

**Министерство образования и науки Украины
Донбасская государственная машиностроительная академия
Кафедра «Обработка металлов давлением»**

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
к самостоятельной работе
по дисциплине
«Автоматизация производственных процессов и
микропроцессорная техника»
(для студентов специальности 7.090404)

Краматорск 2006

ЦЕНТРАЛЬНЫЙ ПРОЦЕССОР И ПАМЯТЬ КОМПЬЮТЕРА

Цель работы:

Ознакомиться с организацией процессора и основной памяти компьютера и их взаимодействием при выполнении команд программы. Получить навыки работы с ассемблерными отладчиками.

Теоретические сведения

Главным средством обработки информации в ПЭВМ является центральный процессор (ЦП), выполненный, как правило, в виде одной микросхемы высокой степени интеграции, в связи с чем его также называют микропроцессором (МП). Программист, работающий с ПЭВМ, имеет доступ к ресурсам МП, к числу которых относятся программно-доступные регистры. Регистры представляют собой запоминающие устройства фиксированной разрядности, расположенные непосредственно на кристалле процессора. В них можно заносить и из них можно считывать данные с помощью команд МП. Совокупность программно-доступных регистров определяет логическую организацию МП.

Регистры используются для управления выполняющейся программой, для адресации памяти и для обеспечения арифметических вычислений арифметико-логическим устройством (АЛУ). Каждый регистр имеет длину в одно слово и адресуется по имени. Биты регистра принято нумеровать справа налево:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
----	----	----	----	----	----	---	---	---	---	---	---	---	---	---	---

На рис.1.1 показана логическая организация, а на рис.1.2 – общая структура МП Intel 8086, в составе которого имеется 14 программно-доступных регистров разрядностью 16 бит. Процессоры последующих поколений этой серии (i80286, i80386 и т.д.) имеют ряд дополнительных регистров, обычно более высокой разрядности. Эти регистры здесь не рассматриваются.

Важным компонентом ПЭВМ является оперативная память (ОП). Логически она организована в виде последовательности байтов с адресами от 0 до $2^n - 1$, где n – число разрядов адреса. Адресуемая область памяти МП составляет 1 Мбайт ($1\text{Мбайт}=1048576=2^{20}$ байт), и, следовательно, формат адреса равен 20 бит. Сам МП манипулирует логическими адресами, содержащими 16-разрядный базовый (сегментный) адрес и 16-разрядное внутрисегментное смещение. Механизм сегментации памяти предполагает разбиение всего адресуемого пространства на области-сегменты по 64 Кбайт каждая (рис.1.3).

Сегментом называется область памяти, которая начинается на границе параграфа, т.е. по любому адресу, кратному 16. Хотя сегмент может располагаться в любом месте памяти и иметь размер до 64 Кбайт, он требует столько памяти, сколько необходимо для выполнения программы.

Одновременно процессор может использовать четыре сегмента :

1. *Сегмент кодов.* Сегмент кодов содержит машинные команды программы, которые будут выполняться. Обычно первая выполняемая команда находится в начале этого сегмента и операционная система передает управление по адресу данного сегмента для выполнения программы, которая адресует данный сегмент.

РЕГИСТРЫ ОБЩЕГО НАЗНАЧЕНИЯ (РОН)				УКАЗАТЕЛЬНЫЕ И ИНДЕКСНЫЕ РЕГИСТРЫ			
15	8	7	0	15		0	
AX	AH		AL	аккумулятор	SP (Stack Pointer)		указатель стека
BX	BH		BL	база	BP (Base Pointer)		указатель базы
CX	CH		CL	счетчик	SI (Source Index)		индекс операнда
DX	DH		DL	данные	DI (Destination Index)		индекс результата

СЕГМЕНТНЫЕ РЕГИСТРЫ				РЕГИСТР АДРЕСА КОМАНД			
15			0	15			0
CS (Code Segment)			сегмент кода	IP (Instruction Pointer)			
DS (Data Segment)			сегмент данных				
SS (Stack Segment)			сегмент стека				
ES (Extra Segment)			сегмент экстракодов				

РЕГИСТР ПРИЗНАКОВ (ФЛАГИ)

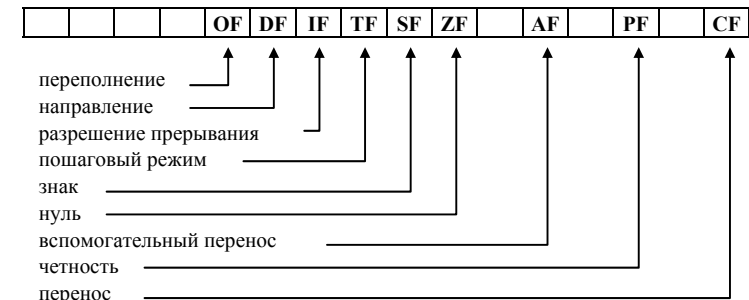


Рис. 1.1 Логическая организация МП Intel 8086

2. *Сегмент данных.* Сегмент данных содержит определенные данные, константы и рабочие области, необходимые программе.
3. *Сегмент стека.* Стек содержит адреса возврата как для программы при возврате в операционную систему, так и для вызовов подпрограмм при возврате в главную программу.
4. *Экстракодовый сегмент* используется для передачи данных между ячейками ОП, не попадающими в непрерывную область объемом 64 К байт.

