

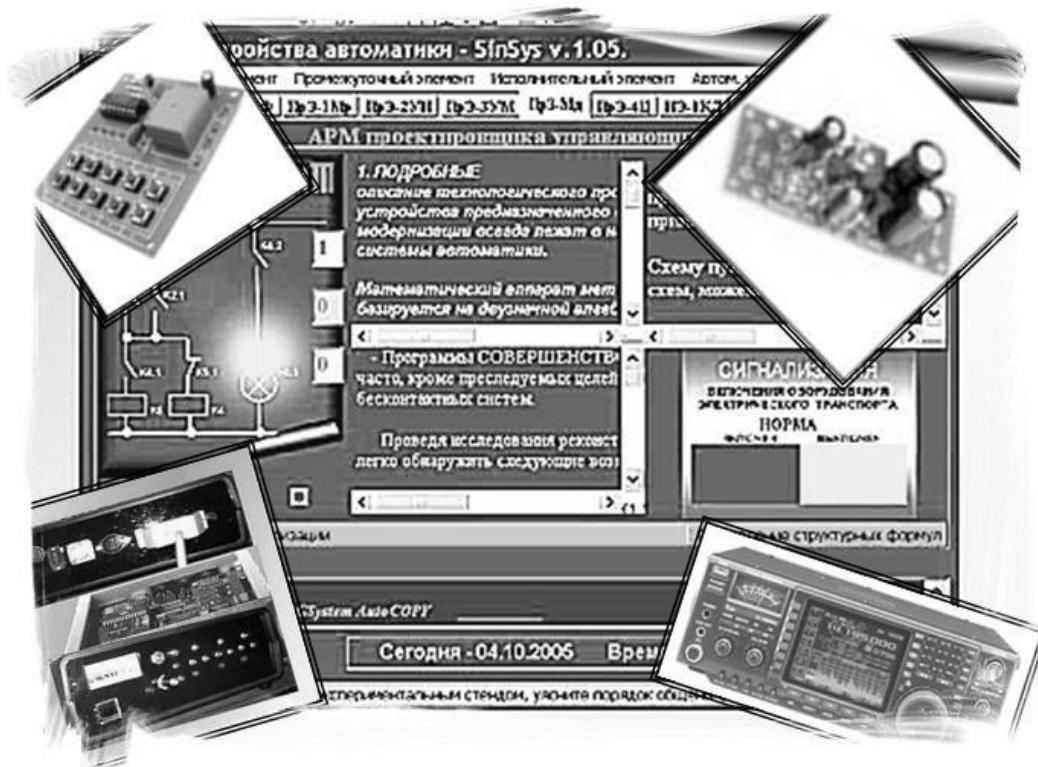
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА

С. М. Єсаулов, О. Ф. Бабічева, Р. В. Воронов

МІКРОПРОЦЕСОРНІ ПРИСТРОЇ В СИСТЕМАХ  
ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

КОНСПЕКТ ЛЕКЦІЙ

*(для здобувачів другого (магістерського) рівня вищої освіти денної та заочної форм навчання спеціальності 141 – Електроенергетика, електротехніка та електромеханіка)*



Харків  
ХНУМГ ім. О. М. Бекетова  
2023

**Єсаулов С. М.** Мікропроцесорні пристрої в системах електричної інженерії : конспект лекцій для здобувачів другого (магістерського) рівня вищої освіти денної та заочної форм навчання спеціальності 141 – Електроенергетика, електротехніка та електромеханіка / С. М. Єсаулов, О. Ф. Бабічева, Р. В. Воронов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2023. – 140 с.

Автори:

канд. техн. наук, доц. С. М. Єсаулов,  
канд. техн. наук, доц. О. Ф. Бабічева,  
канд. техн. наук, ст. викл. Р. В. Воронов

Рецензент

**А. К. Бабіченко**, кандидат технічних наук, професор кафедри автоматизації технологічних систем та екологічного моніторингу Харківського національного технічного університету «ХП»

*Рекомендовано кафедрою електричного транспорту,  
протокол № 7 від 13.12.2021.*

© С. М. Єсаулов, О. Ф. Бабічева, Р. В. Воронов, 2023  
© ХНУМГ ім. О. М. Бекетова, 2023

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ.....	6
ВСТУП.....	7
1 ОСНОВНІ РОЗДІЛИ КУРСУ.....	8
2 ІНФОРМАЦІЙНІ ОСНОВИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ.....	9
2.1 Історія виникнення мікропроцесорних пристроїв.....	9
2.2 Поняття інформації, види сигналів.....	9
2.3 Кодування інформації.....	11
2.4 Контрольні питання.....	15
3 ЛОГІЧНІ ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ.....	17
3.1 Булева алгебра.....	17
3.2 Поняття елементів мікропроцесорних пристроїв.....	19
3.3 Контрольні питання.....	20
4 ЕЛЕМЕНТНА БАЗА МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ.....	20
4.1 Система логічних елементів.....	21
4.1.1 Елемент АБО.....	21
4.1.2 Елемент І.....	22
4.1.3 Елемент НІ.....	23
4.1.4 Тригери.....	24
4.2 Послідовні (накопичувальні) пристрої.....	29
4.2.1 Лічильники й дільники.....	29
4.2.2 Регістри.....	31
4.3 Комбінаційні функціональні схеми.....	32
4.3.1 Дешифратори.....	32
4.3.2 Шифратори.....	35
4.3.3 Мультиплексори.....	36
4.3.4 Демультиплексори.....	38
4.3.5 Компаратори.....	38
4.3.6 Підсилювачі.....	41
4.4 Контрольні питання.....	43
5 ОСНОВИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ АВТОМАТИКИ.....	44
5.1 Арифметичні основи мікропроцесорних пристроїв.....	44
5.1.1 Принципи побудови систем числення.....	44
5.1.2 Переведення чисел з однієї системи в іншу.....	45
5.2 Арифметичні операції.....	46
5.3 Контрольні питання.....	47
6 МІКРОПРОЦЕСОРНА АВТОМАТИКА НА ТЕХНОЛОГІЧНИХ ОБ'ЄКТАХ.....	47
6.1 Архітектура і структура мікропроцесора системи керування.....	47
6.2 Пристрої на базі комп'ютерів і мікро-ЕОМ.....	49

6.3	Структура мікроконтролерів системи керування.....	51
6.4	Принцип програмного керування.....	52
6.5	Контрольні питання.....	54
7	<b>ЕЛЕМЕНТИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ АВТОМАТИКИ ТА ЇХ ХАРАКТЕРИСТИКИ.....</b>	<b>55</b>
7.1	Процесори.....	55
7.2	Мікропроцесори.....	56
7.3	Архітектура мікропроцесорів.....	57
7.4	Мікропроцесорні комплекти.....	58
7.5	Архітектура однокристалного мікроконтролера.....	60
7.6	Функціональні структури мікропроцесорних пристроїв.....	60
7.7	Адресний простір мікропроцесорних пристроїв і мікроконтролерів.....	62
7.8	Пам'ять внутрішня і зовнішня.....	62
7.9	Контрольні питання.....	65
8	<b>ПРОГРАМОВАНА АВТОМАТИКА НА БАЗІ МІКРОПРОЦЕСОРІВ.....</b>	<b>65</b>
8.1	Функціонування мікропроцесорів у часі.....	65
8.2	Формати команд і даних.....	66
8.3	Мова асемблеру.....	67
8.4	Призначення програм і етапи їх створення.....	68
8.5	Структура програмного забезпечення.....	69
8.6	Алгоритми і правила їх складання.....	71
8.7	Система команд мікропроцесорів.....	72
8.8	Способи адресації.....	75
8.9	Контрольні питання.....	75
9	<b>ПРОГРАМУВАННЯ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ НА МОВАХ ВИСОКОГО РІВНЯ.....</b>	<b>76</b>
9.1	Особливості реалізації мов високого рівня.....	76
9.2	Програмування на мові СІ.....	77
9.3	Системи проектування програм.....	79
9.4	Контрольні питання.....	79
10	<b>ПРИСТРОЇ ЗВ'ЯЗКУ З ОБ'ЄКТОМ КЕРУВАННЯ.....</b>	<b>80</b>
10.1	Характеристики мікропроцесорних систем.....	80
10.2	Загальна характеристика інтерфейсів.....	81
10.3	Підсистеми перетворення сигналів в мікропроцесорних пристроях.....	82
10.4	Послідовний і паралельний інтерфейси мікропроцесорних пристроїв.....	89
10.4.1	Загальна характеристика інтерфейсів.....	89
10.4.2	Програмовний послідовний інтерфейс.....	89

10.4.3 Програмовний паралельний інтерфейс.....	90
10.4.4 Послідовна універсальна шина (USB).....	91
10.5 Контрольні питання.....	91
11 ПРИКЛАДИ СКЛАДАННЯ ПРОГРАМ.....	92
11.1 Програма на асемблері.....	92
11.2 Програма на BASIC.....	94
11.3 Програма на C.....	95
11.4 Контрольні питання.....	95
12 ПРОГРАМОВАНІ ЗАСОБИ АВТОМАТИКИ.....	96
12.1 Автоматична система контролю.....	96
12.2 Позиційний одноканальний керуючий пристрій.....	97
12.3 Восьмиканальний керуючий пристрій.....	99
12.4 Автоматична система повторного включення тягової підстанції.....	101
12.5 Автоматизоване робоче місце оператора мийно-прибирального комплексу депо.....	104
12.6 Мікроконтролер діагностики устаткування.....	106
12.7 Контрольні питання.....	108
13 ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ.....	109
13.1 Характеристики експлуатаційної надійності.....	109
13.2 Методи підвищення надійності мікропроцесорних керуючих систем автоматики.....	110
13.3 Організація технічного обслуговування мікропроцесорних пристроїв.....	113
13.4 Контрольні питання.....	115
14 ПРИКЛАДИ.....	115
14.1 Системи числення.....	115
14.2 Синтез безконтактних логічних пристроїв.....	118
14.3 Кодування дискретної інформації.....	122
14.4 Логічні й арифметичні дії в мікропроцесорах.....	123
14.5 Розробка блок-схем алгоритмів.....	125
14.6 Програмування пристроїв автоматики.....	125
СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	129
ДОДАТКИ.....	130

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЛБ – арифметико-логічний блок  
АЛП – арифметико-логічний пристрій  
АПВ – автоматична система повторного включення  
АПК – адресний простір МК  
АПП – адресний простір пам'яті  
АПРФ – адресний простір реєстрового файлу  
АРМ – автоматизоване робоче місце  
АЦП – аналогово-цифровий перетворювач  
БК – блок контролю  
БІФ – блоку інтерфейсу  
БН – блок нормалізації  
ВЕ – виконавчі елементи  
ВІС – великі інтегральні схеми  
ДДНФ – досконала диз'юнктивна нормальна форма  
ДКНФ – досконала кон'юнктивна нормальна форма  
ЗЗП – зовнішні запам'ятовуючі пристрої  
ЛЕ – логічний елемент  
ЛЗЗ – лінії записування-зчитування  
МВР – мова високого рівня  
МК – мікроконтролери  
МПК – мікропроцесорні комплекти  
МПЛ – мультиплексор  
МПП – мікропроцесорні пристрої  
МПС – мікропроцесорна система  
МСЗ – максимальний струмовий захист  
ОП – операційні підсилювачі  
Оп – оперативна пам'ять  
ПВв – пристрої вводу інформації  
ПВив – пристрої виводу інформації  
ПЕ – приймальний елемент  
ПК – пристрій керування  
ПП – периферійні (зовнішні) пристрої  
ППІ – програмовні послідовні інтерфейси  
РЗП – реєстр загального призначення  
САПР – система автоматизованого проектування  
ТО – технологічний об'єкт  
ТП – тягова підстанція  
ФКС – формувач керуючого сигналу  
ЦАП – цифро-аналогового перетворювач  
ШД – шини даних

## ВСТУП

Цей конспект лекцій призначений для студентів усіх форм навчання дисципліни «Мікропроцесорні пристрої в системах електричної інженерії».

Збільшення технологічних можливостей і здешевлення виробів напівпровідникового виробництва привело до того, що мікропроцесорні пристрої (далі – МПП) стають найбільш популярними технічними рішеннями в системах автоматизації різного призначення. Інтеграція в одній мікросхемі великого числа функцій, пов'язаних з прийомом, обробкою, зберіганням даних і керуванням, сприяла розвитку нових типів МПП – мікроконтролерів (далі – МК) на одному кристалі. МК стали невід'ємною частиною нашого життя. Вони популярні на транспорті, різних технологічних об'єктах (далі – ТО) і особливо в побутовій техніці. З багатьма такими пристроями ми спілкуємося щодня. Малий розмір, надійність, низька ціна і висока продуктивність є очевидними достоїнствами МПП, які постійно удосконалюються. Області застосування МПП не мають меж.

Вивчення цифрової техніки і МПП, вживаної на електричному транспорті, має специфічні особливості. Пов'язано це з використанням не тільки сучасного електротехнічного устаткування, але й великого числа функціонуючих пристроїв на контактній-релейній базі, що відзначаються моральним зносом. У зв'язку з цим необхідною є підготовка спеціалістів міського транспорту, здатних організувати правильну експлуатацію цієї техніки.

*Мета курсу* – вивчення основ створення і використання мікропроцесорних систем для керування технологічними процесами і обладнанням міського електричного транспорту.

У результаті вивчення курсу студент повинен знати периферійне устаткування, структуру МПП і МК, вміти експлуатувати обладнання з системами керування на базі МПП, розробляти алгоритми, мати навички програмування завдань керування об'єктами транспорту і знати перспективні напрямки розвитку мікропроцесорної техніки.

## 1 ОСНОВНІ РОЗДІЛИ КУРСУ

**Тема 1.** Інформаційні основи мікропроцесорних пристроїв. Історія виникнення мікропроцесорних пристроїв. Поняття інформації, види сигналів. Кодування інформації.

**Тема 2.** Логічні основи мікропроцесорної техніки. Булева алгебра. Поняття елементів мікропроцесорних пристроїв.

**Тема 3.** Елементна база мікропроцесорних пристроїв. Система логічних елементів. Елементи ЧИ, І, НІ. Тригери. Послідовні пристрої: лічильники й дільники, регістри. Комбінаційні функціональні схеми: дешифратори, шифратори, мультиплексори, демультимплексори, компаратори, підсилювачі.

**Тема 4.** Основи мікропроцесорних пристроїв автоматики. Арифметичні основи МПП. Принципи побудови систем числення. Переведення чисел з однієї системи в іншу. Арифметичні операції.

**Тема 5.** Мікропроцесорна автоматика на технологічних об'єктах. Архітектура і структура МП систем керування. Пристрої на базі комп'ютерів і мікро-ЕОМ. Структура мікроконтролера системи керування. Принцип програмного керування.

**Тема 6.** Елементи МПП автоматики та її характеристики. Процесори, мікропроцесори, мікропроцесорні комплекти. Архітектура мікропроцесорів. Архітектура однокристального мікроконтролера. Універсальні й спеціалізовані мікропроцесори. Адресний простір МП і МК. Пам'ять внутрішня і зовнішня.

**Тема 7.** Програмована автоматика на базі МП. Функціонування МП у часі. Формати команд і даних. Мова асемблера. Призначення програм і етапи їх створення. Структура програмного забезпечення. Алгоритми і правила їх вкладання. Система команд МП і способи адресації.

**Тема 8.** Програмування МПП на мовах високого рівня. Особливості реалізації мов високого рівня. Програмування на мові С. Системи проектування програм. Приклади складання програм. Програми на Basic і С.

**Тема 9.** Пристрої зв'язку з об'єктом керування. Характеристики мікропроцесорних систем. Загальна характеристика інтерфейсів. Підсистеми перетворення сигналів у МПП. Послідовний і паралельний інтерфейси. Послідовна універсальна шина (USB)

**Тема 10.** Приклади складання програм на асемблері, Basic, С.

**Тема 11.** Приклади застосування мікропроцесорних пристроїв на об'єктах міського електричного транспорту.

**Тема 12.** Забезпечення надійності мікропроцесорних пристроїв. Характеристики експлуатаційної надійності. Методи підвищення надійності мікропроцесорних керуючих систем. Організація технічного обслуговування.



## 2 ІНФОРМАЦІЙНІ ОСНОВИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ

### 2.1 Історія виникнення мікропроцесорних пристроїв

Найголовнішим елементом в МПП є *мікропроцесо* електронна схема, що виконує всі обчислення і обробку інформації. Швидкість його роботи багато в чому визначає швидкодію самого технічного пристрою. Початок появи мікропроцесорних пристроїв пов'язаний з мікросхемою Intel 4004 – першим мікропроцесором, створеним в 1971 р. талановитим винахідником Тедом Хоффом.

Потреба в автоматизації обробки даних, зокрема обчислень, виникла давно. У даний час індустрія виробництва мікропроцесорних пристроїв і програмного забезпечення є однією з найбільш важливих сфер економіки всіх розвинених країн. Причини використання мікропроцесорних пристроїв на транспорті:

- невисока вартість мікропроцесорних пристроїв;
- простота використання на локальних і рухомих об'єктах;
- постійно зростаючі можливості з переробки, зберігання і передачі інформації;
- висока надійність, простота ремонту і експлуатації;
- можливість швидкої адаптації до технологічних об'єктів будь-якої складності шляхом заміни програмного забезпечення та ін.

Докладну інформацію див. в Windows - додатку SinSys (меню: вхід, Історія МП техніки).

### 2.2 Поняття інформації, види сигналів

*Інформація* – це сукупність знань, що характеризують стан контрольованого процесу або об'єкта. Вона може передаватися, розподілятися, перетворюватися, зберігатися або безпосередньо використовуватися.

*Сигнал* – це фізичний процес, що відображає інформаційне повідомлення з похибкою, яка не перевищує заданого значення. Сигнал повинен бути придатним для передачі на відстань та його обробки.

Оскільки повідомлення несе інформацію про зміни стану технологічного об'єкта керування, то сигнал повинен мати один або декілька параметрів, що піддаються відповідній зміні.

Якщо технологічний об'єкт керування має тільки одну керовану величину ( $Y$ ) (рис. 2.1), його називають *одновимірним*. Реально ТО на електричному транспорті мають велику кількість і керованих ординат, і відповідне їм число, керуючих дій ( $X$ ), – це *багатовимірні* об'єкти.

До особливостей багатовимірних ТО на електричному транспорті слід віднести застосування різних датчиків-джерел інформації про контрольовані величини (струм, напругу, температуру та ін.).

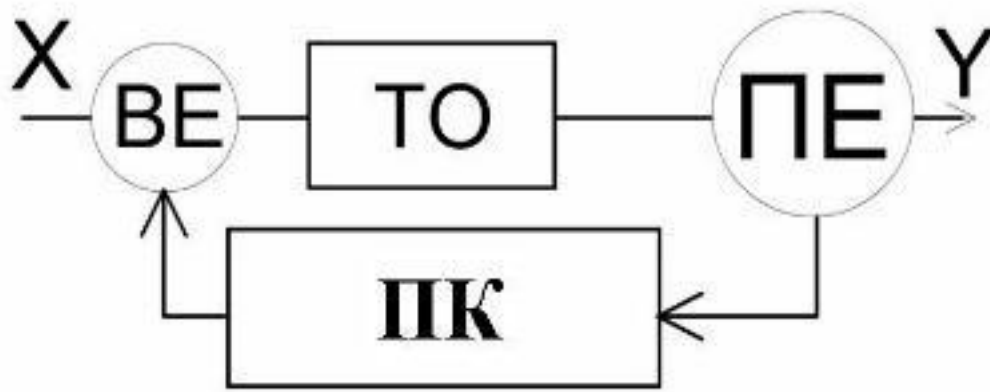


Рисунок 2.1 – Структурна схема одновимірного керованого об’єкта:  
*ТО* – технологічний об’єкт; *ПК* – пристрій керування;  
*ПЕ* – приймальний елемент; *ВЕ* – виконавчий елемент;  
*X* – керуюча величина; *Y* – керована величина

Часто контрольовані величини перетворюються в пропорційні електричні сигнали до їх передачі на різні відстані. Інформаційне повідомлення завжди пов’язане з джерелом і приймачем інформації, сполученими каналом передачі (рис. 2.2).



Рисунок 2.2 – Інформаційна модель каналу передачі

Джерелом і приймачем інформації можуть бути технічні пристрої (ПЕ, МПП, індикатори та ін.). *Каналом передачі* можуть бути сукупність пристроїв, що мають один вхід і один вихід, й призначені для передачі інформації на відстані. Повідомлення можуть мати різноманітні форми: звук, текст, зображення, електрична напруга від ПЕ.

В електричному сигналі можуть модулюватися напруга, сила струму, частота, фаза. Сигнал, в якому нічого не змінюється, не може нести інформацію. При всіх перетвореннях сигналу інформація, що міститься в ньому, мусить зберігатися.

Залежно від характеру зміни сигналу в часі розрізняють сигнали випадкові й детерміновані. *Випадкові* – це ті, поведінку яких у годині неможливо передбачати. Як правило, у пристроях автоматики це перешкоди.

*Детерміновані сигнали* – це ті, поведінку яких у часі можна передбачити заздалегідь із заданою похибкою. Детермінований сигнал може бути заданий математичною моделлю. Наприклад, для генератора синусоїдальної напруги  $U(t)$ :

$$U(t) = U_m \sin(\omega_0 t + \varphi_0), \quad (2.1)$$

де  $U_m$  – амплітуда коливань;

$\omega_0$  – колова частота;

$\varphi_0$  – початкова фаза;

$t$  – час.

*Аналогові* сигнали передають повідомлення у деякому безперервному інтервалі значень фізичного параметра, що кількісно характеризує стан джерела інформації. Як аналогові ПЕ можуть служити терморезистори, фоторезистори, трансформатори струму і напруги та ін.

*Логічні (цифрові, дискретні)* сигнали передають повідомлення тільки про два стани джерела інформації – «ТАК–НІ», «УВІКНЕНО–ВИМКНЕНО». Як логічні ПЕ можуть служити контактні елементи, наприклад реле, які виконують роль джерел електричних сигналів. Формування таким чином двійкового коду знайшло застосування в цифрових пристроях. Кількість інформації в сигналі прийнято вимірювати в бітах (binary digital – двійкова цифра). *Один біт* – це кількість інформації, відповідна одному з двох рівноймовірних повідомлень типу «1» – «0» і означає один двійковий розряд, який може набувати значення логічної одиниці чи логічного нуля.

### 2.3 Кодування інформації

У МПП використовують розрядно-цифрове (цифрове) кодування. Розряди двійкового числа характеризуються вагою, кратною ступеню двійки – 1, 2, 4, 8, ... у напрямку від молодших до старших розрядів), наприклад, для чотирирозрядного двійкового числа маємо:

$$X = X_3 \cdot X_2 \cdot X_1 \cdot X_0 = X_3 \cdot 2^3 + X_2 \cdot 2^2 + X_1 \cdot 2^1 + X_0 \cdot 2^0. \quad (2.2)$$

Якщо двійкове число (BIN)  $X(B) = 1001$ , то отримуємо такий десятковий еквівалент (DEC):

$$X(D) = 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 9. \quad (2.3)$$

Цифровий код передають послідовно в часі (*послідовний код*) за допомогою одного каналу передачі (рис. 2.3) або одночасно (*паралельний код*) за допомогою багатоканальної передачі (рис. 2.4).

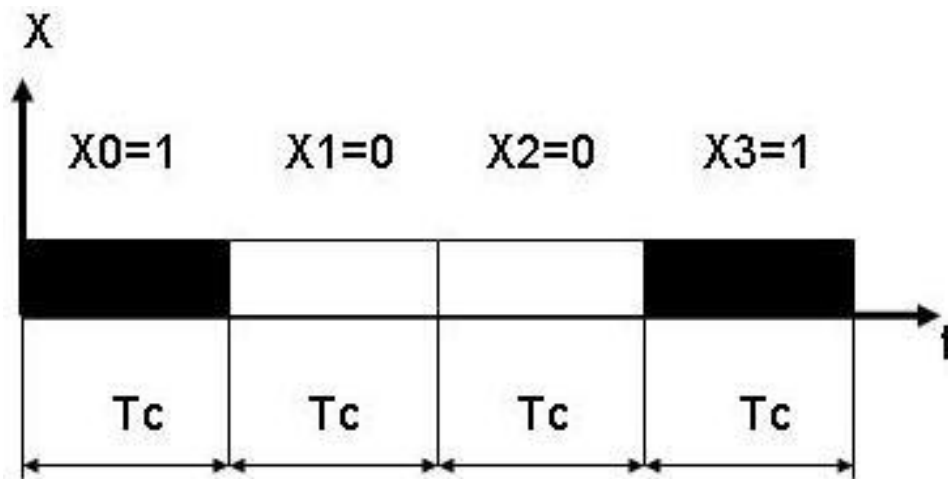


Рисунок 2.3 – Передача інформації послідовним кодом

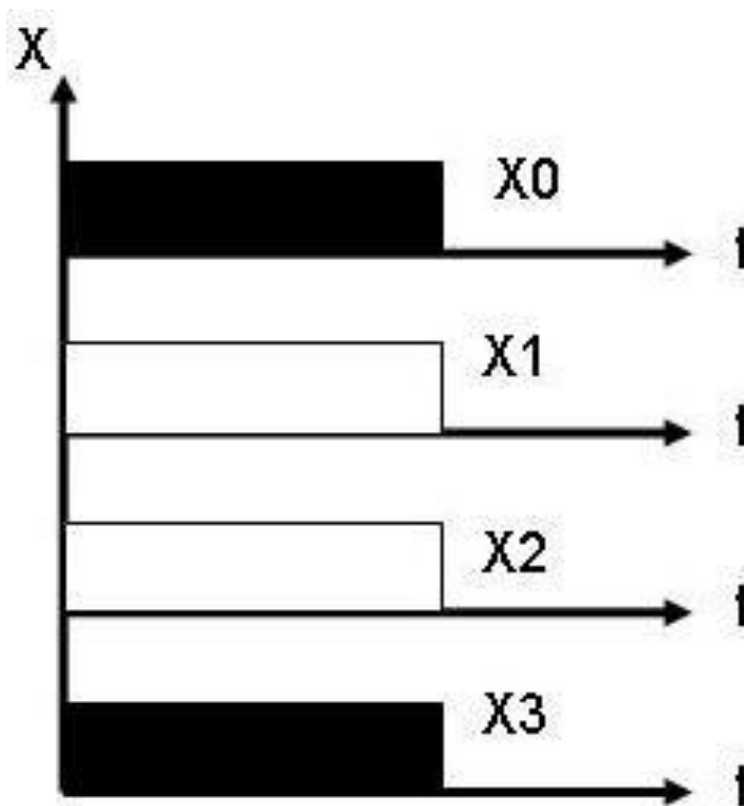


Рисунок 2.4 – Передача інформації паралельним кодом

На практиці послідовний код використовують при передачі інформації на великі відстані, а паралельний – на малі відстані.

У теорії передачі й перетворення інформації встановлені інформаційні міри кількості та якості інформації. Одним із заходів кількості інформації  $I$  є число комбінацій символів, що збігається з числом. Наприклад, для цифр  $q = 1, 2, 3, 4$  можна скласти шість сполучень по дві цифри  $n = 2$ : 12, 13, 14, 23, 24, 34. Таким чином, можна записати:

$$I = q^n. \quad (2.4)$$

Очевидно, що для  $q = 0, 1$  і  $n = 3$  маємо: 000, 001, 010, 011, 100, 101, 110, 111.

Кількість можливих повідомлень довжини  $n$  дорівнює числу розміщень з повтореннями:

$$N = q^n. \quad (2.5)$$

Якщо кількість різних повідомлень дорівнює  $N$ , то кількість інформації за Хартлі складає:

$$I = \log N = n \log q. \quad (2.6)$$

Кількість інформації, що припадає на один елемент повідомлення, називається *ентропією*:

$$H = \frac{I}{n} = \log q. \quad (2.7)$$

Основа логарифму залежить від вибору одиниці кількості інформації. Якщо для бінарної інформації (BIN), то за основу логарифму приймають  $q = 2$ , у результаті чого  $I = n \cdot \log_2 2 = n$ . При довжині  $n = 1$  одержують  $I = 1$  (один біт).

Для зручності використання введені й більші за біт одиниці кількості інформації: 8 символів – 1 байт, 1024 байт – 1 кілобайт (Кбайт), 1024 Кбайт – 1 мегабайт (Мбайт), 1024 Мбайт – 1 гігабайт (Гбайт), при цьому  $1024 = 2^{10}$ .

Максимальна кількість інформації  $I_m$ , яку можна передати за каналом зв'язку (біт):

$$I_m = F_{\max} T_c \log_2 \left( 1 + \frac{P_c}{P_3} \right), \quad (2.8)$$

де  $F_{\max}$  – максимальна частота у спектрі сигналу;

$T_c$  – тривалість сигналу;

$P_c, P_3$  – відповідно середня потужність сигналу й перешкоди.

З формули (2.8) випливає, що збільшення пропускної здатності каналів зв'язку можливе за рахунок розширення смуги частот, тривалості передачі сигналу, покращення відношення  $P_c / P_3$ .

Для аналізу частотних спектрів сигналів застосовують математичний апарат перетворення Фур'є, відповідно до якого сигнал  $U(t)$  можна подати у такому вигляді:

$$U(t) = U_0 + \sum A_n \cos(n\omega t) + \sum B_n \sin(n\omega t). \quad (2.9)$$

Виділення постійної складової  $U_0$  першої гармоніки з частотою  $\omega_1 = \frac{2\pi}{T}$  і вищих гармонік з частотою  $2\omega_1, 3\omega_1$  і т. д. – ряд Фур’є матиме такий вигляд:

$$U(t) = U_0 + \sum U_n \cos(n\omega_1 t + \varphi_n), \quad (2.10)$$

де  $U_0$  – амплітуда;

$\varphi_n$  – фаза  $n$ -ї гармоніки сигналу, яку розраховують:

$$U_n = (A_n^2 + B_n^2)^{0,5}; \quad (2.11)$$

$$\varphi_n = \arctg\left(-\frac{B_n}{A_n}\right); \quad (2.12)$$

$$U_0 = 1/T |U(t)| dt; \quad (2.13)$$

$$A_n = 2/T |U(t)| \cos(\omega_1 t) dt; \quad (2.14)$$

$$B_n = 2/T |U(t)| \sin(\omega_1 t) dt, \quad (2.15)$$

де  $T$  – період сигналу;

$n = 1, 2, 3$  і т.д.

У мікропроцесорних та інших системах керування інформацію передають прямокутними імпульсами з неперервним інформативним параметром. Часово-імпульсне представлення первинного сигналу характеризується періодом слідування  $T$  і тривалістю  $t$  імпульсів. Відношення  $K_3 = t/T$  називають коефіцієнтом заповнення. Для засобів МПП  $K_3 = 0,5$ .

Ефективну ширину частотного спектру сигналу оцінюють за 90 % вмістом потужності, для чого можна використовувати співвідношення:

$$\delta\omega \approx \frac{\omega_1 T}{t} = \frac{\omega_1}{K_3}. \quad (2.16)$$

Таким чином, інформаційний сигнал може мати декілька видів кодування: часово-імпульсне, число-імпульсне, імпульсне-кодове (рис.2.5).

Часово-імпульсне представлення первинного сигналу  $Y(t)$  (рис. 2.5, А) прямокутними імпульсами – тривалість імпульсів, пропорційна значенню сигналу. Число-імпульсне (рис. 2.5, Б) – інформативним параметром є число імпульсів за період. Імпульсно-кодове представлення символів (рис. 2.5, В) –

послідовна посилка в часі п'яти імпульсів різної полярності (застосовується для передачі інформації телетайпом з використанням коду Бодо).

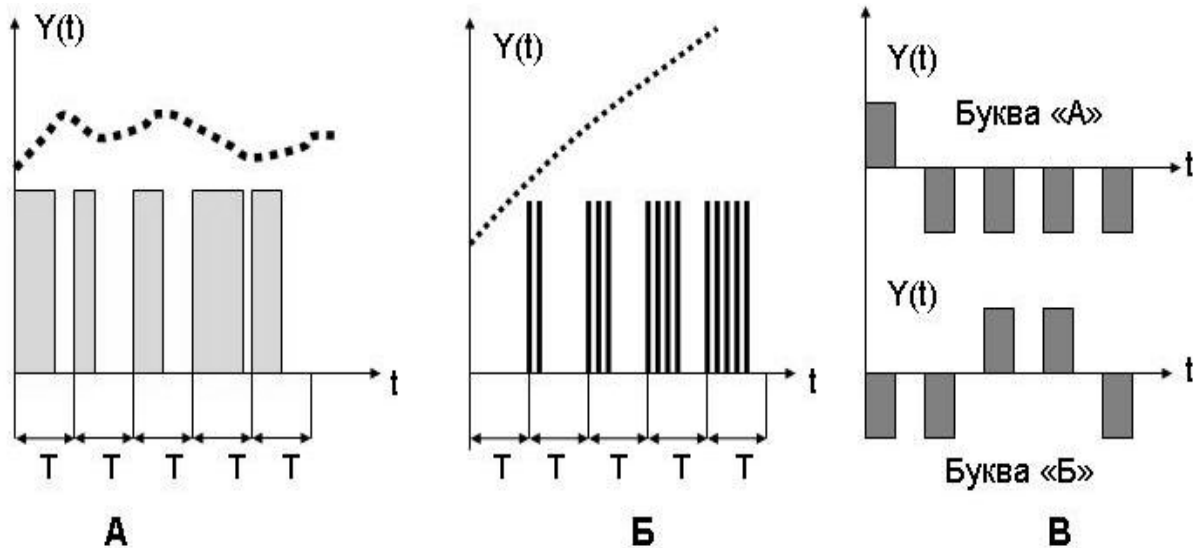


Рисунок 2.5 – Представлення сигналів у мікропроцесорній техніці:  
*A – часово-імпульсне; B – число-імпульсне; B – імпульсно-кодове*

## 2.4 Контрольні питання

1. Чому засоби автоматики називаються мікропроцесорними?
2. Що таке сигнал?
3. Поясніть принципи формування інформаційного сигналу.
4. У чому різниця між одно- і багатовимірними технологічними об'єктами на транспорті?
5. Наведіть приклади джерел кількісної і якісної інформації на транспорті.
6. Поясніть особливості аналогових інформаційних сигналів, наведіть приклади їх формування на транспорті .
7. Що таке випадкові сигнали?
8. Поясніть принципи формування логічних сигналів, наведіть приклади їх формування на транспорті.
9. У чому особливості детермінованих джерел інформації?
10. Поясніть кількість міри інформації – біт, байт, Кбайт та ін.
11. Які існують заходи вимірювання інформації?
12. Поясніть поняття кодування інформації в МПП.
13. Поясніть на прикладах принципи формування в МПП пристроях автоматики послідовного і паралельного кодів.
14. Чому представлення первинного сигналу називають часово-імпульсним?
15. Поясніть числово-імпульсний принцип формування сигналу.
16. У чому особливості імпульсно-кодового представлення інформаційних сигналів?

17. Охарактеризуйте основні системи числення, які використовуються в МПП.
18. Запишіть число 100D у BIN-кодi.



## 3 ЛОГІЧНІ ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ

### 3.1 Булева алгебра

Теоретичною основою мікропроцесорних пристроїв є *алгебра логіки* – наука, яка використовує математичні методи для розв’язання логічних задач. Алгебру логіки називають *булевою* на честь математика Дж. Буля.

Основним предметом булевої алгебри є висловлювання – просте твердження, про яке можна стверджувати: істинне воно (1) або хибне (0). Звичайно прості висловлювання позначають буквами  $X_1, X_2, X_3, \dots, X_n$  які називають змінними (аргументами). За допомогою логічних зв’язків НІ (NOT), АБО (OR), І (AND), ЯКЩО (IF) ТО (to), будують складні висловлювання, які називають булевими (логічними) функціями.

Використання апарата алгебри логіки в схемотехніці засноване на тому, що цифрові елементи характеризуються двома станами і через це можуть бути описані булевими функціями. Стандарт ДСТУ 2533-94 Арифметичні і логічні операції. Терміни і визначення конкретизує основні поняття булевої алгебри в системах оброблення інформації.

*Операція* – це чітко визначена дія над одним або декількома операндами, яка створює результат. Булеву операцію над одним операндом називають одномісною, над двома – двомісною і т.д.

*Заперечення* – це одномісна булева операція  $Y = \overline{X}$  («не X»), результатом якої є значення, протилежне значенню операнда.

*Кон’юнкція* – це булева операція  $Y = X1 * X2$  («X1 і X2»), результатом якої є значення одиниця тоді і тільки тоді, коли значення кожного операнда дорівнює одиниці. Вираз  $X1 * X2$  часто записують як  $X1 \& X2$ .

*Диз’юнкція* – це булева операція  $Y = X1 \vee X2$  («X1 чи X2»), результатом якої є значення нуль тоді і тільки тоді, коли обидва операнди мають значення нуль.

Для булевих операцій заперечення, диз’юнкції, кон’юнкції справедливі закони, властивості й тотожності логіки.

Булеву функцію задають різними способами: словесним описом, часовими діаграмами, геометричними фігурами, графами, таблицями істинності й аналітичними виразами.

Однією з інтерпретацій булевих операцій є схеми, які складаються з ключів ( $X_i$ ), джерела напруги (E) і лампочки (HL). Для реалізації операції диз’юнкції двох змінних  $X1$  і  $X2$  використовують два паралельно з’єднаних нормально розімкнутих ключі (рис.3.1, А).

При натисканні будь-якого ключа ( $X1 = 1$  або  $X2 = 1$ ) або обох разом лампочка HL горить (значення 1).

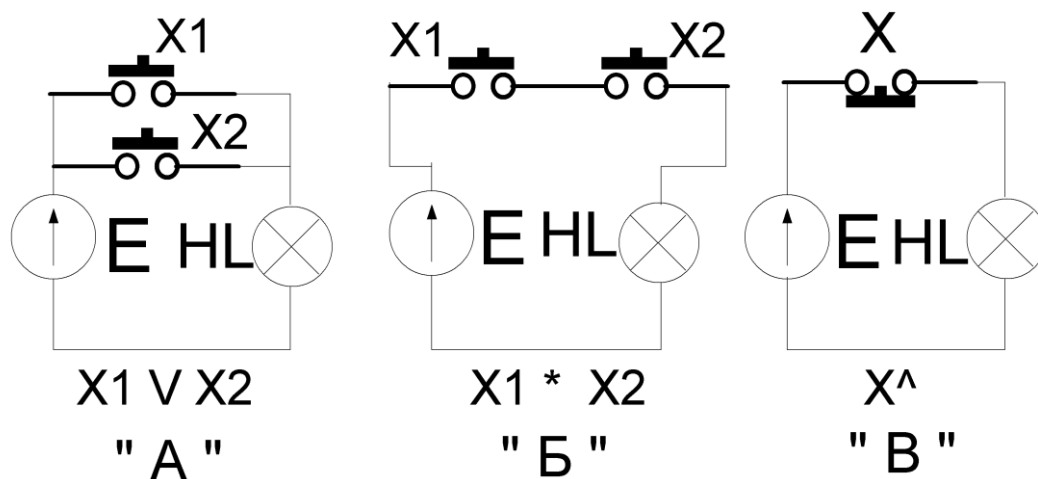


Рисунок 3.1 – Інтерпретація булевих операцій:  
*A – диз'юнкція; Б – кон'юнкція; В – заперечення*

Для реалізації операції кон'юнкції двох змінних  $X1$  і  $X2$  застосовують два послідовно з'єднаних нормально розімкнутих ключі (рис. 3.1, Б). При натисканні одночасно обох ключів ( $X1 = X2 = 1$ ) лампочка горить (значення 1). Для реалізації операції заперечення застосовують нормально замкнутий ключ (рис. 3.1, В). При  $X = 0$  ключ замкнутий і лампочка горить; при  $X = 1$  ключ розмикається і лампочка не горить.

Вище розглянуті операції можна представити у вигляді таблиць істинності:

Таблиця 3.1 – Таблиця істинності операції заперечення

$X$	$Y = \bar{X}$
0	1
1	0

Таблиця 3.2 – Таблиця істинності операцій диз'юнкції і кон'юнкції

$X1$	$X2$	$Y = X1 \vee X2$	$Y = X1 \cdot X2$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Розроблено універсальні форми представлення булевих функцій, які дають можливість одержати аналітичну форму довільної функції безпосередньо з таблиці істинності.

Найбільше поширення одержали *досконала диз'юнктивна нормальна форма* (далі – ДДНФ):

$$Y = f_0 m_0 V \cdot f_1 m_1 V \dots f_n m_n V, \quad (3.1)$$

де  $f_i - 1$  або  $0$ ;

$m_i$  – мінтермі (функція  $n$  змінних, яка дорівнює одиниці тільки на одному наборі аргументів).

*Досконала кон'юнктивна нормальна форма* (далі – ДКНФ):

$$Y = (f_0 VM_0) \cdot (f_1 VM_1) \dots (f_n VM_n), \quad (3.2)$$

де  $M_i$  – макстерм (функція  $n$  змінних, яка дорівнює нулю тільки на одному наборі аргументів).

Важливим етапом проектування мікропроцесорних (далі – МП) схем є мінімізація булевих функцій, тобто знаходження їхніх виражень з мінімальним числом букв. Мінімізація забезпечує побудову економічних схем МПП, оскільки реалізація логіки досягається при найменшій кількості електронних компонентів (елементів).

### 3.2 Поняття елементів мікропроцесорних пристроїв

Технічні засоби МПП залежно від функцій, які вони виконують, поділяться на елементи, функціональні вузли і пристрої. Всі вони призначені для обробки дискретної інформації, тому називаються *цифровими*.

Технічні засоби МПП засновані на інтегральних мікросхемах (далі – ІМС) різного ступеня складності.

*Елементами* в МПП називаються найменші неподільні мікроелектронні вироби, призначені для виконання логічних операцій або зберігання біта інформації. До елементів умовно відносяться і допоміжні схеми-підсилювачі, повторювачі, формувачі та ін. Елементи будуються на основі двопозиційних ключів, що технічно реалізуються найпростіше. Елементи з двома станами називаються *двійковими*.

На входах і виходах двійкового елемента діють напруги, які набувають у сталому режимі двох значень – високого ( $5,0V \geq High \geq 2,5V$ ) і низького ( $0V \geq Low \geq 0,5V$ ) рівнів.

За конструкційно-технологічним виготовленням елементна база складається з *інтегральних мікросхем*. Це – мікроелектронні вироби з високою щільністю упакування резисторів, діодів, транзисторів і з'єднань між ними.

Мікросхеми класифікують за такими ознаками:

- технологією виготовлення – напівпровідникові, гібридні, плівкові;
- конструкційним оформленням – корпусні й безкорпусні;
- формою оброблення інформації – аналогові, цифрові й аналогово-цифрові;
- ступенем інтеграції – малі, середні, великі, надвеликі й ультравеликі;

- типом активних елементів – побудовані на біполярних і МОН-транзисторах;
- областю застосування – широкого застосування і спеціалізовані;
- використовуваними матеріалами – кремнієві, арсенід-галієві;
- перспективними напрямками – оптоелектронні, акустоелектронні та інші.

У малих і ультравеликих ІС розміщується від 100 до 10 млн. компонентів. Елементарні логічні функції реалізуються відповідно логічними схемами (АБО, І, НІ), які є сукупністю взаємозалежних логічних елементів з формальними методами їх опису.

Корпуси мікросхем звичайно містять декілька однотипних логічних елементів, в яких найчастіше декілька входів. При розробці логічних пристроїв завжди треба враховувати сімейства логічних елементів, які в технічній документації відмічені як компоненти позитивної або негативної логіки.

### **3.3 Контрольні питання**

1. Поясніть призначення булевої алгебри.
2. Що таке булева операція?
3. Поясніть логіку типових булевих операцій.
4. Які існують способи завдання булевих функцій?
5. Використовуючи приклад реалізації контактних схем на транспорті, поясніть логіку *I*.
6. Використовуючи реальний приклад, поясніть логіку АБО.
7. У чому сенс логіки НІ?
8. Поясніть призначення таблиці істинності.
9. Для чого використовують мінімізацію булевих виразів?
10. Що таке інтегральна мікросхема?
11. Охарактеризуйте класифікаційні ознаки елементів МПП.

## 4 ЕЛЕМЕНТНА БАЗА МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ

### 4.1 Система логічних елементів

У МПП в основному використовують потенціальну систему елементів. Вона має такі особливості:

- на входах і виходах логічних елементів діють тільки потенціальні сигнали ;
- з виходу одного елемента на вхід іншого передаються як перехідні, так і встановлені значення сигналів;
- реалізується обмежений набір булевих функцій: НІ, АБО, І, НІ-АБО, НІ-І та ін., що полегшує застосування автоматизованих методів проектування (див. програму SinSys / меню: допомога, САПР – система автоматизованого проектування схем).

Потенціальні елементи розрізняють за схемою за технічними ознаками – способом з'єднання транзисторів, діодів і резисторів між собою в межах однієї схеми типового базового елемента. Сукупність із загальною ознакою побудови утворює вид схемної логіки.

#### 4.1.1 Елемент АБО

Логіка роботи *логічного елемента* (далі – ЛЕ) АБО (OR): на два входи X1 і X2 подаються вхідні сигнали (табл.4.1). Функцію Y можна записати для довільного числа вхідних змінних:

$$Y = X1 \vee X2 \vee \dots \vee X_n \vee . \quad (4.1)$$

Схема двоходового елемента АБО, його умовне графічне зображення і часові діаграми роботи показані на рис. 4.1.

Таблиця 4.1 – Залежність вихідного сигналу ЛЕ АБО

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

Високий рівень напруги на виході елемента АБО встановлюється при подачі на один або обидва входи високих рівнів напруги, при яких відкриваються відповідні кремнієві діоди VD1 чи VD2 або обидва разом. При подачі одночасно на обидва входи низьких рівнів напруги діоди закриті, струм у колі навантаження не протікає і вихідна напруга майже дорівнює нулю. Ілюстрація роботи ЛЕ «OR» представлена в програмі SinSys (електронна лабораторія «ПрЭ-Дт»).

