. Команди зсуву. Арифметичні команди. Логічні команди.
4. Регістр ознак, команди викликаючі зміну регістра ознак
5. Команди роботи зі стеком, ПНЗлідовність дій (команд) при роботі зі стеком
6. Призначення внутрішніх вузлів ОЕОМ
7. Призначення й робота із внутрішньою пам'яттю даних ОЕОМ
8. Система переривання ОЕОМ. Призначення портів ОЕОМ
9. Фізичні характеристики вихідних сигналів ОЕОМ

## Короткі теоретичні відомості Додаток №4

### Приклад виконання лабораторної роботи №1

#### Вміст регістрів R1 й R4 скласти й відобразити на індикаторі

```
ORG 0
mov R1,#04h ;записати в R1 число 04
mov R4,#30h ;установити в R4 число 30
mov DPTR,#0B000h ;установити в DPTR адреса Інд. DD17, DD18
mov A,R1 ;записати в A значення R1
add A,R4 ;скласти значення в A и R4 результат суми в A
movx @DPTR,A ;засвітити на Інд. DD17, DD18 число збережене в A
mov A,#00h ;обнулить A
Con: jmp Con ;перехід на зациклення програми
END
```

### Варіанти індивідуальних завдань

Завдання до л.р. №1

Табл.1

№	Текст індивідуального завдання
1	Занести в регістр R4 двійково-десятькове число 0X, у регістр R6 двійково-десятькове число X0, суму чисел відобразити на першому й другому знакомісці статичної індикації .
2	Занести в регістр R3 двійково-десятькове число XX відобразити його на першому й четвертому знакомісці статичної індикації.
3	Занести в регістр У двійково-десятькове число, із частотою 2 Гц виводити це число на першому й другому знакомісці статичної індикації .
4	Занести в Акумулятор двійково-десятькове число XX, у регістр R5 X0, число з Акумулятора відобразити на першому й другому знакомісці статичної індикації, число з R5 відобразити на третьому знакомісці статичної індикації.
5	Занести в регістр R2 двійково-десятькове число 0X, у регістр R5 X0, суму чисел відобразити на другому й третьому знакомісці статичної індикації.
6	Занести в комірку з адресою B0h внутрішньої пам'яті ОЕОМ двійково-десятькове число 0X, у регістр R3 число X0, суму чисел відображати на другому й третьому знакомісці статичної індикації із частотою 0,5Гц.
7	Занести в регістр R0 двійково-десятькове число XX, поперемінно відображати молодшу й старшу тетраду на першому й четвертому знакомісці статичної індикації із частотою 1 Гц.
8	Занести в У двійково-десятькове число X0, у регістр R1 XX, число з У відображати на першому знакомісці статичної індикації із частотою 1 Гц, число з R1 відображати на третьому й четвертому знакомісці статичної індикації із частотою 0,5 Гц.
9	Уважати значення регістра ТСО і відобразити його на третьому й четвертому знакомісці статичної індикації.
10	Занести в регістр R4 двійково-десятькове число 0X, у регістр R3 X0, суму



№	Текст індивідуального завдання
	чисел відобразити на другому й третьому знакомісці статичної індикації з повільним (протягом 5 сек.) загасанням цього числа.
11	Занести в Акумулятор двійково-десятькове число $X0$ , у регістр $B\ 0X$ , суму чисел відобразити на першому й четвертому знакомісці статичної індикації.
12	Занести в регістр $U$ двійково-десятькове число $0X$ , у регістр $R5\ X0$ , два розряди суми (десятки й одиниці) по черзі відображати на першому й другому знакомісці статичної індикації
13	Занести в регістр $R1$ двійково-десятькове число $0X$ , віднімаючи від числа одиницю відображати на третьому знакомісці статичної індикації отримане значення до нуля із частотою 1 Гц.
14	Занести в регістр $R3$ двійково-десятькове число $XX$ , у регістр $R5\ XX$ , поперемінно відображати ці числа на першому й другому знакомісці статичної індикації ( $R3$ ) і на третьому й четвертому знакомісці статичної індикації ( $R5$ ).
15	Занести в регістр $A$ двійково-десятькове число $0X$ , у регістр $R2\ X0$ , число з $A$ відобразити на четвертому знакомісці статичної індикації, число з регістра $R2$ відображати на другому знакомісці статичної індикації із частотою в 0.5 Гц.
*	Занести в регістр $R0$ число $XX$ , вивести на лівій парі знакомісць статичної індикації. Через 2 секунди вивести число $XX$ , занесене в регістр $R1$ . Через 2 секунди на правій парі знакомісць статичної індикації вивести різницю чисел занесених у регістр $R0$ й $R1$ .
*	Занести число $0X$ у регістр $R0$ і число $XX$ у регістр $R1$ . Вивести значення $R1$ на правій парі знакомісць статичної індикації. Потім додавати значення $R1$ до значення $R0$ , доти, поки результат не досягне певного, заздалегідь заданого порога. Здійснити перехід на початок програми. (Час затримки 0,5 секунд).
*	Занести число $0X$ у регістр $R0$ і число $XX$ у регістр $R1$ . Суму вивести на правій парі знакомісць статичної індикації. Потім додавати значення $R1$ до значення $R0$ , доти, поки значення суми не досягне певного, заздалегідь заданого порога. Потім від результату віднімати число занесене в регістр $R2$ , причому $R2$ не дорівнює $R1$ . При досягненні нуля повернутися до додатка значення регістра $R1$ .

## **Лабораторна робота №2. Способи побудови схем відображення інформації**

**Тема:** Схеми відображення інформації

**Мета роботи:** Вивчення схем динамічної й статичної індикації

**Навчальне завдання:** Розробка програм для ОЕОМ сімейства MCS-51 для відображення цифрової інформації на пристроях динамічного й статичного типу, а також на одиничних індикаторах

### ***Порядок виконання лабораторної роботи:***

- 1) Вивчити принцип роботи різних методів відображення.
- 2) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття
- 3) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття з використанням підпрограм.
- 4) Увести програму індивідуального завдання на персональному комп'ютері.
- 5) Вивчити програмно налагоджувальні засоби (ПНЗ) для КР1816ВЕ31.
- 6) За допомогою ПНЗ проаналізувати виконання індивідуальної програми
- 7) Завантажити програму в стенд ОЕОМ. Переконатися в правильному виконанні індивідуального завдання, при негативному результаті здійснити зміну алгоритму або програми. Повторити завантаження програми в стенд ОЕОМ
- 8) Роздрукувати листинг правильно працюючої програми.
- 9) Відповісти на контрольні питання викладача

### ***Контрольні питання:***

1. Розрахунок часу регенерації для динамічного методу відображення.
2. Обґрунтування необхідності застосування різних методів відображення
3. Схемотехнічні рішення для побудови схем відображення інформації
4. Схеми включення одиничних індикаторів .
5. Включення рідиннокрісталічних індикаторів
6. Включення газорозрядних індикаторів

*Короткі теоретичні відомості*  
*Системи відображення інформації*

Найпростішими приладами відображення інформації в цифрових пристроях є світлодіоди й цифрові індикатори.

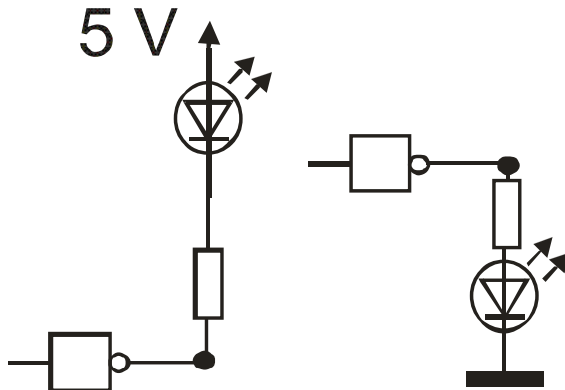


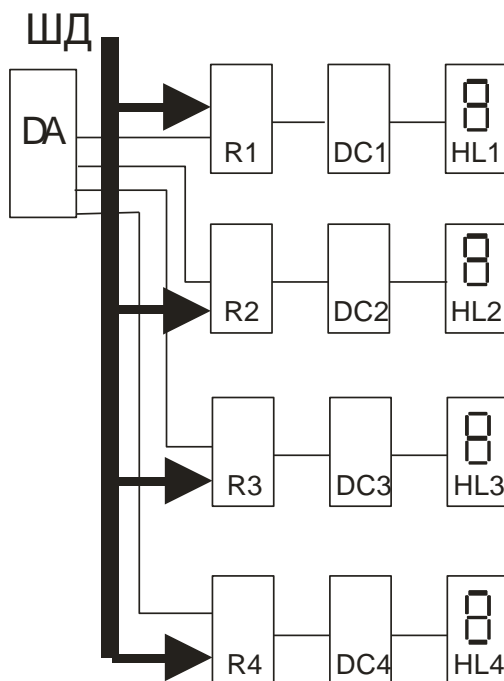
Рис.6. Включення одиничних індикаторів.

У напівпровідникових світлодіодах використовується властивість р-п переходу випромінювати світло у видимій частині спектра при протіканні через нього прямого струму ( $I_{пр}=5-20\text{ма}$ ,  $U_{пр}=2-3\text{У}$ ). Варіанти включення індикаторів на рис.6.

Для відображення цифрової інформації найбільше розповсюджені семисегментні індикатори, у яких зображення цифри становлять із семи лінійних світлодіодних сегментів розташованих у вигляді цифри 8.

На основі світлодіодів і семисегментних індикаторів будуються підсистеми відображення інформації. При побудові підсистем відображення інформації розрізняють два підходи - динамічна й статична схема побудови підсистеми індикації.

Статична індикація полягає в постійному підсвічуванні індикаторів HL1-n від одного джерела інформації рис.6.



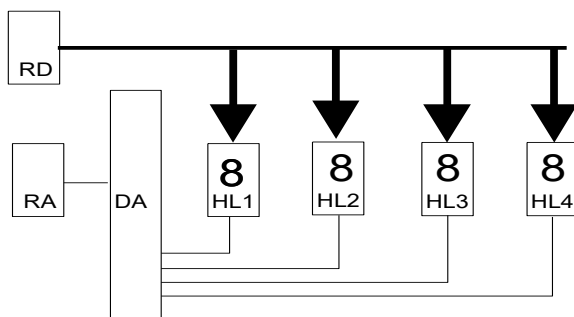
DA - дешифратор адреси, необхідний для вибірки відповідного регістра  
 R1-R4- регістри, у яких тимчасово зберігається значення коду числа для відображення (відповідний регістр вибирається DA).  
 DC1-DC4 - семисегментні дешифратори, що перетворюють двійковий код у семисегментний код.  
 HL1-HL4 - семисегментні індикатори.  
 ШД - шина даних, по ній здійснюється передача даних на індикацію.

Рис.7. Структурна схема статичної індикації.

У такій системі кожен індикатор HL1-n підключений через власний дешифратор DC1-n і регістр-засувку RG1-n до шини даних, вибірка регістрів RG1-n відбувається за допомогою селектора адреси СА. Апаратні витрати при такій організації становлять n пара регістр + дешифратор при n десяткових розрядів індикатора.

Сутність динамічної індикації полягає в почерговому циклічному підключенні кожного індикатора HL1-n до джерела інформації через загальну шину даних, рис 8.

Вибірка індикатора здійснюється дешифратором DA. У регістрі RD зберігається цифровий код, призначений для відображення. У регістрі RA зберігається адреса індикатора.



RD- реєстр даних для тимчасового зберігання відображуваного числа або символу.

RA- реєстр адреси для тимчасового зберігання двійкового коду адреси обраного індикатора.

DA-для перетворення адреси перетворення двійкового коду у позиційний код.

HL1-HL4- семисегментні індикатори.

Рис. 8. Структурна схема динамічної індикації

При такому включенні значно зменшуються апаратні витрати. Але необхідно забезпечити достатній час підсвічування одного індикатора, для того щоб не зменшувалася яскравість. Також необхідно забезпечити таку частоту перебору індикаторів, щоб не було помітно мерехтіння. Переваги такого способу помітні при кількості розрядів індикації більше 5.