Начальный адрес любого сегмента хранится в 16-разрядном сегментном регистре, а обращение к байту или слову внутри сегмента осуществляется с использованием 16-

разрядного смещения. Большинство команд МП оперирует только 16-разрядными смещениями, а адреса сегментов находятся в одном из 4-х регистров: CS – код (программа); DS – данные; SS – стек; ES – экстракод (дополнительные данные).



Рис. 1.2 Структура процессора Intel 8086

Для обращения к любому адресу, заданному в программе, компьютер по специальной схеме складывает адрес в регистре сегмента и смещение. Так как начало сегмента всегда находится на границе параграфа, младшие (правые) четыре бита его начального адреса равны нулю. Специалисты решили, что нет смысла иметь место для битов, которые всегда равны нулю. Поэтому адрес хранится в сегментном регистре как шестнадцатиричное $NNNN_{16}$, а компьютер полагает, что имеются еще четыре нулевых младших бита (одна шестнадцатиричная цифра), т.е. $NNN0_{16}$. Таким образом, $FFFF0_{16}$ позволяет адресовать до $1048560 \approx 2^{20}$ байт.

Сегментные регистры: CS, DS, SS и ES

Каждый сегментный регистр обеспечивает адресацию памяти объемом 64 Кбайт, которая называется текущим сегментом.

1. *Регистр CS.* Регистр сегмента кода содержит начальный адрес сегмента кода. Этот адрес плюс значение смещения в командном указателе (IP) определяет адрес команды,

- которая должна быть выбрана для выполнения.
2. *Регистр DS.* Регистр сегмента данных содержит начальный адрес сегмента данных. Этот адрес плюс смещение, определенное в команде, указывают на конкретную ячейку в сегменте данных.
3. *Регистр SS.* Регистр сегмента стека содержит начальный адрес сегмента стека.
4. *Регистр ES.* Некоторые операции над строками используют дополнительный сегментный регистр для управления адресацией памяти.

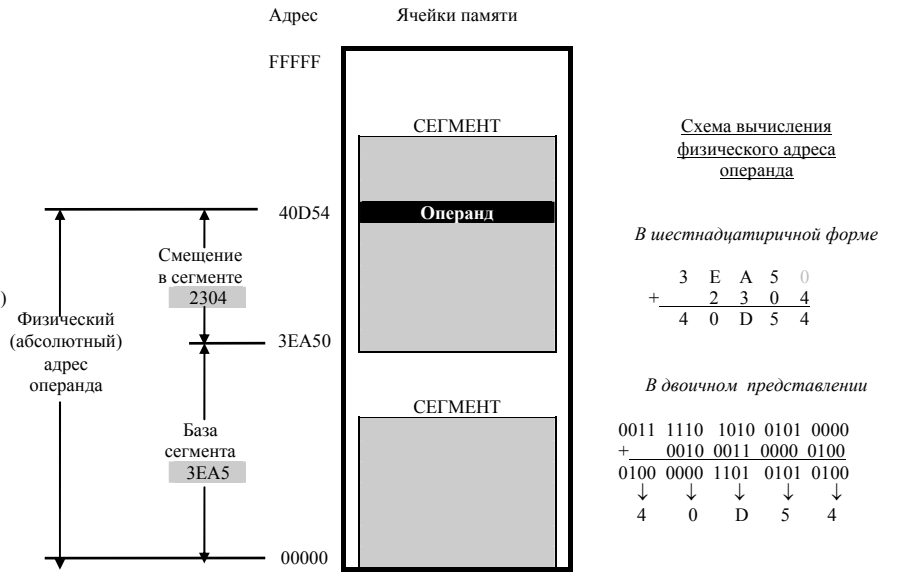


Рис. 1.3 Адресация памяти с использованием механизма сегментации

Регистры общего назначения: AX, BX, CX и DX

Регистры общего назначения (РОН) являются наиболее интенсивно используемыми при вычислениях. Особенность этих регистров состоит в том, что возможна адресация их как одного целого слова (16 байт) или как однобайтовой части. Левый байт является старшей частью (High), а правый – младшей частью (Low). Например, двухбайтовый регистр CX состоит из двух однобайтовых CH и CL, и ссылки на регистр возможны по любому из этих трех имен.

