

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
МІСЬКОГО ГОСПОДАРСТВА імені О. М. БЕКЕТОВА**



І. О. Костенко, О. О. Шкурпела, А. І. Кузнецов

МІКРОПРОЦЕСОРНА ТЕХНІКА

КОНСПЕКТ ЛЕКЦІЙ

*(для здобувачів першого (бакалаврського) рівня вищої освіти
всіх форм навчання зі спеціальності
141 – Електроенергетика, електротехніка та електромеханіка)*

**Харків
ХНУМГ ім. О. М. Бекетова
2025**

Костенко І. О. Мікропроцесорна техніка : конспект лекцій для здобувачів першого (бакалаврського) рівня вищої освіти всіх форм навчання зі спеціальності 141 – Електроенергетика, електротехніка та електромеханіка / І. О. Костенко, О. О. Шкурпела, А. І. Кузнецов ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків : ХНУМГ ім. О. М. Бекетова, 2025. – 87 с.

Автори:

канд. техн. наук, асист. І. О. Костенко,
канд. техн. наук, асист. О. О. Шкурпела,
канд. техн. наук, доц. А. І. Кузнецов

Рецензент

О. Б. Єгоров, кандидат технічних наук, доцент кафедри альтернативної електроенергетики та електротехніки (Харківський національний університет міського господарства імені О. М. Бекетова)

Рекомендовано кафедрою альтернативної електроенергетики та електротехніки, протокол № 5 від 19 листопада 2024 р.

Конспект лекцій складено з метою допомогти студентам спеціальності 141 – Електроенергетика, електротехніка та електромеханіка під час підготовки до практичних занять, лабораторних робіт, екзамену та виконання розрахунково-графічної роботи.

ЗМІСТ

ВСТУП	5
1 ОСНОВИ СХЕМОТЕХНІКИ ЦИФРОВИХ ПРИСТРОЇВ.	
ЛОГІЧНІ ЕЛЕМЕНТИ	6
1.1 Алгебра логіки	6
1.2 Реалізація простих логічних функцій. Логічні елементи ...	9
2 ТРИГЕРИ	16
2.1 Загальні відомості про тригери та їхнє призначення	16
2.2 Тригери на логічних елементах	16
2.3 RS-тригер	19
2.4 Тригер D-типу (D-тригер)	20
2.5 Тригер T-типу (T-тригер)	21
2.6 JK- тригер	22
3 ЦИФРОВІ МІКРОЕЛЕКТРОННІ ПРИСТРОЇ	23
3.1 Поняття про цифрові мікроелектронні пристрої	23
3.2 Реалізація складних логічних функцій	24
3.3 Дешифратори	27
3.4 Мультиплексори	29
3.5 Лічильники імпульсів	30
3.6 Регістри	35
4 ЦИФРО-АНАЛОГОВІ (ЦАП) І АНАЛОГОВО-ЦИФРОВІ (АЦП) ПЕРЕТВОРЮВАЧІ	37
4.1 Будова і робота ЦАП	37
4.2 АЦП	43
5 СХЕМОТЕХНІКА ПРОГРАМОВАНИХ ЦИФРОВИХ ПРИСТРОЇВ. МІКРОПРОЦЕСОРНІ ПРИСТРОЇ	47
5.1 Загальні положення. Дещо з історії процесорів	47

5.2 Особливості роботи і використання мікропроцесорів. Двійкова система числення. Структура мікропроцесорної системи.....	51
5.3 Структура мікропроцесора	56
6 ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ	59
6.1 Загальні відомості	59
6.2 Запам'ятовувальні пристрої для зберігання оперативної інформації (ОЗП)	60
6.3 Запам'ятовувальні пристрої для зберігання постійної інформації (ПЗП)	61
7 ЗАВДАННЯ І ВИКОНАННЯ ПРОГРАМ МІКРОПРОЦЕСОРНОГО ПРИСТРОЮ	64
7.1 Алгоритм. Способи представлення алгоритму	64
7.2 Завдання і виконання програм МПП	65
7.3 Шістнадцяткова система числення	66
7.4 Асемблер	68
8 ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ. МІКРОКОНТРОЛЕРИ	71
8.1 Програмовані логічні контролери	71
8.2 Мікроконтролери	74
8.3 Структура мікроконтролера	79
8.4 Засоби програмування для мікроконтролерів.....	81
8.5 Співставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих	84
СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ	86

ВСТУП

Широке використання мікропроцесорної техніки у всіх сферах людської діяльності, ефективність мікропроцесорних систем пов'язані як з розвитком багатьох суміжних технічних розробок, так і з рівнем підготовки у цій галузі спеціалістів самого різного профілю.

Метою викладання навчальної дисципліни «Мікропроцесорна техніка» є формування у майбутніх фахівців знань з основ мікропроцесорної техніки, функціональних можливостей мікропроцесорних систем і призначення пов'язаних з ними об'єктів.

Основними завданнями вивчення дисципліни «Мікропроцесорна техніка» є опанування будови і принципів роботи пристроїв і систем програмованої логіки з метою застосування у різноманітних пристроях керування.

Згідно з вимогами освітньої програми студенти повинні:

а) знати:

- принципи будови цифрових автоматів з жорсткою і програмованою логікою з позиції порівняльного аналізу ефективності їхнього застосування за конкретної виробничої діяльності майбутнього фахівця;

- побудову взаємозв'язків між окремими вузлами та пристроями, що об'єднуються мікропроцесорною системою;

б) вміти:

- оцінювати техніко-економічну ефективність застосування мікропроцесорних пристроїв;

- формувати технічні вимоги до різного роду електротехнічних пристроїв з мікропроцесорним керуванням;

- укладати алгоритми функціонування пристроїв з мікропроцесорним керуванням;

в) мати компетентності:

- в питаннях застосування мікропроцесорних пристроїв у пристроях та системах енергетики та світлотехніки.

1 ОСНОВИ СХЕМОТЕХНІКИ ЦИФРОВИХ ПРИСТРОЇВ. ЛОГІЧНІ ЕЛЕМЕНТИ

1. Алгебра логіки.

2. Реалізація простих логічних функцій. Логічні елементи.

ОСНОВНІ ПОНЯТТЯ

Цифрові (логічні) ІМС; математична логіка (алгебра Буля); подія; двійкова (логічна) змінна; двійкова (логічна, перемикальна) функція; операція НІ (логічне заперечення, операція інверсії); операція АБО (логічне додавання, диз'юнкція), операція І (логічне множення, кон'юнкція); закони алгебри логіки; тотожності алгебри логіки; правила де Моргана; логічні схеми, логічні елементи; логічні (цифрові) автомати; таблиця істинності; динамічний режим роботи логічних елементів.

1.1 Алгебра логіки

Основу сучасних пристроїв обробки інформації складають **цифрові (логічні) ІМС**.

Аналіз роботи цифрових пристроїв базується на використанні апарату **математичної логіки – алгебри Джорджа Буля (1815–1864 рр.)**. В її основі лежить поняття **події**, що оцінюється з точки зору її настання: вона може статися або не статися.

Тоді кожна **подію** можна вважати **істинною**, що може моделюватися одиницею «1» (високим рівнем напруги при електричному моделюванні), або **хибною** – моделюється нулем «0» (низьким рівнем напруги).

Обробка інформації, поданої у вигляді подій, ведеться у **двійковій системі числення**. Вона має тільки дві цифри: 0 і 1.

Величина, котра може приймати тільки ці два значення, називається **двійковою (логічною) змінною**.

Складна подія, що залежить від декількох двійкових змінних, називається **двійковою (логічною, перемикальною) функцією**:

$$y = f(x_1, x_2, \dots, x_n), \quad \text{де } x = \{1, 0\}. \quad (1.1)$$

Алгебра логіки дозволяє виконувати математичний запис логічних подій і зв'язків між ними, а це дає можливість аналітично описувати будову і роботу цифрових пристроїв (бо цифрові пристрої оброблюють інформацію, представлену у вигляді сигналів, що змінюються за законом дискретної функції).

В алгебрі логіки є три основних **логічних дії (операції, функції)**:

- **операція НІ – логічне заперечення:**

функція має зворотне значення до змінної, від якої вона залежить

$$y = \bar{x} \quad (1.2)$$

(читається: у дорівнює не x);

- **операція АБО – логічне додавання (диз'юнкція):**

функція істинна, якщо істинна хоча б одна з незалежних змінних, що до неї входять

$$y = x_1 + x_2; \quad (1.3)$$

- **операція І – логічне множення (кон'юнкція):**

функція істинна, якщо істинні усі незалежні змінні, що до неї входять

$$y = x_1 \cdot x_2. \quad (1.4)$$

Зверніть увагу: в алгебрі логіки немає операцій віднімання і ділення.

Порядок дій в алгебрі логіки такий: перш за все виконується операція НІ, потім І й насамкінець АБО.

Для зміни порядку дій, як і в звичайній алгебрі, застосовують дужки.

Для алгебри логіки справедливі закони:

- **переставний (комутативний)**

$$x_1 + x_2 = x_2 + x_1 \quad \text{та} \quad x_1 \cdot x_2 = x_2 \cdot x_1; \quad (1.5)$$

- **сполучний (асоціативний)**

$$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3$$

$$\text{та} \quad x_1 \cdot x_2 \cdot x_3 = x_1(x_2 \cdot x_3) = (x_1 \cdot x_2) x_3; \quad (1.6)$$

- **розподільний (дистрибутивний)**

$$x_1(x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3 \quad (1.7)$$

(зазначимо, що такі ж закони діють і в звичайній алгебрі).

Тотожності алгебри логіки:

$$x + x = x \quad \text{та} \quad x \cdot x = x; \quad (1.8)$$

$$x + \bar{x} = 1 \quad \text{та} \quad x \cdot \bar{x} = 0; \quad (1.9)$$

$$x + 0 = x \quad \text{та} \quad x \cdot 0 = 0; \quad (1.10)$$

$$\overline{\bar{x}} = x; \quad (1.11)$$

$$x_1 + x_1 \cdot x_2 + x_1 \cdot x_3 = x_1; \quad (1.12)$$

$$x_1 + \bar{x}_1 \cdot x_2 = x_1 + x_2; \quad (1.13)$$

$$x_1 \cdot x_2 + x_1 \cdot \bar{x}_2 = x_1; \quad (1.14)$$

$$x_1(x_1 + x_2) = x_1; \quad (1.15)$$

$$x_1(\bar{x}_1 + x_2) = x_1 \cdot x_2; \quad (1.16)$$

$$(x_1 + x_2)(x_1 + x_3) = x_1 + x_2 \cdot x_3. \quad (1.17)$$

До основних законів алгебри логіки також відносяться **закони інверсії для логічних додавання та множення – правила де Моргана:**

$$\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2} \quad \text{та} \quad \overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}. \quad (1.18)$$

Ці закони грають важливу роль при синтезі схем цифрових пристроїв, часто призводячи до суттєвого спрощення логічних функцій, а значить і схем пристроїв, що їх реалізують.

Взагалі, знання законів алгебри логіки дозволяє отримувати оптимальну за заданими критеріями схему пристрою, що забезпечує використання мінімального числа уніфікованих елементів, високу швидкодію та надійність.

У практичних цілях алгебру Буля першим у 1938 році застосував один із родоначальників математичної теорії інформації і кібернетики Клод Шеннон (США) при дослідженні електричних кіл з контактними перемикачами.

1.2 Реалізація простих логічних функцій. Логічні елементи

Практична реалізація аналітичного опису подій алгебри логіки виконується у вигляді **логічних схем**, що будуються з **логічних елементів** як **логічні (цифрові) автомати**. При цьому проєктувальника, зазвичай, не цікавить внутрішня будова логічних елементів. Вони розглядаються як функціональні вузли обробки цифрової інформації.

Отже, однією з найпростіших логічних функцій є функція заперечення НІ, яку ще називають **операцією інверсії**.

Графічне позначення елемента, що реалізує таку функцію наведено на рисунку 1.1. Як такий елемент може бути використано, наприклад, транзисторний ключ – підсилювач з СЕ, що працює у ключовому режимі: при високому рівні напруги на його вході на виході матимемо низький і навпаки.

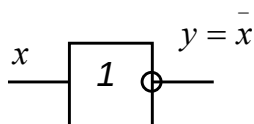


Рисунок 1.1 – Елемент НІ

Функцію, що її виконує логічний пристрій, для полегшення сприйняття часто представляють у вигляді таблиці, яку називають **таблицею істинності**. Кількість стовбців цієї таблиці дорівнює числу змінних, що входять до функції і є ще один стовбець, у якому вказують значення функції для кожної з можливих комбінацій вхідних змінних, а їхньому числу відповідає кількість рядків таблиці. У загальному випадку кількість рядків дорівнює 2^n , де n – число змінних.

Об'єднану таблицю істинності деяких основних логічних функцій, що залежать від двох змінних (усього таких функцій шістнадцять), наведено на рисунку 1.2. На рисунку також представлено інформацію про математичний запис функцій, їхні назви, графічне позначення, можливу реалізацію.

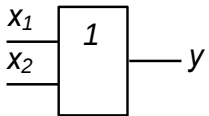
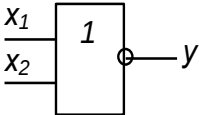
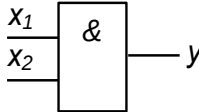
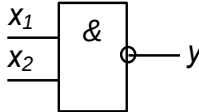
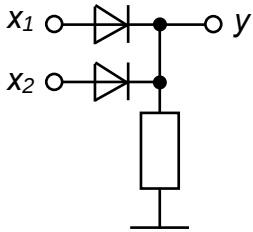
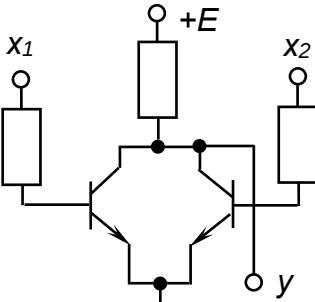
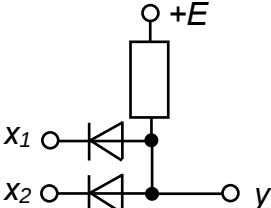
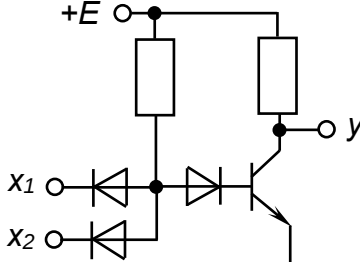
Вхідні змінні		Функція y			
x_1	x_2	АБО	АБО-НІ	I	I-НІ
0	0	0	1	0	1
0	1	1	0	0	1
1	0	1	0	0	1
1	1	1	0	1	0
Математичний запис (формула)		$y = x_1 + x_2 = x_1 \vee x_2$	$y = \overline{x_1 + x_2}$	$y = x_1 \cdot x_2 = x_1 \wedge x_2$	$y = \overline{x_1 \cdot x_2}$
Назва функції		Логічне додавання (диз'юнкція) – функція АБО	Заперечення логічного додавання (стрілка Пірса) – функція АБО-НІ	Логічне множення (кон'юнкція) – функція I	Заперечення логічного множення (штрих Шеффера) – функція I-НІ
Графічне позначення елемента, що реалізує функцію					
Можлива реалізація		 Резисторно-діодна логіка (РДЛ)	 Резисторно-транзисторна логіка (РТЛ)	 Резисторно-діодна логіка (РДЛ)	 Резисторно-діодно-транзисторна логіка (РДТЛ)

Рисунок 1.2 – Деякі логічні функції двох змінних

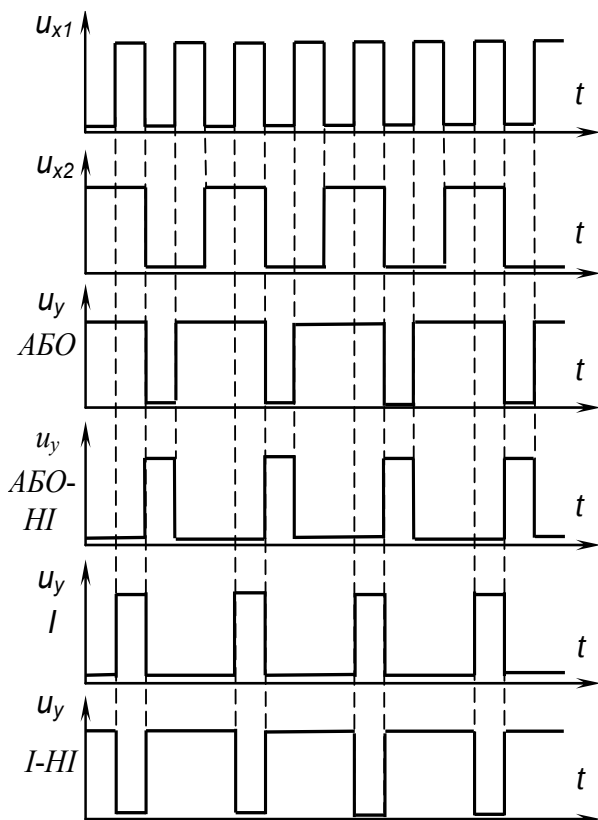


Рисунок 1.3 – Часові діаграми роботи деяких двовходових логічних елементів

З рисунку 1.2 видно, що, наприклад, функція АБО істинна, якщо істинною є хоча б одна із змінних, що до неї входять, а функція І – тільки у випадку істинності обох змінних.