У стенді статична індикація реалізована на чотирьох статичних семисегментних індикаторах HG1 (розряди HG1.0, HG1.1, HG1.2, HG1.3). Звертання до них відбувається, як до комірок пам'яті з адресами A000h(ліва пара знакомісць), B000h(права пара знакомісць).

Динамічна індикація реалізована на платі розширення за допомогою чотирерозрядного семисегментного індикатора HL2. Керування динамічною індикацією здійснюється за допомогою порту В мікросхеми системного контролера (див. схему електричну стенда, додаток 1) , сигнали вибірки відповідного індикатора надходять від лінії порту PC0, PC1 до дешифратора адреси розряду DD3.

Знакосинтезуюча індикація реалізована на платі розширення за допомогою матриці світлодіодів 5x7 HG1. Керування світлодіодною матрицею здійснюється по лініях PA0-PA4 і PC0-PC6. Наприклад для того щоб засвітити крапку з координатами

[1;1] необхідно виставити рівень логічної одиниці на лінії PA0, і рівень логічного нуля на лінії PC0.

Також у стенді є лінійка світлодіодів HL1-HL8 доступ до яких здійснюється як до осередку зовнішнього ОЗУ за адресою 0A006h. Світлодіоди запалюються записом логічних одиниць у відповідні розряди.

### *Приклад програми для статичної індикації*

**Із частотою 1 Гц відобразити на статичному індикаторі число 04:**

```
CSEG
ORG 0
Continue:
    mov A, #0
    mov DPTR, #0A004h
    movx @DPTR, A           ;скасувати гасіння знакомісць ІЗ_інд

    mov A, #04h             ;записати в Акрк. число 04
    mov DPTR, #0A000h       ;установити в DPTR адреса лівої
                             ;пари знакомісць ІЗ_інд
    movx @DPTR, A           ;засвітити число 04
    mov DPTR, #0B000h       ;установити в DPTR адреса правої
                             ;пари знакомісць ІЗ_інд
    movx @DPTR, A           ;засвітити число 04
    CALL ZAD                ;виклик підпрограми затримки

    mov A, #00001111b
    mov DPTR, #0A004h
    movx @DPTR, A           ;погасити всі знакомісця З_інд
    CALL ZAD

    jmp Continue            ;перехід на початок програми
ZAD:
    mov R1, #0FFh
    mov R2, #0FFh
    djnz R2, C4
    djnz R1, C2
    ret                     ;вихід з підпрограми
END
```

**На світлодіодах HL1-HL8 запустити " одиничку, щобіжить," зменшуючи час затримки між засвічуванням світлодіодів до певного значення, після чого засвітити всі світлодіоди. (Приклад програми написаний мовою 3).**

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned int svet,low,i,zad;

    righti=0x33;        //засвітити числа 33 на правом
    lefti=0x33;         //і левом знакомісцях ендикатора HG1
    zad=10000;