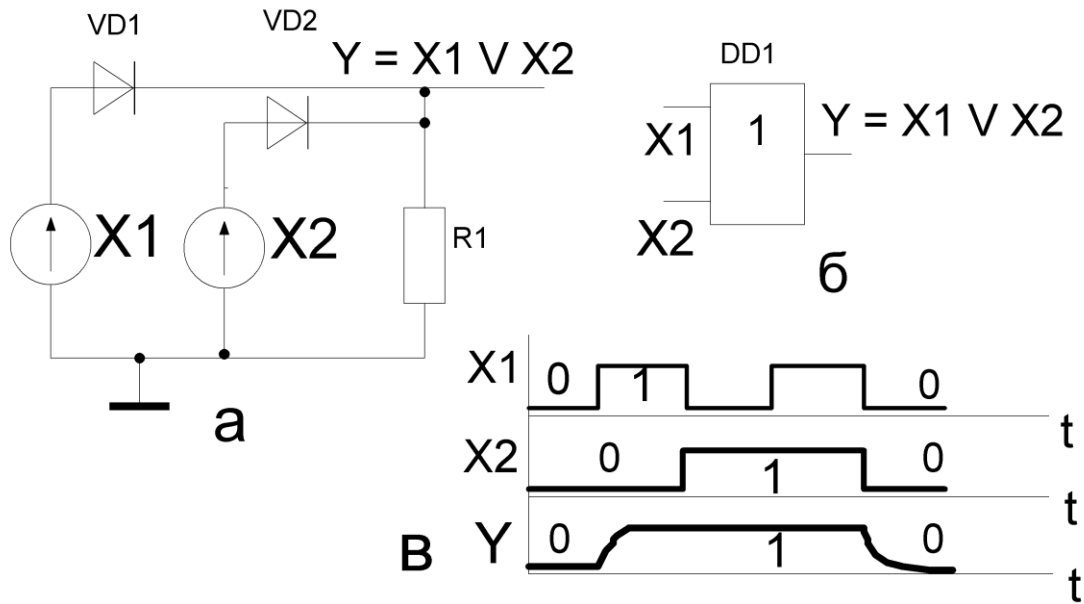


Рисунок 4.1 – Діодний елемент АБО:  
*a* – схема; *б* – умовне графічне зображення; *в* – часові діаграми роботи

#### 4.1.2 Елемент І

Логіка роботи ЛЕ І (AND) подана в таблиці 4.2. На основі логіки одержують вираз для вихідної булевої функції елемента  $Y = X1 \cdot X2$ . Функцію  $Y$  для довільного числа вхідних змінних можна записати:

$$Y = X1 \cdot X2 \cdot \dots \cdot X_n. \quad (4.2)$$

Таблиця 4.2 – Залежність вихідного сигналу ЛЕ І

X1	X2	Y
0	0	0
0	1	0
1	0	0
1	1	1

На рисунку 4.2 наведені схема, графічне зображення і часові діаграми роботи ЛЕ І.

Тривалість фронту вихідного сигналу визначається часом заряду паразитної ємності через резистор R1. Якщо на один із входів, наприклад X1, поданий низький рівень напруги, то діод VD1 відкривається. При цьому від джерела живлення U через резистор R1, відкритий діод VD1 і джерело вхідного сигналу X1 протікає струм і встановлюється низький рівень напруги. Ілюстрація роботи ЛЕ «AND» представлена в програмі SinSys (електронна лабораторія «ПрЭ-Дт»).

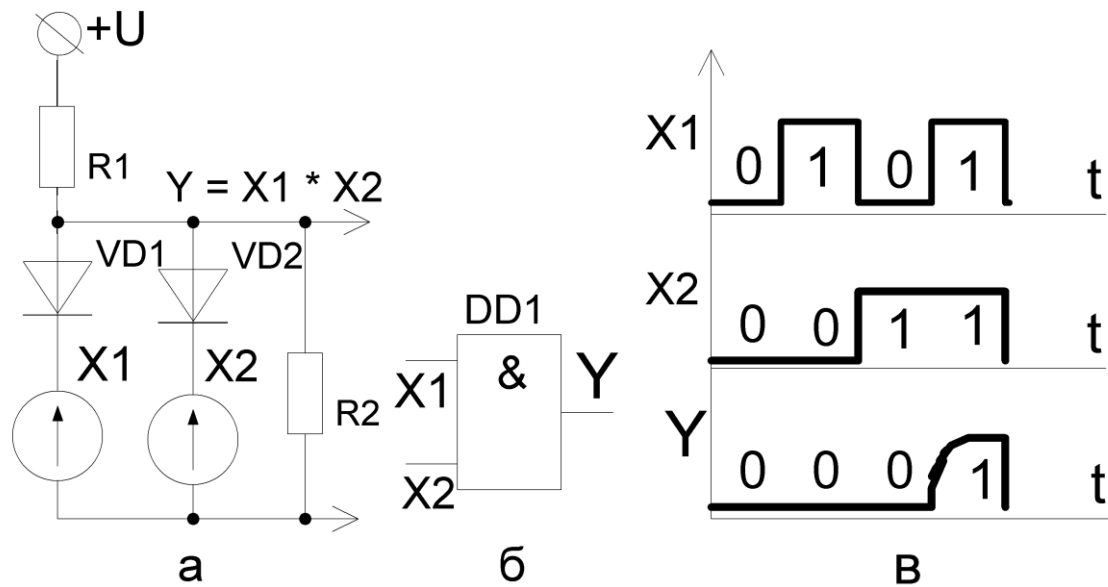


Рисунок 4.2 – Діодний елемент І:  
*a – схема; б – умовне графічне зображення; в – часові діаграми роботи*

#### 4.1.3 Елемент НІ

Логіку роботи елемента НІ (NOT) або інвертора ілюструє таблиця істинності (табл. 4.3). На основі таблиці 4.3 одержують вираз для вихідної булевої функції  $Y = \overline{X}$ .

Таблиця 4.3 – Залежність вихідного сигналу ЛЕ НІ

X	Y
0	1
1	0

Схема елемента НІ, його умовне графічне зображення і часові діаграми роботи показані на рисунку 4.3. Схема елемента НІ (транзисторний ключ): транзистор VT1; резистори в колі колекторного навантаження  $R_k$  і бази  $R_b$ ;  $U$  – джерело живлення.

Високий рівень напруги  $U$  на виході діодного елемента І встановлюється тільки при одночасній подачі на обидва входи високих рівнів напруги при яких закриваються кремнієві діоди VD1 і VD2. При цьому від джерела живлення  $U$  через резистор  $R1$  і  $R2$  протікає струм навантаження:

$$I = \frac{U}{R1 + R2} \quad (4.3)$$

І встановлюється значення високого рівня вихідної напруги  $U_{BP}$ :

$$U_{BP} = IR_2 = \frac{UR_2}{R_1 + R_2}. \quad (4.4)$$

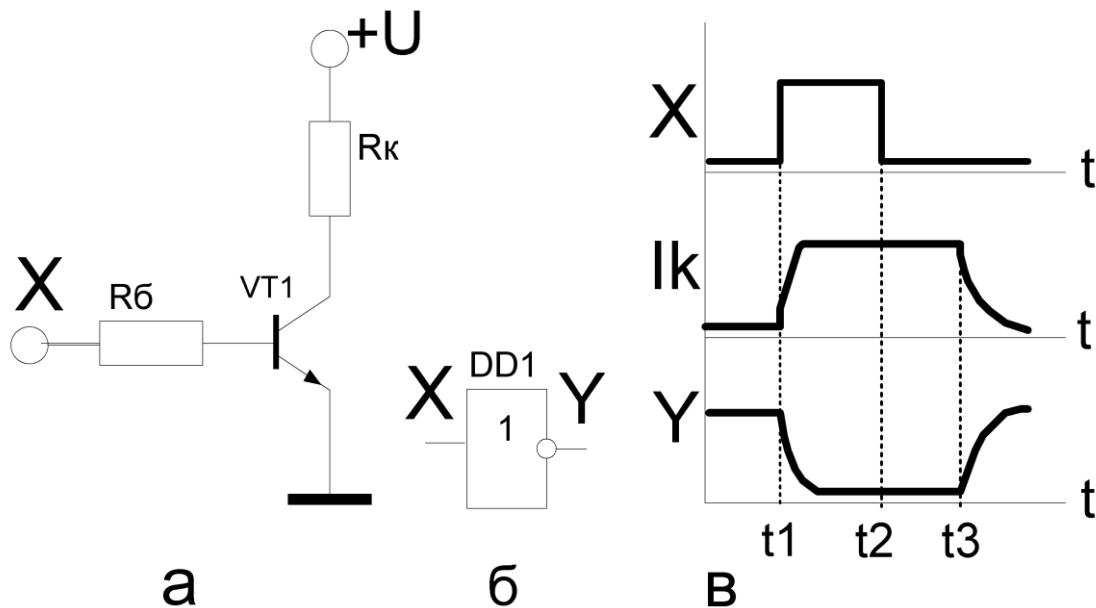


Рисунок 4.3 – Елемент ІІ:

*а – схема; б – умовне графічне позначення; в – часові діаграми роботи*

Транзистор VT1 може знаходитися в трьох основних режимах: відсікання колекторного струму (*закритий стан*), насичення (*відкритий стан*) і активної роботи (*посилення*). У режимі відсікання колекторний і базовий переходи закриті (на вхід поданий низький рівень напруги), у колі колектора протікає дуже малий обернений струм колекторного переходу і на колекторі транзистора встановлюється високий рівень напруги.

У режимі насичення (на вхід поданий високий рівень напруги) на переході база–емітер відбувається пряме падіння напруги. Через колектор протікає максимально можливий струм, який називається *струмом насичення колектора*. У базі накопичується надлишковий заряд. При подачі низького рівня вхідного сигналу транзистор закривається. Колекторний струм залишається постійним в інтервалі часу ( $t_2 - t_3$ ) – *час розсмоктування надлишкового заряду в базі* (рис. 4.3, в). Після закінчення розсмоктування колекторний струм спадає і формується фронт вихідного сигналу інвертора. Ілюстрація роботи ЛЕ «NOT» представлена в програмі SinSys (електронна лабораторія «ПрЭ-Дт»).

#### 4.1.4 Тригер

*Тригер* – це запам’ятовуючий елемент з двома стійкими станами, зміна яких відбувається під дією вхідних сигналів. Елемент призначений для зберігання одного біта інформації, тобто 0 або 1. Схема тригера забезпечує записування, зчитування, стирання та індикацію двійкової інформації, яка



зберігається. На основі тригерів будують типові функціональні вузли МПП – реєстри, лічильники, накопичувальні суматори, мікропрограмні автомати та ін.

#### 4.1.4.1 Асинхронний RS- тригер

*RS-тригером* називають запам'ятовуючий елемент з роздільними інформаційними входами для встановлення його в стан «0» (R-вхід) або в стан «1» (S-вхід). Назва елемента утворена від перших літер слів RESET (скидання) і SET (встановлення). У таблиці переходів RS-тригера (табл. 4.4) прийняті позначення:  $R_t$ ,  $S_t$ ,  $Q_t$  – значення логічних змінних у момент часу  $t$  на входах  $R$ ,  $S$  і виході  $Q$ ;  $Q_{t+1}$  – стан тригера після перемикання; «ЗК» – заборонена комбінація сигналів (одночасне набування  $R = S = 1$ ).

Приклад схеми RS-тригера, його умовне графічне позначення і часові діаграми роботи показані на рисунку 4.4. Особливістю цього тригера є інверсне керування за інформаційними входами. Із діаграм роботи випливає, що елементи НИ-І в схемі перемикаються послідовно. Інтервал часу, коли на обох виходах встановлюються однакові сигнали  $Q \cdot \bar{Q} = 1$  (заштриховані області) – явище «ризик». Тривалість перемикання тригера визначається сумою затримок  $t_3 = 2t$ . Максимальна  $F_{\max}$  і робоча  $F_p$  частоти перемикання тригера відповідно дорівнюють  $F_{\max} = 1/(2t)$  і  $F_p = 1/(3t)$ .

Таблиця 4.4 – Таблиця переходів RS-тригера

$R_t$	$S_t$	$Q_t$	$Q_{t+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	ЗК
1	1	1	ЗК

Ілюстрація роботи RS-тригера представлена в програмі SinSys (електронна лабораторія «ПрЭ-Дт»).

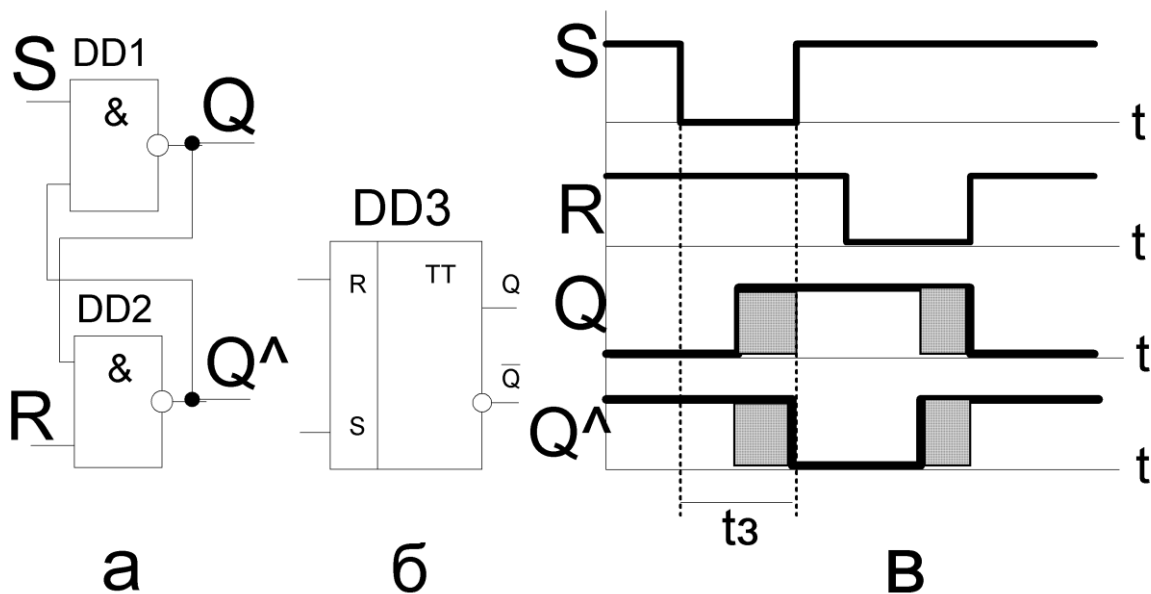


Рисунок 4.4 – Асинхронний RS-тригер на елементах НІ-І:  
*а – схема; б – умовне позначення; в – часові діаграми*

#### 4.1.4.2 Синхронний RS-тригер

Для побудови синхронного RS-тригера на елементах НІ-І у схему (рис. 4.4) вводять додаткові І-НІ елементи, що забезпечують формування  $C$  – синхросигнала. Схема синхронного RS-тригера на чотирьох елементах НІ-І з логічними зв'язками сигналів  $R, S, C, CS^{\wedge}, CR^{\wedge}, Q$  показана на рисунку 4.5.

Додаткові елементи DD3 і DD4 складають схему керування (з прямими входами) асинхронного RS-тригера на елементах DD1, DD2.

При значенні сигналів  $CS = 1$  на виході елемента DD3 встановлюється «0» і тригер перемикається у стан «1». При значенні сигналів  $CR = 1$  на виході елемента DD4 встановлюється «0» і тригер перемикається у стан «0». Комбінація вхідних сигналів  $CSR = 1$  заборонена, оскільки призводить до невизначеного стану тригера.

Заборонені стани (далі – ЗК) рівнів напруги на входах тригерів необхідно завжди враховувати. Особливе це важливо при розробці логічних схем автоматики з використанням цих елементів пам'яті.

Із часової діаграми (рис. 4.5, б) випливає, що час перемикання тригера  $t_{nm} = 3t$ , а тривалість синхросигналу визначається з умови  $t_c = 4t$ . Максимальна  $F_{\max}$  і робоча  $F_p$  частоти перемикання тригера відповідно дорівнюють:  $F_{\max} = 1/(3t)$  і  $F_p = 1/(4t)$ .

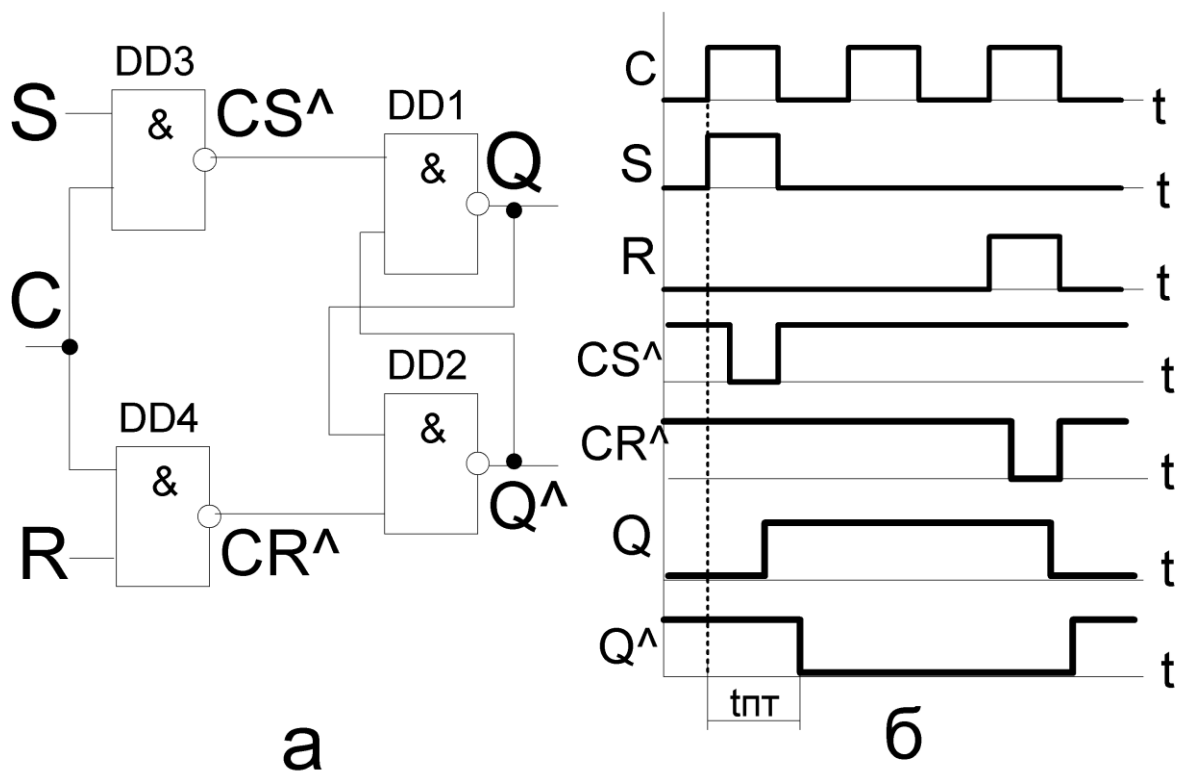


Рисунок 4.5 – Синхронний RS-тригер на елементах НІ-І:  
а – схема; б – часові діаграми

#### 4.1.4.3 Тригери інших типів

*Тригер типу JK* (умовне позначення JK-тригера ілюструє рисунок 4.6, а) називається запам'ятовуючий елемент з двома стійкими станами та інформаційними входами J (аналог S) і K (аналог R), що забезпечують відповідно роздільну установку станів «1» і «0». Він функціонує подібно до RS-тригера, але при збігу сигналів  $JK = 1$  переключається в протилежний стан, тобто реалізує додавання сигналів за модулем два. Таким чином цей тригер не має заборонених комбінацій вхідних сигналів. JK-тригер є універсальним, оскільки може виконувати функції RS-тригера (при роздільному знаходженні сигналів J і K), T-тригера (при одночасній подачі сигналів J і K), D-тригера (при подачі сигналу від входу J через інвертор на вхід K). Зміна станів наведена в таблиці 4.5.

*T-тригер* – елемент з двома стійкими станами та одним інформаційним T-входом. Стан T-тригера змінюється на протилежний після кожного надходження лічильного сигналу на T-вхід. Зміна станів наведена в таблиці 4.6. Умовне позначення T-тригера ілюструє рисунок 4.6, б.

*D-тригер* – синхронний запам'ятовуючий елемент з двома стійкими станами і одним інформаційним D-входом. Цей тригер «слідкує» за зміною сигналу на D-вході під час дії синхросигналу C і зберігає ту інформацію, яка була в момент його закінчення. Наприклад, в RS-тригері такої властивості не має. Умовне позначення D-тригера ілюструє рисунок 4.6, в.

Для затримки інформації в D-тригері на довільне число тактів використовується дозволяючий V-вхід (двоступеневий одноктактний DV-тригер). Якщо  $V = 1$ , то DV-тригер функціонує як звичайний тригер затримки; якщо  $V = 0$ , то робота схеми за входами блокується і DV-тригер зберігає попередню інформацію.

Таблиця 4.5 – Зміна станів JK-тригера

$K_t$	$J_t$	$Q_t$	$Q_{t+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

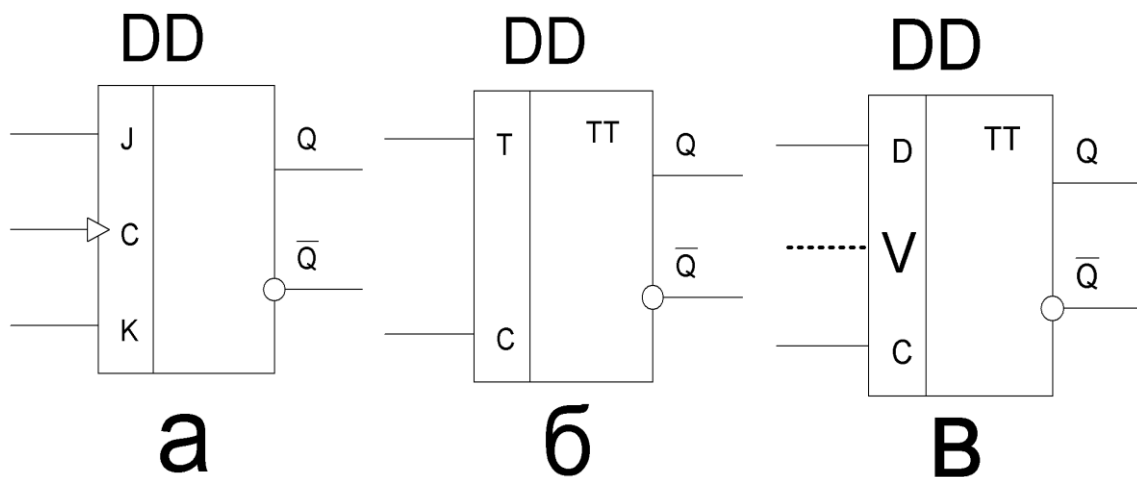


Рисунок 4.6 – Умовні позначення тригерів:  
*а – JK-тригер; б – T-тригер; в – D-тригер*

Таблиця 4.6 – Зміна станів T-тригера

$T_t$	$Q_t$	$Q_{t+1}$
0	0	0
0	1	1
1	0	1
1	1	0

## 4.2 Послідовні (накопичувальні) пристрої

Послідовні пристрої мають пам'ять і при зміні сигналів на вході для визначення сигналів на виході необхідно знати крім набору входніх сигналів і стан, в якому пристрій був до цього.

Послідовні пристрої будують з логічних елементів. Роботу цих елементів аналізують у безрозмірному дискретному часі. Для цього роблять розбиття реального часу на інтервали, які називають тактовими імпульсами. Оновлення інформації на виході послідовного пристрою відбувається у момент початку нового такту.

### 4.2.1 Лічильники та дільники

*Лічильник* – це послідовний пристрій, призначений для підрахунку кількості імпульсів, що надійшли на його вхід. Лічильник, зібраний з  $n$  послідовно з'єднаних тригерів, може підраховувати у двійковому коді  $2^n$  імпульсів. У лічильниках з позиційним кодуванням числовий вираз  $N$  поточного стану лічильника визначається за формулою:

$$N = \sum r_i Q_i = r_n Q_n + r_{n-1} Q_{n-1} + \dots + r_1 Q_1, \quad (4.5)$$

де  $r_i$  – вага  $i$ -го розряду;

$Q_i$  – значення виходу  $i$ -го розряду;

$n$  – число розрядів.

Це число називається коефіцієнтом рахунку і характеризує ємність лічильника (рис. 4.7).

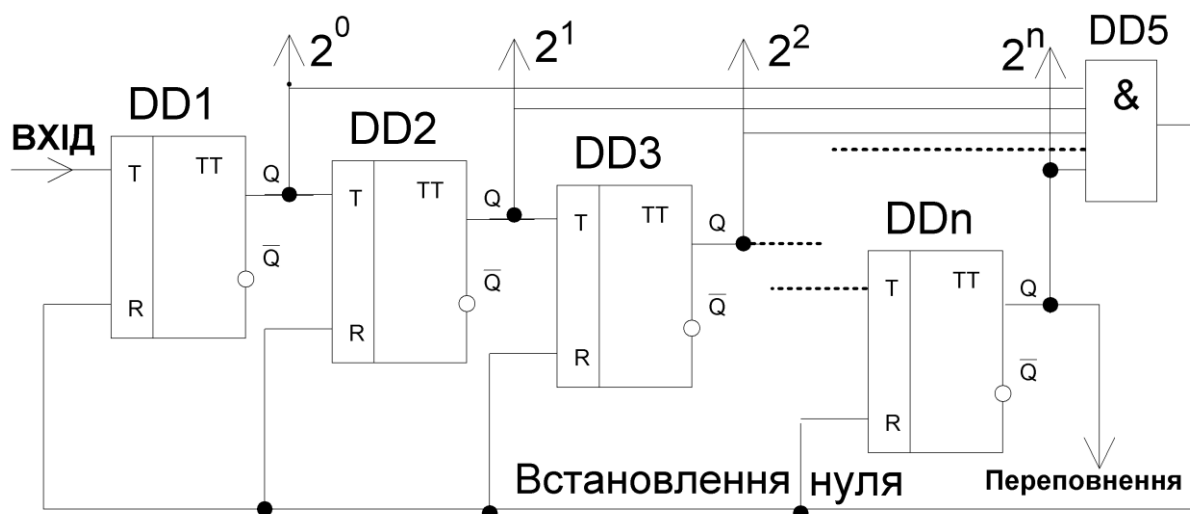


Рисунок 4.7 – Асинхронний двійковий сумуючий лічильник на Т-тригерах

Нульовий стан усіх тригерів приймається за нульовий стан лічильника в цілому. Інші стани нумеруються за числом входніх імпульсів. Коли досягається

коефіцієнт рахунку, відбувається переповнення лічильника, після чого він повертається у нульовий стан (за сигналом високого рівня на виході елемента DD5) і цикл рахунку повторюється.

Лічильники розрізняють залежно від коефіцієнта рахунку на двійкові, двійково-десяткові, десяткові та ін. На виході таких лічильників кількість зареєстрованих імпульсів відображається у відповідному коді (рис. 4.8).

Залежно від напрямку рахування розрізняють лічильники сумуючі, від'ємні, реверсивні. Асинхронні лічильники можна використовувати як дільники частоти, оскільки часта появи сигналів на виході лічильника дорівнює відношенню частоти надходження вхідних імпульсів, поділених на коефіцієнт рахунку, а кожний каскад лічильника поділяє частоту на два.

Лічильники класифікують за такими ознаками:

- способом кодування – позиційні й непозиційні;
- модулем лічби – двійкові, десяткові, з довільним постійним або змінним (програмованим) модулем;
- напрямком лічби – прості (підсумовуючі, піднімальні) й реверсивні;
- способом організації міжрозрядних зв'язків – з послідовним, наскрізним, паралельним і комбінованим переносами (позицією);
- типом використовуваних тригерів – Т, JK, D в лічильному режимі;
- елементним базисом – потенціальні, імпульсні й потенціально-імпульсні.

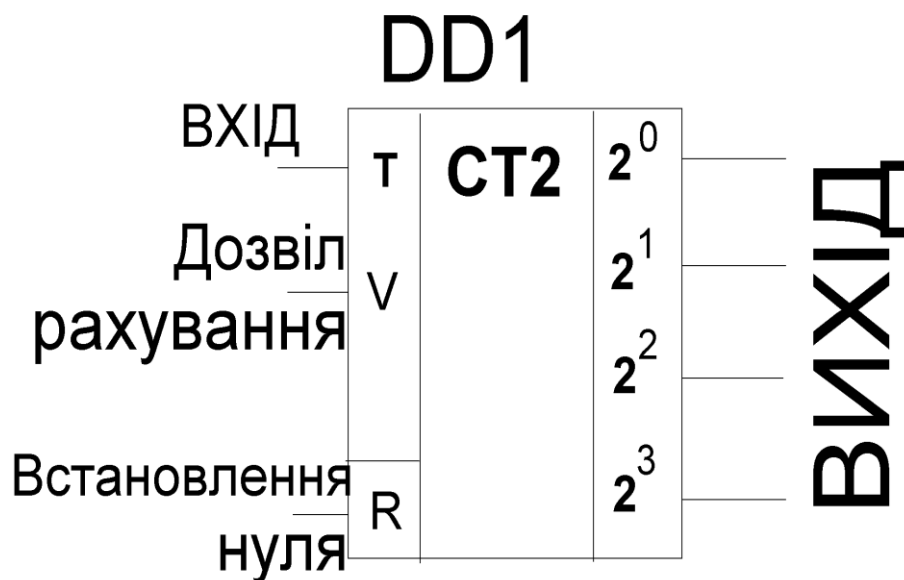


Рисунок 4.8 – Умовне позначення двійкового 4-розрядного лічильника

Лічильники є одним з основних функціональних вузлів МПП і різних цифрових керуючих систем автоматики.

Основне застосування усіх лічильників:

- утворення послідовності адрес команд програм (лічильники команд);
- підрахунок числа циклів при виконанні операцій ділення, множення, зсуву (лічильник циклів);

– одержання сигналів мікрооперацій і синхронізації; аналого-цифрові перетворення і побудова електронних таймерів (годинників реального часу).

#### 4.2.2 Регістри

*Основне призначення регістрів* – зберігання і перетворення багаторозрядних двійкових чисел. Всі регістри підрозділяють на регістри пам'яті, регістри зсуву та багаторежимні буферні регістри.

*Регістри пам'яті* – група синхронних тригерів, кожний з яких зберігає один розряд двійкового числа (рис. 4.9).

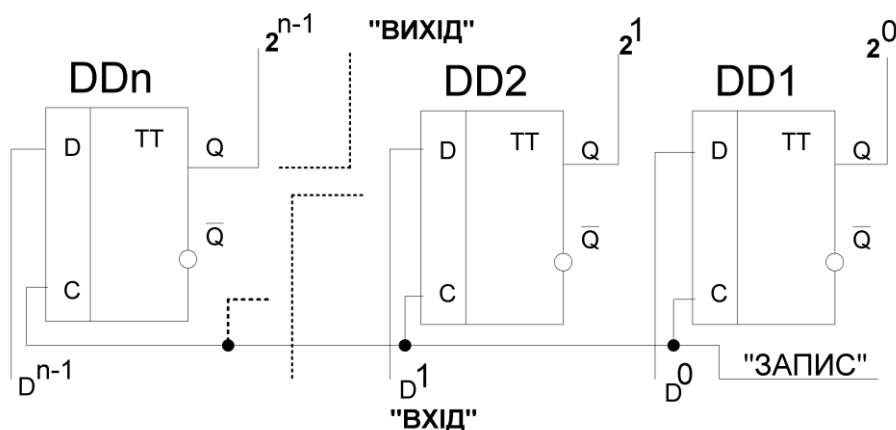


Рисунок 4.9 – Схема регістра пам'яті на D-тригерах

Сигнал, що надходить на вхід кожного тригера, передається на його вихід, тобто записується у пам'яті регістра лише з появою на вході синхронізації кожного тригера імпульсу «ЗАПИС». Таким чином призначення лінії «ЗАПИС» – дозвіл на запам'ятовування вхідної інформації.

Регістри зсуву окрім операції зберігання інформації виконують перетворення послідовного двійкового коду в паралельний і навпаки, здійснюють арифметичні й логічні операції.

При виконанні операції зсуву з надходженням тактового синхроімпульсу відбувається перезапис (зсув) змісту кожного тригера в сусідній розряд без зміни послідовності одиниць і нулів.

Регістри можуть виконувати зсув вліво і вправо. Є також реверсивні регістри. Зсув інформації на один двійковий розряд відповідає арифметичній операції множення або ділення на два.

Багаторежимні буферні регістри (далі – ББР) окрім одиничного й нульового стану спроможні приймати і третє – високоімпедансне. Перехід у високоімпедансний стан відбувається при надходженні відповідного сигналу на спеціальний вхід. ББР використовують в мікропроцесорних пристроях як порти вводу – виводу бінарної інформації. Приклади зображення регістрів на схемах ілюструє рисунок 4.10.

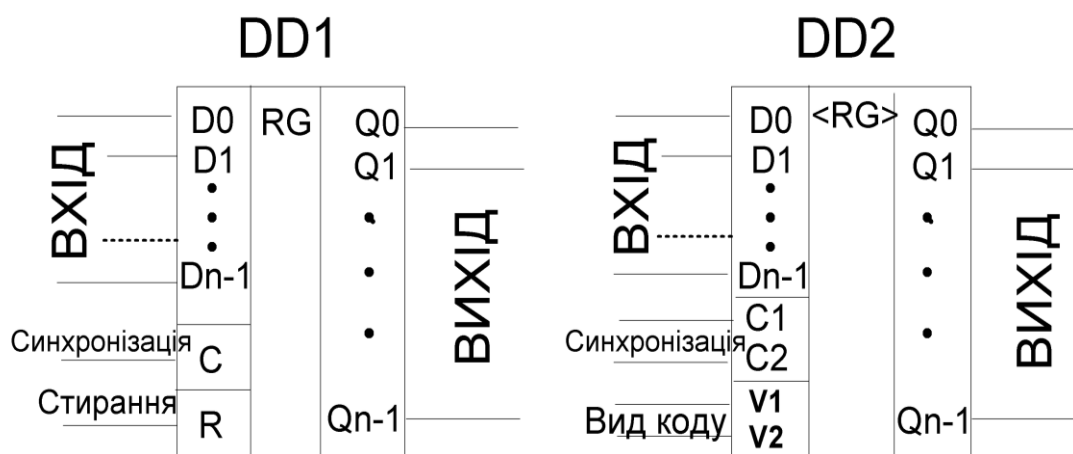


Рисунок 4.10 – Умовне позначення регістрів:  
*DD1 – пам'яті; DD2 – реверсивного зсуву*

У системах синхронізації і пристроях керування на транспорті використовують розподільники тактів, що виробляють серії імпульсів. Розподільники тактів будуються на кільцевих регістрах, в яких вихід молодшого розряду з'єднаний з інформаційним входом старшого, якщо реалізований зсув вправо, або є зв'язком від старшого розряду до молодшого при зсуву вліво.

### 4.3 Комбінаційні функціональні схеми

Схемною ознакою комбінаційних схем є відсутність ланцюгів зворотного зв'язку. До таких пристроїв відносять базові елементи АБО, І, НІ та їх модифікації, мультиплексори, демультимплексори, шифратори, дешифратори, компаратори, підсилювачі та ін. Сигнали на виході цих пристроїв у будь-який момент часу цілком визначаються сполученням сигналів на їх входах і не залежать від їх попереднього стану.

#### 4.3.1 Дешифратори

*Дешифратор* – функціональний вузол МПП, призначений для перетворення кожної комбінації вхідного двійкового коду в керуючий сигнал лише на одному із своїх виходів. У загальному випадку дешифратор має  $n$  однофазних входів і  $m = 2^n$  виходів, де  $n$  – розрядність (довжина) коду, який дешифрується. Дешифратор з максимально можливим числом виходів  $m$  називається повним. Функціонування повного дешифратора описується системою логічних виразів вигляду:



$$\begin{aligned}
 Y_0 &= \overline{X_n} \cdot \overline{X_{n-1}} \dots \overline{X_2} \cdot \overline{X_1} \\
 Y_1 &= \overline{X_n} \cdot \overline{X_{n-1}} \dots \overline{X_2} \cdot X_1 \\
 &\dots \dots \dots \\
 Y_{m-1} &= X_n X_{n-1} \dots X_2 X_1,
 \end{aligned} \tag{4.6}$$

де  $X_i$  – вхідні двійкові змінні;

$Y_i$  – вихідні логічні функції, що являють собою мінтерми  $n$  змінних.

Дешифратори класифікують за такими ознаками:

- способом структурної організації – одноступеневі (лінійні) й багатоступеневі (пірамідальні, прямокутні або матричні);
- форматом вхідного коду – двійкові, двійково-десяткові;
- розрядністю коду, який дешифрується – 2,3,  $n$ ;
- кількістю входів – повні й неповні дешифратори;
- типом використовуваних логічних елементів – І, АБО, НІ-І, НІ-АБО і т.д.

Логічна функція дешифратора позначається буквами DC (decoder). Умовні графічні позначення дешифраторів на електричних схемах показані на рисунку 4.11. У схемах дешифраторів вбудовуються один або два стробуючих (дозволяючих) входи, наприклад  $W$ . Крім того, вхід  $W$  використовується для нарощування розрядності вхідного коду. Повний дешифратор – на  $n$  входів і  $m$  виходів для стислості називають дешифратором „з  $n$  в  $m$ ” (активізується одна з  $m$  вихідних ліній).

На об’єктах електричного транспорту в багатьох пристроях візуального спостереження контрольованих величин використовуються індикатори на основі електролюмінісцентних приладів, рідких кристалів або світлодіодних матриць. Роботу цих приладів можна організувати тільки за допомогою дешифраторів. Фрагмент схеми такого пристрою ілюструє рисунок 4.12.

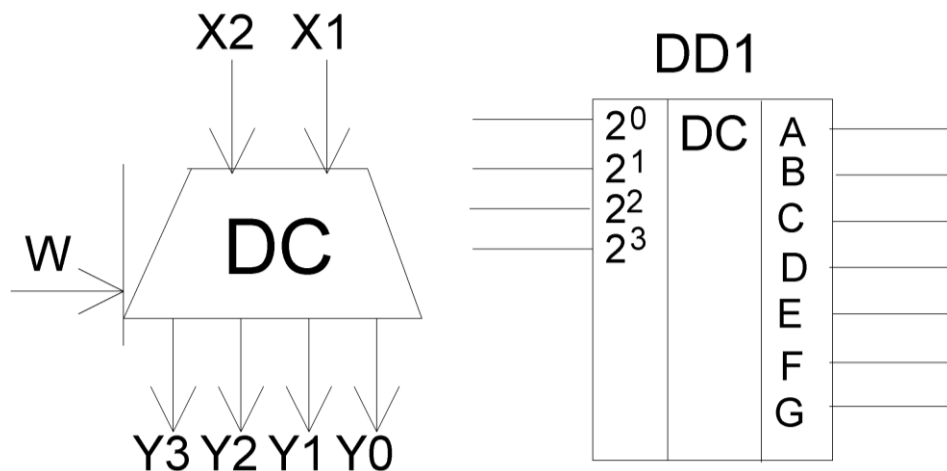


Рисунок 4.11 – Умовні графічні позначення дешифратора: DC – на функціональних схемах; DD1 – на принципових схемах

Включення семи сегментів електролюмінесцентного індикатора залежить від рівнів сигналів на вході дешифратора. Всі можливі варіанти формування «оптичних сигналів» або відображення десяткових цифр на світловому табло розглянуті в таблиці 4.7.

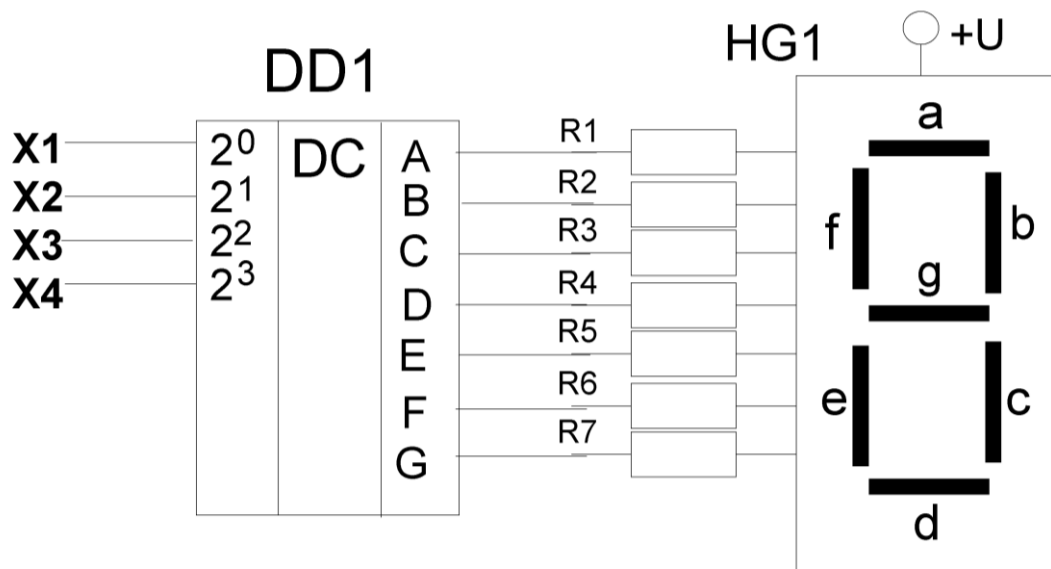


Рисунок 4.12 – Схема застосування дешифратора з оптичним індикатором

Таблиця 4.7 – Включення оптичного індикатора

Вхідні сигнали				Сегменти							Десяткова цифра
X4 $2^3$	X3 $2^2$	X2 $2^1$	X1 $2^0$	A	B	C	D	E	F	G	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	1	0	1	1	9

Дешифратори використовують у всіх мікропроцесорних пристроях. Вони застосовуються для дешифрації коду операції і адреси операнда, розташованих в регістрах команд процесорів МПП. Дешифрація коду операції в МПП визначає тип машинної команди. Дешифрація адреси операнда в оперативній пам'яті забезпечує доступ до вказаної комірки пам'яті для записування або зчитування даних.

Застосовуючи дешифратори в цифрових пристроях автоматики, слід пам'ятати, що всі вхідні змінні  $X1$ ,  $X2$ ,  $Xn$  подаються паралельно на входи дешифраторів.

Приклад фрагмента схеми автоматики мийно-прибирального корпусу (далі – МК) депо з використанням дешифратора для формування сигналів керування рухомими одиницями, що подаються для обслуговування, наведений на рисунку 4.13.

У системі автоматики МК передбачені формувачі сигналів S1 і S2, які взаємопов'язані з постами обслуговування рухомих одиниць (далі – РО). Таблиця істинності (табл. 4.8) ілюструє формування керуючих величин, використовуваних у наведеному фрагменті схеми для сигналізації.

Таблиця 4.8 – Таблиця істинності пристрою автоматики

X1	X2	Y1	Y2	Y3	Y4
0	0	1	0	0	0
0	1	0	0	1	0
1	0	0	0	0	1
1	1	0	1	0	0

Очевидно, що прийняті кольори джерел світла (Yi) забезпечують певну логіку функціонування МК. Такі системи знайшли широке застосування на різних об'єктах електричного транспорту.

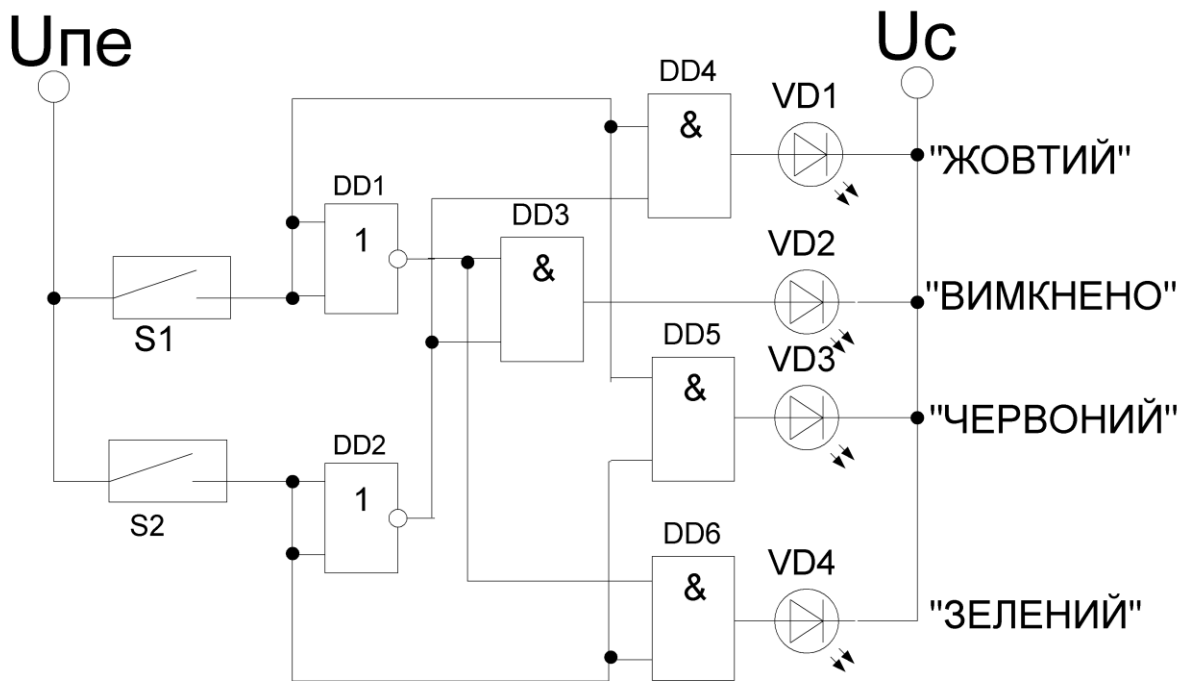


Рисунок 4.13 - Використання дешифратора в системі автоматики МК депо

### 4.3.2 Шифратори

*Шифратором* називається функціональний вузол МПП, призначений для перетворення вхідного  $m$ -розрядного унітарного коду у вихідний  $n$ -розрядний двійковий позиційний код. Двійкові шифратори виконують функцію, обернену функції дешифратора. При активації однієї з вхідних ліній дешифратора на

його виходах формується код, який відображає номер активного входу. Повний двійковий шифратор має  $m = 2^n$  входів і  $n$  виходів. Умовні графічні позначення шифраторів на схемах показані на рисунку 4.14.

