МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

ХАРКІВСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ХАРЧУВАННЯ ТА ТОРГІВЛІ

О.І. ТОРЯНИК, О.Г. ДЬЯКОВ, М.А. ЧЕКАНОВ, Е.О. ІШТВАН

МІКРОПРОЦЕСОРНА ТЕХНІКА

Конспект лекцій

Для студентів напрямку підготовки 6.050502 “Інженерна механіка”

(скорочена форма навчання)

Харків
2013

Укладачі:

ТОРЯНИК Олександр Іванович, д-р. хім. наук, проф.
ДЬЯКОВ Олександр Георгійович, канд. техн. наук, доц.
ЧЕКАНОВ Микола Анатолійович, канд. техн. наук, доц.
ІШТВАН Єгор Олексійович, асистент

Конспект лекцій обговорено та затверджено на засіданні кафедри енергетики та фізики, протокол засідання
№ 11 від «26» лютого 2013 р.

Зав. кафедри _____ д-р. техн. наук, проф. М.І. Погожих
(підпис)

Схвалено науково-методичною комісією факультету обладнання та технічного сервісу, протокол засідання
№ 07 від «25» 03.2013 р.

Голова комісії _____ канд. техн. наук, доц. Д.П. Семенюк
(підпис)

Рецензент _____ д-р. техн. наук, проф. В.О. Потапов
(підпис)

ЗМІСТ

ВСТУП

На теперішній час мікропроцесорна техніка активно входить у наше повсякденне життя, поступово заміщаючи й витісняючи традиційну цифрову техніку. Універсальність, гнучкість, простота проектування апаратури, практично необмежені можливості по ускладненню алгоритмів обробки інформації - все це обіцяє мікропроцесорній техніці велике майбутнє. На частку традиційної цифрової техніки залишаються тільки вузли й пристрої, що вимагають максимальну швидкодію, а також пристрої з найпростішими алгоритмами обробки інформації. Звичайна цифрова техніка сьогодні застосовується для збільшення можливостей мікропроцесорних систем, для їхнього сполучення із зовнішніми пристроями, для збільшення їхніх можливостей, тобто грає, по суті, допоміжну роль. Таким чином, традиційну цифрову техніку у самому недалекому майбутньому, чекає доля аналогової техніки, область застосування якої у свій час сильно звузилася з появою цифрової.

Останнім часом видано чимало підручників, присвячених мікропроцесорній техніці, внутрішній архітектурі мікропроцесорів, мікроконтролерів, мікрокомп'ютерів, принципам організації обміну інформації й методам побудови систем па їхній основі.

Конспект лекцій призначений для студентів факультету обладнання та технічного сервісу, які займаються по скороченій програмі навчання та мають повне уявлення про основи електроніки. Необхідність знайомства студентів з основами мікропроцесорної техніки обумовлена тим, що практично всі сучасні прилади, що використовуються в харчовій промисловості мають мікропроцесорну систему управління. Крім того широке впровадження мікроконтролерів, які мають спрощену систему програмування розширюють межі використання можливостей обладнання. Матеріали даних лекцій також можуть бути використані для першого знайомства з мікропроцесорною технікою і для студентів інших спеціальностей університету.

ЛЕКЦІЯ 1. ОБЧИСЛЮВАЛЬНА ТЕХНІКА УПРАВЛІННЯ ТЕХНОЛОГІЧНИМ ПРОЦЕСАМИ

1.1 Обчислювальна техніка та її значення для розвитку технічного прогресу. Етапи розвитку.

Використання обчислювальної техніки для автоматичного управління різноманітними процесами є головна особливість технічного розвитку сучасного суспільства. Швидкі темпи розвитку цифрових методів обробки інформації поширюють розробку та впровадження в практику обчислень та управління виробництвом мікропроцесорних засобів (персональних комп'ютерів та відповідного периферійного обладнання) обробки інформації. Практично не одна промислова система не працює без тієї чи іншої форми керування. Цифрові електронно обчислювальні машини (ЕОМ) – комп'ютери – грають основну роль; в деяких випадках не існує реальної альтернативи щодо комп'ютерного управління процесами. Всі вони створюються на основі мікропроцесорних систем (МП). Можна виділити чотири основних напрямків використання:

- вбудовані системи контролю та управління;
- локальні системи накоплення та обробки інформації;
- розподіленні системи управління складними об'єктами;
- розподіленні системи паралельних обчислень.

Відносно галузі харчових виробництв найбільше значення мають вбудовані системи контролю та управління. Використання МП у відносно простих схемах керування принципово змінює якість функціонування приладів. Що обслуговуються. Вона дозволяє оптимізувати режими роботи приладів або процесів з метою здобуття відповідних техніко-економічних ефектів. Прямий техніко-економічний ефект може бути виражений в економії енергії споживання, підвищення часу роботи обладнання, зменшенні споживання матеріальних ресурсів. Побічний ефект може полягати у зниженні вимог до персоналу та збільшенням виробництва. Управління обладнанням на основі вбудованих систем управління створює реальні передумови для реалізації повністю автоматичних виробництв.

Для визначення особливої ролі комп'ютерів в управлінні необхідно визначити що розуміється під терміном «процес».Під фізичним процесом розуміють послідовність зміни стану об'єктів фізичного світу. Прикладом процесу може бути виготовлення продуктів харчування на основі певних речовин. Під технічним процесом розуміють зміну фізичних величин які можна вимірювати та змінювати технічними засобами. Це означає що фізичний процес не обов'язково може керуватися зовні, а технічний процес передбачає обробку інформації для досягнення заданої цільової функції.

Будь який фізичний процес має вхід та вихід у вигляді:

- матеріальних компонентів;
- енергії;

- інформації.

Завжди існують сторонні по відношенню до цілі процесу фактори, якими не можливо управляти, але які впливають на процес. Процес виробництва полягає у створенні продукції із сировини з відповідними втратами енергії. Вхідною інформацією є технологічні інструкції які можна представити у вигляді наборів параметрів які можна контролювати. Вихідна інформація це набір даних що вимірюються і які описують стан процесу та його зміну.

Інформація основний компонент управління фізичними процесами. Обробка інформації дозволяє покращити характеристики технічного процесу і вигідна у будь якому випадку. Саме для її обробки і використовуються комп'ютери. Вони виконують дві основних функції: контролюють параметри технічного процесу та ініціюють відповідні сигнали для наступного керування щоб параметри процесу залишалися у заданих межах і при присутності зовнішніх збурень. Управління технічним процесом суттєво відрізняється від звичайній обробки даних. У даному випадку комп'ютерна система повинна швидко реагувати на зовнішні випадки і постійно обробляти зовнішню інформацію не маючи можливості змінити її кількість або швидкість надходження. Одночасно може виникнути необхідність виконання і інших операцій, наприклад вивід даних на дисплей або передачу даних на інший комп'ютер. Цей особливий режим обробки даних здобув спеціальну назву – режим реального часу.

Перший приклад застосування практичного використання комп'ютера для керування технологічним процесом відноситься до 1959 року коли було впроваджено керування нафтохімічного виробництва. Подальший розвиток цього напрямку призвів до створення нової галузі управління на основі комп'ютера.

В 1962 році була запропонована концепція прямого цифрового керування яка полягала у заміні аналогових контурів керування одним центральним комп'ютером.

Створення транзистора дало суттєвий розвиток комп'ютерним технологіям. Із знижкою вартості одиниць обчислювальної потужності стало доцільним впроваджувати комп'ютерне управління і до малих виробництв. Узгодження впливу трьох головних факторів – удосконалення технічної бази комп'ютерів, економічна доцільність їх використання в керуванні простими процесами та розвиток теорії управління – призвело до широкого розповсюдження комп'ютерного управління.

В наш час в управлінні використовують комп'ютери, що мають так звану відкриту архітектуру. В них основну вагомість мають не конкретні апаратні компоненти, а відповідні інтерфейси сполучення. Його у разі потреби можна підбирати спираючись на вимоги максимального використання ресурсів комп'ютера. У мікропроцесорних системах можна швидко змінювати стратегії керування шляхом заміни програми керування. Однак всі переваги керування за допомогою комп'ютера неможливі без чіткого усвідомлення загальних принципів роботи мікропроцесорних систем та їх окремих компонентів. Тільки чітке розуміння роботи окремих компонентів та їх взаємодія між собою дасть

можливість у повній мірі отримати переваги управління за допомогою комп'ютера.

1.2 Особливості цифрового управління технологічним процесами

Комп'ютери, що керують реальним процесами, виконують інші задачі ніж комп'ютери які використовують для звичайній обробки інформації. Основна різниця полягає у тому, що комп'ютер, що використовується для керування повинен робити із швидкістю що відповідає швидкості технологічного процесу (рис. 1.1).

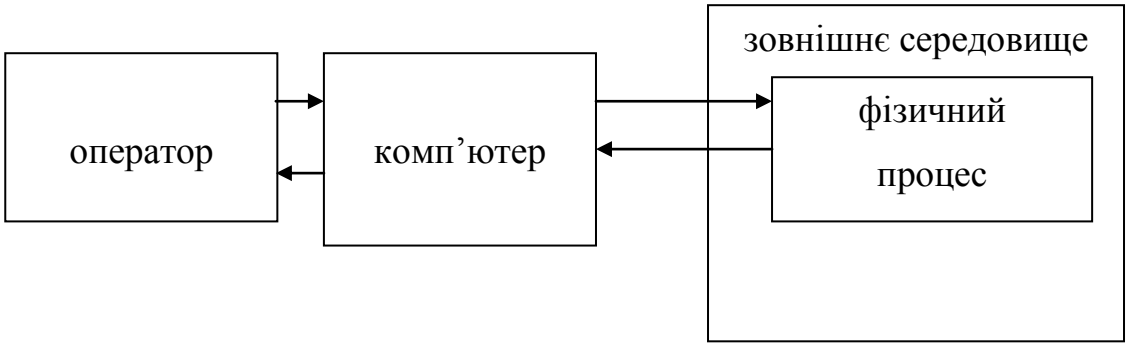

Рис.1.1 – Використання комп'ютера для керування процесом.

Інша головна особливість комп'ютерного управління полягає у тому, що хід виконання програми не можливо передбачити. Зовнішні сигнали можуть переривати або змінювати послідовність виконання дій в програмі. В той же час всі фізичні процеси можна явити у вигляді множини процесів які виконуються паралельно. Тому комп'ютер, що керує таким процесами повинен враховувати цю особливість. Тобто він повинен керувати паралельним процесами що враховувати у його структурі

Керування у режимі реального часу може бути реалізовано шляхом послідовного програмування при якому комп'ютер через певний час опитує відповідні датчики. Під час пауз він не може виконувати інші операції що знижує ефективність його роботи. Коли необхідно керувати ще одним процесом то виникає складність у перемиканні між даним процесами що призводить до ускладнення програми керування. На сучасний момент ця проблема розв'язується шляхом використанні відповідних переривань. Практично сигнал переривання являє собою зовнішній сигнал, що сповіщає о початку відповідної події. Ці події відслідковуються датчиками та примушують комп'ютер переходити від однієї програми до іншої.

У якості простого приклада розглянемо процес регулювання температури речовини у ємності. Температура речовини вимірюється датчиком вихідна напруга якого пропорційна дійсній температурі. Вимірювання відбуваються через певний час і поступають в комп'ютер, що керує даним процесом.

Величина нагріву розраховується за різницею між заданим та виміряним значеннями. В залежності від виконавчого механізму – пристрою, що впливає на процес – змінюється вид сигналу керування що подається на його вхід.

Регулятор температури демонструє основні властивості контуру керування. Температура повинна визначатися з частотою, що визначається сталою часу управління. Коли теплоємність ємкості велика, то стала часу має відносно велике значення. Навпаки, коли теплоємність ємкості мала, а потужність нагрівача велика то стала часу процесу мала і система керування повинна відносно швидко вимірювати температуру і керувати нагрівачем. Все це повинно враховуватися при виборі технічних засобів мікропроцесорної системи.

Доцільно відмітити деякі проблеми, що ускладнюють широке впровадження мікропроцесорної техніки у виробництво. Основним з проблем є:
– нелінійність процесу;
– навколишнє середовище, що може постійно змінюватися;
– зміна умов процесу;
– значні часові затримки.

Всі ці особливості притаманні більшості харчових процесів що ускладнює їх керування і вони повинні враховуватися при створення мікропроцесорних систем. Будь який технологічний процес потребує рішення двох проблем: як керувати процесом і як побудувати мікропроцесорну систему. Система керування повинна не тільки керувати технологічними операціями, а також виконувати ряд додаткових функцій, наприклад, визначати нештатні ситуації та оперативно на них реагувати. Тому знання принципів роботи цифрових пристроїв мікропроцесорних систем є запорукою ефективного їх використання. Знання принципів дії та застосування сучасних мікропроцесорних систем є невід'ємною складовою підготовки фахівців у галузі харчових виробництв.

1.3 Предмет «Мікропроцесорна техніка».

Предмет «Мікропроцесорна техніка» викладається студентам факультету обладнання та технічного сервісу і ставить своєю метою ознайомлення їх з сучасними підходами щодо розуміння роботи та наступного використання елементів мікропроцесорної техніки. Даний курс передбачає лекційні заняття та лабораторний практикум. На лекціях передбачається розглядання та обговорення основних теоретичних питань що пов'язані з етапам створення та використання мікропроцесорних систем з урахуванням їх майбутнього використання у харчових виробництвах. Речовини, що використовуються у харчових виробництвах у переважній більшості являють собою об'єкти, якими відносно складно керувати. Це обумовлено у переважній більшості їх біологічною природою. Для впевненого керування такими об'єктами необхідно мати відповідні датчик які можуть відтворювати без суттєвих похибок стан об'єкту керування. Крім того слід передбачити відносно великий

інформаційний потік для визначення рекомендацій щодо подальшого керування.

Лабораторний практикум проведення досліджень цифрових пристроїв з метою визначення особливостей їх функціонування шляхом застосування відповідних моделюючих програм. Такий підхід дає можливість суттєво поглибити розуміння щодо принципів дії шляхом використання відповідних віртуальних вимірювальних приладів.

В цілому даний курс сприяє розширенню розумінню щодо мікропроцесорної техніки, її використанні у сучасному виробництву і створює умови для розширення меж її використання з урахуванням сучасних тенденцій науково технічного прогресу.

ЛЕКЦІЯ 2. НАДАННЯ ІНФОРМАЦІЇ В МІКРОПРОЦЕСОРНИХ ПРИСТРОЯХ

2.1 Імпульсні струми та напруги.

Розглянуті у попередніх курсах аналогові пристрої що використовуються у техніці, мають ряд недоліків. Найбільш суттєві з них:

1. порівняно невисока точність що обумовлена дією зовнішніх факторів;.
2. для зміни алгоритму роботи необхідно змінити схему пристрою (тобто внести апаратні зміни).

Для подолання цих недоліків використовують принципово інший спосіб подачі та обробки інформації, Він полягає у тому, що інформація подається у вигляді набору імпульсів які чисельно відповідають набору значення інформації. При такому підході напруга сигналів у цифровому пристрої має лише два значення: близьке до нуля вольт Щ умовно називають *логічним нулем* та напругою у декілька вольт, що називають умовно *логічною одиницею.* Такий підхід дозволяє практично повністю уникнути впливу перешкод на корисний сигнал, а використання чисельних методів обробки інформації дозволяє досягти практично будь яку точність.

Крім того зміна алгоритму функціонування досягається програмним шляхом і не потребує зміни електричних схем.

Електричним імпульсом називають струм або напругу, що мають постійне значення протягом короткого часу. Прикладом такого імпульсу є відео імпульс який має постійну складову. Для оцінки імпульсів вводять наступні параметри:

амплітуда імпульсу U_m (максимальне його значення);

тривалість імпульсу t_i;

тривалість переднього фронту t_a (час зростання імпульсу від 0,1 U_m до 0,9 U_m);

тривалість заднього фронту $t_з$ (час спадання імпульсу від 0,9 U_m до 0,1 U_m);

час подавання імпульсів;

скважність $Q=T/ t_a$.

Для реалізації таких сигналів використовується звичайний транзисторний підсилювач який працює у ключовому режимі. На відміну від підсилювачів, що розглядалися раніше у цього підсилювача зсув робочої точки транзистора відносно нуля відсутній. Тому при $U_{вх}=0$ транзистор закритий. Струм колектора також дорівнює нулю а вихідна напруга практично дорівнює напрузі живлення даної схеми. Такий стан носить назву режим відсікання. Коли вхідна напруга перевищує напругу відкривання транзистора то по переходу колектор емітер починає іти струм і напруга на колекторі практично зменшується до нуля. Транзистор переходить у режим насичення тому що струм бази значно більший ніж той що потрібен для відкривання транзистора.

Перевага такої схем полягає у тому що такий підсилювач є найпростішим цифровим пристроєм на основі якого будуються усі інші цифрові схеми. Якщо вважати, що напруга на виході відповідає рівню логічної одиниці, то вихідні напруги схеми будуть відповідати необхідним для роботи інших схем рівнем напруги. Параметри схеми підсилювача обирають таким чином, що до нього можна підключати декілька інших входів таких саме підсилювачів. В цілому для кожного підсилювача є обмеження щодо кількості підключень.

Сучасні імпульсні пристрої які складаються з інверторів можуть мати ще один стан який позначається літерою “Z”. В даному стані вихід інвертора відключений від інших елементів які до нього підключені. Таким чином забезпечується паралельна робота багатьох інверторів на одне навантаження. Це суттєво спрощує схемну реалізацію імпульсних пристроїв, зменшує кількість елементів підвищує надійність роботи всього приладу. Функціонування інвертора з третім станом можна описати можна описати відповідною таблицею. Вона буде наведена при викладі матеріалу наступної лекції.

2.2 Основи двійкової системи обчислення.

Двійкова система числення являє собою теоретичну базу яка використовується в мікропроцесорній техніці. В цифрових пристроях сигнал має лише два значення – 0 і 1 (низький і високий рівень напруги, вмк./вимкн.), тому числа у таких пристроях можна подати лише у двійковій системі обчислення. В табл. 2.1. наведені десяткові та відповідні двійкові числа

Таблиця 2.1

Відповідність двійкових та десяткових чисел.

Десяткове число	Двійкове число			
	Розряди			
	2^3=8	2^2=4	2^1=2	2^0=1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

Двійкова система обрана тому, що електронні пристрої які мають два стана є самим надійними у роботі Крім того двійкова система підходить до обчислення логічних змінних які також мають тільки два стани. Якби електронний пристрій працював безпосередньо з десятковими числами то тоді він повинен бути знаходитися у 10 різних станах які відповідали цифрам від 0 до 9. Принципово це можливо, але пов’язано з відповідними технічними проблемами.

Двійкова система числення є позиційною системою числення і являє собою комбінацію двійкової системи кодування і показової вагової функції з основою рівною 2. Позитивні цілі числа записуються у виді

$$x_{2,2} = \sum_{k=0}^{n-1} a_k 2^k ,$$

де $x_{2,2}$ – двійкове число, перший індекс-основа системи кодування(розмірність множини цифр a=(0,1)), другий індекс-основа вагової показової функт]- n кількість знаків числа, k –порядковий номер цифри.