begin:
    low=1;
    LED_REG=low;        //у регістр світлодіодів записуємо
                        //значення low
    for (svet=1; svet<8; svet++) //номер активного //світлодіода HL1-HL8
    {
        delay16(zad);
        low=low<<1; //зрушити значення low на знак уліво
        LED_REG=low;
    }
    low=256;
    for (svet=1; svet<=8; svet++)
    {
        zad=zad-100;
        if (zad==100)    //зменшуємо затримку
            goto cont;   //якщо значення zad досягло 100
                        //засвітити всі світлодіоди
        else
            delay16(zad);
        low=low>>1;      //зрушити значення low вправо
                        //на один знак вправо
        LED_REG=low;
    }
    goto begin;
cont:
    low=0xFF;
    LED_REG=low;
    for (i=1; i<=100;i++)
        delay16(1000);
    zad=10000;
    goto begin;
}
```

### *Варіанти індивідуальних завдань*

Завдання до л.р. №2

Табл.2.

№.	Текст індивідуального завдання
1	Занести в регістр R1 XXH, віднімаючи від числа "1" відображати результат на динамічному індикаторі в молодшому розряді до нуля із частотою 0,5 Гц. Включати що біжить світодиод на HL1-HL8
2	Занести в У двійково-десятькове число X0, у регістр R1 XXH, число з В. відобразити на знакосинтезуючому індикаторі число з R1 відображати на динамічному індикаторі в старшому розряді із частотою 0, 5 Гц.
3	Включити в шаховому порядку світлодіоди HL1-HL8. Занести в регістр у двійково-десятькове число 0X, у регістр R5 X0, два розряди суми (десятки й одиниці) по черзі відображати на статичному індикаторі й на динамічному індикаторі із частотою 1 Гц..
4	Занести в R6 двійково-десятькове число XXH, в. R5 двійково-десятькове число XX, в R0 двійково-десятькове число XX, відображати ці числа с R5,R6 на динамічному індикаторі, с R0 на статичному індикаторі.
5	Поперемінно відображати на знакосинтезуючому індикаторі числа від 0 до 9 дублювати ці числа на динамічному індикаторі.
6	Занести в регістр А двійково-десятькове число 0X, у регістр R2 X0, число з А відобразити на статичному індикаторі, число з регістра R2 відображати на динамічному індикаторі із частотою 0.6 Гц.
7	Занести в Аккумулятор двійково-десятькове число XX, у регістр R1 XX, молодші два розряди суми чисел відобразити на динамічному індикаторі, при цьому на знакосинтезуючому індикаторі здійснити плавне загоряння числа 5.
8	Занести в регістр R6 число XX перетворити його у двійково-десятькове число відобразити його на динамічному індикаторі відобразити значення регістра R6 на світлодіодах HL1-HL8 і його інверсний стан із частотою 1Гц.
9	Занести в регістр У двійково-десятькове число XX, у регістр R3 XX, різниця чисел відобразити на динамічному індикаторі.
10	Відобразити на знакосинтезуючому індикаторі У. Занести в Аккумулятор число XX, у регістр R5 X0, число з Аккумулятора відобразити на статичному індикаторі, число з R5 відобразити на динамічному індикаторі.
11	Занести в регістр R0 двійково-десятькове число XX, поперемінно відображати молодший і старший розряди на динамічному індикаторі із частотою 0,5 Гц.
12	Занести в регістр R2 двійково-десятькове число XX, у регістр R5 XX, суму відобразити на динамічному індикаторі.
13	Занести в регістр У двійково-десятькове число, із частотою 2 Гц



№.	Текст індивідуального завдання
	виводити це число на статичному індикаторі й одночасно на динамічному індикаторі.
14	Поперемінно включати світлодіоди HL1-HL8. Занести в комірку з адресою 0010h зовнішньої пам'яті ОЕОМ двійково-десятькове число 0X, у рег. R3 XXH, суму чисел відобразити в старшому розряді на динамічному індикаторі.
15	Занести в регістр R1 двійково-десятькове число 0X, у регістр R3 XX, суму відобразити на динамічному індикаторі. Шістнадцятькове число відобразити на HL1-HL8.
*	По черзі активуючи світлодіоди HL1-HL8, на статичному індикаторі паралельно показувати кількість активних світлодіодів. Інтервал між засвечуванням 1 с.
*	Відобразити на динамічному індикаторі слово із чотирьох букв. Вивести його в такий спосіб: Перша літера, із крайнього правого положення пройшовши всі сегменти залишається горіти в крайнім лівому положенні. Наступні літери, аналогічно, пройшовши всі сегменти залишаються за попередніми. Коли слово повністю засвітилося, гасити сегменти по черзі, починаючи із крайнього лівого сегмента з інтервалом 0,5с.

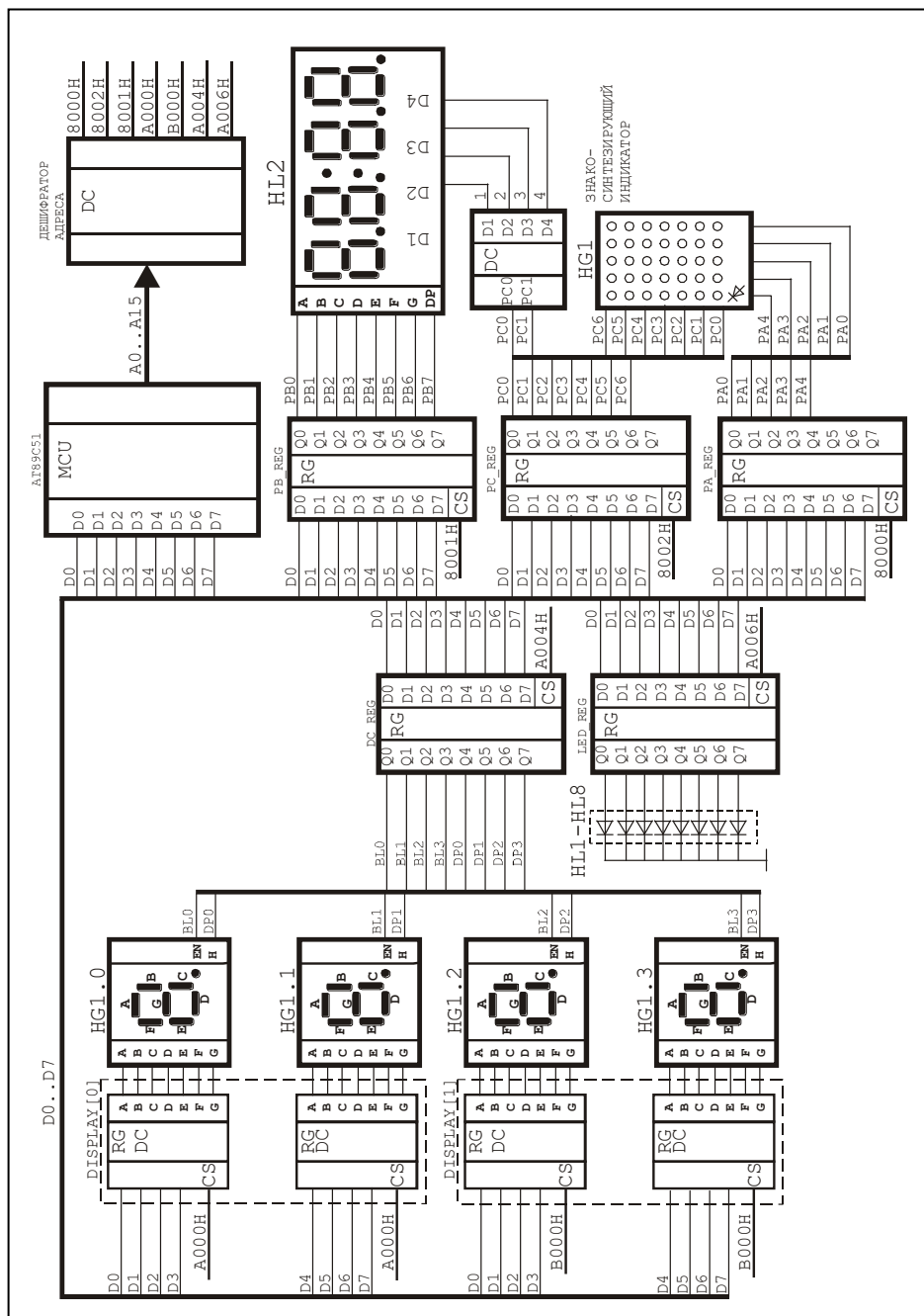


Схема для лабораторної роботи №2

### **Лабораторна робота №3. Система переривань. Опитування дискретних датчиків**

**Тема:** Система переривань. Опитування дискретних датчиків.

**Мета роботи:** Вивчення режимів роботи системи переривання ОЕОМ, програмна обробка дискретних сигналів

**Навчальне завдання:** Вивчення систем переривання режимів вводу дискретної інформації, розробка програм опитування сигналів від датчиків

#### ***Порядок виконання лабораторної роботи.***

- 1) Вивчити систему переривання ОЕОМ K1816BE31, особливості опитування дискретних датчиків з механічними контактами.
- 2) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття.
- 3) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття
- 4) Ввести програму індивідуального завдання на ПК.
- 5) За допомогою ПНЗ проаналізувати виконання індивідуальної програми
- 6) Завантажити програму в стенд ОЕОМ. Переконатися в правильному виконанні індивідуального завдання, при негативному результаті здійснити зміна алгоритму або програми. Повторити завантаження програми в стенд ОЕОМ
- 7) Роздрукувати листинг правильно працюючої програми.
- 8) Відповісти на контрольні питання викладача

#### ***Контрольні питання:***

1. Структура системи переривання ОЕОМ K1816BE31.
2. Призначення, приклади застосування системи переривання
3. Регістри керування, розподіл пам'яті в ОЕОМ K1816BE31.
4. Переривання від таймерів, послідовного порту.
5. Апаратне усунення дребезга контактів для схем з TTL і КМОП.
6. Програмне усунення дребезга контактів
7. Необхідність застосування апаратного або програмного усунення дребезга контактів

#### ***Короткі теоретичні відомості***

**Опитування дискретних сигналів.** Для вводу інформації широко застосовуються кнопкові перемикачі й контактні клавіатури. Сигнал таких перемикачів формується шляхом замикання (розмикання) електричного ланцюга. Сигнал, зформований контактною парою, супроводжується дребезгом, тривалість якого становить ~8-12мс рис. 9.

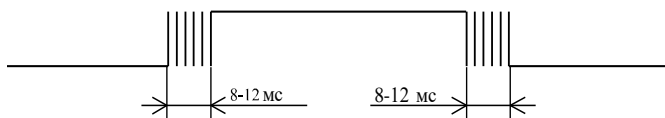


Рис. 9.Сигнал контактної пари

Для усунення дребезга в отриманому сигналі на виході контактної пари встановлюють спеціальні формувачі. Приклад такого формувача заснованого на принципі безпосереднього встановлення RS-тригера зображений на рис. 10.

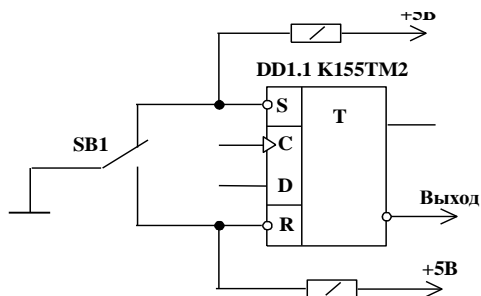


Рис.10.Схема усунення дребезга за допомогою RS-тригера

Для зменшення апаратних витрат застосовують програмне усунення дребезга. Воно полягає в повторному опитуванні контактної пари із затримкою в 12мс, при співпаданні результатів опитування кнопка була натиснута, інакше в результаті першого опитування був зафіксований дребезг.

### **Система переривань ОЕОМ 1816ВЕ31**

Регістр пріоритетів переривання (IP) призначений для встановлення рівня пріоритету переривання для кожного з 5-ти джерел переривання.

Позначення розрядів регістра IP показано в таблиці 3, а призначення наведене нижче.

Регістр пріоритетів переривань IP

7	6	5	4	3	2	1	0
X	X	X	PSP	PT1	PX1	PT0	PX0

PX0 - встановлення рівня пріоритету переривання від зовнішнього джерела /INTO.

PT0 - встановлення рівня пріоритету переривання від T/30.

PX1 - встановлення рівня пріоритету переривання від зовнішнього джерела. /INT1.

PT1 - встановлення рівня пріоритету переривання від T/3 1.

PS - встановлення рівня пріоритету переривання від послідовного порту.

X - резервний розряд.

Наявність у розряді IP "1" установлює для відповідного джерела високий рівень пріоритету, а наявність у розряді IP "0"-низький рівень пріоритету. При читанні резервних розрядів, що відповідають лінії магістралі даних не визначені.

Регістр дозволу переривання (IE) призначений для дозволу або заборони переривань від відповідних джерел. Позначення розрядів регістра IE показано в таблиці 4, а призначення зазначене нижче.

Регістр дозволу переривань IE

7	6	5	4	3	2	1	0
EA	X	X	ES	ET1	EX1	ETO	EXO

EA - керування всіма джерелами переривань одночасно. Якщо EA = 0, то переривання заборонені. Якщо EA = 1, то переривання можуть бути дозволені індивідуальними дозволами EXO, ETO, EXI, ETI, ES.

X - резервний розряд.

ES - керування перериванням від ПНЗлідового порту. ES = 1 дозвіл. ES = 0 - заборона.

ETI - керування перериванням від т/с 1. ET = 1 - дозвіл. ET = 0 - заборона.

EXI - керування перериванням від зовнішнього джерела /INT1. EXI = 1 - дозвіл. EXI = 0 - заборона.

ETO - керування перериванням від т/с 0. ETO = 1 - дозвіл. ETO = 0 - заборона.

EXO - керування перериванням від зовнішнього джерела /INTO.

EXO=1 - дозвіл. EXO = 0 - заборона.

При зчитуванні резервних розрядів відповідні лінії магістралі не визначені.

### *Структура переривань*

Механізм переривань в OMEOM дозволяє автоматично реагувати на зовнішні й на внутрішні події (переповнення таймерів/лічильників; завершення послідовного обміну).

Кожне із зовнішніх переривань /INT0, /INT1 може бути активізоване за рівнем або за фронтом сигналів P3.2, P3.3 за допомогою бітів IT0 та IT1 регістра TCON. При надходженні запиту зовнішнього переривання /INTx встановлюється прапорець IEx регістра TCON. Очищення прапорця IEx відбувається апаратно: при перериванні за фронтом IEx скидається при звертанні до відповідної підпрограми обробки переривання; при перериванні за рівнем прапорець очищається при знятті запиту зовнішнього переривання, тобто в IEx відслідковується стан виводу /INTx.

Щоб зовнішнє переривання за рівнем було розпізнано, необхідно, щоб низький рівень на виводі INTx утримувався протягом не менш 12 періодів сигналу тактової частоти. Якщо ж переривання активізується по переходу зі стану високого рівня в стан низького рівня, то циклу низького рівня повинен передувати цикл високого рівня на виводі /INTx. Якщо зовнішнє переривання активізується за рівнем, запит повинен утримуватися до початку обслуговуючої підпрограми й зніматися перед завершенням цієї підпрограми для запобігання повторного обслуговування.

Переривання від таймерів/лічильників виконуються по прапорах TF0 й TF1 регістра TCON, які встановлюються при переповненні відповідних регістрів таймерів/лічильників (за винятком режиму 3). Очищення прапорів TF0 й TF1 відбуваються при переході до підпрограми обслуговування переривання.

Переривання від послідовного порту виконується по прапоріцю закінчення прийому RI або по прапорцю закінчення передачі TI, які встановлюються в регістрі SCON.

На відміну від всіх інших прапорів, RI й TI скидаються тільки програмним шляхом звичайно в межах підпрограми обробки переривання, де визначається, якому із прапорців RI або TI відповідає переривання.

У випадку одночасного надходження запитів переривання з однаковим рівнем пріоритету, рівним 0 або 1, обробка їх відбувається в порядку внутрішнього опитування прапорців:  
 IEO -> TFO -> TE1 -> TF1 -> (TI + RI)

Встановлення прапорців переривання відбувається наприкінці машинного циклу, а їхнє опитування в наступному циклі. І тільки після виконання останнього циклу поточної команди відбувається апаратний виклик відповідної підпрограми обслуговування, еквівалентний команді LCALL.

У загальному випадку, звертання до підпрограми обслуговування затримується при виконанні хоча б однієї з наступних умов:

- відбувається обробка переривання з таким же або вищим пріоритетом;
- поточний машинний цикл (цикл опитування прапора) не є останнім циклом виконуваної команди;
- виконувана команда поточної програми RETI або будь-яка команда звертання до регістрів IE, IP.

В останній умові після закінчення однієї з вищевказаних команд обов'язково виконається ще одна команда поточної програми перед викликом підпрограми обслуговування переривання.

Прапор переривання, установлений під час дії блокування переривання по одному із трьох зазначених умов і скинутий до їхнього зняття, не викличе обслуговування відповідного запиту переривання.

Підпрограма обслуговування переривання триває до виконання команди RETI по якій відновлюється стан логіки переривання й стан програмного лічильника PC із двох верхніх комірок стека. При використанні команди RET відновлюється тільки стан програмного лічильника, а стан логіки переривання залишається незмінним.

Початкові адреси векторів переривань

<i>Джерело переривання</i>	<i>Адреса</i>
Зовнішнє переривання 0	0003H
Переповнення таймера 0	000BH
Зовнішнє переривання 1	0013H
Переповнення таймера 1	001BH
Послідовний порт	0023H

У складі учбово-налагоджувального стенда є дві окремі кнопки S10-S11, які можуть опитуватися, як програмно, так і за допомогою використання функцій переривань INT0-INT1 відповідно.

У складі стенда є матрична 3x4 клавіатура SW3-SW14. Клавіатура підключена до шини даних OEOM за допомогою мікросхеми буфера DD1 74245(АП6).

Опитування всієї клавіатури відбувається за три рази (за один раз зчитується стан тільки одного стовпця клавіатури). Щоб зробити опитування стовпця клавіатури (SW3,SW6,SW9,SW12; SW4,SW7,SW10,SW13; або SW5,SW8,SW11,SW14) необхідно виставити на відповідній лінії адреси (A0,A1,A2 для першого, другого й третього стовпця відповідно) рівень логічного нуля, а на інших лініях рівень логічної одиниці й прочитати стан буфера клавіатури, підключеного до шини даних OEOM як доступна для читання комірка пам'яті з адресою 9000h. Якщо кнопка натиснута то відповідний біт у ліченому байті буде дорівнює нулю, якщо ж не натиснута те одиниці.

Стовпець (кнопки)	Адреса
1 (SW3,SW6,SW9,SW12)	9006h
2 (SW4,SW7,SW10,SW13)	9005h
3 (SW5,SW8,SW11,SW14)	9003h



## **Приклад програмного опитування дискретного сигналу**

```

CSEG
ORG 0
    jnb p3.2,$                ;опитування натискання кнопки SW15, якщо
                                ;кнопка натиснута, програма виконується
                                ;далі

Continue:
    mov A,#0
    mov DPTR,#0A004h
    movx @DPTR,A              ;скасувати гасіння знакомісць I3_інд

    mov A,#04h                 ;записати в Акк. число 04
    mov DPTR,#0A000h           ;установити в DPTR адреса лівої пари
                                ;знакомісць I3_інд
    movx @DPTR,A               ;засвітити число 04
    mov DPTR,#0B000h           ;установити в DPTR адреса правої пари
                                ;знакомісць I3_інд
    movx @DPTR,A               ;засвітити число 04
    CALL ZAD                    ;виклик підпрограми затримки
    mov A,#00001111b
    mov DPTR,#0A004h
    movx @DPTR,A              ;погасити всі знакомісця 3_інд
    CALL ZAD

    jmp Continue                ;перехід на початок програми
ZAD:                                ;підпрограма затримки
    mov R1,#0FFh
C2:    mov R2,#0FFh
C4:    djnz R2, C4
        djnz R1, C2
        ret                    ;вихід з підпрограми

```

END

**Відобразити на C\_інд. Числа 33.33 й 32.32 при натисканні кнопок SW16, SW15 відповідно (програма написана мовою C).**