Все РОН участвуют в выполнении арифметических и логических операций, представляя операнды и фиксируя результат. Наряду с этим имеются некоторые команды, которые специализируют РОН.

Особенности РОН.

1. *Регистр AX.* Регистр AX является основным сумматором и применяется для всех операций ввода-вывода, некоторых операций над строками и некоторых арифметических операций. Например, команды умножения, деления и сдвига предполагают обязательное использование регистра AX. Некоторые команды генерируют более эффективный код, если они имеют ссылки на регистр AX.

2. *Регистр BX.* Регистр BX является базовым регистром. Это единственный регистр общего назначения, который может использоваться в качестве “индекса” для расширенной адресации.

3. *Регистр CX.* Используется как счетчик для управления числом повторений циклов и для операций сдвига влево или вправо.

4. *Регистр DX.* Регистр DX является регистром данных. Он применяется для некоторых операций ввода-вывода и тех операций умножения и деления над большими числами, которые используют регистровую пару DX:AX.

Регистры SP, BP, SI, DI, IP, флаги

Четыре 16-разрядных указательных и индексных регистра (SP, BP, SI, DI) предназначены для хранения внутрисегментных смещений. Эти же регистры могут участвовать в выполнении арифметических и логических операций над двухбайтными словами.

Регистры SP и BP используются для доступа к данным в текущем сегменте стека, а *SI и DI* - в текущем сегменте данных.

Регистр адреса (или счетчик команд) IP применяется для выбора команд программы в текущем сегменте кода CS. Это единственный регистр, содержимое которого всегда обновляется при выполнении каждой команды программы.

Регистр признаков (флагов) предназначен для хранения данных о различных ситуациях, возникающих в программе, а также для задания режимов обработки данных процессором.

Девять из 16 бит флагового регистра являются активными и определяют текущее состояние машины и результаты выполнения команд программы. Фактически, регистр признаков представляет собой совокупность нескольких одноразрядных регистров, объединенных в один регистр стандартного объема (16 бит), при этом содержимое добавленных 7 бит устанавливается всегда одинаковым и игнорируется процессором. Многие арифметические, логические команды и команды сравнения изменяют состояние флагов.

При программировании на Ассемблере наиболее часто используются флаги OF, SF, ZF, CF для арифметических операций и операций сравнения, а флаг DF – для обозначения направления в операциях над строками.

Назначение флаговых битов:

Флаг	Назначение
OF (Переполнение)	Указывает на переполнение старшего бита при арифметических командах.
DF (Направление)	Обозначает левое или правое направление пересылки или сравнения строковых данных
IF (Прерывание)	Указывает на возможность внешних прерываний.
TF (Пошаговый режим)	Обеспечивает возможность работы процессора в пошаговом режиме.
SF (Знак)	Содержит результирующий знак после арифметических операций (0 – плюс, 1 – минус)
ZF (Ноль)	Показывает результат арифметических операций (0 – ненулевой, 1 – нулевой результат).
AF (Дополнит. перенос)	Содержит перенос из 3-го бита (используется для специальных арифметических операций).
PF (Четный паритет)	Показывает четность количества битов "1" в младшем байте данных (1 – четное и 0 – нечетное).
CF (Перенос)	Содержит перенос из старшего бита после арифметич. операций, а также выпавший бит при сдвигах.

Ассемблерный отладчик

Для отображения и модификации содержимого внутренних регистров МП и ячеек оперативной памяти, а также для отладки программ используют специальные программные средства – ассемблерные отладчики (Debug, AfdPro и т.п.).

В данной лабораторной работе используется ассемблерный отладчик ТИС (файл *tic.com*), имеющий достаточно удобный интерфейс и развитые средства отладки программ.

Основной рабочий экран отладчика разбит на такие функциональные зоны (рис. 1.4):

1. *Командное окно* – служит для ввода команд управления работой отладчика. Загрузка отлаживаемой программы в рабочую область памяти осуществляется командой L_<имя программы>.

2. *Программное окно* – отображает команды отлаживаемой программы в обычном виде (мнемоники Ассемблера) и в виде машинных кодов. Загруженная программа



Рис. 1.4 Функциональные окна ассемблерного отладчика

представлена в этом окне в виде 4-х колонок:

- адрес (смещение в сегменте кодов ОП) старшего байта команды;
- машинный код команды (1...6 байт);
- мнемоническое имя операции;
- операнд или операнды (если есть).

Перемещаться по тексту загруженной программы можно с помощью клавиш управления курсором.

3. *Процессорное окно* – отображает содержимое регистров МП, в том числе флажков, а также вершину стека. В этом окне можно непосредственно изменять содержимое регистров МП.