Переклад чисел з однієї системи у іншу виконується за відомим правилами арифметики.

Наприклад: $235 = 2 \cdot 10^2 + 3 \cdot 10^1 + 5 \cdot 10^0 = 200 + 30 + 3$

Аналогічно: $1011 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 1 \cdot 8 + 0 \cdot 4 + 1 \cdot 2 + 1 \cdot 1 = 11.$

2.3 Представлення чисел в обчислювальні машині

Представлення чисел в обчислювальні машині. В обчислювальних машинах традиційно використовують дві форми представлення чисел:

природна форма або форма з фіксованою зап’ятою;

нормальна форма, або форма з плаваючою зап’ятою.

Приклад (природна форма) $451{,}23=0{,}45123\cdot10^3$.

Для представлення чисел з фіксованою зап’ятою (природна форма) використовується наступне представлення числа:

Знак числа 0 – «+» 1 – «-»	Ціла частина числа							Дробова частини числа							

Зап’ята не відображається але її місце строго фіксовано. Лівий (старший) розряд призначений для зберігання інформації про знак числа і для запису числа не використовується. З фіксованою зап’ятою числа відображаються з постійним для всіх чисел положенням зап’ятої що відокремлює цілу частину від дробової. Така форма спрощує апаратну реалізацію ЕОМ, зменшує час виконання маминих операцій, однак при рішенні задач постійно треба слідкувати що всі дані знаходилися у межах діапазону представлення. Якщо цього не дотримуватися, то можливо переповнення розрядної сітки в результат обчислення буде невірний. Від цих недоліків вільні ЕОМ які використовують форму представлення чисел плаваючою зап’ятою.

При використанні нормальної форми числа відображаються наступним чином

$$X = \pm M \cdot P^{\pm r},$$

де M – мантиса числа (правильна дріб в межах $(0{,}1 \le M \le 1)$, r – порядок числа(ціле),P – основа системи числення.

Нормалізована експоненціальна запис числа має вигляд $a = m^{pg}$, де q – ціле число (позитивне негативне або нуль), m – дріб ціла частина якої має одну цифру. Величина m – мантиса числа, q – порядок числа.

Нормальна форма має великий діапазон чисел і є основною в сучасних ЕОМ. Нормальна форма представлення числа має наступний вигляд

Знак Мантиси								Знак порядку							

Поле мантиси Поле порядку

У конкретній ЕОМ межа представлення чисел з плаваючою зап’ятою залежить від основи системи числення та кількості розрядів для представлення порядку.

2.4 Дії над двійковими числами..

Дії над двійковими числами. Арифметичні операції над числами виконуються подібно діям з десятковими починаючи з молодшого розряду. Всі арифметичні операції виконуються на використанні наступних таблиць

Таблиця 2.2.

Таблиця операцій

Додавання			Віднімання			Множення		
+	0	1	-	0	1	*	0	1
0	0	1	0	0	¯11	0	0	0
1	1	10	1	1	0	1	0	1

Розглянемо приклади виконання арифметичних операцій на прикладі дій х цілими числами.

Складання двійкових чисел. Ця операція є непоширеною, що виконується в мікропроцесорних системах. Операція складання виконується в стовпчик для чого числа записують одне під іншим. Складання починають справа наліво. Наприклад:

01011 =13 В даному випадку було складено 11 та 10.
01010 =10 Результатом є число 21.
10101 =21

Віднімання двійкових чисел.

У мікропроцесорних системах немає від’ємних у звичайному розумінні чисел. Там є тільки рівні 0 та 1 (логічний 0 та логічна одиниця). Однак математика дозволяє представляти від’ємні числа без використання негативного знака. Негативні числа у мікропроцесорних системах можна представити у зворотному або додатковому коді. Для представлення числа у зворотному коді необхідно інвертувати всі його розряди і далі провести операцію додавання.

Принцип такого представлення полягає у тому, що негативне число це таке число яке необхідно додати до позитивного числа тієї величини щоб здобути нуль. Щоб не використовувати негативний знак необхідно спочатку

визначити весь діапазон чисел з якими необхідно робити. Таким чином представлення негативні числа у мікропроцесорних системах у особливому виді який має назву додатковий код. Для переводу числа у додатковий код необхідно зробити наступні дії:

представити число за допомогою наборів 0 та 1;

інвертувати кожний розряд числа (замінити 0 на 1 та 1 на 0);

додати 1 до молодшого розряду.

Для того щоб цифровий пристрій знав з яким числами (позитивними чи негативними) треба виконувати арифметичні операції перед числом вводять додатковий розряд в який заносять 0 коли число позитивне, або – 1, коли число негативне. Наприклад число 0.1001 дорівнює числу 9, а 1.0101 дорівнює числу -5. Таким чином комп'ютер розуміє які алгоритми обробки інформації треба використовувати.

Розглянемо приклади.

1. 0.0110 =6 переходимо до додаткового коду 0.0110
   1.0011 =-3 - 1.1101
   результат обчислення *1*. 0.0011

В результаті складання виникає одиниця переносу *1* яка не враховується. Результат складання є число позитивне (знаковий розряд дорівнює 0). Позитивне число у звичайному та додатковому коді однакові тому результат дорівнює 3.

2. 1.0110=-6 переходимо до додаткового коду 1.1010
   1.0010=-2 1.1110
   результат обчислення *1*.1.1000

Результат обчислення визначається додатковим кодом. Для переходу у звичайний код необхідно про інвертувати знайдене число і додати до молодшого розряду одиницю. У даному випадку число також буде дорівнювати 1.1000.

2. 0.00100=4 переходимо до додаткового коду 0.00100
   1.10011=-19 1.01101
   результат обчислення *1*.1.10001.

Результат обчислення даного прикладу також визначається додатковим кодом. Для переходу у звичайний код необхідно про інвертувати знайдене число і додати до молодшого розряду одиницю. У даному випадку число також буде дорівнювати 1.0111=-15. Використання додаткового коду спрощує виконання арифметичних операцій. Якби машина рокотала з прямими кодами позитивних та негативних чисел то для виконання арифметичних операцій необхідно було виконати ряд додаткових дій. При використанні додаткового коду то операція додавання зводиться до порозрядного додавання. Для комп'ютерного

представлення цілих чисел використовується один, два або чотири байти, тобто комірка пам’яті буде мати вісім, шістнадцять або тридцять два розряди відповідно.

Таким чином для реалізації операції віднімання в мікропроцесорних системах реалізують операцію додавання.

Операція множення двійкових чисел полягає у визначенні знака та абсолютної величини числа. Для визначення знаку добутку необхідно просумувати знакові розряди співмножників без формування переносу (так зване сумування по модулю 2). Для знаходження абсолютного значення добутку необхідно провести звичайне порозрядне множення. Розглянемо приклад. Необхідно помножити два числа 1101_2 та 1011_2

```
 1101  1 - співмножник
x
- 1011  2 - - співмножник
 ———
  1101  1-ий частковий добуток
 1101   2-ий частковий добуток
0000    3-ий частковий добуток
1101    4-ий частковий добуток
 ————————
10001111  добуток
```

З наведеного прикладу видно, що здійснюється шляхом сумування з одночасними зсувами чисел. В цифрових пристроях процес сумування часткових добутків має послідовний характер: формується один з часткових добутків, до нього з відповідним зсувом додається наступний частковий добуток , до знайденої суми двох часткових добутків додається з відповідним зсувом наступний частковий добуток і т.д., доки не прочумуються всі часткові добутки. Існують і інші алгоритми реалізації операції добутку.

З наведено прикладу видно, що основна операція при знаходженні добутку це є операція складання. Ділення двійкових чисел також можна звести до відповідної операції складання з використанням негативних чисел.

2.5 Шістнадцятирічна та восьмерична системи числення у мікропроцесорних системах.

Сучасна мікропроцесорна техніка використовує лише двійкову систему числення. Однак чоловіку важко сприймати довгі записи нулів та одиниць. Крім того переводити числа з двійкової системи у десяткову відносно важко і тому фахівці використовують інші системи числення (восьмирічну та шістнадцятирічну). Особливо широко при виводі інформації з мікропроцесорних систем використовується шістнадцятирічна система числення. При її використанні записи чисел більш компактні. В даній системі

основа дорівнює 16 тобто використовують 16 символів для запису чисел. Це наступні символи: цифри від 0 до 9 а далі букви латинського алфавіту від A до F. Кожний символ шістнадцятирічного числа потребує 4 разрядів двійкового числа для запису символів. При програмування мікропроцесорних систем також широко використовують восьмеричну систему числення. В таблиці 2.3 наведені наведено співвідношення між числами двійкової, восьмирічної, десяткової та шістнадцятирічної систем числення

Таблиця 2.3

Таблиця відповідності чисел різних систем числення

10-ая	8-ая	2-ая	16-ая	10-ая	8-ая	2-ая	16-ая
0	0	00000000	0	8	10	00001000	8
1	1	00000001	1	9	11	00001001	9
2	2	00000010	02	10	12	00001010	A
3	3	00000011	3	11	13	00001011	B
4	4	00000100	4	12	14	00001100	C
5	5	00000101	5	1	15	00001101	D
6	6	00000110	6	14	16	00001110	E
7	7	00000111	7	15	17	00001111	F

Для переводу чисел з двійкової системи в 16 –ую систему необхідно число розбити на тетради (по 4 розряди) і записати кожну тетраду еквівалентним двійковим числом. Наприклад:

$1001\ 1110_2=9E_{16}$; 2. $0010\ 0010_2 =22_{16}$.

Для переводу числа з 16 –ої системи у 2 необхідно кожну цифру 16 –го числа записати відповідним розрядами двійкового коду. Наприклад:

1. $164_{16} =101100100_2$.

Для переводу двійкового числа у восьмирічну систему його треба розбити на трійки і кожну з них замінити на відповідне восьмирічне число. Ці операції необхідно робити при розробці програм роботи мікропроцесорних систем.

ЛЕКЦІЯ 3. ОСНОВНІ ПОНЯТТЯ АЛГЕБРИ ЛОГІКИ, ЛОГІЧНІ ОПЕРАЦІЇ ТА ЇХ РЕАЛІЗАЦІЯ.

3.1 Алгебра логіки.

Алгебра логіки це розділ математичної логіки в якому вивчаються логічні операції на висловленнями. Передбачається, що висловлення можуть бути тільки істинними або хибними. Основним елементам з яким оперує алгебра логіки є висловлення яка і є логічною змінною. Вона може приймати два значення 0 або 1. Існують три основних логічних операції. Це логічне заперечення (операція НІ), логічне множення (операція І), логічне додавання (операція АБА). Крім того алгебра логіки визначається наступною системою аксіом:

$$\begin{cases} x = 0, \quad коли\ x \neq 1 \\ x = 1, \quad коли\ x \neq 0 \end{cases} \quad \begin{cases} \overline{0} = 1 \\ \overline{1} = 0 \end{cases} \quad 0+1=1+0=1$$

$$\begin{cases} 1+1=1 \\ 0+0=0 \end{cases} \quad \begin{cases} 0 \cdot 0 = 0 \\ 1 \cdot 1 = 1 \end{cases} \quad 1 \cdot 0 = 0 \cdot 1 = 0$$

3.2 Основні тотожності логічних обчислень.

$$X+X=X, \quad (X \cdot X)=X \quad , \; X \cdot \overline{X} = 0, \quad X + \overline{X} = 1, \quad \overline{\overline{X}} = X.$$

При проведенні еквівалентних перетворень з метою спрощення логічних виразів використовується теорема де Моргана:

$$\overline{X_1 \cdot X_2} = \overline{X}_1 + \overline{X}_2; \quad \overline{X_1 + X_2} = \overline{X}_1 \cdot \overline{X}_2; \quad X + \overline{X} \cdot Y = X + Y.$$

Операції з 0 та 1: $X \cdot 1 = X, \; X \cdot 0 = 0, \; X + 0 = X, \; X + 1 = 1$

Розглянуті вище три основних логічних операції аналізувати та спрощувати будь які логічні вирази. Для їх реалізації необхідно розглянути відповідні схеми які дають можливість реалізувати ці операції Ці операції виконуються на відповідних електронних пристроях в основі багатьох з них лежить транзистор, що працює у ключовому режимі з додаванням деяких функціональних елементів. Розглянемо реалізацію логічних операцій з використанням електронних схем.

3.3 Реалізація логічних виразів за допомогою цифрових схем.

Бурхливий розвиток обчислювальної техніки, широке її впровадження у всі сфери діяльності людини потребує знайомства фахівців будь – яких галузей з основними її елементами та принципами роботи окремих приладів. Розглянемо реалізацію основних логічних операцій на стандартних елементах комп'ютерної техніки.

Всі сучасні ЕОМ оперують цифровою інформацією, що представлена у двійковій системі. Це означає, що числа які обробляє комп'ютер, можна представити за допомогою лише двох цифр: 0 та 1. Двійкова система вибрана тому, що електроні пристрої які її реалізують, повинні мати тільки два стійких стана що забезпечує високу надійність роботи в умовах промислових перешкод. Крім того, двійкова система ідеально підходить для обчислення функцій логічних змінних тому що останні можуть приймати тільки два значення „істинне" та „хибне". З логічних змінних можна скласти логічні повідомлення істинність або хибність яких можна визначити однозначно. Кожне логічне повідомлення може бути представлено математичним еквівалентом, *логічною функцією.* Логічні повідомлення об'єднують між собою за допомогою логічних операцій.

Розглянемо три основних логічні операції.

1. *Операція НІ* (логічне заперечення або інверсія). Логічне заперечення функції *Y* позначається як $Y=\bar{x}$ і визначається таблицею істинності:

X	*Y*
0	*1* .
1	*0*

Робота даного елемента пояснюється діаграмою сигналів, що наведені на рис. 3.1

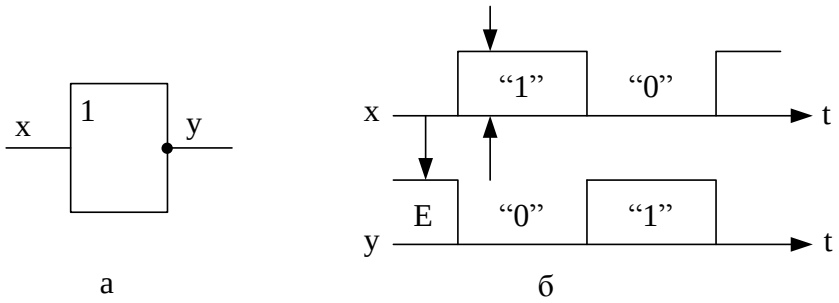

Рис. 3.1 – Принцип роботи інвертора

У якості такого елемента використовують звичайний підсилювач на транзисторі. При надходженні на його вхід сигналу транзистор відкривається і

на його виникає низький потенціал. Тобто, коли вхідний сигнал дорівнює „одиниці” вихідний сигнал елемента буде дорівнювати „нулю”. І навпаки, коли вхідний сигнал дорівнює „нулю” вихідний сигнал дорівнює „одиниці”. Таким чином даний елемент виконує інверсію сигналу, тобто перевертає його за фазою на 180 градусів Тут X – вхідний, а Y вихідний сигнал електронної схеми, яка реалізує операцію НІ. Таку операцію може виконувати кнопка з нормально замкненими контактами. Якщо вона не натиснута то коло замкнене і навпаки. Цю саму операцію виконує ключовий транзисторний підсилювач. Деякі логічні елементи мають вбудовані інвертори.

Якщо використовується інвертор з додатковим станом «Z», то таблиця істинності має вигляд

X	Z	Y
0	0	
1	0	
0	1	
1	1	

Використання додаткового стану дає можливість паралельного підключення цифрових приладів до інформаційних шин без використання логічних ключів.

2. *Операція АБО* (логічне додавання, або диз’юнкція) позначається символами „+” або „v”. Кількість аргументів може бути два, три і більше. При наявності двох незалежних аргументів X_1 та X_2 записується як $Y=X_1 + X_2$. Реалізована функція буде дорівнювати одиниці коли хоча би одна з незалежних величин дорівнює одиниці. Даний логічний елемент може бути реалізований за допомогою відповідних електронних схем. Робота даного елемента визначається за допомогою таблиці істинності:

X_1	X_2	Y
0	0	0
0	1	1 .
1	0	1
1	1	1

Умовне позначення на схемах та часова діаграма функціонування логічного елемента, що реалізує операцію АБО, наведені на рис. 3.2. Сигнал на виході цього елемента з’являється одночасно з появою будь – якого вхідного сигналу. Електрична схема, що реалізує цю операцію, може бути виконана і на напівпровідникових діодах. Даний елемент можна розглядати як дві паралельно з’єднані кнопки керування і тому прилад, який до них підключений, буде ввімкнено при вмиканні однієї з них Сучасна комп’ютерна техніка реалізує ці

операції на базі ІМС з використанням інверторів. Часто ця операція використовується з подальшою інверсією вихідної змінної тобто до виходу даного елемента підключають інвертор. Отримана в результаті функція АБО-НІ реалізується одним спеціальним елементом і має іншу таблицю істинності. Більш детальну інформацію про схемотехнічні рішення реалізації даної операції та її використання у сучасних комп'ютерних схемах можна отримати у спеціальній літературі.

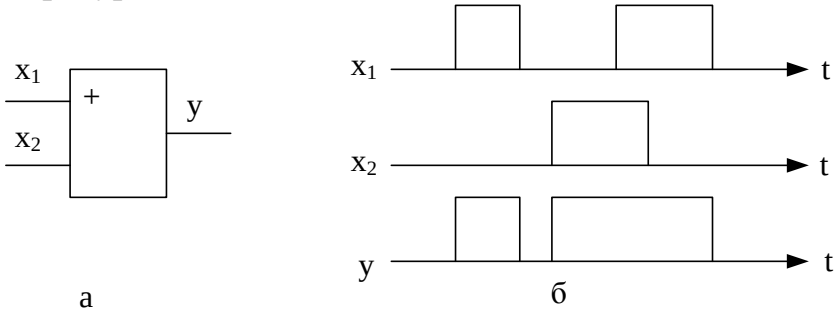

Рис. 3.2. – Принцип роботи елемента АБО

Кількість входів елементів може бути і більше двох і тоді він реалізує функцію виду $Y = X_1 + X_2 + X_3 + \ldots$ Кількість входів прийнято вказувати в позначенні елемента, наприклад: чотиривходовий елемент – 4 АБО-НІ.

3. *Операція І* (логічне множення, або кон'юнкція). Для її позначення використовуються символи „&", „х". Записується ця операція як $Y=X_1 \cdot X_2$. і визначається таблицею істинності:

X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Дану функцію можна реалізувати як послідовне з єднання двох кнопок з нормально розімкненими контактами. Якщо їх ввімкнути послідовно з електричною лампою, то вона буде горіти коли включені дві кнопки. Таким чином, логічна одиниця виникає на виході елемента лише тоді коли на всіх ходах присутні одиничні сигнали.

Умовне позначення на схемах та часова діаграма функціонування логічного елемента, що реалізує операцію І, наведені на рис. 3.3.