Якщо вхідні сигнали змінювати з часом, як це показано на рисунку 1.3, можна отримати часові діаграми роботи, що відповідають **динамічному режиму роботи** логічних елементів. У загальному випадку кількість вхідних змінних (кількість входів) логічних елементів, необхідних для реалізації складних логічних функцій, може бути будь-якою. Реально у елементів, що випускаються у

вигляді ІМС, вона зазвичай складає 2 (чотири елементи в одному корпусі ІМС, що мають спільні кола живлення), 3 (три елементи), 4 (два елементи), 8 (один елемент). Частіше це елементи І-НІ, АБО-НІ.

За елементною базою, на якій виконано логічні елементи, їх підрозділяють на резисторно-діодні (РДЛ – резисторно-діодна логіка), резисторно-транзисторні (РТЛ), резисторно-діодно-транзисторні (РДТЛ), транзисторно-транзисторні (ТТЛ), на К-МОН комплементарних транзисторах (К-МОН-логіка) і деякі інші.

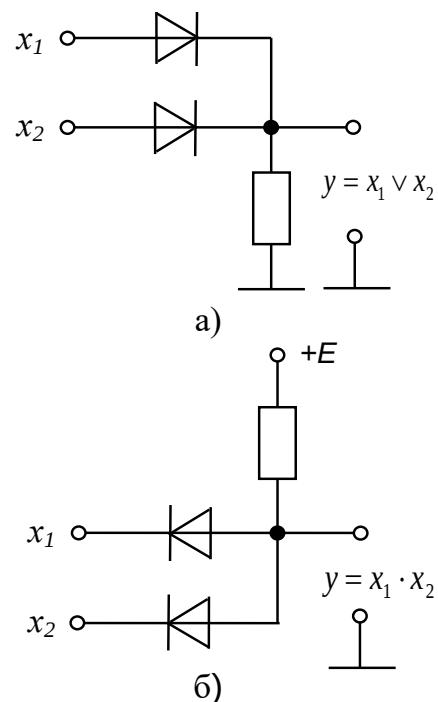


Рисунок 1.4 – Резисторно-діодні елементи 2АБО (а) та 2І (б)

Схеми двохходових резисторно-діодних елементів 2АБО та 2І наведено на рисунку 1.4, а і рисунку 1.4, б відповідно.

При своїй схемній простоті вони мають суттєвий недолік: падіння напруги на діодних ключах не дозволяє реалізовувати складні логічні функції з послідовним вмиканням великого числа елементів за прийнятних значень напруги джерела живлення. Необхідно забезпечувати проміжне підсилення сигналів.

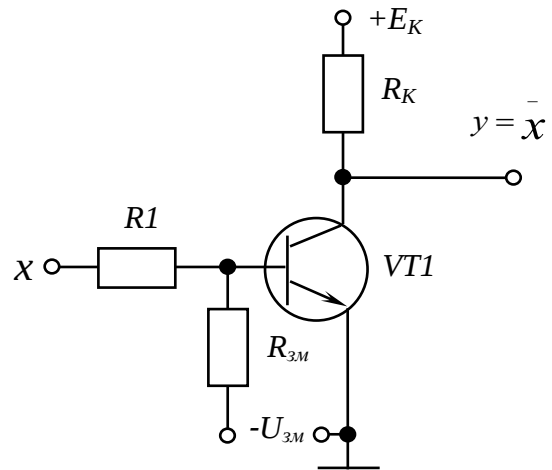


Рисунок 1.5 – Інвертор (елемент НІ)

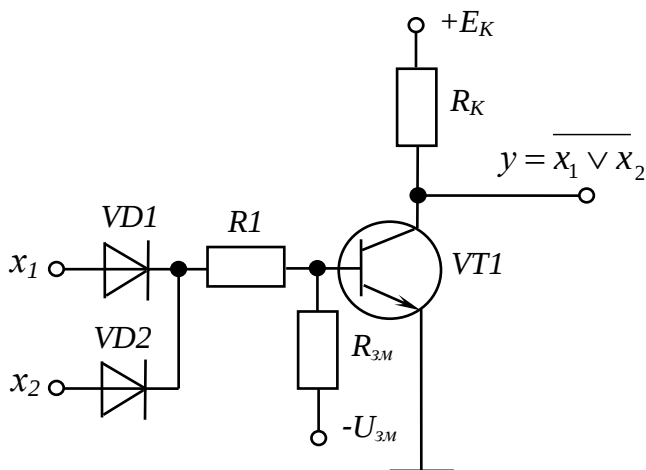


Рисунок 1.6 – РДТЛ елемент 2АБО-НІ

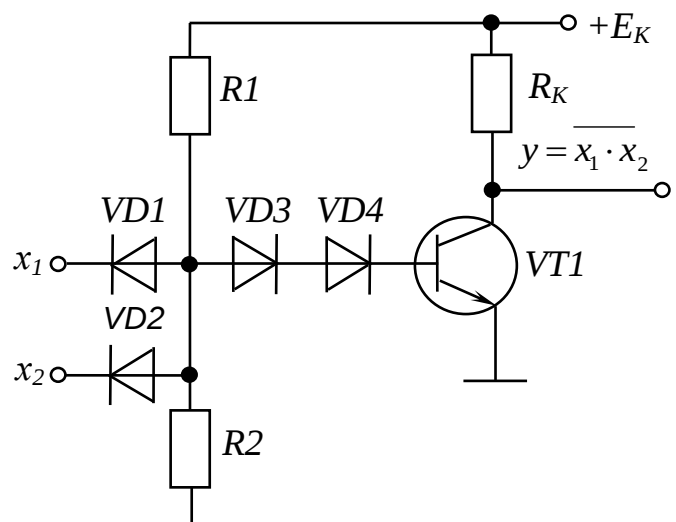


Рисунок 1.7 – РДТЛ елемент 2І-НІ

Підсилення забезпечують елементи, побудовані на основі транзисторних ключів. Наприклад, це **інвертор**, схема якого наведена на рисунку 1.5. Зверніть увагу: подача невеликої негативної напруги зміщення U_{3M} забезпечує надійне закривання транзистора – збільшує завадостійкість елемента.

Схеми діодно-транзисторних елементів 2АБО-НІ та 2І-НІ наведено на рисунку 1.6 і рисунку 1.7 відповідно.

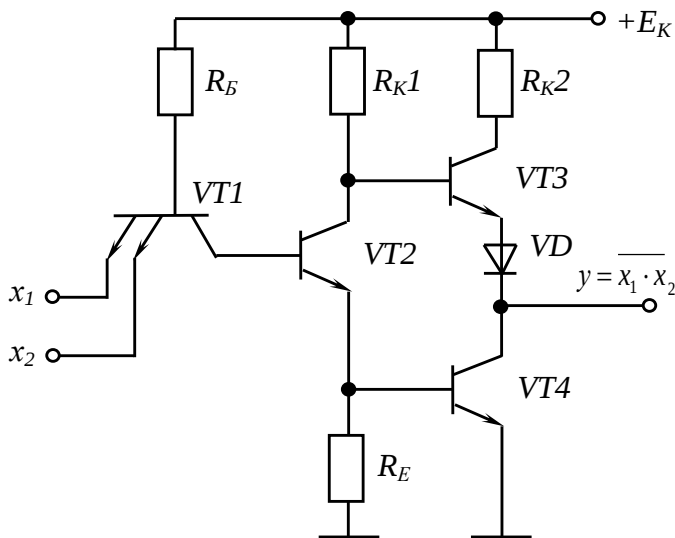


Рисунок 1.8 – TTL елемент 2І-НІ

наявність на вході багатоемітерного транзистора VT1, що є набутком інтегральної технології і замінює вхідний діодний вузол елементів РДТЛ.

Елемент TTL також має складний двотактний вихідний каскад, що дозволяє збільшити навантажувальну здатність елемента – знижує вплив опору колекторного резистора на значення напруги вихідного сигналу, що відповідає 1.

Діод VD (як пороговий елемент з напругою відкриття близько 0,6 В) надійно забезпечує закритий стан транзистора VT3 при відкритому VT2 (падіння на якому складає 0,2 - 0,4 В).

Розсмоктування зарядів у базі насиченого транзистора при відкритті триває значний час. Для його зменшення транзистору забезпечують стан, коли він закритим знаходиться на межі режимів насичення і активного. Для цього в ІМС застосовують діод Шоттки, що вмикається паралельно до переходу база-колектор транзистора (анодом до бази). Швидкодія при цьому значно підвищується. Така структура називається **транзистором Шоттки**.

Діоди VD3 і VD4 у елемента І-НІ забезпечують виключення відкриваючої дії на транзистор напруги, що падає на діодах VD1 або VD2 (замінюють $U_{зм}$).

Широке розповсюдження знайшли елементи TTL. Схему двоходового TTL елемента 2І-НІ наведено на рисунку 1.8.

Відмінною його ознакою є

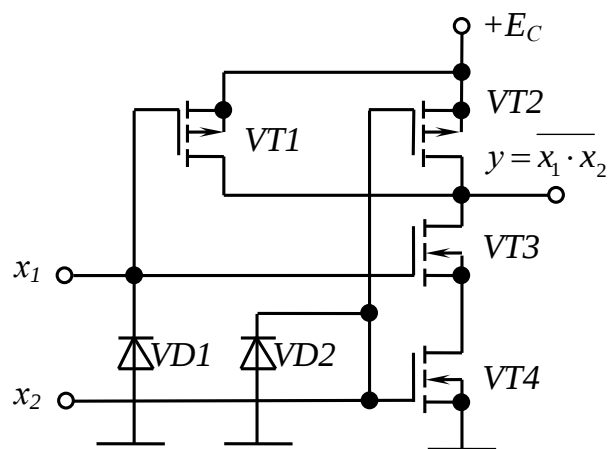


Рисунок 1.9 – Елемент 2І-НІ К-МОП-логіки

На рисунку 1.9 наведено схему двовходового елемента І-НІ, виконаного на комплементарних К-МОН транзисторах. Як видно з рисунку, елемент складається тільки з чотирьох МОН-транзисторів, що одночасно виконують і роль резисторів, бо опір їхнього каналу становить від десятків до сотень Ом.

Вихід елемента транзисторними ключами $VT1$ або $VT2$ підмикається до шини живлення, а $VT3$ і $VT4$ – до нульової шини.

Діоди $VD1$ і $VD2$ забезпечують захист вхідних кіл ІМС від подачі від’ємної напруги.

Зазначимо, що захисні ланцюжки встановлюють на входах багатьох видів ІМС з метою їхнього захисту як від напруги недопустимої полярності, так і від перевищення вхідною напругою допустимого значення. Наприклад, від дії статичної електрики у разі дотику людини до виводів ІМС.

Елементи К-МОН прості у виготовленні (а значить дешеві), мають більшу завадостійкість ніж елементи ТТЛ, а за частотними властивостями у останній час наближаються до них.

Перевагою К-МОН логіки є ще й те, що вона працездатна у широкому діапазоні змін напруги живлення. Отже, якщо для ІМС ТТЛ типове значення напруги живлення становить $5\text{ В} \pm 5\%$, то для ІМС К-МОН вона може становити від 3 В до 15 В.

2 ТРИГЕРИ

1. Загальні відомості про тригери та їхнє призначення.
2. Тригери на логічних елементах.
3. *RS*-тригер.
4. Тригер *D*-типу (*D*-тригер).
5. Тригер *T*-типу (*T*-тригер).
6. *JK*-тригер.

ОСНОВНІ ПОНЯТТЯ

Тригери; тригери в інтегральному виконанні; вихід прямий; вихід інверсний; вхід інформаційний; вхід тактовий (синхронізуючий;) вхід прямий; вхід інверсний; вхід потенціальний; вхід імпульсний; тригери асинхронні, тригери синхронні; таблиця переходів; стан тригера (нульовий, одиничний, невизначений); двовходовий асинхронний *RS*-тригер з прямими і інверсними входами; *D*-тригер; *T*-тригер; *JK*-тригер.

2.1 Загальні відомості про тригери та їхнє призначення

Основою послідовнісних логічних пристроїв (пристроїв з пам'яттю) є тригери. Тригер забезпечує запам'ятовування елементарного об'єму дискретної інформації – 1 біт.

Тригери (від англійського «*trigger*» – заскочка) – це спускові імпульсні пристрої з позитивним зворотним зв'язком, що мають два сталих стани рівноваги і можуть переходити із одного стану в інший під дією сигналу, який перевищує за значенням деякий рівень – поріг спрацьовування пристрою.

Тригери можуть бути побудовані на напівпровідникових приладах, що мають ділянку з негативною крутістю характеристики (наприклад, на тиристорах). Сучасні тригери зазвичай будують на основі двокаскадних

підсилювачів з позитивним зворотним зв'язком. Тригери в інтегральному виконанні будують на логічних цифрових елементах.

Використовуються тригери для таких цілей:

- 1) перетворення імпульсу довільної форми у прямокутну, тобто застосовуються як формувачі імпульсів прямокутної форми (тригери Шмітта);
- 2) створення електронних реле;
- 3) створення пристроїв підрахунку імпульсів і ділення частоти надходження імпульсів;
- 4) зберігання інформації у двійковому коді.

2.2 Тригери на логічних елементах

Тригери в інтегральному виконанні будуються з простих логічних елементів типу АБО-НІ, І-НІ. Зазвичай мікросхема вміщує 1÷4 тригери із спільними колами живлення, а інколи і спільними колами синхронізації або керування.

У загальному випадку тригер складається з логічного пристрою керування та власне тригера як елемента пам'яті. Є велика кількість різноманітних схем тригерів з різними функціональними можливостями.

Узагальнену структурну схему тригера зображено на рисунку 2.1.

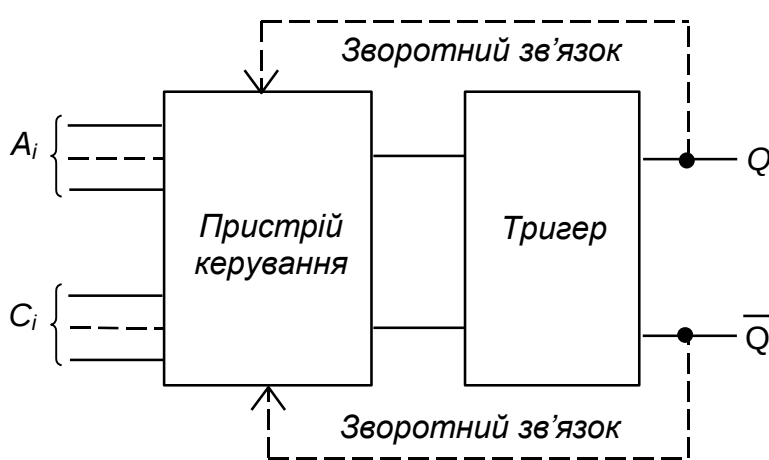


Рисунок 2.1 – Структурна схема тригера

Пристрій керування призначений для перетворення сигналів, що надходять до входів A_i , у вигляд, придатний для керування власне тригером, що виконує функцію елемента пам'яті.

Тригер має два виходи:

Q – **прямий** (одиничний), $\overline{Q}$

– **інверсний** (нульовий). Входи A_i називаються **інформаційними**, а входи C_i – **тактовими** або **синхронізуючими**.

За способом занесення інформації тригери поділяються на **асинхронні**, що змінюють свій стан одразу після надходження сигналу на певний інформаційний вхід, і **синхронні** (тактовані), які спрацьовують не тільки за наявності сигналів на інформаційних входах, а лише після надходження синхронізуючого (тактового) сигналу на певний вхід синхронізації.

Описують роботу тригерів (і послідовнісних пристроїв взагалі) також за допомогою логічних функцій або частіше задля наочності за допомогою **таблиць переходів**. У таблицях вказують всі можливі комбінації сигналів на інформаційних входах у певний момент часу t^i і стан, в який перейде тригер під дією цих сигналів у наступний момент часу t^{i+1} . Причому, наступний момент часу у асинхронного тригера настає одразу після зміни комбінації сигналів на інформаційних входах, а у синхронного – після надходження тактового сигналу (зазвичай це імпульс) на відповідний вхід синхронізації.