4, 5. *Окна памяти* – имеют одинаковое назначение и дают удобную возможность одновременного доступа к двум различным участкам памяти.

Содержимое памяти выведено в виде последовательности строк из 8 (правое окно) или 16 (нижнее окно) ячеек памяти. В начале каждой строки указан регистр, адресующий данный сегмент и адрес (смещение внутри этого сегмента) первой из ячеек в строке). Для перехода в другой сегмент достаточно, расположив курсор на имени сегментного регистра, указать при помощи клавиатуры другое. Аналогично в любой строке можно непосредственно указать и смещение в этом сегменте искомой ячейки.

6. *ASCII – окно*. Представляет содержимое памяти в виде символьных кодов.

7. *Строка подсказок*. Определяет действия, закрепленные за каждой из 10 функциональных клавиш.

Таблица 1.1

Основание системы счисления		
"10"	"16"	"2"
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Примечание: для компактности вся числовая информация представлена на экране отладчика в шестнадцатичном формате (соответствие между шестнадцатичными, десятичными и двоичными числами приведено в табл.1.1)

Порядок выполнения работы

1. Создать в своей директории поддиректорий **MPT_LAB1** и скопировать в него файлы **tic.com** и **lab1.exe**.

2. Запустить на выполнение ассемблерный отладчик и загрузить в командном окне программу **lab1.exe**. Освоить перемещение по тексту программы, по окнам отладчика, модификацию содержимого регистров МП и ячеек памяти и поиск нужной ячейки памяти по указанному смещению в данном сегменте.

3. Выполнить пошагово 5 первых команд программы. Считая, что сегменты программы следуют в памяти вплотную один за другим, определить расположение сегментов кода, данных и стека. Для расчета использовать данные из регистров CS, DS, SS, SP; длину сегмента кода определить по смещению последней команды программы "RET Far". Результаты изобразить в протоколе по аналогии с рис. 1.5.

4. Выполнив команды работы со стеком PUSH и POP, проследить в процессорном окне порядок заполнения/разгрузки вершины стека. Проследить за изменением содержимого счетчика команд IP при пошаговом выполнении программы, в том числе при переходах, циклах и вызовах подпрограмм.

5. Выделив в программном окне указанную преподавателем команду пересылки данных, найти ее машинное представление в памяти. Определить и найти на экране соответствующие источники и приемники информации. Проверить, выполнив команды в пошаговом режиме. Записать в протокол команду, содержимое источников и приемников данных до и после выполнения команды.

6. Для выбранной арифметической команды определить операнды и рассчитать результат. Переведя результат в двоичный формат, определить и показать преподавателю, каким будет состояние флажков SF, ZF, CF, AF и PF. Произвести проверку, выполнив команду в отладчике

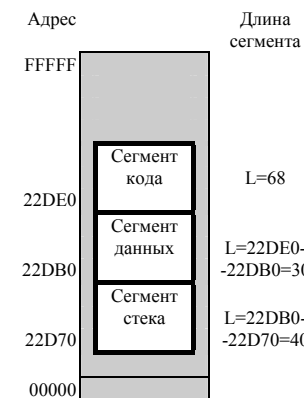


Рис.1.5 Размещение в памяти сегментов программы
CS=22DE; DS=22DB; SS=22D7; смещение "RET Far"=68

Содержание протокола работы

- ☞ Цель работы, краткие теоретические сведения.
- ☞ Схема размещения в памяти сегментов программы **lab1.exe** с расчетами их длин и базовых адресов (п.3).
- ☞ Расчеты к
- ☞ выполнению команд программы (пп. 5, 6).

ДАТЧИКИ

Мета роботи

Ознайомлення з 4 типами датчиків, їх характеристиками та визначення їх метрологічних параметрів.

Теоретичні відомості

Датчик – елемент вимірювальної системи, призначений для вимірювання параметру контролюемого об'єкту та перетворення його в сигнал, зручний для підсилення або передачі. Найбільш поширеними є датчики, що перетворюють неелектричну величину в електричну. Метрологічні характеристики датчиків визначають такі параметри:

Статична характеристика датчика – являє собою залежність зміни вихідної величини від вхідної, тобто $Y = F(X)$, де X – вхідна величина, Y – вихідна величина.

Чутливість датчика – відношення приросту вихідної величини до приросту вхідної, тобто $S = \Delta Y / \Delta X$.

Поріг чутливості датчика – найменше значення вхідної величини, яке спричиняє появу сигналу на виході. Цей параметр пов'язаний з поняттям зони нечутливості, тобто зони, в межах якої зміна вхідного сигналу не викликає зміни вихідного.

Інерційність датчика – час, протягом якого вихідна величина приймає значення, що відповідає вхідній величині.