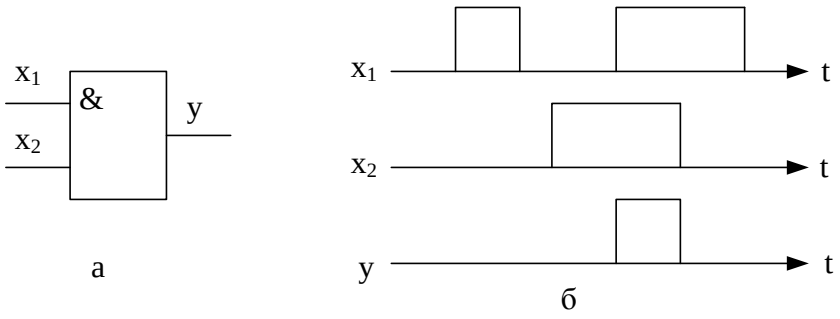

Рис. 3.3 – Принцип роботи елемента І

Для того щоб вихідна функція даного елемента була істинною (дорівнювала значенню „одиниця”) необхідно щоб істинними були всі вхідні сигнали.

4. Операція виключне АБО

Ця операція може бути визначена як нерівнозначність і широко використовується у сучасних мікропроцесорних системах. Сигнал на її виході приймає значення 1 лише тоді, коли сигнали на входах відрізняються. Такій логічній ситуації відповідає вираз

$$Y = (\overline{X}_1 \cdot X_2) + (X_1 \cdot \overline{X}_2).$$

Відповідний логічний елемент має два входи та один вихід і таку функцію використовують для порівняння двох змінних за значеннями. Таблиця істинності для даної функції має вигляд

X_1	X_2	Y
0	0	0
1	0	1
0	1	1
1	1	0

Практичне застосування алгебри логіки полягає у тому, що вона дає можливість реалізувати заданий порядок роботи цифрових приладів. Використання розглянутих законів дає можливість не тільки провести синтез цифрових приладів але і провести їх реалізацію з використанням мінімальної кількості цифрових елементів. Розглянемо приклади реалізації цифрових приладі на основі використання алгебри логіки.

Припустимо, що необхідно синтезувати елемент НІ з елемента 2І-НІ. Даний елемент має рівняння $Y = \overline{X_1 \cdot X_2}$. Згідно теореми де Моргана можна

записати $Y = \overline{X}_1 + \overline{X}_2$. Якщо об'єднати входи між собою то можна записати $Y = \overline{X} + \overline{X} = \overline{X}$. Таким чином ми отримали нову функцію.

Коли необхідно збільшити кількість входів логічного елемента та можна не вистачає то їх можна збільшити. Наприклад, потрібен тривходовий елемент (3І-НІ) який потрібно створити на базі двоходового («І-НІ). Очевидно, що $\overline{\overline{x}} = x$ і можна записати $y = \overline{x_1 \cdot x_2 \cdot x_3} = \overline{\overline{x_1 \cdot x_2} \cdot x_3} = \overline{\overline{x_1 \cdot x_2} \cdot 1 \cdot x_3}$. В даний логічний запис було введено 1 яка з наведених вище тотожностей не може змінити логічну функцію. Такий підхід до синтезу логічних схем оснований на застосуванні логічних перетворень за вище наведеними правилами. Електрична схема, що відповідає даним рівнянням, приведена на рис 3.4. і вона еквівалентна схемі (3І-НІ)

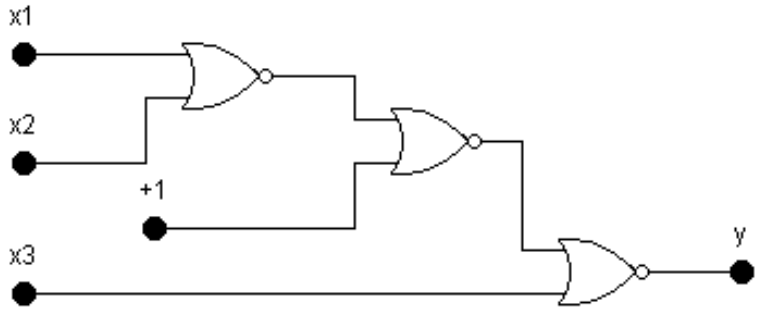

Рис. 3.4 – Синтезована електрична схема .

Логічні елементи, що реалізують логічні операції НІ, АБО, І створюють на дискретних напівпровідникових приладах або ІМС. У сучасних комп'ютерних системах логічні операції реалізуються виключно на ІМС.

Рівняння логічних операцій розв'язуються згідно з правилами алгебри логіки, яка є аналогом звичайної алгебри. Особливістю даної алгебри є те, що аргументи і функції можуть приймати лише два значення:0 та 1. Алгебра логіки дозволяє математично описати будь – яке функціонування технічного пристрою та виконати оптимізацію його роботи.

В алгебрі логіки існує наступний порядок виконання дій: спочатку виконується операція НІ, потім І і наприкінці АБО. Для зміни порядку операції використовують дужки. Для спрощення логічних виразів використовують ряд тотожностей які дають можливість спростити логічні рівняння, звести до мінімуму кількість логічних елементів що необхідні для виконання логічних функцій. При аналізі складних логічних виразів доцільно використовувати розроблені для цих цілей спеціальні діаграми, наприклад діаграму Вейча.

Таким чином розглянути операції алгебри логіки дозволяють будувати різноманітні логічні схеми які можуть описати необхідну послідовність роботи окремого приладу. Щодо реалізації логічними елементами то вона може бути здійснена різними логічними системами.

На основі логічних виразів створюється комбінаційні цифрові схеми які дають можливість при виконанні певних умов змінювати алгоритм роботи як окремого приладу, так і системи керування в цілому. Комбінаційні схеми мають велику швидкодію і дозволяють оперативно реагувати на певні сигнали що створюються під час роботи цифрових пристроїв.

Більш детальну інформацію про використання у сучасних комп'ютерних схемах розглянутих елементів можна отримати у спеціальній літературі.

ЛЕКЦІЯ 4. КОМБІНАЦІЙНІ ТА ПОСЛІДОВНІ СХЕМИ.

4.1 Комбінаційні схеми мікропроцесорної техніки.

Комбінаційні схеми призначені для виділенні з інформаційного потоку певних числових комбінацій при надходженні яких цифровий пристрій повинен змінити провести якусь дію або змінити режим роботи. Ця потреба особливо виникає при роботі мікропроцесорних систем в режимі реального часу коли послідовність операцій залежить не тільки від програми роботи приладу а і від стану навколишнього середовища..

Дешифратор є типовим комбінаційним пристроєм. Він перетворює кодовані двійкові вхідні сигнали у сигнали керування виконавчими пристроями, пристроями відображення інформації т.п.

У загальному випадку дешифратор має декілька входів (виходячи з розрядності двійкових чисел, що треба декодувати.). Кожній комбінації вхідних сигналів відповідає може відповідати комбінація вихідних сигналів. Розглянемо двійковий дешифратор. Таблиця істинності має вигляд

$x1$	$x2$	$y1$	$y2$	$y3$	4
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Для кожного вихідного сигналу запишемо відповідні умови його виникнення:

$$y1 = \overline{x1} \cdot \overline{x2}; \quad y2 = \overline{x1} \cdot x2; \quad y3 = x1 \cdot \overline{x2}; \quad y4 = x1 \cdot x2 .$$

Якщо реалізувати знайдені рівняння логічними елементами то можна створити прилад який буде керувати іншими приладами в залежності від комбінації поданих вхідних сигналів. Розглянемо таблицю істинності для

дешифратора який перетворює триразрядний двійковий код у відповідні вихідні сигнали. Таблиця істинності має вигляд що наведена у таблиці 4.1. З даної таблиці видно що треба знайти вісім логічних рівнянь. Кожне рівняння складається з трьох двійкових розрядів. Для кожного вихідного розряду можна записати логічне рівняння. Наприклад:

$$Q0 = \bar{x}1 \cdot \bar{x}2 \cdot \bar{x}3; \quad Q6 = x1 \cdot x2 \cdot \bar{x}1.$$

Таблиця 4.1

Таблиця істинності дешифратора 3х8.

Входи			Виходи							
x3	x2	x1	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0		1	0	0	0
1	0	1	0	0	0		0	1	0	0
1	1	0	0	0	0		0	0	1	0
1	1	1	0	0	0		0	0	0	1

Дешифратори у складі логічних елементів набувають деяких особливостей: інверсні виходи (замість одиниці на обраному виході є нуль, а на інших – одиниці), наявність спеціальних додаткових входів, що дозволяють роботу схеми т.д. Дешифратори використовують у схемах індикації, у складі складних обчислювальних систем для обирання (адресації) конкретного елемента.

Мультиплексом – це електронний перемикач який має n адресних входи, 2^n інформаційних входи і один вихід. На вихід мультиплексора надходить лише один сигнал з того входу номер якого збігається з двійковим числом на адресних входах. За допомогою мультиплексора можна реалізовувати різні логічні функції. Розглянемо, наприклад, мультиплексор 8х1. Він має 8 інформаційних сигналів, 3 розряди для керування, вихідний сигнал і вхід для дозволу роботу мультиплексора.Функціонування мультиплексора описується наступними рівняннями

$$Y = \left(\begin{array}{l} \overline{C} \cdot \overline{B} \cdot \overline{A} \cdot D0 + \overline{C} \cdot \overline{B} \cdot A \cdot D1 + \overline{C} \cdot B \cdot \overline{A} \cdot D2 + \overline{C} \cdot B \cdot A \cdot D3 + \\ + C \cdot \overline{B} \cdot \overline{A} \cdot D4 + C \cdot \overline{B} \cdot A \cdot D5 + C \cdot B \cdot \overline{A} \cdot D6 + C \cdot B \cdot A \cdot D7 \end{array} \right) \cdot \overline{G}$$

На мультиплексові можна реалізувати логічні функції однак попередньо треба визначити які сигнали та логічні константи треба подавати на вхід мультиплексора. Припустимо, що треба реалізувати функцію

$$F = C \cdot \overline{B} \cdot \overline{A} + \overline{C} \cdot B \cdot \overline{A} + C \cdot B \cdot A + A \cdot B \cdot \overline{C}\ .$$

Ця функція визначена для 8 комбінацій тому буде використаний мультиплексом 8x1. Побудуємо для нього таблицю істинності що наведена нижче. З неї видно, що для реалізації функції на інформаційний вхід з номером N слід подати сигнал, значення якого дорівнює відповідному значенню функції F. Це означає, що на входи з номерами 1,2,5,6 слід подати рівень логічного нуля, а на інші рівень логічної одиниці. Таким чином при подачі комбінацій логічних рівнів на управляючі входи мультиплексора до його виходу підключиться вхід, значення сигналу на якому буде відповідати значенню функції.

Таблиця 4.2.

Таблиця істинності мультиплексора 8x1.

N	C	B	A	F
0	0	0	0	0
1	0	0	1	
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Схемна реалізація наведена на рис 4.1.

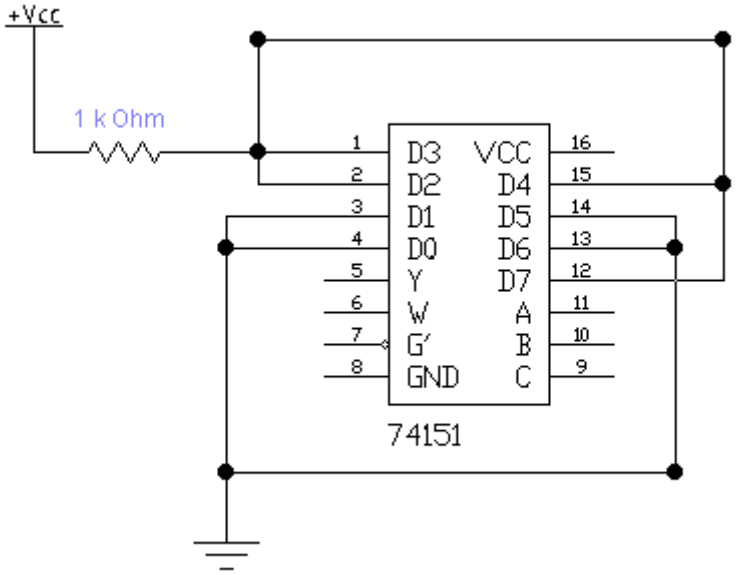

Рис.4.1 – Схема підключення мультиплексора.

Більш детальну інформацію щодо використання мультиплексора для реалізації логічних функцій можна довідатися к відповідній літературі.

Для реалізації операції складання двійкових чисел використовується суматор. Він є типовою комбінаційною схемою. Для його реалізації необхідно скласти таблицю істинності де записати вимоги до значення суми S і переносу P. Одноразрядний сумматор може складати два розряди за наступною таблицею істинності

$x1$	$x2$	S	P
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Згідно таблиці запишемо рівняння для S та P

$$S = \bar{x}1 \cdot x2 + x1 \cdot \bar{x}2; \quad P = x1 \cdot x2 .$$

Схема одноразрядного суматора, що побудований за знайденим рівнянням наведено на рис 4.2.

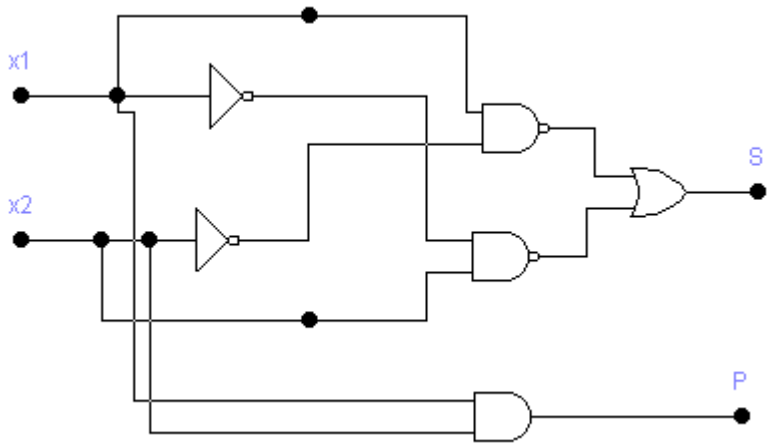

Рис. 4.2 – Схема одноразрядного суматора.

4.2 Послідовні схеми мікропроцесорної техніки.

Розглянуті схеми комбінаційних приладів дозволяють з інформаційного потоку виділяти певні комбінації при надходженні яких повинні спрацьовувати ряд приладів. В той же час існують технологічні процеси хід яких залежить не тільки від програми керування , а також від умов які виникають у процесі роботи. Тому у складі мікропроцесорних систем є прилади які дозволяють

запам’ятовувати та зберігати інформацію на певний час. Для зміни інформації необхідно подати зовнішній сигнал. Такі прилади носять назву *тригери.* Він має один або декілька входів і два виходи – прямий Q та інверсний $\overline{Q}$. Коли на першому виході сигнал буде відповідати „1”, то на другому виході обов’язково буде „0”.

Розглянемо декілька найбільш поширених типів тригерів які є основою сучасної обчислювальної техніки.

Найпростіший тригер – це асинхронний RS – тригер. Він має два входи S (set – встановити) і R (reset – знову встановити, тобто скинути) і два виходи Q та $\overline{Q}$, Позначення RS – тригера наведено на рис.4.3.

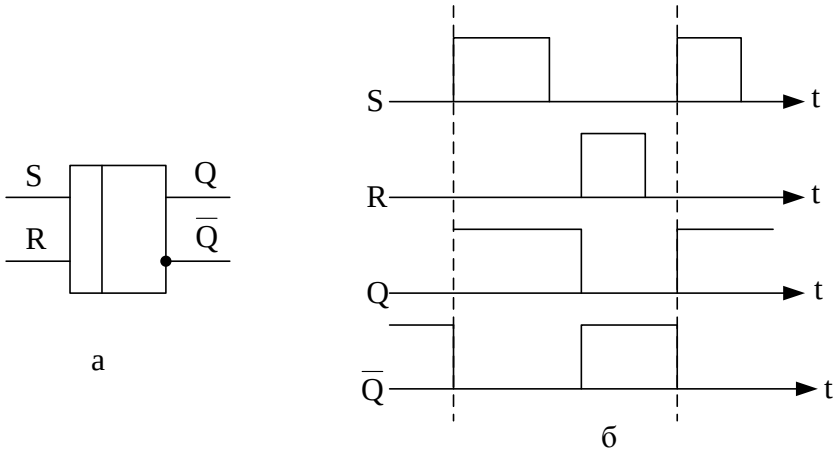

Рис. 4.3 – Принцип роботи RS тригера.

Для розуміння роботи тригера позначимо Q_n – вихідний сигнал тригера до поступлення вхідного сигналу, а літерою Q_{n+1} – вихідний сигнал після дії вхідного сигналу. Робота тригера може бути описана наступною таблицею істинності де позначення:

S	R	$\overline{S}$	$\overline{R}$	Q_n	$\overline{Q}_n$	Q_{n+1}	$\overline{Q}_{n+1}$
0	0	1	1	1	0	1	0
0	1	1	0	1	0	0	1
0	0	1	1	0	1	0	1
1	0	0	1	0	1	1	0
1	1	0	0	1	0	*З.К*	*З.К*

На практиці доцільно використовувати таблицю істинності у скороченому вигляді:

S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	*З.К*

Позначення З.К означає заборонену комбінацію.

Часова діаграма роботи тригера, що пояснює його функціонування наведена на рис. 4.3. З діаграми видно, що тригер змінює свій стан тільки при надходженні сигналів на входи R та S.

Лічильний Т – тригер має один тактовий вхід (Т) та два виходи Q та $\overline{Q}$. У момент приходу тактового імпульсу стан тригера змінюється на протилежний. Роботу даного тригера можна описати наступною таблицею істинності:

T	Q_n
0	Q_n
1	Q_{n+1}
0	Q_{n+1}
1	Q_n

Умовне позначення на схемах та часова діаграма дії Т – тригера наведено на рис. 4.4.

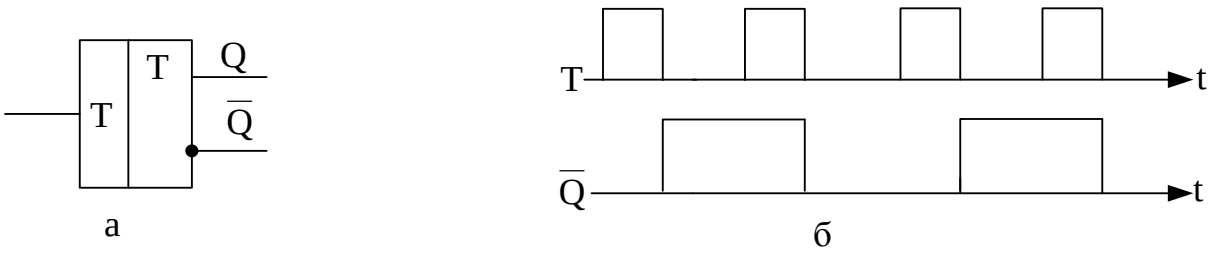

Рис. 4.4 – Принцип роботи Т тригера.