Одним з основних застосувань шифратора – є ввід даних з клавіатури, наприклад, десяткових цифр. Клавіатура використовується для вводу текстів програм. Шифратори, які при одночасному натисканні декількох клавіш виробляють код тільки старшої цифри, називаються пріоритетними. Пріоритетні шифратори, які призначені для пошуку старшої (лівої) одиниці в слові та формування на виході двійкового номера шуканого розряду, називаються покажчиками старшої одиниці. Їх застосовують у пристроях нормалізації чисел в системах з пріоритетним обслуговуванням запитів на переривання роботи МПП.

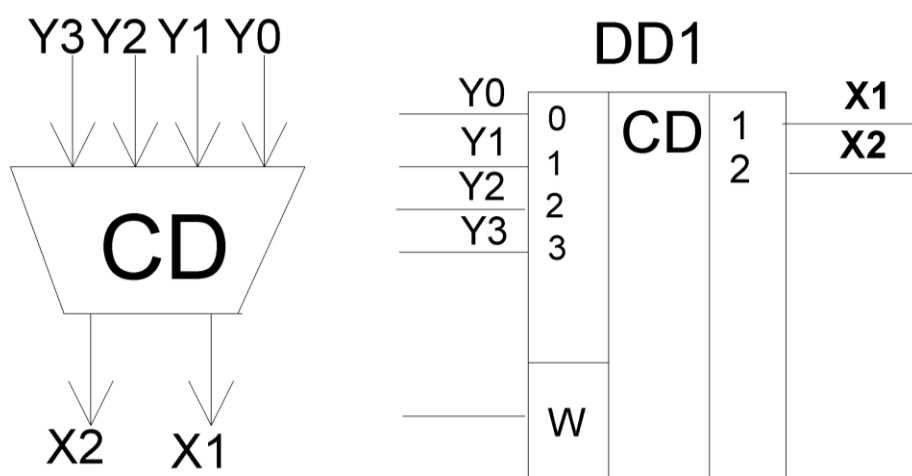


Рисунок 4.14 – Умовні графічні позначення шифратора: CD – на функціональних схемах; DD1 – на принципових схемах

### 4.3.3 Мультиплексори

*Мультиплексор* – функціональний вузол МПП, призначений для почергової комутації (перемикання) інформації від одного з  $n$  входів на загальний вихід. Номер конкретної вхідної лінії, що підключається до виходу в кожний такт машинного часу, визначається адресним кодом  $A_0, A_1, \dots, A_{m-1}$ . Зв'язок між числом інформаційних  $n$  і адресних  $m$  входів визначається співвідношенням  $n = 2^m$ . Таким чином, мультиплексором реалізує керовану передачу даних від кількох вхідних ліній в одну вихідну.

Приклад реалізації мультиплексора на логічних елементах показаний на рисунку 4.15. Сигнал на вході мультиплексора залежить від стану адресного входу  $A$ . Залежність вихідного сигналу  $Y$  від рівнів вхідних величин  $X_1, X_2, A$  ілюструє таблиця 4.9.

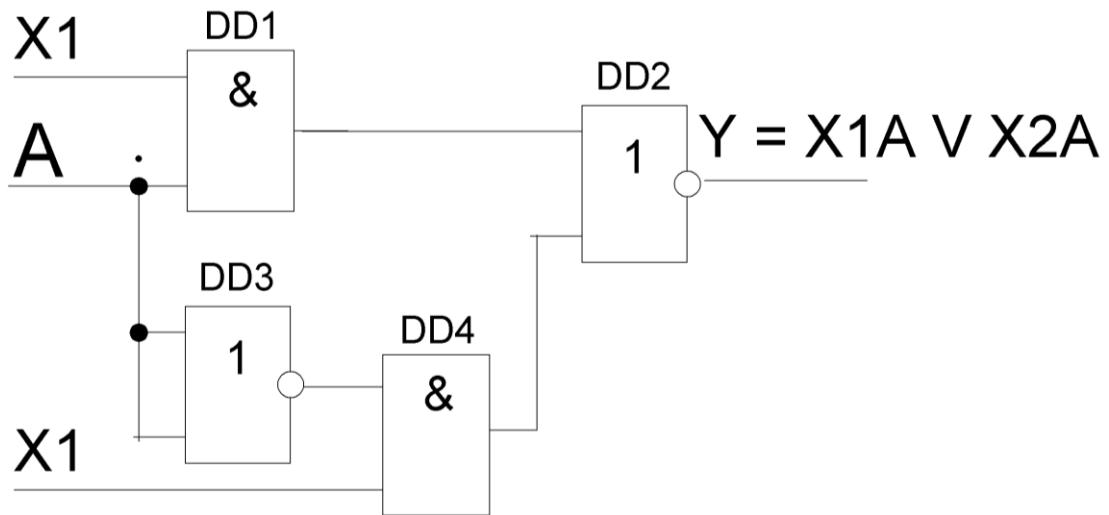


Рисунок 4.15 – Логічна структура мультиплектора типу 2 : 1

Таблиця 4.9 – Залежність вихідних сигналів мультиплектора від вхідних

X1	X2	A	Y
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Умовне графічне позначення мультиплексорів показано на рисунку 4.16. Функція мультиплексорів записується буквами MUX (multiplexor).

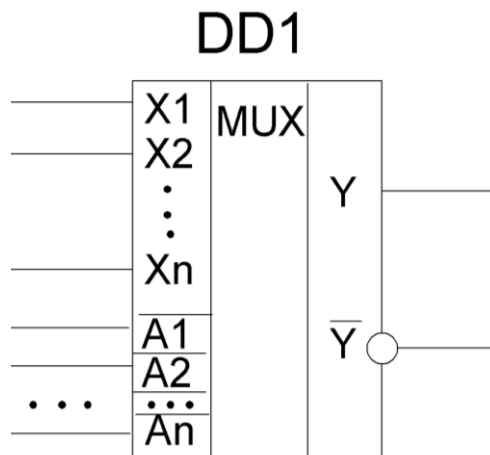


Рисунок 4.16 – Умовне графічне позначення мультиплектора

Мультиплектори застосовують для таких операцій: комутації як окремих ліній, так і груп ліній (шин); перетворення паралельного коду в послідовний; реалізації логічних функцій; побудови схем порівняння, генераторів кодів.

#### 4.3.4 Демультимплектори

*Демультимплектор* – функціональний вузол МПП, призначений для комутації (перемикання) сигналу з одного інформаційного входу  $D$  на один з  $n$  інформаційних виходів. Номер виходу, на який в кожний такт машинного часу передається значення вхідного сигналу, визначається адресним кодом  $A_0, A_1, A_2, \dots, A_{m-1}$ . Адресні входи  $m$  та інформаційні входи  $n$  пов'язані співвідношенням  $n = 2^m$  або  $m = \log_2 n$ .

Демультимплектор виконує функцію, обернену функції мультиплектора. Стасовно мультиплексорів і демультимплекторів користуються також терміном «селектори» даних.

Сигнал на виході демультимплектора залежить від стану адресного входу  $A$ . Приклад схеми демультимплектора на базі логічних елементів поданий на рисунку 4.17.

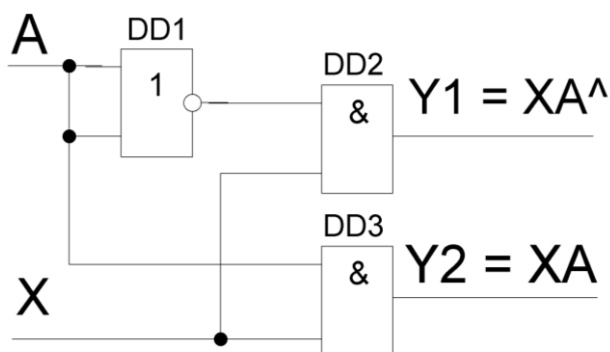


Рисунок 4.17 – Логічна структура демультимплектора типу 1 : 2

Залежність вихідних сигналів  $Y$  і від рівнів вхідних величин  $X$ ,  $A$  ілюструє таблиця 4.10.

Таблиця 4.10 – Залежність вихідних сигналів мультиплектора від вхідних

X	A	Y1	Y2
0	0	0	0
0	1	0	0
1	0	1	0
1	1	0	1

В умовних графічних позначеннях (рис. 4.18) функція демультимплектора позначається буквами DMX (demultipexor).

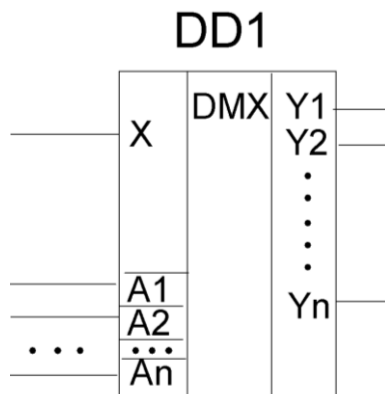


Рисунок 4.18 – Умовне графічне позначення демультиплексора

Демультиплексори використовують для таких операцій:

- комутації як окремих ліній, так і багаторозрядних шин;
- перетворення послідовного коду в паралельний;
- реалізації логічних функцій та інших.

Демультиплексори часто позначають: «1 → n».

Приклади, що характеризують роботу пристроїв автоматики реалізованих на логічних елементах, можна знайти на сторінках електронної лабораторії програми SinSys (**ПРЕ-4Ц, ПрЕ-Дз, ПРЕ-МД**).

#### 4.3.5 Компаратори

*Компаратор* – це пристрій порівняння. Контроль і корекція результатів операцій є важливою умовою грамотної експлуатації устаткування на транспорті. Контроль може бути програмним або апаратним. До апаратних методів відносяться дублювання операцій і відновлення вхідних сигналів. Схеми порівняння призначені для вироблення ознак відношень між двійковими словами (числами).

Ознаки відношень записують у вигляді:

$$F_i := A * K \text{ або } F_{A*K}; \quad (4.7)$$

$$F_i := A * B \text{ або } F_{A*B}, \quad (4.8)$$

де  $A$  і  $B$  – двійкові або двійково–десяткові числа;

$K$  – двійкова константа;

$i$  – номер відношення;

$*$  – операція відношення вигляду  $=$ ,  $>$ ,  $<$ ,  $>=$ ,  $<=$  та ін.;

$F_i$  – функція, що задає результат відношення (1 – якщо відношення виконується, тобто істинне, 0 – якщо відношення не виконується, тобто помилкове).

Функція компаратора позначається буквами COMP (comparator) або знаками «= =»

Основними відношеннями вважаються «рівне», «більше», «менше». Схеми, що реалізують ці відношення, називають схемами порівняння «на більше» або «на менше».

Фрагмент схеми пристрою автоматики тягової підстанції електричного транспорту ілюструє роботу системи максимального струмового захисту (далі – МСЗ). Схема реалізована за допомогою однорозрядного компаратора (рис. 4.19).

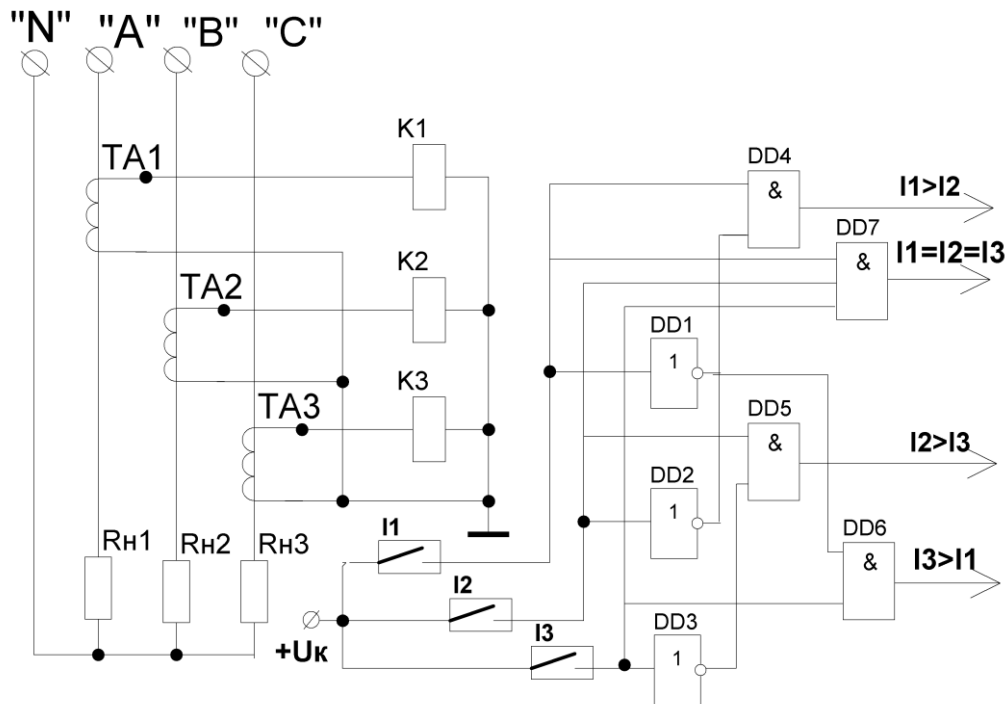


Рисунок 4.19 – Фрагмент схеми системи МСЗ с компаратором на ЛЕ

МСЗ застосовується для контролю фазних струмів. У схемі використана гальванічна розв'язка між джерелом високої напруги (А, В, С) і пристроєм автоматики ( $U_K$ ). Логічна схема МСЗ забезпечує надійний контроль струмів  $I_1$ ,  $I_2$ ,  $I_3$  і відповідне відношення між ними:  $I_1 > I_2$ ,  $I_2 > I_3$ ,  $I_3 > I_1$ ,  $I_1 = I_2 = I_3$ . Порівняння цих величин може бути розширено при відповідному доопрацюванні пристрою. Очевидно, що ці технічні рішення знаходять широке розповсюдження на транспорті, де часто реалізуються залежності, визначувані відношенням поточних значень величин з їх заданими пороговими значеннями.

У подібних схемах, коли, наприклад, два  $n$  – розрядних числа дорівнюють, коли попарно дорівнюють між собою всі розряди цих чисел. Якщо числа  $A$  і  $B$  4 – розрядні, то ознакою їх рівності буде виконання умови  $A_3 = B_3$ ;  $A_2 = B_2$ ;  $A_1 = B_1$ ;  $A_0 = B_0$ , де  $A_0$  і  $B_0$  – молодші розряди.

Застосовуючи елемент порівняння для кожної пари відповідних розрядів, факт рівності обох чисел  $A = B$  буде встановлено, коли  $E = E_3 E_2 E_1 E_0 = 1$ . Якщо  $E = 0$ , то  $A$  не рівне  $B$ .

Порівняння може бути реалізоване й на інших схемотехнічних принципах.



Умовне графічне позначення  $n$  – розрядного компаратора ілюструє рисунок 4.20.

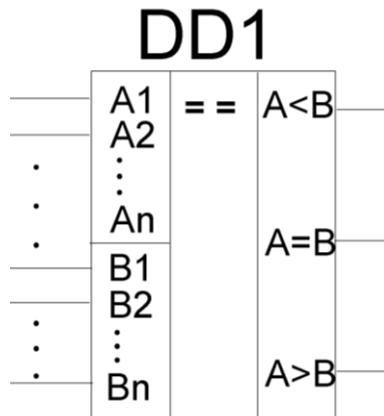


Рисунок 4.20 – Умовне графічне позначення  $n$  – розрядного компаратора

#### 4.3.6 Підсилювачі

*Підсилювачами* електричних сигналів називаються операційні підсилювачі (далі – ОП), призначені для виконання різних операцій над електричними величинами. У пристроях автоматики підсилювачі виконують перетворення сигналів, нормалізацію їх рівнів, обчислювальні дії, узгодження вузлів з різними електричними параметрами та ін. Широка номенклатура ОП, використовуваних в цифровій автоматичі, підтверджує їх невід’ємний зв’язок з МПП. Вихідні струми ОП не перевищують 5 мА, а підсилювачі потужності дозволяють керувати багатьма виконавчими елементами на об’єктах електричного транспорту, забезпечуючи вихідні струми, визначувані сотнями і тисячами ампер.

Інтегральні ОП використовують у пристроях генерування, фільтрації, модуляції і демодулювання сигналів спільно з мікроконтролерами та іншими МПП, вживаними в автономних пристроях автоматики безпосередньо на об’єктах і рухомих одиницях електричного транспорту.

На електричних принципових схемах підсилювачі зображаються як показано на рисунку 4.21.

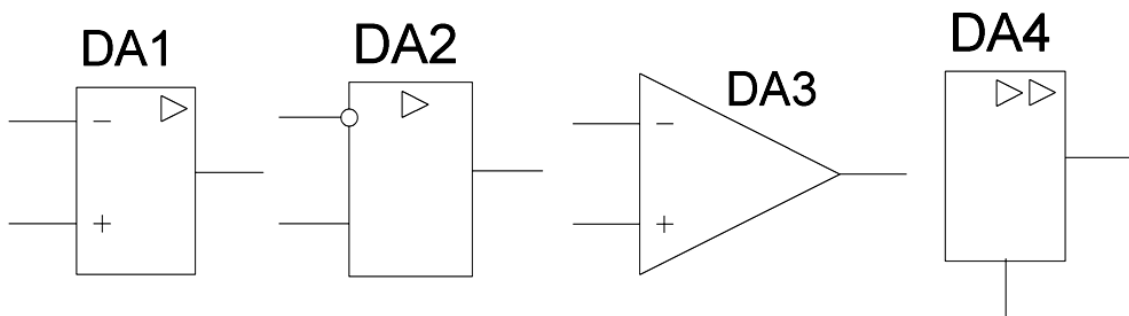


Рисунок 4.21 – Умовне графічне позначення підсилювачів:  
*DA1 – DA3 – напруги; DA4 – потужності*

На рисунку 4.22 представлений фрагмент схеми керування електричним двигуном М з живленням від джерела змінної напруги.

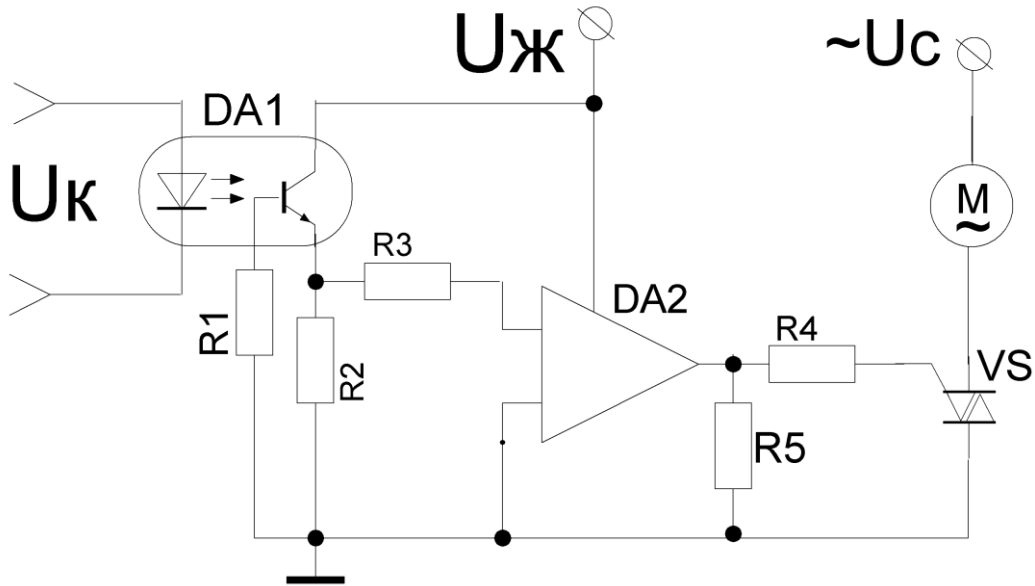


Рисунок 4.22 – Схеми керування ВЕ з використанням симістора

У схемі оптопара  $DA1$  формує інформаційний дискретний сигнал, який перетворюється підсилювачами напруги  $DA2$ . Комутатор ланцюга живлення електродвигуна реалізований на симісторі  $VS$ .

На рисунку 4.23 зображений підсилювач, використовуваний для підключення електричної лампи  $EL1$ . Даний пристрій відрізняється застосуванням гібридної схеми, в якій підсилювачі напруги і потужності реалізовані відповідно на інтегральній мікросхемі  $DA2$  і транзисторі  $VT1$ .

Розглянуті схеми знайшли широке застосування на різних об'єктах електричного транспорту. Дуже важливою їх рисою є реалізація гальванічної розв'язки між пристроєм автоматики і потужним виконавчим елементом. Доцільність таких схем обумовлена підвищеними вимогами до захисту МПП від дії зовнішніх електричних ланцюгів.

Програми для розрахунку реальних підсилювачів напруги і потужності представлені в Windows-додатку SinSys (ПрЕ-1МР, ПрЕ-2УН, ПрЕ-3УМ).

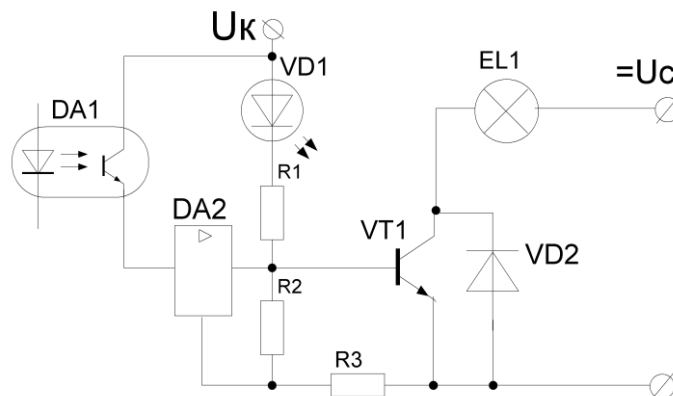


Рисунок 4.23 – Схеми підключення навантаження до оптодатчику

#### 4.4 Контрольні питання

1. Перерахувати особливості системи логічних елементів.
2. Поясніть роботу логічних елементів АБО, І, НІ.
3. Поясніть роботу асинхронного RS-тригера.
4. Поясніть роботу синхронного RS-тригера.
5. Поясніть, чому JK-тригер називають універсальним.
6. Укажіть на відмінності між тригерами типів JK, D, T.
7. Що таке послідовнісні пристрої?
8. Поясніть роботу лічильника й дільника.
9. Поясніть області застосування лічильників у МПП.
10. Поясніть будову і роботу регістрів у МПП.
11. Що таке комбінаційні пристрої?
12. Поясніть призначення і будову дешифратора.
13. Наведіть класифікацію дешифраторів.
14. Поясніть відомі приклади застосування дешифраторів у МПП.
15. Поясніть призначення шифраторів у МПП.
16. Що таке мультиплексор?
17. Поясніть роботу пристрою з мультиплексором.
18. Що таке демультиплексор?
19. На прикладі поясніть доцільність використання демультиплексора.
20. Що таке компаратор?
21. Де використовуються компаратори на транспорті?
22. Поясніть назначення підсилювачів у МПП.
23. Наведіть приклад нормалізатора інформаційного сигналу з підсилювачем.
24. Поясніть призначення підсилювачів у пристроях дискретного керування потужними виконавчими елементами на транспорті.
25. Запропонуйте схему передачі  $2^n$  команд керування на  $n$  ВЕ, використовуючи логічні елементи.

## 5 ОСНОВИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ АВТОМАТИКИ

### 5.1 Арифметичні основи мікропроцесорних пристроїв

#### 5.1.1 Принципи побудови систем числення

Числова інформація в мікропроцесорних пристроях автоматики характеризується: системою числення (двійкова, десяткова та ін.); видом числа (числа дійсні, комплексні, масиви); типом числа (змішане, ціле, дробове); формою представлення числа (місце коми) – з природною (змінною), фіксованою, плаваючою комами; розрядною сіткою і форматом числа; способом кодування чисел; алгоритмами виконання арифметичних операцій.

Системою числення називається сукупність цифр і правил для записування чисел. Запис числа у деякій системі числення називається його кодом. Усі системи числення поділяються на позиційні й непозиційні. Для запису чисел у позиційній системі числення використовують певну кількість графічних знаків (цифр і букв), які відрізняються один від одного. Число таких знаків  $q$  називається основою позиційної системи числення. В МПП використовують позиційні системи з різною основою.

Число в позиційній системі можна представити поліномом:

$$A_M = a_k q^k + a_{k-1} q^{k-1} + \dots + a_0 q^0 + a_{-1} q^{-1} + \dots + a_{-m} q^{-m}, \quad (5.1)$$

де  $q$  – основа системи числення;

$q^i$  – вага позиції;

$a_i$  – цифри в позиціях числа;

$0, 1, \dots, k$  – номери розрядів цілої частини числа;

$-1, -2, \dots, -m$  – номери розрядів дробової частини числа.

Позиційні системи з однаковою основою в кожному розряді називаються однорідними. Оскільки на значення  $q$  немає ніяких обмежень, то теоретично можлива нескінченна множина позиційних систем числення. Використовуючи ексклюзивний калькулятор програми SinSys (електронна лабораторія САК-1t) можна провести експерименти з перекладу десяткових чисел в будь-які задані системи числення, які матимуть свої назви. Наприклад: двійкова (BIN) – система числення з основою 2 (цифри 0 і 1); шістнадцяткова (HEX) – система числення з основою 16 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

У позиційних системах числення значення кожної цифри визначається її зображенням і позицією або розрядом в числі. *Номер позиції* – номер розряду. *Число позицій* – називається розрядністю числа і збігається з довжиною числа.

У непозиційних системах числення значення кожної цифри не залежить від її позиції. Найвідомішою непозиційною системою є римська (I, V, X, L, C, D, M), наприклад: III – 3; LIX – 59; DLV – 555.

Недоліком непозиційної системи є відсутність нуля і формальних правил запису чисел і відповідно арифметичних дій з ними.

Система числення повинна забезпечувати:

1. Можливість представлення будь-якого числа в заданому діапазоні.
2. Однозначність, стислість запису числа і простоту виконання арифметичних операцій.
3. Досягнення високої швидкодії машини в процесі обробки інформації.

### 5.1.2 Переведення чисел з однієї системи в іншу

Для переведення цілого числа з однієї системи числення в іншу необхідно поділити початкове число на нову основу за правилами: одержана перша остача є значенням молодшого розряду в новій системі, а першу частку необхідно знову ділити. Цей процес продовжується до появи неподільної частки. Результат записують у порядку, оберненому їхньому одержанню.

Приклад: початкове число 14 (DEC) необхідно перевести в двійкову (BIN) систему:

$$14 : 2 = 7 + 0 \text{ (остача);}$$

$$7 : 2 = 3 + 1;$$

$$3 : 2 = 1 + 1;$$

$$1 : 2 = 0 + 1.$$

Результат:  $14D \Rightarrow 1110B$ .

Для переведення правильного дробу з однієї системи числення в іншу необхідно, діючи за правилами початкової системи, помножити перевідне число на основу нової системи; від результату відокремити цілу частину, а дробову частину, що залишилася, знову помножити на цю основу. Процес повторюється до одержання заданого числа цифр. Результат записують як цілі частини добутку в порядку їхнього одержання.

Приклад: початкове число 0,625 необхідно перевести у двійкове число з точністю до четвертого знаку:

$$0, \quad 625 \\ \quad \quad 2 \quad \quad 0;$$

$$1 \quad 250 \\ \quad \quad 2 \quad \quad 1;$$

$$0 \quad 500 \\ \quad \quad 2 \quad \quad 0;$$

$$1 \quad 000 \\ \quad \quad 2 \quad \quad 1;$$

$$0 \quad 000 \quad 0.$$

Результат:  $0,625D \Rightarrow 0,1010B$ .

Для переведення змішаних чисел у двійкову систему потрібно окремо переводити їхні цілу і дробову частини.

## 5.2 Арифметичні операції

Всі операції в МПП виконуються в арифметико–логічному пристрої (далі – АЛП). Числа, які беруть участь в операціях, називаються *операндами*. Основною операцією в АЛП є додавання. Операція віднімання замінюється додаванням операндів. Операції множення і ділення зводяться до багатократних додавань і зсувів (табл. 5.1).

Таблиця 5.1 – Операції в АЛП

Додавання	Віднімання	Множення	Зсув вправо	Зсув вліво
$0 + 0 = 1$	$0 - 0 = 0$	$0 * 0 = 0$	0100	0100
$0 + 1 = 1$	$1 - 0 = 1$	$0 * 1 = 0$	0010	1000
$1 + 0 = 1$	$1 - 1 = 0$	$1 * 0 = 0$	-	-
$1 + 1 = 10$	$0 - 1 = 11$	$1 * 1 = 1$	-	-

Операція додавання в АЛП виконується функціональним вузлом – суматором. Він складається з окремих логічних схем, які називаються однорозрядними суматорами. Операнди надходять з виходів регістрів операндів.

Машинні зображення від’ємних операндів одержують за правилами представлення чисел, наприклад, в оберненому коді (табл. 5.2). Прямий, обернений і доповняльний коди використовують для записування знака числа, заміни операції віднімання чисел додавання їхніх кодів, а також для визначення переповнення розрядної сітки. Для представлення знака числа у них відводиться знаковий розряд, який розташовується зліва від числа і відділяється комою. У знаковий розряд записується нуль – для позитивного числа і одиниця – для негативного.

Операція множення чисел складається з декількох циклів. Цикли завершуються зсувом на один розряд. Наприклад, зсув вліво на один розряд еквівалентний множенню на два.

Для простоти виконання арифметико–логічних операцій в МПП є функціональний вузол призначений для перетворення, наприклад, двійкового коду з однієї форми в іншу.

Таблиця 5.2 – Прямий, обернений і додатковий коди

Коди для додатних чисел				Коди для від’ємних чисел			
DEC	прямий	обернений	додатковий	DEC	прямий	обернений	доповняльний
+0	0,000	0,000	0,000	-0	1,000	1,111	0,000
+1	0,001	0,001	0,001	-1	1,001	1,110	1,111
+2	0,010	0,010	0,010	-2	1,010	1,101	1,110
+3	0,011	0,011	0,011	-3	1,011	1,100	1,101

## 5.3 Контрольні питання

1. Що таке система числення?
2. Які існують системи числення?
3. Поясніть принцип створення позиційних систем числення.
4. Поясніть спосіб перетворення десяткового коду в бінарний.
5. Переведіть задане (DEC) число в BIN систему і перевірьте результат.
6. Переведіть задане (HEX) число в BIN систему і перевірьте результат.
7. Що таке операнд?
8. Поясніть функції АЛП в МПП.
9. Поясніть операції додавання, віднімання, множення, зсуву.
10. Поясніть призначення прямого і оберненого кодів.

## 6 МІКРОПРОЦЕСОРНА АВТОМАТИКА НА ТЕХНОЛОГІЧНОМУ ОБ'ЄКТІ

### 6.1 Архітектура і структура мікропроцесора системи керування

*Архітектура* – це найбільш загальні принципи побудови МПП автоматики. Вона охоплює питання: функціональний склад технічних і програмних засобів та їхню взаємодію у процесі обробки інформації і формування керуючих величин, що впливають на технологічний об'єкт.

*Мікропроцесорною системою автоматики* називають будь-яку обчислювальну, інформаційно-керуючу або керуючу систему, центральним пристроєм обробки інформації якої є мікропроцесор.

У загальному вигляді структуру системи автоматичного керування технологічним об'єктом можна зобразити схемою, що ілюструє взаємозв'язок технологічного об'єкта з мікропроцесорним пристроєм автоматики, оснащеним необхідним периферійним обладнанням (рис. 6.1).

*Технологічний об'єкт* (далі – ТО) *керування* – сукупність технологічного обладнання і реалізованого на ньому з відповідними інструкціями технологічного процесу.

МПП виконує функції збору даних про стан ТО (датчики фізичних величин *ПЕ1*, *ПЕ2*, *ПЕп*), регулювання і керування процесу (за допомогою виконавчих пристроїв *ВЕ1*, *ВЕ2*, *ВEm*), відображення інформації, оптимізацію технологічних величин та ін.

Схема на рисунку 6.1 реалізується за допомогою мікропроцесорного пристрою, використання якого дозволяє:

- замінити аналогові технічні засоби об'єктів транспорту на цифрові там, де перехід до цифрового обладнання підвищує точність і розширює функціональні можливості системи керування;

- замінити технічні засоби з «жорсткою» логікою на програмні («гнучкі»), тобто які передбачають можливості зміни їх внутрішньої структури;

- забезпечити перехід до децентралізованого керування ТО з підвищеною надійністю.

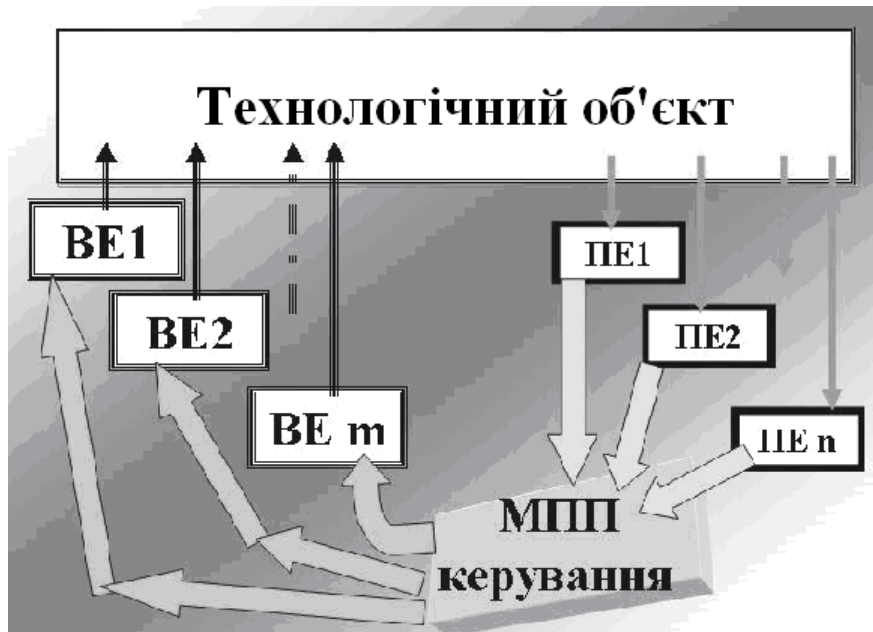


Рисунок 6.1 – Структура мікропроцесорної системи керування

МПП на об'єктах електричного транспорту забезпечують декілька основних режимів функціонування.

*Автоматизована інформаційна система* – це найбільш проста форма. Результати обчислень при такому режимі відображаються на екрані відеотерміналу, друкуються, відтворюються будь-яким способом.

*Радник-оператор* – МПП працює в розімкненому контурі, тобто система забезпечує тільки відображення інформації про стан об'єктів транспорту. Оператор (диспетчер) може прийняти рішення на основі рекомендацій МПП або проігнорувати їх. Регулювання параметрів здійснюють локальні пристрої автоматики або сам оператор через пристрої ручного керування.

*Супервізорне керування* – МПП працює в замкнутому контурі (рис. 6.1). Оператор тільки спостерігає за її діями і втручається лише при відмовах у роботі пристроїв системи керування.

*Безпосередньо цифрове керування* – найбільш складний режим, при якому вихідні сигнали від МПП надходять прямо на ВЕ (рис. 6.1). Необхідність оператора в такій системі повністю виключена.

Перехід від централізованих систем керування до децентралізованих пов'язаний з ускладненням ТО, підвищенням вимог до їх швидкодії та надійності в роботі, зниженням вартості й експлуатаційних витрат.

*Функціональна децентралізація* – це розподіл складного процесу або системи на підсистеми за функціональними ознаками, що мають самостійну мету функціонування.

*Топологічна децентралізація* означає територіальний розподіл процесу або системи на локальні підсистеми.

Оскільки розподілені системи мають щонайменше дві функціонально пов'язані підсистеми, то внаслідок децентралізації утворюються багатопроцесорні системи керування.



У розподілених системах керування прийняті три основні топологічні структури організації взаємодії підсистем: радіальна, кільцева, магістральна.

## 6.2 Пристрої на базі комп'ютерів і мікро-ЕОМ

Більшість перших мікропроцесорних пристроїв автоматики були створені на базі комп'ютерів. Основні принципи проектування комп'ютерів були запропоновані в 1946 р. американськими математиками Дж. Нейманом, К. Голдстайном і А. Берксом. Загалом нейманівська архітектура має такі основні ознаки:

- наявність одного обчислювача, що має процесор, пам'ять, засоби вводу-виводу інформації (периферійні пристрої) та керування;
- використання двійкової системи числення як для представлення інформації, так і для виконання арифметико-логічних операцій;
- розміщення в єдиній спільній пам'яті команд і чисел фіксованої довжини;
- лінійну структуру адресації комірок пам'яті, що вимагає наявності в процесорі лічильника команд;
- централізоване автоматичне послідовне зчитування команд із пам'яті та інтерпретацію їх процесором.

Класична архітектура комп'ютера вміщує (рис. 6.2):

- арифметико-логічний пристрій (АЛП);
- оперативну пам'ять (далі – ОП);
- засоби зберігання і вводу-виводу інформації: зовнішні запам'ятовуючі пристрої (далі – ЗЗП); пристрої вводу інформації (далі – ПВВ); пристрої виводу інформації (далі – ПВив);
- пристрої керування (далі – ПК).

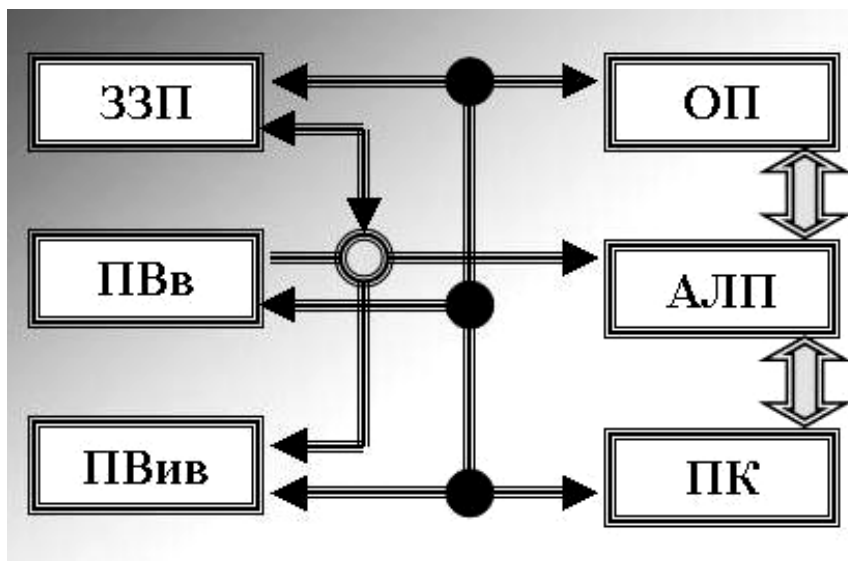


Рисунок 6.2 – Класична архітектура комп'ютера

Арифметико-логічний пристрій призначений для виконання арифметичних і логічних операцій, передбачених системою команд даного

МПП. До складу АЛП входять регістри і комбінаційні схеми. Дані (операнди) для обробки в АЛП надходять з ОП.

Пристрій керування зчитує і дешифрує у відповідній послідовності команди, формує і подає керуючі сигнали для інших пристроїв МПП. До ПК віднесено і пульт оператора.

*Оперативна пам'ять* (далі – ОЗП, RAM – random access memory) – призначена для часового зберігання програм і даних, в ній виконуються операції записування і читання інформації. Крім ОП, використовують також постійну пам'ять (далі – ПЗП, ROM – real only memory) і регістри АЛП. ОЗП, ПЗП і АЛП називають внутрішньою пам'яттю МПП (рис. 6.3). Процесор і ОП разом створюють ядро комп'ютера у будь-якому МПП автоматички.



Рисунок 6.3 – Внутрішня пам'ять МПП

*Операції вводу–виводу* – це обмін інформацією між ядром машини і ПП. Операція вводу передає інформацію з ПП в ядро МПП, а операція виводу – навпаки.

Пристрої вводу і виводу інформації розглядають як єдину функціональну частину МПП. Різні за своїми функціями, принципами побудови й характеристиками ПВВ і ЗП разом створюють групу периферійних пристроїв.

До ПВВ інформації входять:

- клавіатура, призначена для вводу програм і даних та керування роботою машини;
- ручні маніпулятори – миша, джойстик;
- пристрої мовного вводу, призначені для керування МПП за допомогою мовних команд.

До ПВив інформації відносять:

- принтери (матричні, струминні, лазерні);
- дисплеї (на електронно-променевих трубках, на рідких кристалах, плазмові, люмінесцентні);
- плотери (графопобудовники), призначені для друкування, креслення;

- синтезатори звуку і мови, призначенні для перетворення аналогових сигналів у цифровий код і навпаки;
- пристрої виводу інформації.

Зв'язок між функціональними частинами МПП здійснюють за допомогою *інтерфейсу* – сукупності шин, сигналів, допоміжних мікросхем та алгоритмів, призначених для обміну інформацією між пристроями МПП.

Виділяють три шини (рис. 6.4):

- адреси (далі – ША), призначена для передачі адреси комірок ОП і регістрів ПП;
- даних (далі – ШД), призначена для передачі даних;
- керування (далі – ШК), призначена для передачі керуючих сигналів від процесора до пристроїв і навпаки.

Аналогічно влаштовані керуючі мікро-ЕОМ, до складу яких входять: мікропроцесор, підсистеми вводу-виводу сигналів, оперативний запам'ятовуючий пристрій, постійний запам'ятовуючий пристрій та інші функціональні блоки.

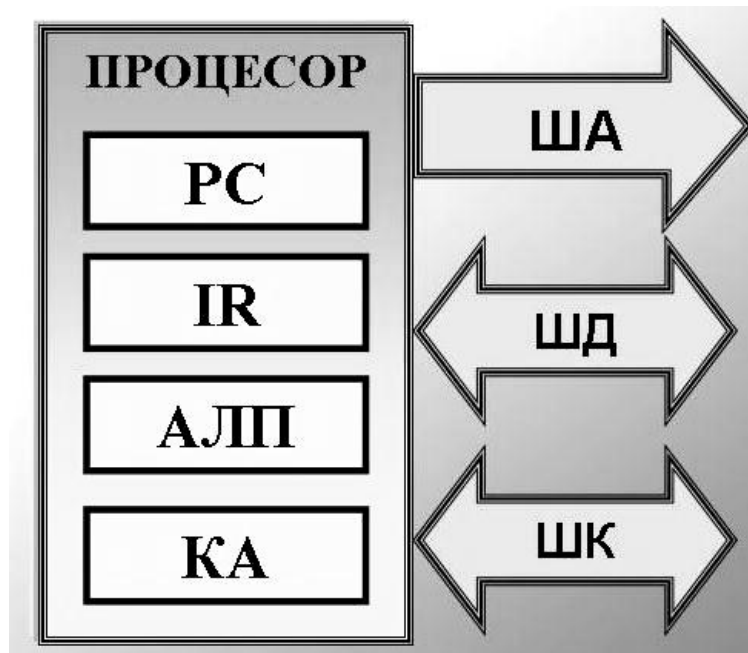


Рисунок 6.4 – Структура процесора

### 6.3 Структура мікроконтролера системи керування

Всі мікроконтролери побудовані за однією схемою. Система керування, що складається з лічильника команд і схеми декодування, виконує прочитування і декодування команд з пам'яті програм, а операційний пристрій відповідає за виконання арифметичних і логічних операцій.

*Інтерфейс вводу-виводу* дозволяє обмінюватися даними з периферійними пристроями.

Для зберігання програм і даних використовується пристрій пам'яті, який входить до складу будь-якого мікроконтролера. Пам'ять мікроконтролерів реалізована за Гарвардською архітектурою, що має на увазі розділення пам'яті

команд і даних. Це означає, що звернення до команд здійснюється незалежно від доступу до даних. Перевагою такої організації є підвищення швидкості доступу до пам'яті.

У ряді мікроконтролерів передбачений одночасний доступ до пам'яті даних і до пам'яті команд. Це суттєво підвищує швидкість обробки програм. Загальну структуру мікроконтролера ілюструє рисунок 6.5.

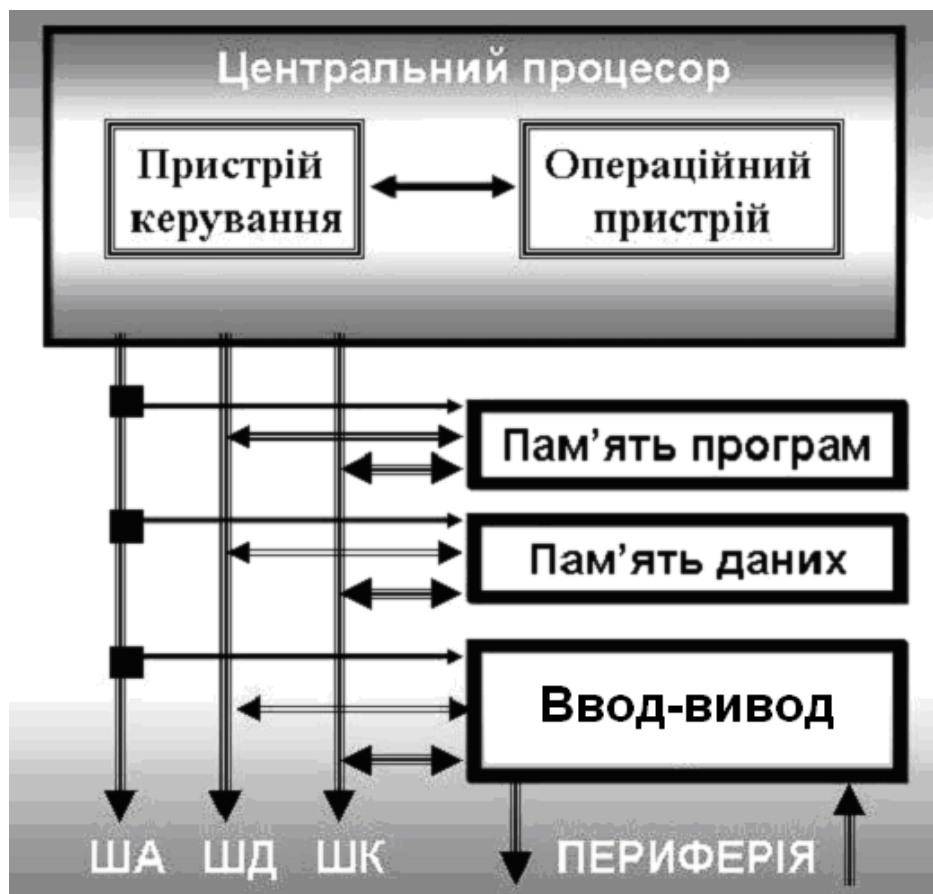


Рисунок 6.5 – Структура мікроконтролера

#### 6.4 Принцип програмного керування

У МПП реалізують принцип програмного керування. Для вирішення завдання автоматизації будь-якого технологічного об'єкта розробляють алгоритм на основі числових методів обчислення. Алгоритм переводиться на мову, властиву конкретному мікропроцесорному пристрою у вигляді програми – мовної конструкції, яка є впорядкованою послідовністю описів і команд, призначених для обробки інформації. Програма записується в Оп у вигляді машинних слів, що кодуються цифрами 0 і 1. Код операції поступає в регістр команд IR (Instruction Register) і потім дешифрується, а дані – в регістри АЛП (рис. 6.4).