```

#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    begin:
        DC_REG=0x0F;                //гасити індикатор HG1
        if (!P3_2)
        {
            new_dotsi=0x20;
            lefti=0x32;
            righti=0x32;              //якщо кнопка
                                      //SW15 натиснута вивести 32.32
        }
        if (!P3_3)
        {
            new_dotsi=0x20;
            lefti=0x33;
            righti=0x33;              //якщо кнопка
                                      //SW16 натиснута вивести 33.33
        }

        goto begin;
}

```

## Варіанти індивідуальних завдань

Завдання до л.р. №3

Табл.3.

№	Текст індивідуального завдання
1	Підрахувати й відобразити на статичному індикаторі кількість натискань кнопки SW15
2	Реалізувати опитування клавіатури. Номер клавіші відображати шляхом засвічування відповідної крапки на знаковисинтезуючому індикаторі.
3	Реалізувати опитування клавіатури. Номер клавіші ПНЗлідовно відображати в кожному розряді динамічному індикаторі.
4	По натисканні SW15 запускати вогонь, що біжить, на світлодіодах HL1-HL8, при відпусканні плавне загоряння числа 3 на знаковисинтезуючому індикаторі.
5	По натисканні SW16 включити секундомір з відображенням на статичному індикаторі значення секунд при відпусканні запускати “тінь, що біжить, ” на світлодіодах HL1-HL8
6	Реалізувати опитування клавіатури. Номер клавіші відобразити позиційним кодом на світлодіодах HL1-HL8, з відображенням значення кнопки на динамічному індикаторі.
7	Реалізувати програму введення чотиризначного числа із клавіатури, використовуючи статичному індикаторі й дублюючи значення натиснутої кнопки на знаковисинтезуючому індикаторі.
8	Реалізувати опитування клавіатури після 2-х натискань SW16. Номер клавіші відображати на динамічному індикаторі.
9	По натисканні SW15 запускати будь-яке значення, що біжить, на знаковисинтезуючому індикаторі, а по натисканні SW16 запалити всі крапки в шаховому порядку.
10	Відображати значення секунд на статичному індикаторі. По перериванню INT0 зупинити секундомір і засвітити світлодіоди HLn (n-непарне)
11	Відображати число 7543 на динамічному індикаторі. По перериванню INT1 засвітити світлодіоди HLn (n-парне)
12	На статичному індикаторі відобразити число 5555. По перериванню INT0 відображати “шахматку” на знаковисинтезуючому індикаторі, По перериванню INT1 відобразити на статичному індикаторі число 3333
13	Після натискання SW15 реалізувати програму введення тризначного числа із клавіатури з відображенням на статичному індикаторі.
14	По натисканні SW16 запускати “тінь, що біжить, ” знаковисинтезуючому індикаторі, а при повторному натисканні SW16 згасити всі крапки.
15	Реалізувати опитування клавіатури. Номер клавіші індицировать двійковим кодом на світлодіодах HL1-HL8

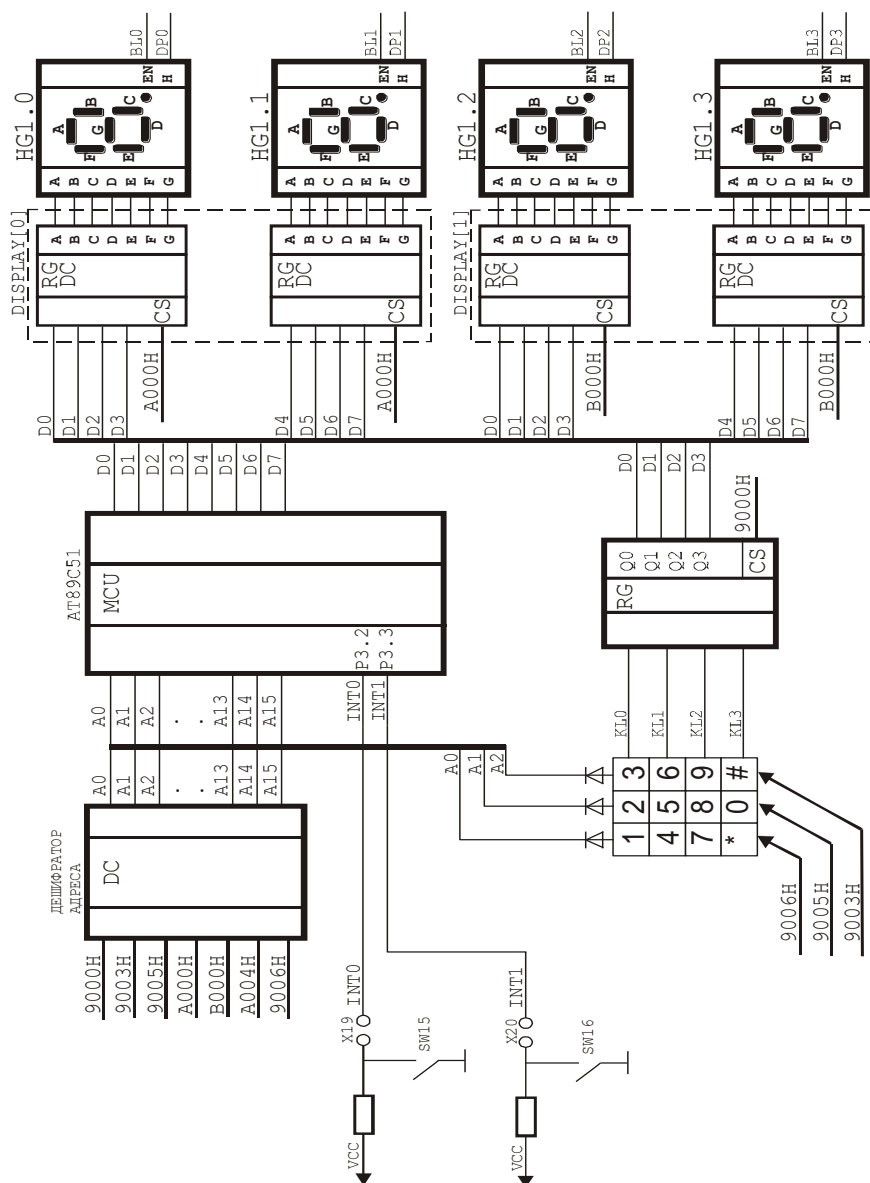


Схема до лабораторної роботи №3.

## **Лабораторна робота №4. Цифро-аналогове перетворення**

**Тема:** Цифро-аналогове перетворення

**Мета роботи:** Вивчення методів цифро-аналогового перетворення

**Навчальне завдання:** Розробка програм для формування різних аналогових сигналів.

### ***Порядок виконання лабораторної роботи***

- 1) Вивчити структурну схему модуля ЦАП на платі розширення
- 2) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття
- 3) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття
- 4) Ввести програму індивідуального завдання на персональному комп'ютері.
- 5) За допомогою ПНЗ проаналізувати виконання індивідуальної програми
- 6) Завантажити програму в стенд ОЕОМ. Переконатися в правильному виконанні індивідуального завдання (формування заданої форми сигналу) використовуючи осцилограф, при негативному результаті здійснити зміна алгоритму або програми. Повторити завантаження програми в стенд ОЕОМ
- 7) Роздрукувати листинг правильно працюючої програми.
- 8) Відповісти на контрольні питання викладача

### ***Контрольні питання:***

1. Методи й типи ЦАП.
2. Статичні параметри ЦАП
3. Поняття дискретності, квантування, що дозволяє здатність ЦАП
4. Характеристика перетворення, нелінійність ЦАП.
5. Напруга зсуву нуля, допустиме напруження на виході ЦАП.
6. Динамічні параметри ЦАП
7. Фактори, що впливають на похибку ЦАП
8. Апаратні реалізації ЦАП
9. Приклади практичного застосування ЦАП

### ***Короткі теоретичні відомості***

Цифро-аналогові перетворювачі (ЦАП) виробляють напруга або струм, функціонально пов'язані з керуючим кодом. Застосовуються ЦАП для формування вихідних аналогових сигналів цифрових вимірювальних й обчислювальних пристроїв. Для перетворення двійкового коду в аналоговий сигнал звичайно формуються струми, пропорційні вагам розрядів коду, і потім

підсумуються ті зі струмів, які відповідають ненульовим розрядам вхідного коду.

Застосовуються в основному два методи ЦАП: підсумовування одиничних еталонних величин і підсумовування еталонних величин, ваги яких розрізняються. У першому при формуванні вихідної аналогової величини використовується тільки одна еталонна величина вагою в один квант. У другому методі застосовуються еталонні величини з вагами, що залежать від номера розряду, і в підсумовуванні беруть участь тільки ті еталонні величини, для яких у відповідному розряді вхідного коду встановлена одиниця.

У випадку використання на вході двійкового позиційного коду значення всіх розрядів надходить одночасно, і робота таких ЦАП описується вираженням

$$X = \frac{P}{2^b - 1} (a_0 \cdot 2^0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + \dots + a_{b-1} \cdot 2^{b-1})$$

де  $X$  - аналогова величина.

$a_i$  – коефіцієнти відповідних двійкових розрядів, які приймають дискретні значення одиниця або нуль.

$P$  - опорний сигнал

$b$  - число розрядів

У перетворювачах з опорної напруги формуються еталонні величини, що відповідають значенням розрядів вхідного коду, які підсумуються й утворюють дискретні значення вихідної аналогової величини.

Класифікація ЦАП може бути проведена за ознаками.

Способу формування вихідної напруги (з підсумовування напруг, розподілом напруг, підсумовуванню струмів).

Роду вихідного сигналу (зі струмовим виходом, виходом по напрузі).

Полярності вихідної напруги (постійному, змінному) і т.д. Основні структури використовувані в ЦАП інтегрального виконання -це:

Структури з підсумовуванням струмів.

ЦАП - зі зваженими резисторами в ланцюгах еміттерів.

ЦАП - зі зваженими резисторами в ланцюгах навантаження.

ЦАП зі сходовою матрицею  $R=2R$  у ланцюгах емітерів транзисторів джерел струмів.

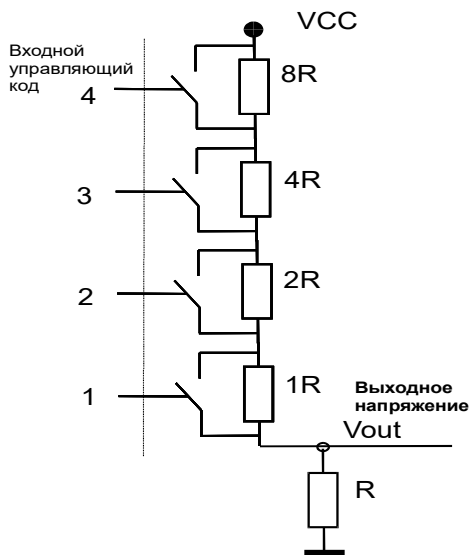
ЦАП з вихідною сходовою матрицею  $R=2R$

Параметри ЦАП.

Число розрядів  $n$  керуючого коду, номінальний вихідний струм, час установлення вихідного сигналу після зміни вхідного керуючого коду, погрішність повної шкали, погрішність лінійності, диференціальна нелінійність.

Погрішності ЦАП можуть бути виражені у відсотках або інших відносних одиницях, а також у частках кванта

Спрощена схема ЦАП для розуміння роботи перетворення вхідного коду в струм (напруга) певної величини.



Якщо співвідношення резисторів ЦАП як  $8R$ ,  $4R$ ,  $2R$ ,  $1R$ , то при включенні всіх комутаторів, відповідно до закону Ома напруга в крапці OUT буде рівнятися напрузі  $VCC$ . Якщо включити комутатор резистора  $R8$ , напруга в крапці OUT буде дорівнює  $VCC/2$ . Аналогічно можна знати значення напруги, при різних комбінаціях вхідного коду.

У складі стенда є мікросхема ЦАП AD7801. Вона являє собою восьмирозрядний ЦАП (з паралельним інтерфейсом) з розмахом вихідної напруги рівним напрузі живлення. Для спостереження вихідного сигналу з мікросхеми ЦАП необхідно підключити осцилограф до BNC роз'єму й замкнути перемичку J4. Доступ до ЦАП здійснюється як до комерки зовнішнього ОЗП за адресою 0F000h.

### **Приклад виконання програми цифро-аналогового перетворення**

```
ORG 0
Begin:
    mov A, #0                ;записати в Акк. 0
    mov DPTR, #0F000h        ;установити в DPTR адрес Цап
    movx @DPTR, A            ;установити на ЦАП код з Акк.

    call ZAD                  ;виклик підпрограми затримки
continue:
    inc A                    ;збільшити значення Акк. на одиницю
    movx @DPTR, A            ;установити на ЦАП код з Акк.
    call ZAD
    cjne A, #255d, continue  ;якщо код не досяг максимального значення
                                ;перейти на мітку continue
                                ;інакше перехід на мітку Begin
    jmp Begin
ZAD:
    mov R4, #005h
C2:   mov R2, #0FFh
C3:   djnz R2, C3
      djnz R4, C2
      ret                    ;виходи з підпрограми
END
```

### **Сформувати пилкоподібні сигнали. \_ (програма написана мовою C).**

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned char y;
    DC_REG=1;
    lefti=0x0D;
    righti=0xAC;
    y=0;
    DAC_REG=y;
    delay16(10);
begin:
    for (y=1; y<255; y++) //збільшувати код Цап на
                        //одиниці до максимального
    {
        DAC_REG=y;
        delay16(10);
    }
    for (y=255; y>1; y--) //зменшувати код Цапа на
                        //одиниці до мінімального
    {
        DAC_REG=y;
        delay16(10);
    }
    goto begin;
}
```



### **Варіанти індивідуальних завдань**

Завдання до л.р. №4

Табл.4.