Стани тригера в таблицях переходів зазвичай вказують так:

0 – тригер має сигнал на виході $Q = 0$ (**нульовий стан**) незалежно від сигналів на входах;

1 – тригер має сигнал на виході $Q = 1$ (**одиничний стан**) незалежно від сигналів на входах;

Q_i – стан тригера не змінюється при зміні сигналів на входах;

$\bar{Q}_i$ – стан тригера змінюється на протилежний при зміні сигналів на входах;

X – **невизначений стан** тригера, коли він після зміни сигналів на входах рівноможливо може опинитися в нульовому ($Q = 0$) або в одиничному ($Q = 1$) стані.

Стверджують, що навіть за найпростішої конфігурації тригерного пристрою, яка має один інформаційний вхід і два виходи, можна отримати

25 функціональних різновидів тригерів. При двох входах їх буде вже 625. Практично ж застосовують 6–8 типів.

Найбільш розповсюджені з них *RS*-тригери, *D*-тригери, *T*-тригери, *JK*-тригери. Часто тригери будують як комбіновані: *RSD*-тригер, *RST*-тригер і т. п.

2.3 *RS*-тригер

Умовні позначення **двовходових асинхронних *RS*-тригерів з прямими** (такими, що реагують на наявність 1) і **інверсними** (такими, що реагують на наявність 0) **входами** наведено на рисунку 2.2.

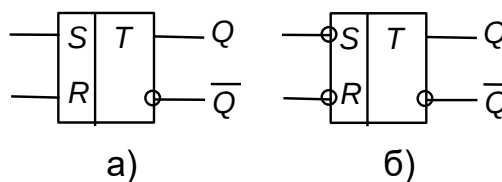


Рисунок 2.2 – Умовні позначення *RS*-тригера з прямими (а) й інверсними (б) входами

Свою назву *RS*-тригер отримав від перших літер англійських слів «set» – встановлювати (*S*) та «reset» – відновлювати (*R*).

S – інформаційний вхід, призначений для установаження тригера в одиничний стан ($Q = 1$), а *R* – вхід, призначений для повертання тригера у нульовий стан ($Q = 0$).

Роботу тригерів описують відповідні таблиці переходів наведені в таблиці 2.1.

Таблиця 2.1 – Таблиці переходів *RS*-тригерів
а) з прямими входами б) з інверсними входами

t_i		t_{i+1}
<i>S</i>	<i>R</i>	<i>Q</i>
1	0	1
0	1	0
0	0	Q_i
1	1	X

t_i		t_{i+1}
<i>S</i>	<i>R</i>	<i>Q</i>
1	0	0
0	1	1
1	1	Q_i
0	0	X

Схеми таких RS -тригерів, побудованих на елементах І-НІ та АБО-НІ зображено на рисунку 2.3.

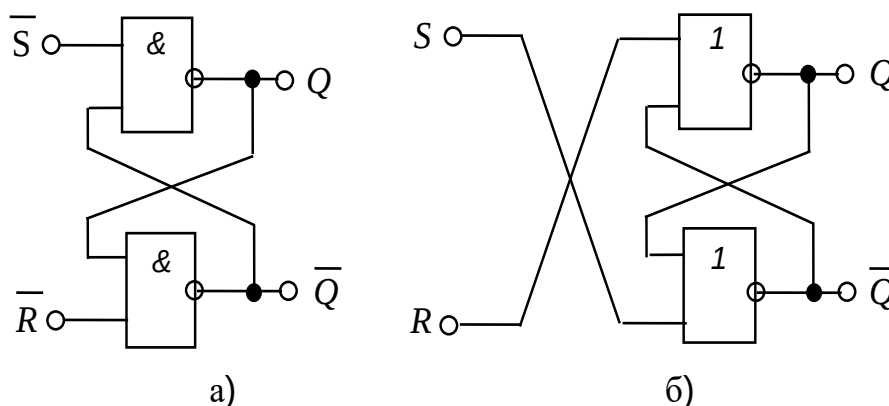


Рисунок 2.3 – RS -тригер
з інверсними входами на елементах І-НІ (а)
та з прямими входами на елементах АБО-НІ (б)

Схема і умовне позначення синхронного RS -тригера з прямими входами, побудованого на елементах І-НІ, наведено на рисунку 2.4.

Варто зазначити, що **тактові входи** бувають **потенціальні прямі**, як у даному випадку (тригер змінює свій стан при надходженні сигналу 1 на вхід C), та **інверсні** (тригер змінює стан при надходженні сигналу 0), або імпульсні, також прямі й інверсні (коли тригер змінює свій стан при зміні сигналу на тактовому вході з 0 на 1 або з 1 на 0 відповідно).

2.4 Тригер D -типу (D -тригер)

D -тригер (від англійського *delay* – затримка) має два входи: D – інформаційний та C – тактовий (синхронізуючий): D -тригер синхронний. А це значить, що інформація, яка надходить на вхід D , запам'ятовується лише при надходженні синхронізуючого імпульсу на вхід C , тобто із затримкою на час надходження останнього. Тому D -тригер ще називають тригером затримки.

Умовне позначення D -тригера з прямим імпульсним входом синхронізації та таблиця переходів наведено на рисунку 2.5, а часові діаграми його роботи – на рисунку 2.6.

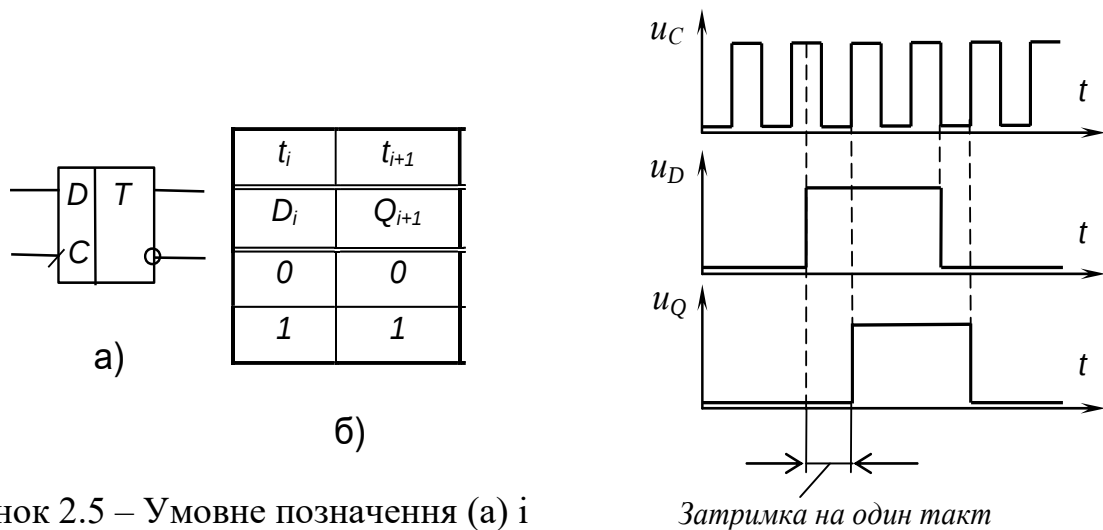


Рисунок 2.5 – Умовне позначення (а) і таблиця переходів (б) D -тригера

Рисунок 2.6 – Часові діаграми роботи D -тригера

2.5 Тригер T -типу (T -тригер)

T -тригер (від англійського «toggle» – перекидатись) ще називають тригером поділювачем на два або лічильним тригером. Тригер має тільки один тактовий вхід, а його стан змінюється на протилежний з надходженням на вхід кожного імпульсу. Цей тригер використовують для лічення та ділення частоти

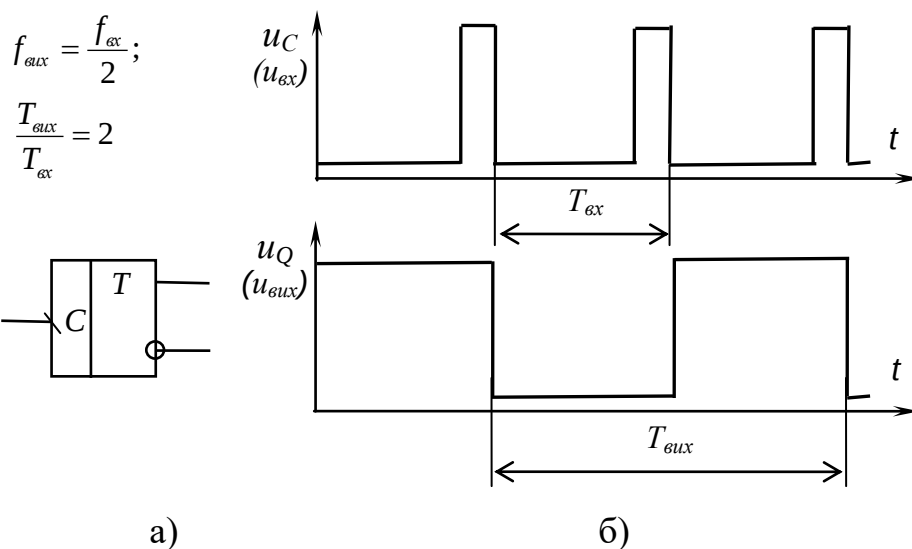


Рисунок 2.7 – Умовне позначення (а) та часові діаграми роботи (б) T -тригера

імпульсів.

Умовне позначення та часові діаграми роботи T -тригера з інверсним імпульсним входом наведено на рисунку 2.7.

2.6 JK -тригер

Синхронний **JK -тригер** має два інформаційних входи J (*jump* – стрибок) і K (*kill* – відключення) та тактовий C . Умовне позначення та таблиця переходів JK -тригера з прямим імпульсним тактовим входом наведено на рисунку 2.8.

JK -тригер є універсальним, бо він може виконувати роль RS -тригера, якщо використовувати вхід J як S , а K як R (при цьому таблиця переходів RS -тригера відповідає першим трьом рядкам таблиці переходів JK -тригера). Якщо задати одиниці на обох інформаційних входах, JK -тригер стає T -тригером. Схеми використання JK -тригера як T -тригера та D -тригера зображено на рисунку 2.9.

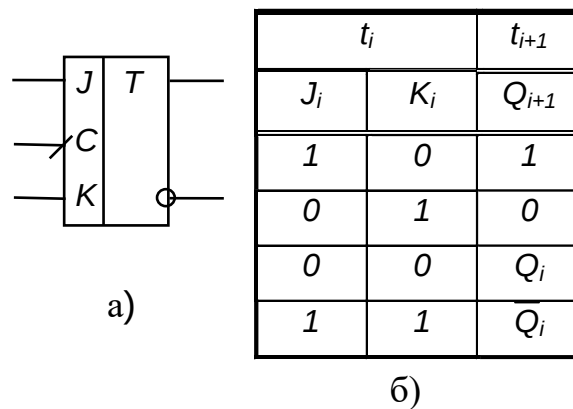


Рисунок 2.8 – Умовне позначення (а) та таблиця переходів (б) JK -тригера

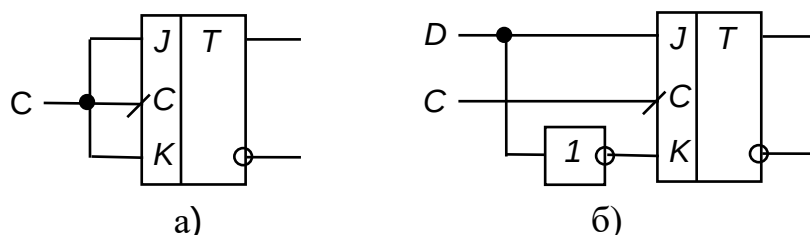


Рисунок 2.9 – Використання JK -тригера як T -тригера (а) та D -тригера (б)

3 ЦИФРОВІ МІКРОЕЛЕКТРОННІ ПРИСТРОЇ

- 1. Поняття про цифрові мікроелектронні пристрої.**
- 2. Реалізація складних логічних функцій.**
- 3. Дешифратори.**
- 4. Мультиплексори.**
- 5. Лічильники імпульсів.**
- 6. Регістри.**

ОСНОВНІ ПОНЯТТЯ

Цифрові мікроелектронні пристрої; комбінаційні пристрої; послідовнісні пристрої; функціонально повна система логічних елементів (базис); дешифратор; двійковий дешифратор; двійково-десятковий дешифратор; двійково-семисегментний дешифратор; мультиплексор; лічильник імпульсів (простий, віднімаючий, реверсивний, послідовний, паралельний, двійково-десятковий); послідовний двійковий код; двійково-десятковий код; регістр.

3.1 Поняття про цифрові мікроелектронні пристрої

Цифрові мікроелектронні пристрої становлять дискретні цифрові автомати, виконані на ІМС і призначені для обробки інформації, що представлена у вигляді цифрового коду. Вони використовуються для створення цифрових інформаційних, вимірювальних систем та систем керування.

Усі цифрові пристрої поділяються на два великих класи: комбінаційні і послідовнісні.

Комбінаційні пристрої реалізують функції, які залежать тільки від комбінації змінних, що до них входять, у певний момент часу і не залежать від стану пристрою у попередній момент часу. Найпростішими прикладами таких пристроїв є логічні елементи.

Послідовнісні (від слова «послідовність») **пристрої** реалізують функції, що залежать не тільки від комбінації вхідних змінних у певний момент часу, а

ще й від стану пристрою у попередній момент часу: вони мають пам'ять. Найпростішими прикладами таких пристроїв є тригери.

Основними (найбільш вживаними) мікроелектронними цифровими пристроями є:

- 1) дешифратори;
- 2) мультиплексори;
- 3) лічильники імпульсів;
- 4) регістри;
- 5) цифро-аналогові та аналого-цифрові перетворювачі.

Будуються ці пристрої на логічних елементах і тригерах.

3.2 Реалізація складних логічних функцій

Складні логічні функції реалізують на ІМС простих логічних елементів.

Мінімальний набір логічних елементів, що реалізують деякі прості логічні функції і за наявності необмеженої кількості яких можна реалізувати наскільки завгодно складну логічну функцію, називають **функціонально повною системою логічних елементів або базисом**.

Найбільш відомими функціонально повними системами є:

- 1) елементи, що реалізують функції алгебри Буля – І, АБО, НІ;
- 2) елемент, що реалізує функцію штрих Шеффера – І-НІ;
- 3) елемент, що реалізує функцію стрілка Пірса – АБО-НІ.

Якщо уважно подивимось на таблицю істинності логічних елементів (див. рис. 1.2), то побачимо, що, наприклад, елемент І для одиниць є елементом АБО для нулів. Тобто, якщо для прямих значень

$$y = x_1 \cdot x_2,$$

то для інверсних $\bar{y} = \bar{x}_1 + \bar{x}_2.$ (3.1)

Для елемента І-НІ можна записати

$$y = \overline{x_1 \cdot x_2} = \bar{x}_1 + \bar{x}_2, \quad (3.2)$$

а для елемента АБО-НІ

$$y = \overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2} . \quad (3.3)$$

Вирази (3.2) і (3.3) називають правилами де Моргана.

Якщо задати, наприклад, x_2 рівним відповідно 1 або 0, то матимемо

$$y = \overline{x_1 \cdot x_2} = \overline{x_1 \cdot 1} = \overline{x_1} , \quad (3.4)$$

$$\text{або} \quad y = \overline{x_1 + x_2} = \overline{x_1 + 0} = \overline{x_1} . \quad (3.5)$$

Отже, ми можемо використовувати багатовходові логічні елементи з інверсією на виході як інвертори, задаючи на всіх, крім одного входу, сигнал 1 (*const 1*) або 0 (*const 0*).

Якщо на всі входи цих елементів подати одну й ту ж змінну, також матимемо інвертори:

$$y = \overline{x_1 \cdot x_1} = \overline{x_1} ,$$

$$y = \overline{x_1 + x_1} = \overline{x_1} . \quad (3.6)$$

Коли кількості входів конкретного логічного елемента не вистачає, можна вчинити так: якщо, наприклад, потрібен тривходовий елемент, а маємо тільки двовходові, то тривходовий можна реалізувати на основі таких співвідношень.

Оскільки очевидно, що

$$\overline{\overline{x}} = x , \quad (3.7)$$

$$\text{то} \quad y = \overline{x_1 \cdot x_2 \cdot x_3} = \overline{\overline{\overline{x_1 \cdot x_2 \cdot x_3}}} = \overline{\overline{x_1 \cdot x_2 \cdot 1 \cdot x_3}} , \quad (3.8)$$

$$\text{або} \quad y = \overline{x_1 + x_2 + x_3} = \overline{\overline{\overline{x_1 + x_2 + x_3}}} = \overline{\overline{x_1 + x_2 + 1 + x_3}} . \quad (3.9)$$

Отже, реалізувати тривходовий елемент можна на трьох двовходових, один з яких повинен працювати як інвертор.