Команди програми розміщені в Оп лінійно (одна за одною) і виконуються послідовно. Номер команд в Оп визначається програмним лічильником РС (Program Counter). Керуючий автомат (далі – КА) виробляє множину керуючих сигналів, що подаються на всі пристрої МПП. Регістр команд, програмний

лічильник і керуючий автомат входять до складу МПП автоматики. Послідовне керування зумовлене наявністю одного процесора.

У МПП використовують певний список команд, наприклад:

- пересилки даних між регістрами АЛП, регістрами і Оп;
- арифметичних операцій над двійковими числами;
- логічних операцій;
- установлення відношень – більше, менше та ін.;
- керування програмою;
- ввід–вивод даних між ядром МПП і ПП;
- перетворення з одного формату в інші.

МПП, вживані для автоматизації процесів і виробництв, часто мають невеликий набір команд, визначуваний декількома десятками.

МПП залежно від їх складності й призначення працюють в таких режимах (рис. 6.6):

- однопрограмному – програма завантажуються в Оп і виконується до одержання результату;
- багатопрограмному – в пам'ять завантажуються декілька програм. Коли виконання однієї з програм зупиняється, МПП переключається на виконання іншої програми;
- пакетному – формується пакет задач, які потім зчитуються в Оп групами і виконуються в багатопрограмному режимі;
- розподілу часу, але колективного користування – доступ до МПП автоматики користувачів за допомогою власного терміналу;
- реального часу – забезпечується взаємодія МПП із зовнішніми об'єктами в заданому темпі.



Рисунок 6.6 – Режими роботи МПП автоматики

Згідно з деревом мікроелектронних пристроїв (рис. 6.7) існує три підходи до архітектури МП. МПП на базі МП із складним набором команд CISC (Complex Instruction Set Computers) на транспорті звичайно використовуються на центральних диспетчерських пунктах. Такі пристрої реалізують багатопрограмні режими.



Рисунок 6.7 – Дерево мікропроцесорних пристроїв

На локальних об'єктах – тягових підстанціях, пунктах контролю стану контактної мережі, на рухомих одиницях та ін. знайшли застосування МПП з архітектурою RISC (Reduced Instruction Set Computers) – із скороченим набором команд. Такі МПП реалізують частіше за все однопрограмні режими. Основна перевага мікроконтролерів з такою архітектурою полягає в обмеженому наборі команд і, як наслідок, дуже високій швидкості вирішення завдань автоматизації технологічних процесів. Програмування завдань для таких пристроїв спрощується, що й забезпечило їх популярність у розробників мікропроцесорної техніки.

### 6.5 Контрольні питання

1. Дайте визначення архітектури МПП.
2. Дайте визначення структури МПП.
3. Дайте визначення технологічного об'єкта керування.
4. Поясніть режими функціонування МПП на ТО електротранспорту.
5. Поясніть нейманівську архітектуру МПП.
6. Поясніть архітектуру комп'ютера і мікро-ЕОМ.
7. Що таке внутрішня пам'ять МПП?
8. Що відноситься до пристроїв ввід інформації МПП?
9. Що таке пристрої вивод інформації?
10. Перерахувати і пояснити назначення шин в МПП.
11. У чому особливості архітектури мікроконтролера?
12. Охарактеризуйте принцип програмного керування.
13. Поясніть особливості МПП з CISC архітектурою і їх застосування на об'єктах транспорту.
14. У чому особливості МПП з RISC архітектурою і їх застосування на транспорті?
15. Поясніть режими роботи МПП автоматики.

## 7 ЕЛЕМЕНТИ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ АВТОМАТИКИ ТА ЇХ ХАРАКТЕРИСТИКИ

### 7.1 Процесори

Універсальні мікропроцесорні пристрої поділяються на три функціонально пов'язані апаратні частини: процесор, пам'ять, периферійні пристрої.

*Процесор* – це основна функціональна частина МПП, яка інтерпретує і виконує команди (реалізує програмно-керований процес обробки даних). Процесор для виконання основних функцій називають *центральним* (далі – ЦП). Спеціалізований процесор, призначений для керування зовнішніми пристроями (накопичувачами, дисплеями та ін.), називають *контролером*. *Пам'ять* – сукупність різних пристроїв, призначених для приймання, зберігання і видачі двійкової інформації. *Периферійні* – пристрої для формування вхідних і вихідних величин.

Процесор характеризується архітектурою, до якої відносять:

- список арифметико-логічних операцій (система команд);
- типи і формати команд і даних;
- організацію адресного простору пам'яті й периферійних пристроїв;
- способи адресації команд і даних;
- функції складових частин і структуру зв'язків з іншими компонентами

МПП та режими роботи.

Процесор складається з пристрою керування, арифметико-логічного пристрою та блоку інтерфейсу (далі – БІФ) для з'єднання із пам'яттю, периферійними компонентами (рис. 7.1).

Обробка даних здійснюється в арифметико-логічному пристрої, який містить арифметико-логічний блок (далі – АЛБ), блок контролю (далі – БК) і місцевий пристрій керування. До складу процесора входить також регістр загального призначення (далі – РЗП). Через БІФ реалізується взаємодія з оперативною пам'яттю.

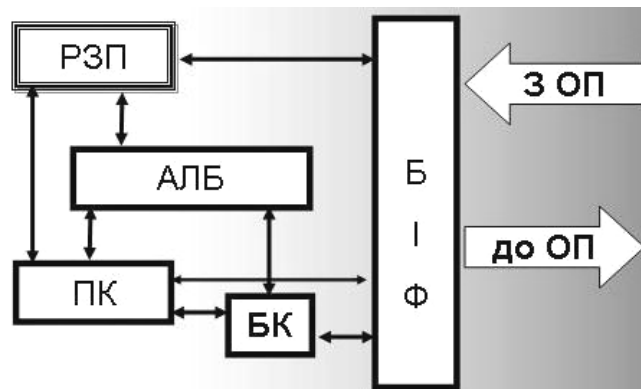


Рисунок 7.1 – Структура процесора

У РЗП зберігаються початкові дані, проміжні й кінцеві результати, адреси даних, константи, які необхідні в процесі виконання команд. Всі операції в

АЛП реалізуються як послідовності мікрооперації над двійковими словами, кожна з яких є сукупністю булевих операцій над бітами слів. В АЛП реалізуються такі мікрооперації:

- передачі слів між регістрами і регістрами й пам'яттю;
- додавання двох слів, декремент (-1) або інкремент (+1) слова;
- арифметичні, логічні й циклічні зсуви вправо чи вліво;
- логічні операції АБО, І та ін.;
- перетворення кодів слів.

ПК керує процесом обробки даних і взаємодією всіх пристроїв МПП. ПК має регістр і дешифратор команд, програмний лічильник для завдання адреси наступної команди, блок керування і схеми синхронізації, діагностики й контролю.

ПК послідовно зчитує код команди з пам'яті і розміщує його в регістр команд (інструкцій). БК дешифрує команду і формує послідовності керуючих сигналів. Для виконання однієї мікрооперації в АЛП необхідний один керуючий сигнал. В одному машинному такті реалізується сукупність мікрооперацій–мікрокоманд. Множина мікрокоманд створює мікропрограму команди. Кожна команда має свою мікропрограму, час виконання якої називається *командним циклом*.

Розрізняють апаратні, мікропрограмні й комбіновані БК. *Апаратні БК* побудовані на основі схемної логіки, *мікропрограмні* – на основі програмовної логіки (пам'ять мікропрограм), *комбіновані* використовують обидва способи реалізації.

## 7.2 Мікропроцесори

*Мікропроцесор* (далі – МП) – процесор, складові частини якого мініатюризовані й розміщені в одній або декількох мікросхемах.

Історія МП почалася в 1971 р., коли фірма Intel розробила чотирирозрядний процесор 4004, а через рік – восьмирозрядний 8008.

До характеристик МП інтегральних мікросхем (далі – ІМС) відносяться: розміри кристала і кількість транзисторів у ньому, тип корпусу і кількість виводів.

МП, призначений для використання функціонально повного набору операцій, є універсальним, інакше – спеціалізованим.

За конструкцією розрізняють такі МП:

- однокристальні – виконані у вигляді єдиної мікросхеми;
- багатокристальні (модульні) – побудовані з мікросхем, кожна з яких виконує функції МПП;
- розрядно-нarrowувальні (секційні) – розрядність їхня може збільшуватися за рахунок декількох МП секцій, об'єднаних спільними шинами.

Усі МП мають засоби сполучення із зовнішніми пристроями - інтерфейси (далі – ІФ).

Залежно від режиму роботи розрізняють такі МП:

- одно- й багатопрограмні;



- мультипроцесори (одночасно активні декілька процесорів);
- конвеєрні (команди виконуються послідовно рядом МПП);
- матричні (мають спеціальну архітектуру на обробку числових масивів);
- співпроцесори (для розширення списку команд самостійно не використовуються);
- периферійні – для вводу–виводу інформації;
- асоціативні (характер обробки даних визначається змістом самих даних).

За видом оброблюваної інформації розрізняють цифрові й аналогові МП. В аналогових – на вході використовує аналогово-цифровий пристрій (далі – АЦП), а на виході – схема цифро-аналогового перетворювача (далі – ЦАП).

### 7.3 Архітектура мікропроцесорів

*Архітектуру мікропроцесора* характеризують: список команд, формати команд, способи адресації; розрядність і ємність адресованої пам'яті; структура регістрів і їхні функції та ін.

Першими склалися архітектури МП: з акумулятором, регістром загального призначення (РЗП), зі стековою організацією і комбіновані.

*Структура всіх МП* містить однакові пристрої, вузли і блоки, об'єднані спільною внутрішньою шиною даних; АЛБ; ПК; IR – регістр команд (інструкцій); FL – регістр ознак (прапорців); EAR – регістр виконавчої адреси; БІФ – блок інтерфейсу з вихідними шинами адреси, даних, керування.

В архітектурі МП з акумуляторами (рис. 7.2) додатково використовують індексний регістр X і покажчик стека SP; регістр R виконує функції акумулятора А. При виконанні арифметико-логічних операцій перший операнд попередньо розміщується в А, а другий – з оперативної пам'яті (Оп) безпосередньо поступає на вхід АЛБ. Результат операції розміщується в А, а в FL – записуються ознаки операції.

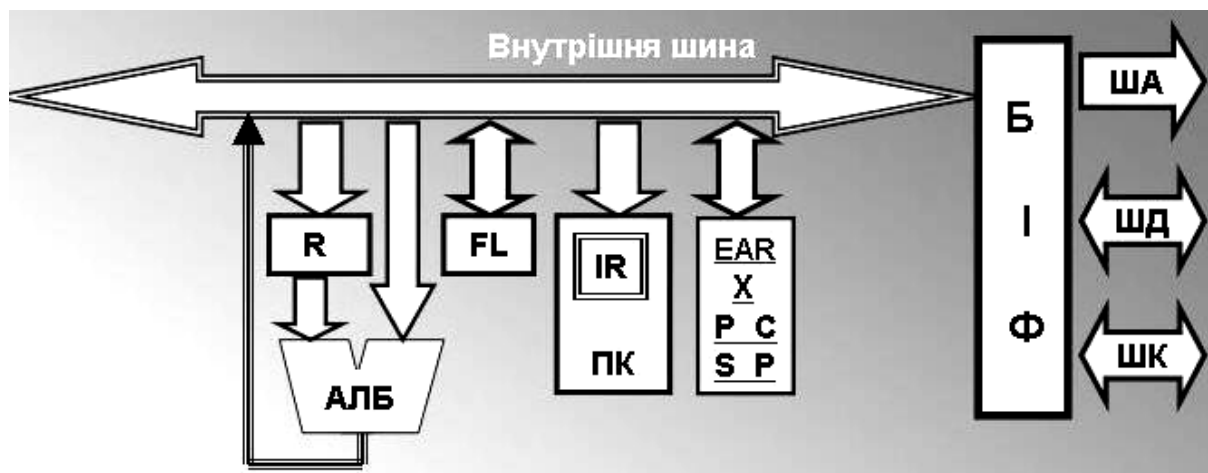


Рисунок 7.2 – Структура МП з акумулятором і стеком

В архітектурі МП з РЗП використовують від 8 до 16 регістрів, кожний з яких може виконувати функції акумулятора.

У МП зі стековою архітектурою відсутні акумулятор і РЗП, а R виконує функції регістра часового зберігання даних ОП на час виконання операції. Читання даних і їхнє записування у стекову пам'ять здійснюють за допомогою покажчика стека SP (стек – структура даних «останній прийшов – перший вийшов»).

## 7.4 Мікропроцесорні комплекти

Для побудови МПП використовують мікропроцесорні комплекти (далі – МПК) й великі інтегральні схеми (ВІС) (рис. 7.3). Основні параметри більшості МПК (серія, підгрупа МП, швидкодія в MIPS, розрядність в бітах, споживана потужність, число мікросхем в серії) ілюструє таблиця 7.1. Залежно від виду і типу базової програмної ВІС можна виділити такі МПК:

- з одним або кількома однокристальними мікропроцесорами: серії КР580, КР581, КР1810;
- з багатокристальними МП: К581, К588;
- із секціонованими МП: К583, К1800, К1802.

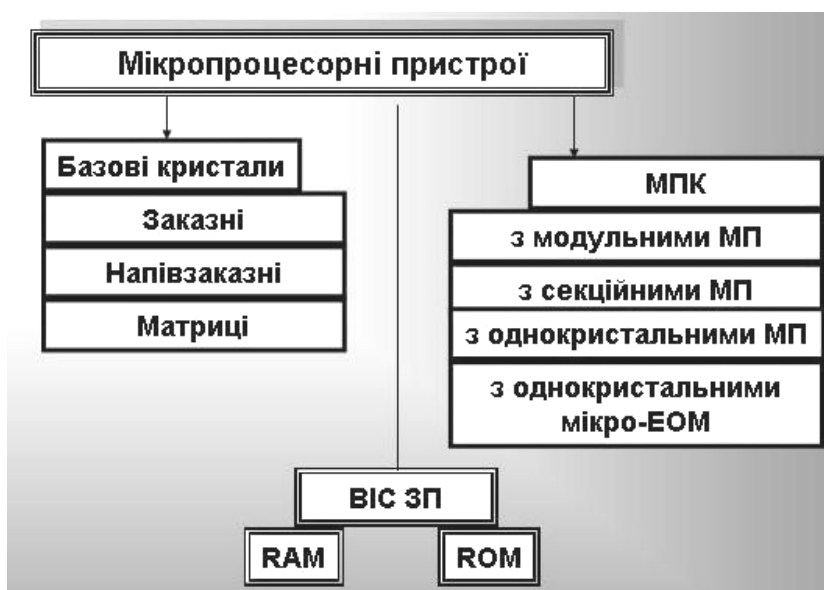


Рисунок 7.3 – Склад мікропроцесорних пристроїв

Таблиця 7.1 – Загальна характеристика процесорів і мікропроцесорів

Серія МПК	Тип ІМС	F, MIPS	n, біт	P, Вт	Число ВІС, шт.
1	2	3	4	5	6
К580	ВМ80	0,5	8	1,5	21
К581	ВЕ1	0,4		0,9	6
К583	ВС1	0,1	8	0,6	11
К584	ВМ1А	0,5	4	0,25	4
К588	ВС2А	0,5	16	0,05	21
К1800	ВС01	3,0	4	1,7	10

Продовження таблиці 7.1

1	2	3	4	5	6
K1802	BC1	0,8	8	1,4	15
K1810	BM86	0,7	16	0,9	22
K1813	BE1	1,2	28		1
K1820	BE1	0,5	8	0,6	2
K1827	BE1	0,5	16		11
Z86x02 (Zilog)		0,2	14	0,06	
Z86x30 (Zilog)		0,4	24	0,075	
AT89xx (Atmel)		< 0,6	8	0,035	
HC11 (Semicond/ Motorola)		< 0,5	8/16	0,124	
PIC (Microchip)		0,5	13/9	0,015	

Однокристалні МП мають фіксовану структуру і систему команд. Призначені для побудови мікро- і міні-ЕОМ. Однокристалні мікро-ЕОМ на транспорті виконують функції контролю, керування і діагностики.

Швидкодія в MIPS – для операцій типу реєстр – реєстр.

Модульні МП орієнтують на реалізацію конкретних алгоритмів, наприклад, оброблення цифрових сигналів.

Секціоновані МП забезпечують побудову високопродуктивних універсальних МПП методом нарощування числа секцій.

Поява МП була еволюційним процесом переходу в новий якісний стан обчислювальної техніки і систем автоматизації технологічних процесів і виробництв.

До техніко-економічних переваг МП відносять:

1. Високу швидкодію і надійність, малу потужність споживання, малі розміри й масу, низку вартість.
2. Простоту технічного обслуговування, складання і розширення МПП.
3. Можливість побудови потужних МПП з паралельною обробкою інформації.
4. Реалізацію розподільних систем контролю, діагностики і керування.

Розвиток МП у часу ілюструє таблиця 7.2 для виробів різних фірм.

У перших МП рівень інтеграції визначався тисячами транзисторів на кристалі. У сучасних МП ця величина досягла десятків мільйонів. При цьому транзисторні складки можуть працювати на частотах більше 2500 Мгц. Нові покоління МП з'являються кожні два роки.

## 7.5 Архітектура однокристального мікроконтролера

*Однокристальні мікроконтролери* – є базова основа для створення сучасних інформаційно-керуючих систем на транспорті. Узагальнена структурна схема МК містить операційний пристрій: арифметико-логічний вузол і регістр, прапори. Пристрій керування МК містить лічильник команд РС (Program Counter) і вузол синхронізації та керування.

Таблиця 7.2 – Вироби фірм

Назва МП	Рік випуску	$n$ , біт	Адресний простір	$F$ , MIPS	$N_{TP}$
4004 (Intel)	191	4	640 байт	0,1	2 800
8008 (Intel)	1972	4		0,15	3 500
8080 (Intel)	1974	8	64 Кбайт	5,0	5 000
8086 (Intel)	1978	16	1 Мбайт	5-10	29 000
Pentium	1993	32	4 Гбайт	75-200	3,1 млн.
Pentium MMX	1997	32	4 Гбайт	166-200	4,5 млн.
Itanium(Merced)	2000	64			
AVR (Atmel)	2005	16/8	128 – 4 096 байт	0 – 0,4	
PIC12xxxx (Microchip)	2005	10/9	$\leq 24$ байт	0,1	
AT91M ... (Atmel)	2005	24/16	8-256 байт	$<1,65$	
Z180/380 (Zilog)	2006	16/8	$<8192$	$<0,825$	

*Адресний простір* – це безліч елементів пам'яті, до яких може звертатися завдання;  $N_{TP}$  – число транзисторів у МП.

Система пам'яті МК–ПЗП для зберігання програм і ОЗП для даних і стека (структура даних за принципом «останній прийшов – першим вийшов»), виконаних у вигляді регістрового файлу. Пристрої вводу-виводу в МК представляються декількома програмованими портами Port's (рис. 7.4).

На кристалі МК зроблені також ряд периферійних пристроїв: таймери/лічильники, аналогові компаратори, послідовний інтерфейс та ін.

## 7.6 Функціональні структури мікропроцесорних пристроїв

За кількістю виконуваних програм розрізняють одно- і багатопрограмні МПП. В *однопрограмних МПП* в кожний момент часу виконується тільки одна програма. Перехід до наступної програми відбувається тільки по завершенні поточної програми.

У *багатопрограмних МПП* виконуються кілька програм. МПП такого типу спроможні контролювати стан одноразово декількох об'єктів керування або вирішувати задачі різних видів.

Універсальні МПП можна використовувати для вирішення різних завдань, але часто застосування таких пристроїв для виконання завдання одного конкретного виду є надто дорогим для споживача.

Спеціалізовані МПП дають змогу різко збільшити продуктивність і скоротити витрати на створення та експлуатацію системи керування. Як правило, такі МПП придатні для виконання задач лише одного виду, наприклад, мікроконтролери для керування локальним устаткуванням на рухомих одиницях, засобами захисту від коротких замикань, системами обмеження і вибору оптимальної швидкості руху, обліку і економії електричної енергії, та багатьох інших. Це найпоширеніші технічні рішення на базі ВІС.

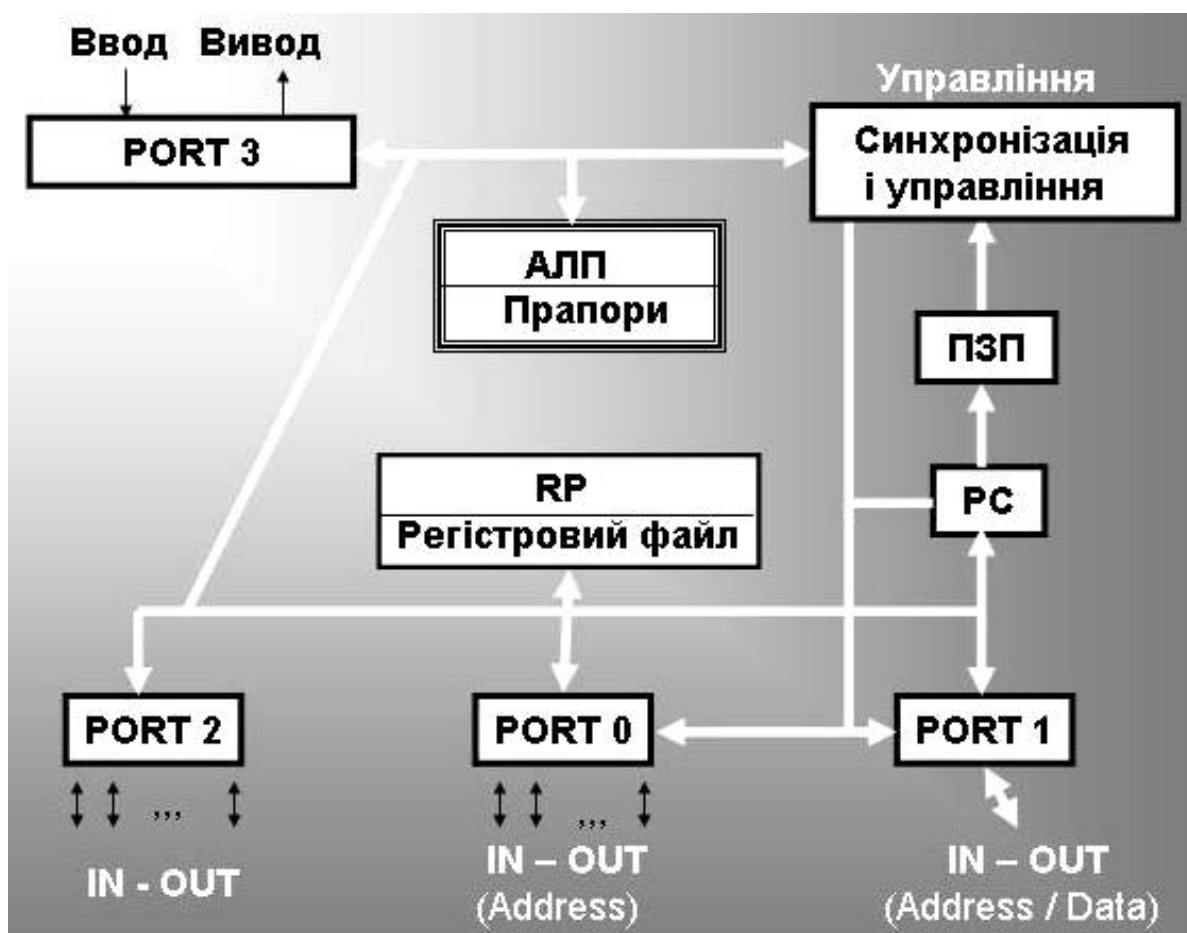


Рисунок 7.4 – Узагальнена структурна схема МК

Для вирішення подібних завдань використовують *програмовані логічні інтегральні схеми* (далі – ПЛІС – Programmable Logic Device, PLD). ПЛІС навіть в єдиному екземплярі дозволяє створювати спеціалізовані програмовані технічні засоби автоматики. Такі пристрої реалізуються на простій друкарській платі малих розмірів, їх впровадження здійснюється в найкоротші терміни. Пристрої відрізняються високою надійністю, а схеми мають надійний захист від копіювання. На транспорті такі технічні рішення знайшли широке розповсюдження. Пов'язано це з тим, що фірми-розробники таких технічних засобів гарантують їх обслуговування, ремонт і модифікацію, оскільки тільки вони мають доступ до всіх секретів такого устаткування.

Програмовані логічні контролери на ПЛІС, за допомогою яких можна отримати різноманітні схеми комутації вхідних і вихідних пристроїв та інші завдання програмним шляхом без зміни внутрішньої структури центрального пристрою обробки інформації знаходять широке розповсюдження в локальних системах автоматики і в мережах різних технічних і інформаційних служб на транспорті.

### **7.7 Адресний простір мікропроцесорів і мікроконтролерів**

*Під адресним простором МП* розуміють загальну множину адрес регістрів пам'яті, доступних для програм, які виконує МПП. Розмір адресного простору визначається кількістю ліній зв'язку адресної шини і виражається в одиницях мінімальних адресованих елементів пам'яті – байтах, або в похідних одиницях. Наприклад, якщо адресна шина має 16 ліній зв'язку, то розмір адресного простору становить  $2^{16} = 65535$  байт.

Для полегшення орієнтування в адресному просторі МПП складають карту пам'яті, в якій всі адреси регістрів пам'яті розподіляють на окремі області. У загальний адресний простір мікроЕОМ часто включають і адреси пристроїв вводу-виводу.

*Адресний простір МК* (далі – АПМК) складається з адресних просторів реєстрового файлу (далі – АПРФ) і пам'яті (далі – АПП). У АПРФ включені периферійні реєстри, реєстри керування, порти вводу-виводу і реєстри загального призначення, які утворюють ОЗП МК. АПП утворено елементами внутрішньої і зовнішньої пам'яті програм для зберігання програмного коду і елементами зовнішньої пам'яті для зберігання даних і розміщення стека.

*Приклад розподілу пам'яті мікропроцесорного пристрою* має наступну структуру: ПЗП операційної системи (програма монітор), реєстри контролера прямого доступу, реєстри контролера дисплея, реєстри інтерфейсу, реєстри клавіатури, екранна область (дисплей), робочі осередки програми монітор або верхня межа стека, ОЗП користувача.

*Стек МК* може розташовуватися у внутрішньому ОЗП або в зовнішній пам'яті даних. Розміщення стека програмується розробником. При функціонуванні МК стек може переповнюватися, але в цьому випадку результат роботи МПП буде непередбачуваний.

### **7.8 Пам'ять внутрішня і зовнішня**

*Пам'яттю МПП* називається сукупність різних пристроїв, призначених для приймання, зберігання і видачі двійкової інформації.

Пам'ять МПП класифікують за функціональним призначенням, видом носія інформації, способом організації доступу до інформації. За функціональним призначенням пам'ять поділяється на дві групи: зовнішню і внутрішню.

*Зовнішні ЗП* – для тривалого зберігання великих масивів інформації з великою ємністю і малою швидкістю. Зовнішня пам'ять містить в собі накопичувачі на дисках, оптичних дисках та ін.

*Внутрішні ЗП* призначені для зберігання програм і даних, які виконуються в поточний момент часу. До внутрішньої пам'яті відносяться:

- *регістрові ЗП*, які використовують регістри загального призначення процесора (мають невелику інформаційну ємність і швидкість роботи процесора);

- *кеш-пам'ять*, яка служить для зберігання копій інформації, що використовуються в поточних операціях обміну (висока швидкість);

- *оперативні*, що характеризуються високою швидкістю та інформаційною ємністю. У процесі роботи МПП інформація із зовнішньої пам'яті при необхідності переписується в оперативний ЗП (ОЗП);

- *постійні* – будуються на напівпровідникових ВІС. У постійній пам'яті інформацію можна тільки прочитувати. ОЗП – основна пам'ять усіх МПП;

- *спеціалізовані види пам'яті* – багатопортові, відеопам'ять та ін.

*Елементи пам'яті* – мікросхеми різного ступеня інтеграції. Основа сучасних ЗП – елементи статичного або динамічного типу.

*Статичні ЗП* – запам'ятовування біта інформації виконують тригери.

*Динамічні ЗП* – інформація зберігається у вигляді зарядів на дуже малій ємності ( $C = 0,01-0,05$  пФ), яка створена між стоком і підкладкою транзистора. Заміна інформації в динамічному ЗП забезпечується регенерацією пам'яті.

Покоління мікросхем динамічної пам'яті змінюються через 4–5 років. Фірми IBM, Siemens, Motorola та інші створюють ультра – ВІС з ємністю від 4 Кбіт до десятків Мбіт.

Напівпровідникова пам'ять має різні структури.

*Адресні* (словникові) лінії служать для вводу в кожний розряд вибраної комірки пам'яті (далі – КП).

*Адресні й розрядні лінії* (лінії записування-зчитування, далі – ЛЗЗ) називаються лініями вибірки. Якщо довжина адресного коду дорівнює  $N$ , то кількість слів  $M$ , що зберігаються в пам'яті як окремі одиниці даних, визначається із співвідношення  $M = 2^N$ .

Пам'ять з послідовним доступом – буфер FIFO і стек. Буфер FIFO – це пам'ять для зберігання черг даних (списків). Слова з черги вибирають в порядку їхнього надходження. Моменти записування слів у буфер і зчитування з нього задаються зовнішніми керуючими сигналами незалежно один від одного. Дані поступають в темпі джерела інформації, а зчитуються з частотою, необхідною для приймача.

*Кеш-пам'ять* (Cache – тайник) це засіб копіювання і зберігання блока даних основної пам'яті у процесі виконання програм. Кеш-пам'ять побудована на швидкодіючих тригерах, але має невелику ємність порівняно з основною динамічною пам'яттю. Кеш зберігає обмежене число даних і тегів. *Тег* містить інформацію про фізичну адресу і стан даних.

При кожному зверненні до основної пам'яті спеціальний контролер перевіряє за тегом наявність цієї копії в кеші. Якщо вона є, то виробляється сигнал Hit (кеш-попадання) і звернення відбувається до кеш-пам'яті. Структуру кеш-пам'яті ілюструє рисунок 7.5.

Мікроконтролери містять у своїй структурі флеш-пам'ять. *Флеш-пам'ять* (Flash Memory) використовують на транзисторах з електричним стиранням і записуванням інформації. Вона відноситься до постійної пам'яті, але ряд архітектурних і функціональних особливостей дозволяє виділити флеш-пам'ять в окремий клас. Флеш використовує поряд з традиційними адресними і керуючими сигналами спеціальні команди. Інформація у мікросхемах флеш записується і зберігається в блоках визначного розміру, іноді – призначення.

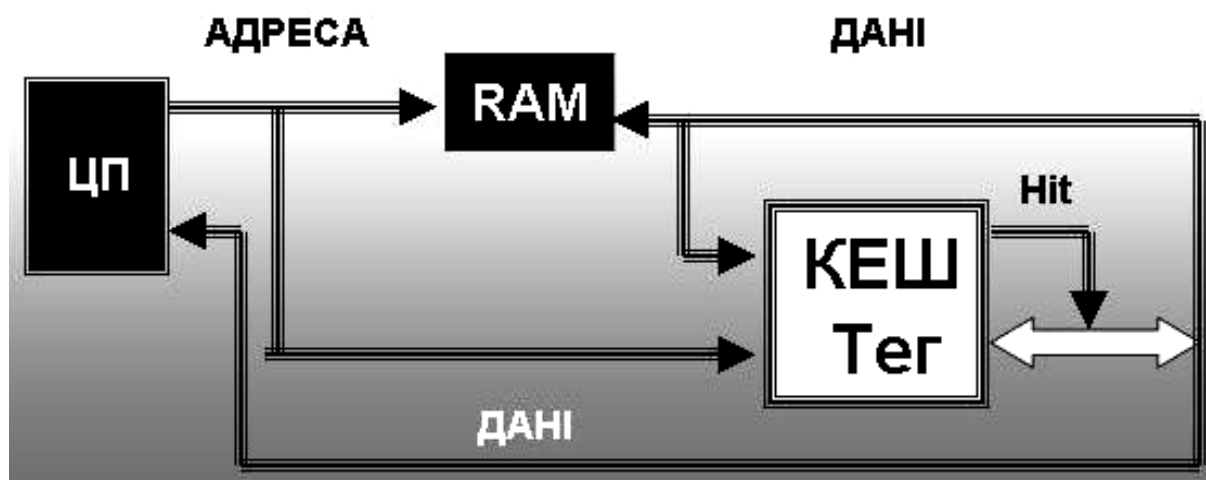


Рисунок 7.5 – Структура кеш-пам'ять

При цьому стирання інформації здійснюється або для всієї пам'яті разом, або для великих блоків. Флеш-пам'яті розрізняють:

- файлова флеш-пам'ять (Flash File) – масив розділений на блоки однакового розміру (симетрична архітектура). Ємність мікросхем 4–32 Мбіт, час доступу – 70–150 нс;
- флеш з несиметричною архітектурою (Boot Block) – масив розділений на блоки різного розміру. Ємність різного розміру, час доступу 75–150 нс, гарантується  $10^5$  циклів стирання-програмування;
- флеш з можливістю стирання тільки всього масиву (Bulk Erase). Мікросхеми мають однобайтову організацію ємністю 32–256 Кбайт, час доступу 65–200 нс;
- флеш з можливістю записування інформації за різними напругами програмування (Start Voltage);
- флеш, які зберігають по два біти (Strata Flash). Це нові елементи пам'яті, що дозволяють на ємності 32 Мбіт записувати інформацію 64 Мбіт. Залежно від заряду елемент пам'яті має чотири значення порогових напруг. При зчитуванні інформації струм елемента пам'яті має також чотири значення, їх основні вихідні схеми формують дворозрядний код.



## 7.9 Контрольні питання

1. Що таке процесор?
2. Що таке центральний процесор?
3. Що таке пам'ять МПП?
4. Назвіть периферійні пристрої МП.
5. Охарактеризуйте архітектуру процесора.
6. Поясніть назначення АЛБ, ПК, РЗП, БІФ процесора.
7. Що таке мікропроцесор?
8. Охарактеризуйте архітектуру мікропроцесора.
9. Охарактеризуйте структуру МП з акумулятором і стеком.
10. Назвіть типи мікропроцесорів.
11. Перелічіть режими роботи МП.
12. Поясніть назначення мікропроцесорних комплектів.
13. Охарактеризуйте склад мікропроцесорних пристроїв.
14. Перелічіть техніко-економічні переваги МП.
15. Поясніть структурну схему МК.
16. Які є функціональні структури МПП?
17. Що таке ПЛІС?
18. Поясніть адресний простір МП і МК.
19. Для чого призначена внутрішня пам'ять?
20. Для чого призначена зовнішня пам'ять?
21. Які принципи побудови постійної пам'яті?
22. Які переваги має динамічна пам'ять порівняно зі статичною?
23. Що таке кеш-пам'ять?
24. Поясніть назначення тег в кеш-пам'яті.
25. Що таке флеш-пам'ять?
26. Назвіть параметри пам'яті.

## 8 ПРОГРАМОВАНА АВТОМАТИКА НА БАЗІ МІКРОПРОЦЕСОРІВ

### 8.1 Функціонування мікропроцесорів у часі

Функціонування МП у часі визначається внутрішнім генератором тактових (синхронізуючих) імпульсів (далі – ГТІ). Збудження його може задаватися зовнішнім кварцовим резонатором, RC- чи LC-ланцюгами, або зовнішнім генератором, які підключаються до входів МП.

Внутрішній ГТІ працює і видає вихідний сигнал SYNC частотою вдвоє меншою, ніж Умп на входах мікропроцесора (рис. 8.1).

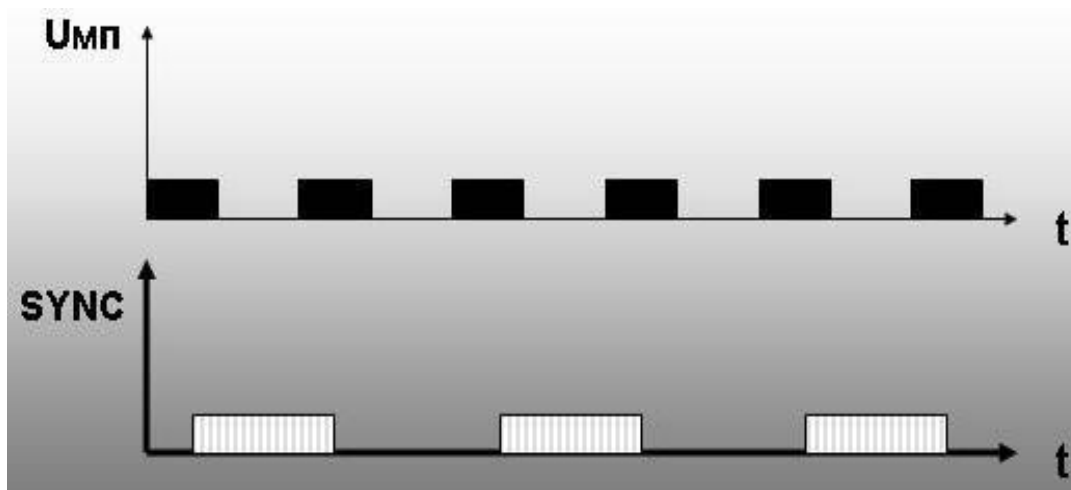


Рисунок 8.1 – Діаграма сигналів  $U_{mp}$  і SYNC

Період проходження сигналів SYNC визначає машинний такт  $T_c$ , з яким пов'язаний стійкий стан мікропроцесора. Команди виконуються за машинними циклами. Залежно від типу команди для її виконання потрібно звичайно не більше п'яти машинних циклів. У МК ці процеси виконуються набагато швидше.

При тактовій частоті  $F_c = 5$  МГц тривалість такту  $T_c = 0,2$  мкс.

Блок керування і синхронізації МП забезпечує виконання в тактах типових машинних, які наведені в таблиці 8.1.

Таблиця 8.1 – Типові машинні дії

Номер такту	Типові дії в такті
T1	Видача адреси на ША
T2	Аналіз сигналів: «Готовність», «Запит прямого доступу до пам'яті»
Tи	Перехід в режими очікування при «Готовність = 0»
T3	Обмін даними. Виконання операцій
T4, T5	Виконання внутрішніх дій для ряду операцій

## 8.2 Формати команд і даних

Формат команд МП залежить від типу операції і може бути одно-, дво- або трибайтовим (рис. 8.2). Перший байт містить код операції, інші – дані чи адресу операнда. Дво- і три байтові команди зберігаються в послідовних комірках оперативної пам'яті. Для читання команди з пам'яті вказують адресу її першого байта.

У МП дані представляються у форматі восьмибітового двійкового слова (числа). Є команди, що оперують з 16-бітовими даними, які запам'ятовуються в двох послідовних комірках пам'яті. В арифметичних операціях МП інтерпретує байт даних як число із знаком в доповнювальному коді. Старший біт D7 такого

числа є знаковим (0 – для додатного числа, 1 – для від’ємного). Інші сім (D6–D0) бітів створюють значення числа.

Числа у форматі байта змінюються в діапазоні: для додатних чисел від 0 до плюс 127; для від’ємних чисел – від мінус 1 до мінус 128. Якщо знакова інтерпретація чисел не обмовлена: 0–255.

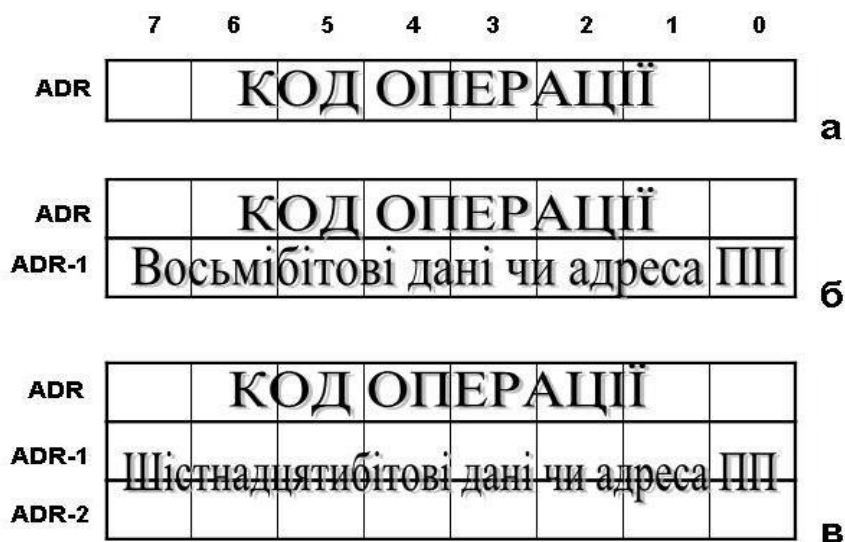


Рисунок 8.2 – Формат команд:  
а – одно-; б – дво-; в – трибайтовий

### 8.3 Мова асемблеру

Систему команд мікропроцесора описують на мові асемблеру.

*Мова асемблеру* (Assembly або Assembler) – це машинно-орієнтований засіб програмування. Програма на мові асемблеру називається *початковою*. Асемблер трансліює початкову програму в символах у двійковій машинній коді. Така програма називається *об’єктною*, її можна завантажувати в пам’ять для виконання.

Програма готується у вигляді послідовності речень, які називаються *операторами* мови асемблеру. Кожний оператор записується в одному окремому рядку і після трансляції породжує одну машинну команду.

*Рядок оператора має п’ять полів*: адреса, мнемокод (код операції), мітка, операнд (и), коментарі.

Усі оператори записуються за допомогою символів Американського стандартного коду для обміну інформацією (ASCII – American Standart Code for Information Interchange).

Мітки звичайно відділяються від поля мнемоніки двокрапкою. Наприклад, МІТКА: *MOV A, B* – команда пересилки, *JMP МІТКА* – команда переходу.

У полі операндів розміщують один або два операнди. Двійкові дані закінчуються символом В, десяткові – D (або може опускатися), шістнадцяткові – H. Якщо в шістнадцятковому числі перший знак є літера, то перед нею

пишуть цифру 0. Наприклад, МІТКА: *MVI A,0EH* – шістнадцяткових, *MVI B,15* – десяткових, *MVI L, 1111B*; – двійкових.

У полі коментарів, яке відділяють попереднього поля точкою з комою, вміщують опис призначення оператора.

У мові асемблеру використовують директиви, які призначені для передачі інформації, що використовується при трансляції. Основні директиви: керування трансляцією *ORG* (початок програми, адреса), *END* (кінець, закінчення програми); визначення даних та резервування в пам'яті області даних *DB* (дозволяє організувати символічну адресацію даних), *DW* (задає послідовність 16-розрядних даних: молодший байт – адреса мітки, а старший байт – адреса на одиницю більше), *DS* (визначення пам'яті – кількість комірок або байтів пам'яті, які резервуються для запам'ятовування даних). Наприклад, МІТКА1: *DB 15, 1, 24, 50* – запам'ятати чотири байти даних; МІТКА2: *DB «DATA»* – дані як п'ять символів; *DATA: DW 1100H, 0222H, TABLE: DS 64; 64* байт даних.

## 8.4 Призначення програм і етапи їх створення

Мікропроцесорні пристрої автоматики на транспорті часто вирішують типові задачі на різних технологічних об'єктах. Наприклад, завдання захисту силового обладнання від струмів короткого замикання мають багато близьких технічних рішень при їх реалізації як на рухомих одиницях, так і на стаціонарних тягових підстанціях. Очевидно, що при розробці таких пристроїв завжди передбачається їх універсальність для застосування на різних об'єктах.

Програмовані технічні рішення завжди повинні володіти універсальністю, яка досягається адаптацією апаратної частини устаткування МПП його програмними засобами.

Більшою гнучкістю володіють комп'ютерні системи, які мають схожу структуру. Апаратні модулі комп'ютера, що розташовуються поблизу мікропроцесора, легко тиражуються, упрощуються і модернізуються в дуже короткі терміни.

МПП на базі RISC- і CISC – мікропроцесорів стали невід'ємною частиною сучасного локального силового устаткування на транспорті. На їх основі створюються вбудовані системи жорсткого контролю різних технологічних величин і справності устаткування на основі систематичної діагностики вузлів і механізмів.

Побудова програм для МПП автоматики на транспорті забезпечує швидкодію систем контролю, керування і взаємозв'язку їх із службами транспортних підприємств.

До складу програмного забезпечення МПП входять як програми розробника керуючої системи, так і створені користувачем.

Програмні засоби забезпечують:

– організацію спільного функціонування усіх елементів системи (операційна система);

– створення користувачем робочих програм (система програмування, редагування і налагодження);

- використання програм, що накопичуються у процесі їх створення користувачем (бібліотека прикладних програм);
- контроль правильності функціонування апаратних засобів (система контролю і діагностування);
- проектування програмних і апаратних засобів за допомогою зовнішньої системи автоматизованого проектування.

Оснoву програмного забезпечення мікро-ЕОМ складає операційна система (далі – ОС), за допомогою якої відбувається керування всіма апаратними й програмними засобами.

Програми ОС зберігаються у пристроях масової пам'яті і передаються до ОЗП МПП.

*Програма* – це послідовність команд, що забезпечує виконання МПП певних дій. Команди програми і дані зберігаються в регістрах пам'яті і закодовані у двійковій формі. Команди послідовно зчитуються із ОЗП і виконуються апаратними засобами. Програма, подана у двійковій формі – об'єктна програма.

Основні етапи створення програми:

1. Складання алгоритму програми.
2. Написання програми на символічній мові.
3. Трансляція початкового тексту програми.
4. Перевірка програми.
5. Завантаження програми в ОЗП і відпрацьовування.
6. Запис в накопичувач або реалізація.