На основі цих тригерів будують лічильники імпульсів.

D – тригер. Цей тригер має тактовий інформаційний вхід D та синхронізуючий вхід С. Зміна стану цього тригера відбувається при появі сигналу на вході С при умові що на вході D присутній сигнал. Роботу даного тригера можна описати наступною таблицею істинності:

C	D	Q_n
0	1	Q_n
1	1	Q_{n+1}.
0	0	Q_{n+1}
1	0	Q_n

Умовне позначення на схемах та часова діаграма роботи D тригера наведена на рис.4.5.

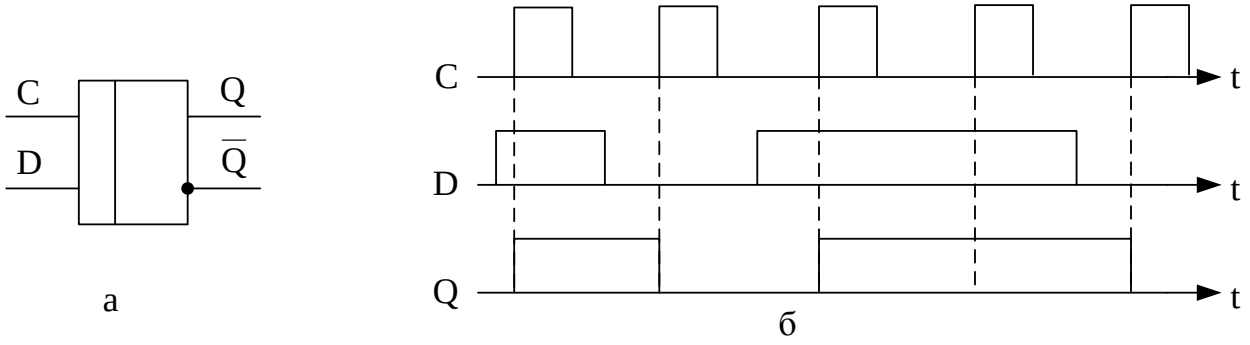

Рис. 4.5 – Принцип роботи D тригера.

З таблиці видно, що тригер запом’ятовує сигнал що присутній на вході D і запямятовує його при приході синхроімпульсу (С). Сигнал на виході тригера зміниться лише тоді коли, зміниться інформація на вході D і прийде наступний синхроімпульс.

Лічильники імпульсів

Лічильники імпульсів призначені для лічби кількості імпульсів, що поступають на їх вхід. Результат лічби фіксується в них в двійковому коді. Лічильник може бути побудований шляхом послідовного з’єднання тригерів. На рис. 4.6 показано приклад побудови триразрядного лічильника на основі Т – тригера.

При надходженні імпульсів на вхід першого тригера стан його починає змінюватись і викликає зміну подальших тригерів. Послідовність зміни стану тригерів під час лічби вхідних імпульсів наведена у табл.7.1. З таблиці видно, що стан розрядів X0,X1,X2 лічильника імпульсів у будь – який час являє собою запис кількості імпульсів що надійшли до лічильника у вигляді двійкового коду. Після запису числа $N=2^3-1=7$ лічильник переходить у нульовий стан.

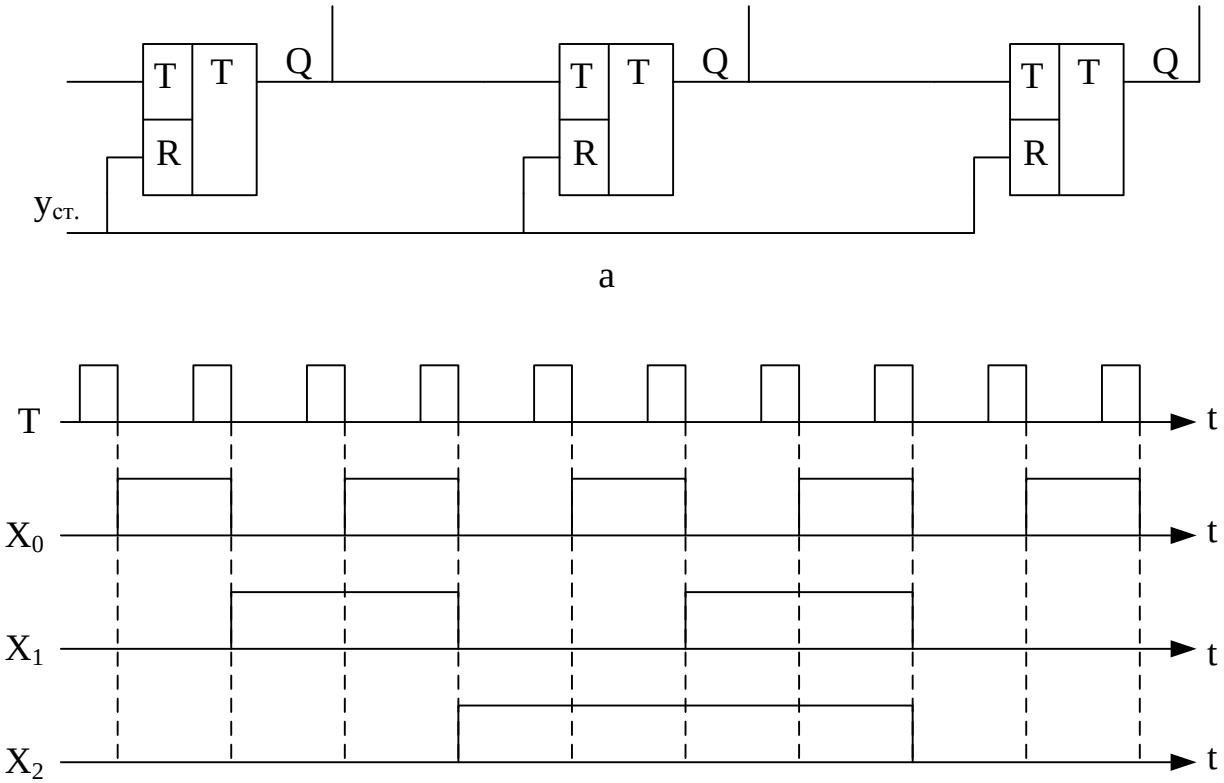

Рис.4.6 – Лічильник імпульсів на Т тригері

Таблиця 4.3.

Таблиця станів тригерів лічильника імпульсів

№ імпульсу	X2	X1	X0
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6		1	0
7	1	1	1
8	0	0	0

Лічильники імпульсів можна побудувати також на основі D тригера. Приклад побудови лічильника наведено на рис. 7.28.

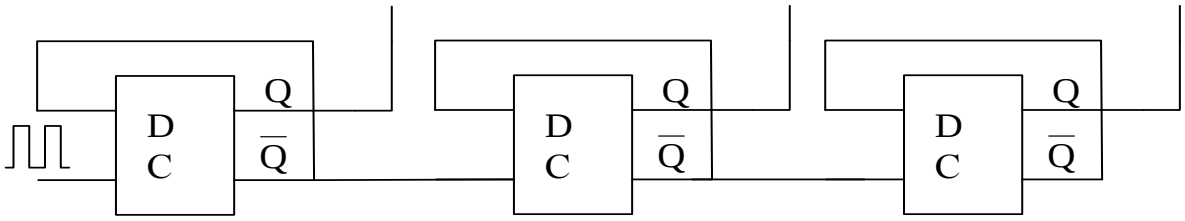

Рис.4.7 – Лічильник на основі D тригера

Регістри призначені для запам’ятовування інформації і наступному її зберіганні у вигляді багато розрядних двійкових чисел та видачі їх за зовнішньою командою. Їх відносять до елементів короткочасної (оперативної пам’яті.).

Регістри будуються на основі тригерів і від способу запису та видачі інформації бувають:

1) послідовні – запис інформації виконується послідовно одного двійкового розряду за іншим через один вхід;

2) паралельні – запис інформації виконується одночасно (паралельно) у всі розряди;

3) послідовно-паралельні – можуть працювати як послідовні або паралельні, залежно від сигналу на спеціальному вході керування.

Крім регістрів зберігання існують регістри зсуву які дозволяють за приходом імпульсу пересувати інформацію вліво або вправо. Регістри зсуву можуть використовуватися у схемах множення або ділення чисел. Регістри можуть видавати інформацію у прямому або зворотному коді знаходять широке застосування у мікропроцесорних системах.

ЛЕКЦІЯ 5. ОБРОБКА ІНФОРМАЦІЇ В МІКРОПРОЦЕСОРНИХ СИСТЕМАХ

5.1 Цифро – аналогові та аналого – цифрові перетворювачі.

При використанні комп'ютера для керування технічними об'єктами, проведення наукових досліджень тощо, необхідно проводити перетворення інформації. Використанні комп'ютера для керування промисловими об'єктами потребує перетворення цифрової інформації, з якою оперує комп'ютер, у аналоговий сигнал, який здійснює керування виконавчим (у більшості випадків) електричним пристроєм. При проведенні досліджень навпаки необхідно перетворювати сигнал вимірювання, який за своєю природою є у більшості випадків аналоговим, у цифровий код який може обробляти комп'ютер. Для виконання таких задач використовуються цифро – аналогові та аналого – цифрові перетворювачі, які виконані у вигляді ІМС. Ці перетворювачі узгоджують інформацію, з якою працює комп'ютер, з інформацією зовнішніх приладів.

Цифро – аналоговий перетворювач (ЦАП). Принцип цифро –аналогового перетворення простий: кожний біт інформації вносить свою частку у загальний сигнал, що виникає на виході аналогово суматора. Загальний сигнал, що виникає на виході суматора визначається сумою сигналів від кожного біта. Таким чином ЦАП перетворює цифровий двійковий код $Q_n\ Q_{n-1}...Q_2Q_1Q_0$ в аналогову величину – вихідну напругу або струм. Роботу триразрядного ЦАП можна описати наступним рівнянням:

$$A = e(Q_0 \times 1 + Q_1 \times 2 + Q_3 \times 8),,$$

де A – вихідний аналоговий сигнал, e – напруга або струм , яка відповідає „вазі" наймолодшого розряду двійкового числа.

Спрощена схема ЦАП наведена на рис 5.1. Якщо один з розрядів має рівень одиниці (наприклад замкнено ключ А), то на загальному резисторі(R_i) виникає спад напруги. Резистори R1 – R4 створюють разом з резистором Ri подільник напруги. Для даної схеми резистори R1 – R4 створюють двополюсник з еквівалентним опором $R_{ек}$. Причому R4>>Ri. У такому випадку напруга на виході знаходиться з виразу

$$U_{вих} = E\frac{Ri}{R_{ек} + Ri} \approx E\frac{Ri}{R_{ек}}.$$

Загальна провідність двополюсника дорівнює

$$\frac{1}{R_{tr}} = \frac{A}{8R} + \frac{B}{4R} + \frac{C}{2R} + \frac{D}{1R}.$$

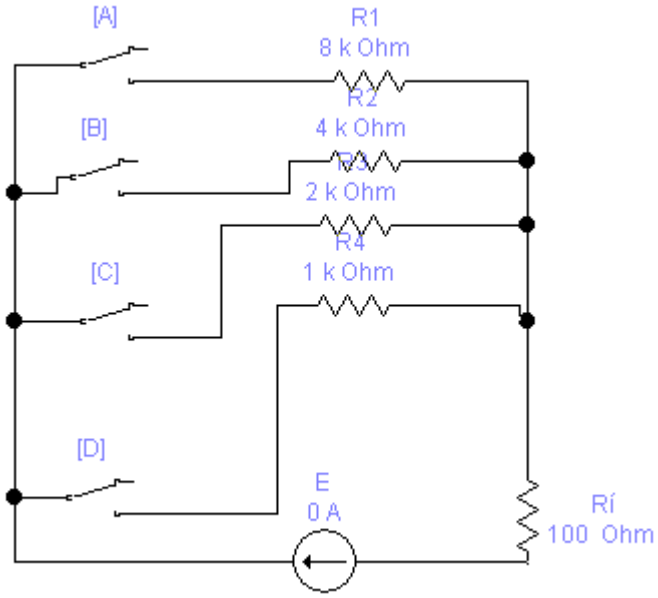

Рис.5.1 – Спрощена схема ЦАП

Якщо один з розрядів дорівнює нулю то відповідний член у рівнянні двополюсника також дорівнює нулю Тому можна остаточно записати

$$U_{вих} = E\frac{Ri}{8R}(A \cdot 1 + B \cdot 2 + C \cdot 4 \cdot D \cdot 8)$$

Більш детально із схемами конкретної реалізації ЦАП, особливостями їх використання можна ознайомитись у спеціальній літературі.

Аналого – цифровий перетворювач (АЦП) перетворює аналоговий сигнал, що надходить із зовнішнього об'єкта, у цифровий двійковий сигнал. Існую багато схем АЦП. Загальний принцип їх роботи базується на різних методах порівняння аналогово сигналу, що надходить зовні, та коду, що генерується перетворювачем. В результаті цієї операції кожному значенню вхідного аналогово сигналу буде відповідати деяке значення цифрового коду. Таким чином буде відбуватися постійне перетворення вхідного аналогово сигналу у цифровий (двійковий) код. Прикладом АЦП є „миша", яка використовується у персональному комп'ютері.

Існують різ схеми АЦП для вводу аналогової інформації у мікропроцесорні пристрої. За принципом дії їх можна поділити на дві великі групи: перетворювачі що використовують послідовний двійковий код; перетворювачі, що використовують паралельний двійковий код.

Перетворювачі першої групи (послідовний перетворювач) при подачі на них аналогового сигналу перетворюють його у код і запам'ятовують. Після перетворення інформації на виході АЦП з'являється сигнал про закінчення

процесу перетворення. Після цього інформація може бути видана по запитанню іншого пристрою. Видача інформації починається при приході на вхід імпульсів опитування і на кожний імпульс АЦП видає нульовий або одиничний сигнал. Таким чином на вхідному регістрі пристрою накопичується двійковий код величина якого пропорційна величині аналогового сигналу.

Перетворювач другого типу Паралельний перетворювач) має вихід сигналів у вигляді декількох розрядів (8,10,16 біт). Перетворений сигнал одночасно поступає на певний прилад. Швидкодія перетворювача другого типу значно вище ніж першого. Однак він потребує більш складної схемної реалізації.

Існують датчики які самі являються перетворювачами аналогово сигналу у цифровий код. До них можна віднести датчики температури у яких частота вихідного сигналу пропорційна температурі середовища де вони розташовані. Такий сигнал можна безпосередньо вводити у мікропроцесорну систему використовуючи відповідні порти комп'ютера.

5.2 Використання комп'ютерної технології для автоматизації вимірювань.

Розглянуті основні елементи дозволяють створювати на їх основі складні електроні прилади, які використовуються як для обробки інформації так і для керування різноманітними технічними об'єктами.

Фахівці з галузі товарознавство широко використовують комп'ютерні технології, які набули широкого впровадження при дослідженнях якісних характеристик харчових товарів і для розробки нових продуктів. Основна сфера використання комп'ютерів – це автоматизація проведення наукових досліджень. Це значно підвищує точність одержання результатів при проведенні досліджень і дає можливість зібрати необхідний матеріал для подальшої обробки.

Розглянемо приклад використання персонального комп'ютера (ПК) для побудови вимірювальної системи, що досліджує рухомість води у харчових продуктах. Сучасний метод визначення цього параметра базується на використання спектрометрів ЯМР імпульсного типу. Проведення подібних досліджень з використанням спектромета ЯМР характеризується малим часом вимірювання та невеликою масою зразка, необхідного для проведення дослідження. Вимірюваним параметром під час проведення досліджень із застосуванням спектрометрів ЯМР є амплітуда луна – сигналу на виході вимірювального пристрою, яка отримана після збудження зразка радіо імпульсами з заданими параметрами. Зазвичай, для фіксації луни – сигналу використовується осцилограф. Проте за цієї методики визначення точного значення амплітуди луни – сигналу залежить не тільки від якості настроювання вимірювальної апаратури, але і від суб'єктивних навичок дослідника. Крім того, слід зазначити, що вимірювання сигналу проводиться на тлі перешкод, що спотворюють його форму, а також і амплітуду, що значно знижує точність подальшого обчислення вимірюваного параметра, Використовуючи

комп’ютерну технологію можна значно спростити процедуру вимірювання та підвищити точність вимірювання.

Структурна схема вимірювального комплексу на основі персонального комп’ютера зображена на зображена на рис.5.1. В даному комплексі використовується АЦП для перетворення сигналу ЯМР та логічні схеми для керування спектрометром. Зв’язок комп’ютера та спектрометра відбувається через спеціальну логічну схему, що керується безпосередньо процесором, та зовнішній паралельний порт.

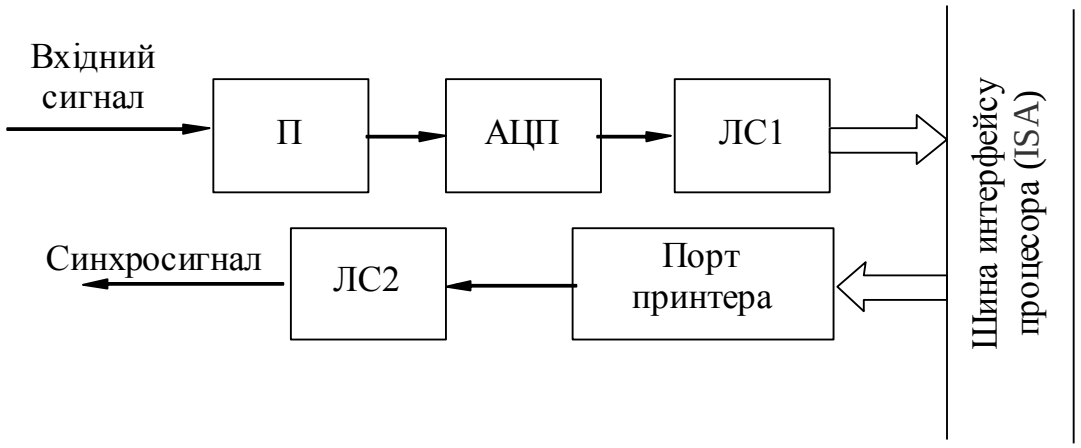

Рис. 5.1 - Структурна схема вимірювального комплексу

Посилений сигнал від спектрометра ЯМР надходить на АЦП паралельного типу (розрядність перетворювача становить 10 біт) і потім у вигляді паралельного двійкового коду через логічну схему (ЛС1), що являє собою шинний формувач з інтерфейсом ISA, надходить через загальну шину комп’ютера у процесор для обробки та наступного запису в пам'ять. В адресному просторі пристроїв комп'ютера АЦП займає три адреси і сприймається процесором як зовнішній пристрій. Сигнал про початок вимірювання через паралельний порт комп'ютера і далі через логічну схему (ЛС2) надходить на спектрометр ЯМР для запуску його у роботу Така послідовність дій, виконуваних комп'ютером і логічними схемами, забезпечує повну автоматизацію процесу вимірювання амплітуди спінової луни спектрометра ЯМР.