<i>№</i>	<i>Текст індивідуального завдання</i>
1	Сформувати пілкоподібну напругу із частотою повторення 50 Гц. Відображати на статичному індикаторі число сгенерованих імпульсів.
2	По натисканні SW3 сформувати трикутні імпульси, передній фронт 20мсек задній 10 мсек, кожен 10 імпульс відображати на динамічному індикаторі.
3	По натисканні SW4 сформувати трапецієподібні імпульси, передній фронт 13мсек задній 15 мсек щосекунди запалювати світлодіод HL1.
4	Сформувати синусоїду із частотою повторення 120 Гц.
5	Сформувати пілкоподібну напругу із частотою повторення 200Гц і тривалістю переднього фронту 2 мсек.
6	По натисканні SW5 сформувати синусоїду із частотою повторення 100Гц.
7	Сформувати прямокутні імпульси, із тривалістю 25мсек і шпаруватістю 4
8	По натисканні SW16 сформувати трикутні імпульси, передній фронт 25мсек задній 5 мсек.
9	Сформувати синусоїду із частотою повторення 300 Гц. По натисканні SW15 змінити частоту на 100 Гц.
10	Сформувати два прямокутних імпульси, однією максимальною амплітудою тривалістю й другий 2/3 амплітуди максимальної з періодом повторення 40 Гц.
11	Сформувати прямокутні імпульси, із тривалістю 25мсек по натисканні SW6 на клавіатурі сформувати трикутні імпульси.
12	Сформувати синусоїду із частотою повторення 70 Гц, по натисканню SW7 на клавіатурі прямокутні імпульси, із тривалістю 25мсек і шпаруватістю 2.
13	По натисканні SW15 сформувати трикутні імпульси, передній фронт 15мсек задній 40 мсек.
14	По натисканні SW9 на клавіатурі сформувати трапецієподібні імпульси, передній фронт 20сек задній 20 мсек
15	Сформувати три прямокутних імпульси, один 1/3 максимальної амплітуди, 2-ий 2/3 максимальної амплітуди, 3-ій максимальної амплітуди з періодом повторення 100 Гц.



## **Лабораторна робота №5. Аналого-цифрове перетворення**

**Тема:** Аналого-цифрове перетворення

**Мета роботи:** Навчитися вимірювати аналогову величину.

**Навчальне завдання:** Розробка програм виміру аналогових величин для різних методів виміру й типів АЦП

### ***Порядок виконання лабораторної роботи***

- 1) Вивчити структурну схему модуля АЦП на платі розширення
- 2) Розробити алгоритм і програму для виконання індивідуального завдання до початку лабораторного заняття
- 3) Ввести програму індивідуального завдання на персональному комп'ютері.
- 4) За допомогою ПНЗ проаналізувати виконання індивідуальної програми
- 5) Завантажити програму в стенд ОЕОМ. Переконавшись в правильному виконанні індивідуального завдання, змінити значення подаваної напруги на вхід АЦП, повторити перетворення, при негативному результаті здійснити зміну алгоритму або програми.
- 6) Повторити завантаження програми в стенд ОЕОМ
- 7) Роздрукувати листинг правильно працюючої програми.
- 8) Відповісти на контрольні питання викладача

### ***Контрольні питання***

1. Методи й типи АЦП.
2. Статичні параметри АЦП
3. Поняття дискретності, квантування, що дозволяє здатність
4. Характеристика перетворення, диференціальна нелінійність АЦП, відхилення коефіцієнта перетворення.
5. Напруга зсуву нуля.
6. Динамічні параметри АЦП
7. Час перетворення, час затримки запуску, час циклу перетворення, максимальна частота перетворення.
8. Поняття апертурний час.
9. Фактори, що впливають на похибку АЦП
10. Апаратні реалізації АЦП.
11. Приклади практичних застосувань АЦП
12. Побудова схем АЦП за допомогою мікросхем ЦАП

### ***Короткі теоретичні відомості.***

Аналогово-цифрові перетворювачі (АЦП) застосовуються у вимірювальних системах і вимірювально-обчислювальних комплексах для узгодження аналогових джерел вимірюваних сигналів із цифровими пристроями обробки й представлення результатів виміру.

Існують різні методи побудови АЦП. Вони відрізняються по складності реалізації, перешкодостійкості, швидкодії.

У системах де основним критерієм є швидкодія застосовують АЦП паралельного перетворення. Але АЦП цього типу досить складні в реалізації. Для  $n$ -розрядного АЦП необхідно  $2^n - 1$  компараторів і паралельний дільник напруги, що виробляє  $2^n - 1$  рівнів квантування.

Для реалізації систем з високою перешкодостійкістю застосовують інтегруючі АЦП. Такий АЦП складається із двох перетворювачів. Вимірювана напруга перетворює в тривалість імпульсу, а потім тривалість імпульсу перетворює в цифровий код.

Одним з найпоширеніших є АЦП, побудований на цифро-аналоговому (ЦАП) перетворювачі. Схема цього АЦП наведена на рис. 12.

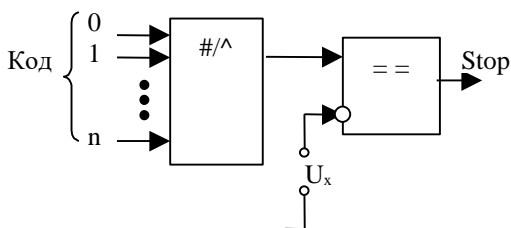


Рис 12.Схема АЦП побудованого на ЦАП

Код формується лічильником, при організації жорсткої логіки, або програмно, якщо АЦП працює в складі обчислювального комплексу. Вхідний код перетворює в аналоговий сигнал за допомогою ЦАП. Напруга з виходів ЦАП надходить на один із входів компаратора. На інший вхід подається вимірювана напруга  $U_x$ . У момент, коли напруга ЦАП буде дорівнює вимірюваному, компаратор формує сигнал 'Stop' який свідчить про закінчення циклу виміру.

При формуванні коду використовуються різні алгоритми. Найпростішим алгоритмом є порозрядне врівноважування. При такому підході код міняється від мінімального шляхом збільшення одиниці молодшого розряду доти, поки напруга ЦАП не

зрівняється з вимірюваною напругою. Недоліком порозрядного врівноважування є маленька швидкодія.

Для скорочення часу перетворення застосовується метод половинних наближень. Врівноважування починається зі старшого розряду. У цьому розряді встановлюється одиниця й читається стан компаратора. Якщо напруга ЦАП більше вимірюваної то розряд скидається, а якщо менше, те розряд зберігає свій стан. Далі в такий же спосіб обробляється наступний розряд. Перетворення закінчується тоді, коли будуть оброблені всі розряди.

У системах спостереження, за якими або параметрами часто необхідно безупинно зчитувати стан датчика. Це забезпечується малим часом перетворення за рахунок застосування що стежить АЦП. Суть даного алгоритму полягає в тім, що спочатку код формується методом половинних наближень. А після порівнянь із вимірюваною напругою АЦП відслідковує зміна напруги. Якщо напруга росте то код порозрядно збільшується доти, поки напруга ЦАП не зрівняється з вимірюваним, і навпаки.

У стенді АЦП побудований на мікросхемах AD7801(восьмирозрядний ЦАП) і LM358 (ОП як компаратор) (див. схему стенда). Стан компаратора можна вважати з виводу порту P1.7 однокристальної ЕОМ, про закінчення циклу перетворення також свідчить світіння світлодіода підключеного до виходу компаратора. Доступ до ЦАП здійснюється як до комірки зовнішнього ОЗП за адресою 0F000h.

Вимірювана напруга  $U_x$  формується змінним резистором (якщо встановлено перемичку J5), або джерело сигналу підключається до клеммнику.

У розширеній комплектації стенд ПНЗтавляється з інтегральним десятибітним АЦП із паралельним інтерфейсом AD7813. Доступ до АЦП AD7813 здійснюється як до осередку зовнішнього ОЗУ за адресою 0E000h.

### **Приклад програми для аналого-цифрового перетворення.**

```
ORG 0
Begin:
    mov A, #0 ;записати в Акк. 0
nextTest:
    mov DPTR, #0F000h ;установити в DPTR адреса ЦАП
    movx @DPTR, A ;установити на ЦАП код з Акк.
    mov r3, #50;
z2:    djnz r3, z2
jnb    P1.7, ShowResult ;перевірка спрацьовування компаратора
;збільшити значення Акк. на одиницю
    inc A
    cjne A, #0ffh, nextTest ;якщо код не досяг максимального значення
ShowResult:
    mov DPTR, #0B000h ;вивід значення коду на індикатор GH1
    movx @DPTR, A
    call ZAD
    jmp Begin ;інакше перехід на мітку Begin
ZAD:   ;підпрограма затримки
    mov R4, #0ffh
C2:    mov R2, #0FFh
C3:    djnz R2, C3
    djnz R4, C2
    ret ;виходи з підпрограми
END
```

### **(програма мовою Cі) .**

```
#include <8051.h>
#include "..\ev8031.lib\ev8031.c"
#include "..\ev8031.lib\bitdef.h"

int main()
{
    unsigned char y,x,z;
    DC_REG=1;
    lefti=0x0A;
    righti=0xDC;
start:
    if (!P3_2) //запустити програму при
                //натисканні кнопки SW15
    {
begin:
        y=0;
        DAC_REG=y;
        for (y=0; y<255; y++) /*подавати код на ЦАП
збільшуючи його на 1 доти поки напруга із Цап не
зрівняється з невідомою напругою*/
        {
            DAC_REG=y;
            delay16(10);
            if (!P1_7)
            {
                x=y/100;
                lefti=x;
                z=x*100;
                x=y-z;
                righti=x; //вивести код на індикатор HG1
                delay16(300);
                goto begin;
            }
        }
        goto begin;
    }
    goto start;
}
```

### *Варіанти індивідуальних завдань*

Завдання до л.р. № 5

Табл.5.