Для реалізації всіх етапів програмісти зобов'язані мати повний комплект програмних і технічних засобів для перевірки кінцевого продукту і перенесення його в пам'ять МПП за допомогою пристроїв, що називаються програматорами.

Найбільш поширеними є *програма тори* – пристрої для виготовлення інтегральних мікросхем пам'яті для МПП.

Програматори часто виконують у вигляді зовнішніх пристроїв, що підключаються до МПП через порт сполучення і в діалоговому режимі завантажують в них необхідні програмні продукти.

## 8.5 Структура програмного забезпечення

Створення програм не є окремим напрямом в технічних науках, суть тут полягає в тому, щоб використовувати якийсь метод, що дозволить одержати добре написану, читабельну програму.

Програмування прикладних завдань вважає за краще:

1. Створення структурованої програми, яка зручна при налагодженні програми.
2. Включення в структуру об'єктних модулів, раніше створених незалежно один від одного, що вимагає обов'язкового розбиття будь-якого проекту на декілька окремих частин.

3. Програмні продукти експлуатуються не його творцями, а іншими користувачами, які повинні добре розуміти структуру використовуваного програмованого технічного рішення.

Існує декілька фаз створення реальних проєктів:

1. Аналіз з метою ясного розуміння призначеного МПП.
2. Проєктування визначає спосіб досягнення необхідних характеристик.
3. Реалізація досягається кодуванням проєкту на мові програмування.
4. Тестування підсистем проєкту.
5. Налаштування окремих функцій.
6. Тестування системи всього проєкту в цілому.
7. Документація для користувачів програмним продуктом.

Структура конструювання програм ведеться «зверху–вниз». Ідея полягає в послідовному розбитті конкретного пристрою на все менші й менші частини. Такий підхід забезпечує систематичний аналіз програмного завдання, що є деревом, де компоненти нижнього ряду є складові частини компонента верхнього ряду (рис. 8.3).

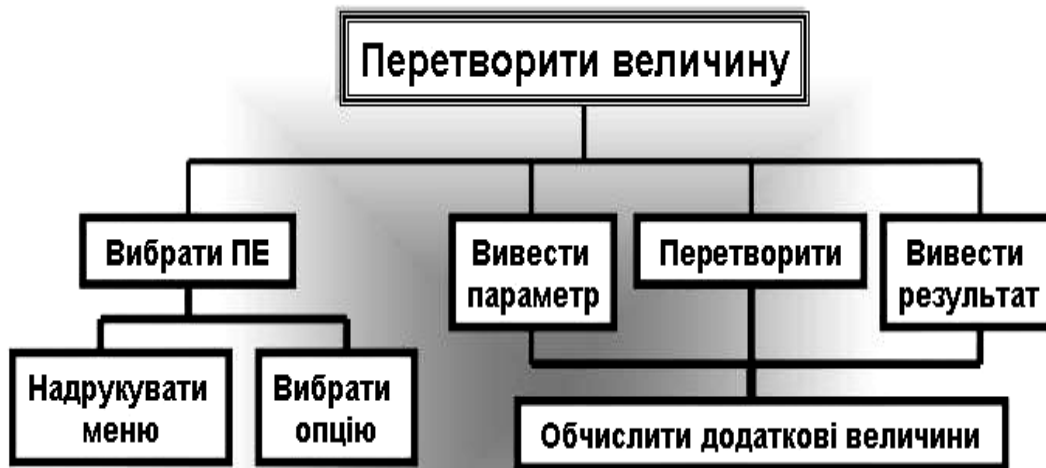


Рисунок 8.3 – Структура програмування перетворення фізичної величини

Правила структурного програмування:

- модулі програми повинні бути невеликими й легкими для розуміння;
- модулі програми повинні мати одну точку входу на початку і одну точку виходу в кінці;
- програми повинні використовувати комбінації типових конструкцій:
  - а) послідовності двох або декількох операцій наступні по черзі одна за одною;
  - б) вибір однієї складової частини з двох або більш можливих;
  - в) повтор однієї або декількох операцій за певними умовами.

Поетапна розробка елементів програми дозволяє з особливою ретельністю вдаватися в подробиці керованих процесів. Такий підхід примушує активно вивчати технологічні процеси, виявляти їх недоліки і намічати шляхи подальшого вдосконалення як реальних об'єктів, так і технічних засобів автоматизації, що створюються на базі мікропроцесорів.

## 8.6 Алгоритми і правила їх складання

*Алгоритм* – точний припис, що визначає послідовність операцій процесу перетворення початкових даних у потрібний результат. Алгоритм створюється в три етапи:

1. Складається концептуальна схема алгоритму, яку отримують при простому переліку в логічному порядку всіх головних подій технологічного процесу. Концептуальна схема містить загальні вирази, які показують, що повинно бути зроблено для отримання потрібного результату.

2. Складається функціональна схема алгоритму. Вона визначає, як це може бути досягнуто. Це найбільш докладний з документів, в якому визначені всі логічні зв'язки.

3. Складається структурна схема команд, що являє собою покажчик для кодування програми для конкретної мікро-ЕОМ, мікроконтролера або іншого МПП.

Алгоритм записується у вигляді послідовності дій над вхідною інформацією. Ці дії називаються операторами. Алгоритм може бути записаний у словесній формі за кроками або зображений графічно. Прийнята послідовність графічного зображення алгоритму – зверху вниз і зліва направо. Концептуальна схема алгоритму розробляється у вигляді набору великих модулів таким чином, щоб вона, бажано, вмістилася на одній сторінці аркуша. При складанні функціональної схеми кожен модуль деталізується і перетворюється у множину функціонально-самостійних схем.

При написанні програм на символічній мові (асемблері) всі елементи програми відображаються у символічній формі, яка передає їх зміст. Мікропрограми представляють направленим графом, що складається з вершин: «Початок», «Кінець», операторних (відповідають одній мікрокоманді кожна) та умовних (еквівалентних умовним переходам до двох міток). Якщо операторні й умовні вершини графа містять повний опис мікрооперацій і булевих виразів (логічних умов), то його називають *змістовним*.

Опрацьовування завдань зображують у вигляді блок-схем (або граф мікропрограм). У кожному графі або блоці міститься опис однієї або декількох операцій. Надаючи блокам різну форму, створюється зорове представлення певних операцій. Блок-схема є корисним допоміжним засобом для опису того, як працює конкретна програма. Графічний вигляд підкреслює логіку програми, процес рахунку і керування програмою. Блок-схема не залежить від мов програмування, а тому корисна при створенні продуктів на будь-яких мовах програмування.

Стрілки на блок-схемах мікропрограм показують напрям розвитку подій у програмі.

Існують три основні типи блок-схем:

1. *Системні блок-схеми*. У системній блок-схемі показується, які пристрої використовуються для вводу, висновку і зберігання даних. Алгоритм зображується у вигляді єдиного блоку (рис. 8.4, а).

2. *Укрупнені блок-схеми (основні)*. У таких блок-схемах алгоритм опрацьовується на принциповому рівні (рис. 8.4, б).

3. *Докладні блок-схеми (детальні)*. У блок-схемах цього типу відбувається подальша деталізація алгоритму, описаного укрупненою блок-схемою. Кожен блок представляє одну або дві–три команди (рис. 8.4, в).

Всі вершини, крім початкової, мають довільне (не менше одного) число входів.

Якщо операторні й умовні вершини графа містять повний опис мікрооперацій і булевих виразів (логічних умов), то його називають змістовним.

Мова, призначена для описування мікропрограм в термінах операторів, називається *мовою мікропрограмування*. Оператори в мікропрограмі виконуються послідовно від початку до кінця. Для зміну такого порядку використовують оператори переходу і мітки.

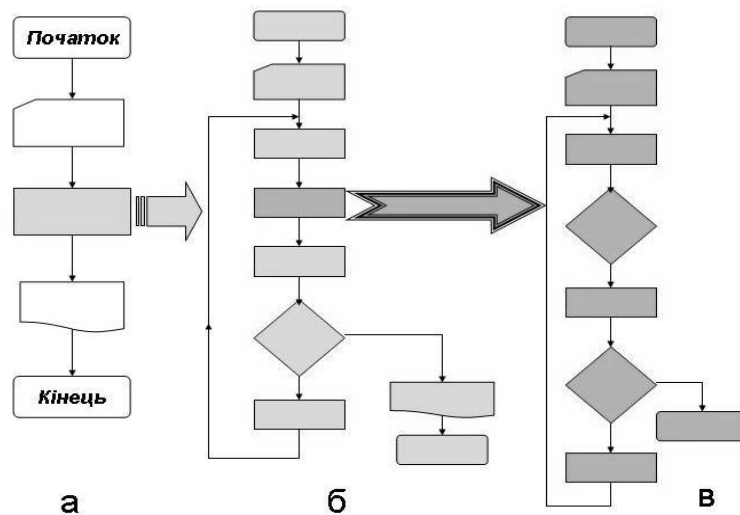


Рисунок 8.4 – Вирішення задач у вигляді блок-схем:

*а – послідовні або лінійні; б, в – з використанням циклів і галуження*

Графи мікропрограм, написаних на мовах високого рівня, завжди мають менш докладний вигляд, ніж для мови асемблера. Пов'язано це з тим, що оператори мов високого рівня завжди відповідають декільком машинним кодам.

## 8.7 Система команд мікропроцесорів

Системи команд різних мікропроцесорів відрізняються між собою. Звичайно вони налічують декілька десятків команд. Всю систему команд можна розділити на п'ять функціональних груп:

- пересилки;
- арифметичні операції;
- логічні операції;
- передачі керування;
- керування стеком, вводом-виводом і станом МП.



Команди описуються символічними назвами. Більшість мнемонік – це аббревіатури англійських слів, які характеризують основні функції команд, наприклад для мікропроцесорів KP580BM80A (аналог Intel 8080), K1821BM85A (аналог Intel 8085A), Z80 фірми Zilog та ін.:

- MOV (MOVE) – передати, переслати;
- ADD – додавати;
- SUB (SUBtract) – відняти;
- LDA (Load Accumulator) – завантажити акумулятор;
- STA (Store Accumulator) – запам'ятати акумулятор;
- ACI (Add with Carry Immediate) – додавання з перенесенням, безпосереднє;
- JNZ (Jump if Non Zero) – перейти, якщо не нуль.

Систему всіх команд МП (KP580) ілюструє рисунок 8.5.

Команди пересилання наказують розміщувати або переміщувати дані між блоками МП (див. дод. табл. А.1).

Команди обробки даних підрозділяють на арифметичні й логічні.

До групи команд арифметичних дій відносять команди арифметичного складання, віднімання, *інкрементування* (додавання одиниці) і *декрементування* (віднімання одиниці).

Команди арифметичного складання (ADD, ADI) і віднімання (SUB, SUI) виконують з двома операндами, один з яких розміщується в акумуляторі, а другий – у будь-якому іншому програмно доступному регістрі, або задається у другому байті команди. Результат виконання команди завжди розміщується в акумуляторі.

До підгрупи команд логічних дій відносять команди логічного множення «І» (ANA), логічного складання «АБО» (ORA), заперечення «НІ» (CMA), виключного складання - виключного «АБО» (XRA).

Логічні дії виконують побітно, а результат завжди розміщують в акумуляторі. До команд логічних дій відносять також команди циклічного зсуву вліво (RAL), вправо (RAR), зсуву вліво (RLC) і вправо (RRC) з перенесенням.

Команди керування за напрямком обчислювального процесу підрозділяють на команди безумовного і умовного переходу. Система команд МП завжди має принаймні одну команду безумовного переходу (JMP B2 B3 – передача керування за адресою, що вказана у другому B2 і третьому B3 байтах команди).



Рисунок 8.5 – Розподіл команд МП за функціональним призначенням

Адреса переходу завантажується у програмний лічильник.

Команди умовних переходів передають керування за адресою, яка вказана в команді, якщо виконується деяка умова. В іншому випадку виконується наступна команда, записана у програмі за командою умовного переходу.

До групи команд керування відносять команди безумовного переходу до підпрограм (CALL B2 B3) і безумовного повернення з підпрограми (RET).

Команди умовного переходу до підпрограм та виходу з них виконуються за тими ж ознаками, що й команди умовних переходів. Це трибайтні команди, де в другому й третьому байтах записана адреса регістру пам'яті, до якого треба здійснити перехід.

До спеціальних команд відносять команди вводу (IN B2) і виводу (OUT B2), зупинки МП (HLT), відсутності операції (NOP), роботи із стеком та ін. У командах вводу-виводу в байті B2 вказують адресу пристрою вводу-виводу. Всі дії по вводу-виводу даних при виконанні цих команд відбуваються через регістр-акумулятор.

Всі команди виконуються за певне число тактів. Враховуючи кількість тактів, визначають тривалість виконання будь-яких фрагментів програм або всієї програми в цілому. Слід пам'ятати, що в циклічних та ін. алгоритмах тривалість програм залежить від виконання умов (команди: CC, CNC, CZ, CNZ, CP, CM, CPE, CPO, RC, RNC, RZ, RNZ, RP, RM, RPE, RPO). Якщо умова не виконується, то число тактів звичайно менше (указується в чисельнику), ніж при виконанні умов (у знаменнику).

## 8.8 Способи адресації

Приклади способів адресації:

– *неявний*, який не вимагає операндів (HLT – зупинка, EI – дозвіл переривання, NOP – немає операцій).

– *Прямий* – адреса операнда визначається значеннями другого і третього байтів команди: LDA 1213 (виконавча адреса 1213 знаходиться в команді); ADD AL, BL (DATA8) – обидва операнди мають пряму регістрову адресацію; JMP CX (DATA16) – безумовний перехід на адресу CX; JMP 2530 – перехід на адресу 2530.

– *Безпосередній* – операнд задається безпосередньо в команді: ADD AX, DATA16; SUB BL, DATA8.

– *Регістровий* – як операнд використовують зміст регістра, вказаного в першому байті команди.

– *Непрямий* – адреса операнда визначається змістом регістрової пари (таких, як BC, DE, HL).

– *Індексна адресація* (з регістром D1): MOV D1,5500H.

Адресація з покажчиком стека називається *непрямою з автоіндексуванням*. При виконанні деяких команд застосовуються комбіновані адресації, наприклад, регістрова для акумулятора і непряма для комірки за спеціальною адресою M (комірка пам'яті, адреса якої задає зміст регістрової HL-пари, тобто  $M = (H,L)$ ).

Вивчення правил застосування команд мови асемблер вимагає часто глибокого вивчення чужих налагоджених програмних продуктів, функціонування яких зрозуміле при використанні конкретного МПП з відомою структурою.

## 8.9 Контрольні питання

1. Поясніть призначення генератора тактових імпульсів в МПП.
2. Поясніть формати команд.
3. Охарактеризуйте мову асемблер.
4. Поясніть оператори мови асемблер.
5. Поясніть призначення програмних засобів.
6. Яке призначення операційної системи або програми монітор в МПП?
7. Які основні етапи створення програм?
8. Поясніть фазу створення програмних проєктів.
9. Які існують алгоритми і етапи їх створення?
10. Поясніть призначення блок-схем програм.
11. Охарактеризуйте блок-схеми мікропрограм.
12. Назвіть і поясніть групи системи команд мови асемблер.
13. Охарактеризуйте список команд восьмирозрядного мікропроцесора.
14. Поясніть призначення програматорів.

## 9 ПРОГРАМУВАННЯ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ НА МОВАХ ВИСОКОГО РІВНЯ

### 9.1 Особливості реалізації мов високого рівня

Розвиток мікропроцесорної техніки призвів до реалізації мов високого рівня. *Мови високого рівня* (далі – МВР) в порівнянні з машинними мовами і мовами асемблер легше сприймаються і освоюються в коротші терміни. Доцільність застосування МВР обумовлена підвищенням продуктивності праці програмістів, спрощенням супроводу програмних продуктів, забезпеченням можливості постійного вдосконалення програм. Завдяки невеликим об'ємам оперативної пам'яті і доцільності підвищення швидкості функціонування МПП забезпечується *трансляція* МВР (перетворення програми в еквівалентну програму в машинних кодах). *Компіляція* (Compilation) – це трансляція програми на МВР в об'єктний модуль (самостійна частина програми), здійснювана спеціальною програмою або технічними засобами, що називаються *компілятором*. Звичайно сучасний компілятор виконує компіляцію, трансляцію, перевірку і налагодження програм.

Компілятор виконує функції лексичного і синтаксичного аналізу, а також перетворення у форму зворотного польського запису (арифметичний або логічний вираз записується послідовністю спочатку операнди, а потім знак операції, запропонованих польським математиком Я. Лукашевичем).

МВР Basic, що входить в стандартний набір програмного забезпечення, широко використовується через простоту реалізації і близькість до природного спілкування. Так, математичний запис  $(ax^2 + by) : \ln y^3$  на мові Бейсик матиме вигляд:

$$(A \cdot X^2 + B \cdot Y) / \text{LOG}(Y^3) \quad (9.1)$$

Для роботи програми всі оператори завжди мають мітки, роль яких виконують цілі числа.

МВР Basic містить операторів, що дозволяють проводити обчислення, управляти ходом виконання програм, здійснювати безумовні й умовні передачі керування, виконувати операції в циклі, працювати з підпрограмами (табл. 9.1).

МВР відрізняються між собою. Програми на популярній мові Pascal значно швидші ніж Basic, але мають певну орієнтацію при використанні. У різних МВР можливе представлення чисел в різних форматах і зміна їх, наприклад, при виводу на друк. Існують засоби для роботи з графічною інформацією, обміну з периферійними пристроями, роботи в мережах та багато іншого.

Таблиця 9.1 – Інтерпрітація операторів Basic

Оператор	Короткий переклад	Оператор	Короткий переклад
GOTO	Перейти в мітку	RESTORE	Повторне використання DATA
IF	Умовний перехід	INPUT	Ввести дані
FOR	Для параметра	PRINT	Вивести дані
NEXT	Кінець тіла циклу	REM	Коментарі
GOSUB	Перейти в мітку і запам'ятати мітку для повернення	RANDOMIZE RND	Випадкові числа
DATA	Ввести блок числових даних	STOP	Зупинка (може бути у середині програми)
READ	Витягти числову інформацію з блоку даних для привласнення їх змінним	END	Кінець програми

## 9.2 Програмування на мові СІ

МВР СІ (або С) був вперше створений на початку 70-х років і успішно розвивається. СІ компілюють у виконуваний файл в двійкових кодах спеціально для конкретного мікропроцесора. Для створення програми необхідно мати певне програмне забезпечення. У комплект програм звичайно входять текстовий редактор, бібліотечні файли, компілятор, компоновальник, файли допомоги і приклади програм.

Програми на мові СІ портативні. Якщо необхідно змінити програму, то для цього змінюється декілька рядків коду. Компілятори СІ добре функціонують як на 8-розрядних мікропроцесорах, так і на суперкомп'ютерах.

Особливий інтерес до МВР СІ програмісти почали проявляти, коли фірми РІС, Atmel, Motorola та ін. створили мікроконтролери з можливістю розробки для них програм на цій мові. Враховуючи, що асемблерні мови орієнтовані на конкретні типи мікропроцесорів і логіку їх роботи, мова СІ позбавлена подібних недоліків. СІ використовується для програмування будь-яких мікропроцесорів і особливо мікроконтролерів, для яких є компілятор з мови СІ. Вивчивши СІ, можна легко переходити від одного сімейства МПП до інших, витрачаючи на це менше часу на розробку.

Розробка програм на СІ включає декілька етапів:

1. Визначення мети програми (вимоги користувача):
  - а) описати все, що повинна робити програма;
  - б) визначити початкові умови;
  - в) точно визначити всі можливі операції з величинами і текстом;

г) встановити кінцеву інформацію, яку слід одержати в результаті роботи програми.

2. Побудова програми з використання певної структури.

3. Запис коду – переклад ідеї програми на мову СІ.

Для створення початкового коду використовується будь-який текстовий редактор.

Коментарі в початкових текстах програм наводяться всередині похилих дужок /\* ... \*/. Вкладені коментарі в СІ, які допустимі, наприклад в Basic, не вирішуються.

Всі функції наводяться на початку програми і є її частиною. Всі змінні декларуються перед рядком *main()*. Змінні величини, доступні для функцій, позначаються *global* або *local*. Типи даних визначаються завжди наперед. Якщо змінні є двобайтові цілі числа, то позначаються *int*, символні – *char*, чотирьохбайтові речовинні – *float* і т.д. На цьому етапі програма одержує ім'я з розширенням «\*.c» (наприклад, «test.c»).

4. *Компіляція* – переклад початкового тексту програми в послідовність машинних двійкових кодів. Слід враховувати спрямованість роботи компіляторів для конкретних мікропроцесорів.

Результат роботи компілятора полягає у створенні нового тексту програми з назвою, наприклад, «test.obj».

5. *Компоновка* або *складання* – виконується тільки в тому випадку, якщо початковий текст програми скомпільований без помилок. Компонувальник (спеціальна програма) збирає проєкт, що розробляється, і представляє його з назвою, наприклад, «test.exe».

6. *Робота програми* – перевіряється за допомогою додаткових програм для тестування працюючої програми з певними вхідними величинами (струм в мережі, швидкість обертання валу двигуна, температура устаткування і т.д.).

7. *Тестування і налагодження програми* - перевірка всіх завдань, які повинна вирішувати програма. При цьому:

а) обробляється весь діапазон вхідних, вихідних та інших даних;

б) перевіряються кінцеві результати;

в) оцінюються режими функціонування всього комплексу МПП;

г) визначається стійкість МПП при багатократному старті програми;

д) перевіряється вплив на програму інших програмних продуктів, якщо вони використовуються МПП, або операційного середовища.

8. *Збереження і зміна програми* – усунення виникаючих проблем, пов'язаних:

а) помилками, що з'являються при вводі неперевірених даних від ПЕ;

б) використанням програми в іншому операційному середовищі;

в) усуненням впливу на програму інших програмних продуктів;

г) впливом зовнішніх чинників (ВЕ, перешкод, електричних розрядів блукаючих струмів і т.д.).

Для спрощення процедури усунення різних помилок в програмному проєкті важливо в нього включати докладні коментарі і ретельно вести документацію, що розкриває всі етапи створення програми.

### 9.3 Системи проєктування програм

Для програмування мікроконтролерів створені програмні пакети-компілятори, які є набором інструментальних засобів, призначених для програмування RISC – мікроконтролерів певних сімейств, наприклад, AVR, PIC. Комплекти компіляторів розраховані для створення програм на ЯМР C і C++. Окрім вирішення всіх завдань, відмічених в розділі 9.2, компілятори дозволяють завантажувати готові програми в той чи інший програматор, а також емалювати виконання програм на тому чи іншому мікроконтролері.

*Емулятори* (Emulator) – імітатори функціонування всієї або частини системи засобами іншої системи без втрати функціональних можливостей або спотворення одержуваних результатів. Емулятори дозволяють детально налагоджувати програмні продукти без фізичної реалізації МПП, що має величезне значення, оскільки досягається економія матеріальних і грошових коштів на стадії проєктування засобів автоматики на транспорті.

Пакети-компілятори містять бібліотеки «стандартних фраз», використовуваних програмістами при вирішенні типових завдань. Регулярне звертання до бібліотеки скорочує час набору тексту програм і виключає помилки, яких не можна уникнути при роботі з клавіатурою.

Автоматичне збереження в компіляторах виключає втрату будь-яких вдалих розробок.

Використання віконних редакторів компіляторів дозволяє користувачу контролювати або змінювати роботу мікроконтролерів. За допомогою пакетів-компіляторів легко змінити зміст осередків пам'яті, втручатися в роботу вузлів, що ініціюють сигнали керування, які поступають на виконавчі пристрої. Налагодження програм може виконуватися в покроковому режимі з контролем вмісту регістрів (перевірка на низькому рівні) і змінних (перевірка на програмному рівні).

Оскільки багато пакетів, компілятори розроблені для запуску в середовищі Windows 95/98/Me/NT4/2000/XP або Linux, то їх використання можливе на будь-якому побутовому комп'ютері.

Особливий інтерес викликають графічні технології створення програм. Вони дозволяють вводити початкові коди програм на площини робочого вікна у вигляді алгоритму з деревовидною структурою, що нагадує блок-схему алгоритму. В результаті вся логічна структура програми стає повністю наочною. Сприйняття створюваного таким чином малюнка програми розкриває нові можливості для розробки програм. Особливо такий підхід ефективний при програмуванні маловивчених об'єктів або знов проєктованих автоматизованих об'єктів на транспорті, наприклад рухомих одиниць з новими тяговими двигунами або автономними джерелами енергії та ін.

### 9.4 Контрольні питання

1. Поясніть особливості мов високого рівня.
2. Що таке трансляція?

3. Що таке компіляція?
4. Поясніть можливості MBR Basic.
5. Поясніть популярність мови СІ.
6. Назвіть і поясніть етапи розробки програм на СІ.
7. Що таке компоновка програм?
8. Охарактеризуйте сучасні компілятори.

## 10 ПРИСТРОЇ ЗВ'ЯЗКУ З ОБ'ЄКТОМ КЕРУВАННЯ

### 10.1 Характеристики мікропроцесорних систем

*Мікропроцесорною системою* (далі – МПС) називають сукупність інформаційно-обчислювальних засобів. У МПС входить один або декілька мікропроцесорів. Для побудови МПС потрібен обмежений набір мікроелектронних ресурсів, оскільки реалізація більшості функцій покладається на програмне забезпечення. Таким чином, МПС – це цифрові системи оброблення інформації і керування, функціональні можливості яких визначаються програмним забезпеченням, а взаємозв'язок із зовнішнім середовищем здійснюється *периферійними (зовнішніми) пристроями* (далі – ПП). На основі МПС на транспорті реалізують локальні системи контролю, спостереження, діагностики, керування, інформаційні мережі, системи штучного інтелекту та багато інших.

МПП у системах об'єднані між собою внутрішньосистемним інтерфейсом і взаємодіють за адресним принципом.

*Інтерфейс (Interface)* – сукупність уніфікованих інформаційно-логічних, технічних і програмних засобів, які забезпечують оптимальний алгоритм взаємодії всіх компонентів МПП або сполучення їх між собою.

*Інформаційно-логічні вимоги* визначають структуру і склад ліній і сигналів, способи керування та формати даних, адрес і команд, протоколів обміну для різних режимів і фаз роботи.

*Внутрішньосистемний інтерфейс* найчастіше реалізують на основі спільної системної шини, за якою передаються адреси, дані й команди. Системна шина може мати окремі шини для адрес і даних або суміщені шини адрес/даних, які передаються мультиплексним способом.

Мікропроцесорні системи поділяються на універсальні, спеціалізовані й багатопроцесорні (рис. 10.1).





Рисунок 10.1 – Склад мікропроцесорних систем

Універсальні МПС орієнтовані на розв’язання широкого класу задач:

- *мікро-ЕОМ* – комп’ютер малих розмірів на основі мікропроцесора. Це найбільш поширений клас МПП для різних технічних систем на транспорті;
- *ПЕОМ* – персональні мікро-ЕОМ, розраховані на масового користувача;
- *робочі станції* – різні автоматизовані робочі місця;
- *сервери* – виконують частину функцій з обслуговуванням груп користувачів (розподіл ресурсів пам’яті, баз даних, принтерів та ін.);
- *мейнфрейми* – великі універсальні комп’ютери;
- *кластерні системи* – об’єднання машин з єдиним механізмом керування і програмного забезпечення, які забезпечують розподіл ресурсів, високу готовність, зручність розширення конфігурації.

Спеціалізовані МПС орієнтовані на вирішення спеціалізованих задач керування чи обробки інформації у складі технічних систем, цифрової обробки сигналів, обробки графіки.

Багатопроцесорні системи забезпечують функціонування багатьох процесорів під спільним керуванням.

*Основними технічними характеристиками* МПС є: розрядність, ємність пам’яті, продуктивність, число зовнішніх пристроїв та їхня пропускну здатність, функції системи і склад програмного забезпечення.

## 10.2 Загальна характеристика інтерфейсів

Всі сучасні МПС мають магістрально-модульну організацію, яка заснована на принципах агрегування і уніфікації. *Модульність (агрегування)* – це розділення МПС на прості функціонально й конструктивно закінчені блоки, які називаються модулями (модуль мікропроцесора, модуль пам’яті та ін.).

*Уніфікація* – оптимізація складу модулів, зв'язків між ними та їх конструктивного оформлення.

*Конструктивні вимоги* вказують на тип з'єднувальних елементів і розподіл ліній за їхніми контактами, геометричні розміри плат, каркаса та інші ознаки.

*Сигнальна лінія* – провідник (електричний ланцюг), який фізично поєднує джерело, і приймач інформації. Сукупність сигнальних ліній, за якими сигнали однакового функціонального призначення, називають *шиною*. Розрізняють шини даних, адреси, керування.

Інтерфейс класифікують за такими ознаками.

1. За функціональним призначенням:

- внутрішні – міжплатові й системні;
- зовнішні – для периферійних пристроїв, локальних мереж,

розподілених систем керування.

2. За напрямком обміну інформацією:

- симплексні – обмін в одному напрямку;
- напівдуплексні – почерговий обмін у двох напрямках;
- дуплексні – одночасний обмін у двох напрямках;
- мультиплексні – обмін реалізується за допомогою спільної шини

(магістралі), в якій в кожний момент часу взаємодіють джерело і приймач інформації.

3. За структурою:

- *радіальні* – реалізують за допомогою двонаправлених шин і заданого пріоритету абонентів (робочі станції, периферійні пристрої, схеми автоматики);
- *кільцеві* – де кожний абонент пов'язаний з двома сусідніми. У кільці можуть одночасно циркулювати декілька повідомлень від джерел до приймачів;
- *каскадні* – мають ланцюгове підключення абонентів, які обслуговуються в порядку їхнього підключення до ліній інтерфейсу;
- *магістральні* – використовуються в колективній шині (магістралі).

Інформація в магістралі доступна всім абонентам, які до неї підключені.

4. За способом передачі інформації в часі:

- синхронні – з фіксованою тривалістю операцій обміну;
- асинхронні – зі змінною тривалістю операцій обміну;
- синхронно-асинхронні – з комбінованим поєднанням двох способів.

5. За розрядністю даних розрізняють інтерфейси з *паралельним* (словами), *послідовним* (бітами) і *паралельно-послідовним* обмінами.

*Необхідно пам'ятати*, що термін «інтерфейс» застосовують як до апаратури МПП, так і до програмного забезпечення (оформлення екранного інтерфейсу, наприклад, з розміщеними на ньому всіх органів керування доступних користувачу програмно-керованого технічного засобу).

### **10.3 Підсистеми перетворення сигналів у мікропроцесорних пристроях**

Для здійснення функцій збору інформації керуючу систему обладнують ПЕ і підсистемою вводу/виводу аналогових і дискретних сигналів. На об'єктах

транспорту велика кількість контрольованих величин мають змінний характер в часі (величини струму, напруги, потужності та ін.), у зв'язку з чим МПП вимагають оснащення їх підсистемами вводу аналогової інформації.

Виконавчі механізми на транспорті також мають безліч конструкцій. Для керування ними в даний час використовуються інтелектуальні приводи, в основі яких лежать програмовані мікропроцесорні пристрої. Для ефективного керування виконавчими елементами МПП на транспорті оснащуються підсистемами виводу керуючих сигналів.

Порогові значення струмів, напруги, температури окремих вузлів і механізмів в системах автоматики на транспорті контролюються безліччю реле, що формують дискретні аналогові сигнали тільки в критичні моменти. Вихідні сигнали з таких пристроїв носять звичайно випадковий характер. Для вводу дискретної інформації такого вигляду в МПП використовуються підсистеми дискретних сигналів.

У загальному вигляді підсистема вводу аналогових сигналів має (рис. 10.2): датчики, кількість яких на транспорті визначається десятками, блок нормалізації (далі – БН), у складі якого підсилювач і джерело опорної напруги, мультиплексор (далі – МПЛ), аналогово-цифровий перетворювач та порт вводу МПП (порт інтерфейсу). Блок нормалізації забезпечує формування уніфікованих сигналів з ПЕ, чим досягається задана точність і надійність вимірюваних величин. Застосування мультиплексора дозволяє використовувати один АЦП для декількох каналів вимірювання величин, що скорочує витрати на реалізацію пристрою.

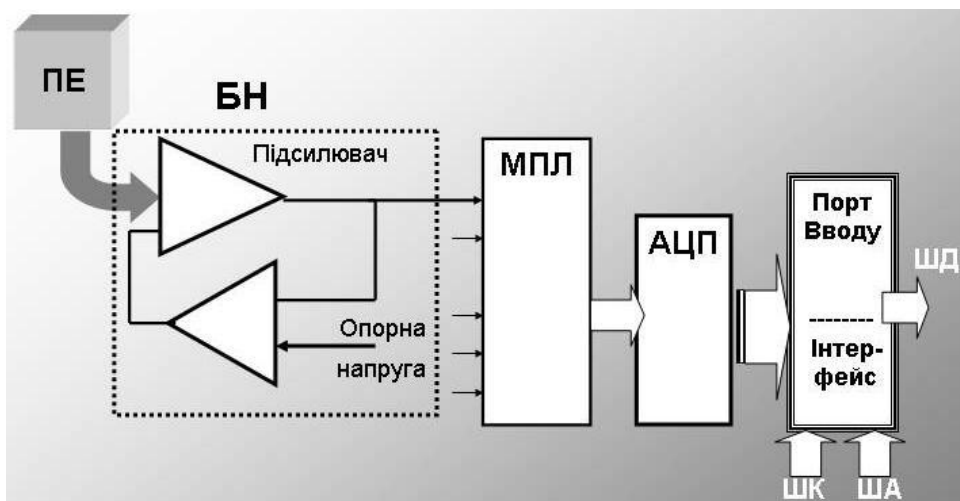


Рисунок 10.2 – Структура підсистеми вводу аналогових сигналів

АЦП виконує перетворення аналогової величини в цифровий код. Вибір типу АЦП треба робити, виходячи з мети його застосування з урахуванням таких найбільш важливих характеристик:

– *дозволюча здатність* – це мінімальне значення аналогового сигналу, яке може розпізнаватися перетворювачем:

$$\Delta U = \frac{|U_2 - U_1|}{2^n}, \quad (10.1)$$

де  $\Delta U$  – дозволяюча здатність, В;

$U_1, U_2$  – початкове й кінцеве значення діапазону вхідного аналогового сигналу, В;

$n$  – розрядність АЦП.

– *точність* (методична похибка) – у процесі квантування вхідного сигналу за рівнем відбувається його округлення до найближчого молодшого розряду цифрового коду, тобто виникає похибка квантування, що знаходиться у межах  $\pm 0,5\Delta U$ ;

– *час перетворення* (період квантування)  $\Delta t$  – проміжок часу, необхідний для перетворення в цифрову форму одного значення вхідного сигналу.

У промисловості рідко застосовують АЦП з розрядністю вище 12. Це пов'язано з тим, що індикатори друку результатів обмежують цю точність своїм дозволом на екранах або форматах друкарських аркушів.

Сучасні АЦП працюють за допомогою імпульсно-кодової модуляції. Безперервний сигнал представляється у вигляді послідовності відліків, які беруться через певний проміжок часу (з певною частотою дискретизації). Цю функцію виконує пристрій вибірки-зберігання. Запам'ятовуючи миттєве значення вхідного сигналу від ПЕ, цей пристрій забезпечує зберігання величини, відлік часу процесу оцифрування і уявлення амплітуди у формі кодового слова з певною кількістю розрядів.

Найбільш часто в МПП використовують АЦП порозрядного врівноваження. Принцип його дії заснований на послідовному порівнянні за допомогою компаратора вхідної напруги і напруги, що виробляється цифро-аналоговим перетворювачем (ЦАП), який входить до складу ЦАП.

Якщо на вхід  $n$  - розрядного ЦАП потрапляє код  $0\dots 001$ , то на його виході буде напруга, що відповідає елементарному кванту  $\Delta U$  (рис. 10.3).

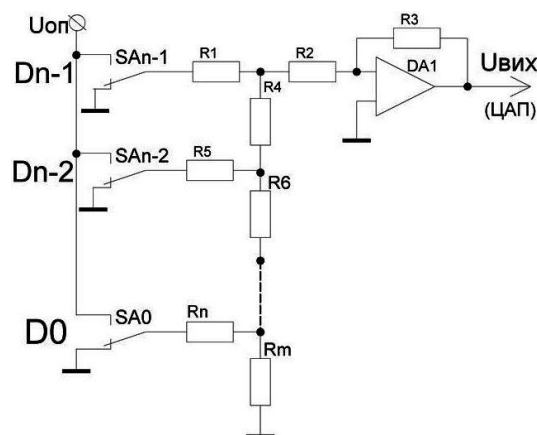


Рисунок 10.3 – Схема ЦАП:

$U_{on}$  – еталонна напруга;  $SA_0 - SA_{n-1}$  – електронні ключі;  $DA1$  – суматор

Резистори вимірювальної схеми підбрані таким чином, що вихідний сигнал  $U_{вих}$  пристрою при замиканні будь-якого ключа змінюється на величину, еквівалентну ваговому коефіцієнту, тобто,

$$U_{вих} = \Delta U (D_{n-1} \cdot 2^{n-1} + D_{n-2} \cdot 2^{n-2} + \dots + D_1 \cdot 2^1 + D_0 \cdot 2^0). \quad (10.2)$$

Таким чином реалізується принцип підсумовування аналогових величин, пропорційних вагам розрядів вхідного цифрового коду, розрядні коефіцієнти яких рівні одиниці.

Перетворення аналогових величин в цифрові коди є вимірювальними процесами. Перетворення відбувається шляхом порівняння перетворюваної величини з набором еталонних величин.

Існує декілька способів формування вихідного коду АЦП. Найбільший розвиток одержали послідовні АЦП з ЦАП в ланцюзі негативного зворотного зв'язку (НЗЗ). Прикладом такого пристрою може служити АЦП (рис. 10.4), що складається з компаратора DA1 і схеми керування F(t) або цифрового автомата.

В основі роботи такого АЦП лежить принцип порівняння вхідного сигналу  $U_{вх}$  з вихідним сигналом ЦАП  $U_{DA2}$  на виході операційного підсилювача DA2 (суматора). При рівності сигналів  $U_{DA2}$  і  $U_{вх}$  компаратор DA1 формує логічний сигнал, що зупиняє процес урівноваження. При цьому на виході АЦП буде сформований N – код, відповідний вхідної напруги:

$$U_{вх} = U_{он} \cdot N / N_0, \quad (10.3)$$

де  $U_{он}$  – напруга початку шкали відліку;

$N_0$  – код початку шкали відліку;

$N$  – код вимірюваної величини.

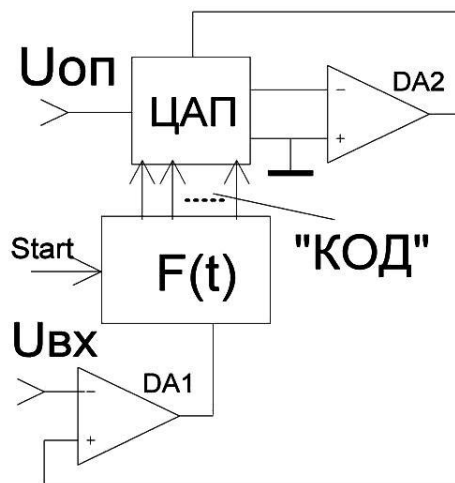


Рисунок 10.4 – Схема АЦП:

DA1 – компаратор; DA2 – суматор; F(t) – цифровий автомат

Статична похибка таких АЦП визначається похибкою ЦАП і компаратора.

Залежно від способу врівноваження шуканої величини існують різні АЦП, наприклад, розгортаючого, стежачого і порозрядного врівноваження.

Підсистему виводу аналогового сигналу з МПП до виконавчого механізму ілюструє рисунок 10.5.

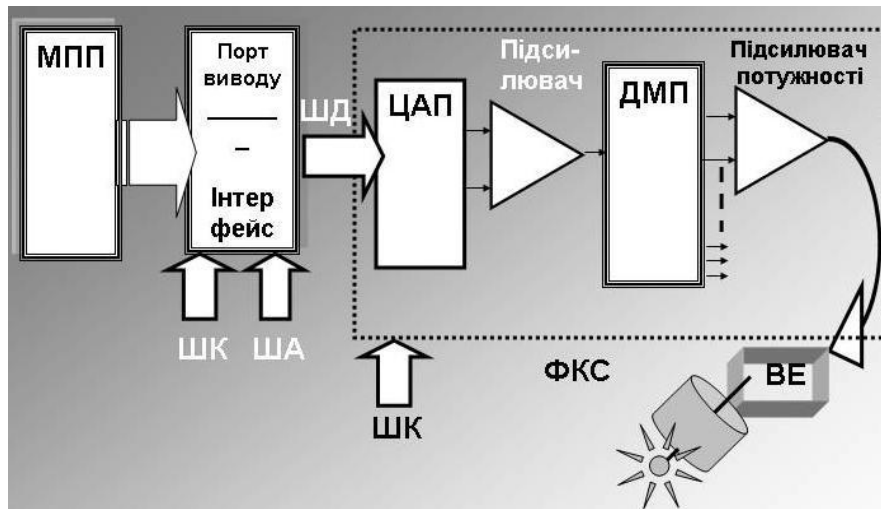


Рисунок 10.5 – Структурна схема підсистеми виводу аналогового сигналу

Формувач керуючого сигналу (далі – ФКС), що підключається до порту МПП, містить ЦАП, підсилювач напруги, демультимплексор і підсилювач потужності. Оскільки виконавчі елементи (далі – ВЕ) можуть мати різне призначення і електричні параметри, то очевидно, що підсилювачі потужності потрібні для них абсолютно різних конструкцій.

Сучасні підсилювачі потужності часто містять автономні мікропроцесорні пристрої, що забезпечують ефективну їх експлуатацію в різних кліматичних умовах, при різких змінах навантаження та інших контрольованих і неконтрольованих чинників.

Дискретні ПЕ виявляють перехід контрольованого параметра певного порогового значення. Вони можуть знаходитися лише в одному з двох станів, наприклад, «УВИМКНЕНО» або «ВИМКНЕНО».

Дискретні ПЕ підрозділяються за принципом дії на контактні й безконтактні. Перші мають контакт, який підключають до порту вводу МПП.

До безконтактного типу відносять ПЕ, створені на основі індукційних, оптичних та інших ефектів з характером зміни вихідного сигналу від контрольованої величини у вигляді стрибка, наприклад, фоторезистор. Для покращення роботи таких ПЕ використовують, як правило, компаратор, що зрівнює вихідний сигнал датчика  $U_{ПЕ}$  з деяким пороговим рівнем  $U_{ОП}$ . При  $U_{ПЕ} = U_{ОП}$  на виході компаратора відбувається зміна логічного сигналу. Приклади підключення дискретних ПЕ до порту МПП наведені на рисунку 10.6.

Зону нечутливості безконтактного ПЕ регулюють, вибираючи певне співвідношення між резисторами  $R1$  і  $R2$ . Подібні схеми використовуються при створенні диференціальних ПЕ з двома датчиками, що широко застосовується в системах захисту силового обладнання від помилкових спрацьовувань на різних об'єктах транспорту.

Основна функція пристроїв дискретного виводу – функція ключа. При цьому кожний двійковий розряд вихідних даних має самостійний зміст, тобто кожний біт вихідного слова може використовуватися для керування виконавчим елементом незалежно від інших (рис. 10.7).

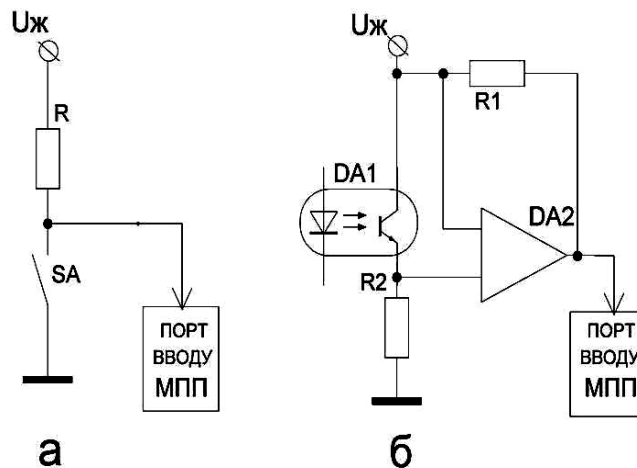


Рис. 10.6 – Схеми підключення дискретних ПЕ до МПП:  
*а – контактного типу; б – безконтактного типу*

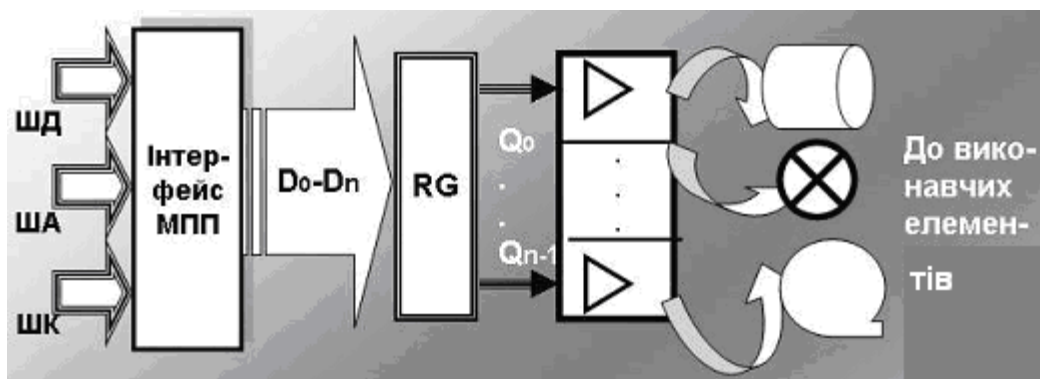


Рисунок 10.7 – Структурна схема підсистеми дискретного виводу МПП

Для узгодження електронних схем МПП з виконавчими пристроями застосовують підсилювачі. Транзисторні варіанти підсилювачів потужності є найпоширенішими на об'єктах транспорту (рис. 10.8). Номенклатура потужних транзисторів і тиристорів, що випускаються різними фірмами, в даний час дозволяє створювати схеми для ВЕ із струмами навантажень, кратними тисячам ампер.