По характеру отримання сигналу від вимірюваної величини датчики поділяють на параметричні, в яких зміна вимірюваної величини викликає зміну якогось параметру (наприклад, опору, індуктивності, ємності, тиску, т. ін.), і генераторні, в яких зміна вимірюваної величини викликає генерацію сигналу (поява терморушійної сили, фотоструму, т. ін.).

По характеру залежності вихідного сигналу від вхідного розрізняють датчики пропорційні, циклічні, імпульсні. Схеми ввімкнення вимірювальних та перетворюючих елементів датчика можуть бути диференційні, компенсаційні, мостові та ін.

Технічне забезпечення: блок живлення з стабілізованою постійною напругою +5в, +12.5в, -12.5в, мультиметр цифровий, потенціометричний датчик шляхового положення, терморезистивний датчик, тензорезистивний датчик, термopара.

Порядок виконання роботи

Вся робота розрахована на 4 години. З них 2 години – дослідження потенціометричного та терморезистивного датчиків; 2 години – тензорезистивний датчик та термopара.

1.1. Дослідження потенціометричного датчика.

Ознайомитись зі схемою датчика та намалювати її, заміряти параметри (величини опорів) елементів датчика. Підключити датчик до джерела живлення та зняти залежність вихідної напруги від шляхового положення датчика для 4-ох схем ввімкнення (мал. 1.1):

- з загальною точкою та опором навантаження, що дорівнює вхідному опору мультиметра;
- те ж з додатковим опором навантаження;
- з середньою точкою без додаткового навантаження;
- з середньою точкою та додатковим навантаженням.

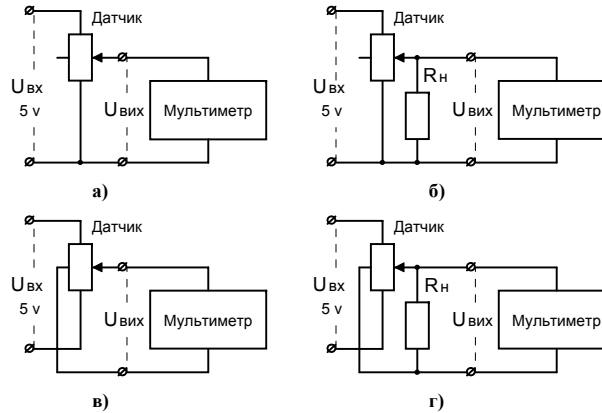
По результатам дослідження побудувати сімейство характеристик в координатах $\phi - U_{\text{вих}}$, визначити середню чутливість для кожної з схем, обчислити максимальне відхилення від лінійності, визначити внутрішній опір мультиметра.

1.2. Дослідження терморезистивного датчика.

Ознайомитись та замалювати схему датчика, підключити його до джерела живлення, регулюванням змінного опорів збалансувати міст. Заміряти опори кожного плеча моста, перевірити умову його балансу. Помістити терморезистор в середовище з відомою температурою і повторити зроблені раніше вимірювання. Визначити чутливість датчика.

Зміст протоколу роботи

- ♦ Мета роботи та коротке викладення основних теоретичних відомостей;
- ♦ Схема потенціометричного датчика;
- ♦ Графіки характеристик потенціометричного датчика та розрахунки чутливості для 4 схем ввімкнення, відхилення від лінійності та внутрішнього опорів мультиметра;
- ♦ Розрахунок чутливості терморезистивного датчика.



Мал. 1.1 Схеми ввімкнення потенціометричного датчика

ПІДСИЛЮВАЧІ СИГНАЛІВ ДАТЧИКІВ

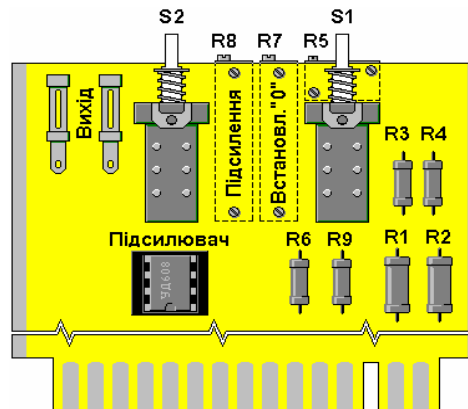
Мета роботи

Ознайомлення з підсилювачем вихідного сигналу тензометричного датчика мостового типу, виконаного на базі операційного напівпровідникового підсилювача в інтегральному виконанні типу К140УД608.