Автоматизація процесу вимірювання досягається за рахунок використання прикладного програмного забезпечення (ПЗ), що об’єднує як роботу комп’ютера, так і додаткових блоків. Завдання режиму роботи виконується з використанням "меню", в якому передбачаються відповіді тільки "так" чи "ні". Інформація видається на екран дисплея у визначеній послідовності, що виключає її пропуск і як наслідок неправильне функціонування вимірювального комплексу.

На рис. 5.2 показано інформацію, що записана комп’ютером та виведена на екран дисплея під час реєстрації всього вимірювального процесу. Під час

одного вимірювання комп'ютер реєструє 640 значень аналогового сигналу, що надходить з електронних блоків спектрометра ЯМР. З рисунку видно, що на вимірювальний сигнал накладені перешкоди, які обумовлені високою чутливістю приймальної апаратури спектрометра.

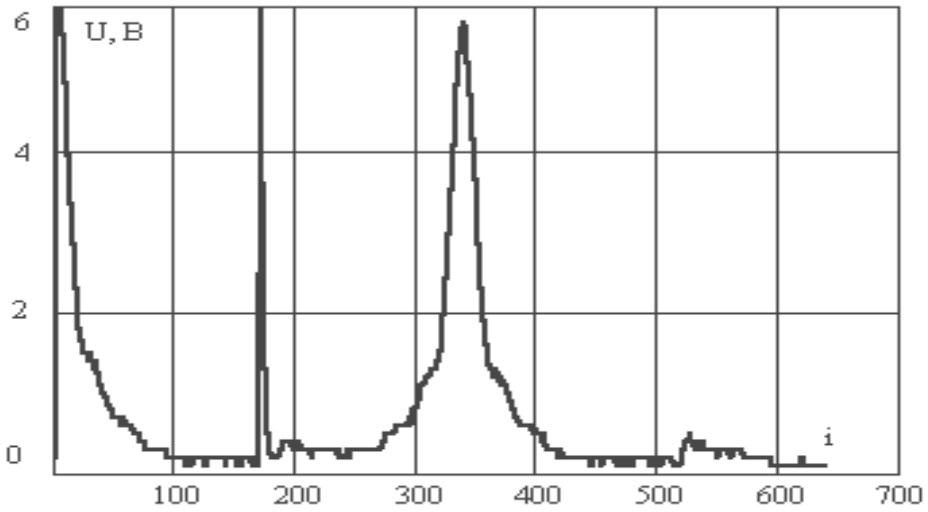

Рис. 5.2 - Запис сигналів зондуючих імпульсів та луни-сигналу процесу вимірювання установки ЯМР

Перші два імпульси на осцилограмі це зондуючі імпульси генератора установки ЯМР. Інтервал часу між ними складає 0,01 сек. Останній імпульс це луна – сигнал амплітуду якого треба визначити для подальших досліджень.

На рис. 5.3 показано окремо записаний луна-сигнал ЯМР з інтервалом опитування АЦП у 10 мсек.

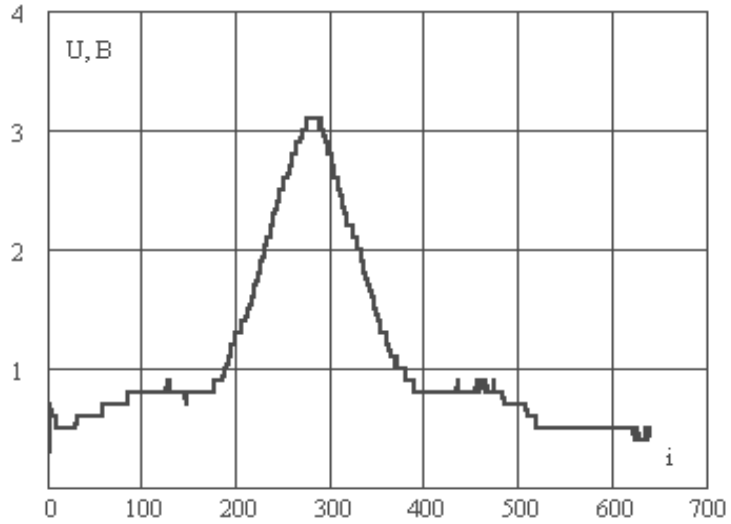

Рис. 5. 3 - Запис сигналу спінової луни спектрометра ЯМР

Для зменшення впливу імпульсних перешкод на результат вимірювання значення амплітуди для подальшого розрахунку використовують середнє значення декількох вимірювань. Записаний корисний сигнал зберігається у комп'ютері у вигляді числового файла і може бути використаний для подальшого дослідження відповідним програмним забезпеченням.

Кількість вимірювань при необхідності також задається та контролюється програмою керування вимірювальної системи. Програмування програмного прикладного забезпечення для керування додатковими приладами виконане па програмі „Турбо Паскаль". Для безпосередній обробки інформації, що надходить до АЦП, з метою підвищення швидкості вимірювання, використовувались програмні модулі побудовані на мові асемблер.

Наведений приклад використання комп'ютера для проведення наукових досліджень показує доцільність використання комп'ютерних технологій в проведенні досліджень пов'язаних з визначеннями окремих показників харчових продуктів та науково – дослідній роботі.

ЛЕКЦІЯ 6. ОСНОВНІ ВІДОМОСТІ ПРО МІКРО -ЕОМ

6.1 Мікро ЕОМ. Структура, призначення основних елементів.

В процесі розвитку та удосконалення засобів мікроелектроніки та обчислювальної техніки були створені спеціалізовані програмовані великі інтегральні схеми які мають назву мікропроцесори. Вони створили передумови для розробок і впровадження у промисловість контролюючих, керуючих та обробляючих систем на базі мікро -ЕОМ. Ці прилади можуть бути безпосередньо вбудовані у технологічне обладнання і дозволяють суттєво підвищити рівень автоматизації технологічних процесів, сприяють зростанню продуктивності виробництва, економії сировини, матеріалів і енергії.

Масовий випуск відносно недорогих мікропроцесорних наборів дозволяв створювати високоефективні спеціалізовані мікро - ЕОМ для систем промислової автоматики, автоматизованого керування технологічними процесами тощо. Причому, спеціалізація щодо налаштування до різного виду обладнання полягала шляхом застосування різного програмного забезпечення.

В наш час випускається велика кількість мікро -ЕОМ з різноманітними функціональними можливостями для найрізноманітнішого застосування, починаючи від окремих приладів до систем автоматизованого управління (АСУ) складними технологічними об'єктами.

В наш час більшість мікропроцесорних систем (МПС) будуються на основі мікро контролерів, які мають невелику вартість та великі можливості. Основа сучасної МПС запропонована Джоном фон Нейманом ще в 1945році і реалізує концепцію зберігаючої програми: в якій програми і дані зберігаються в одній і тієї ж пам'яті. Дії, що виконуються нею визначаються, блоком керування та аріфметично- логічним пристроєм. У відповідності з програмою

МП вибирає з пам’яті і виконує команди послідовно. Адреса чергової команди задається лічильником адресу. Як розвиток класичної архітектури , слід розглянути Гарвардську архітектуру, Вона заснована на розподілі пам’яті програм і даних з відповідним розподілом шин. Це дозволяє підвищити швидкість роботи і продуктивність МПС. Основні структури МП можна поділити на:

CISC – мікропроцесор з повним набором інструкцій, до яких відноситься процесор К580. Склад та призначення регістрів неоднорідні, широкий набір команд ускладнює декодування інструкцій, на що витрачаються апаратні ресурси. Зростає кількість тактів що необхідні для виконання програми. Все це призводить до ускладнення системи команд і як наслідок програмування. Даний процесор буде детальніше розглянути.

RISC – це МП зі скороченою системою команд Вони мають набір однорідних регістрів універсального призначення. Система команд відрізняється відносною простотою, апаратурна реалізація такої архітектури дозволяє з невеликими витратами декодувати і виконувати інструкції за мінімальну кількість тактів роботи.

DSP – цифровий сигнальний процесор використовується для пристроїв що передбачають великий обсяг обробки інформації. Вони мають 6 шин та декілька банків пам’яті.

Незважаючи на широкі можливості щодо застосування мікро -ЕОМ має структуру що складається з трьох основних частин: пам'ять, центральний процесора, інтерфейс вводу /виводу. Центральний процесор побудований як правило на одній мікросхемі і має назву мікропроцесор (МП). Він має у своєму складі декілька регістрів для утримання інформації, арифметично- логічний пристрій (АЛУ) для виконання арифметичних та логічних операцій, а також пристрій керування. Швидкість роботи МП визначається тактовим генератором який працює на фіксованій частоті. С состав МП входить програмний лічильник який при виконанні команд переходить від однієї комірки паняті до іншої де знаходиться програма. У переважній більшості програми виконуються не безпосередньо у тому порядку в якому вони записані у комірках пам’яті. Часто зустрічаються переходи уперед або назад в залежності від результаті виконання команд.

Логіка роботи дешифратора команд та пристрою керування реалізована у кристалі МП і визначаються особливостями його роботи.

В мікро -ЕОМ що побудовані на основі мікропроцесора зв’язки між окремим блоками здійснюються за допомогою шин. Під шиною розуміється група ліній передачі яка мають функціональну загальність. Передача даних по шині здійснюється послідовно. Типові зв’язки у мікро -ЕОМ здійснюються трьома шинами. Це шини адресу (ША), даних (ШД)) і управління (ШУ).Шини адресу та управління є одно направленими. В них сигнали ідуть від центрального процесора до інших блоків. Шина адресу має 16 розрядів. Дані, що передаються ША формуються у процесорі і потрібні для визначення шляхів передачі в середовищі мікро -ЕОМ. Вони необхідні для вибору комірок пам’яті куди необхідно занести або зчитати інформацію.

Шина даних є двунаправленою шиною. Дані по ліням цієї шини можуть передаватися до будь якого пристрою від процесора і навпаки.

При такій організації роботи передача даних має особливості:

- вихід будь якої схеми що не посилає дані у даний час, повине знаходитися у стані з великим опором (стан Z), щоб не втратити інформаці.;

- кожний прилад, що підключений до шини має свою адресу або номер;

Типова загальна схема мікро -ЕОМ що використовує в своєму складі мікропроцесор наведена на рис. 6.1.

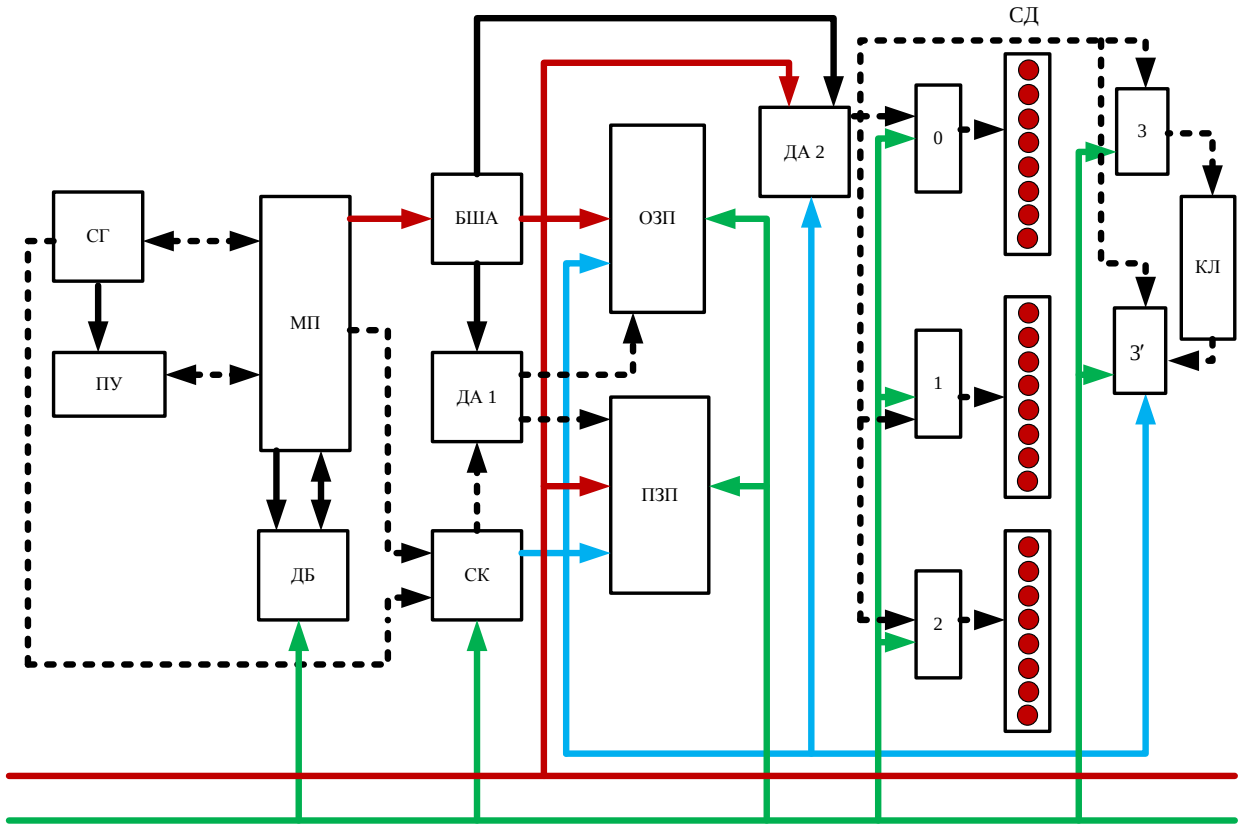

Рис. 6.1 – Типова структурна схема мікро -ЕОМ

На даній схемі наведені основні блоки мікро -ЕОМ. Управління роботою здійснюється двома основними блоками схемою покрокового управління програми (ПУ) та схемою системного контролера (СК). Перш схема призначена для аналізу виконання команд, що вводяться програмою. Системний контролер призначений для організації передачі інформації при роботі мікропроцесора. Процесор, що наведений на рисунку має ША, ШД, а також деякі сигнали керування з яких СК формує управляючи сигнали. До них належать: ЧИТАННЯ, ЗАПИС, ВИІВІД НА ЗОВНІШНИЙ ПРИСТРІЙ, ВВОД З ЗОВНІШНЬОГО ПРИСТРОЯ.

При організації шинного зв’язку треба вірно оцінити величини струмового навантаження по кожній шині. Для цього до шин приєднують

буфери шини адреса (БША). Під буфером розуміють спеціальну схему яка забезпечує електричне узгодження кіл передачі сигналів.

Шина даних має спеціальний двонаправлений буфер (ДБ) що забезпечую відповідну передачу сигналів. Дешифратори (ДА1), (ДА2) призначені для дешифрування адресів оперативного запам’ятовуючого пристрою (ОЗП) та постійного запам’ятовуючого пристрою (ПЗП). Із зовнішніми пристроями мікро -ЕОМ з’єднується за допомогою портів виводу (0,1,2). Для контролю проходження інформації порти обладнані відповідною індикацією. Порти (3 та 3') призначені для з’єднання з клавіатурою.

6.2 Пристрої пам’яті.

Магнітні та оптичні пристрої для зберігання даних. Питання зберігання та виймання інформації дуже важливі для кожної комп’ютерної системи. Сучасна дискова пам'ять дозволяє видобувати інформацію у будь якому порядку незалежно від порядку її запису. Магнітна запис на вінчестерах дозволяє зберігати великі об’єми інформації на протязі довгого часу. Наряду з цим використовується оптична запис яка дає можливість робити носії знімними, що спрощує перенос інформації.

Оперативні запам’ятовуючи пристрої. В основі даних пристроїв лежать тригерні схеми які дозволяють зберігати біти інформації. Їх можна об’єднати у відповідні регістри для запам’ятовування цілих байтів інформації. В якості паняті можна використовувати і регістри зсуву в яких реалізується принцип «перший прийшов – перший вийшов».

Практично на одному кристалі можна створити мільйони тригерів з можливістю прямого доступу до кожного.

Існують два основних типа ОЗП. Статичний ОЗП (SRAM) яке створюється з використанням тригерів у комірках. В динамічних ОЗП (DRAM) двійкове число зберігається у вигляді заряду на затворі МОП транзистора. Це дозволяє створити дуже компактну схему. Така комірка складається з декількох транзисторів і працює наступним чином: при приході сигналу з ліні] D високий логічний рівень з виходу схеми D відкриває транзистор Т1 і відбувається заряд або розряд конденсатора С відповідно тому який рівень сигналу на вході – високий або низький. Роль ємності С грає паразитна ємність затвору наступного транзистора Т2. Коли ємність С заряджена, то транзисторТ2 відкритий і його стік має низький потенціал. Коли відбувається звертання до цієї комірки, транзистор Т3 відкритий. Він з’єднує вихідну шину із стоком транзистора Т2 і створює на ній низький рівень коли С заряджений, заряд не може зберігатися довго і тому кожні 2 мс. Відбувається цикл регенерації який полягає у перезапису інформації. Приклад комірки динамічного ОЗП на один біт наведено на рис.6.2

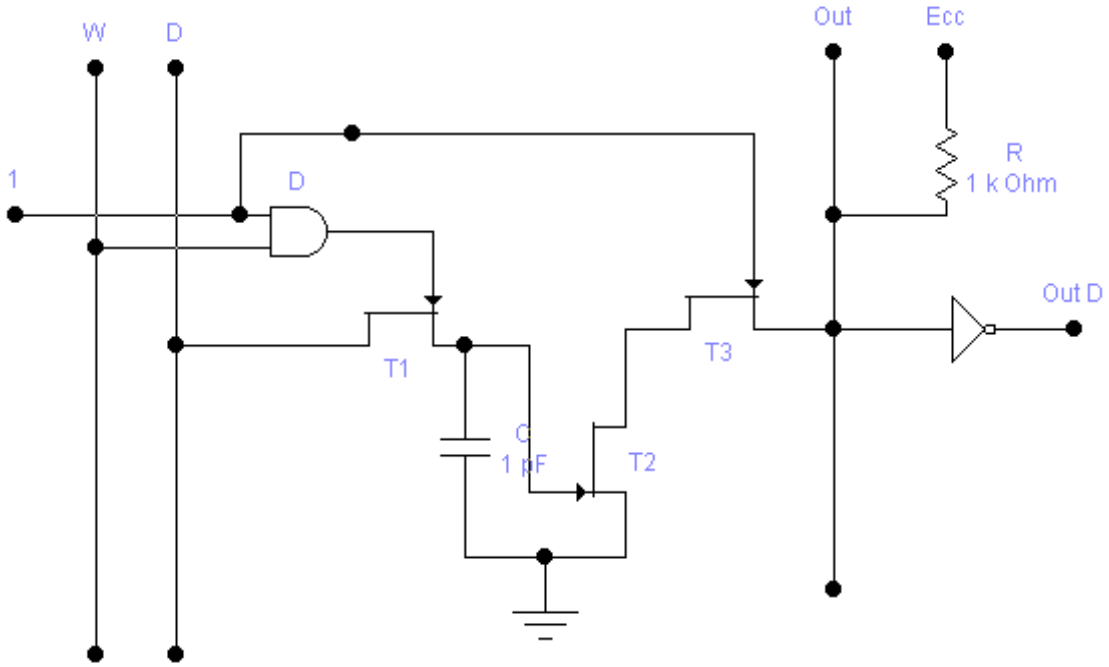

Рис. 6.2 – Комірка динамічного ОЗП на один біт.