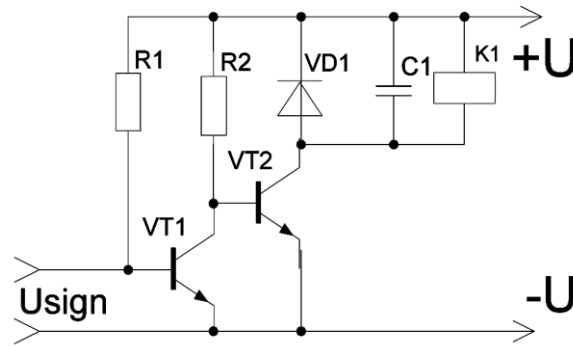


Рисунок 10.8 – Типова схема транзисторного підсилювача

У ланцюгах змінного струму на тягових підстанціях, в депо та інших об'єктах електричного транспорту найбільшою популярністю користуються симісторні підсилювачі потужності. Такі підсилювачі реалізують схеми з гальванічною розв'язкою, яку необхідно завжди мати між силовими блоками і електронними пристроями автоматики, особливо при використанні мікропроцесорів. Приклад симісторного підсилювача для керування електродвигуном ілюструє рисунок 10.9.

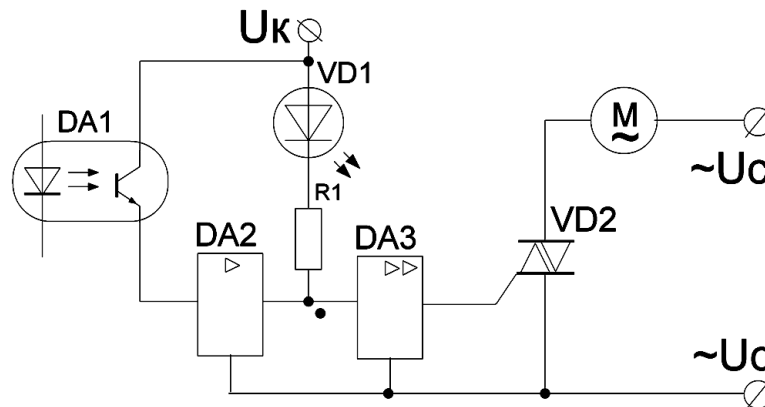


Рисунок 10.9 – Оригінальна схема симісторного підсилювача

Включення симісторного пристрою керування відображає світлодіод VD1, який використовується як локальний світловий індикатор подачі керуючого сигналу на електродвигун. Популярність таких індикаторів пов'язана з тим, що їх розміщення не вимагає додаткових монтажних робіт на щитах керування і організації лінії зв'язку від виконавчих пристроїв до засобів сигналізації.

Для більшості мікропроцесорних пристроїв, що випускаються серійними партіями, фірми-розробники створили підсилювачі з певними заданими технічними характеристиками. Номенклатура таких пристроїв забезпечує вибір необхідного з них для надійної експлуатації електроприводів із самими різними виконавчими механізмами.

Особливий інтерес викликають розробки, в яких мікропроцесорні пристрої забезпечують керування, наприклад, пневматичними або гідравлічними виконавчими елементами.



## 10.4. Послідовний і паралельний інтерфейси мікропроцесорних пристроїв

### 10.4.1 Загальна характеристика інтерфейсів

Сучасні периферійні пристрої підключаються до МПП і комп'ютерів за допомогою спеціалізованих адаптерів або контролерів.

*Адаптер* – це засіб сполучення деякого зовнішнього пристрою певною шиною. *Контролер* також виконує подібні функції, але він здатний на самостійні дії після одержання команд від процесора. Сьогодні терміни «адаптер» і «контролер» вважають майже синонімами. Усі зовнішні інтерфейси мають у своєму складі адаптер або контролер.

Мікропроцесорні комплекти завжди мають програмовні інтерфейсні контролери (адаптери). Наприклад, комплекти серії KP580 і K1810 мають:

- контролер прямого доступу до пам'яті – VT87;
- послідовний інтерфейс – VB51;
- паралельний інтерфейс – VB55 та ін.

### 10.4.2 Програмовний послідовний інтерфейс

У МПП для послідовного обміну інформацією між мікропроцесором і терміналом, наприклад, телетайпом, дисплеєм та ін. використовують програмовні послідовні інтерфейси (далі – ППП). ППП містить передавач (Transmitter – TX) і приймач (Receiver – RX). TX одержує від мікропроцесора по шині даних (ШД) дані (Data) в паралельному коді і передає їх послідовно за лінією TxD в термінал. Приймач одержує від терміналу за лінією RxD дані в послідовному коді, перетворює їх у паралельні слова і пересилає за ШД в мікропроцесор. Система обміну може бути асинхронною чи синхронною, симплексною, напівдуплексною чи дуплексною.

*Асинхронний режим* – послідовний обмін даними у вигляді одиночних символів (букв, цифр, знаків), які програмують на довжину 5–10 біт. Передавач автоматично обрамляє кожний символ додатковими службовими бітами: нульовим старт-бітом (початок передачі); інформаційними бітами, починаючи з молодшого розряду D0; бітом контролю на парність чи непарність; одиничним стоп-бітом (кінець передачі) тривалістю 1; 1,5; 2 біт. У такому форматі дані поступають від терміналу, а приймач ППП автоматично виключає службові біти. Часовий інтервал між символами при асинхронному обміні – фіксований.

При *синхронному обміні* стартові й стопові біти виключаються, а синхронізація здійснюється за допомогою одного або двох синхросигналів, які попередньо записуються в ППП. При синхронному прийманні приймач виконує побітне порівняння вхідного потоку даних із записаними синхросимволами, доки не виявить їхнього проходження. Після цього він інтерпретує кожну групу довжиною 5–8 біт (залежно від програмування) як символ. Символи ідуть один за одним безперервно. При неготовності даних передавач продовжує роботу, посилаючи для підтримки синхронізації спеціальні коди.

Синхронний режим обміну використовують для пересилок масивів даних; він підвищує швидкість обміну, яка вимірюється числом бітів, що пересилаються за секунду. Одиницю біт за секунду (біт/с) називають *бодом*.

Послідовна передача у всіх сучасних моделях мікроконтролерів та інших МПП здійснюється за стандартом RS232. Аналогічно називають послідовний інтерфейс – RS232 або COM1, COM2 і т.д. Вихідний рівень з боку TX: «0» відповідає сигналу від  $-15\text{ В}$  до  $-3\text{ В}$ , а «1» – від  $+3\text{ В}$  до  $+15\text{ В}$ . Проміжок від  $-3\text{ В}$  до  $+3\text{ В}$  відповідає невизначеному значенню.

Всі комунікації через COM – порти (Communications Port) організовуються програмними засобами.

Довжина сполучного кабелю не повинна перевищувати 15,24 м. Але на практиці ця відстань може бути значно більшою.

### 10.4.3 Програмовний паралельний інтерфейс

Паралельний інтерфейс призначений для організації паралельного обміну інформацією між МПП і периферією (клавіатура, індикатори та ін.).

*Паралельна передача даних* – система передачі, в якій кожен біт двійкового слова передається за своїм власним дротом (одиначна лінія). Таким чином, для передачі 8-розрядних слів буде потрібно 8 провідників. Один додатковий дріт необхідний як загальний або дріт заземлення (GND). З метою зменшення взаємних перешкод від ліній зв'язку кожен дріт утворює скручену пару з дротом GND.

Програмовний паралельний адаптер (далі – ППА) має:

- двонаправлений буфер даних, який підключається до системної шини МПП;
- блок сполучення і керування;
- регістри, які разом із серійними буферами створюють порти вводу–виводу. Порти зв'язані з периферією каналами;
- регістр керуючого слова.

Усі регістри, буфери і канали – восьмирозрядні.

Паралельний порт звичайно називають Centronics або LPT (Line Printer). Кожен порт має своє ім'я (наприклад, LPT1, LPT2) і свій адаптер, який може керувати регістрами: регістр вхідних даних; регістр статусу (звичайно статус принтера); регістр керування (керування принтером).

Порт паралельного виводу даних може працювати в нестандартному (незвичайному) режимі. Прикладом такого режиму є маніпуляція бітами даних, що вводяться (або виводяться). Вивод команд добре виконується за допомогою сигналів Data1–Data8 порту. Ввод сигналів від зовнішніх пристроїв виконується через виводи порту – ACK, BUSY, PE, ERROR. Рівні логічних «0» і «1» складають 0–0,5 В і 2,5–5 В відповідно.

Для комунікацій через LPT-порт використовується кабель із скрученими парами. Якісна робота зовнішніх пристроїв забезпечується при довжині ліній не більше 1,8 м.

Незалежно від призначення, наприклад, порту LPT1 для підключення до нього пристроїв друку електронних варіантів різних документів, цей порт часто застосовується для взаємодії МПП з периферійним обладнанням. Вирішення таких нетрадиційних задач завжди пов'язане із завданням створення оригінального блоку сполучення із зовнішніми пристроями-датчиками і виконавчими механізмами. Очевидно, що блок сполучення в такому МП-комплекті керується тільки програмними засобами.

#### 10.4.4 Послідовна універсальна шина (USB)

Шина USB (Universal Serial Bus) була розроблена в 1996 р. для розширення архітектури комп'ютерів. Шина забезпечує: високу швидкість передачі (від 12 до 480 Мбіт/с), одночасне підключення великої кількості пристроїв (до 127), зручність операцій, надійний зв'язок, а також можливість негайного підключення пристроїв.

Окрім сигнальної пари кабель USB має лінії передачі напруги живлення 5 В. Периферійні пристрої, що підключаються до шини, можуть живитися безпосередньо від шини (допустимий струм навантаження до 100 мА).

Протокол обміну даними за USB складний. З цієї причини багато фірм проводять необхідні власні дослідження і пропонують оригінальні розробки. У порівнянні з COM і LPT портами пропускну спроможність шини постійно нарощується.

Всі створені раніше периферійні пристрої, призначені для роботи через COM або LPT порти, до МПП з USB шиною підключаються до неї через спеціальні перетворювачі COM-USB, LPT-USB.

Довжина кабелю при високошвидкісному режимі не повинна перевищувати 5 м.

### 10.5 Контрольні питання

1. Що таке адаптер?
2. Що таке контролер?
3. Поясніть призначення послідовного інтерфейсу.
4. Поясніть асинхронний режим послідовного обміну даними.
5. Поясніть синхронний режим послідовного обміну даними.
6. Що таке інтерфейс RS232?
7. Дайте характеристику логічним "0" і "1" COM порта.
8. Поясніть призначення паралельного інтерфейсу.
9. Поясніть компоненти паралельного адаптера.
10. Що таке нестандартний режим LPT порта?
11. Дайте характеристику логічним "0" і "1" Centronics порта .
12. Що таке шина USB?
13. Які особливості USB шини в порівнянні з COM портом?
14. Які рівні уніфікованих сигналів використовуються в периферійних пристроях?

15. Як підключаються МПП з COM і LPT портами до USB шини?

## 11 ПРИКЛАДИ СКЛАДАННЯ ПРОГРАМ

### 11.1 Програма на асемблері

При складанні програм на асемблері кожна лінія програми підрозділяється на поля:

1. Адреса.
2. Код.
3. Мітка.
4. Мнемокод.
5. Коментарій.

У полі «Адреса» вказується адреса комірки пам'яті ОЗП до якої записується код операції або операнд. Усі адреси і коди записують у шістнадцятковій системі числення.

Міткою позначають окремі програми або їх фрагменти. Як правило, мітки проставляють в лініях, де вказані команди переходу. Мітка може мати обмежене число символів (звичайно 6).

У полі «Мнемокод» записують мнемонічне позначення операції, яку треба виконати.

У полі «Коментарій» дають необхідні пояснення виконаних операцій.

Наприклад, для створення програми складання трьох чисел 23D, 42D, 18D складемо концептуальну схему алгоритму:



Рисунок 11.1 – Концептуальна схема алгоритму

Функціональну схему алгоритму створимо з урахуванням структури типового мікропроцесора, зобразивши її на рисунку 11.2. Слід пам'ятати, що для запису програми необхідно перетворити десяткові числа у шістнадцяткові еквіваленти: 23D >> 17H; 42D >> 2AH; 18D >> 12H .

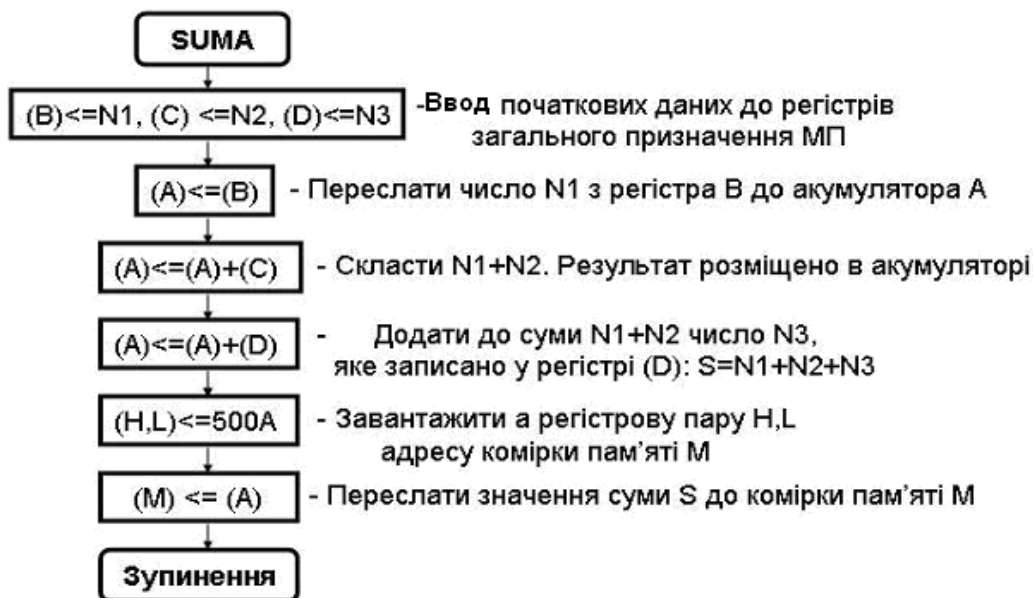


Рисунок 11.2 – Функціональна схема алгоритму

Програму на асемблері представимо у вигляді лістингу, використовуючи типову заготівку з раніше згаданими полями (табл. 10.1). Використовуючи інформацію про час виконання команд в МП, можна визначити час виконання всієї програми при певній тактовій частоті генератора МПП.

При підготовці програми за допомогою програми-редактора асемблера поле «КОД» не заповнюється. Ці коди «з'являються» після трансляції програми, якщо редактор не виявить помилок в тексті. Зберігаються програми придатними для їх завантаження в пам'ять МПП у вигляді кодованих текстів, прикладом яким і є поле «Код» лістингу. Якщо виконати проглядання програми в пам'яті МПП (ця операція завжди можлива за певною командою, наприклад: L 8 050 > Enter), то вона на екрані дисплея буде представлена в наступному вигляді:

8050 06 17 0E 2A 16 12 78 81 82 0A 50 77 76 FF FF FF . (11.1)

Таблиця 10.1 – Програма на асемблері

Адреса	Код	Мітка	Мнемокод	Коментарій
1	2	3	4	5
8050	06	SUMA:	MVI B,17	Завантажити: (B) <= N1
8051	17			
8052	0E		MVI C,2A	(C) <= N2
8053	2A			
8054	16		MVI D,12	(D) <= N3
8055	12			
8056	78		MOV A,B	Отримати суму (A) <= (B)
8057	81		ADD C	(A) <= (A) + (C)

Продовження таблиці 10.1

1	2	3	4	5
8058	82		ADD D	(A) <= (A) + (D)
8059	21		LXI H,500A	Запам'ятати результат (H,L) <= 500A
805A	0A			
805B	50			
805C	77		MOV M,A	(500A) <= (A)
805D	76		HLT	Зупинка

Очевидно, що перша група цифр  $i$  є адреса елемента пам'яті, з якою  $i$  планувалася нашим лістингом завантаження програми. Всі подальші осередки містять коди програми відповідно до лістингу. Такі тексти нерідко використовують для вивчення чужих готових програм при освоєнні асемблера.

При лінійній схемі алгоритму час виконання програми  $t_{np}$  дорівнюватиме сумі часу виконання окремих команд:

$$t_{np} = 1/F \cdot (n_1 + n_2 + \dots + n_m), \quad (11.2)$$

де  $F$  – частота тактових імпульсів, Гц;

$n$  – число тактів для команди;

$m$  – кількість команд.

## 11.2 Програма на BASIC

Завдання, що ілюструється алгоритмом на рисунку 10.1, на мові Basic матиме наступний вигляд:

```

100 REM ПРОГРАМА ВИЗНАЧЕННЯ СУМИ ТРЬОХ ВЕЛИЧИН
110 REM ВВОД ДАНИХ
120 PRINT «ВСІ ВЕЛИЧИНИ ВВОДЯТЬСЯ В ДЕСЯТКОВІЙ СИСТЕМІ»
130 INPUT B
140 INPUT C
150 INPUT D
160 REM ОБЧИСЛЕННЯ
170 LET A = B+C+D: REM ПРИ СЛОВІ LET ОБЧИСЛИТИ ЗА ФОРМУЛОЮ
180 PRINT «ВИВОД РЕЗУЛЬТАТУ (A) = »: A
190 END

```

Так текст програми виглядає на екрані дисплея. Після рядка «190» набирається слово RUN (виконати) і натискається клавіша <ENTER> – програма запускається. Якщо редактора мови Бейсік не виявив у тексті помилки, програма запропонує ввести початкові дані й виконує розрахунок, результат якого буде надрукований на екрані дисплея. Помилки в тексті програми редактор пропонує виправити, відзначаючи номер рядка після слова ERROR.

Очевидно, що програмування на мові Бейсік простіше, що дозволяє освоїти його в короткі терміни. Для програмування на мові BASIC можна скористатися програмним продуктом QBASIC 4.5 (Quik), який передбачений для установки на ПК з операційною системою Windows 95/98/Me.

### 11.3. Програма на C

Для програмування на мові Сі (С++ поліпшена версія мови С) слід пам'ятати, що кожна змінна спочатку оголошується. Оголошення всіх величин краще всього поміщати на початку функції. Інструкцію слід забезпечувати короткими коментарями, позначаючи їх символом «//». Інструкція оголошення будь-якої змінної має вигляд «Тип Ім'я Змінна;». Оскільки компілятор мови Сі розрізняє прописні й рядкові букви, то про це завжди слід пам'ятати. Основними числовими типами мови є: int (цілий) і float (дробовий). Функція printf забезпечує вивод повідомлень на екран. Вивод кількості цифр в результатах задається величиною «n».

Реалізацію алгоритму заданого на рис.10.1 ілюструє наступний текст:

```
// Вычисление суммы трех чисел
#include <stdio.h>
#include <conio.h>
void main()
{
float B,C,D; // начальные величины
float A;     // сумма
int n;      // количество суммарных членов
printf("Ввести начальные данные -> ");
A=B+C+D;
printf("Результат расчета: \n", A);
printf("\n Для завершения нажать <Enter>");
getch();
}
```

Рисунок 11.3 – Реалізація алгоритму

Після виправлення помилок програма транслюється і в кодованому вигляді «зашивається» в пам'ять МП, наприклад, мікроконтролера. Для цієї операції використовується програматор конкретного мікроконтролера.

Можливості мови Сі значно вищі, ніж мови Basic. Очевидно, що її освоєння вимагає великих витрат часу. Для вивчення мови Сі існує навчальна програма «Мандрівник», яка має докладні коментарії на англійській мові.

### 11.4 Контрольні питання

1. Поясніть поля лістингу програми на асемблері.

2. Поясніть призначення концептуальної схеми алгоритму.
3. Поясніть призначення функціональної схеми алгоритму.
4. Як виглядає програма в пам'яті МПП?
5. Чи можна визначити час виконання програми, використовуючи її лістинг?
6. Який вигляд має лістинг програми на мові Basic?
7. Поясніть призначення слів RUN, PRINT, INPUT, FOR, GOTO, GOSUB, END.
8. Які особливості програмування на мові Сі?
9. Пояснити популярність мови Сі при програмуванні МПП автоматики.

## 12 ПРОГРАМОВАНІ ЗАСОБИ АВТОМАТИКИ

### 12.1 Автоматична система контролю

До часто контрольованих технологічних величин на електричному транспорті слід віднести величини струму і напруги. Для створення мікропроцесорного пристрою для контролю будь-якого параметра  $X$  перш за все необхідно вибрати приймальний елемент. Вихідний сигнал датчика-перетворювача повинен бути нормованим. Пристрій порівняння повинен забезпечувати формування вихідного сигналу, коли контрольована величина відрізняється від заданого його значення  $X_0$ .

Якщо значення параметра  $X$  дорівнює або перевищує максимально припустиме значення  $X_0$ , то, наприклад, від дискретного датчика-перетворювача надходить сигнал логічної одиниці. Згідно з цим МПП вмикає виконавчий елемент  $K$ . Датчик  $I$  підімкнуто до біту D4 порту вводу № 3, а виконавчий пристрій  $K$  до біту D6 порту виводу № 4.

Формати вхідного слова МП:

D7	D6	D5	<b>D4 (від S)</b>	D3	D2	D1	D0
<b>Порт вводу №3</b>							

Формат вихідного слова МП:

<b>Порт виводу №4</b>							
D7	<b>D6 (до K)</b>	D5	D4	D3	D2	D1	D0

Згідно з прийняти форматом вхідного слова, якщо  $X < X_0$ , то від ПЕ  $S$  буде надходити сигнал логічного нуля і до акумулятора МП через порт вводу № 3 за командою вводу буде записано число 0. Якщо  $X \geq X_0$ , то від ПЕ  $S$  буде записано двійкове число 00010000B, тобто десяткове 16D. Таким чином, для перевірки стану ПЕ треба порівняти вхідне слово з числом 16D. Якщо в результаті виконання команди порівняння у розряд  $Z$  регістру ознак буде записано «0», то це означає, що від ПЕ  $S$  надходить сигнал логічного нуля. У протилежному разі, коли в розряд  $Z$  регістру ознак буде записана «1», то від ПЕ  $S$  надходить сигнал логічної одиниці. Для вмикання виконавчого пристрою  $K$  у



порт виводу № 4 можна записати двійкове число 01000000В, тобто 64D. У цьому разі в біт D6 порту виводу буде записана логічна одиниця. Для вимкнення ВЕ у порт виводу можна записати будь-яке двійкове число з нулем у біті D6, наприклад, 00000000В.

Функціональна схема розглянутого алгоритму роботи пристрою має вигляд, наведений на рисунку 12.1.

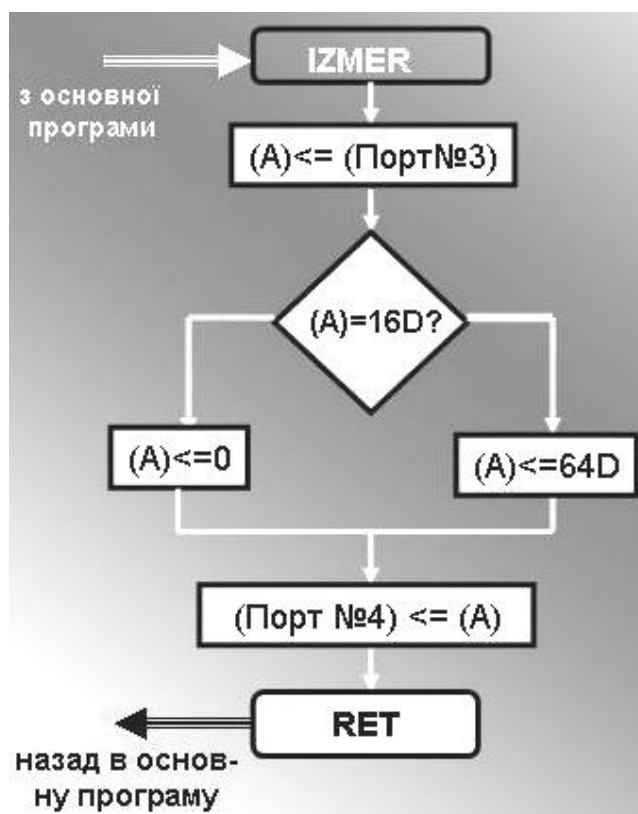


Рисунок 12.1 – Функціональна схема алгоритму контролю параметра об’єкта

Оскільки розглянутий алгоритм вимірювання технологічних величин можна використовувати в різних МПП, то такий програмний продукт можна рекомендувати для використання у вигляді підпрограми з міткою «IZMER». Лістинг програми контролю параметра  $X$  на асемблері представлений в таблиці 12.1.

## 12.2 Позиційний одноканальний керуючий пристрій

Цей пристрій – розробка лабораторії мікропроцесорних пристроїв ХНУМГ. Пристрій реалізований на базі мікро-ЕОМ з 8-розрядним мікропроцесором. Він дозволяє управляти силовими виконавчими механізмами залежно від поведінки контрольованої вхідної величини. Алгоритм роботи пристрою близький до ілюстрації на рисунку 12.1, але відрізняється тільки умовою порівняння параметра  $X > X_0$ .

Таблиця 12.1 – Лістинг підпрограми контролю параметра

Адреса	Код	Мітка	Мнемокод	Коментарій
8000 8001	DB 03	IZMER	IN 03	(A) <= (Порт №3)
8002 8003	FE 10		CPI 10	Поріняти (A) з числом 16D (16D = 10H)
8004 8005 8006	CA 0C 80		JZ 800C	Перехід, якщо (A)-16D = 0 на 800C
8007 8008	3E 00		MVI A,00	Вимкнути К (A) <= 0
8009 800A 800B	C3 0E 80		JMP 800E	Перехід на 800E
800C 800D	3E 40		MVI A,40	Вмикнути К (A) <= 64D (64D = 40H)
800E 800F	D3 04		OUT 04	(Порт № 4) <= (A)
8010	C9		RET	Повернення до основної програми

Перевищення поточним значенням величини X його граничного значення  $X_0$  характерне для всіх автоматичних вимикачів, якими оснащуються рухомий склад та інші об'єкти електричного транспорту. Параметром  $X_0$ , наприклад, вибирається струм уставки автоматичних вимикачів, встановлюваних в електричних ланцюгах захисту від коротких замикань. Для реалізації пристрою в пам'ять мікро-ЕОМ слід ввести наступний текст програми:

```

100 PRINT «KONTROL X»: INPUT K
200 KEY OFF: CLS: REM OCHISTKA EKRANA
300 B=1-K: REM SRAVNIT S USTAVKOJ TOKA
400 B=0: GOTO 1000: REM VKL IE
500 B=1: GOTO 100: REM POVTOR IZMERENIE
1000 MOTOR ON
1100 PRINT «KZ NA LINII»: REM INFO NA EKRANE
1200 FOR I=0 TO 1000: NEXT
1300 MOTOR OFF : REM PROVERKA KZ=0
1400 GOTO 100: CLS

```

На відміну від існуючих автоматичних вимикачів даний пристрій може періодично перевіряти стан лінії. Якщо виконуватиметься умова  $X \leq X_0$  (умова усунення короткого замикання), то програмований автоматичний вимикач підключить лінію до джерела живлення. Очевидно, що в серійних автоматичних вимикачах такий режим реалізувати неможливо.

Алгоритм роботи МПП легко змінити, вносячи відповідні зміни в програму. Пропоноване МПП легко перетворити на сигнальний пристрій, автомат включення устаткування в різні часові інтервали і т.п.

Зовнішній вигляд пристрою ілюструє рисунок 12.2.



Рисунок 12.2 – Одноканальний керуючий пристрій на базі мікро-ЕОМ

### 12.3 Восьмиканальний керуючий пристрій

Даний пристрій розроблений в лабораторії мікропроцесорних пристроїв ХНУМГ (рис. 12.3).

Мікропроцесорний пристрій призначений для автоматизації різних технологічних процесів. Одним з напрямків його застосування був вибраний процес обробки води, багато разів використовуваної при обслуговуванні рухомого транспорту в мийно-прибиральному корпусі (далі – МК) депо. Система багаторазового очищення технічної води знайшла широке застосування на таких об'єктах, забезпечуючи економію засобів підприємства, оскільки вода з міської водопровідної мережі використовується в малих кількостях.



Рисунок 12.3 – Восьмиканальний керуючий пристрій

Оскільки очищення технічної води пов'язане з тривалими процесами вилучення з неї різних домішок і шкідливих речовин, то очевидно, що процеси повинні бути автоматизовані. МПП забезпечує автоматизацію локальної лабораторії, контролюючої всі процеси водообробки в МК депо.

Програма восьмиканального МПП реалізує наступний алгоритм контролю якості очищеної води і устаткування для обслуговування транспорту:

- 1 – відключення контактної мережі МК;
- 2 – включення насоса подачі проби води для аналізу фотометричним методом;
- 3 – виконання процесу підготовки проби води для її контролю;
- 4 – виконання аналізу проби води і визначення середнього значення проведених вимірювань;
- 5 – логічна оцінка якості води: подача напруги в мережу живлення насосів МК при оцінці якості води, відповідній нормі й формування сигналу «НОРМА»;
- 6 – формування сигналу «ЗАБОРОНА» при залишковому вмісті домішок у воді що перевищують нормований рівень;
- 7 – підключення до контактної мережі МК джерела живлення зниженої напруги і відключення системи блокування всього електричного устаткування МК.

Для створення системи автоматичного контролю і керування МК восьмиканальний мікропроцесорний пристрій комплектується фотометричним датчиком-перетворювачем з мостовою вимірювальною схемою, оснащеною підсилювачем напруги. Такий пристрій забезпечує нормування вихідного сигналу аналізатора якості води. Всі виконавчі пристрої, що входять в комплект устаткування, підключаються до МПП за допомогою підсилювачів (див. рис. 10.8, 10.9), що забезпечують гальванічну розв'язку силових і керуючих ліній зв'язку системи автоматики.

Для формування часових затримок, які доцільно виконувати при керуванні виконавчими елементами на описаному вище об'єкті, в МПП реалізована підпрограма «ТІМЕ». Приклад лістингу «ТІМЕ» поданий у таблиці 12.2.

Для визначення числа  $N$  (записується у пару реєстрів) треба зробити розрахунок часу виконання підпрограми «ТІМЕ». Слід врахувати, що ця операція виконуватиметься один раз. Допустимо, що тактова частота МП 2 МГц. Час виконання програми ( $T$ ) визначимо за формулою:

$$T = t_{LXI} + (t_{MOV} + t_{ORA} + t_{JZ/NI} + t_{JMP}) \cdot (N - 1) + t_{MOV} + t_{ORA} + t_{JZ/NI} + t_{RET} \cdot (12.1)$$

З урахуванням потрібної кількості тактів для виконання кожної команди, враховуючи формулу (11.2), одержимо:

$$T = t \frac{1}{2 \cdot 10^6} \cdot (10 + (4 + 4 + 7 + 6 + 10) \cdot (N - 1) + 4 + 4 + 10 + 10) = 15,5(N - 1) + 19 \text{ (мкс)}.$$

Після вирішення останнього рівняння відносно числа  $N$  запишемо:

$$N = \frac{T - 3,5}{15,5}. \quad (12.2)$$

Таблиця 12.2 – Підпрограма формування витримку часу

Адреса	КОД	Мітка	Мнемокод	Коментарій
8100	01	TIME:	LXI B,N	(B,C) ← N
8101				← Молодша частина N
8102				← Старша частина N
8103	78		MOV A,B	(A) ← (B)
8104	B1		ORA C	(A) ← (A) V (C)
8105	CA		JZ 810C	Якщо (A) = 0, то перехід на адрес 810C
8106	0C			
8107	81			
8108	0B		DCX B	(B,C) ← (B,C) – 1
8109	C3		JMP 8103	Перехід на адрес 8103
810A	03			
810B	81			
810C	C9		RET	Повернення до основної програми

Оскільки час виконання підпрограми повинен дорівнювати необхідні витримці часу, наприклад  $0,5 \text{ с}$  ( $0,5 \cdot 10^6 \text{ мкс}$ ) або  $1 \text{ с}$  ( $1 \cdot 10^6 \text{ с}$ ), то

$$N1 = \frac{0,5 \cdot 10^6 - 3,5}{15,5} = 32258;$$

$$N2 = \frac{1 \cdot 10^6 - 3,5}{15,5} = 64516.$$

Після перетворення  $N1$  і  $N2$  у шістнадцяткові еквіваленти 7E02, FC04, відповідно ці величини і треба внести в підпрограму «TIME». Враховуючи, що в основній програмі реалізується декілька часових затримок, очевидно, що підпрограму «TIME» необхідно кілька разів копіювати з новими значеннями величини  $N$ .

Аналогічно діють, якщо потрібне формування часових затримок, що становлять, наприклад, декілька секунд. Для цього підпрограму «Time = 1с» повторюють необхідну кількість раз.

Для виконання підпрограми «TIME» і формування необхідної витримки часу потрібно в листигу підпрограми записати в полі «КОД» для рядка з адресою 8101 молодшу частину числа  $N$ , тобто 02(04), а для рядка з адресою 8102 – старшу частину, тобто 7E(FC).

#### 12.4 Автоматична система повторного включення тягової підстанції

Цей пристрій розроблений в лабораторії мікропроцесорних пристроїв ХНУМГ. Пристрій демонструє приклад організації автоматизованого робочого

місця (далі – АРМ) оператора тягової підстанції (далі – ТП). Популярність таких АРМ обумовлена величезними можливостями мікропроцесорної техніки, особливо, персональних комп'ютерів.

Програма «SauTP» для АРМ оператора ТП реалізує наступний алгоритм:

1. Контроль величини струму в лініях контактної мережі (передбачено три лінії).
2. Відключення випрямного агрегату ТП від контактної мережі при короткому замиканні або перевищенні допустимого значення струму в лінії.
3. Діагностику контактної мережі при відключеному випрямному агрегаті.
4. Запуск автоматичної системи повторного включення (АПВ-1) з регульованою часовою затримкою.
5. При неусувних неполадках в лінії включається друга часова затримка системи повторного включення (АПВ-2).
6. При неусувних неполадках система діагностики продовжує контролювати стан ураженої контактної мережі після реалізації класичних варіантів АПВ-1 і АПВ-2 (реалізує АПВ-3, АПВ-4 ...АПВ-N).
7. При усуненні неполадок в лінії забезпечується формування керуючого сигналу, що поступає на відповідний лінійний вимикач.
8. Передбачені автоматичний кільцевий і селективний режими контролю і керування ліній контактної мережі.

Система контролює всі лінії контактної мережі в автоматичному режимі, дозволяє змінювати часові уставки АПВ. Системою ведеться електронний журнал всіх дій системи в автоматичному режимі і оператора, якщо він використовує ручний режим, передбачений системою.

Код програми підготовлений в середовищі сучасної мови С++. Об'єктно-орієнтоване програмування на С++ дозволяє створювати програмні продукти для керуючих систем Windows 95/98/Me і вище. Можливість створення барвистого і зручного для користування інтерфейсу відкриває перспективу використання таких Windows-додатків на будь-яких комп'ютерах і ноутбуках (рис. 12.4).

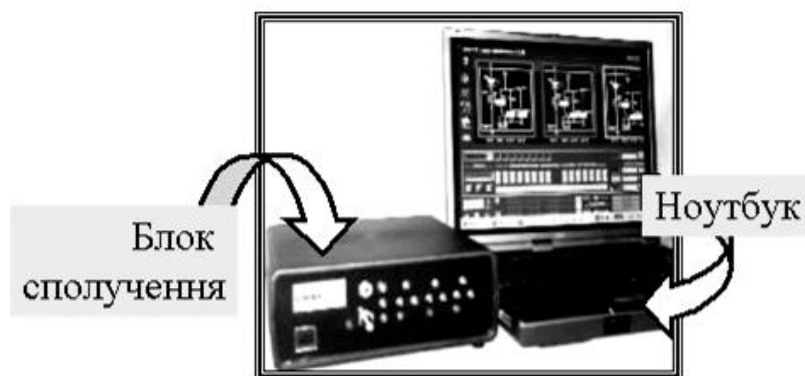


Рисунок 12.4 – Система керування «SauTP» для тягової підстанції з ноутбук

Інтерфейс користувача програми «SauTP» (рис. 12.5) має фрагмент оперативної схеми тягової підстанції і всі органи керування, забезпечуючи

оператору АРМ вибір необхідного режиму функціонування системи автоматики або ручним способом впливати на всі доступні виконавчі елементи. Система оснащена засобами сигналізації, що дозволяють оператору спостерігати за її діями і при необхідності приймати власні рішення, особливо при виникаючих неполадках на ТП і в контактній мережі.

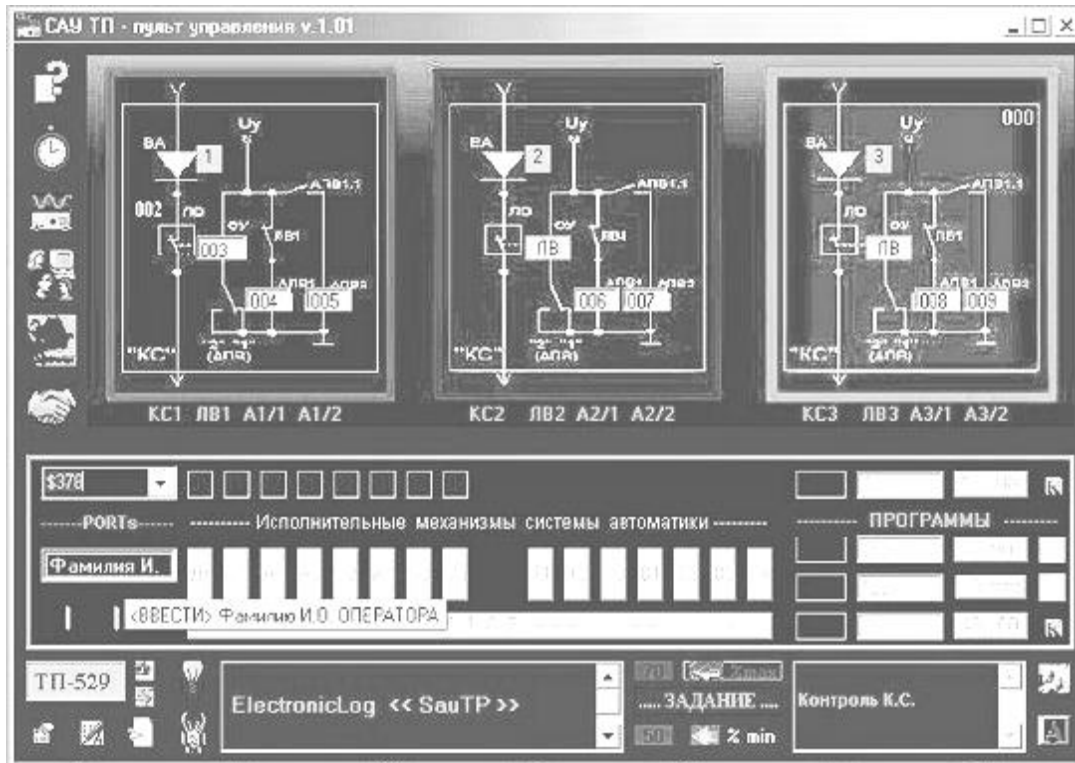


Рисунок 12.5 – Призначений для користувача інтерфейс програми «SauTP»

Для створення АРМ ТП необхідно до LPT-порту ПК підключити аналогово-цифровий перетворювач. 4-розрядний АЦП послідовного наближення зображений на рисунку 12.6.

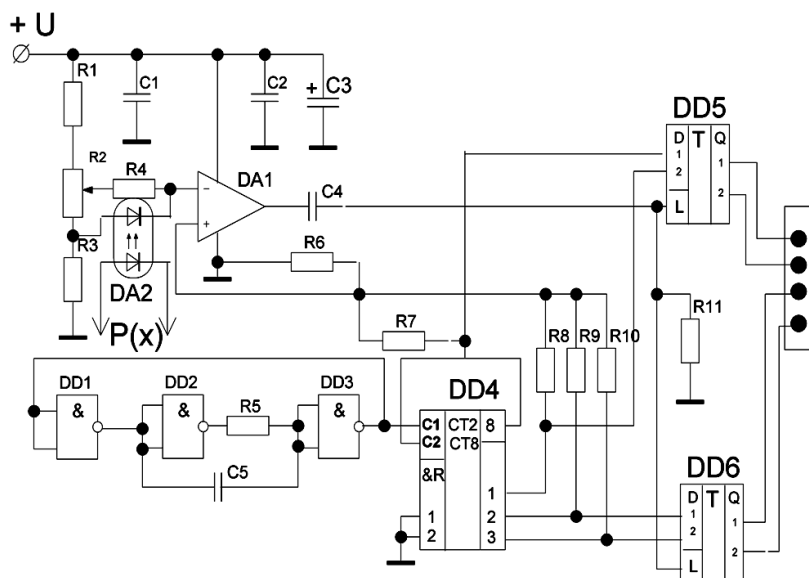


Рисунок 12.6 – Чотири розрядний ЦАП

ПЕ ЦАП може служити типовий трансформатор струму або оптичний перетворювач. ПЕ ступінчастого типу на оптоперетворювачах найбільш поширені в сучасних системах автоматики. Аналоговий сигнал з датчика DA2 поступає на інвертуючий вхід компаратора DA1, а на прямий вхід – напруга з виходу ЦАП, зібраного на мікросхемах DD1, DD2. Як тільки напруга на виході ЦАП зрівняється з напругою, яка подана на вході DA1, останній формує короткий позитивний імпульс. Цей імпульс подається на вхід DD3 і вирішує видачу коду ЦАП з DD2, який поступає на порт LPT ПК. Розглянутий принцип формування бінарного коду залежить від напруги на вході АЦП, а змінюється від величини струму  $P(x)$  в контактній мережі.

АЦП підключається до клем LPT-порту: 10,11,12,13.

Інші клеми LPT-порту ПК призначені для підключення до них підсилювачів з гальванічною розв'язкою (рис. 10.8, 10.9), призначених для формування керуючих сигналів: 1 – включення щита силового устаткування; 2 – живлення датчика струму; 3, 4 – АПВ-1/1, АПВ1/2; 5, 6 – АПВ2/1, АПВ2/2; 7, 8 – АПВ3/1, АПВ3/2; 9 – до керуючої обмотки лінійного вимикача; 14 – від ПЕ-1; 16 – від ПЕ-2; 17 – від ПЕ-3; 21-25 – земля.

Вибір часових затримок всіх АПВ передбачений в інтервалі від 1 с. до 999 с. Значення уставок регулюються оператором у відповідних вікнах інтерфейсу. Величина уставки по струму рівна 2400 А. Регулювання цього завдання допустиме в інтервалі від -40% до +40%.

Тривале використання таких програм на технологічних об'єктах звичайно приводить до їх постійного вдосконалення. З цієї причини з'являються оновлені версії, в яких усуваються виявлені недоліки і користувач знаходить нові функціональні можливості.

## **12.5 Автоматизоване робоче місце оператора мийно-прибирального комплексу депо**

Windows-додаток «SauMUK» розроблений в лабораторії мікропроцесорних пристроїв ХНУМГ для організації АРМ оператора МК депо. Інтерфейс користувача ілюструє лінію обслуговування рухомого складу депо, на постах якого розміщено різне технологічне устаткування (рис. 12.7).

Оператор МК на різних постах включає електрообладнання і вимикає його при завершенні відповідних робіт.

Дана програма ілюструє можливість автоматизації процесу керування всім устаткуванням на постах МК депо. Алгоритм роботи АРМ МК забезпечує:

- 1 – автоматичний і ручний режими керування устаткуванням МК;
- 2 – контроль потенціалу на кузові рухомої одиниці (РО);
- 3 – керування електричними приводами воріт на вході і виході МК;
- 4 – включення автономного низьковольтного джерела живлення контактної мережі в приміщенні МК і засобів контролю якості води;
- 5 – включення і виключення мийних барабанів;
- 6 – керування насосами подачі технічної води з місткостей локальної водоочисної установки;



7 – включення і виключення вентиляторів для видалення залишків води з поверхні РО.

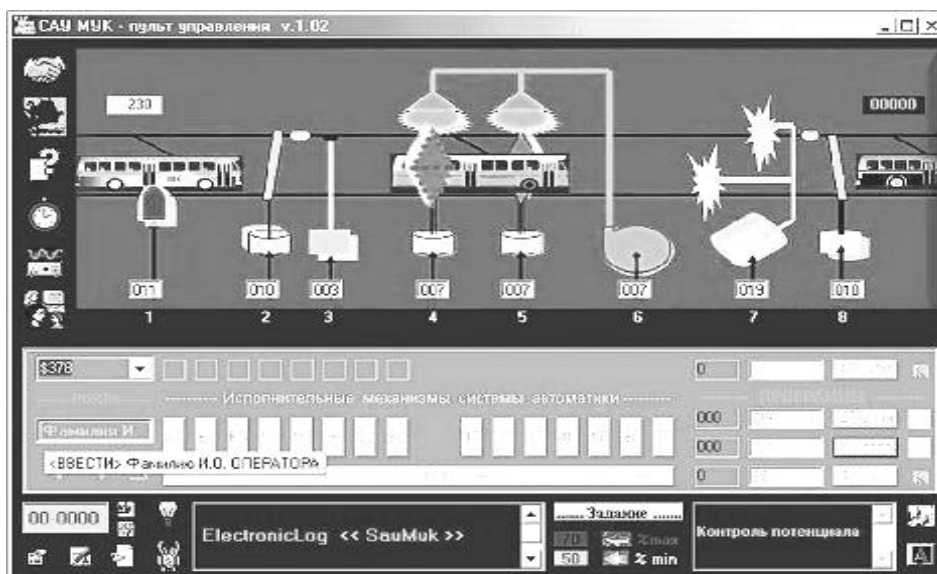


Рисунок 12.7 – Інтерфейс користувача Windows - додатку "SauMUK"

Для підключення до ПК периферійного устаткування використовуються АЦП (рис. 12.6) і підсилювачі з гальванічною розв'язкою (рис. 10.8, 10.9). Вищезгадані пристрої підключаються до 25-контактного розйому LPT-порту ПК відповідно (табл. 12.1).

Завдання за часом на постах обслуговування РО може бути вибрано в інтервалі від 1 с до 999 с. Вибір часових затримок дозволяє автоматизувати процеси включення і виключення устаткування, використовуваного при обслуговуванні рухомого транспорту, пересування якого в мийно-прибиральному корпусі здійснюється з постійною швидкістю.