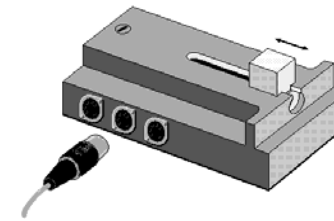
Теоретичні відомості

Вихідні сигнали первинних аналогових датчиків, як правило, дуже малі, і безпосереднє їх вимірювання та, особливо, реєстрація, практично неможливі. Тому перед реєструючою апаратурою, чи, власне, датчиком, встановлюється підсилювач, функція якого – підсилити сигнал до рівня та потужності, достатніх для його реєстрації.

Тип підсилювача залежить від характеру вихідного сигналу датчика. Для більшості датчиків вихідним сигналом є напруга (термopара, фотодіод та ін.), або зміна якогось параметру, що досить просто перетворюється в напругу (параметричні датчики). Підсилення таких сигналів виконується за допомогою електронновакуумних, напівпровідникових та, рідше, магнітних підсилювачів.

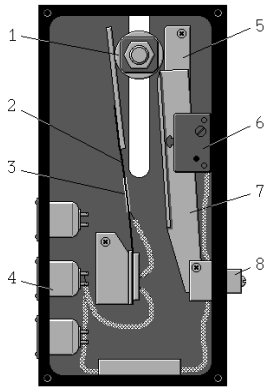


Мал. 2.1 Плата підсилювача



Мал. 2.2 Датчик переміщення (холограф)

- | | |
|-------------------------|-----------------------|
| 1 - нажимний ролик; | 2 - пружна балка; |
| 3 - тензорезистори; | 4 - штекерні роз'єми; |
| 5 - направляюча планка; | 6 - мікровимикач |
| 7 - важіль; | 8 - кронштейн |



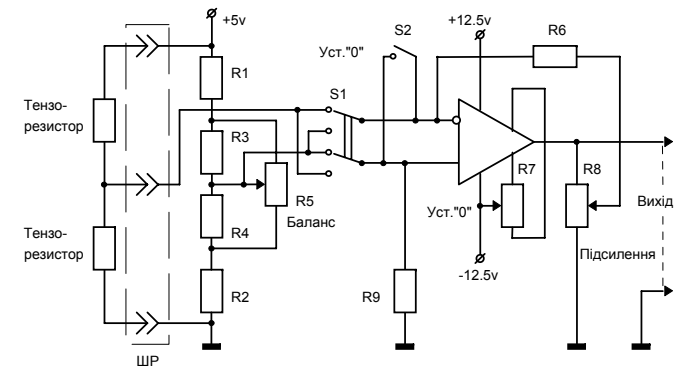
Технічне забезпечення роботи: тензометричний холограф, плата підсилювача, блок живлення, цифровий вольтметр, штангенциркуль.

Порядок виконання роботи

- Ознайомитись та замалювати схеми холографа та плати підсилювача.
- Встановити мінімальний коефіцієнт підсилення.
- Підключити підсилювач та холограф до блоку живлення і за допомогою органів регулювання підсилювача збалансувати його (встановити вихідний сигнал підсилювача мінімально можливим).
- Виконати тарування холографа, тобто знайти залежність вихідної напруги від переміщення холографа.
- Визначити коефіцієнт підсилення
- Змінити (збільшити) коефіцієнт підсилення і повторити тарування холографа та відповідні розрахунки.
- Перевести підсилювач в режим компаратора.
- Проаналізувати реакцію підсилювача в цьому режимі на зміну положення холографа.
- Зробити та занотувати висновки.

Зміст протоколу роботи

- ♦ Мета роботи та коротке викладення основних теоретичних відомостей;
- ♦ Схеми датчика та підсилювача;
- ♦ Графіки тарування холографа при різних коефіцієнтах підсилення;
- ♦ Розрахунки коефіцієнту підсилення;
- ♦ Графік, що ілюструє роботу компаратора.



Мал. 2.3 Схема підсилення сигналів тензодатчиків

АПАРАТУРА ДЛЯ РЕЄСТРАЦІЇ СИГНАЛІВ АНАЛОГОВИХ ДАТЧИКІВ

Мета роботи

Ознайомитись з світлопроменевим багатоканальним осцилографом та будовою дзеркальних гальванометрів.

Теоретичні відомості

Сигнали аналогових датчиків реєструються за допомогою електроннопроменевих, світлопроменевих осцилографів та самописців. Перші, як правило, використовуються для реєстрації параметрів, що швидко змінюються (динамічних). Для отримання "твердої копії" фотографують екран осцилографу безпосередньо в момент проведення досліду (при використанні звичайного електроннопроменевого осцилографа), або після проведення досліду (осцилограф з запам'ятовуючою трубкою).