З наведених прикладів видно, що елементи І-НІ чи АБО-НІ дійсно дозволяють реалізувати логічну функцію будь-якої складності.

Нехай потрібно реалізувати функцію

$$y = \overline{(x_1 + x_2 + x_3) \cdot x_4 + x_4 + x_5} . \quad (3.10)$$

У загальному випадку це можна зробити за допомогою схеми, наведеної на рисунку 3.1.

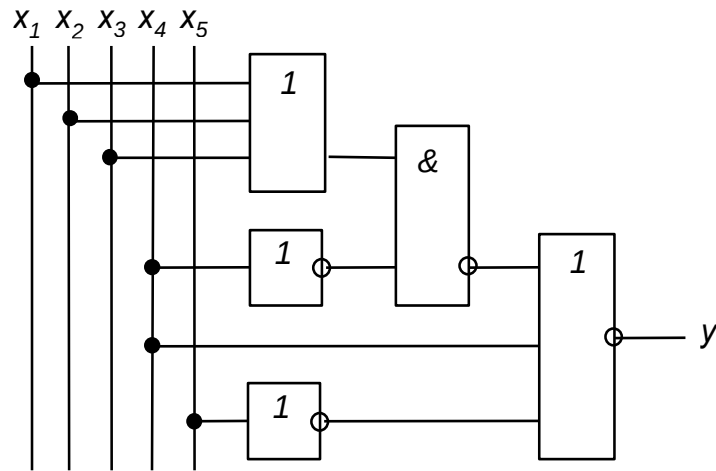


Рисунок 3.1 – Приклад схемної реалізації логічної функції

Для забезпечення отримання простої (економічної в реалізації), швидкодіючої схеми, складні комбінаційні пристрої спочатку описують за допомогою логічних функцій (у вигляді математичних формул або таблиць істинності). Потім ці **функції мінімізують** на основі законів алгебри логіки з урахуванням специфіки стандартних ІМС логічних елементів, що будуть використані для схемної реалізації. Це можна робити як вручну, що достатньо складно, так і з використанням спеціальних програм на ЕОМ.

Так, якщо, наприклад, вираз (3.10) необхідно реалізувати на двовходових елементах І-НІ, то спочатку отримаємо еквівалентний мінімальний вираз в базисі І-НІ

$$\begin{aligned}
 y &= \overline{(x_1 + x_2 + x_3) \cdot x_4 + x_4 + x_5} = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 + x_4 + x_5} = \\
 &= \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_4 \cdot x_5} = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5} = \\
 &= \overline{x_1 \cdot 1 \cdot x_2 \cdot 1 \cdot 1 \cdot x_3 \cdot 1 \cdot x_4 \cdot 1 \cdot x_5 \cdot 1}.
 \end{aligned} \tag{3.11}$$

Отриманому виразу відповідає схема, наведена на рисунку 3.2. Вона виконана на трьох ІМС К561ЛА7.

Зауважимо, що на таких схемах відносно ланцюгів живлення ІМС на вільному полі в правій частині схеми в технічних вимогах дають вказівки щодо підмикання відповідних контактів або вказують відповідні контакти ІМС як нелогічні виводи елементів і позначають місця їх підмикання (див. рис. 3.2 – контакти 7 і 14). На входи елементів, що не задіяні, подають сигнали *const 1* або *const 0*, а їхні виходи залишають вільними (див. рис. 3.2 – елементи DD3.2 і DD3.3).

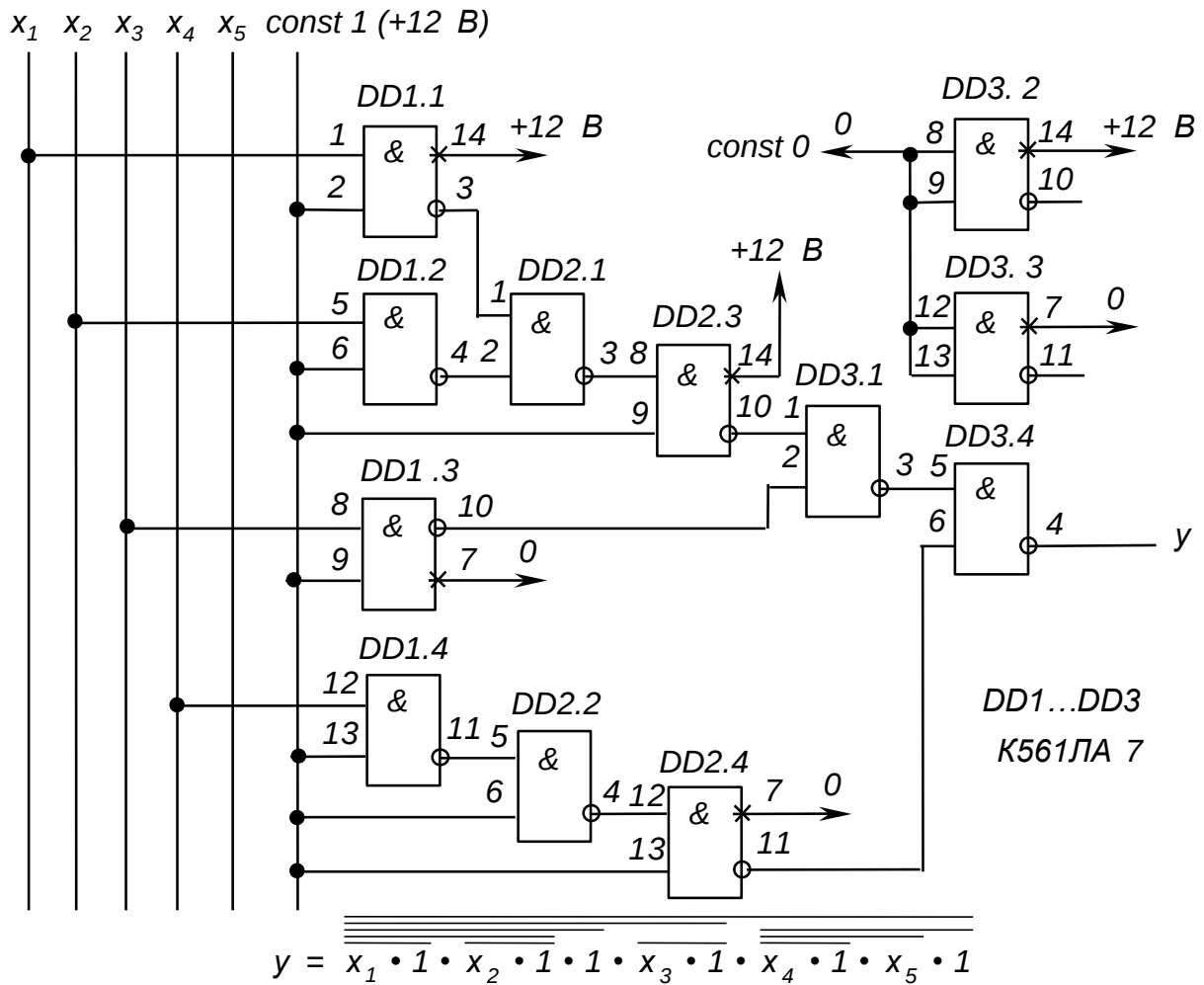


Рисунок 3.2 – Схемна реалізація логічної функції на однотипних елементах

3.3 Дешифратори

Дешифратори (декодери) є комбінаційними пристроями, призначеними для перетворення кодованих двійкових входних сигналів у сигнали керування виконавчими пристроями, пристроями відображення інформації і т. п.

У загальному випадку дешифратор має декілька входів (за кількістю розрядів двійкових чисел, які необхідно декодувати) і декілька виходів.

Кожній комбінації входних сигналів відповідає певна комбінація вихідних (зрозуміло, що дешифратори, як комбінаційні пристрої будуються на логічних елементах і їх випускають у вигляді ІМС).

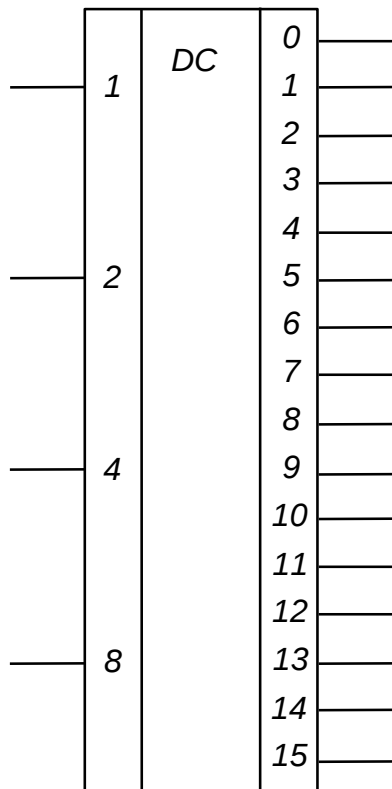


Рисунок 3.3 – Двійковий дешифратор

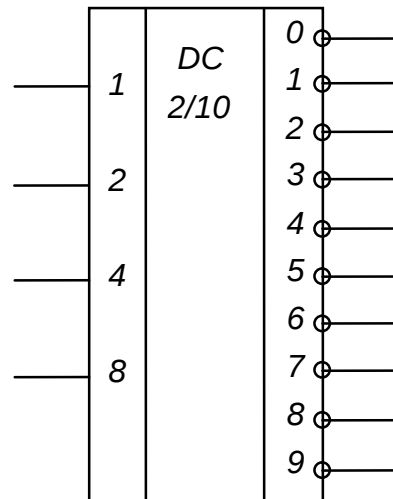


Рисунок 3.4 – Двійково-десятковий дешифратор

Наприклад, **двійковий дешифратор**, умовне позначення якого наведено на рисунку 3.3, має чотири входи ($n=4$) з ваговими коефіцієнтами 1, 2, 4, 8, що відповідає чотирьом розрядам послідовного двійкового коду ($2^0, 2^1, 2^2, 2^3$), і шістнадцять виходів: від 0 до 15 ($N=2^n=2^4=16$). Кожній

комбінації нулів і одиниць на входах відповідає одиниця на відповідному виході.

Наприклад,

$$y_5 = x_1 \cdot \overline{x_2} \cdot x_4 \cdot \overline{x_8}$$

$$(5=1 \cdot 2^0 + 0 \cdot 2^1 + 1 \cdot 2^2 + 0 \cdot 2^3). \quad (3.12)$$

У **двійково-десятькового дешифратора** з інверсними виходами, умовне позначення якого наведено на рисунку 3.4, кожній з перших десяти двійкових

комбінацій (двійково-десятковий код) відповідає нуль на відповідному виході. Такі дешифратори у вигляді ІМС застосовують для керування десятковими неоновими індикаторами, у яких десяткові знаки становлять фігурні катоди неонові лампи.

Двійково-семисегментний дешифратор, зображений на рисунку 3.5,а, перетворює двійкову комбінацію вхідних сигналів у комбінацію вихідних, необхідну для вмикання відповідної комбінації сегментів семисегментного десятикового індикатора – рисунок 3.5,б.

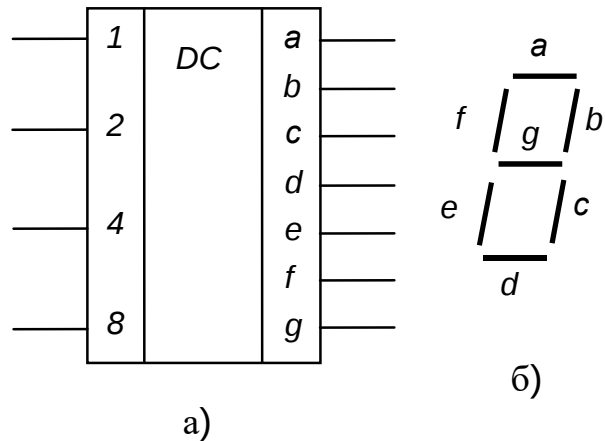


Рисунок 3.5 – Двійково-семисегментний дешифратор (а) і схема розміщення сегментів індикатора (б)

3.4 Мультиплексори

Мультиплексори (комутатори) – це комбінаційні пристрої, які підмикають до виходу вхід (передають на вихід інформацію з входу), номер якого задає комбінація нулів і одиниць на адресних входах.

Схема чотиривходового мультиплексора і його таблиця істинності наведені на рисунку 3.6.

Логічна функція, що її реалізує цей мультиплексор, така:

$$F = A\bar{x}\bar{y} + B\bar{x}y + Cx\bar{y} + Dxy. \quad (3.13)$$

На основі мультиплексорів може бути реалізована будь яка логічна функція, тому їх часто застосовують у складі програмованих ВІС.

Зауважимо, що К-МОН мультиплексори забезпечують комутацію не тільки цифрових, а також і аналогових сигналів.

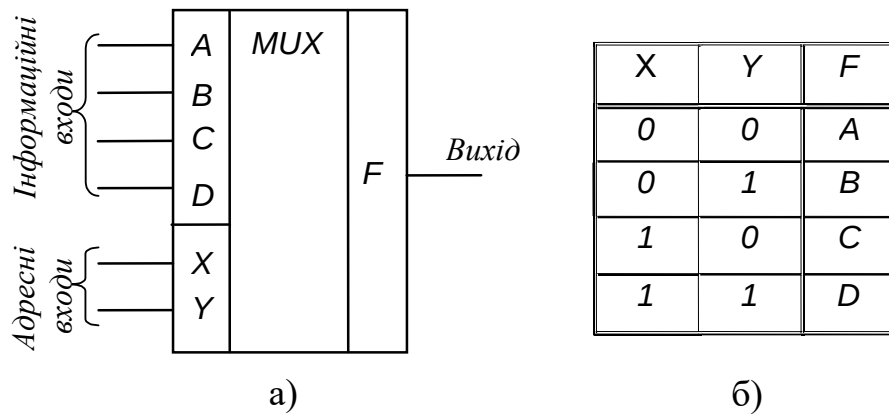


Рисунок 3.6 – Чотиривходовий мультиплексор (а)
і його таблиця істинності (б)

3.5 Лічильники імпульсів

Однією з найбільш розповсюджених операцій у пристроях інформаційно-обчислювальної і цифрової вимірювальної техніки є фіксування кількості імпульсів – підрахунок їхньої кількості. Реалізують таку операцію лічильники імпульсів. Лічильники також забезпечують представлення інформації про кількість імпульсів у вигляді двійкового коду (завдяки принципів побудови).

Лічильники бувають **прості (підсумовуючі)**, у яких код збільшується на одиницю після надходження на вхід кожного імпульсу; **віднімаючі**, у яких код відповідно зменшується після надходження на вхід кожного імпульсу) і **реверсивні** (суміщують властивості підсумовуючих і віднімаючих – можуть працювати в тому або іншому режимі за зовнішньою командою). Лічильники зазвичай будують на основі тригерів.

Схему **чотирирозрядного підсумовуючого послідовного двійкового лічильника**, виконаного на комбінованих *RST*-тригерах з імпульсними інверсними входами наведено на рисунку 3.7, його умовне позначення – на рисунку 3.8, часові діаграми роботи – на рисунку 3.9, таблиця переходів – у таблиці 3.1.

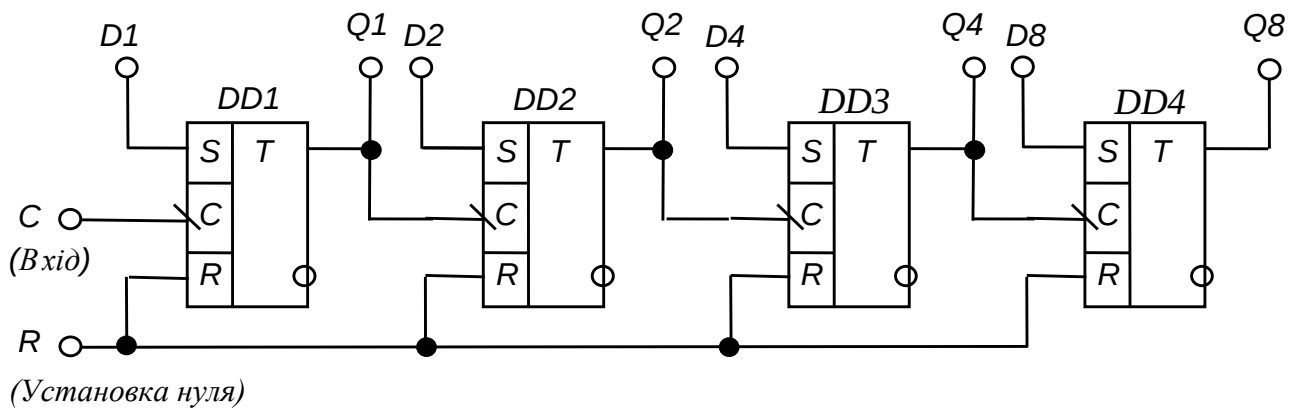


Рисунок 3.7 – Чотирирозрядний послідовний двійковий лічильник

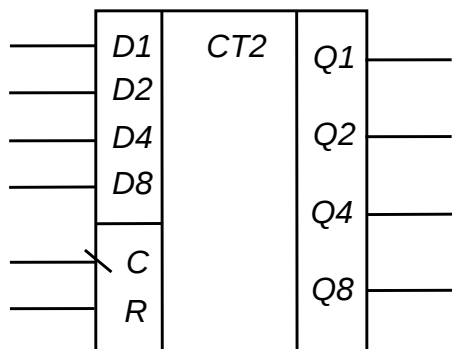


Рисунок 3.8 – Умовне позначення чотирирозрядного послідовного двійкового лічильника

Таблиця 3.1 – Таблиця переходів чотирирозрядного послідовного двійкового лічильника

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

Лічильник називається **послідовним** тому, що вихід тригера кожного попереднього розряду з'єднано з лічильним входом тригера наступного, в результаті чого передача інформації – перемикування тригерів розрядів лічильника – відбувається послідовно одного за одним. Це визначає низьку швидкодію лічильника.