<i>№</i>	<i>Алгоритм для реалізації</i>
1	Порозрядне врівноважування, значення відображати на статичному індикаторі.
2	Метод половинних наближень, значення відображати на динамічному індикаторі.
3	Слідкуючий з первісним порозрядним врівноваженням, відображати на статичному індикаторі.
4	Слідкуючий з первісним половинним врівноваженням, відображати на статичному індикаторі.
5	По натисканню кнопки SW15 запустити АЦП половинних наближень, значення відображати на динамічному індикаторі.
6	По натисканню кнопки SW3 на клавіатурі запустити АЦП порозрядного зрівноважування, значення відображати на статичному індикаторі.
7	По натисканню кнопки SW4 на клавіатурі запустити АЦП типу, що стежить, з первісним порозрядним зрівноважуванням, значення відображати на динамічному індикаторі.
8	По натисканню кнопки SW5 на клавіатурі запустити АЦП типу, що стежить, з первісним половинним наближенням, значення відображати на статичному індикаторі.
9	По натисканню кнопки SW6 на клавіатурі запустити АЦП половинних наближень результат відобразити на динамічному індикаторі по натисканню кнопки SW3
10	По перериванню INT0 запустити АЦП порозрядного зрівноважування результат відобразити на статичному індикаторі по натисканню кнопки SW3
11	По натисканню кнопки SW7 запустити АЦП типу, що стежить, з первісним порозрядним зрівноважуванням результат відобразити по натисканню кнопки SW8
12	По натисканню кнопки SW8 запустити АЦП типу, що стежить, з первісним половинним наближенням результат відобразити на динамічному індикаторі по натисканню кнопки SW9

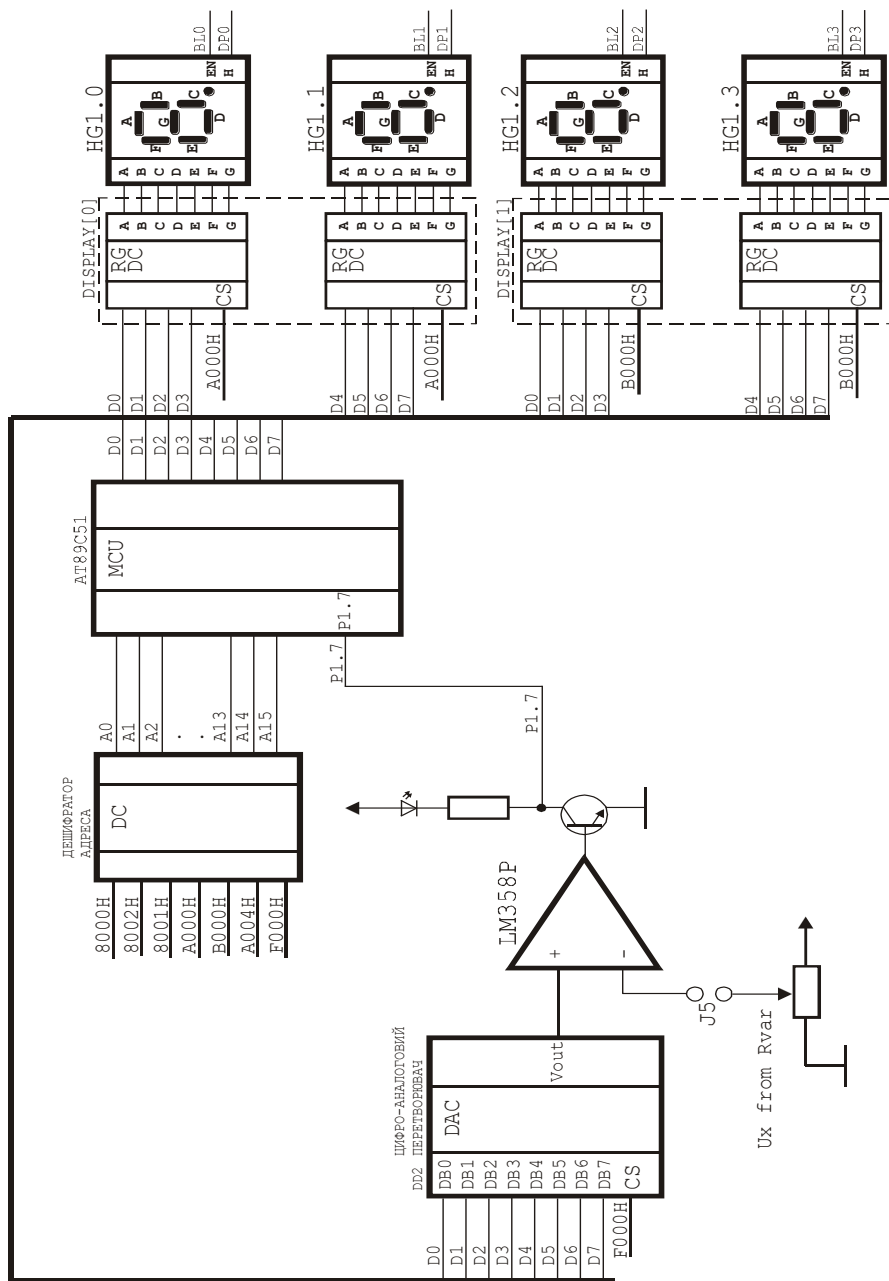


Схема до лабораторної роботи №5



## **Лабораторна робота №6. Обробка частотних і тимчасових сигналів**

**Тема:** Обробка частотних і тимчасових сигналів

**Мета роботи:** Вивчення методів частотного перетворення

**Навчальне завдання:** Навчитися здійснювати вимір частоти, періоду, тривалість дискретних сигналів за допомогою програмувальних лічильників/ таймерів ОЕОМ 1816ВЕ31, а також з використанням зовнішнього еталонного генератора.

### ***Порядок виконання лабораторної роботи.***

- 1) Вивчити методи частотного перетворення (апаратні й за допомогою програмувальних лічильників таймерів)
- 2) Вивчити методи виміру часових інтервалів.
- 3) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття.
- 4) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття
- 5) Ввести програму індивідуального завдання на персональному комп'ютері.
- 6) За допомогою ПНЗ проаналізувати виконання індивідуальної програми
- 7) Завантажити програму в стенд ОЕОМ. Переконалися в правильному виконанні індивідуального завдання, змінити значення вимірюваної частоти, повторити перетворення, при негативному результаті здійснити зміну алгоритму або програми.
- 8) Роздрукувати листинг правильно працюючої програми.
- 9) Відповісти на контрольні питання викладача

### ***Контрольні питання.***

1. Методи й типи частотного й часового перетворення.
2. Параметри частотних перетворень.
3. Фактори, що впливають на похибку частотного перетворення.
4. Поняття роздільна здатність частотного перетворення.
5. Вимір періоду.
6. Характеристика перетворення, нелінійність частотного перетворення.
7. Апаратні реалізації частотного й часового перетворення.
8. Приклади практичного застосувань частотного й часового перетворення

### ***Короткі теоретичні відомості***

У системах автоматичного керування часто доводиться вимірювати такі величини як: частота  $f$ , період  $T$ , тривалість  $\tau$ , зсув фаз  $\varphi$ . Для цього застосовується перетворення частота-код. Залежно від того, який саме параметр потрібно виміряти застосовують різний підхід.

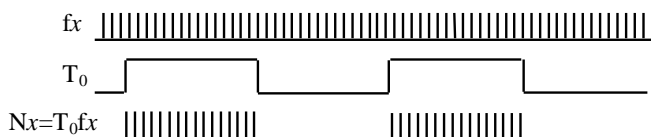


Рис.13.Визначення частоти імпульсів.

При визначенні частоти  $f_x$  відбувається рахунок імпульсів вхідного сигналу, на протязі фіксованого відрізка часу рис13. Такий метод прийнятний для виміру високих частот більше 100Гц. Верхня межа обмежується швидкодією елементів схеми й розрядністю лічильників.

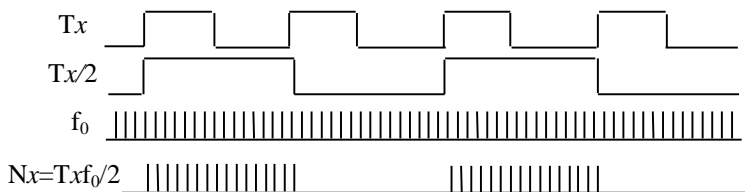


Рис.14.Визначення періоду проходження методом підрахунку кількості імпульсів відомої частоти.

Визначення частот нижче 100Гц замінюється визначенням періоду  $T_x$ . При цьому відбувається підрахунок імпульсів фіксованої частоти  $f_0$  за інтервал рівному або кратний вимірюваному періоду мал. 14.

У такий же спосіб вимірюється тривалість імпульсу  $\tau_x$ , відмінність складається лише в тім, що при визначенні тривалості імпульсу непотрібно ділити частоту на два.

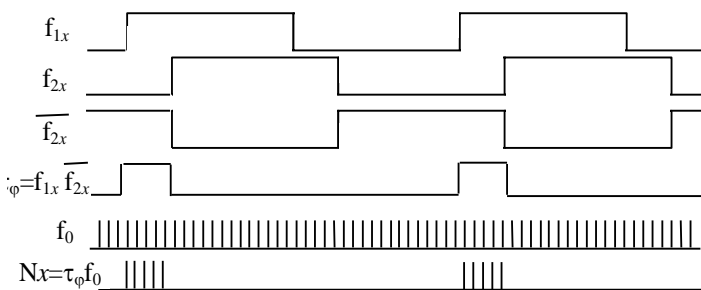


Рис.15.Визначення зсуву фаз

При визначенні зсуву фаз часовий інтервал  $\tau_\phi$  формують шляхом кон'юнкції сигналів  $f_{1x}$  й  $f_{2x}$  рис. 15, отриманий інтервал часу вимірюють методом описаним вище.

Схеми, що реалізують такий алгоритм, наведені на рис. 16.

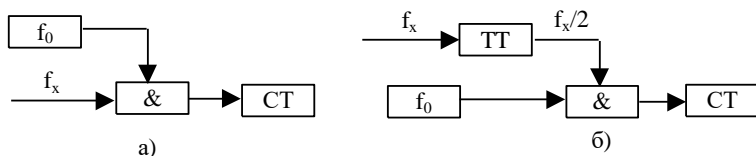


Рис.16.Схеми для визначення частоти а) низкою, б) високої

У лабораторному стенді при наявності плати розширення до входу ОЕОМ Т0 підключений постійний генератор частоти, а до входу Т1 підключений генератор зі змінюваною частотою. Сигнали цих генераторів можна спостерігати на осцилографі, підключеному до BNC роз'єм. Для спостереження сигналів Т0 замкнути перемичку J1, а для Т1 перемичку J3 Таймери/лічильники (Т/З) ОЕОМ призначені для підрахунку зовнішніх подій, для отримання програмно керованих часових затримок, виконання часозадаючих функцій ОЕОМ.

До складу блоку Т/І з входять:

- 1) два 16-ти розрядних регістри Т/З 0, Т/З 1;
- 2) 8-ми розрядний регістр режимів Т/З (ТМ0D);
- 3) 8-ми розрядний регістр керування (ТСО);
- 4) схема інкремента; \_\_\_\_\_
- 5) схема фіксації INT0, INT1, Т0, Т1;

6) схема керування прапорцями;

7) логіка керування Т/С.

Два 16-розрядних регістри Т/Л 0 і Т/Л 1 виконують функцію зберігання вмісту рахунку. Кожний з них складається з пари 8-ми розрядних регістрів, відповідно ТН0, ТЛ0 і ТН1, ТЛ1. Причому регістри ТН0,ТН1 - старші, регістри ТЛ0,ТЛ1 - молодші 8 розрядів. Кожний з 8-ми розрядних регістрів може бути використаний, як РОН, якщо Т/З не використовуються (біт TR0 для Т/З 0 і біт TR1 для Т/З 1 у регістрі керування TCON дорівнює "0").

Регістр режимів Т/З (TMOD) призначений для прийому й зберігання коду, що визначає:

- один з 4-х можливих режимів роботи кожного Т/З;
- роботу як таймери або лічильників;
- керування Т/Л від зовнішнього виводу.

Регістр режимів TMOD

M0-M1(0-1,4-5) - один з 4-х режимів лічильників

Т/Л (2,6) - '0' таймер '1' лічильник

GATE - дозвіл керування від зовнішнього виводу '0' заборонене '1' дозволено(INT0,INT1)

Регістр керування (TCON) призначений для прийому й зберігання коду керуючого слова. Регістр TCON не можна використати в якості РОН. Схема інкремента призначена для збільшення на 1 у кожному машинному циклі вмісту регістрів Т/З 0, Т/З 1, для яких:

- установлений режим таймера й рахунок дозволений;
- установлений режим лічильника, рахунок дозволений і на відповідному вході схеми Т0 (для Т/Л 0) або Т1 (для Т/Л 1) зафіксований рахунковий імпульс у попередньому машинному циклі.