В світлопроменевому осцилографі сигнали датчиків після підсилення записуються на фоточутливому матеріалі (плівці, папері) за допомогою сфокусованого променя, відбитого від дзеркальця спеціального дзеркального гальванометра (шлейфа). В процесі запису світлочутливий матеріал протягується з певною швидкістю в напрямку, перпендикулярному відхиленню променя. Частота процесів, що можуть бути зареєстровані світлопроменевим осцилографом, знаходиться в межах від 0 до декількох сотень герц.

В самописцях сигнали датчиків після підсилення реєструються на звичайному або спеціальному діаграмному папері за допомогою капілярного пишучого елемента спеціальними чорнилами. Частота реєструємих самописцями процесів, як правило, не вище декількох герц.

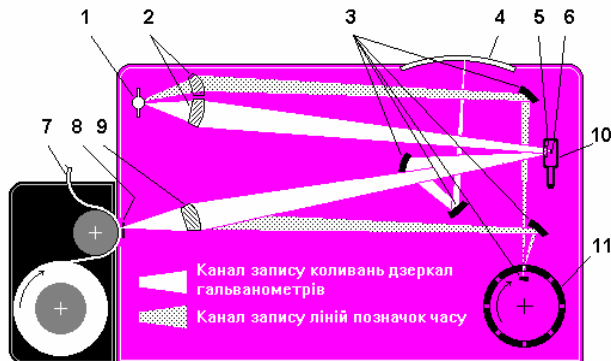
Технічне забезпечення роботи : світлопроменевий осцилограф Н117/1 з комплектом гальванометрів, дзеркальний гальванометр з внутрішньою магнітною системою, викрутка.

Порядок виконання роботи

1. Ознайомитися з конструкцією осцилографа.
2. Скласти оптичну та механічну схему осцилографа
3. Розібрати гальванометр та зробити його ескіз.

Зміст протоколу роботи

- ♦ Мета роботи та коротке викладення основних теоретичних відомостей;
- ♦ Ескізи осцилографа та гальванометра;
- ♦ Оптична схема осцилографа;
- ♦ Кінематична та електрична схеми механізму протягування світлочутливого паперу.



Мал. 3.1 Оптична схема осцилографа Н117/1 (без каналів повздовжнього графлення та регулювання положень гальванометрів)

- | | | |
|------------------------------------|-----------------------------|--------------------|
| 1 - джерело світла (ртутна лампа) | 2 - циліндричні конденсори; | 3 - дзеркала; |
| 4 - екран; | 5 - сферична лізна; | 6 - дзеркальце; |
| 8 - пластина з щільовими отворами; | 9 - об'єктив; | 10 - гальванометр; |
| 11 - барабан позначника часу | | |

ВИЗНАЧЕННЯ ЧУТЛИВОСТІ ТЕНЗОМЕТРИЧНОГО ДАТЧИКА ОПОРУ

Мета роботи

Ознайомитись з застосуванням, особливостями і основними характеристиками тензометричних датчиків.

Теоретичні відомості

Тензометричні датчики опору призначені для визначення пружних деформацій (розтягання, стиснення, згинаючих та крутих моментів) деталей машин та конструкцій в лінійному та площинно-напруженому стані при дії на них статичних та динамічних навантажень. Основою конструкції датчиків є константовий дріт діаметром 0.01...0.05 мм, зигзагоподібно складений між склесними смужками паперу або синтетичної плівки. Останнім часом, крім дротяних, використовують датчики, в яких робочий елемент виконується з фольги товщиною 4...12 мкм, а також напівпровідникові тензодатчики.

Для вимірювання деформацій датчик наклеюється на відповідним чином підготовлену поверхню контролюємої деталі. При дії на контролюєму деталь навантажень, разом з нею буде деформуватись і датчик. При цьому за рахунок зміни довжини провідників датчику, площі його поперечного перетину та фізичних властивостей матеріалу провідника, змінюється й опір датчика. По величині зміни опору датчика можна визначити і деформацію, тобто:

$$\Delta R_d = k \epsilon,$$

де ΔR_d – абсолютна зміна опору датчика,

ϵ – відносна деформація датчика (деталі),

k – коефіцієнт пропорційності, що залежить від чутливості датчика.

Чутливість датчика, або коефіцієнт передачі – відношення приросту вихідної величини (ΔY) до приросту вхідної (ΔX):

$$S = \frac{\Delta Y}{\Delta X}.$$

Для тензодатчика опору

$$S = \frac{\Delta R / R}{\epsilon},$$

де R – опір ненавантаженого датчика.

Абсолютна величина зміни опору ΔR для провідникових (дротяних та фольгових) датчиків досить мала і безпосереднє її вимірювання за допомогою омметра не дає достатньої точності. Тому, як при вимірюванні деформацій деталей, так і при визначенні чутливості датчика, його вмикають в мостову схему (міст Вінстона). Живлять міст постійним або змінним струмом та вимірюють величину вихідного сигналу розбалансованого при навантаженні мосту. Вихідний сигнал мосту, як правило, підсилюють для підвищення точності вимірювання.