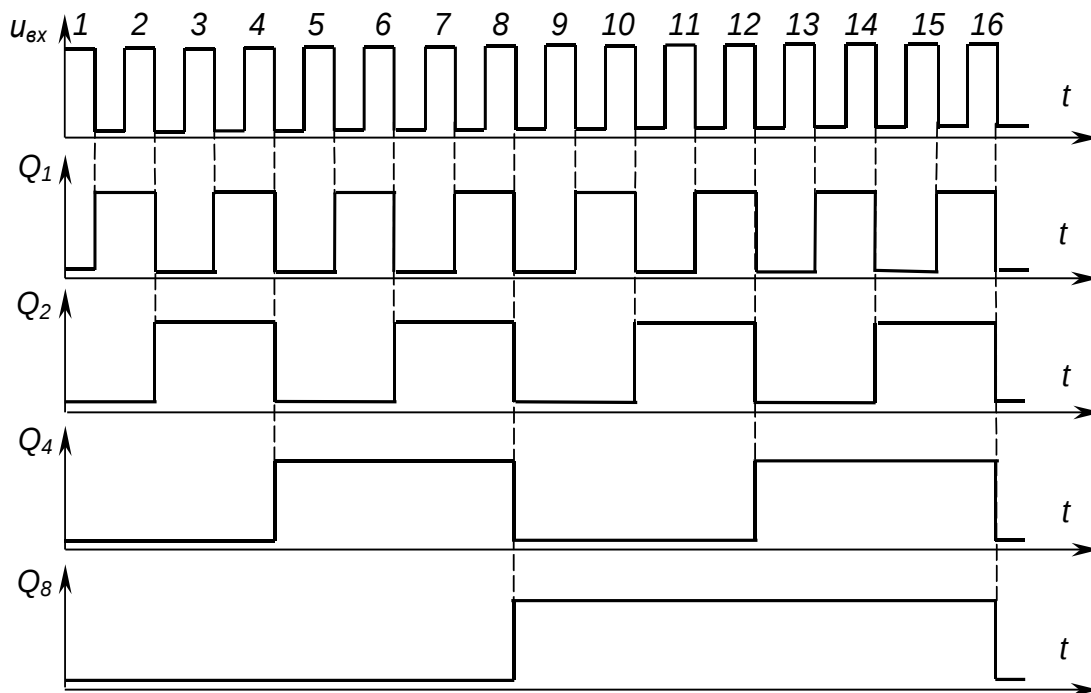


Рисунок 3.9 – Часові діаграми роботи чотирирозрядного підсумовуючого послідовного двійкового лічильника з імпульсним інверсним лічильним входом

У **паралельних** лічильників інформація з розряду в розряд передається за допомогою спеціальної комбінаційної схеми, а входи синхронізації тригерів з'єднано разом і перемикання всіх тригерів відбувається одночасно.

Як видно з таблиці переходів і часових діаграм, при безперервній роботі лічильника на його виходах Q_1 , Q_2 , Q_4 , Q_8 формується **послідовний двійковий код**.

Вхід R призначений для установки лічильника в нульовий стан (у даному разі – подачею сигналу логічної 1).

Входи D_1 , D_2 , D_4 , D_8 призначені для паралельного занесення чисел у лічильник – для завдання початкового стану, що відрізняється від нульового.

Послідовну роботу лічильника можна порушити, обмеживши кількість його станів. Це можна зробити, вводячи зворотний зв'язок, як, наприклад, показано на рисунку 3.10. Тепер при досягненні десятого стану лічильник одразу переходить в нульовий стан – ми отримали **двійково-десятковий лічильник**, який має не 16, а 10 станів і формує на виходах **двійково-**

десятковий код. Його таблиця переходів наведена в таблиці 3.2, а часові діаграми роботи на рисунку 3.11.

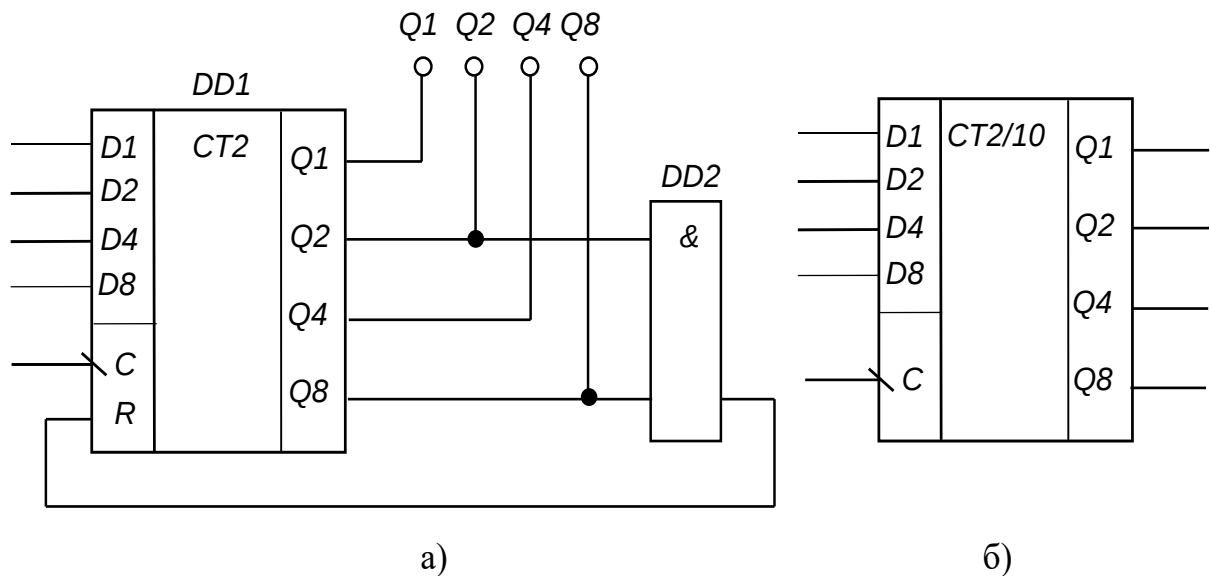


Рисунок 3.10 – Отримання двійково-десятькового лічильника з двійкового (а) і умовне позначення двійково-десятькового лічильника (б)

Таблиця 3.2 – Таблиця переходів чотирирозрядного послідовного двійково-десятькового лічильника

Стан	Q_8	Q_4	Q_2	Q_1
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	0	0	0	0

Застосування таких лічильників разом з двійково-десятьковими або двійково-семи-сегментними дешифраторами дозволяє на основі стандартних ІМС будувати схеми керування багаторозрядними десятковими індикаторами (розряди десяткові, а у межах десяткового розряду рахунок двійковий).

Він має входи:

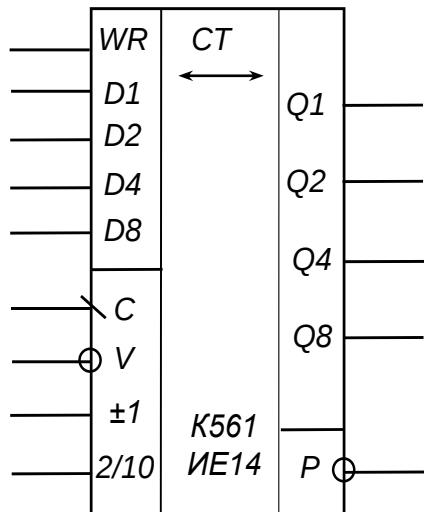


Рисунок 3.13 – Реверсивний лічильник K561IE14

- 1) запису інформації з входів паралельного вводу інформації $D_1, D_2, D_4, D_8 - WR$;
- 2) паралельного вводу інформації – D_1, D_2, D_4, D_8 ;
- 3) синхронізації (тактовий) – C ;
- 4) дозволу лічення – $\bar{V}$;
- 5) напрямку лічення (підсумовування чи віднімання) – ± 1 ;
- 6) завдання роботи в двійковому чи двійково-десятковому коді – $2/10$.

Виходи:

- 1) інформаційні – Q_1, Q_2, Q_4, Q_8 ;
- 2) переносу (займу) в старший (із старшого) розряду – $\bar{P}$.

3.6 Регістри

Регістри призначені для запам'ятовування і зберігання інформації, представленої у вигляді багаторозрядних двійкових чисел (двійкового коду) і їх видачі за зовнішньою командою – це елементи короткочасної (оперативної) пам'яті.

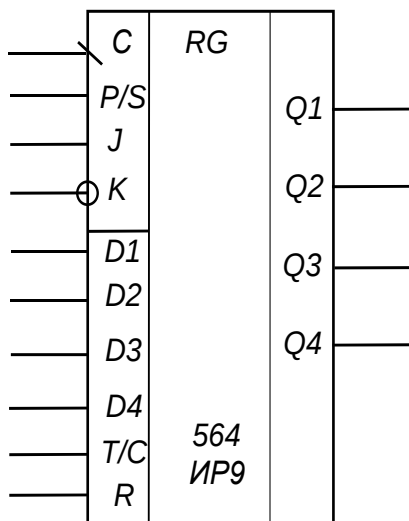
Залежно від способу запису і видачі інформації регістри бувають:

- 1) послідовні – запис інформації в яких виконується послідовно одного двійкового розряду за іншим через один вхід;
- 2) паралельні – запис інформації в яких виконується одночасно (паралельно) у всі розряди;
- 3) послідовно-паралельні – можуть працювати як послідовні або паралельні, залежно від сигналу на спеціальному вході керування.

На рисунку 3.14 як приклад наведено умовне позначення універсального регістру типу 564ІР9.

Він має входи:

- 1) синхронізації (тактовий) – C ;
- 2) завдання режиму роботи (паралельний – послідовний) – P/S ;
- 3) послідовного вводу інформації (входи JK -тригера першого розряду) – $J, \bar{K}$;
- 4) паралельного вводу інформації – D_1, D_2, D_3, D_4 ;
- 5) завдання видачі інформації в прямому або інверсному коді – T/C ;
- 6) установки нульового стану R .



Виходи: Q_1, Q_2, Q_3, Q_4 .

При видачі інформації у послідовному коді, останній знімається з виходу Q_4 .

На завершення варто сказати, що основні схемотехнічні прийоми побудови пристроїв цифрової техніки не змінилися за десятиліття її існування до цих пір.

Рисунок 3.14 – Універсальний
регістр 564ІР9

4 ЦИФРО-АНАЛОГОВІ (ЦАП) І АНАЛОГО-ЦИФРОВІ (АЦП) ПЕРЕТВОРЮВАЧІ

1. Будова і робота ЦАП.

2. АЦП.

ОСНОВНІ ПОНЯТТЯ

Цифро-аналогові перетворювачі (ЦАП); аналого-цифрові перетворювачі (АЦП); опорна напруга; матриця R-2R; метод перетворення; підсилювач з програмованим коефіцієнтом підсилення; перетворювач струму у напругу; ЦАП на основі методу перерозподілу заряду; АЦП послідовного наближення; функціонально закінчений АЦП; АЦП паралельного кодування.

Для узгодження цифрових пристроїв вимірювання і керування, що працюють з інформацією, представленою у двійковому коді, з виконавчими пристроями і датчиками, які мають аналогові відповідно вхідні та вихідні сигнали, застосовують **цифро-аналогові і аналого-цифрові (ЦАП і АЦП) перетворювачі.**

4.1 Будова і робота ЦАП

ІМС ЦАП зазвичай становлять резисторні матриці. Матриця може бути виконана як дільник напруги з співвідношенням опорів кратним 2^n : R-2R-4R-8R-16R і т. д. (застосовують рідко, бо технологічно важко забезпечити точні значення великих опорів резисторів, що відповідають старшим розрядам). Частіше матриця ЦАП становить дільник струмів з співвідношенням опорів R-2R. Також до ІМС, зазвичай, входять транзисторні ключі (наприклад, на польових К-МОН-транзисторах), що забезпечують вмикання потрібної комбінації резисторів за сигналами двійкового коду.

Зокрема, ІМС К572ПА1 (виконана за К-МОН-технологією) призначена для перетворення 10-розрядного прямого паралельного двійкового коду на цифрових входах у струм на аналоговому виході. Її спрощену структуру наведено на рисунку 4.1. Видно, що, окрім власне резисторної матриці R-2R, вона має ключі і вузли керування ними *БК*.

Вхідним аналоговим сигналом ЦАП є **опорна напруга** $U_{оп}$, що визначає значення напруги, яка відповідає молодшому двійковому розряду. Вхідний струм I_0 , що протікає під дією $U_{оп}$, ділиться за двійковим законом у вузлах

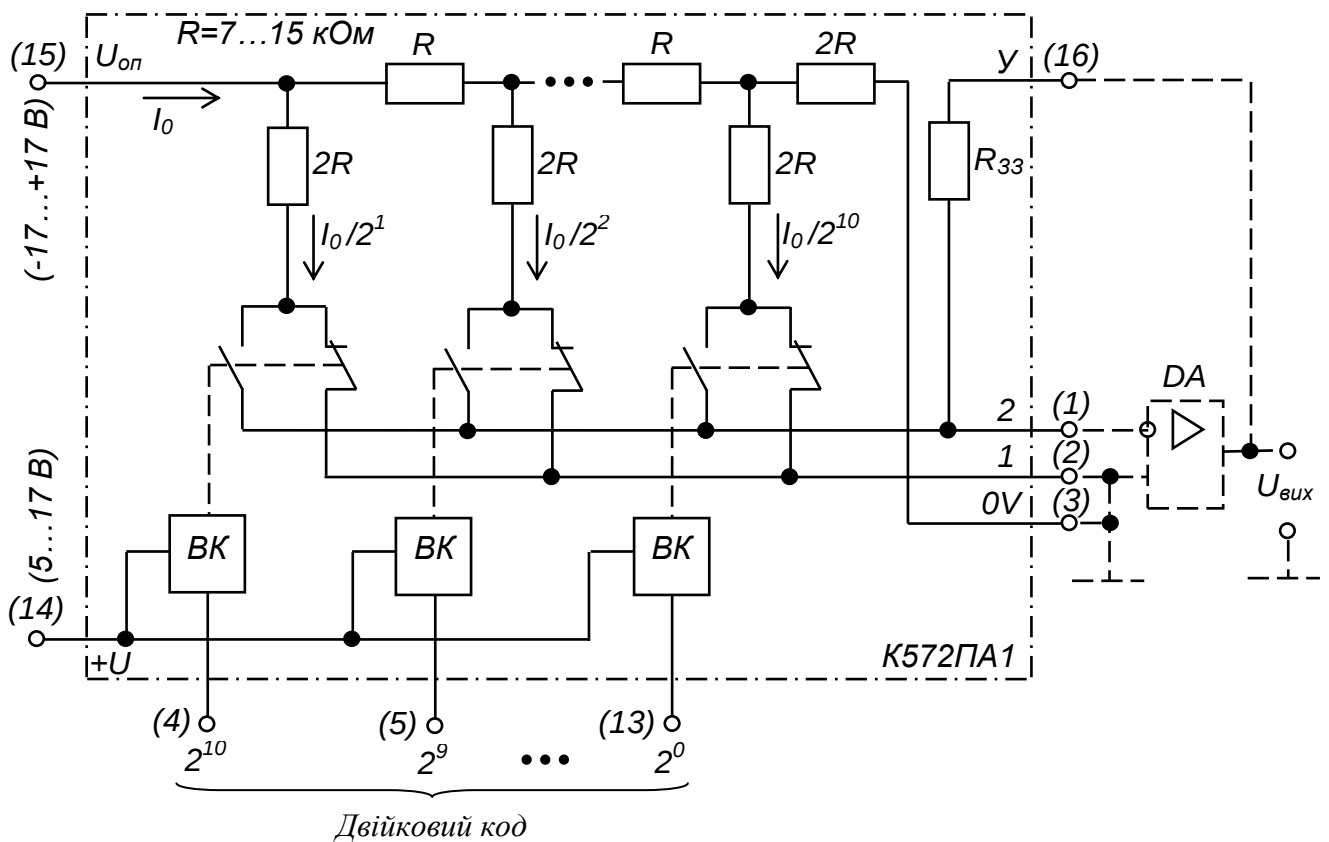


Рисунок 4.1 – Структура ІМС К572ПА1

резисторної **матриці R-2R**. З рисунку 4.2 можна бачити: приведені до місць перетинів *A-A*, *B-B* та *C-C* опори частини резисторів, що відтинаються у напрямку стрілок, дорівнюють $2R$ і тому значення струмів, які протікають через резистори $2R$, зменшуються удвічі зліва на право.