Регістр керування TCON

TR - біт включення Т/С '0' виключений

TF - прапор переповнення Т/С

IT - визначає вид переривання по вході INT '0' по L-рівню, '1' по фронту

IE - прапорець запиту на переривання

## Приклад програми визначення частоти

```
$nolist
#include (c:\asm51\compile\mod51)
$list
;вимір частоти в Гц
;розрахована на зовнішню частоту T0 6817Гц
;підрахунок ведеться за 1з, результат у вигляді XXXX Гц
mode equ 55h ;набудовує T1 counter, T0 counter
str equ 50h ;запис в TCON дозволяє рахунок
ORG 0000h
BEG: MOV TL1,#00h ;обнуління
MOV TH1,#00h ;T/31
MOV TL0,#0f0h ;предвстановлення T/30
MOV TH0,#0E4h ;розрахована на 1з
MOV TMOD,#MODE ;ініціалізація режимів таймерів
MOV TCON,#STR ;запуск рахунку
JNB TFO,$ ;перевірка прапора переповнення T/30
CLR TR1 ;якщо відбулося переповнення те
CLR TR0 ;зупинити рахунок T/31 і T/30
MOV R0,TL1 ;зчитування результату виміру
MOV R1,TH1 ;R0 містить молодшу частину а R1 старшу
CALL IND ;виклик підпрограми індикації
MOV R3,#01h ;тимчасова затримка
M: MOV R4,#01h ;на двох регістрах
DJNZ R4,$ ;з декрементом
DJNZ R3,M ;у вкладеному циклі
JMP BEG ;перехід на наступний цикл виміру
IND: ;підпрограма індикації, робить переклад результату
;із двійкової системи у двійково-десяткову
;i відображення
JMP M1
L: DB 01h,02h,04h,08h,16h,32h,64h,28h,56h,12h,24h,48h,96h
HI: DB 00h,00h,00h,00h,00h,00h,00h,01h,02h,05h,10h,20h,40h
M1: MOV R2,#00h
MOV R6,#00h
MOV R7,#00h
NACHALO_L: MOV A,R0
JNB ACC.0,AGAIN_L
MOV A,R2
MOV DPTR,#L
MOVC A,@A+DPTR
ADD A,R6
DA A
MOV R6,A
JNB PSW.7,PLUS_NEXT_L
MOV A,R7
ADD A,#01H
DA A
MOV R7,A
PLUS_NEXT_L: MOV A,R2
MOV DPTR,#HI
MOVC A,@A+DPTR
ADD A,R7
DA A
MOV R7,A
AGAIN_L: MOV A,R0
RRC A
MOV R0,A
INC R2
CJNE R2,#08h,NACHALO_L
NACHALO_HI: MOV A,R1
JNB ACC.0,AGAIN_HI
MOV A,R2
MOV DPTR,#L
```

```

        MOVC A,@A+DPTR
        ADD  A,R6
        DA   A
        MOV  R6,A
        JNB  PSW.7,PLUS_NEXT_HI
        MOV  A,R7
        ADD  A,#01H
        DA   A
        MOV  R7,A
PLUS_NEXT_HI:  MOV  A,R2
        MOV  DPTR,#HI
        MOVC A,@A+DPTR
        ADD  A,R7
        DA   A
        MOV  R7,A
AGAIN_HI:  MOV  A,R1
        RRC  A
        MOV  R1,A
        INC  R2
        CJNE R2,#0Dh,NACHALO_HI
        MOV  A,R6
        MOV  DPTR,#0B000h          ;Вивід на індикацію
        MOVX @DPTR,A              ;молодшої частини коду
        MOV  A,R7
        MOV  DPTR,#0A000h          ;Вивід на індикацію
        MOVX @DPTR,A              ;старшої частини коду
        RET
        END

```

**Виміряти частоту, змінного генератора, регульовану резистором R4 (програма написана мовою 3) .**

```

int main()
{
    unsigned int i;
    xdata unsigned int freq[16];
    unsigned char f_index=0,index;

start:

    TL1=0; TH1=0;    // обнулити t/c1
    TL0=0;           // 7,37MHz/12 = 614400Hz 0,01sec => 6144d або 1800h
    TH0=0xe8;        // предствановлення t/c0 = 10000h - 1800h = e800h
    TMOD=_01010001;  // c/t1-counter, c/t0-timer, mode 1 for both
    TCON=_01010000;  // почали
    while(!TF0);      // очікуємо переповнення t/c0 (через 0,1 сек)
    TCON=0;           // стіп
    i=(TH1<8) | TL1;   // запам'ятовуємо

    if (++f_index == 16) f_index=0;
    freq[f_index] = i;    // вус
    i=0;                 // ред
    for (index=0; index<16; index++){ // ня
        i = i + freq[f_index];        // ім
    }
    i = i >> 4;
    display_int(i,_01000000);
    delay16(20000);
    if (key_scan() == 12) goto start;
    lights_off();
}

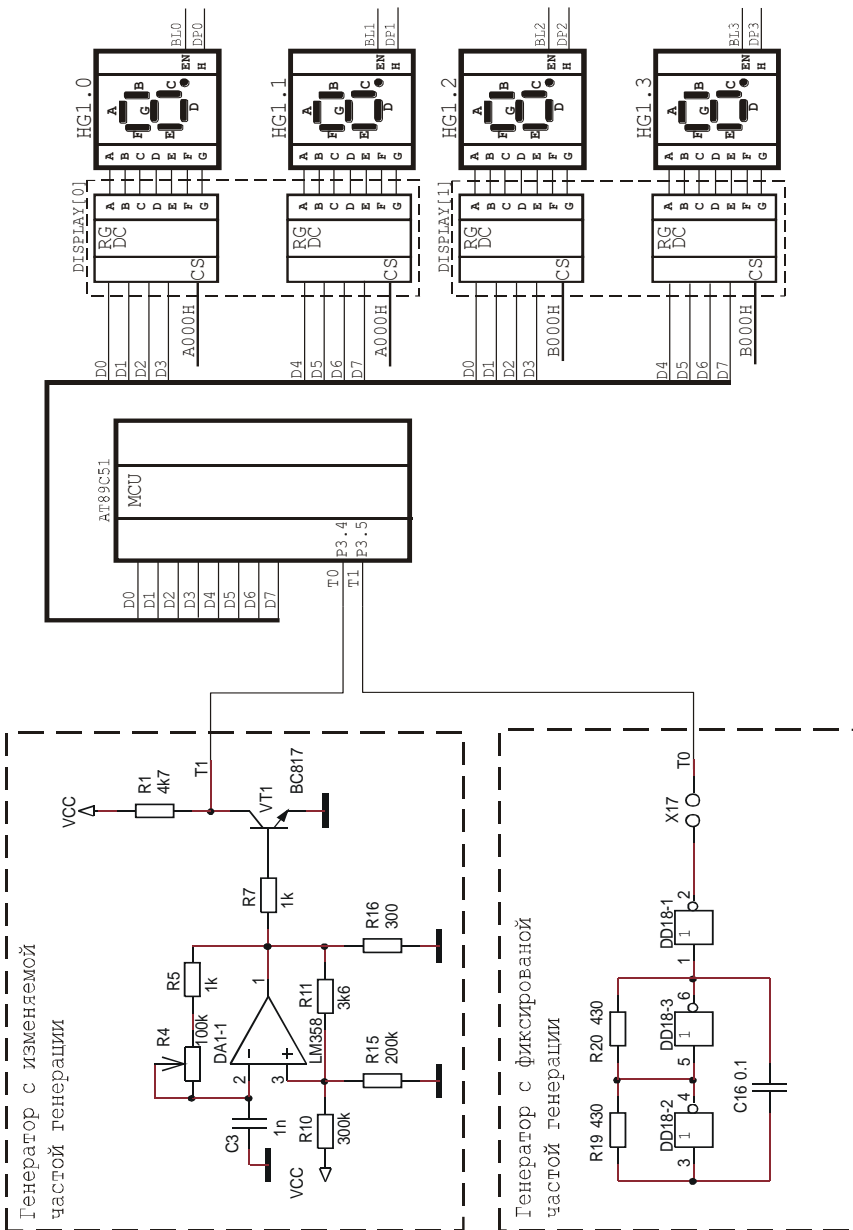
```

### *Варіанти індивідуальних завдань*

Завдання до л. р. №6

Табл.6.

<i>№</i>	<i>Текст індивідуального завдання</i>
1	Виміряти частоту T0, відобразити на статичному індикаторі.
2	Виміряти частоту T1 відносно ГФЧ, відобразити на статичному індикаторі.
3	Виміряти частоту T1 щодо частоти роботи процесора відобразити на статичному індикаторі.
4	Виміряти різницю частот ГФЧ і ГЗЧ відобразити на динамічному індикаторі.
5	З використанням внутрішніх Т/Л забезпечити плавне загорання числа 55 на статичному індикаторі.
6	По черзі відображати 48 на другому й третьому знакомісці статичної індикації й на першому й четвертому знакомісці статичної індикації с інтервалом 1с. Часові інтервали формувати ГФЧ
7	Підрахувати й відображати на динамічному індикаторі кількість натискань SW3 за 10с. Часові інтервали формувати Т/Л ОЕОМ
8	Відображати на другому знакомісці статичної індикації числа від 1 до 9 протягом 1с, і паузою між відображеннями 1с. Часові інтервали формувати Т/Л ОЕОМ
9	При натисканні кнопки SW4 включити секундомір з відображенням на динамічному індикаторі, з використанням Т/З ОЕОМ
10	Відображати миготливі числа на чотирьох розрядах статичної індикації з різним часом відображення паузи задавати рівні 0.25, 0.5, 0.75, 1 сек. За допомогою Т\Л ОЕОМ.
11	Підрахувати кількість натискань кнопки SW6 за 5с. Часові інтервали формувати Т/Л ОЕОМ
12	Виміряти час між натисканнями кнопок SW7 й SW8.Час у сек. Відобразити на статичній індикації.
13	При натисканні кнопки SW9 відобразити в секундах час її натискання на статичній індикації.
14	Виміряти період T0 відобразити на статичній індикації.
15	Виміряти період T1 відобразити на статичній індикації.





## **Лабораторна робота №7. Вивчення інтерфейсу I2C**

**Тема:** Вивчення інтегральних мікросхем з послідовним вводом-виводом інформації.

**Мета роботи:** Навчитися здійснювати обмін між пристроями за допомогою інтерфейсу I2C.

**Навчальне завдання:** Розробка програми для OEOM KP1816BE31 формуюча I2C інтерфейс.

### ***Порядок виконання лабораторної роботи***

1) Вивчити методи послідовної передачі даних між інтегральними мікросхемами.

2) Розробити алгоритм для виконання індивідуального завдання до початку лабораторного заняття

3) Розробити програму для виконання індивідуального завдання до початку лабораторного заняття

4) Ввести програму індивідуального завдання на персональному комп'ютері.

5) За допомогою ПНЗ проаналізувати виконання індивідуальної програми

6) Завантажити програму в стенд OEOM. Переконавшись в правильному виконанні індивідуального завдання, при негативному результаті здійснити зміну алгоритму або програми. Повторити завантаження програми в стенд OEOM

7) Роздрукувати листинг правильно працюючої програми.

8) Відповісти на контрольні питання викладача

### ***Короткі теоретичні відомості***

#### ***Термінологія шини I<sup>2</sup>C***

**Передавач** - елемент, що ПНЗилає дані в шину.

**Приймач** - елемент, що отримує дані із шини.

**Master** - елемент, що ініціює передачу, генерує сигнали синхронізації, і завершує передачу.

**Slave** (підлеглий) - елемент, адресований майстром

**Multi-master** - система з більш ніж одним майстром.

**Арбітраж** - процедура, що гарантує, що, якщо більше ніж один майстер одночасно спробують управляти шиною, тільки один з них одержить повне керування шиною й забезпечить передачу даних без помилок.

**Синхронізація** - синхронізація передачі даних між елементами.

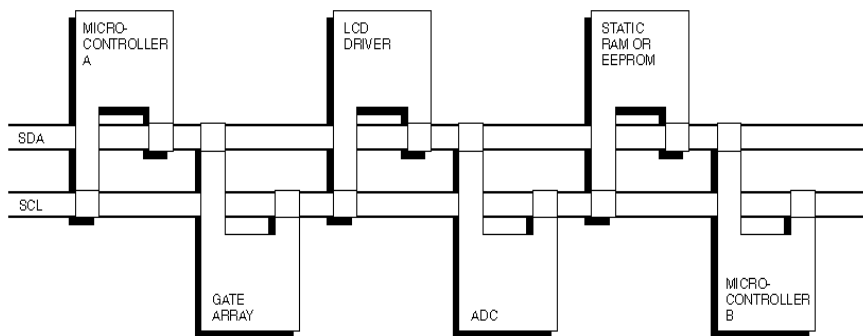


Рис.17. Приклад конфігурації шини I<sup>2</sup>C, яка використовує два мікроконтролери

### *Загальні характеристики*

SDA й SCL - двонаправлені лінії з відкритим стоком (колектором), з'єднані з позитивною живлячою напругою через резистор (див. рис. 18). Коли шина вільна, обидві лінії перебувають у стані "1". Дані можуть передаватися по шині I<sup>2</sup>C зі швидкістю до 100 kbit/s у стандартному режимі, або до 400 kbit/s у швидкому режимі. Число інтерфейсів, з'єднаних із шиною винятково залежить від ємності шини, максимальне значення якої - 400р.