Незважаючи на лопаткове ускладнення динамічний ОЗП має менші розміри та дешевше ніж статичний ОЗП. Основною галуззю використання статичних ОЗП є енергонезалежні ОЗП, в яких використовуються КМОП-схеми.

В мікро -ЕОМ використовуються фіксовані дані та програми. Вони зберігаються у постійних запам'ятовуючих пристроях (ПЗП) які мають позначення (ROM). Вони дуже корисні коли дані або програми розмножуються у багатьох примірниках. Вони схожі на динамічні ОЗП і являють собою решітку з транзисторів (звичайно МОП-транзистори з каналом n-типу). Конфігурація комірок збереження встановлюється під час виготовлення. У комірці пам'яті яка повинна зберігати логічну одиницю встановлюється транзистор, а у тих комірках де повинен зберігатися логічний нуль транзисторів немає.

У програмованих ПЗП (PROM) транзистори встановлюються у всі комірки і там встановлюють плавкі вставки. Користувач сам перепалює непотрібні вставки і записує власні дані. Для цього використовується спеціальний програма тор який послідовно перебігає адреси і подає струм для розплавлення вставки.

При роботі з експериментальними зразками доцільно використовувати ПЗП в яких можна змінювати інформацію (EPROM) не змінюючи саму мікросхему. В таких ПЗП в якості комірок пам'яті використовують n- канальні МОП транзистори з індукованим каналом які мають спеціальний ізольований плаваючий затвор. Перед програмуванням всі транзистори ставлять у режим відсічки що відповідає запису логічно гної одиниці. Програмування потребує ретельного контролю і полягає у подачі відносно великої напруги (25 В) між

стоком та допоміжним затвором. В результаті подачі напруги відбувається лавинний пробій в ізолюючім шарі оксиду між каналом та плаваючим затвором. Після зняття напруги пробій зупиняється, у плаваючому затворі залишається заряд і транзистор відпирається що відповідає запису у цій комірці нуля. Щоб знищити заряд цей кристал необхідно підвергнути дії ультрафіолетового світла, під дією якого заряд стікає з плаваючого затвору , дані стираються і схема знов готова для програмування.

Існують прилади де знищення заряду відбувається шляхом подачі напруги відповідного значення (EEPROM). Такі прилади потребують меншого часу на програмування. У популярних картах паняті (флєш) можна переписувати дані при звичайному рівні напруги 5 В. Вони являють собою енергонезалежні ОЗП.

Постійні запам’ятовуючи пристрої можна розглядати як компактні комбінаційні логічні схеми. Будь яка комбінаційна логічна схема працює за певною таблицею, що утворена відповідними логічними рівняннями. Якщо використовувати вхідні біти в якості розрядів двійкового числа

6.3 Мікропроцесор. Структура, призначення основних елементів.

Структура мікропроцесора (МП) зображена на рис 6.3. Основними блоками МП є арифметично логічний пристрій (АЛП) з регістром акумулятором, блок регістрів загального призначення (РЗП) із схемою вибірки, регістр команд з дешифратором команді формувачем машинних циклів, регістри тимчасового зберігання даних W та Z, прапор омий регістр, пристрій керування та синхронізації, буфери шини даних та адресу, буфер акумулятора, схема збільшення та зменшення.

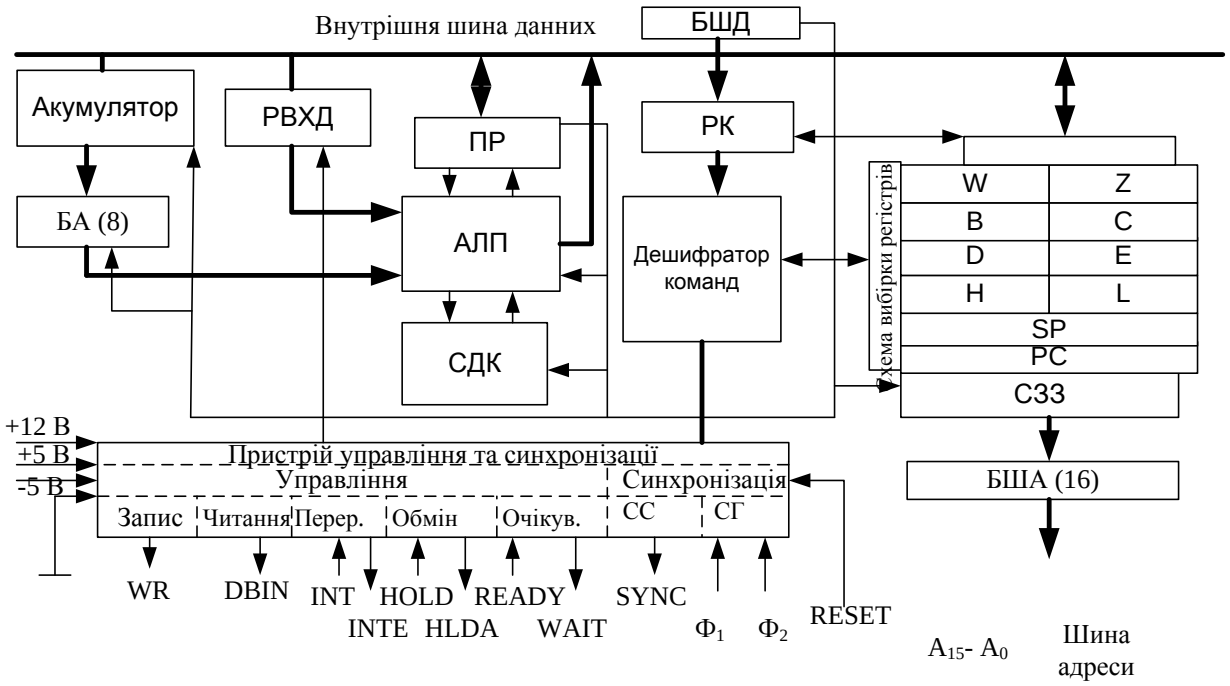

Рис. 6.3 – Структура мікропроцесора.

Програмісту доступні регістри B,C,D,E,H,L (РЗП) один регістр A, 16 розрядний регістр стеку, 16 розрядний регістр лічильника команд. Програмно недоступні регістри Y та Z.

Регістри загального призначення (РЗП) мають розмірність в один байті використовуються для збереження даних при виконанні обчислення. При обробці16 розрядних чисел можливо об'єднання регістрів (B,C);(D,E);(H,L).

Аккумулятор спеціальний однобайтовий регістр (A). Він є джерелом одного з операндів при виконанні арифметичних дій,

Регістр команд – однобайтовий регістр де зберігається код команди, яка виконується.

Лічильник команд –двубайтний регістр або програмний лічильник. Він зберігає адресу наступної команди, яка повинна бути виконана наступною. Лічильник команд автоматично збільшує адрес, який він зберігає в залежності від довжини команди. На вміст цього регістра можна подіяти тільки шляхом виконання команд які змінюють порядок виконання програми.

Показник стеку – двубайтний регістр який позначається SP. Стек це відповідним чином організована ділянка оперативної пам'яті що виділяє програміст для тимчасового зберігання вмісту внутрішніх регістрів.

Прапорів регістр – вміщує 5 двійкових розрядів які мають назву пропори. Іноді він має назву регістра ознак .Значення прапора вказує на результат виконання певної операції. Мікропроцесор КР580ІК80А має наступні значення прапорів: Прапор нуля (Z – zero), прапор переносу (C – carry), прапор знаку (S – sign), прапор парності (P – parity), прапор додаткового переносу (AC – auxiliary carry). Прапори встановлюються або скидаються автоматично.

ЛЕКЦІЯ 7. ПРОГРАМУВАННЯ МІКРОПРОЦЕСОРІВ

7.1 Основні команди мікропроцесора.

Мікропроцесор має 16 розрядну шину адресу та 8 розрядну шину даних яка може передавати дані в обох напрямках. Виконання кожної команди здійснюється у певній послідовності яка синхронізується по часу сигналами тактового генератора. Програма являє собою процес розв'язування задачі з урахуванням можливостей МП. Результат опису рішення задачі іменують алгоритмом. При розробці програм доцільно використовувати блок-схеми алгоритм ва які допомагають користувачу наочно уявити роботу програми.

Підпрограма – частина програми що використовується при виконанні програми декілька разів. Текст підпрограми записується програмістом один раз. Для виконання підпрограми достатньо вказати відповідну команду виклику, що адресується до області пам'яті, де розташована підпрограма.

Команда МП – це двійкове слово, що змушує МП виконувати певні дії.

Команда МП повинна містити наступну інформацію:

- повідомляти МП, що робити (виконувати додавання, очистку, пересилання, зсув і т.п.);
- - вказувати адресу, тобто місцеположення даних, що обробляються.

Наприклад, команди можуть інформувати МП про наступне: додати до вмісту акумулятора копію вмісту деякої області пам'яті, очистити акумулятор, перемістити дані з регістру А до регістра В, виконати зсув вмісту акумулятора і т.п. З наведених прикладів команд слідує, що МП одержує від команди інформацію не тільки про те що робити, але і про те, де знаходяться дані – об'єкти маніпулювання.

Команда складається з коду операції (КОП) і адреси. Код операції повідомляє МП, що робити; адреса вказує місцеположення даних, що беруть участь в операції. Якщо довжина команди складає два або три слова, то перше з них – це код операції, а друге і третє – адреси.

Однак є команди і без адреси. Наприклад, команді, що наказує МП зупинити роботу, адреса не потрібно. Тип звернення (адресації) до даних прийнято називати засобом адресації. Розглянемо види адресації

Неявна адресація. Однобайтні команди не адресуються до даних, розташованих в пам'яті; вони оперують даними, завантаженими в регістр, регістрову пару або даними, які зберігаються в області пам'яті, адреси якої знаходиться у регістровій парі. Наприклад, 1-байтовая команда пересилення даних з регістра А до регістра В складається з коду операції, адреси джерела даних (регістра А) і адреси приймальника даних (регістра В). Адреси джерела і приймальника вказані в команді неявно; інколи говорять, що вони «вбудовані» в команду. Ось чому така адресація називається неявною. Приклад розташування інформації уданій команді наведено на рис.7.1

1	0	0	0	0	1	0	1
КОП (код операції)		Адреса регістра А			Адреса регістра В		

Рис. 7.1 – Структура однобайтної операції з неявною адресою

Однобайтні команди виконуються швидше будь-яких інших команд. Відомо, що для виконання будь-якої команди МП необхідно здійснити дві послідовності операцій, що мають назву *цикл вибірки і виконання*. У випадку однобайтової команди МП витрачає на це два мікроцикли: один – на операцію вибірки, другий – на операцію виконання. Інакше кажучи, команди з неявною адресацією виконуються найбільш швидко.

Безпосередня адресація. Код операції команди з *безпосередньою адресацією* розміщується у першому байті, за кодом операції ідуть слідом дані, що займають 1 або 2 байта.

Пряма адресація. Команди з *прямою адресацію* можуть мати довжину, що дорівнює 2 або 3 байти. Перший байт – це код операції, другий і, якщо є третій – для адреси. Адреса вказує, область пам’яті, в якій знаходяться дані, що підлягають обробці.

7.2 Приклад програмування мікропроцесора

Прикладом використання такої адресації може служити команда записи вісту акумулятор в пам'ять за адресою 000FH (рис.7.2). Ця адреса, задана в двійковій формі, займає другий і третій байти 3-байтової команди.

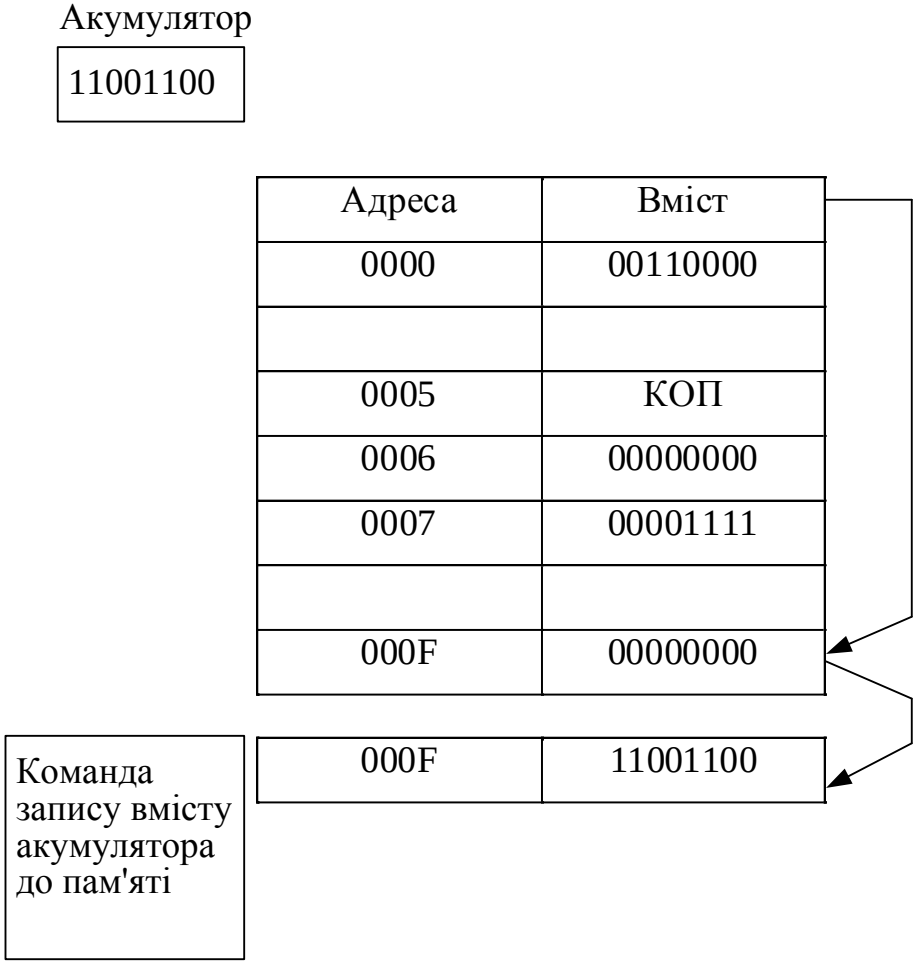

Рис. 7.2 – Принцип реалізації команди з прямою адресацією

Адреси задаються у шістнадцятковій формі, а вміст областей пам’яті, що адресуються, - у двійковій. Код операції команди запису вмісту акумулятора до памяті знаходиться в області пам’яті з адресою 0005, а області з наступними адресами 0006 і 0007 містять інформацію про місцеположення необхідних даних, тобто це адреса 000FH. Перед виконанням команди вміст акумулятора дорівнює 1100 1100. В області пам’яті з адресою 000F міститься двійкове число 00000000. Після виконання команди копія вмісту акумулятора виявляється розміщеною в області пам’яті з адресою 000FH.

Посередня адресації. Більшість МП володіє ще одним засобом адресації до пам'яті, що реалізується командами довжиною тільки в одне слово. Така адресація називається *посередньою*. Окрім коду операції в такій команді вказується номер регістра вміст якого – адреса місцеположення даних в пам'яті. Так, при використанні посередньої адресації у 8-разрядному МП відповідна команда вказує, в якій регістровий парі міститься адреса місцеположення даних в пам'яті.

Машинні коди і Асемблер. Мікропроцесор сприймає інформацію у вигляді двійкових чисел (машинних кодів). Разом з тим, безпосереднє укладання програм у двійкових кодах або у вигляді шістнадцяткових (Нех) чисел надто важке, програма практично є нечитабельною. (Наприклад: 01 0001 0011 1100, С3 00 .).Тому кожній машинній команді ставиться у відповідність мнемонічний запис з декількох англійських літер – стисло пояснюючий сенс команди. Сукупність таких літерних команд утворить систему команд даного МП на мові Асемблер. Формат команди залежить від архітектури МП. ЇЇ довжина може бути кратна байту або довільною.

Повна група команд мікропроцесора складається з 4-х груп.

1. Група команд пересилання даних і завантаження регістрів *(команди переміщення даних)* служать для пересилання даних в різноманітні пристрої зберігання інформації мікро -ЕОМ, а також для пересилання даних з цих пристроїв. До областей зберігання інформації відносяться як комірки пам'яті, так і регістри.

Команди пересилання даних, можливо, слідувало б називати командами «копіювання», тому що практично вони здійснюють переміщення саме копії даних. Так, наприклад, команда STA F16 переміщує дані з акумулятора в область пам'яті (А16). Після виконання даних команди і в означеній комірці пам'яті, і в акумуляторі знаходяться одні і ті ж дані.

Команди бувають одно-, двох- і трибайтні. При цьому перший байт в будь якому випадку – код команди, а інші байти – константи або адреси. Слід звернути увагу, що в трибайтних командах число або адреса містяться в зворотному порядку – спочатку молодший байт, після цього старший байт. Наприклад, в команді 1.4 LDA 0A73FH – відповідні їй коди містяться наступним чином: 3А 3F А7 – код команди LDA, 3F - молодший і А7 – старший байт адреси. МП, прочитавши код дво- або трибайтної команди, автоматично зчитує вміст однієї або двох наступних комірок памяті і завдяки цьому – необхідні для виконання команди параметри. Будь-яких особливостей команди 1 групи не мають.

2. Група команд арифметичних і логічних операцій. Більшість з них виконують операції над даними, що заздалегідь поміщені в акумулятор. В акумуляторі ж і залишається результат операції. При цьому, в залежності від результату, в регістрі стану (ознак) відповідні розряди встановлюються в 0 або1.

3. Група команд переходів. Команди переходів дозволяють організувати розгалуження програм, формувати цикли і т.п. операції. Це істотно поширює можливості програміста при розв'язанні різноманітних задач. Переходи можуть

бути як безумовними (JMP A16, CALL A16), так і умовними. Якщо умова, що задана в команді, не виконується, МП автоматично переходить до наступної команди.

Команди виклику підпрограм відрізняються від команд переходів тим, що раніше, ніж перейти до виконання підпрограми, МП автоматично заносить адресу наступної команди основної програми в спеціально організуємо область пам'яті – стек. В кінці підпрограми за командою RET МП витягає зі стека цю адресу, встановлює її в лічильник – регістр команд, і з нього продовжує виконання програми.

1. Група команд посередньої (непрямої) адресації. Команди посередньої адресації дозволяють економити обсяг пам'яті, що відведена під програму (бо більшість команд однобайтні), а також підвищити продуктивність МПС при організації дій з масивами чисел, що розміщені в пам'яті. При цьому початкова адреса завантажується в регістрову пару, а наступні адреси формуються командами INR R або DCR R (звичайно всередині циклу).

Система команд МП містить також ряд спеціальних команд. Наприклад команда NOP дозволяє зарезервувати комірку пам'яті під можливі доповнення до програми, а також при необхідності реалізувати невелику часову затримку, так як МП витрачає деякий час на її розпізнання і перехід до читання наступної команди.