Таблиця 12.1 – Розподіл контактів LPT1-розйому ПК

LPT-контакт	Призначення	LPT-контакт	Призначення
1	2	3	4
1	Блок контролю потенціалу	14	Освітлення постів
2	Привод вхідних воріт	15	Контроль якості води
3	Автономний блок живлення КМ	16	Світлофор
4	Мийні барабани – 1	17	Фільтри очистки води
5	Мийні барабани – 2	18	–

Продовження таблиці 12.1

1	2	3	4
6	Водяний насос	19	–
7	Сушильна установка	20	–
8	Привод вихідних воріт	21	GND
9	Дистанційний пульт керування	22	GND
10	АЦП ПЕ	23	GND
11	АЦП ПЕ	24	GND
12	АЦП ПЕ	25	GND
13	АЦП ПЕ	-	

### 12.6 Мікроконтролер діагностики устаткування

Даний пристрій розроблений в лабораторії мікропроцесорних пристроїв ХНУМГ і призначений для ілюстрації діагностики різного устаткування за допомогою автономного мікроконтролера (рис.12.8).



Рисунок 12.8 – Мікроконтролер діагностики устаткування

Використання мікроконтролера сімейства AVR (8 біт) обумовлено можливістю створення автономного пристрою, який з фотометричним або резистивним датчиком може проводити відбраковку виробів, які пройшли ремонт, перевірку, налаштування та інше обслуговування.

Контролер може бути виконаний як самостійний виріб або підключатися до локальної мережі автоматизованої ділянки, наприклад депо, з цехом з ремонту устаткування рухомих одиниць.

Оскільки діагностика устаткування виконується в певних умовах за допомогою спеціальних засобів, то очевидно, що для автоматизації таких процесів необхідне оригінальне програмне забезпечення. Приклад тексту програми для прошивки AVR - мікроконтролера, призначеного для автоматизації діагностики пневматичної системи РО, має наступний вигляд:

```
:100000003BC05FB62395441C331C221C111C001CF2
:10001000819B01C04E0E283261F5442D84D0052FFE
:10002000442D429580D0F52E432D7DD05068E52E8D
```

:10003000432D429578D0D52E422D75D0C52E422D18  
:10004000429571D05068B52E412D6DD0A52E412D11  
:10005000429569D0952E402D66D0852E402D429533  
:1000600062D0752E22270024112422243324442414  
:1000700066245FBEA89518954FE041BD112740E466  
:100080004BBF42E045BF40E84FBF49EF41BB46E0B0  
:1000900042BB4FEF47BB40E748B91FBDE1E0222715  
:1000A0000241124222433244424A8954FE3742EE1  
:1000B00046E0842E4BE5942E4FE4A42E46E6B42E63  
:1000C0004DE6C42E4DE7D42E47E0E42E4FE7F42E44  
:1000D0004FE6042FF6E0F2BB78BA69D0F7E0F2BB46  
:1000E00088BA65D0FEE0F2BB98BA61D0FFE0F2BBFF  
:1000F000A8BA5DD0F6E1F2BBB8BA59D0F7E1F2BB4A  
:10010000C8BA55D0FEE1F2BBD8BA51D0FFE1F2BB7C  
:10011000E8BA4DD0F6E2F2BBF8BA49D0F7E2F2BB4A  
:1001200008BB45D0D7CFA8954F704030F1F0413093  
:10013000F1F04230F1F04330F1F04430F1F045306D  
:10014000F1F04630F1F04730F1F04830F1F049304D  
:10015000F1F04A30F1F04B3011F14C3021F14D30DB  
:1001600021F14E30D1F04F30E1F050E01FC056E0A9  
:100170001DC05BE51BC05FE419C056E617C05DE615  
:1001800015C05DE713C057E011C05FE70FC05FE621  
:100190000DC05FE30BC050E009C053E707C051E753  
:1001A00005C050E403C05CE001C058E10895C3E01D  
:1001B000BFEF8A8950000BA95E1F7CA95C9F7662088  
:1001C00009F00895222700241124222433244424F2  
:0401D0006394089597

Зовнішній вигляд інтерфейсу Windows-додатку для ПК оператора АРМ показаний на рисунку 12.9.

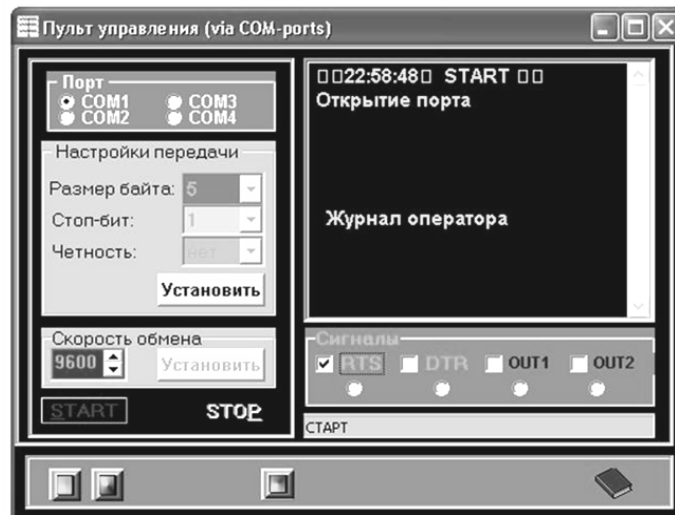


Рисунок 12.9 – Интерфейс АРМ для керування мікроконтролером діагностики

Використовуючи необхідну кількість таких автономних мікроконтролерів, можна створювати складні автоматичні системи діагностики, застосування яких забезпечує в короткий час проводити діагностику різного устаткування на РО. Прикладом може бути інтерфейс програми діагностики на рисунку 12.10.

Кожен мікроконтролер такої автоматизованої системи, оснащений індивідуальною оригінальною програмою і приймальними елементами, які

встановлюються на технологічному устаткуванні або стендах, призначений для випробування устаткування.

Програма ПК оператора АРМ ділянки діагностики забезпечує взаємозв'язок зі всіма мікроконтролерами через СОМ-порт. Очевидно, що такі мікропроцесорні пристрої відзначаються складністю. Проте, використання їх дозволяє суттєво скоротити час у визначенні неполадок і їх усуненні, забезпечуючи тим самим безпеку експлуатації рухомого транспорту при перевезенні пасажирів.



Рисунок 12.10 – Інтерфейс програми діагностики РО на базі автономних мікроконтролерів

Аналогічно створюються системи контролю пасажиропотоків, розробки маршрутних розкладів та їх коригування залежно від реальних ситуацій протягом доби і в триваліші періоди.

Особливий інтерес до таких пристроїв виявляється на об'єктах з автономним електричним живленням. Пов'язано це з незначними енерговитратами для функціонування мікроконтролерів на локальних об'єктах з необмеженим часом експлуатації.

Всі програми вищерозглянутих мікропроцесорних пристроїв входять в пакет навчальних програм «SinSys» версії v.1.12.55. Програми можуть використовуватися для експериментів як в навчальних цілях, так і для реалізації їх на об'єктах за допомогою персональних комп'ютерів з операційними системами Windows 98/Me/2000/XP.

## 12.7 Контрольні питання

1. Поясніть призначення підпрограм у програмах.
2. Що таке позиційний керуючий пристрій?
3. Як підключається виконавчий елемент до мікро-ЕОМ?
4. Поясніть особливості багатоканальних керуючих пристроїв.
5. Поясніть особливості сучасного програмування під ОС Windows.
6. Поясніть призначення інтерфейсу користувача Windows-додатків.

7. Охарактеризуйте відомі алгоритми систем автоматики на базі персональних комп'ютерів.

8. Охарактеризуйте особливості систем автоматики, що проектуються на базі мікроконтролерів.

9. Поясніть особливості LPT- і COM – портів ПК.

10. Які відомі мікропроцесорні системи автоматики, вживані на об'єктах електричного транспорту?

## 13 ЗАБЕЗПЕЧЕННЯ НАДІЙНОСТІ МІКРОПРОЦЕСОРНИХ ПРИСТРОЇВ

Великі економічні витрати через відмови в системах керування, а також строгі вимоги щодо безпеки на транспорті змушують приділяти особливу увагу створенню науковообґрунтованої організації експлуатації МПП на транспортних підприємствах.

У загальному випадку можна виділити такі види обслуговування МПП: зберігання; монтаж та налагодження на місці експлуатації; введення в експлуатацію; обслуговування під час нормальної роботи; планово-профілактичні роботи; усунення несправностей (ремонт); обслуговування програмного забезпечення та баз даних.

### 13.1 Характеристики експлуатаційної надійності

Висновок про реальні можливості пристрою керування на базі МП можна зробити тільки з урахуванням характеристик його експлуатаційної надійності.

Надійність є властивістю системи (або МПП) зберігати свої параметри у часі в заданих межах в певних умовах експлуатації. Надійність насамперед є функцією часу. Через деякий інтервал часу може статися подія, після якої система не виконуватиме своїх функцій у заданому обсязі й виходить з ладу. *Відмова* — це подія, що полягає в частковій або повній втраті роботоздатності виробу. Відмови бувають повні й переміжні.

За характером зміни вихідних параметрів системи відмов поділяються на *поступові* й *раптові*. Відмови можуть бути також *передбачуваними* або *випадковими*.

Безвідмовність системи керування на базі МП та МПП характеризують середнім часом  $T_c$  напрацювання на відмову:

$$T_c = \frac{t_1 + t_2 + \dots + t_n}{n}, \quad (13.1)$$

де  $t_1$  — час роботи до першої відмови;

$t_2$  — час роботи між першою і другою відмовами;

$t_n$  — час роботи між  $(n-1)$  та  $n$ -ю відмовами;

$n$  — число відмов.

Інтенсивність потоку відмов є середнім числом відмов в одиницю часу:

$$\lambda = \frac{1}{T_c}. \quad (13.2)$$

Типова крива параметра  $\lambda$  має вигляд, представлений на рисунку 13.1.

Для цієї кривої характерні три періоди: 1 – період пропрацювання; 2 – період нормальної роботи; 3 – період спрацювання і старіння.

Для періоду нормальної роботи, коли відмови мають випадковий характер і  $\lambda = \text{const}$ , ймовірність безвідмовної роботи протягом періоду часу  $t$  можна визначити за формулою:

$$P(t) = e^{-\lambda t} = e^{-\{t/T_c\}}. \quad (13.3)$$

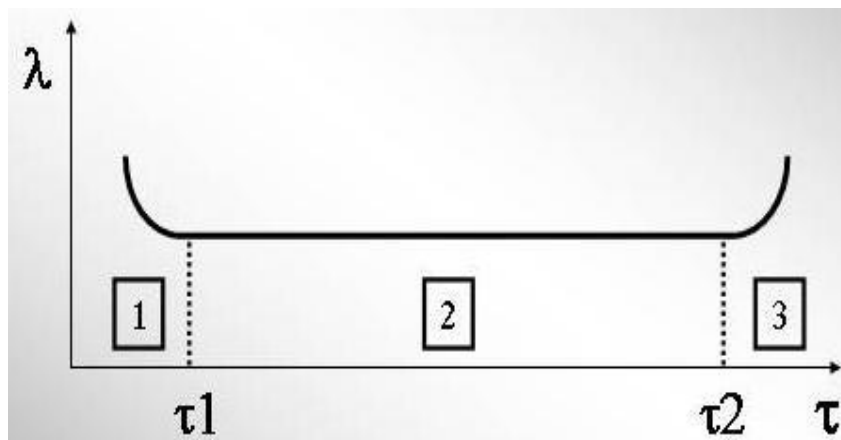


Рисунок 13.1 – Графік зміни інтенсивності потоку відмов системи

Якщо тривалість інтервалів між відмовами є випадковою величиною, яка має експоненційний розподіл, то ймовірність, що за час  $t$  відбудеться рівно  $k$  подій, визначається виразом:

$$P_k(t) = \frac{(\lambda t)^k}{k!} e^{-\lambda t}. \quad (13.4)$$

*Збої* – це короткочасне самоусуване порушення нормального функціонування системи або пристрою внаслідок дії на деякі елементи зовнішніх перешкод. Через збої виникає перекручення інформації.

### 13.2 Методи підвищення надійності мікропроцесорних керуючих систем автоматики

Резервування застосовують у випадках, коли треба забезпечити високий рівень надійності (насамперед безвідмовності) системи при недостатньо надійних її елементах. Залежно від виду використовуваної надмірності розрізняють структурне, інформаційне, часове й функціональне резервування.

*Структурне* – це резервування з використанням додаткових елементів структури системи. Резервні елементи приймають на себе функції основних елементів у разі відмови останніх.

*Інформаційне* – це резервування з використанням резервів інформації. Цей вид резервування застосовують, коли виникнення відмови або збою призводить до втрати або перекручення деякої частини інформації, що обробляється або переказується.

*Часове* – це резервування з використанням резервів часу. Структура системи і характер її функціонування у цьому випадку такі, що відмови й збої протягом деякого обмеженого часу не порушують роботи системи, а цей час відводять на ремонт елементів, які вийшли з ладу, або забезпечують багаторазове вирішення задачі за тією самою програмою.

*Функціональне* – це резервування з використанням функціональних резервів. Суть його полягає в тому, що при відмові одних елементів інші починають виконувати й додаткові для себе функції елементів що відмовили.

Найбільш поширеним є структурне резервування. Розрізняють два його види: загальне і окреме (поелементне)

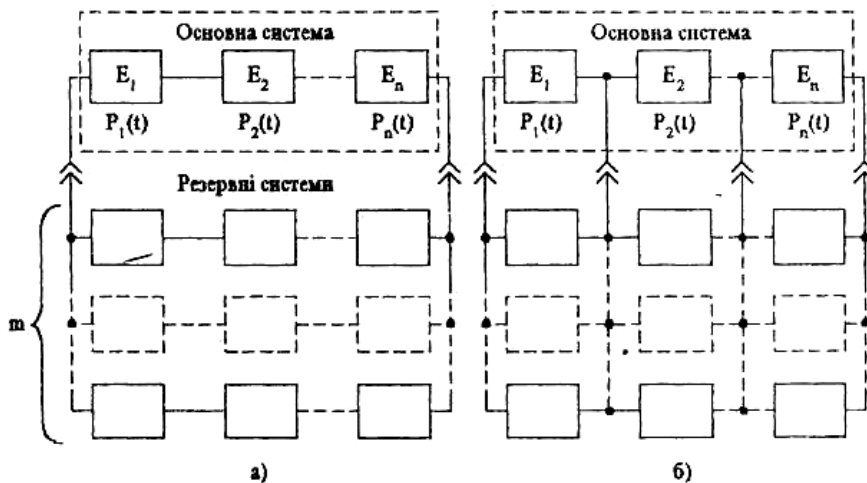


Рисунок 13.2 – Схеми структурного резервування:  
а – загальне; б – поелементне

Коли припустити, що резервні й резервовані елементи одного й того ж функціонального призначення мають однакові ймовірності безвідмовної роботи, то для всього з'єднання при загальному резервуванні:

$$P_3(t) = 1 - \left[ 1 - \prod_{i=1}^n p_i(t) \right]^{m+1}, \quad (13.5)$$

а по елементному:

$$P_n(t) = \prod_{i=1}^n \left\{ 1 - [1 - p_i(t)]^{m+1} \right\}. \quad (13.6)$$

**Приклад.** Мікропроцесорний вимірювальний пристрій складається з послідовно з'єднаних трьох блоків: датчика, мікропроцесорного блоку і блоку відображення інформації, ймовірність безвідмовної роботи яких дорівнює відповідно  $p_1(t) = 0.850$ ;  $p_2(t) = 0.980$ ;  $p_3(t) = 0.950$ .

Зробити розрахунок і порівняти ймовірність безвідмовної роботи всього з'єднання при загальному й поелементному резервуванні вимірювального приладу однією резервною системою.

При загальному резервуванні за формулою (13.5) маємо:

$$P_3(t) = 1 - [1 - p_1(t) \cdot p_2(t) \cdot p_3(t)]^{m+1} = 1 - [1 - 0.850 \cdot 0.980 \cdot 0.950]^2 = 0.956. \quad (13.7)$$

При поелементному резервуванні за формулою (13.6) визначимо:

$$\begin{aligned} P_n(t) &= \left\{ 1 - [1 - p_1(t)]^{m+1} \right\} \cdot \left\{ 1 - [1 - p_2(t)]^{m+1} \right\} \cdot \left\{ 1 - [1 - p_3(t)]^{m+1} \right\} = \\ &= [1 - (1 - 0.850)^2] \cdot [1 - (1 - 0.980)^2] \cdot [1 - (1 - 0.950)^2] = 0.975 \end{aligned} \quad (13.8)$$

Розрахунок показує, що надійність системи з поелементним резервуванням вища, ніж із загальним.

Поелементне резервування має перевагу з економічної точки зору, тому що будь-яка система містить в собі деяке число елементів, надійність яких значно вища надійності решти елементів. Тому резервування елементів з високою надійністю недоцільне.

Для підвищення надійності однопроцесорних керуючих систем застосовують додаткові апаратні й програмні засоби для виявлення помилок, що з'явилися внаслідок збоїв, за рахунок часової надмірності. Часткове розпізнавання і повне коригування наслідків збоїв виконується спеціальною підпрограмою контролю та діагностування. При цьому роблять розбиття обчислювального процесу на окремі ділянки і на початку кожної з них виконують дану підпрограму, що забезпечує повторення цієї ділянки, якщо на ній було зафіксовано збій. Повторення цієї ділянки основної програми і є коригуванням обчислювального процесу.

Якщо керуюча система має один процесор, то його відмова призведе до відмови всієї системи. Для забезпечення надійності такої системи необхідно ввести структурну надмірність, внаслідок чого керуюча система стає багатопроцесорною.

Наприклад, у двопроцесорній керуючій системі можливі три варіанти функціонування:



1) один процесор є ведучим, а другий знаходиться у навантаженому резерві. У разі відмови ведучого процесора організується переключення на резервний процесор і обчислювальний процес затримується на час становлення факту відмови і виконання переключення;

2) обидва процесори виконують обчислювання паралельно один одному за одним алгоритмом. При виникненні відмови одного з процесорів час на переключення не витрачається і затримки обчислювального процесу не буде;

3) обчислювання виконується одночасно на двох процесорах, але за різними напрямками розпаралеленого алгоритму. Це значно підвищує швидкість обчислювання, але потребує додаткових апаратних і програмних засобів. Якщо виникає відмова одного процесора, то обчислення може продовжити другий процесор, але швидкість обчислювання стає значно меншою.

Надійність керуючої системи значною мірою обумовлена надійністю програмного забезпечення. *Втрата* або *пропускання* команд, невірне їх виконання може призвести до повної або часткової відмови керуючої системи. Основою створення надійного програмного забезпечення є введення резервування, виявлення моменту програмної відмови або збою і повернення до відповідної ділянки програми, де відмови ще не було (рис. 13.3).

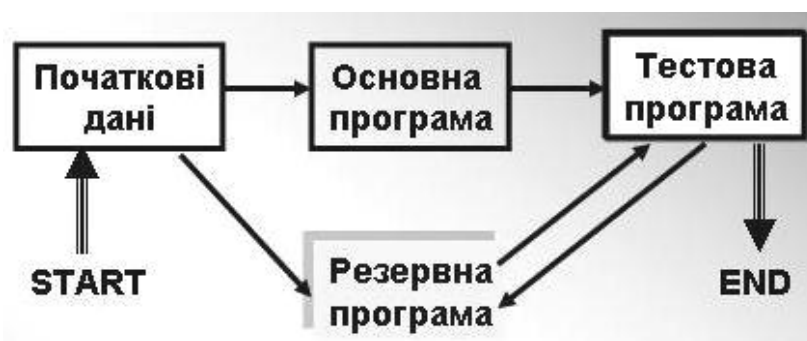


Рисунок 13.3 – Структура програмного забезпечення, стійкого до відмов

Для виключення одночасної програмної відмови основні й резервні алгоритми повинні використовувати різні схеми обчислювання.

### 13.3 Організація технічного обслуговування мікропроцесорних пристроїв

У системах керування, де пристроєм обробки інформації є мікропроцесор, найбільш часто відмови виникають у периферійному обладнанні (датчиках, виконавчих пристроях, приладах масової пам'яті та відображення інформації), у блоках електроживлення, в елементах з електромеханічними контактами та в контактних з'єднаннях (конвекторах). Тому є важливим своєчасне проведення технічного обслуговування (далі – ТО) або профілактичних робіт, під час яких виявляють пристрої, блоки й елементи з параметрами, наближеними до граничних які заміняють або регулюють.

*Технічне обслуговування* – комплекс операцій або видів робіт з підтримання роботоздатності або справності технічних засобів при їх використанні за призначенням, зберіганні або транспортуванні.

Організація ТО передбачає підготовку обслуговуючого персоналу, планування та проведення робіт з ТО, забезпечення персоналу запасними частинами, інструментами, приладдям і матеріалами, систематичне ведення відповідної експлуатаційної документації.

Для технічних засобів керуючих систем періодичність ТО заснована на календарному принципі, що передбачає проведення робіт з ТО після проходження певного терміну (доба, тиждень, місяць, квартал, півріччя, рік), незалежно від інтенсивності використання пристроїв та їх напрацювання.

ТО здійснюється шляхом періодичної перевірки всіх пристроїв системи з метою визначення їх роботоздатності і виявлення несправностей. При цьому перевірки підлягають як працюючі пристрої, так і пристрої, що зберігаються на складі.

*Ремонтпридатність МП* або МПП – це ступінь пристосування пристрою до попередження, виявлення та усунення відмов, вимірюється середнім часом відновлення роботоздатності після відмови.

Коефіцієнт використання МПП визначають за формулою:

$$k_{\text{в}} = \frac{T_c}{T_c + t_{\text{в}} + t_n}, \quad (13.9)$$

де  $t_{\text{в}}$  – середній час відновлення роботоздатності (ремонт);

$t_n$  – середній час профілактичних робіт на одну відмову.

*Коефіцієнт готовності* пристрою або системи показує відношення середнього часу справної роботи до середнього часу відновлення пристрою:

$$k_r = \frac{T_c}{T_c + t_{\text{в}}}. \quad (13.8)$$

Прийнято, що під час ТО відбувається повне відновлення надійності пристрою або системи. ТО виконується через рівні проміжки часу  $T_n$ , причому тривалість ТО повинна бути набагато менша, ніж  $T_n$ .

*Середній час безвідмовної роботи* для систем з ТО (профілактикою) визначається за формулою:

$$T_{\text{сн}} = \frac{T_n}{1 - P(T_n)}, \quad (13.9)$$

де  $T_n$  – періодичність ТО (профілактичних робіт);

$P(T_n)$  – імовірність безвідмовної роботи пристрою за період між ТО.

З урахуванням періодичності ТО у відповідному підрозділі підприємства, що забезпечує експлуатацію мікропроцесорних керуючих систем, складають календарний графік ТО.

Основними видами робіт з ТО є перевірка робоздатності пристроїв, внутрішній і зовнішній огляд пристроїв з очищенням та кріпленням всіх контактів, окраска корпусів приладів, якщо вони знаходяться під дією агресивного середовища. При виявленні відмов у роботі пристроїв проводять поточний ремонт методом заміни несправного блока з його наступним ремонтом у спеціальній майстерні.

Для проведення ТО МП використовують спеціальну контрольно-діагностичну апаратуру.

При виконанні ТО ведуть відповідну документацію, до якої записують зміст робіт і параметри стану технічних засобів, вказують дані осіб, які виконують ТО. Основним документом є картка обліку, яку оформляють на кожний пристрій системи. Для МПП або МП оформляють журнал обліку, в якому фіксують всі відмови та збої, що виникають під час експлуатації, дані про проведення профілактичних і ремонтних робіт. На основі цих даних коригують плани і уточнюють методику проведення ТО з метою підвищення надійності окремих пристроїв і керуючої системи в цілому.

### 13.4 Контрольні питання

1. Поясніть причини відмови в системах автоматики.
2. Дайте характеристику періодам роботи МПП до їх відмови.
3. Які існують методи підвищення надійності МПП?
4. Поясніть особливості одно- і двопроцесорних керуючих систем.
5. Нарисуйте і поясніть структуру програмного забезпечення, стійкого до відмов.
6. Поясніть організацію технічного обслуговування МПП.
7. Що таке ремонтпридатність МПП?
8. Що таке коефіцієнт готовності?
9. Які основні види робіт з ТО МПП?
10. Як оформляються результати ТО МПП?

## 14 ПРИКЛАДИ

### 14.1 Системи числення

**Мета:** засвоїти принципи перетворення чисел різних систем числення DEC, BIN, HEX і їх застосування для розрахунків параметричних величин цифрових перетворювачів: аналого-цифрових (АЦП) і цифро-аналогових (ЦАП).

### АЛГОРИТМ ПЕРЕТВОРЕННЯ:

1. Розділити число на основі системи числення.
2. Записати частинне від ділення і залишок.
3. Якщо частинне від ділення НЕ РІВНЕ НУЛЮ, слід повторити перший пункт.
4. Якщо частинне від ділення РІВНЕ НУЛЮ, то залишки від ділення записати в зворотному порядку.

**Приклад.** 57 – в десятковій системі (DEC) слід перетворити в двійкову (BIN):

$$57(10 \dots X(B))?$$

$$57 : 2 = 28 + 1$$

$$28 : 2 = 14 + 0$$

$$14 : 2 = 7 + 0$$

$$7 : 2 = 3 + 1$$

$$3 : 2 = 1 + 1$$

$$1 : 2 = 0 + 1$$

Таким чином одержали двійкове число:  $57D \rightarrow 111001B$ .

### АЛГОРИТМ ЗВОРОТНОГО ПЕРЕТВОРЕННЯ:

1. Необхідно кожен розряд помножити на його ВАГОВИЙ КОЕФІЦІЄНТ.
  2. Скласти одержані добутки:
- |       |       |       |       |       |       |       |       |                        |
|-------|-------|-------|-------|-------|-------|-------|-------|------------------------|
| D7    | D6    | D5    | D4    | D3    | D2    | D1    | D0    | – позначення БІТ       |
| $2^7$ | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | – ваговий коефіцієнт   |
| 128   | 64    | 32    | 16    | 8     | 4     | 2     | 1     | – величина коефіцієнта |

**Приклад.** Перетворити число 111001B – > ?D:

$$1 * 1 = 1$$

$$0 * 2 = 0$$

$$0 * 4 = 0$$

$$1 * 8 = 8$$

$$1 * 16 = 16$$

$$1 * 32 = 32$$

$$32 + 16 + 8 + 0 + 0 + 1 = 57$$

Таким чином одержали двійкове число:  $111001B \rightarrow 57D$ .

### ПЕРЕТВОРЕННЯ В ШІСТНАДЦЯТКОВУ СИСТЕМУ:

Необхідно знати представлення десяткових чисел від нуля до 15 у системах числення з основою 2, 16, 2–10 (табл. 14.1).

Таблиця 14.1 – Представлення чисел у різних системах числення

DEC	BIN	HEX	Bin-Dec
1	2	3	4
0	0	0	0000
1	1	1	0001
2	10	2	0010
3	11	3	0011
4	100	4	0100

Продовження таблиці 14.1

1	2	3	4
5	101	5	0101
6	110	6	0110
7	111	7	0111
8	1000	8	1000
9	1001	9	1001
10	1010	A	0001 0000
11	1011	B	0001 0001
12	1100	C	0001 0010
13	1101	D	0001 0011
14	1110	E	0001 0101
15	1111	F	0001 0110

**Приклад.** Перетворити число **60D** → ?H:

$$60 : 16 = 3 + 12 \quad 12 \rightarrow \text{CH}$$

$$3 : 16 = 0 + 3 \quad 3 \rightarrow 3\text{H}$$

Таким чином одержали: **60D** → **3CH**.

АЛГОРИТМ ЗВОРОТНОГО ПЕРЕТВОРЕННЯ:

**3CH** → ?D

$$\text{CH} \rightarrow 12\text{D} * 16^0 = 12 * 1 = 12;$$

$$3\text{H} \rightarrow 3\text{D} * 16^1 = 3 * 16 = 48$$

$$3\text{CH} \rightarrow 48 + 12 = 60\text{D}.$$

ПЕРЕТВОРЕННЯ «HEX → BIN»:

1. Необхідно кожен розряд 16-річного числа виразити двійковою тетрадою скориставшись табл.14.1.

2. Старшу тетраду (справа) доповнити нулями.

3. Всі тетради записати в послідовності від старшої до молодшої.

**Приклад.** Перетворити число **3CH** → ?B:

$$\text{CH} \rightarrow 1100\text{B};$$

$$3\text{H} \rightarrow 0011\text{B};$$

$$3\text{CH} \rightarrow 00111100\text{B}.$$

ПЕРЕТВОРЕННЯ «BIN → HEX»:

**Приклад.** Перетворити число **00111100B** → ?H:

бінарне число слід розбити на тетради: 0011 1100;

замінімо тетради, скориставшись таблицею 14.1:

$$0011\text{B} \rightarrow 3\text{H};$$

$$1100\text{B} \rightarrow \text{CH}.$$

$$00111100\text{B} \rightarrow 3\text{CH}.$$

Для перевірки перетворень слід скористатися програмою SinSys.



Рисунок 14.1 – Інтерфейс САК-1т програми SinSys з перетворювачем чисел в різні системи числення

**Завдання.** Виконати перетворення числової інформації відповідно до індивідуального завдання.

## 14.2 Синтез безконтактних логічних пристроїв

**Мета:** познайомитися з алгеброю релейних ланцюгів, принципами моделювання схем автоматики, мінімізацією математичних описів і використанням їх для розробки безконтактних пристроїв автоматики на логічних елементах.

Алгебра логіки виникла на підставі залежностей і законів алгебри логіки (аксіоми і закони алгебри релейних ланцюгів – АРЛ).

Таблиця 14.2 – Функції і аксіоми релейних ланцюгів

№ з/п	ФОРМУЛА	СХЕМА	НАЙМЕНУВАННЯ
1	2	3	4
<b>ЗАКОНИ</b>			
1	$X \vee 0 = X$	14.3.1	<i>Закон нульової безлічі:</i> $X \vee 0 = X$ $X = Y$ - логічне складання
2	$X * 0 = 0$	14.3.2	Завжди $Y=0$ .
3	$X \vee 1 = 1$	14.3.3	<i>Закон універсальної множини :</i> від стану контакту $X$ не залежить

Продовження таблиці 14.2

1	2	3	4
4	$X * 1 = X$	14.3.4	<i>Закон універсальної множини :</i> повністю залежить від стану X.
5	$X * X = XX = X$	14.3.5	<i>Закон повторення (I):</i> логічне множення (кон'юнкція) – приводяться в дію обидва контакти одночасно
6	$X \vee X = X$	14.3.6	Логічне складання (АБО) – диз'юнкція
7	$X * \bar{X} = 0$	14.3.7	<i>Закон логічної суперечності:</i> контакт з інверсією (ланцюг ніколи не включиться)
8	$X \vee \bar{X} = 1$	14.3.8	<i>Закон суперечності:</i> одинична логіка – завжди ланцюг включений
9	$X = \bar{\bar{X}}$	–	Подвійне заперечення
<b>ЛОГІЧНІ ФУНКЦІЇ ЗМІННИХ</b>			
10	$X1 \vee X2 = X2 \vee X1$	14.3.10	Від зміни місць результат не залежить
11	$X1 \cdot X2 = X2 \cdot X1$	14.3.11	Від зміни місць результат не залежить
12	$(X1 \vee X2) \vee X3 = X1 \vee (X2 \vee X3)$	–	Сполучний закон
13	$(X1 \cdot X2) \cdot X3 = X1 \cdot (X2 \cdot X3)$	14.3.13	Сполучний закон
<b>ЗАКОНИ ПЕРЕТВОРЕННЯ ЛОГІЧНИХ ФУНКЦІЙ</b>			
14	$X1 \cdot (X2 \vee X3) = X1 \cdot X2 \vee X1 \cdot X3$	–	Правило розкриття дужок
15	$X1 \vee (X2 \cdot X3) = (X1 \vee X2) \cdot (X1 \vee X3)$	14.3.15	Правило розкриття дужок
16	$X1 X2 \vee X1 \bar{X2} = X1 (X1 \vee X2) \cdot (X1 \vee \bar{X2}) = X1$	14.3.16	Правило склеювання .
17	$X1 \vee X1 X2 = X1$ $X1 (X1 \vee X2) = X1$	14.3.17	Правило поглинання
18	$\overline{(X1 X2)} = \bar{X1} \vee \bar{X2}$ $\overline{(X1 \vee X2)} = \bar{X1} \cdot \bar{X2}$	14.3.18	Закон Де Моргана дозволяє перейти від логічного множення до логічному складанню і назад
«V» - логічне складання (диз'юнкція); «*» - логічне множення (кон'юнкція)			

Еквівалентні схеми, що ілюструють логіку ланцюгів (табл.14.2), представлені на рисунку 14.2.

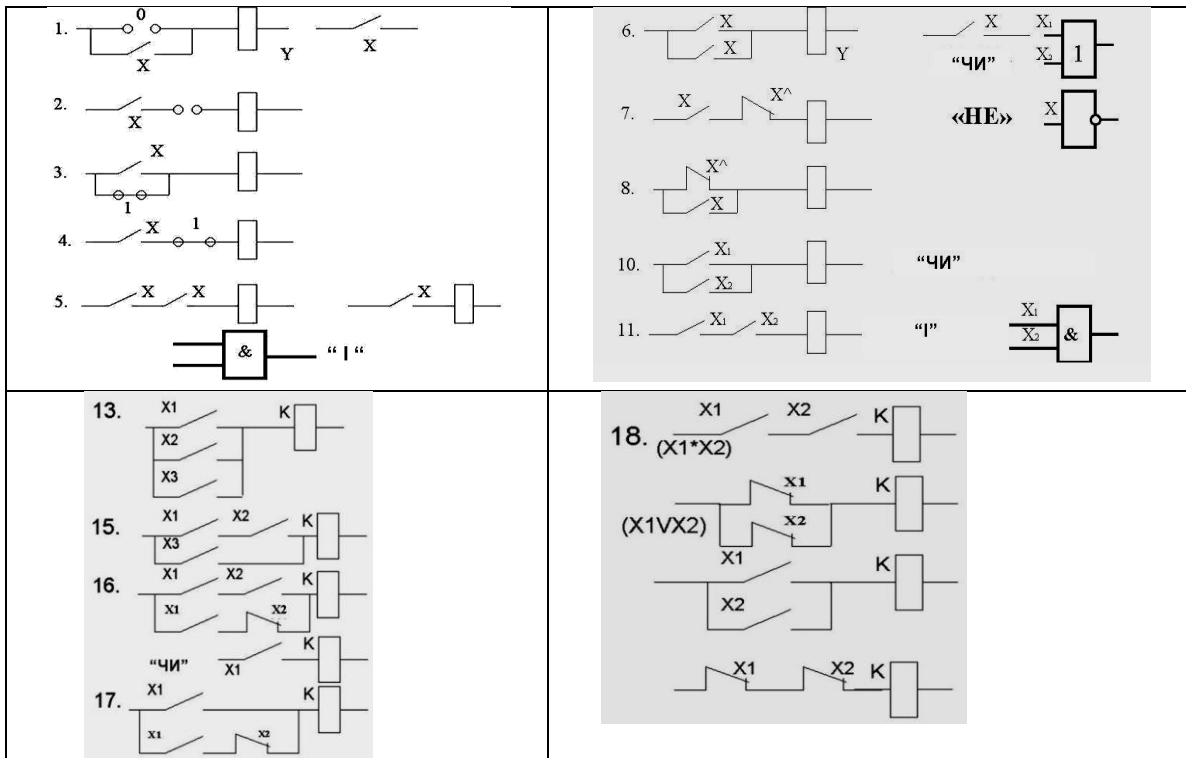


Рисунок 14.2 – Еквівалентні схеми ланцюгів

Для синтезу (складання) функціональної схеми автоматичного пристрою необхідно:

1. Скласти словесний опис технологічного процесу керування.
2. Визначити аргументи  $X_i$  (вхідні сигнали) і функції  $Y_i$  (вихідні сигнали) пристрою автоматики, за якими слід скласти таблицю істинності для всіх можливих варіантів наборів аргументів.
3. Записати логічне рівняння в аналітичній формі.
4. Виконати мінімізацію логічного рівняння з використанням АРЛ.
5. Приступити до складання функціональної схеми пристрою автоматичного керування в прийнятій базисі логічних елементів.

**Приклад.** При вході в РО автомат повинен включити освітлення без участі водія. Датчик контролю присутності людини містить виконавчий пристрій-реле з контактами, включеними в ланцюг живлення освітлювальних приладів. Треба скласти схему алгоритму роботи автомата, ввести позначення елементів.

1. *Описати алгоритм роботи автомата* із згадкою позначень, використовуваних в схемі елементів (рис. 14.3).

а) Якщо контакт  $X_3$  замкнута і замкнута один з контактів  $X_1$  або  $X_2$ , то контакти реле  $Y$  замикають ланцюг живлення лампи;

б) Якщо  $X_3$  розімкнений, реле  $Y$  знеструмлене, а його контакти розімкнені, ланцюг живлення лампи розірваний. Цей режим зберігається при будь-яких положеннях контактів  $X_1$  і  $X_2$  (не залежить від стану контактів  $X_1$  і  $X_2$ ).



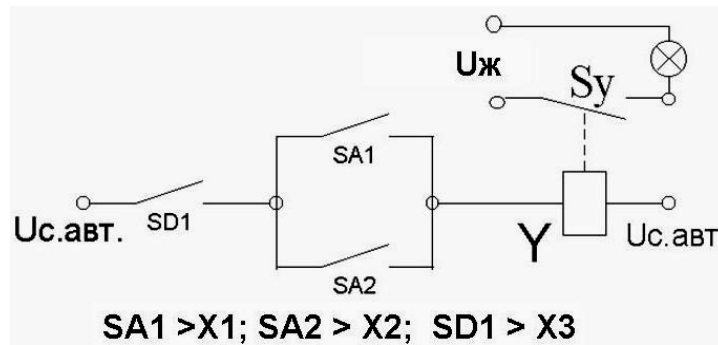


Рисунок 14.3 – Контактно-релейна схема пристрою

Використовуючи двійкову систему числення для позначення всіх варіантів, приймаючи умову «замкнуті» контакти = 1, «розімкнені» = 0 складемо таблицю істинності. Кількість варіантів визначимо  $N = 2^n$ , де  $n$  – число аргументів.

Таблиця 14.3 – Таблиця істинності

№ з/п	X3	X2	X2	Y
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

*Визначити, в якій формі вигідніше записати аналітичне рівняння.*

Необхідно враховувати, що:  $X \cdot X$  – кон'юнкція,  $X \vee X$  – диз'юнкція,  $\bar{X}$  – інверсія.

Виділимо в таблиці 14.3 значення Y, рівні 1. Якщо аргумент рівний 1, то він запишеться в рівнянні без інверсії –  $X$ , інакше – з інверсією  $\bar{X}$ . Маємо логічне рівнянь в аналітичній формі:

$$Y = X_1 \cdot \bar{X}_2 \cdot X_3 \vee \bar{X}_1 \cdot X_2 \cdot X_3 \vee X_1 \cdot X_2 \cdot X_3 . \quad (14.1)$$

*Провести мінімізацію логічного рівняння.*

Скористаємося правилом склеювання: виберемо для склеювання групу аргументів –  $X_1 X_2 X_3$ .

А)  $X_1 \cdot \bar{X}_2 \cdot X_3 \vee X_1 \cdot X_2 \cdot X_3 = X_1 \cdot X_3$  – виключений аргумент з інверсією.

Б)  $\bar{X}_1 \cdot X_2 \cdot X_3 \vee X_1 \cdot X_2 \cdot X_3 = X_2 \cdot X_3$  – виключені  $X_1$  і  $\bar{X}_1$ .

У результаті наше початкове логічне рівняння після перетворень набуде вигляду:

$$Y = X_1 X_3 \vee X_2 X_3. \quad (14.2)$$

Винесемо аргумент  $X_3$  за дужки і запишемо рівняння:

$$Y = X_3 (X_1 \vee X_2), \quad (14.3)$$

в якому маємо мінімум аргументів за значенням.

Користуючись таблицею 14.2 і варіантами еквівалентних схем (табл. 14.3), складемо функціональну схему пристрою (рис. 14.4). Реалізуючи окремі функції перевірте роботу пристрою.

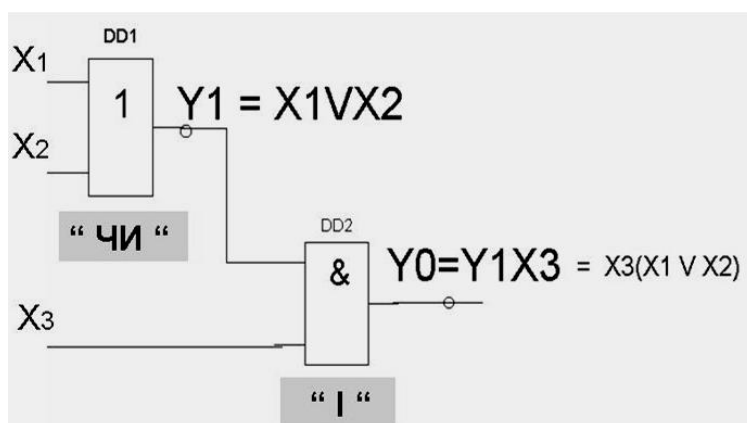


Рисунок 14.4 – Функціональна схема пристрою на логічних елементах

**Завдання.** Для закріплення теми необхідно вирішити аналогічне завдання згідно з індивідуальним варіантом.

### 14.3 Кодування дискретної інформації

**Мета:** засвоїти принципи бінарного кодування інформації в цифрових пристроях автоматики.

Використовуючи принцип адресного керування виконавчими пристроями за допомогою дешифратора, скласти схему керування чотирма ВЕ за двома лініями зв'язку (рис. 14.5).

Застосовуючи будь-які додаткові елементи логіки, самостійно допрацювати схему автоматичного пристрою, який забезпечить реалізацію таблиці істинності (табл. 14.4).

Таблиця 14.4 – Таблиця істинності керуючого пристрою

Адреса (код)	Аргументи		Функції			
	X1	X2	Y1	Y2	Y3	Y4
0000	1	1	1	0	0	0
0001	1	0	0	1	0	0
0010	0	1	0	0	1	0
0011	0	0	0	0	0	1

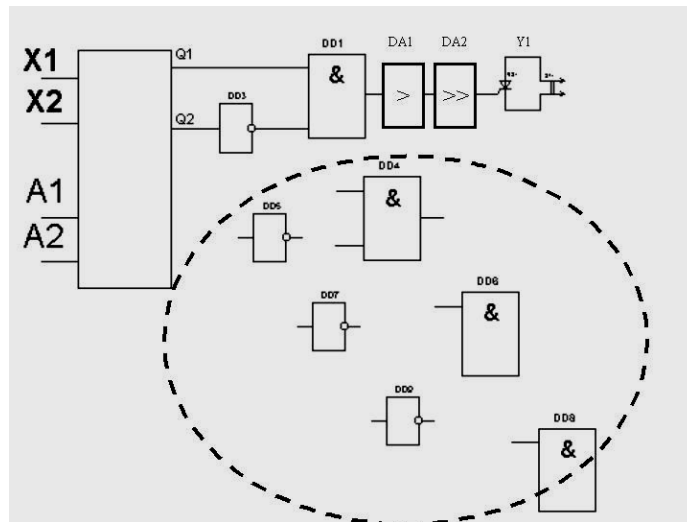


Рисунок 14.5 - Проект пристрою для включення чотирьох ВЕ

При цьому слід врахувати, що при:  $Y1 = 1$  включається світлова сигналізація;  $Y2 = 1$  – електродвигун;  $Y3 = 1$  – проміжне реле;  $Y4 = 1$  – переривчастий звуковий сигнал.

Розгляньте варіанти, коли вхідний сигнал поступає за 3, 4,  $n$  інформаційними лініями до цифрового пристрою автоматики.

**Завдання.** Використовуючи варіант індивідуального завдання, доповнити розроблений логічний пристрій автоматики: 1 – асинхронним і 2 – синхронним елементами пам'яті.

#### 14.4 Логічні й арифметичні дії в МП

**Мета:** засвоїти принципи реалізації логічних і арифметичних дій в мікропроцесорах, їх кодування і використання в мовах програмування.

МП КР580 (рис. 14.6) виконує логічні операції, які реалізують відомі електронні логічні елементи: AND, OR, NOT, а також що "виключає OR" і комбінації NAND, XOR.

1. Виконання логічних операцій.

$A \wedge B$  «І - AND» – команда ADD

1	0	1	1	0	0	1	1
0	1	1	1	0	0	1	0

перше число розміщується в регістрі акумулятора (A), а друге – в регістрі швидкої пам'яті (B). Результат розміщується в (A):

0	0	1	1	0	0	1	0
---	---	---	---	---	---	---	---

$A \vee B$  «АБО - OR» – команда ORA

1	0	1	1	0	0	1	1
1	1	1	1	0	0	1	1

- перше число розміщується в (A), а друге – в (B).

Результат розміщується в (A):

1	1	1	1	0	0	1	1
---	---	---	---	---	---	---	---

**«Вимикальне АБО – XOR» – операція доповнення до парності:**

1	0	1	1	0	0	1	1
0	1	1	1	0	0	1	0
1	1	0	0	0	0	0	1

розміщення чисел за схемою  $A > B > A$ .

**Заперечення або інверсія «НІ - NOT»:**

1	0	1	1	0	0	1	1
0	1	0	0	1	1	0	0

розміщення чисел за схемою  $A > A$ .

**Команди «Зрушення» вмісту акумулятора вліво: RLC**

0	1	1	0	1	0	1	1
1	1	0	1	0	1	1	0

Команда RRC - реалізується аналогічно, але із зрушенням управо.

**2. Виконання арифметичних операцій.**

Всі арифметичні операції виконуються на основі бінарної арифметики.

При програмуванні початкові числа представляються в 16-річній системі.