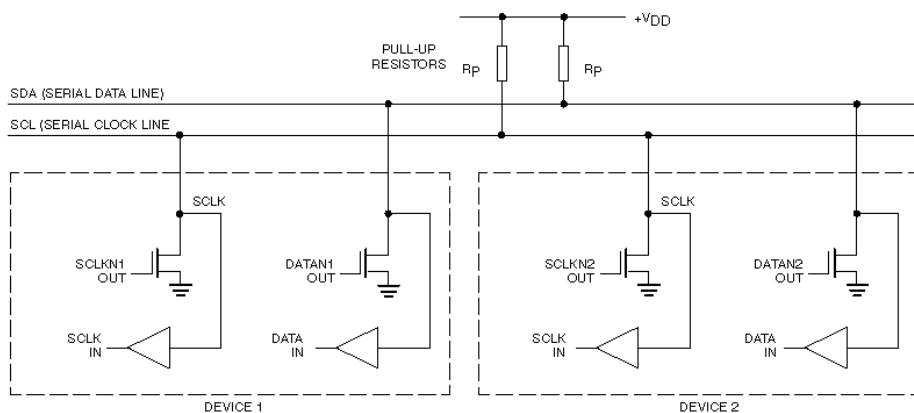


Рис.18. З'єднання I<sup>2</sup>C пристроїв

### *Передача біта*

Через розмаїтість різних технологій (CMOS, NMOS, біполярні прилади) які можуть бути використані із шиною I<sup>2</sup>C, рівні логічного “0” й “1” не встановлені й залежать від величини живлячої напруги VDD. Для передачі одного біта даних використовується один імпульс сигналу синхронізації, при цьому рівень на лінії SDA повинен бути незмінним протягом високого рівня на лінії SCL, і може змінюватися тільки при низькому рівні на SCL (Малюнок 19). Виключеннями є два особливих стани - START й STOP.

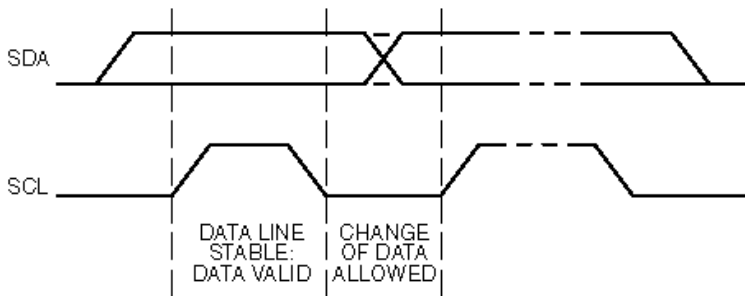


Рис.19.Передача біта.

### **START й STOP.**

Існують два особливих стани шини I<sup>2</sup>C - start й stop, які служать для індикації початку й кінця передачі й відповідно переходу шини в неактивний стан. Слід зазначити, що доти, поки не встановлений стан start, сигнали на лініях SDA й SCL можуть бути зовсім довільними (малюнок 19). Це дозволяє, зокрема, використати одну лінію SDA і кілька ліній SCL (наприклад, при недостатці, адрес на одній шині).

**Стан START** - перехід від “1” до “0” на лінії SDA при “1” на лінії SCL.

**Стан STOP** - перехід від “0” до “1” на лінії SDA при “1” на лінії SCL.

Ці два стани завжди генеруються майстром.

Детектування станів start й stop у спеціалізованих елементах звичайно відбувається апаратно. При повністю програмній реалізації шини I<sup>2</sup>C у мікроконтролерах без апаратної I<sup>2</sup>C-частини необхідно як мінімум 2 рази перевіряти стан лінії SDA.

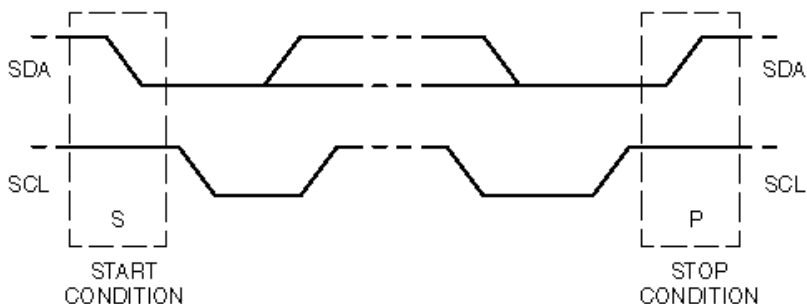


Рис.20.START й STOP стану.

### *Передача даних*

Всі передачі формуються 8-розрядними байтами. Число байтів, які можуть бути передані за одну передачу не обмежено. Кожен байт повинен супроводжуватися бітом підтвердження (ACK). Дані передаються починаючи зі старшого біта (MSB), (рис. 20).

Якщо приймач не може отримати інший повний байт даних, він не видає сигнал ACK, що використовується передавачем для синхронізації або сигналізації про несправності приймача (або його відсутності).

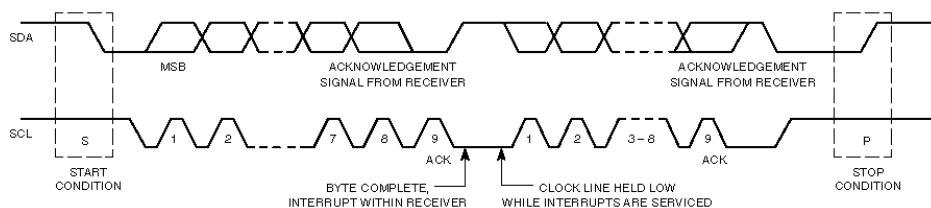


Рис.21.Передача даних по шині I<sup>2</sup>C.

### **ACKNOWLEDGE**

Для підтвердження передачі байта передавач установлює лінію SDA в “1” протягом синхронізуючого імпульсу. Приймач

при цьому повинен виставити “0” на SDA (рис. 22). Природньо при цьому повинні бити витримані певні часові співвідношення.

Звичайно, приймач, який був адресований, повинен генерувати сигнал ACK після того, як кожен байт був отриманий, за винятком того, коли повідомлення починається з адресою CBUS.

Коли підлеглий приймач не підтверджує підлеглу адресу (наприклад, пристрій не готовий, тому що ще виконує деяку внутрішню функцію), лінія SDA даних повинна бути залишена в “1”. Майстер потім може видати стан STOP, щоб перервати передачу.

Якщо підлеглий приймач підтверджує підлеглу адресу, але, якийсь час пізніше не може отримувати більше байти даних, майстер повинен призупинити передачу.

При прийомі останнього байта в серії, замість сигналу ACK майстер може виставити стан STOP, при цьому підлеглий передавач повинен звільнити лінію даних.

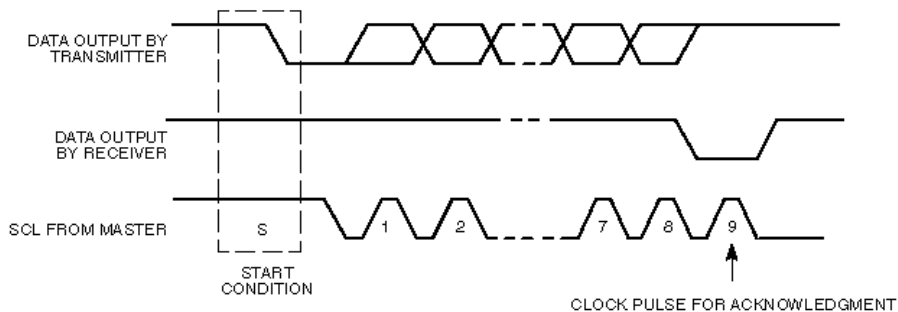


Рис.22.Підтвердження передачі.

### *Контрольні питання*

1. Призначення й сфера застосування інтегральних пристроїв з послідовним інтерфейсом
2. Часові параметри сигналів по інтерфейсу I2c.
3. Протокол обміну по I2c
4. Призначення зовнішніх адресних ліній на мікросхемах з I2c.
5. Використання декількох пристроїв на одному інтерфейсі.
6. Методи адресації різних пристроїв з I2c

## **Лабораторна робота №8. Програмування ПНЗлідового порту OEOM**

**Тема:** Програмування послідовного порту OEOM.

**Мета роботи:** Навчитися здійснювати передачу даних по інтерфейсу RS232.

**Навчальне завдання:** Розробка програми для OEOM KP1816BE31 на базі навчального стенда для зв'язку й передачі інформації від стенда на персональний комп'ютер і назад і зв'язку двох навчальних стендів між собою за допомогою інтерфейсу RS232

### ***Порядок виконання лабораторної роботи***

- 1) Вивчити методи ПНЗлідової передачі даних між периферійними пристроями.
- 2) Розробити алгоритм і програму для виконання індивідуального завдання до початку лабораторного заняття
- 3) Ввести програму індивідуального завдання на персональному комп'ютері.
- 4) Завантажити програму в стенд OEOM. Переконатися в правильному виконанні індивідуального завдання, при негативному результаті здійснити зміну алгоритму або програми. Повторити завантаження програми в стенд OEOM
- 5) Роздрукувати листинг правильно працюючої програми.
- 6) Відповісти на контрольні питання викладача

### ***Контрольні питання***

1. Призначення й сфера застосування передачі даних по послідовному інтерфейсу
2. Фізична характеристика сигналів в інтерфейсі RS232.
3. Часові параметри, визначення швидкості обміну інтерфейсу RS232.
4. Протокол обміну інтерфейсу RS232.
5. Синхронні й асинхронні канали послідовної передачі даних.
6. Регістри Т/Л використовувані при ініціалізації послідовного порту OEOM.
7. Адресація послідовних портів PC

### Перелік літератури

- 1.Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC п/р У.Томпкинса и Дж. Уэбстера, Москва, "Мир",1992
- 2.Быстродействующие интегральные микросхемы ЦАП и АЦП п/р А.К.Марцинкявичуса, Москва, "Радио и связь",1990
- 3.Основы автоматики, импульсной и вычислительной техники П.М.Грицевский, А.Е.Мамченко, Москва, "радио",1989
- 4.Интегральная электроника в измерительных устройствах В.С.Гутников Ленинград "Энегроатомиздат",1990
- 5.В.И.Зубчук Справочник по цифровой схмотехнике, Киев "Тэхника"1990
- 6.С.В.Якубовский "Цифровые и аналоговые интегральные микросхемы" Справочник. Москва "Радио и связь",1990
- 7.В.П.Шило "Популярные цифровые микросхемы" Москва "Радио и связь" 1988
- 8.Ф.В.Кушнир "Радиотехнические измерения" Москва "Связь" 1986
- 10.Абрайтис В.Б. "Быстродействующие интегральные микросхемы" "Электронная промышленность", 1989
11. Шевкопляс Б.В. Микропроцессорные структуры , "Радио и связь" 1990
- 12.Г. Виглеб. Датчики. М.: Мир, 1989.
- 13.Г.П.Липовецкий. Однокристалльные микроЭВМ. М.: МП "БИНОМ",1992.
- 14.С.Т. Хвощ. Микропроцессоры и микроЭВМ в системах автоматизации и управления. Справочник. М.: Машиностроение, 1987.
- 15.В.С. Гутников. Интегральная электроника в измерительных устройствах. Л.: Энергоатомиздат, 1988.
- 16.Аналоговые интегральные схемы: Справ./ А.Л.Булычѳв. В.И.Галкин, В.А.Прохоренко.- Мн.: Беларусь, 1993.- 382с.
- 17.Однокристалльные микро-ЭВМ. Техническое описание и руководство по применению / Г.П. Литвинский., Москва , 1982.
- 18.Ю.В. Новиков Разработка устройств сопряжения