Команда HLT застосовується у випадку необхідності зупинки роботи МП. Поновлення функціонування МП в цьому випадку можливо тільки після подачі сигналу скидання RES або запиту на переривання.

3. Приклад програмування мікропроцесора. Побудуємо програму що може сумувати перші 20 чисел натурального ряду. Для програми треба застосувати команду MVI r, яка здійснює безпосереднє завантаження у будь який регістр дані які знаходяться у другому байті команди (у даному випадку необхідно завантажити число 0 D). Для здійснення операції S=S+N використовується команда ADD r яка виконує операцію сумування вмісту будь якого регістру з вмістом акумулятора та наступного занесення результату операції в акумулятор. Для виконання операції N=N+1 у машинній мові передбачена спеціальна команда INR r, яка здійснює збільшення вмісту регістру на одиницю. Для створення циклу можна використати команду JNZ, яка здійснює перехід у програмі при відсутності нуля в результаті будь якої попередньої операції (для нашої задачі це операція віднімання: 21-N). Програму можна змінити наступним чином, Замість операції сумування послідовності чисел які збільшуються можна сумувати послідовність чисел які зменшуються від 5 D до 0D. Тоді вміст регістру , що зберігає число N на деякому кроці програми перетвориться у нуль і цей ре6зультат можна буде використати при організації умовного переходу. Програма роботи наведена в таблиці 7.1.

Таблиця 7.1

Програма роботи

Адреса комірки	Код операції або вміст другого або третього байту	Мітка	Мнемокод	Коментар
014 000	076		MVI A 000Q	Завантаження регістра А числом OD
014 001	000			Число 0
014 002	026		MVI D 000Q	Завантаження регістра числом 20D
014 003	024			Число 20D
014 004	202	M1	ADD D	Складання вмісту регістрів А та D
014 005	025		DCR D	Зменшення на одиницю вмісту регістра
014 006	302		JNZ M1	Перехід до мітки M1 при відсутності нуля
014 007	004			Мітка M1 (адрес)
014 010	014			Мітка M1 (адрес)
014 011	323		OUT 000Q	Вивід даних з регістра А в порт P000
014 012	000			Номер порта
014 013	166		HLT	Зупинка програми

Таким чином схема алгоритму буде мати вигляд

Крок 1. Заносимо число 0D в акумулятор (A).

Крок 2. Заносимо число 20D (у восьмирічній системі числення 24 Q) у будь який регістр, наприклад регістр D/

Крок 3. Сумуємо вміст регістрів A та D.

Крок 4. Зменшуємо вміст регістра D на одиницю.

Крок 5. Якщо вміст регістру D не дорівнює нулю, здійснюємо перехід до кроку 3; в іншому разі переходимо до наступного кроку.

Крок 6. Переписуємо вміст регістра A в порт P000.

Крок 7. Кінець роботи програми.

На основі даного алгоритму розробимо програму рішення цієї задачі. Тільки замість команди INR r (збільшення вмісту регістра на одиницю) використаємо команду DCR r (зменшення вмісту регістра на одиницю), яка буде виконуватися притаманно регістру D. Крім того необхідно використати ще дві команди: OUT – двубайтна команда виводу даних з акумулятора у порт виводу який визначається адресою що розташована у другому байті команди, та однобайтова команда HLT – зупинка програми. Коди команд можна визначити у спеціальній літературі або довідниках.

Для розташування програми потрібно 12 комірок паняті з номерами від 014Q000Qдо 014 Q 13Q, у перші дві комірки з номерами 014000 та 014001 розташуємо обидва байти першої команди MVI A 000Q завантаження акумулятора (регістра A) числом 0 D. У першу комірку розташуємо код команди 076, а у другу – вміст другого байта даної команди – восьмирічний код числа 0D (число 000). У наступні дві комірки з номерами 014 002 та 014 003 вмістимо два байти наступної команди що завантажує регістр D десятковим числом 20. Восьмирічний код даної операції 026 буде знаходитися у у першій комірці, а восьмирічний код 024 десяткового числа 20 – у другій. В останню комірку з номером 014 013 внесемо восьмирічний код 166 команди HLT що зупиняє виконання програми. При організації циклу використана мітка M1 що показує на операцію яку необхідно виконати коли вміст регістру D не дорівнює нулю.

ЛЕКЦІЯ 8. ОСНОВНІ ВІДОМОСТІ ПРО МІКРОКОНТРОЛЕРИ.

8.1 Структура мікро контролера.

Слово контролер означає прилад керування у більш широкому сенсі ніж у теорії управління. Під управлінням звичайно розуміють дії що направлені на корекцію відхилень системи від заданого стану. При використанні контролера ми розуміємо виконання більш складних функції управління. В наш час МП з RISC –архітектурою є елементами одно кристальних МПС і носять назву мікро контролери (МК). На відміну від універсальної ЕОМ мікроконтролер (МК) призначений для керування конкретним технічним пристроєм. Розглянемо структуру та основні зв’язки МК типу PIC 16F87X. Вона наведена на рис. 8.1.

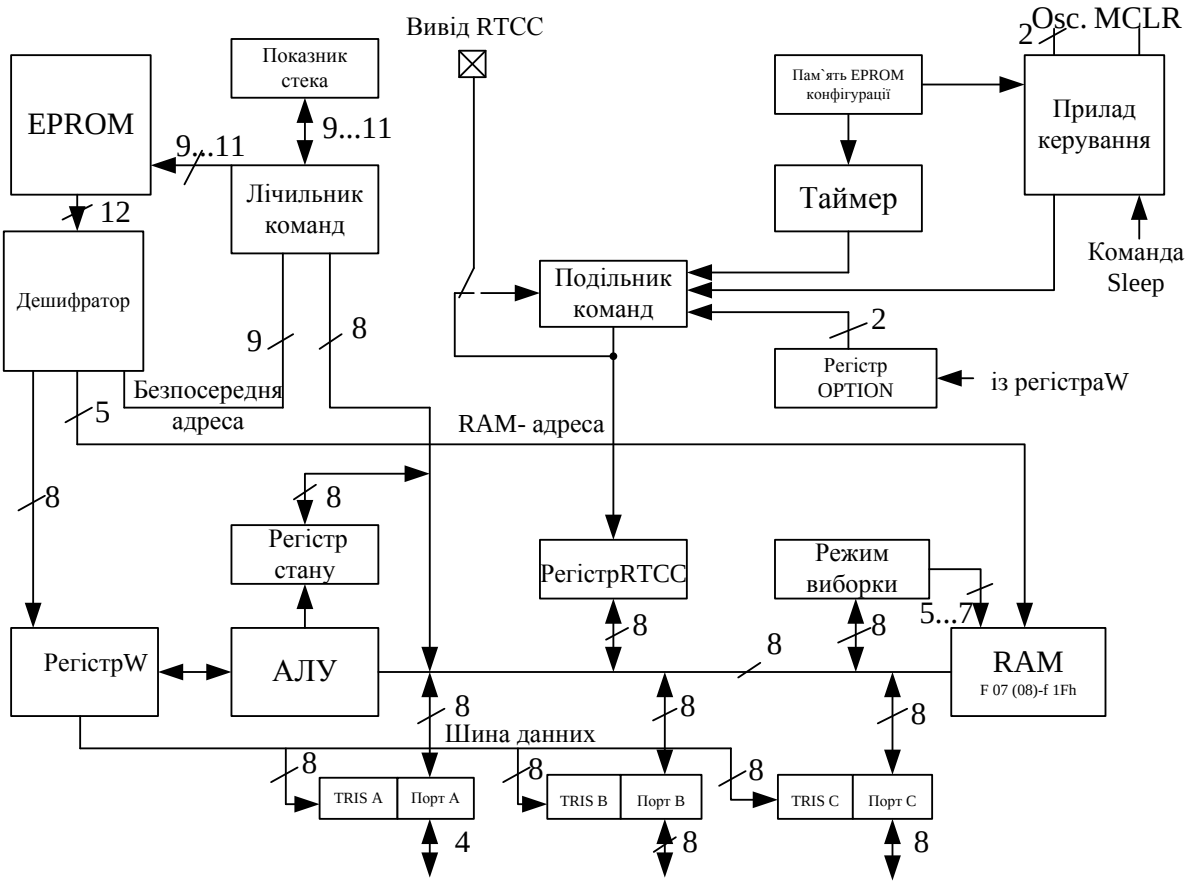

Рис. 8.1 – Структурна схема мікроконтролера.

Він є представником «Гарвардської» архітектури і має 12 розрядну шину для програм і 8 розрядну для даних Завдяки такої концепції виконання команд потребує 4 такти приладу керування. В основі МК лежить Арифметично-логічний пристій з робочим регістром W та регістром стану STATUS. Пам'ять програми виконана на EPROM і пов’язана з блоком кодування та декодування команд 12 розрядною шиною. Лічильник команд має розрядність 9-12 біт в залежності від виконання.

8.2 Призначення основних елементів.

Розглянемо окремі елементи даної схеми. Пам'ять RAM включає 8 регістрів загально призначення які призначені для виконання певних задач і які доступні для користовача.

До регістрів спеціального призначення відносяться наступні регістри.

Регістр W. Це важливий робочий регістр через який проходить весь обмін інформацією. У мікропроцесорах йому відповідає акумулятор.

Регістр OPTION. Він визначає умови користування регістром лічильника реального часу, дільника частоти та сторожового таймера.

Регістр TRIS визначає напрямок проходження сигналу через кожний вивід порту. Розряди регістра відповідають розрадам регістра порту. Коли в розряді регістра TRIS встановлена логічна 1, відповідний вивід порту працює як вхід, а в випадку лог. 0 – як вихід.

Під приладом керування на схемі позначений тактовий осцилятор який тактує роботу МК. В залежності від зовнішніх елементів частота роботи може бути від 32 кГц до 20 МГц.

Довгочасна пам'ять даних дозволяє прочитати і записати байт інформації. При запису байта автоматично стирається попередня інформація і записуються нові дані.

До додаткових функцій слід віднести наступні:

WDT – вартовий таймер який призначений для попередження наслідків від випадкового збою програми. Принцип дії полягає у періодичному скиданні таймера під керуванням програми або зовнішнього впливу до того як закінчиться витримка його часу і не відбудеться скидання МП. Коли програма працює нормально та команда скидання повинна регулярно виконуватися, охороняючи процесор від скидання. Якщо МП вийшов за межі програми або зациклився на якійсь програмі, команда скиданні не буде виконана в термін встановленого часу і відбудеться повне скидання МП, що поверне систему на початок роботи. Даний таймер не використовує зовнішніх компонентів і працює на вбудованому генераторі навіть при відключенні генератора тактової частоти процесора.

Режим зниженого енергоспоживання що дозволяє зменшити споживання за умови виконання певної команди.

8.3 Периферійні пристрої.

Мікроконтролери мають достатньо великий набір внутрішніх периферійних пристроїв (ПП).Програмування ПП означає керування відповідними бітами у керуючому регістрі. Для цього в нього записується відповідний байт.

Універсальні порти вводу виводу. Мікро контролер має три вбудованих порти: PORT A, PORT B, PORT C з побітовим індивідуальним програмуванням.

Програмування здійснюється керуючими регістрами TRIS які дозволяють вводити і виводити інформацію

Набір таймерів. Ці вузли дозволяють здійснювати відлік певних інтервалів часу або вести підрахунок певних подій. Даний МК має набір з декількох восьми та шістнадцяти розрядних таймерів/ лічильників.

Даний МП має у своєму складі модуль аналого цифрового перетворювача що суттєво спрощує реалізацію системи керування технічним об'єктом. Модуль 8 або 10 розрядного АЦП через комутатор обслуговує 5(8) аналогових входів. В основі модуля лежить АЦП. Вхідний сигнал із зовнішніх входів вибирається за допомогою комутатора (мультиплексора). Результати перетворення зберігаються у відповідних регістрах. Більш детальна інформація про технічні можливості даного МК наведена у відповідній технічній документації.

8.4 Основні типи команд та особливості програмування.

Система команд даного МК складається з 35 команд. Кожна команда складається з 14 бітів і розподіляється на : код операції і один або два операндів. Система команд містить в собі байт –орієнтовані і біт орієнтовані команди (операції з окремими бітами).

Всі команди виконуються протягом командного циклу. Один командний цикл складається з чотирьох тактів. Команди можна об'єднати у наступні умовні групи:

1) Команди пересилання даних і завантаження регістрів.
2) Команди арифметичних та логічних операцій.
3) Команди переходів.
4) Спеціальні команди.

Команди першої групи не мають особливостей і подібні командам МП які були розглянуті вище.

Команди другої групи виконують операції над даними, що заздалегідь завантажені у робочий регістр W. Результат операції можна зберігати у ньому ж або у регістрі, що бере участь в операції. В залежності від результатів операції у регістрі STATUS встановлюються значення бітів переносу – C, десяткового переносу – DC, і нуля – Z.

Переходи у програмі можуть бути як безумовними та і умовними, Наприклад команда INCFSZ f,d означає, що треба додати 1 до вмісту регістра f і якщо результат не дорівнює нулю , то виконати наступну команду вибрану під час виконання поточної, якщо результат 0, то пропустити (у другому циклі виконується команда NOP).

Команда COLL використовується для переходу в підпрограму за адресою, що задається у команді, а команда RETURN – для повернення з підпрограми. Адреса команди, що слідує за командою COLL запам'ятовується у спеціально організованих регістрах, що називаються стеком адресів повернення. В кінці програми RETURN МК витягає зі стеку адресу , встановлює її у лічильник регістру команд і з нього продовжує виконання програми. Таким чином, після

повернення з підпрограми виконання основної програми продовжується з наступної після COLL команди. В МК стек апаратний і використовується тільки для зберігання адреси повернення. До спеціальних команд відноситься команда SLEEP яка призначена для переводу МК у режим зниженого енергоспоживання. Після виконання цієї команди тактовий генератор процесора вимикається і назад у робочий режим процесор можна перевести або за входом скидання або за спрацюванням вартового таймера , або за перериванням.

Є також дві команди зсуву: RRF – зсув праворуч і LRF – зсув ліворуч.

Даний МК має особливості в організації переривань. Вони можуть бути реалізовані від 14 джерел. Організація переривань це досить складна задача розробки програми і потребує певного досвіту.

Укладання тексту програм, його редагування з наступною трансляцією здїйснюється на персональному компютері з використанням спеціального програмного забезпечення. Стосовно даних МП це програма MPLAB яка вільно розповсюджується. Однак треба мати програматор з відвідною програмою Можливо також програмування МК у готовому приладі.

Розглянемо приклад множення двох чисел. Вона має наступну послідовність операторів.

1) Командою CLRW заносимо 0 в регістр W.
2) Командою MOVLW D1 заносимо число D1 в регістр W.
3) Командою MOVLWF REG1 заносимо число D1 в регістр REG1.
4) Командою MOVLW D2 заносимо число D2 в регістр W.
5) Командою MOVLF REG2 заносимо число D2 в регістр REG2.
6) Командою ADDWF REG1,d (d=0 щоб зберегти результат вW) складаємо вміст робочого регістра і REG1
7) Командою DECF REG2,d (d=1 щоб результат зберегти у регістрі 2) зменшуємо значення другого співмножника на 1
8) Командою BTFSS STATUS,Z перевіряємо умову «дорівнює результату 0 або ні». Контролюємо біт Z – ознаку нуля у регістрі стану STATUS

Коли результат Z=1команда переходу пропускається і виконується команда запису результату у регістр 3.

В цілому програмування для МК потребує певних навичок та великої уваги. Доцільно використовувати мову Асемблера тому що вона може знаходити похибки програми на стадії вводу (визначати синтаксичні похибки) що може зменшувати час налагоджування програми.

ЛЕКЦІЯ 9. ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ

9.1 Програмовані логічні контролери

В наш часу системах промислової автоматики знайшли широке застосування так звані програмовані логічні контролери (ПЛК). До виникнення мікропроцесорної техніки контролерами називали багатопозиційні перемикачі для комутації електричних кіл. Такі механічні контролери широко використовували в релейно-контактних схемах керування промисловими об'єктами. Принципові схеми таких систем керування нагадували дробину, стояки якої утворювали два проводи, що підводили електричне живлення, а ступені – послідовне ввімкнені контакти реле, блок-контакти пускачів електродвигунів, контакти натискних кнопок, шляхових вимикачів і котушки електромагнітних апаратів (реле, пускачів, електромагнітів та інших пристроїв). Деякі контакти такої ступені могли бути зашунтовані контактами, ввімкненими до них паралельно. Коли низка таких контактів утворювала електричне коло, котушка збуджувалася і, якщо це була обмотка реле чи пускача, створювала перемикання контактів в інших ступенях схеми керування або у силових колах електроустаткування.

Програмовані ПЛК на базі мікропроцесорної техніки спочатку використовувалися для заміни таких систем з релейною логікою. Їх перевагою були меньки габарити і гнучкість, оскільки релейна автоматика вимагала жорсткого монтажу і будь-яка зміна в логіці функціонування релейної автоматики приводила до необхідності зміни всього монтажу, що було економічно невигідно. Але завдяки можливостям мікропроцесорної техніки ПЛК знайшли застосування в індустрії не тільки для перемикань електричних кіл, але й для розв'язку інших прикладних задач , пов'язаних з обчисленнями, обробкою та порівнянням аналогових сигналів. На рис. 9.1. показано спрощену блок-схему ПЛК з модулями дискретних (тобто таких, що можуть мати лише два стани: ввімкнено або вимкнено) вводів/виводів, інтегрованих до блоку центрального процесора. Характерно, що виробники ПЛК виготовляють модулі вводу/виводу, які можуть бути інтегровані до єдиного з центральним процесорним пристроєм корпусу, або виготовлені у вигляді окремих модулів, які монтують до стійки (стелажу) разом з іншими блоками контролера, причому стойка облаштована електричними рознімами і при вставленні до стійки модулів ПЛК відбувається одночасно й їх електричне з'єднання.

Центральний процесор зчитує з модуля вводу інформацію про стани контактів зовнішніх пристроїв, виконує програму користувача, записану до його внутрішнього пристрою пам'яті і посилає відповідні команди керування зовнішніми пристроями через модуль виводу. Для роботи центрального процесора та сполучених з ним мікросхем потрібна низьковольтна постійна напруга.