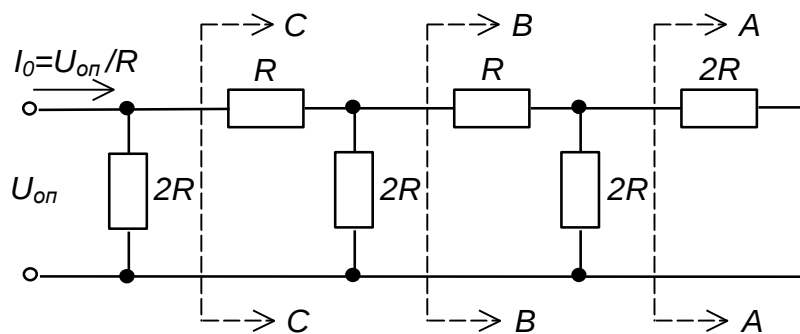


Рисунок 4.2 – Резисторна матриця R-2R

Метод перетворення полягає в підсумовуванні відповідно до заданих значень двійкового коду усіх розрядних струмів, зважених за двійковим законом і пропорційних значенню опорної напруги.

Струми віток резисторної матриці через ключі на К-МОН-транзисторах надходять на два аналогові виходи: на той чи інший – у залежності від значень двійкових розрядів коду на відповідних цифрових входах. Значення струмів складають:

$$\text{для першого виходу} \quad I_1 = \frac{U_{on}}{R_{екв}} \sum_{i=1}^{10} 2^{-i} a_i ; \quad (4.1)$$

$$\text{для другого виходу} \quad I_2 = \frac{U_{on}}{R_{екв}} \sum_{i=1}^{10} 2^{-i} \bar{a}_i , \quad (4.2)$$

де U_{on} – опорна напруга;

$R_{екв}$ – еквівалентний опір резисторної матриці, що має номінальне значення 10 кОм (фактично – від 7 до 15 кОм);

a_i – пряме значення (для $a_i=1$) двійкового коду на i -тому вході;

$\bar{a}_i$ – інверсне значення (для $a_i=0$) двійкового коду на i -тому вході.

Двійковий закон розподілення струмів у вітках матриці виконується за умови однаковості потенціалів обох її виходів. Це забезпечується підмиканням першого виходу до інвертуючого входу допоміжного ОП, охопленого ВЗЗ, а другого – до неінвертуючого входу ОП і нульової точки схеми (як відомо, потенціали інвертуючого і неінвертуючого входів ОП, охопленого ВЗЗ, за умови роботи у лінійному режимі, майже однакові).

Отже, матриця $R-2R$ підмикається до ОП, утворюючи з ним інвертуючий підсилювач з програмованим коефіцієнтом підсилення. Його можна розглядати також як перетворювач струму, що надходить з виходу резисторної матриці, у напругу на виході ОП. У результаті, кожному значенню двійкового коду на цифрових входах ЦАП відповідає деяке значення напруги на виході ОП, оскільки, забезпечуючи різницю потенціалів між своїми входами близькою до нуля, ОП відводить через резистор зворотного зв'язку R_{33} сумарний струм, що надходить до виходу (1) ІМС. Цей струм складається зі струмів резисторів $2R$, підімкнених ключами до виходу (1) відповідно до значення двійкового коду і визначається розрядами $a_i = 1$.

Варто зазначити, що за практичної реалізації подібних пристроїв, задля зменшення взаємних впливів, що можуть призводити до збоїв у роботі, так звані «цифрову землю» і «аналогову землю» (обидві відповідають нульовому потенціалу схеми) потрібно виконувати у вигляді окремих провідників.

Аналітичний вираз, що зв'язує значення напруги на виході ОП зі значенням двійкового коду на цифрових входах ЦАП, такий

$$U_{вих} = U_{он} R_{33} (2^{b-1} a_1 + 2^{b-2} a_2 + \dots + 2^{b-i} a_i + 2^0 a_b) / 2^b R, \quad (4.3)$$

де R_{33} – опір резистора зворотного зв'язку;

b – кількість розрядів перетворювача (у цьому разі – 10), причому старшим є розряд з індексом b .

Значення вихідної напруги в кінцевій точці діапазону становить

$$U_{вих\ макс}^* = U_{он} R_{33} (1 - 2^{-b}) / R, \quad (4.4)$$

а найменше при усіх $a_i = 0$ – дорівнює нулеві.

Розрахунковий приріст вихідної напруги при зміні вхідного коду на одиницю молодшого розряду (крок квантування) складає

$$h = U_{он} R_{33} / 2^b R. \quad (4.5)$$

Реально найбільше значення вихідної напруги при значеннях усіх розрядів двійкового коду $a_i = 1$ становить

$$U_{\text{вих макс}} = \frac{U_{\text{он}}}{R_{33}} - h, \quad (4.6)$$

бо десятирозрядний ЦАП працює з числами, що у десятковій системі числення відповідають від 0 до 1023.

Для забезпечення стабільності роботи задають $R_{33} = R$, а сам резистор R_{33} розміщують на кристалі ІМС разом з матрицею R-2R.

Електричну принципову схему цифро-аналогового перетворювача, виконаного на основі ІМС К572ПА1, наведено на рисунку 4.3.

Якщо задати $U_{\text{он}} = 10,24 \text{ В}$, то отримаємо $U_{\text{вих макс}}^* = 10,24 \text{ В}$, а $h = 10 \text{ мВ}$

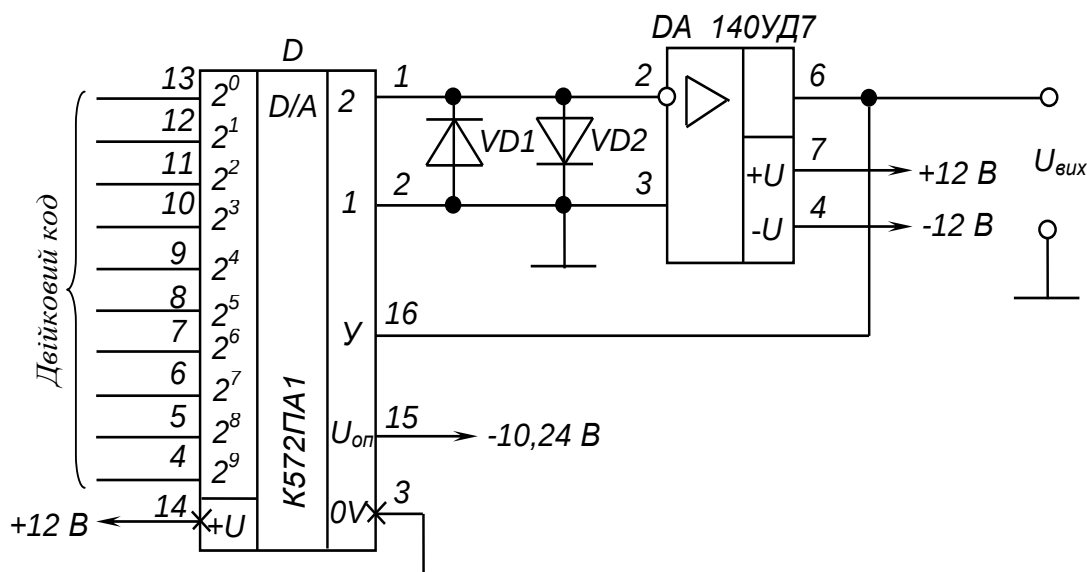


Рисунок 4.3 – Цифро-аналоговий перетворювач на основі ІМС К572ПА1

за номінального значення вхідного струму матриці 1 мА (фактично – від 0,5 до 2 мА).

Точнісні показники перетворювача залежать від точності виконання співвідношень $R_{33}/R = 1$ та $R/2R = 0,5$ для усіх ланок матриці.

Якщо формулу (4.3) переписати, вважаючи

$$R_{33}/2bR = k,$$

$$(2^{b-1} a_1 + 2^{b-2} a_2 + \dots + 2^{b-i} a_i + 2^0 a_b) = X,$$

отримаємо:
$$U_{\text{вх}} = k U_{\text{он}} X, \quad (4.7)$$

звідки видно, що цей ЦАП може бути використано також як помножувач аналогової величини $U_{\text{он}}$ (що може приймати значення від мінус 17 до +17 В) на двійкове число X .

Більш точними є **ЦАП на основі методу перерозподілу заряду**. Вони будуються на основі конденсаторної матриці з співвідношенням ємностей кратним 2^n .

Такий ЦАП має три етапи роботи.

На першому етапі ОП, що також входить до його складу, працює як повторювач, а усі конденсатори матриці підімкнено до вхідного контакту пристрою і накопичують заряд, пропорційний вхідній напрузі.

На другому етапі схема керування перемикає конденсатори від входу на нульовий провідник. Тепер на вхід ОП, що працює вже як нуль-орган, подається напруга, що дорівнює вхідній $U_{\text{вх}}$ але з протилежним знаком.

Ну і на третьому етапі розпочинається перерозподіл заряду, коли до джерела опорної напруги $U_{\text{он}}$ підмикається спочатку конденсатор старшого розряду матриці. На вході нуль-органу отримуємо напругу, що дорівнює $(0,5 U_{\text{он}} - U_{\text{вх}})$. Схема керування, залежно від того, змінився стан на виході нуль-органу чи ні, залишає конденсатор у попередньому стані або знову перемикає його на нульовий провідник. Надалі те ж саме робиться з іншими конденсаторами, доки напруга на вході нуль-органу не знизиться до нуля і зміниться стан на його виході. Тоді відношення сумарної ємності, підімкненої до нульового провідника, до ємності, що була підімкнена до $U_{\text{он}}$, буде еквівалентна відношенню $U_{\text{вх}}/U_{\text{он}}$.

Підвищена точність ЦАП забезпечується тим, що пристрій час від часу переводиться у режим самоналаштування. У цьому режимі до кожного з конденсаторів матриці підмикаються допоміжні конденсатори малої ємності доти, доки сумарна ємність не стане більшою за суму ємностей конденсаторів

молодших розрядів на ємність конденсатора самого молодшого розряду. При цьому точність налаштування складає 0,25 ємності цього розряду.

За енергетичних застосувань, коли важливим є не власне значення напруги на навантаженні, а значення енергії, що передається до нього, як виконавчий пристрій ЦАП застосовують широтно-імпульсний регулятор.

4.2 АЦП

Світ, що нас оточує, є аналоговим. До аналогових відносять процеси, що змінюються за законом безперервної функції. Так, звуки та зображення надходять до наших органів відчуття у вигляді коливань – звукових або електромагнітних. Ці коливання сприймаються органами відчуттів (слух, зір) і у вигляді імпульсів передаються до мозку. Але інформація, що передається аналоговим способом, легко викривляється у тракті передачі та потребує величезних об'ємів пам'яті при використанні в техніці. Спосіб «оцифровування» інформації спрощує процеси передачі і обробки інформації. Оцифровування – це процес перетворення аналогової інформації в цифрову. До цифрових відносять процеси, що змінюються за законом дискретної функції – зазвичай, це двійковий цифровий код. Техніка, що працює з такою інформацією називається цифровою. Цифрова інформація легко контролюється, дає стабільну і регульовану якість обробки і представлення процесів. Вона потребує менших об'ємів пам'яті для зберігання. Отже, для світу техніки цифрова інформація підходить набагато більше, ніж аналогова.

Процес оцифровування аналогової інформації проходить два основні етапи. На першому аналогова інформація розбивається на невеликі рівні частини. На другому етапі кожна частина аналізується і зашифровується спеціальними алгоритмами у коди з послідовності одиниць і нулів.

АЦП може бути побудовано на основі ЦАП, лічильника імпульсів і компаратора. Спрощену структурну схему такого АЦП наведено на рисунку 4.4.

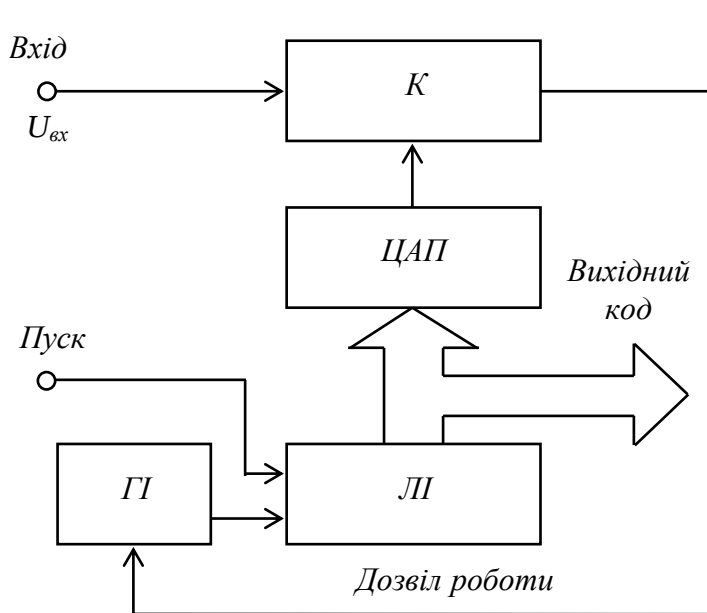


Рисунок 4.4 – Структурна схема АЦП

Цикл перетворення аналогового сигналу, представленого як напруга $U_{вх}$, у двійковий код складається з таких операцій.

Напруга $U_{вх}$ подається на вхід пристрою – один з входів компаратора K . Сигнал з виходу компаратора дозволяє роботу генератора імпульсів $ГІ$.

Сигнал $Пуск$ встановлює нульовий стан і дозволяє роботу

лічильника імпульсів $ЛІ$, що починає заповнюватись імпульсами $ГІ$.

Код з виходу $ЛІ$ подається на цифрові входи $ЦАП$ (входи керування ключами). У результаті з виходу $ЦАП$ ступінчасто зростаюча напруга надходить на другий вхід компаратора. Після досягнення цієї напругою значення $U_{вх}$ компаратор забороняє роботу генератора, а на виході $ЛІ$ маємо прямий паралельний двійковий код, що відповідає значенню $U_{вх}$.

Реально, для збільшення швидкості перетворення, застосовують більш складний алгоритм. До входів $ЦАП$ замість лічильника підмикається регістр пам'яті. Після пуску схема керування встановлює регістр у стан, якому відповідає одиниця у старшому розряді і нулі у всіх інших. При цьому на виході $ЦАП$ формується напруга, що дорівнює половині діапазону перетворення. Якщо вона менша за $U_{вх}$, то в старший розряд регістра записується нуль, а у другий за старшинством одиниця. Це відповідає напрузі на виході $ЦАП$, рівній половині попередньої. У разі перевищення $U_{вх}$ цієї напруги одиниця встановлюється у третьому за старшинством розряді і вихідна напруга $ЦАП$ збільшується у 1,5 рази. Описана процедура повторюється доти, доки на виході $ЦАП$ не сформується напруга, що відрізняється від $U_{вх}$ не більш ніж на ту, що відповідає одиниці молодшого розряду $ЦАП$.

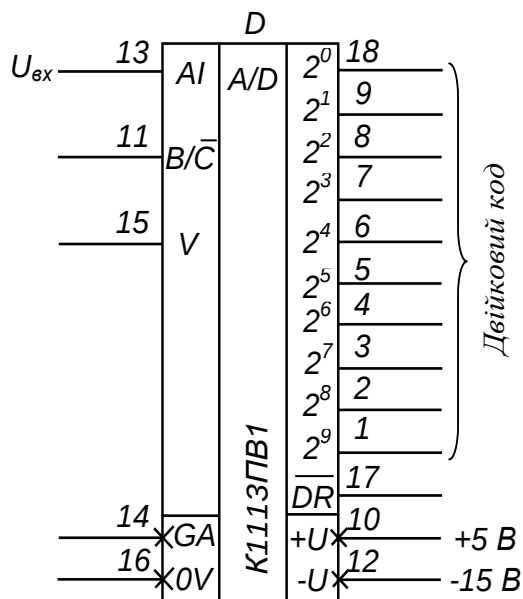


Рисунок 4.5 – ІМС АЦП К1113ПВ1

Перетворювачі, що працюють за таким алгоритмом, називають **АЦП послідовного наближення**.