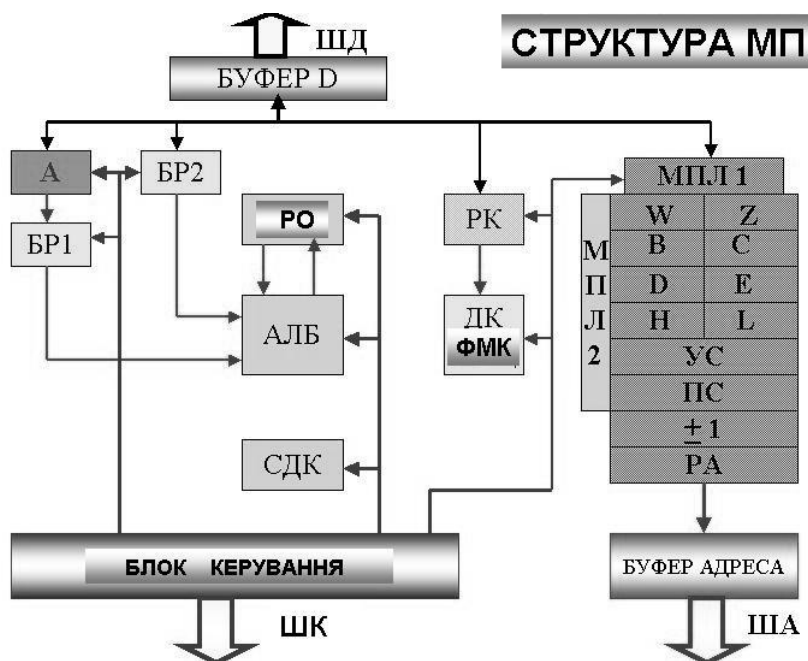


Рисунок 14.6 – Структура МП КР580:

*A – реєстр акумулятора; АЛБ – арифметично-логічний блок; РО – реєстр ознак; РК - реєстр команд; ДКФМЦ – дешифратор команд і формувач машинних кодів; СДК – схема десяткової корекції; МПЛ1, МПЛ2 – мультиплектори; В, С, D, E, H, L, W, Z – блок реєстрів загального призначення; ВУС – показчик стека; ПС – програмний лічильник; РА – реєстр адреси; БР1, БР2 – буферні реєстри*

У РО МП при виконанні арифметичних операцій біт С розглядається як 9-й розряд А – акумулятора.

S	Z	-	AC	-	P	-	C
---	---	---	----	---	---	---	---

**Приклад.** Виконати складання чисел. При складанні  $B8 + A2$ :  $8 + 2 = 10 > A$ ;  $B > 11$ ,  $A > 10$ ;  $11 + 10 = 21$ ;  $21 - 16 = 5 + 1$ , тут маємо «1», що відображає переповнювання акумулятора, яка переміститься в біт С регістра ознак.

У двійкових кодах: команди операцій «суми» – ACI, ADC, і «віднімання» – SBI, SBB та ін. виконуються:

« 1 »				« A »			
0	0	0	1	1	0	1	0
« 4 »				« 9 »			
0	1	0	0	1	0	0	1

Суму цих чисел запишемо для акумулятора:

0      1      0      1      2      0      1      1

Пояснити дії МП з величиною «2».

**Завдання.** Виконати арифметичні й логічні операції з числовою інформацією відповідно до індивідуального завдання.

## 14.5 Розробка блок-схем алгоритмів

**Мета:** навчитися створювати блок-схеми алгоритмів для вирішення різних завдань автоматизації технологічних об'єктів за допомогою програмованих МПП.

Різноманіття блоків часто зводиться до декількох типових: блоки початку і кінця алгоритму; блоки обробки даних; блоки перевірки умов.

Всі блоки відрізняються між собою і графічно зображуються: овальною, прямокутною формами і у вигляді ромба.

**Завдання.** Розробити блок-схему алгоритму вирішення задачі керування ВЕ, розглянутої в розділі 14.2, відповідно індивідуального завдання за допомогою мікропроцесорного пристрою.

## 14.6 Програмування пристроїв автоматики

**Мета:** освоїти основні етапи синтезу пристроїв автоматики на базі мікропроцесорів, що включають розробку блок-схеми алгоритму й написання програми.

**Приклад.** Класична схема комплекту пристрою автоматики складається з (рис. 14.7): ПЕ, перетворювача контрольованої величини (далі – ПКВ) в нормований сигнал, пристрою обробки початкової інформації МП, формувача керуючого сигналу (далі – ФКС) і ВЕ.



Рисунок 14.7 – Схема комплекту пристрою автоматизи

1. Концептуальна схема алгоритму має такий вигляд:

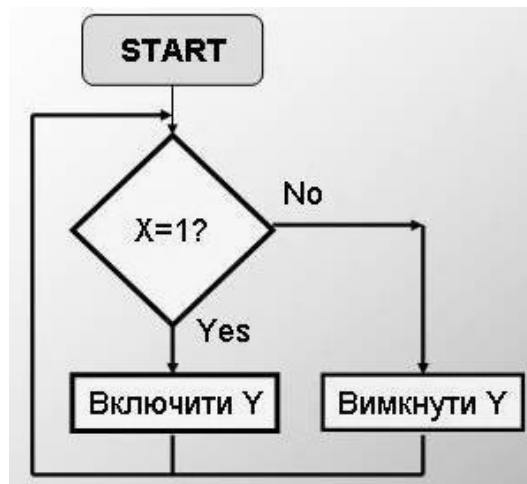


Рисунок 14.8 – Концептуальна схема алгоритму

Припустимо (умова задається): формати вхідного сигналу (слова) – 0 0 0 0 0 1 0 0 (портал введення), а вихідного – 0 0 1 0 0 0 0 0 (портал виводу).  
 Задаємо: портал введення – «27D» > 1ВН ( $27 - 16 = 16 + 11 = 1$  ВН) і портал введення – «42D» > 2АН ( $42 - 16 - 16 = 32 + 10 = 2$  АН).

2. Підготуємо функціональну схему алгоритму:



Рисунок 14.9 – Функціональна схема алгоритму

3. Підготуємо листинг програми на асемблері в прийнятому форматі таблиці.

Таблиця 14.5 – Листинг програми «START»

Адреса	Код	Мітка	Мнемокод	Коментар
8020 8021	DB 1B	STAR T:	IN27	Ввод – «27D» #27D ► 1BН
8022 8023	06 04		MVI B,04	Записати в регістр В число 04 (при X = 1, 04 – 04 = 0; при X = 0, 00 – 04 = –04)
8024	90		SUB B	Відняти з А вміст В і записати в А
8025 8026 8027	CA 2F 80		JZ 802F	Якщо результат 0, то перейти за адресою 802F (задається ПІЗНІШЕ!)
8028 8029	3E 00		MVI A,00	Вимкнути ВЕ. #42
802A 802B	D3 2A		OUT 42D	42D ► 2АН
802C 802D 802E	C3 20 80		JMP 8020	Перехід на адресу 8020
802F 8030	3E 20		MVI A,32	Включити ВЕ (A < 32D = 20H)
8031 8032	D3 2A		OUT 42	Порт виводу (42D = 2A)
8033 8034 8035	C3 20 80		JMP 8020	Перехід на адресу 8020

Для підготовки тексту програми слід скористатися документацією для МП КР580 (всі МП мають таку довідкову інформацію), в якій є перелік всіх команд. Коментарії в листингу пояснюють операції, що плануються для реалізації керування ВЕ за допомогою МПП.

**Завдання.** Набрати текст програми, виконати компіляцію. виправити виявлені помилки. Внести в програму зміни, ілюструючи своє індивідуальне вирішення відомої задачі. Проглянути код програми в пам'яті МПП за допомогою спеціальної програми для редагування текстів.

Для вивчення готових програмних продуктів користуються кодами програм, які легко проглянути за допомогою програм-переглядачів пам'яті МПП. Фрагмент програми у двійкових машинних кодах (об'єктна програма) ілюструє таблиця 14.6.

Таблиця 14.6 – Фрагмент програми у машинних кодах

ADDRESS	CODE															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
<b>01F0 &gt;&gt;</b>	<b>00</b>	<b>01</b>	<b>02</b>	<b>03</b>	<b>04</b>	<b>05</b>	<b>06</b>	<b>07</b>	<b>08</b>	<b>09</b>	<b>0A</b>	<b>0B</b>	<b>0C</b>	<b>0D</b>	<b>0E</b>	<b>0F &lt;&lt; 01FF</b>
<b>0200</b>	CD	B9	F8	CB	5D	B4	2F	32	33	04	21	C3	7E	22	3D	B4
<b>0210</b>	CD	18	F8	C3	1C	01	21	D0	01	CD	18	F8	21	0D	B1	CD
<b>0220</b>	18	F8	21	43	B2	C0	18	F8	CD	03	F8	FE	31	CA	49	00
<b>0230</b>	1B	39	2C	34	20	2B	77	79	62	63	72	69	74	65	20	69

Продовження таблиці 14.6.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0240	67	6F	72	6F	73	74	78	3A	20	20	2E	11	39	21	39	38
0250	31	20	70	20	4F	2B	62	20	63	2B	64	2B	69	21	74	2B
0260	32	20	20	6B	6F	6E	65	63	2B	2D	2B	22	33	22	2B	2B
0270	00	11	59	20	22	00	1B	59	20	5E	01	05	21	6E	30	0A
0280	CD	18	F8	CD	F7	B2	4C	CD	09	F8	40	CD	09	F8	CD	21
0290	F8	C2	1D	02	C9	A7	CA	D0	02	11	00	00	36	00	19	00
02A0	7E	FE	2A	C8	FE	21	C8	FE	38	C8	18	C9	0A	A7	CA	F3
02B0	02	3D	02	1E	2A	73	C9	2A	8D	02	0E	10	7C	29	E6	60
02C0	EA	04	03	23	00	C2	FC	02	7C	PE	20	FA	FC	02	22	80
02D0	02	C9	A7	01	AO	C2	1A	03	01	A1	E6	0F	11	01	00	FE
02E0	81	C8	11	40	00	FE	06	C8	13	FE	07	C8	13	FE	03	C8
02F0	11	11	FF	FE	0C	C8	13	FE	00	C8	13	FE	09	C8	11	FF
0300	FP	FE	0C	C8	13	C9	3E	3A	32	30	7E	32	7E	76	C9	C3
0310	0E	07	C0	09	F8	C0	09	F8	C1	C9	63	2A	37	04	21	70
0320	14	C2	5E	03	E1	C9	3A	8F	02	02	C0	4F	03	C9	AF	C0
0330	12	03	2A	3D	04	3A	39	04	06	76	CD	06	02	C2	83	03
0340	2A	3D	04	22	3D	04	01	39	04	F5	CC	66	03	0E	3A	3A
0350	33	04	A7	CA	98	03	0E	7F	F1	19	01	31	04	CC	98	02
0360	01	39	04	1E	24	CD	EC	02	3A	31	04	E3	CD	1E	F8	E3
0370	21	81	02	C0	18	F8	3A	31	04	CD	13	F8	E1	03	11	18
0380	1D	19	D1	22	42	02	21	40	02	CD	18	F8	E1	CD	3A	03
0390	3A	51	04	4F	C9	3E	01	C0	12	03	2A	5F	04	3A	3A	04
03A0	06	24	C0	06	02	C2	E1	03	2A	3F	04	22	3F	04	01	3A
03B0	04	F3	CC	66	03	0E	3A	3A	33	04	A7	CA	00	04	0E	7F
03C0	F1	19	01	3C	04	CC	98	02	01	3A	04	1E	76	CD	EC	02
03D0	3A	3C	04	ES	CD	1E	F8	E3	21	86	02	C0	18	F8	3A	3C
03E0	04	CD	13	F8	E1	D3	11	18	1D	19	01	22	42	02	21	40
03F0	02	CD	18	F8	E1	CD	5A	03	3A	3C	04	4F	C9	3E	91	D3
0400	A3	AF	32	39	04	32	3A	04	32	31	04	32	3C	04	C9	0E
0410	3A	8F	02	C0	4F	03	C9	AF	C0	03	2A	3D	04	3A	39	04
0420	23	21	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF

**Завдання.** Відновити лістинг фрагмента програми відповідно індивідуальному завданню. Оформити лістинг, використовуючи приклад (табл. 14.7) фрагменту з адреси XXXX до XXXX.

Таблиця 14.7 – Фрагмент відновленого лістингу програми (1103–120B)

Адреса	Код	Мнемокод	Коментар
1103	4F	MOV C, A;	переслати з А в С
1104	D630	SUI 30H;	
1106	FE0D	CPI 0DH;	
1108	D20611	JNC 1106;	
1111	CD09FB	CALL FB09;	
1114	80	ADD B;	
1115	47	MOV B, A;	
1116			

Для написання програм і різних експериментів з готовими програмами доцільно використовувати редактори й переглядачі програм, що входять в комплект Windows-додатку «SinSys» і вище або інших вільно поширюваних навчальних програм, але при цьому слід враховувати їх призначення тільки для конкретних мікропроцесорів.



## СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Єсаулов С. М. Мікропроцесорні пристрої електротранспорту, мікропроцесорні пристрої транспортних засобів, мікропроцесорні пристрої систем автоматизації електроприводів : конспект лекцій із завданнями до практичних робіт : для студентів 4–5 курсів усіх форм навчання за спеціальністю 141 – Електроенергетика, електротехніка та електромеханіка / С. М. Єсаулов, О. Ф. Бабічева ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 135 с.
2. Мікропроцесорні пристрої електротранспорту, мікропроцесорні пристрої транспортних засобів, мікропроцесорні пристрої систем автоматизації електроприводів : метод. вказівки до самост. вивч. : для студентів 4–5 курсів усіх форм навчання за спеціальністю 141 – Електроенергетика, електротехніка та електромеханіка / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; [уклад. С. М. Єсаулов, О. Ф. Бабічева]. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 53 с.
3. Мікропроцесорні пристрої електротранспорту, мікропроцесорні пристрої транспортних засобів, мікропроцесорні пристрої систем автоматизації електроприводів : метод. вказівки для викон. лаб. робіт : для студентів 4–5 курсів усіх форм навчання за спеціальністю 141 – Електроенергетика, електротехніка та електромеханіка / Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова ; [уклад. С. М. Єсаулов, О. Ф. Бабічева. – Харків : ХНУМГ ім. О. М. Бекетова, 2017. – 82 с.

## ДОДАТОК А

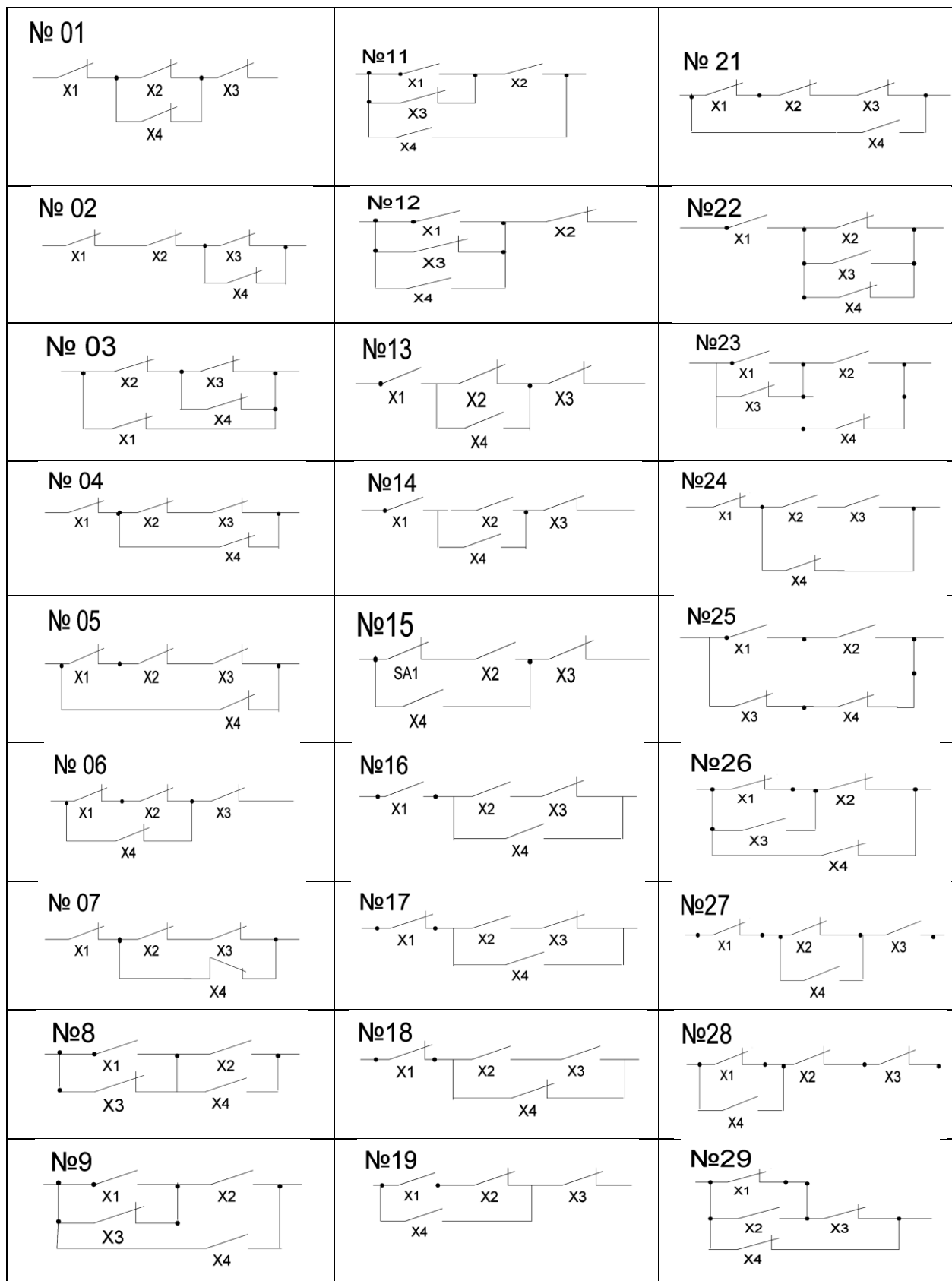
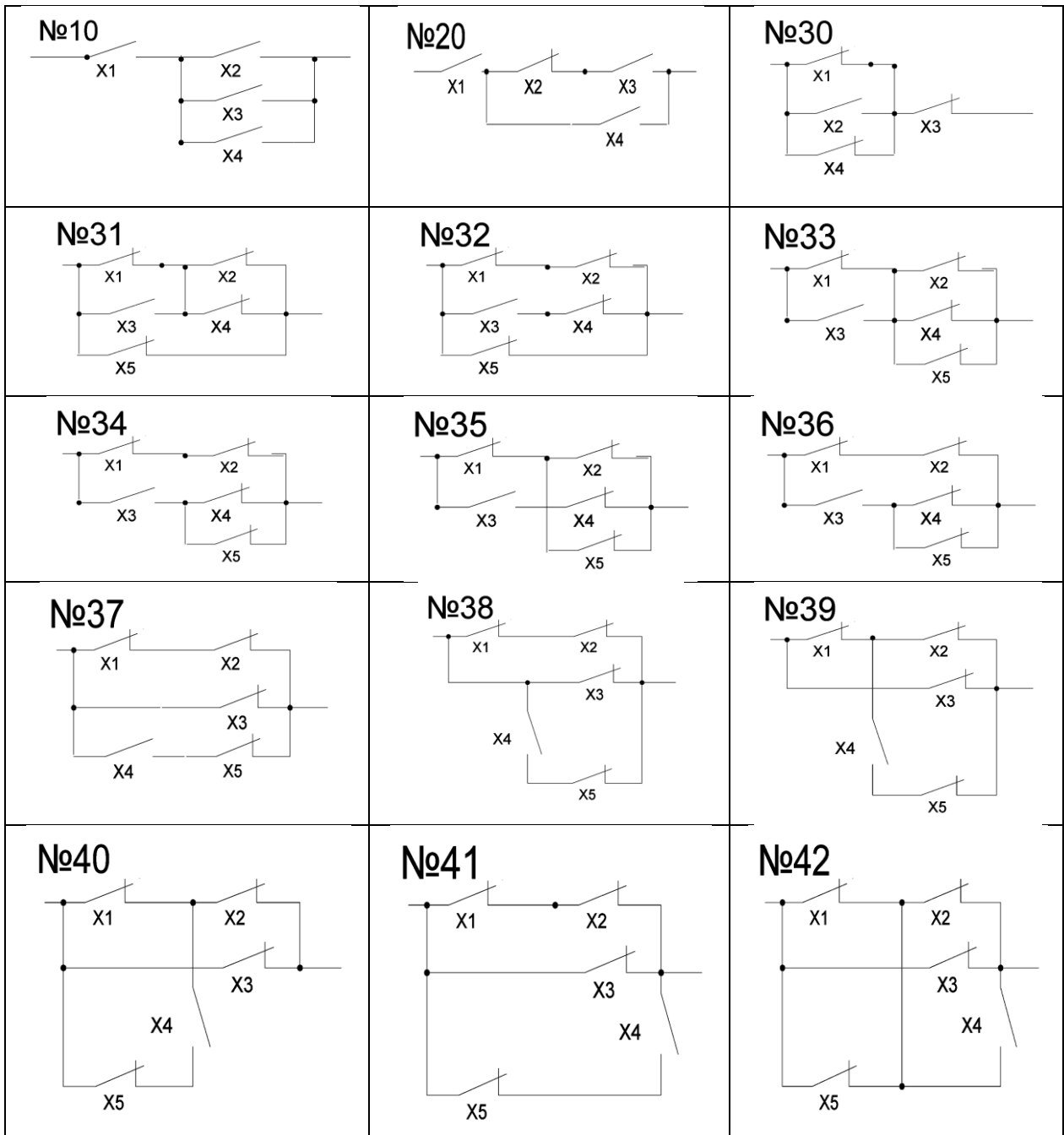


Рисунок А.1 – Варіанти завдань



Продовження рисунка А.1

Таблиця А.1 – Система команд мікропроцесора

Мнемокод	Код операції	Кількість		Зміст команди
		циклів	тактів	
1	2	3	4	5
				Пересилання:
MOV A,A	7F	1	4	(A)←(A)
MOV A,B	78	1	4	(A)←(B)
MOV A,C	79	1	4	(A)←(C)
MOV A,D	7A	1	4	(A)←(D)
MOV A,E	7B	1	4	(A)←(E)
MOV A,H	7C	1	4	(A)←(H)
MOV A,L	7D	1	4	(A)←(L)
MOV A,M	7E	2	7	(A)←(M(H,L))
MOV B,A	47	1	4	(B)←(A)
MOV B,B	40	1	4	(B)←(B)
MOV B,C	41	1	4	(B)←(C)
MOV B,D	42	1	4	(B)←(D)
MOV B,E	43	1	4	(B)←(E)
MOV B,H	44	1	4	(B)←(H)
MOV B,L	45	1	4	(B)←(L)
MOV B,M	46	2	7	(B)←(M(H,L))
MOV C,A	4F	1	4	(C)←(A)
MOV C,B	48	1	4	(C)←(B)
MOV C,C	49	1	4	(C)←(C)
MOV C,D	4A	1	4	(C)←(D)
MOV C,E	4B	1	4	(C)←(E)
MOV C,H	4C	1	4	(C)←(H)
MOV C,L	4D	1	4	(C)←(L)
MOV C,M	4E	2	7	(C)←(M(H,L))
MOV D,A	57	1	4	(D)←(A)
MOV D,B	50	1	4	(D)←(B)
MOV D,C	51	1	4	(D)←(C)
MOV D,D	52	1	4	(D)←(D)
MOV D,E	53	1	4	(D)←(E)
MOV D,H	54	1	4	(D)←(H)
MOV D,L	55	1	4	(D)←(L)
MOV D,M	56	2	7	(D)←(M(H,L))
MOV E,A	5F	1	4	(E)←(A)
MOV E,B	58	1	4	(E)←(B)
MOV E,C	59	1	4	(E)←(C)
MOV E,D	5A	1	4	(E)←(D)
MOV E,E	5B	1	4	(E)←(E)

Продовження таблиці А.1

1	2	3	4	5
MOV E,H	5C	1	4	(E)←(H)
MOV E,L	5D	1	4	(E)←(L)
MOV E,M	5E	2	7	(E)←(M(H,L))
MOV H,A	67	1	4	(H)←(A)
MOV H,B	60	1	4	(H)←(B)
MOV H,C	61	1	4	(H)←(C)
MOV H,D	62	1	4	(H)←(D)
MOV H,E	63	1	4	(H)←(E)
MOV H,H	64	1	4	(H)←(H)
MOV H,L	65	1	4	(H)←(L)
MOV H,M	66	2	7	(H)←(M(H,L))
MOV L,A	6F	1	4	(L)←(A)
MOV L,B	68	1	4	(L)←(B)
MOV L,C	69	1	4	(L)←(C)
MOV L,D	6A	1	4	(L)←(D)
MOV L,E	6B	1	4	(L)←(E)
MOV L,H	6C	1	4	(L)←(H)
MOV L,L	6D	1	4	(L)←(L)
MOV L,M	6E	2	7	(L)←(M(H,L))
MOV M,A	77	2	7	(M(H,L))←(A)
MOV M,B	70	2	7	(M(H,L))←(B)
MOV M,C	71	2	7	(M(H,L))←(C)
MOV M,D	72	2	7	(M(H,L))←(D)
MOV M,E	73	2	7	(M(H,L))←(E)
MOV M,H	74	2	7	(M(H,L))←(H)
MOV M,L	75	2	7	(M(H,L))←(L)
MVI A, B <sub>2</sub> (data)	3E	2	7	(A)←B <sub>2</sub>
MVI B, B <sub>2</sub> (data)	06	2	7	(B)←B <sub>2</sub>
MVI C, B <sub>2</sub> (data)	0E	2	7	(C)←B <sub>2</sub>
MVI D, B <sub>2</sub> (data)	16	2	7	(D)←B <sub>2</sub>
MVI E, B <sub>2</sub> (data)	1E	2	7	(E)←B <sub>2</sub>
MVI H, B <sub>2</sub> (data)	26	2	7	(H)←B <sub>2</sub>
MVI L, B <sub>2</sub> (data)	2E	2	7	(L)←B <sub>2</sub>
MVI M, B <sub>2</sub> (data)	36	3	10	(M(H,L))←B <sub>2</sub>
LXI B, B <sub>2</sub> , B <sub>3</sub> (data)	01	3	10	(B, C)←B <sub>3</sub> B <sub>2</sub>
LXI D, B <sub>2</sub> , B <sub>3</sub> (data)	11	3	10	(D, C)←B <sub>3</sub> B <sub>2</sub>
LXI H, B <sub>2</sub> , B <sub>3</sub> (data)	21	3	10	(H, C)←B <sub>3</sub> B <sub>2</sub>
LXI SP, B <sub>2</sub> , B <sub>3</sub> (data)	31	3	10	(SP)←B <sub>3</sub> B <sub>2</sub>
LDAX B	0A	2	7	(A)←(M(B,C))
LDAX D	1A	2	7	(A)←(M(D,E))
LHLD B <sub>2</sub> B <sub>3</sub> (addr)	2A	5	16	(H, L)←(M(B <sub>3</sub> B <sub>2</sub> ))

Продовження таблиці А.1

1	2	3	4	5
LDA B <sub>2</sub> B <sub>3</sub> (addr)	3A	4	13	(A)←(M(B <sub>3</sub> B <sub>2</sub> ))
STA B <sub>2</sub> B <sub>3</sub> (addr)	32	3	13	M(B <sub>3</sub> B <sub>2</sub> )←(A)
STAX B	02	1	7	M(B, C)←(A)
STAX D	12	1	7	M(D, E)←(A)
				<b>Арифметичне складання:</b>
ADD A	87	1	4	(A)←(A)+(A)
ADD B	80	1	4	(A)←(A)+(B)
ADD C	81	1	4	(A)←(A)+(C)
ADD D	82	1	4	(A)←(A)+(D)
ADD E	83	1	4	(A)←(A)+(E)
ADD H	84	1	4	(A)←(A)+(H)
ADD L	85	1	4	(A)←(A)+(L)
ADD M	86	2	7	(A)←(A)+(M(H, L))
ADC A	8F	1	4	(A)←(A)+(A)+C
ADC B	88	1	4	(A)←(A)+(B)+C
ADC C	89	1	4	(A)←(A)+(C)+C
ADC D	8A	1	4	(A)←(A)+(D)+C
ADC E	8B	1	4	(A)←(A)+(E)+C
ADC H	8C	1	4	(A)←(A)+(H)+C
ADC L	8D	1	4	(A)←(A)+(L)+C
ADC M	8E	1	4	(A)←(A)+(M(H, L))+C
				<b>Арифметичне віднімання:</b>
SUB A	97	1	4	(A)←(A)-(A)
SUB B	90	1	4	(A)←(A)-(B)
SUB C	91	1	4	(A)←(A)-(C)
SUB D	92	1	4	(A)←(A)-(D)
SUB E	93	1	4	(A)←(A)-(E)
SUB H	94	1	4	(A)←(A)-(H)
SUB L	95	1	4	(A)←(A)-(L)
SUB M	96	2	7	(A)←(A)-(M(H, L))
SBB A	9F	1	4	(A)←(A)-(A)-C
SBB B	98	1	4	(A)←(A)-(B)-C
SBB C	99	1	4	(A)←(A)-(C)-C
SBB D	9A	1	4	(A)←(A)-(D)-C
SBB E	9B	1	4	(A)←(A)-(E)-C
SBB H	9C	1	4	(A)←(A)-(H)-C
SBB L	9D	1	4	(A)←(A)-(L)-C
SBB M	9E	2	7	(A)←(A)-(M(H, L))-C
				<b>Подвійне складання:</b>
DAD B	09	3	10	(H, L)←(H, L)+(B, C)
DAD D	19	3	10	(H, L)←(H, L)+(D, E)

Продовження таблиці А.1

1	2	3	4	5
DAD H	29	3	10	$(H, L) \leftarrow (H, L) + (H, L)$
DAD SP	39	3	10	$(H, L) \leftarrow (H, L) + (SP)$
				<b>Збільшення на одиницю:</b>
INR A	3C	1	4	$(A) \leftarrow (A) + 1$
INR B	04	1	4	$(B) \leftarrow (B) + 1$
INR C	0C	1	4	$(C) \leftarrow (C) + 1$
INR D	14	1	4	$(D) \leftarrow (D) + 1$
INR E	1C	1	4	$(E) \leftarrow (E) + 1$
INR H	24	1	4	$(H) \leftarrow (H) + 1$
INR L	2C	1	4	$(L) \leftarrow (L) + 1$
INR M	34	3	10	$(M(H, L)) \leftarrow (M(H, L)) + 1$
INX B	03	1	6	$(B, C) \leftarrow (B, C) + 1$
INX D	13	1	6	$(D, E) \leftarrow (D, E) + 1$
INX H	23	1	6	$(H, L) \leftarrow (H, L) + 1$
INX SP	33	1	6	$(SP) \leftarrow (SP) + 1$
				<b>Зменшення на одиницю:</b>
DCR A	3D	1	4	$(A) \leftarrow (A) - 1$
DCR B	05	1	4	$(B) \leftarrow (B) - 1$
DCR C	0D	1	4	$(C) \leftarrow (C) - 1$
DCR D	15	1	4	$(D) \leftarrow (D) - 1$
DCR E	1D	1	4	$(E) \leftarrow (E) - 1$
DCR H	25	1	4	$(H) \leftarrow (H) - 1$
DCR L	2D	1	4	$(L) \leftarrow (L) - 1$
DCR M	35	3	10	$(M(H, L)) \leftarrow (M(H, L)) - 1$
DCX B	0B	1	6	$(B, C) \leftarrow (B, C) - 1$
DCX D	1B	1	6	$(D, E) \leftarrow (D, E) - 1$
DCX H	2B	1	6	$(H, L) \leftarrow (H, L) - 1$
DCX SP	3B	1	6	$(SP) \leftarrow (SP) - 1$
DAA	27	1	4	Десяткова корекція акумулятора
				<b>Інвертування акумулятора:</b>
CMA	2F	1	4	$(\bar{A}) \leftarrow (A)$
				<b>Встановити біт переносу:</b>
STC	37	1	4	$C \leftarrow 1$
				<b>Інвертування переносу:</b>
CMC	3F	1	4	$\bar{C} \leftarrow C$
				<b>Зсув:</b>
RLC	07	1	4	вліво з переносом
RRC	17	1	4	вправо з переносом
RAL	17	1	4	циклічний улів

Продовження таблиці А.1

1	2	3	4	5
RAR	1F	1	4	циклічний управо
				<b>Логічне множення:</b>
ANA A	A7	1	4	$(A) \leftarrow (A) \wedge (A)$
ANA B	A0	1	4	$(A) \leftarrow (A) \wedge (B)$
ANA C	A1	1	4	$(A) \leftarrow (A) \wedge (C)$
ANA D	A2	1	4	$(A) \leftarrow (A) \wedge (D)$
ANA E	A3	1	4	$(A) \leftarrow (A) \wedge (E)$
ANA H	A4	1	4	$(A) \leftarrow (A) \wedge (H)$
ANA L	A5	1	4	$(A) \leftarrow (A) \wedge (L)$
ANA M	A6	2	7	$(A) \leftarrow (A) \wedge (M(H, L))$
				<b>Виключно АБО:</b>
XRA A	AF	1	4	$(A) \leftarrow (A) \vee (A)$
XRA B	A8	1	4	$(A) \leftarrow (A) \vee (B)$
XRA C	A9	1	4	$(A) \leftarrow (A) \vee (C)$
XRA D	AA	1	4	$(A) \leftarrow (A) \vee (D)$
XRA E	AB	1	4	$(A) \leftarrow (A) \vee (E)$
XRA H	AC	1	4	$(A) \leftarrow (A) \vee (H)$
XRA L	AD	1	4	$(A) \leftarrow (A) \vee (L)$
XRA M	AE	2	7	$(A) \leftarrow (A) \vee (M(H, L))$
				<b>Логічне АБО:</b>
ORA A	B7	1	4	$(A) \leftarrow (A) \vee (A)$
ORA B	B0	1	4	$(A) \leftarrow (A) \vee (B)$
ORA C	B1	1	4	$(A) \leftarrow (A) \vee (C)$
ORA D	B2	1	4	$(A) \leftarrow (A) \vee (D)$
ORA E	B3	1	4	$(A) \leftarrow (A) \vee (E)$
ORA H	B4	1	4	$(A) \leftarrow (A) \vee (H)$
ORA L	B5	1	4	$(A) \leftarrow (A) \vee (L)$
ORA M	B6	2	7	$(A) \leftarrow (A) \vee (M(H, L))$
				<b>Порівняти:</b>
CMP A	BF	1	4	$(A) - (A)$
CMP B	B8	1	4	$(A) - (B)$
CMP C	B9	1	4	$(A) - (C)$
CMP D	BA	1	4	$(A) - (D)$
CMP E	BB	1	4	$(A) - (E)$
CMP H	BC	1	4	$(A) - (H)$
CMP L	BD	1	4	$(A) - (L)$
CMP M	BE	1	4	$(A) - (M(H, L))$
				<b>Безпосередні дії:</b>
ADI B <sub>2</sub> (data)	C6	2	7	$(A) \leftarrow (A) + B2$
ACI B <sub>2</sub> (data)	CE	2	7	$(A) \leftarrow (A) + B2 + C$
SUI B <sub>2</sub> (data)	D6	2	7	$(A) \leftarrow (A) - B2$



Продовження таблиці А.1

1	2	3	4	5
SBI B <sub>2</sub> (data)	DE	2	7	(A)←(A)–B2–C
ANI B <sub>2</sub> (data)	E6	2	7	(A)←(A)∧B2
XRI B <sub>2</sub> (data)	EE	2	7	(A)←(A)∨B2
ORI B <sub>2</sub> (data)	F6	2	7	(A)←(A)∨B2
CPI B <sub>2</sub> (data)	FE	2	7	(A)–B2
JMP B <sub>2</sub> B <sub>3</sub> (addr)	C3	3	10	Перехід за адресою B <sub>3</sub> B <sub>2</sub>
				<b>Перехід якщо:</b>
JNZ B <sub>2</sub> B <sub>3</sub> (addr)	C2	2/3	7/10	не нуль (Z=0)
JZ B <sub>2</sub> B <sub>3</sub> (addr)	CA	2/3	7/10	нуль (Z=1)
JNC B <sub>2</sub> B <sub>3</sub> (addr)	D2	2/3	7/10	немає переносу (C=0)
JC B <sub>2</sub> B <sub>3</sub> (addr)	DA	2/3	7/10	є перенос (C=1)
JPO B <sub>2</sub> B <sub>3</sub> (addr)	E2	2/3	7/10	непарність (P=0)
JPE B <sub>2</sub> B <sub>3</sub> (addr)	EA	2/3	7/10	парність (P=1)
JP B <sub>2</sub> B <sub>3</sub> (addr)	F2	2/3	7/10	додатність (S=0)
JM B <sub>2</sub> B <sub>3</sub> (addr)	FA	2/3	7/10	від'ємність (S=1)
PCHL	E9	1	6	(PC)←(H,L)
CALL B <sub>2</sub> B <sub>3</sub> (addr)	CD	5	17	Виклик підпрограми
				<b>Виклик підпрограми якщо:</b>
CNZ B <sub>2</sub> B <sub>3</sub> (addr)	C4	2/5	9/18	не нуль (Z=0)
CZ B <sub>2</sub> B <sub>3</sub> (addr)	CC	2/5	9/18	нуль (Z=1)
CNC B <sub>2</sub> B <sub>3</sub> (addr)	D4	2/5	9/18	немає переносу (C=0)
CC B <sub>2</sub> B <sub>3</sub> (addr)	DC	2/5	9/18	є перенос (C=1)
CPO B <sub>2</sub> B <sub>3</sub> (addr)	E4	2/5	9/18	непарність (P=0)
CPE B <sub>2</sub> B <sub>3</sub> (addr)	EC	2/5	9/18	парність (P=1)
CP B <sub>2</sub> B <sub>3</sub> (addr)	F4	2/5	9/18	додатність (S=0)
CM B <sub>2</sub> B <sub>3</sub> (addr)	FC	2/5	9/18	від'ємність (S=1)
RET	C9	3	10	Повернення з підпрограми
				<b>Повернення з підпрограми якщо:</b>
RNZ	C0	1/3	6/12	не нуль (Z=0)
RZ	C8	1/3	6/12	нуль (Z=1)
RNC	D0	1/3	6/12	немає переносу (C=0)
RC	D8	1/3	6/12	є перенос (C=1)
RPO	E0	1/3	6/12	непарність (P=0)
RPE	E8	1/3	6/12	парність (P=1)
RP	F0	1/3	6/12	додатність (S=0)
RM	F8	1/3	6/12	від'ємність (S=1)
RST 0	C7	3	12	Переривання
RST 1	CA	3	12	-//-
RST 2	B7	3	12	-//-

Закінчення таблиці А.1

1	2	3	4	5
RST 3	BA	3	12	-//-
RST 4	E7	3	12	-//-
RST 5	EF	3	12	-//-
RST 6	F7	3	12	-//-
RST 7	FF	3	12	-//-
				<b>Операції із стеком:</b>
PUSH B	C5	3	11	помістити до стеку
PUSH D	D5	3	11	-//-
PUSH H	E5	3	11	-//-
PUSH PSW	F5	3	11	-//-
POP B	C1	3	10	добути із стеку
POP D	D1	3	10	-//-
POP H	E1	3	10	-//-
POP PSW	F1	3	10	-//-
XTHL	E3	5	16	обмін вершини стеку з H,L
XTHG	EB	5	16	
SPHL	F9	1	6	(SP) ← (H, L)
OUT B <sub>2</sub> (port)	D3	3	10	вивод даних
IN B <sub>2</sub> (port)	DB	3	10	ввод даних
DI	F3	1	4	переривання заборонено
EI	FD	1	4	переривання дозволено
NOP	00	1	4	немає операції
HLT	76	1	7	зупинка

Примітка. А – реєстр акумулятор; В, С, D, E, H, L – реєстри загального призначення; В,С; D,E; H,L – реєстрові пари; М – комірка пам'яті; SP – показчик стеку; PC – лічильник команд; PSW – слово стану програми; Z, C, P, S – біти регистру ознак.

Таблиця А.2 – Таблиця визначення команд за об'єктним кодом

СТЦ	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
<b>0</b>	NOP	LXI B, B <sub>2</sub> , B <sub>3</sub>	STAX B	INX B	INR B	DCR B	MVI B, B <sub>2</sub>	RLC	—	DAD B	LDAX B	DCX B	INR C	DCR C	MVI C, B <sub>2</sub>	RRC
<b>1</b>	—	LXI D, B <sub>2</sub> , B <sub>3</sub>	STAX D	INX D	INR D	DCR D	MVI D, B <sub>2</sub>	RAL	—	DAD D	LDAX D	DCX D	INR E	DCR E	MVI E, B <sub>2</sub>	RAR
<b>2</b>	—	LXI H, B <sub>2</sub> , B <sub>3</sub>	SHLD B <sub>2</sub> , B <sub>3</sub>	INX H	INR H	DCR H	MVI H, B <sub>2</sub>	DAA	—	DAD H	LHLD B <sub>2</sub> , B <sub>3</sub>	DCX H	INR L	DCR L	MVI L, B <sub>2</sub>	CMA
<b>3</b>	—	LXI SP, B <sub>2</sub> , B <sub>3</sub>	STA B <sub>2</sub> , B <sub>3</sub>	INX SP	INR M	DCR M	MVI M, B <sub>2</sub>	STC	—	DAD SP	LDA B <sub>2</sub> , B <sub>3</sub>	DCX SP	INR A	DCR A	MVI A, B <sub>2</sub>	CMC
<b>4</b>	MOV B, B	MOV B, C	MOV B, D	MOV B, E	MOV B, H	MOV B, L	MOV B, M	MOV B, A	MOV C, B	MOV C, C	MOV C, D	MOV C, E	MOV C, H	MOV C, L	MOV C, M	MOV C, A
<b>5</b>	MOV D, B	MOV D, C	MOV D, D	MOV D, E	MOV D, H	MOV D, L	MOV D, M	MOV D, A	MOV E, B	MOV E, C	MOV E, D	MOV E, E	MOV E, H	MOV E, L	MOV E, M	MOV E, A
<b>6</b>	MOV H, B	MOV H, C	MOV H, D	MOV H, E	MOV H, H	MOV H, L	MOV H, M	MOV H, A	MOV L, B	MOV L, C	MOV L, D	MOV L, E	MOV L, H	MOV L, L	MOV L, M	MOV L, A
<b>7</b>	MOV M, B	MOV M, C	MOV M, D	MOV M, E	MOV M, H	MOV M, L	HLT	MOV M, A	MOV A, B	MOV A, C	MOV A, D	MOV A, E	MOV A, H	MOV A, L	MOV A, M	MOV A, A
<b>8</b>	ADD B	ADD C	ADD D	ADD E	ADD H	ADD L	ADD M	ADD A	ADC B	ADC C	ADC D	ADC E	ADC H	ADC L	ADC M	ADC A
<b>9</b>	SUB B	SUB C	SUB D	SUB E	SUB H	SUB L	SUB M	SUB A	SBB B	SBB C	SBB D	SBB E	SBB H	SBB L	SBB M	SBB A
<b>A</b>	ANA B	ANA C	ANA D	ANA E	ANA H	ANA L	ANA M	ANA A	XRA B	XRA C	XRA D	XRA E	XRA H	XRA L	XRA M	XRA A
<b>B</b>	ORA B	ORA C	ORA D	ORA E	ORA H	ORA L	ORA M	ORA A	CMP B	CMP C	CMP D	CMPE	CMP H	CMP L	CMP M	CMP A
<b>C</b>	RNZ	POP B	JNZ B <sub>2</sub> B <sub>3</sub>	JMP B <sub>2</sub> B <sub>3</sub>	CNZ B <sub>2</sub> B <sub>3</sub>	PUSH B	ADI B <sub>2</sub>	RST 0	RZ	RET	JZ B <sub>2</sub> B <sub>3</sub>	—	CZ B <sub>2</sub> B <sub>3</sub>	CALL B <sub>2</sub> B <sub>3</sub>	ACI B <sub>2</sub>	RST 1
<b>D</b>	RNC	POP D	JNC B <sub>2</sub> B <sub>3</sub>	OUT B <sub>2</sub>	CNC B <sub>2</sub> B <sub>3</sub>	PUSH D	SUI B <sub>2</sub>	RST 2	RC	—	JC B <sub>2</sub> B <sub>3</sub>	IN B <sub>2</sub>	CC B <sub>2</sub> B <sub>3</sub>	—	SBI B <sub>2</sub>	RST 3
<b>E</b>	RPO	POP H	JPO B <sub>2</sub> B <sub>3</sub>	XTHL	CPO B <sub>2</sub> B <sub>3</sub>	PUSH H	ANI B <sub>2</sub>	RST 4	RPE	PCHL	JPE B <sub>2</sub> B <sub>3</sub>	XCHG	CPE B <sub>2</sub> B <sub>3</sub>	—	XRI B <sub>2</sub>	RST 5
<b>F</b>	RP	POP SW	JP B <sub>2</sub> B <sub>3</sub>	DI	CP B <sub>2</sub> B <sub>3</sub>	PUSH PSW	ORI B <sub>2</sub>	RST 6	RM	SPHL	JM B <sub>2</sub> B <sub>3</sub>	EI	CM B <sub>2</sub> B <sub>3</sub>	—	CPI B <sub>2</sub>	RST 7

*Електронне навчальне видання*

**ЄСАУЛОВ** Сергій Михайлович  
**БАБІЧЕВА** Ольга Федорівна  
**ВОРОНОВ** Роман Володимирович

## **МІКРОПРОЦЕСОРНІ ПРИСТРОЇ В СИСТЕМАХ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ**

### **КОНСПЕКТ ЛЕКЦІЙ**

*(для здобувачів другого (магістерського) рівня вищої освіти денної  
та заочної форм навчання  
спеціальності 141 – Електроенергетика, електротехніка та  
електромеханіка)*

Відповідальний за випуск *О. Ф. Бабічева*  
*За авторською редакцією*  
Комп'ютерне верстання *Р. В. Воронов*

План 2022, поз. 76Л

---

Підп. до друку 24.03.2023. Формат 60 × 84/16.  
Ум. друк. арк. 8,1.

Видавець і виготовлювач:  
Харківський національний університет  
міського господарства імені О. М. Бекетова,  
вул. Маршала Бажанова, 17, Харків, 61002  
Електронна адреса: office@kname.edu.ua  
Свідоцтво суб'єкта видавничої справи:  
ДК № 5328 від 11.04.2017.