Таким чином, для визначення чутливості датчика, потрібно:

- наклеїти його на деформуєму деталь,
- контролювати механічним методом відносну деформацію цієї деталі,
- вимірювати вихідний сигнал мосту після його підсилення з відомим коефіцієнтом.

В роботі для визначення чутливості на пружну балочку наклеїти 4 тензодатчики – два з одного боку та два – з другого. Кожна пара з'єднана між собою послідовно і створює в схемі по одному плечу мосту. Балочка спирається своїми кінцями на дві опори. Її середина прогинається гвинтовим механізмом. Величина цього прогину контролюється кутом повороту гвинта з відомим кроком різьби t . Датчики з базою (робочою довжиною) L розміщені симетрично відносно до опор, відстань між якими L . При такому центральному зосередженому навантаженні балочки, середня відносна деформація датчиків, як функція її найбільшого прогину (посередині) Y_{max} , визначається виразом:

$$\epsilon_{cp} = \frac{3h Y_{max}}{L^2} \left(2 - \frac{B}{L}\right),$$

де h – товщина балочки.

На увігнутій частині балочки датчики піддаються стискуванню (їх опір зменшується), а на протилежній – розтягуються, і їх опір збільшується. Послідовне з'єднання верхньої та нижньої пари датчиків в схемі мосту компенсує їх температурну нестабільність, а підключення їх в схему мосту вдвічі підвищує рівень вихідного сигналу. Чутливість тензодатчика в підсумку визначається виразом:

$$S = \frac{2U_g L^2}{U_n \cdot 3h Y_{max}} \cdot \frac{1}{2 - \frac{B}{L}},$$

де U_n – напруга живлення мосту,

U_g – вихідний сигнал розбалансу мосту.

В свою чергу розбаланс мосту можна визначити таким чином:

$$U_g = \frac{U_{nux}}{k} = \frac{U_{nux}}{\frac{R_{33}}{R_d} + 1}$$

де U_{nux} – напруга на виході підсилювача,

k – коефіцієнт підсилення підсилювача,

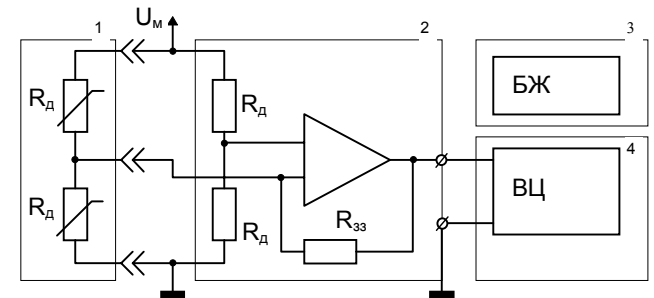
R_n – опір зворотнього зв'язку операційного підсилювача,

R_d – опір тензодатчика.

Технічне забезпечення роботи : балочка з тензодатчиками, плата підсилювача, блок живлення, цифровий вольтметр.

Порядок виконання роботи

1. Ознайомитись з приладами до роботи та визначити їх характеристики (розмір балочки, параметри тензорезисторів, підсилювача, напруги живлення мосту та ін.).
2. Зібрати електричну схему за малюнком 4.1.
3. Обертанням гвинта навантажити балочку до вибору зорів в опорах.
4. Подати живлення на схему і збалансувати мостову схему органами настройки підсилювача (на схемі не показано). Навантажити балочку (прогнути її на 0,5...2 мм з кроком, заданим керівником роботи), фіксуючи за допомогою цифрового вольтметра вихідний сигнал при кожному навантаженні. Повторити ті ж дії при розвантаженні балочки з тим же кроком. Обчислити при кожному навантаженні чутливість.



Мал. 4.1 Схема до визначення чутливості тензодатчиків

- | | |
|---------------------------|---|
| 1 – балочка з напівмостом | 2 – плата масштабіного операційного підсилювача |
| 3 – блок живлення | 4 – вольтметр цифровий |

Зміст протоколу роботи

- ♦ Короткі теоретичні відомості по тензодатчикам.
- ♦ Виконати ескіз з вказанням розмірів балочки та датчиків.
- ♦ Навести схему вимірювання коефіцієнту підсилення.
- ♦ Результати обчислення коефіцієнту чутливості звести до таблиці:

	Y _{max1}			Y _{max2}			Y _{max3}		
	ε _{cp}	U _{nux}	S	ε _{cp}	U _{nux}	S	ε _{cp}	U _{nux}	S
Навантаження									
Розвантаження									

- ♦ За результатами обчислень побудувати графік $S = f(\epsilon_{cp})$ та зробити висновки.