На рисунку 4.5 наведено умовне позначення ІМС **функціонально закінченого** (такого, що не вимагає використання допоміжних елементів)

АЦП послідовного наближення К1113ПВ1. Він призначений для роботи з мікропроцесорними

пристроями і є сумісним з мікропроцесорами, що працюють з

ТТЛ-рівнями сигналів. Його вихідні вузли мають три стани: два логічних (0 та 1) і третій – стан високого імпедансу – стан відімкнення, що забезпечує просте спраження з шиною даних мікропроцесора.

Режим роботи АЦП у мікропроцесорній системі визначається сигналами керування від мікропроцесора.

При надходженні на вхід гасіння і перетворення $B/\overline{C}$ логічного нуля, АЦП починає цикл перетворення вхідної напруги $U_{вх}$ у двійковий код.

Після завершення перетворення на виході готовності даних $\overline{DR}$ з'являється сигнал логічної 1, що є запитом для мікропроцесора на прийом коду. У вихідному стані і стані перетворення на цьому виході утримується сигнал 1.

Після прийому коду мікропроцесор подає на вхід $B/\overline{C}$ сигнал логічної 1, що встановлює АЦП у вихідний стан, після чого він готовий до нового циклу перетворення.

Цей АЦП може обробляти вхідну інформацію у вигляді однополярної напруги до 10,24 В, або двополярної $\pm 5,12$ В. У однополярному режимі на вхід

керування зсувом нуля V потрібно подати сигнал логічного 0, а у двополярному – залишити його вільним.

Точності перетворення $\pm 1/2$ одиниці молодшого розряду цифрового коду досягають вмиканням між виводом «аналогової землі» GA і корпусом OV резистора опором від 5 до 50 Ом (використовують резистор змінного опору).

На завершення варто зазначити, що елементарним пристроєм перетворення аналогової величини у дискретну є компаратор, який фіксує факт перевищення однієї напруги іншою і може мати на виході сигнали, що відповідають логічним 0 або 1. При цьому найбільш швидкодіючим є побудований на основі лінійки компараторів АЦП паралельного кодування. Напруга $U_{вх}$ тут подається на перші входи всіх компараторів одразу. На другі їхні входи подається напруга з дільника, що складається з резисторів одного номіналу. Таким чином значення напруги, з яким відбувається порівняння $U_{вх}$ двома сусідніми компараторами, відрізняється на напругу, що відповідає молодшому розряду. Схема шифрування формує вихідний код, який відповідає старшому компаратору з тих, що спрацювали. Цей код записується в вихідний регістр пам'яті. Але апаратні затрати цього методу надзвичайно великі. Наприклад, для восьмирозрядного АЦП потрібно 255 компараторів і близько 3×10^4 активних компонентів (транзисторів).

5 СХЕМОТЕХНІКА ПРОГРАМОВАНИХ ЦИФРОВИХ ПРИСТРОЇВ. МІКРОПРОЦЕСОРНІ ПРИСТРОЇ

- 1. Загальні положення. Дещо з історії процесорів.**
- 2. Особливості роботи і використання мікропроцесорів. Двійкова система числення. Структура мікропроцесорної системи.**
- 3. Структура мікропроцесора.**

ОСНОВНІ ПОНЯТТЯ

Алгоритм; жорстка логіка; програмована логіка; мікропроцесор (МП); структура процесора; арифметико-логічний пристрій (АЛП); пристрій керування (ПК); програма; пристрій вводу і виводу (ПВВ); запам'ятовувальний пристрій (ЗП); постійний запам'ятовувальний пристрій (ПЗП); оперативний запам'ятовувальний пристрій (ОЗП); десяткова система числення; двійкова система числення; мікропроцесорний пристрій (МПП); інтерфейс; засоби сполучення (ЗС); магістральна схема; шина адреси (ША), шина даних (ШД); шина керування (ШК); логічний стан; третій (нелогічний) стан; структура МП; генератор тактових імпульсів (ГТІ); регістри; акумулятор; регістр стану; лічильник команд; регістр адреси пам'яті; регістр команд; робочі регістри; буферні регістри інтерфейсу; вихідні підсилювачі; пристрій керування.

5.1 Загальні положення. Дещо з історії процесорів

З розвитком технології і схемотехніки ІМС виникла можливість створення на одному кристалі цілого електронного пристрою, наприклад, радіоприймача, вимірювального приладу чи пристрою керування. Але, зрозуміло, що при цьому ІМС (ВІС) стають вузько спеціалізованими. І якщо у випадку застосування пристроїв масового призначення, що продукуються десятками тисяч, (радіоприймач, вимірювальний прилад) з цим можна змиритися, то з пристроями керування виникають великі проблеми.

Перш за все, **алгоритм** (послідовність виконання дій, що веде до успіху) керування часто є індивідуальним (наприклад, для кожного з автоматичних маніпуляторів, що працюють на складальному конвеєрі з виробництва автомобілів). Крім того, у процесі експлуатації технологічного устаткування часто виникає необхідність у заміні всього алгоритму керування або його частин (добавлення або вилучення операцій, зміна послідовності їхнього виконання). Застосування спеціалізованих ВІС у такому разі з економічної точки зору недоцільне.

Але й побудова пристроїв керування на основі універсальних ІМС середнього ступеня інтеграції, хоча це й менше коштує, теж не є вдалим виходом, бо зміни алгоритму при цьому ведуть до зміни електричної схеми пристрою керування (повністю або у деяких її частинах).

До речі, пристрої керування з незмінним алгоритмом називають пристроями з **жорсткою логікою**.

Усунути протиріччя між ступенем складності ВІС і можливістю її універсального використання вдалося шляхом застосування програмування виконуваних функцій (у найпростішому випадку це, наприклад, програмування коефіцієнта перерахунку лічильника імпульсів).

Зараз основним типом програмованих ВІС, що мають найбільшу інформаційну потужність, є мікропроцесор.

Мікропроцесор (МП) – це програмно керований пристрій обробки цифрової інформації, виконаний у вигляді однієї (рідше декількох) ВІС. За функціями МП аналогічний центральному процесору цифрової ЕОМ. Його можна вважати одним з найбільших досягнень сучасної мікроелектроніки.

Тут варто зазначити, що спроби автоматизувати обчислювальні роботи робились дуже здавна і займалися цим найвидатніші мислителі і дослідники. Але рівень техніки і технології не дозволяв створити відповідні пристрої. І тільки з середніх віків починаються зрушення у цій справі.

Наприклад, у 1642 році фізик Б. Паскаль побудував механічну машину, що виконувала операції додавання та віднімання (надалі було застосовано до

п'ятдесяти таких машин). Подібні пристрої розробляли німецький математик Г. В. Лейбніц і російський математик П. Л. Чебишев (орієнтовані на розв'язання рівнянь вищої математики), а пізніше російський інженер В. Т. Орднер. «Колесо Орднера» стало основою арифмометрів. Далі були електромеханічні пристрої – на електромагнітних реле – та, нарешті, електронні – спочатку на лампах, а потім на транзисторах та ІМС.

Найбільш близьким прообразом сучасних цифрових ЕОМ є механічна “аналітична машина” англійського математика Ч. Беббіджа, запропонована ним у 1833 році для автоматизації розрахунку різного роду таблиць (навігаційних, артилерійських, технологічних – подібних до таблиць Брадїса) з метою виключення суб'єктивних помилок розрахівників. Це був надзвичайно складний пристрій, що складався з чотирьох тисяч деталей і важив три тони. Пізніше вона була реалізована його сином (додавання двох чисел тривало 2 секунди, а множення – хвилини). У цій машині вперше було запропоновано і реалізовано ідею **процесора**, що почергово виконує у заданій послідовності наперед задані математичні операції над числами. **Структура процесора** Ч. Беббіджа, наведена на рисунку 5.1, не зазнала суттєвих змін і до нашого часу.

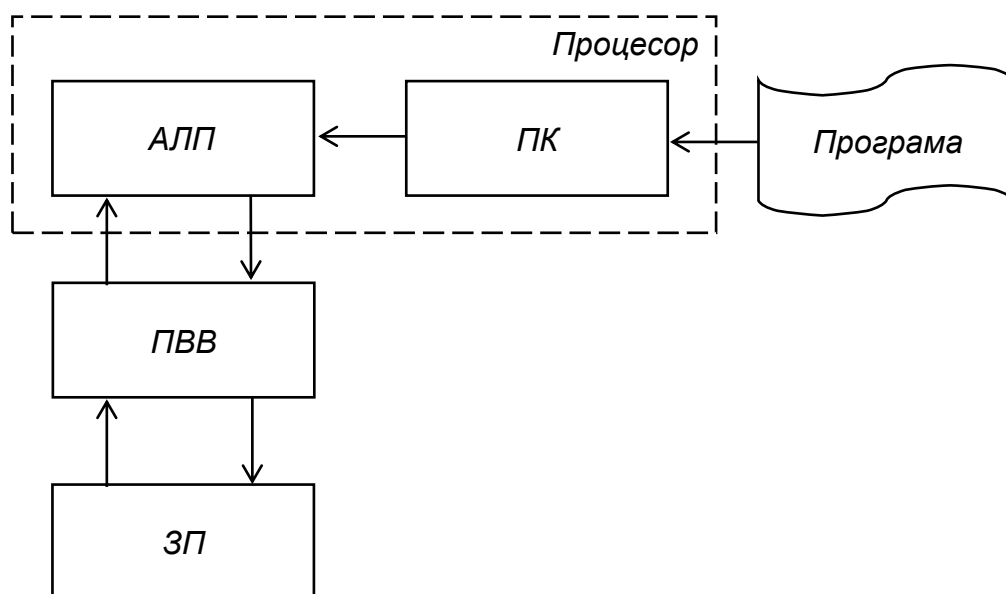


Рисунок 5.1 – Структура процесора

Арифметико-логічний пристрій АЛП тут виконує арифметичні або логічні дії над числами (кодами) у порядку, що задається **пристроєм керування** ПК згідно з **програмою**, яка вводиться до нього (наприклад, з перфоративної стрічки).

Практичне застосування процесора можливе за наявності **пристрою вводу і виводу** чисел ПБВ і **запам'ятовувального пристрою** ЗП, необхідних для вводу вихідних даних, виводу результатів та збереження проміжних результатів.

Повну механічну обчислювальну машину, що мала процесор, ПБВ і ЗП, побудовано 1936 року у Німеччині під керівництвом К. Цузе. Тоді ж з'являється й перша алгоритмічна мова для написання програм для цієї машини. З цього часу розпочинається стрімкий розвиток процесорів, як основної ланки обчислювальної машини.

Зокрема, 1938 року також під керівництвом К. Цузе створено процесор на електромагнітних реле, що збільшило швидкість виконання операцій у десятки разів.

1946 року у США побудовано першу електронну обчислювальну машину ENIAC, що містила 18000 електронних ламп (додавання і віднімання тривало 200 мікросекунд, а множення 2800 мікросекунд).

Одну з перших радянських ЕОМ, призначену для серійного виробництва, «Стрела» було створено у 1953 році. Вона містила 6 000 ламп, споживала потужність 150 кВА і розміщувалася у приміщенні площею 300 квадратних метрів: уніфіковані її комірки (логічні елементи, тригери та ін.) містилися у 6 вертикальних стійках висотою 2,5 метри та у пульті оператора. Вона виконувала 2 000 операцій за секунду. Її постійно обслуговували 5–7 техніків.

Вже у 1955 році з'являються транзисторні процесори, а з початку 1960-х років – виконані на ІМС середнього ступеня інтеграції (процесор виконувався десь на п'ятдесяти ІМС).

Для появи мікропроцесора (процесора у мікрвиконанні – у вигляді ІМС) необхідна була не тільки технологічна база, що вже з'явилася на кінець 1960-х

років, а й усвідомлення, розуміння того, що програмно керована логіка більш ефективна і простіша у виконанні при реалізації складних функцій, ніж схеми з жорсткою логікою.

Наприклад, у 1969 році співробітнику фірми «Intel» (США) М. Хоффу надійшла пропозиція від японської фірми «Nippon Calculating Machine Corporation» розробити комплект ІМС для виконання усіх математичних дій і розрахунку усіх стандартних функцій універсальним калькулятором з роздруковуванням результатів BASICOM. Передбачалося створення дванадцяти ІМС на 2 000 транзисторів кожна, що мали б виконувати відповідні групи з задуманих функцій. Однак М. Хофф запропонував використати лише одну універсальну ІМС з програмним керуванням, що виконувала б усі задані функції, а за необхідності й інші.

Єдиним недоліком було збільшення часу обчислень, що незворотно за програмної реалізації (порівняно з жорсткою логікою). Але для калькулятора це було несуттєво: збільшення тривалості обчислень з десятків мікросекунд до десятків мілісекунд користувачем не помічалось.

Отже, 1971 року фірма «Intel» створила мікропроцесор INTEL 4004, що виконував операції з чотирирозрядними двійковими числами і вміщував 2 250 транзисторів, мав тактову частоту 108 кГц та коштував 200 доларів.

З'явившись у мікрокалькуляторах, мікропроцесори (**програмна логіка**) стали застосовуватись у найрізноманітніших обчислювальних і керуючих пристроях, витісняючи схеми з жорсткою логікою.

5.2 Особливості роботи і використання мікропроцесорів. Двійкова система числення. Структура мікропроцесорної системи

Особливістю сучасних процесорів є те, що вони працюють з числами, представленими не у **десятковій**, а у **двійковій системі числення**: число

представляється не як сума ступенів числа 10, помножених на відповідні числа від 0 до 9

$$N_{10} = a_1 \cdot 10^0 + a_2 \cdot 10^1 + a_3 \cdot 10^2 + \dots + a_{n+1} \cdot 10^n, \quad (5.1)$$

де $a_i = \{0;1;2;3;\dots;9\}$,

а як сума ступенів числа 2, помножених на 0 або 1

$$N_2 = a_1 \cdot 2^0 + a_2 \cdot 2^1 + a_3 \cdot 2^2 + \dots + a_{n+1} \cdot 2^n, \quad (5.2)$$

де $a_i = \{0;1\}$.

У результаті основою такого процесора є двостабільний (такий, що має два стани), а не десятистабільний елемент.

Прикладом двостабільного елемента, призначеного для запам'ятовування інформації є тригер, а десятистабільного – може бути двійково-десятковий лічильник. Двійково-десятковий лічильник, що має 10 станів, побудовано на чотирьох тригерах. В той же час на чотирьох тригерах може бути побудовано пристрій (наприклад, також лічильник або регістр), що має 16 станів. Звідси видно, чому за основу взято двійкову систему числення. Її застосування при побудові цифрових електронних засобів з обробки інформації дозволяє уникнути зниження їхніх потенційних можливостей більш ніж на третину. Отже двійкова система при технічній реалізації є більш економною з точки зору апаратних затрат.

У 1946 році ВМС США замовили двом університетам – Гарвардському і Принстонському – розробку цифрової обчислювальної машини для систем керування артилерією лінкорів. Гарвардська структура мала два пристрої пам'яті: програм і даних. Це забезпечувало високу швидкодію, але вимагало використання двох трактів передачі інформації. Оскільки такі машини тоді будували на електронно-вакуумних лампах, то введення до їхнього складу будь-якого вузла вело до значного збільшення габаритів і зменшення надійності всієї системи. Отже, перемогла прінстонська структура – структура фон Неймана, за якої дані і програма зберігаються в одному і тому ж ЗП і для обміну даними між АЛП й ЗП та між ПК й ЗП використовуються однакові

процедури. Після звернення МП до ЗП перший код вважається кодом команди, а наступні – адресами, даними чи командами у контексті виконання першої команди. Структуру фон Неймана застосовують при побудові і сучасних універсальних процесорів. Але зараз, особливо в мікроконтролерах, про які піде мова дещо пізніше, застосовують і гарвардську структуру, що значно підвищує їхню швидкодію. А надійність систем визначається дуже високою надійністю ВІС – надійність мікросхеми значно вища за надійність навіть однієї лампи.

На основі МП виконуються **мікропроцесорні пристрої** (МПП), перш за все мікроЕОМ – пристрій, що містить МП, запам'ятовувальні пристрої, органи керування і засоби зв'язку з зовнішніми пристроями – інтерфейс.