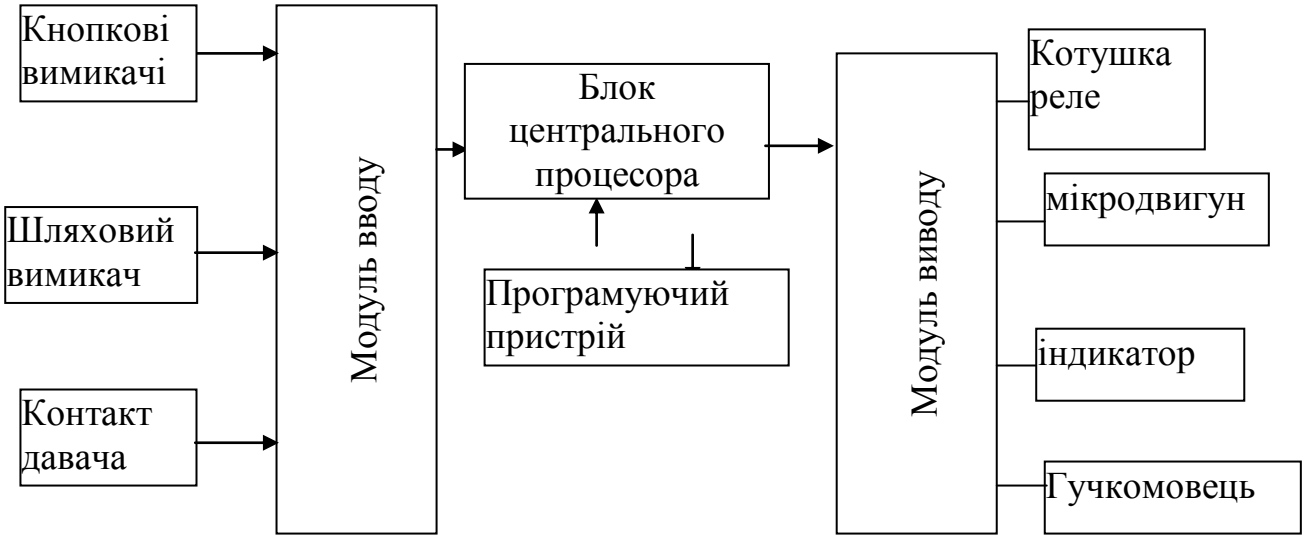

Рис 9.1 – Блок-схема програмованого логічного контролера.

Якщо ж мікросхеми виготовлені з використанням так званої комплементарної метал-окисел-напівпровідник (КМОН) логіки - напруга живлення може бути у діапазоні 3-18 В. Цю напругу забезпечує блок живлення, розміщений у модулі центрального Центральний процесор має розніми для сполучення його кабелями з іншими модулями ПЛК, а також операційні ключі. Типовими є позиції операційного перемикача, який дозволяє обрати для ПЛК один з режимів: OFF – ПЛК вимкнено; RUN – дозвіл виконувати програму користувача без внесення до неї змін; PROGRAM - дозвіл створювати, міняти та видаляти програми користувача, всі виводи відключені.

Модулі вводу/виводу, які часто позначають скорочено як I/O-модулі (від початкових літер слів input – ввід та output – вивід), формують інтерфейс, яким поле зовнішніх пристроїв зв’язане з контролером. Пристрої поля зовнішніх дискретних вводів – це контакти натискних кнопок, шляхових вимикачів, здавачів та перемикачів інших типів, які є жорстко змонтовані до терміналів вхідних модулів. Реальні пристрої виводу – це двигуни малої потужності, пускачі двигунів, котушки електромагнітних реле, соленоїдних клапанів та світлові індикатори, жорстко змонтовані з терміналами модулів виводу. Назви «поле зовнішніх пристроїв» та «реальні пристрої» вжиті для того, щоб відрізняти ці фізично існуючи пристрої від їх логічних зображень у внутрішній програмі користувача, яка дублює їх функції.

Модулі вводу з’єднують сигнали від керованої машини чи технологічного процесу, які можуть мати рівень, наприклад, 120 В змінного струму і перетворюють їх у сигнали постійного струму з рівнем напруги, наприклад, 5 В, який може бути використаний контролером. Модулі інтерфейсу виводу перетворюють сигнали контролера з рівнем 5 В постійної напруги у зовнішні сигнали, наприклад 120 В змінного струму, які використовують для керування

машиною або процесом. Всі вводи і виводи можна поділити на цифрові або дискретні та аналогові. Цифрові вводи та виводи функціонують завдяки дискретним змінам станів сигналів або їх рівня. Аналогові вводи – виводи змінюють електричну напругу неперервно в деякому заданому діапазоні.

9.2 Особливості використання контролерів

Особливості використання ПЛК. Конструкція ПЛК суттєво відрізняється від конструкції звичайної обчислювальної машини. На лицевій панелі розташований дисплей та декілька функціональних клавіш для програмування, редагування і відображення інформації. Силовий блок, що пов'язаний безпосередньо з об’єктом керування може бути розташований окремо. Основними технічними показниками ПЛК є кількість та технічні характеристики каналів вводу/виводу та види сигналів, що можуть оброблятися. При розв’язуванні задачі з використанням ПЛК повинні використовуватися зовнішні пристрої (контакти давача тиску, давача температури, натискної кнопки). Ці пристрої жорстко змонтовані до виводів модуля вводу відповідно формату адресації виробника ПЛК, який згідно з заданою йому програмою здійснює логічне керування тим самим процесом шляхом заміни кожного реального контакту внутрішнім логічним контактам. Типова ступінчаста логічна схема керування показана на рис. 9.2. Розташування логічних контактів є подібним розташуванню жорстко змонтованої релейної схеми. Індивідуальні символи контактів представляють собою команди програми ПЛК, а числа біля них представляють адресу команди.. Коли процесор програмують, ці команди вводять одну за одною до пам’яті контролера з клавіатури терміналу програмувального пристрою (пульта оператора). Введена програма зберігається у пам’яті контролера.

Для запуску програми ПЛК переводять до режиму ВИКОНАННЯ (RUN) , який можна ще назвати режимом операційного циклу. Протягом виконання кожного операційного циклу контролер досліджує стан пристроїв вводу даних, виконує програму користувача і змінює стани виводів відповідно до отриманих станів вводів. Кожну ступінь логічної діаграми можна розглядати як набір нормально відкритих (НВ) контактів. Якщо всі логічні контакти закриті, показана на рис.9.2 логічна котушка 011 збуджується і встановлює на виводі 011 логічну одиницю

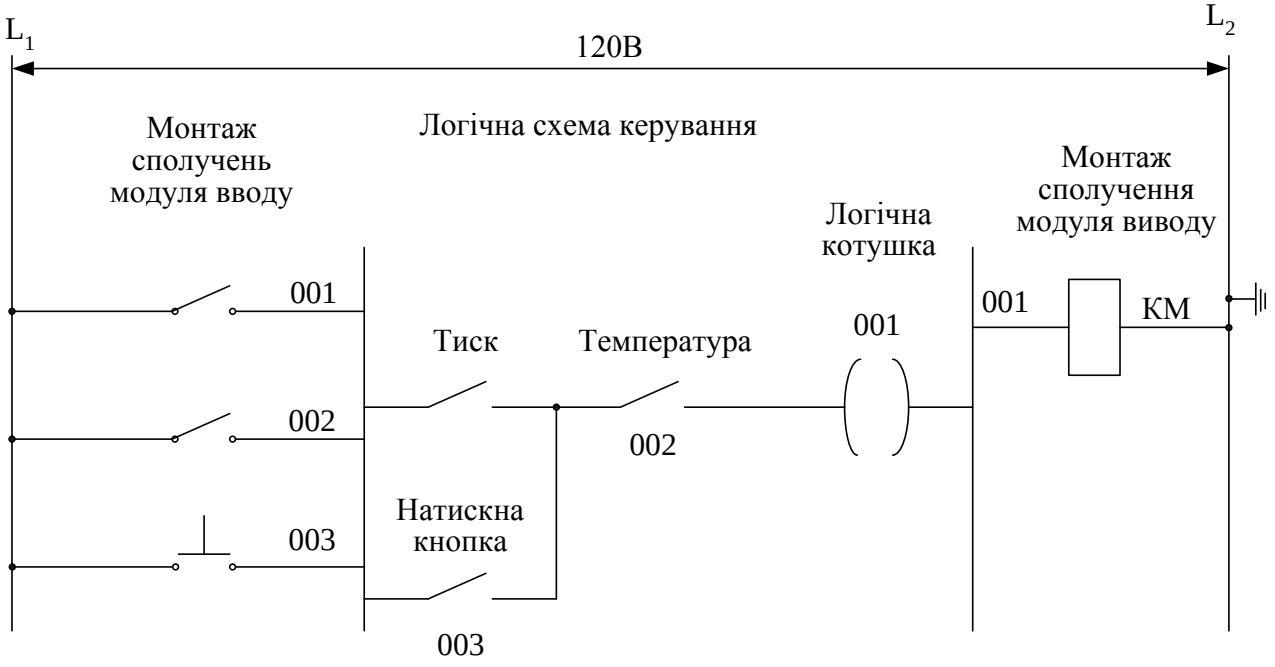

Рис 9.2 – Схема вмикання приладу від ПЛК

У показаній ступінчастій схемі керування котушка 011 збуджена, коли закриті контакти 001 та 002 або коли закритий контакт 003. Виконання принаймні однієї з цих умов забезпечує неперервний шлях зліва направо в межах даної ступені логічної схеми керування, що призводить до вмикання (збудження) логічної котушки і вмикання магнітного пускача КМ.

9.3 Програмування контролерів

Програмування контролерів. Кожний ПЛК постачається із власним програмним забезпеченням різноманітного рівня складності. Програмування можливе з використанням:

- принципу релейно-контактних схем;
- діаграм функціональних блоків;
- спеціальної мови за типом асемблера.

Розглянемо більш детально використання програмування ПЛК на основі використання релейно-контактних схем. Даний підхід оснований на тому, що упер важній більшості обладнання яке використовується у харчовій промисловості логіку роботи реалізують шляхом використання релейно-контактних схем. Це пояснюється тим, що робота цього обладнання відбувається за відносно простими алгоритмами і використання і використання універсальних комп’ютерних систем керування просто недоцільне з точки зору економіки . К той же час використання ПЛК може бути оправдано як з економічної точки зору так і з складністю задач, що повинні розв’язуватися. Розглянемо основні підходи щодо програмування ПЛК.

Як і при програмуванні будь-якого мікропроцесора для програмування ПЛК слід планувати організацію його пам’яті. Рисунок 9.3 показує типову

організацію пам’яті ПЛК відому як карту пам’яті. Індивідуальні розділи карти пам’яті, їх порядок і довжина можуть бути фіксованими або змінними в залежності від виробника і конкретної моделі ПЛК.

Умовні адреси програми

0	Програма користувача
400	Зображення станів вводів
600	Зображення станів виводів
900	Числова та надоперативна пам’ять
1050	Стан таймерів
1300	Стан лічильників
… .	Інші функції

Рис 9.3 – Організація пам’яті ПЛК

В цілому карту пам’яті можна розподілити на три поля: поле таблиці даних, в якому фіксуються розміщення та стан вводів/виводів, реле, таймерів, лічильників; поле програми користувача і поле допоміжної пам’яті, в якому зберігаються програми, необхідні для функціонування контролера (подібно до програми «Монітор».

Розділ пам’яті «програма користувача» - це місце внесення та зберігання програмованої логічної схеми керування, подібної до схеми релейно-контактного керування. Програма користувача містить логіку, яка виконує операції керування зовнішніми пристроями. Ця логіка складається з команд у форматі логічної драбини,
подібної до логіки драбини релейно-контактних схем.

Таблиця даних зберігає необхідну для виконання програми користувача інформацію про стан вводу/виводу, стан лічильників, таймерів, а також значення вихідних чи проміжних даних. Дані стану – це інформація типу «ввімкнути/вимкнути» (ON/OFF), представлена у вигляді логічних одиниць (1) або логічних нулів (0), які виставляються у відповідних конкретних бітах слова виводу. При виставленні на певному виході логічної одиниці процесор

встановлює на вході відповідного модуля виводу напругу високого рівня наближену до значення своєї робочої напруги (наприклад напругу біля 5В), а при встановленні логічного нуля – напругу низького рівня, яка звичайно не перевищує 0,3-0,8 В, чим забезпечує активний або пасивний стан вихідного силового пристрою (наприклад симистору).

Таблиця даних може бути умовно розподілена відповідно до типу збережуваної інформації на три зони: зону зображень даних вводів, зону зображень даних виводів і зону зберігання даних лічильника та таймера. Таблиця зображень даних вводів зберігає інформацію про стан дискретних вводів модуля інтерфейсу вводу.

Програма користувача виконується циклічно по трьох типових етапах: читання стану вводів – аналіз стану вводів згідно із задавлю програмою – корекція стану виводів – і далі знову читання стану вводів.

Під мовою програмування ПЛК розуміють метод повідомлення користувачем інформації ПЛК Логіка релейно-контактної дробини була першою і найбільш популярною мовою, що застосовується у ПЛК. Нині більшість ПЛК використовує цю мову.

Логіка релейно-контактної дробини – це графічна мова програмування, яка дозволяє імітувати вигляд змонтованої релейної схеми. Це полегшує експлуатацію ПЛК технічним персоналом і не потребує спеціальних знань щодо програмування. Релейну схему можна відтворити на дисплеї де вона стає наочною що полегшує роботу оператора. Для транслювання логіки релейно-контактних схем до логіки контактної символіки використовують три базових символи які являють базисний набір команд для виконання функцій подібно функціям реле.

Таблиця 9.1

Базисні символьні команди ПЛК

EXAMINE ON (перевірити на умову “ввімкнено»)	] [	XIC
EXAMINE OFF (перевірити на умову «вимкнено»)	]/[	XIO
OUTPUT ENERGIZE Збудити вивід	( )	OTE

Команди EXAMINE ON та EXAMINE OFF являють собою будь який ввід логічного керування. Вводом може бути перемикач або кнопка, Адреса обох команд – бітового рівня. Біт стану дорівнює 1 (ON) або 0 (OFF). Для команди EXAMINE ON стан біту досліджується на умову рівності його одиниці, а для команди EXAMINE OFF на умову рівності цього біта нулю.

Основна функція програми що представлена логічною драбиною керування виводами на підставі інформації про стани вхідних умов

Таймери та лічильники у ПЛК виглядають як команди виводу, Їх призначення – створення програмованої затримки часу. В релейній автоматиці для цього використовують різні реле часу – електромагнітні, пневматичні,

електронні та інш.У МПС таймери виконують у у вигляді окремих ВІС які мають 1-3 лічильника, в які можна програмним шляхом завести інформацію. При надходження зовнішніх сигналів стан лічильника змінюється і коли він стане рівним нулю, таймер видає сигнал на вихід.

Команди обробки даних надають ПЛК деякі властивості обчислювальної системи які суттєво розширюють можливості релейно-контактних схем. Кожний ПЛК може виконувати передачу даних, оперування даними з використанням математичних функцій, перетворення даних та їх порівняння. Команди передачі даних здійснюють передачу інформації з одного регістру до другого і можуть призначити будь-яку адресу у пам'яті. Команди порівняння даних порівнюють дані, що зберігаються у двох словах і приймають рішення відповідно за програмою.

Команди тестового обмеження порівнюють тестові значення із значеннями обмежень і при необхідності видають відповідний сигнал.

Вище наданий обмежений перелік команд ПЛК що дає деяку уяву про принципи програмування ПЛК та їх особливості. З ростом технічного прогресу можливості ПЛК щодо виконання різноманітних функцій керування розширюються, Також змінюється підхід і до програмування його роботі з метою спрощення діалогу між ПЛК і користувачем. Суттєво спрощує програмування ПЛК шляхом використання рівнянь що дописують роботу приладу логічними рівняннями. При такому підході не треба безпосередньо знати конкретні мови програмування. Треба тільки вміти описати роботу приладу стандартними рівняннями. Можливі і інші підходи які будуються на постійному розвитку систем програмування.

9.4 Застосування ПЛК у системах автоматизації технологічних об'єктів.

Застосування ПЛК у системах автоматизації технологічних об'єктів. Сучасною тенденцією побудови систем автоматизації технологічних об'єктів є широке впровадження персональних комп'ютерів (ПК). Вартість сучасних МПС порівняно невелика і тому доцільно їх використання як на рівні релейно-контактних схем так і для реалізації відносно складних законів управління. Сучасні ПЛК мають комунікаційні засоби у вигляді портів RS-232 (послідовний порти) що дозволяє їх з'єднання з більш потужною обчислювальною машиною. Контролери можна об'єднувати у мережу і підключати до загальної ЕОМ вищого рівня. При цьому будуть вирішуватися задачі як технічного спрямування так і зодчі організаційного управління (задачі автоматизованого збору інформації). Таким чином можна створювати ієрархічні системи керування не тільки технологічним обладнанням, а також створювати передумови для розробки системи управління виробництвом у цілому.

Такі підходи є актуальними і для підприємств харчової та переробної промисловості котрі мають справу при виробництві продукції з великою кількістю факторів, що впливають на якість продукції. Поширення рівня автоматизації виробництва поступова створює умови для більш повного

використання автоматизованих систем керування і подальший перехід на технології де людини буде тільки керувати автоматичними приладами.

ЛІТЕРАТУРА

1. Мілих В.І. Шавьолкін О.О. Електротехніка, електроніка та мікропроцесорна техніка: Підручник. За ред..В.І. Мілиха. 2-е вид – Каравела, 2008. –688 с

2. Основы микропроцессорной техники : учебное пособие / Ю. В. Новиков, П. К. Скоробогатов. — 4-е изд., испр. — М. : Интернет-Университет Информационных Технолотй; БИНОМ. Лаборатория знаний, 2009. - 357 с.

3. Будіщев М.С. Електротехніка, електроніка та мікропроцесорна техніка. - Львів: Афіша, 2001. С.423.

4. Суэмацу Ё. Микрокомпьютерные системы управления,/ Пер. с яп.; под ред. Ёсифуми Амэмия. – М.: Издательский дом «Додека-XXI», 2002. – 256с.

5. Сато Ю. Обработка сигналов, /Пер. с яп..; под.ред. .Ёсифуми Амэмия. – М.: Издательский дом «Додека-XXI», 2002. – 176с.

6. Торяник О.І. Електрогтехніка та основи електроніки. [Текст] :навчальний посібник / О.І. Торянрик, О.Г. Дьяков, М.А. Чеканов – Харків, : ХДУХТ, 2009. – 164 с.

7. Электротехника и электроника в экспериментах и упражнениях: Практикум на Electronics Workbench: в 2 т./Под общей ред. Д.И. Панфилова – Т.2:Электроника. – М.: ДОДЕКА,2000. – 288 с.

8. Поначевний Б.Ш., Свергун Ю.Ф. Загальна електротехніка: теорія і практикум,–К.: Каравела, 2003,–С 400.

9.Прянишников В.А. Электроника, Полный курс лекций, Санкт–Петербург, «КОРОНА принт », 2004. С. 415.

Навчальне видання

Укладачі:
ТОРЯНИК Олександр Іванович
ДЬЯКОВ Олександр Георгійович
ЧЕКАНОВ Микола Анатолійович
ІШТВАН Єгор Олексійович

МІКРОПРОЦЕСОРНА ТЕХНІКА
Конспект лекцій

Для студентів напрямку підготовки 6.050502 “Інженерна механіка”
(скорочена форма навчання)

Підп. до друку р. Формат 60х84 1/16. Папір офсет. Друк офс.
Умов. друк. арк. 2,7. Тираж прим. Зам. №

Видавець і виготовлювач
Харківський державний університет харчування та торгівлі.
61051. Харків-51, вул. Клочківська, 333.
Свідоцтво суб’єкта видавничої справи ДК №4417 від 10.10.2012 р.