Інтерфейс (англ. «*interface*» – засіб спряження, сполучення) є сукупністю уніфікованих технічних і програмних засобів, необхідних для підключення зовнішніх пристроїв. Він забезпечує перетворення сигналів МП у сигнали, що сприймаються зовнішніми пристроями і навпаки, підсилення сигналів та становить апаратні засоби і набір програм передачі даних (уніфікований протокол обміну інформацією).

Якщо мікроЕОМ призначена для керування деяким об'єктом, то вона доповнюється засобами сполучення (узгодження) з об'єктом (датчики, аналого-цифрові і цифро-аналогові перетворювачі, виконавчі пристрої і т. п.). Сукупність мікроЕОМ і засобів сполучення називають **мікропроцесорною системою**. Структуру такої системи приведено на рисунку 5.2.

МП є мозком МПП, у якому відбувається обробка інформації (команд і даних), що представляються у двійковому коді.

МП може виконувати дуже складні завдання з обчислень і керування, уміючи виконувати лише елементарні логічні й арифметичні операції, операції пересилання даних, порівняння двох чисел і деякі інші, шляхом багаторазового їхнього повторення відповідно до заданої користувачем програми (визначеної послідовності команд).

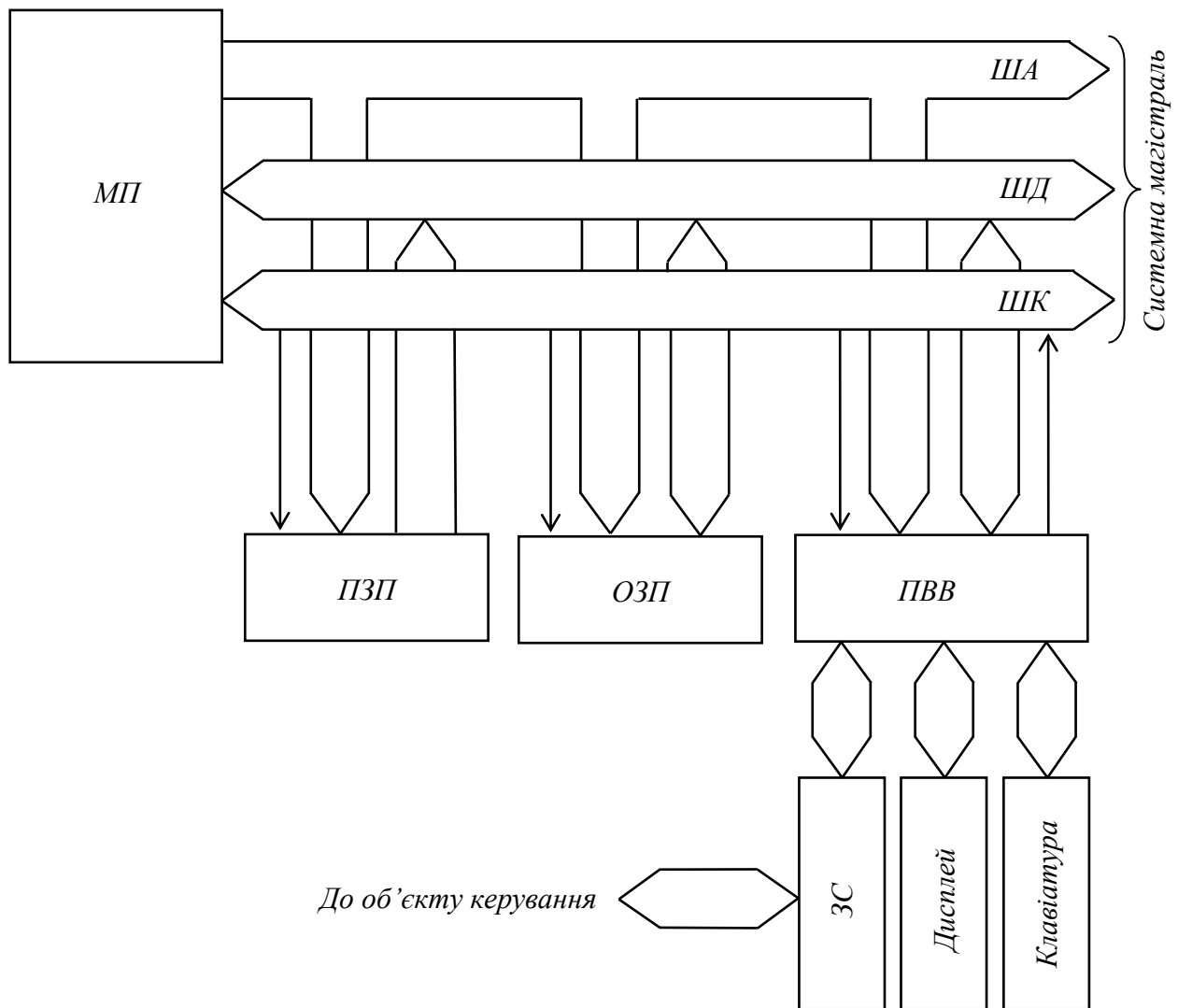


Рисунок 5.2 – Мікропроцесорна система

Для забезпечення виконання операцій МП містить пристрої вибірки інформації з пам'яті і її дешифрування, арифметико-логічний пристрій, що представляє сукупність схем, що реалізують арифметичні і логічні операції над даними, пристрій керування, який забезпечує виконання операцій МП, різні регістри, що слугують для тимчасового зберігання (надоперативний ЗП) і перетворення даних і команд, тактовий генератор, що задає темп роботи МП.

Для зберігання інформації є два види ЗП: **постійний запам'ятовувальний пристрій ПЗП** та **оперативний запам'ятовувальний пристрій ОЗП**.

ПЗП призначений для зберігання інформації, що заноситься в нього при виготовленні МПП і зберігається при відключенні живлення. Завдяки цьому МПП після вмикання знає, як підготувати себе до роботи і що потрібно робити при одержанні тих чи інших зовнішніх команд і дій. ПЗП може працювати тільки у режимі видачі інформації. Вміст ПЗП можна змінити заміною його ІМС на інші з новим набором програм або перепрограмуванням ІМС, якщо вони це допускають.

ОЗП призначений для тимчасового зберігання даних і програм користувача, проміжних результатів роботи і працює як у режимі запису, так і у режимі видачі інформації. Якщо інформацію, що міститься в ОЗП, необхідно зберегти довгостроково, то її потрібно записати у зовнішній пристрій пам'яті (наприклад, на магнітофон або жорсткий диск).

ПВВ забезпечує зв'язок МП із зовнішніми пристроями – дисплеєм, клавіатурою (засоби сполучення з оператором) та **засобами сполучення** з керованим об'єктом ЗС (датчики, виконавчі пристрої, АЦП, ЦАП і т. п.).

Зв'язок між розглянутими модулями МПП здійснюється на основі **магістральної схеми**. Відповідно до неї всі модулі підключені до системної магістралі, що складається із **шини адреси ША, шини даних ШД і шини керування ШК**. Кожна із шин становить набір з деякого числа провідників – ліній.

По ША МП повідомляє, з яким з модулів чи з якою коміркою пам'яті (вказується їхній код – адреса) він буде працювати у певний момент часу.

Розрядність ША (число провідників) визначає число модулів, з яким може взаємодіяти МП. Зазвичай вона складає 16 (число модулів до 2^{16} – комірок пам'яті ПЗП чи ОЗП, модулів вводу, виводу і т. ін.).

По ШК повідомляється характер взаємодії: введення даних чи їх вивід (запис чи читання).

По ШД дані надходять у процесор і виводяться з нього, тобто ШД двонаправлена. У кожний конкретний момент часу ШД використовується тільки для вводу або тільки для виводу.

Оскільки до ШД приєднано всі модулі одночасно, то, щоб виключити їхній взаємний вплив, підмикання до провідників шини виконується за допомогою елементів, що мають три стани: **два – 1; 0 (логічні) і третій – стан відключення від шини (нелогічний)**. МП вибирає один модуль для виводу даних на ШД, задаючи його адресу на ША. Виходи інших модулів при цьому знаходяться у третьому стані – відімкнені від ШД.

Розрядність ШД визначає розрядність двійкових даних (розрядність слова даних), з якими може оперувати МП. Зазвичай ШД має 4, 8, 16 або 32 розряди: МП обробляє двійкові числа (слова) з 4, 8, 16 або 32 розрядів – біт (8 біт складають байт, $2^{10} = 1\,024$ біт – кілобайт).

Типовий МП зазвичай виконують у вигляді ВІС, що має до 40 електричних виводів. Наприклад, у восьмирозрядного МП: 8 виводів забезпечують підмикання до ШД, 16 – до ША, 5-7 – до тактового генератора і джерела живлення, решта – до ШК. Є й мікроЕОМ у вигляді ВІС. Їх називають однокристальними.

5.3 Структура мікропроцесора

Структурну схему МП наведено на рисунку 5.3.

МП містить три основні пристрої: арифметико-логічний пристрій (АЛП), регістр даних (РД) і пристрій керування (ПК). Ці та інші пристрої, що приймають участь у пересиланні даних, з'єднано шинами внутрішньої магістралі: ША, ШД і ШК.

Синхронізує і задає темп роботи (швидкодію) МП **генератор тактових імпульсів (ГТІ)**.

Арифметико-логічний пристрій АЛП становить комбінаційний пристрій у якому відбувається виконання арифметичних або логічних операцій (за відповідними командами) над даними.

Регістри забезпечують виконання функцій МП шляхом зберігання кодів даних, адрес, команд та ін.

Акумулятор є основним регістром МП при діях з даними. Більшість арифметичних і логічних операцій з двома словами даних здійснюється з використанням АЛП і акумулятора.

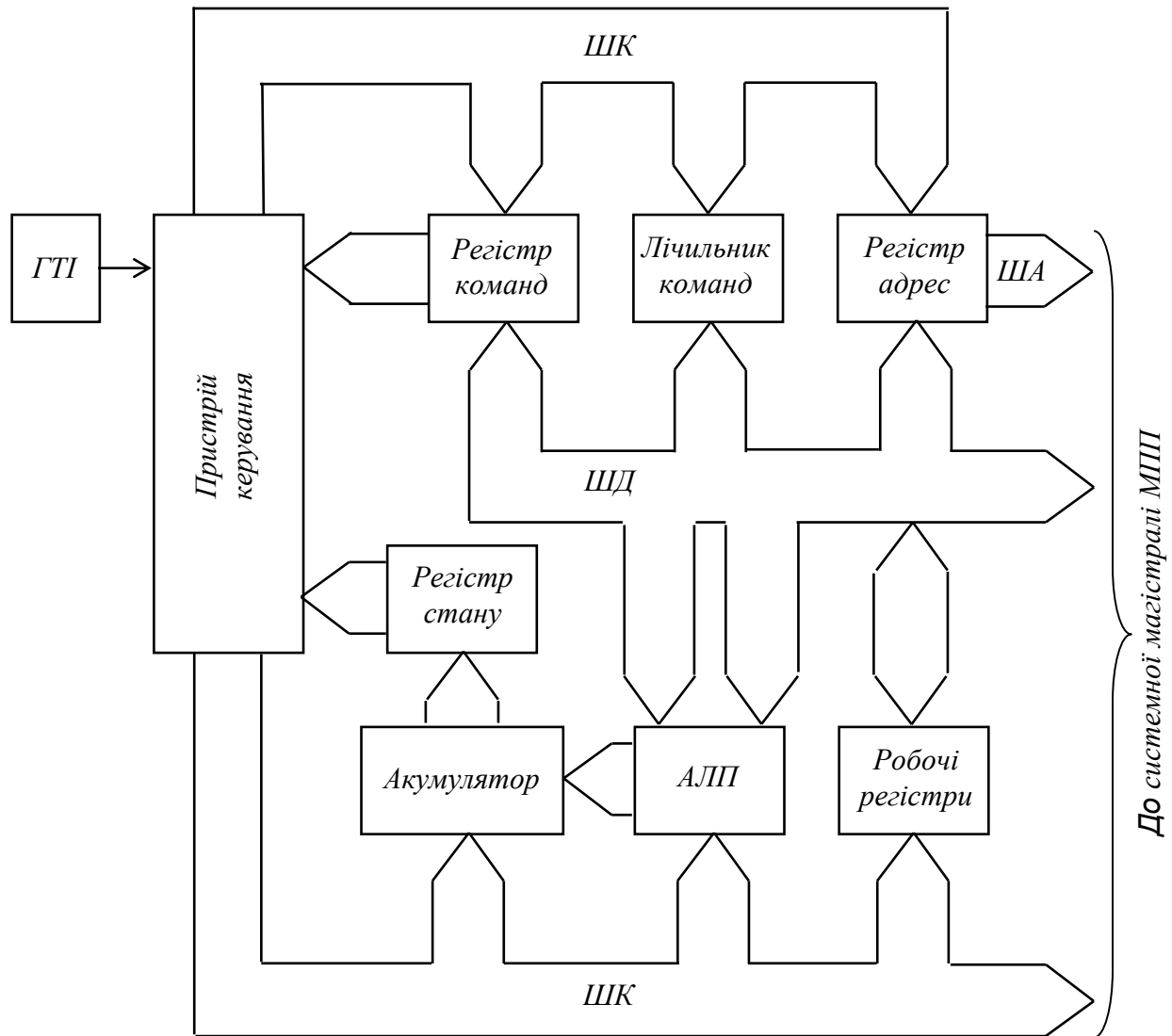


Рисунок 5.3 – Структура мікропроцесора

У **регістрі стану** (ознак або прапорців) зберігаються ознаки результату побіжної операції АЛУ (нульовий, додатний, від’ємний та ін.). Ці ознаки використовуються під час виконання наступних команд, наприклад, для розгалуження програми.

Лічильник команд зберігає номер команди, що виконується і тієї, що буде виконуватись наступною. Перед початком виконання програми до нього заноситься адреса її першої команди.

Регістр адреси пам'яті містить адресу пам'яті, що буде використана МП у наступному такті роботи. Його виходом є ША.

Регістр команд зберігає код команди, що виконується у цей момент.

Робочі реєстри є внутрішньою надоперативною пам'яттю МП і застосовуються для проміжного зберігання кодів даних і адрес, забезпечуючи підвищення швидкодії.

Підключення внутрішньої магістралі МП до системної магістралі МПУ відбувається через **буферні реєстри інтерфейсу і вихідні підсилювачі**.

Пристрій керування узгоджує роботу всіх вузлів МП.

6 ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ

1. Загальні відомості.

2. Запам'ятовувальні пристрої для зберігання оперативної інформації (ОЗП).

3 Запам'ятовувальні пристрої для зберігання постійної інформації (ПЗП).

ОСНОВНІ ПОНЯТТЯ

Інформаційна ємність; біт; слово (байт); організація ЗП; швидкодія ЗП; ЗП RAM-типу; статичні ЗП – SRAM; динамічні ЗП – DRAM; енергозалежна пам'ять; енергонезалежна пам'ять; ЗП для зберігання постійної інформації – ROM-типу; масочні ЗП типу ROM(M); ЗП типу PROM; ЗП типу EPROM; репрограмовані ІМС пам'яті; транзистори ЛІЗМОН-типу; ЗП типу EEPROM; флеш-пам'ять.

6.1 Загальні відомості

Запам'ятовувальні пристрої (далі – ЗП) забезпечують зберігання програми роботи, вихідних даних і результатів обробки, обмін інформацією між окремими частинами мікропроцесорної системи. Будуються вони на напівпровідникових ІМС. Мікросхеми пам'яті складають біля 40 % у загальному обсязі випуску ІМС. Також ЗП входять до складу ВІС: мікропроцесорів, мікроконтролерів та ін.

Максимально можливий об'єм інформації ЗП визначається його **інформаційною ємністю** в бітах (**біт** – мінімальний об'єм цифрової інформації, що може приймати значення 0 або 1) або **словах** (**байтах**, що складаються з декількох біт, наприклад, з чотирьох або восьми). Біт зберігається у **запам'ятовувальному елементі** (далі – ЗЕ), а слово – у **запам'ятовувальній комірці**, що є групою ЗЕ, до яких можливе лише одночасне звернення.

Інформаційна ємність вимірюється, наприклад, в кілобітах ($1 \text{ Кбіт} = 2^{10} = 1024 \text{ біт}$), кілобайтах ($1 \text{ Кбайт} = 2^{10} = 1024 \text{ байт}$), або в мегабітах ($1 \text{ Мбіт} = 2^{20} = 1\,048\,576 \text{ біт}$), мегабайтах ($1 \text{ Мбайт} = 2^{20} = 1\,048\,576 \text{ байт}$) і т.п.

Організація ЗП визначає, скільки слів і якої розрядності він зберігає. Так, наприклад, при ємності 512 біт може зберігатися 64 восьмирозрядних слова, або 128 чотирирозрядних.

Швидкодія ЗП оцінюється часом зчитування і запису.

Цифрові обчислювальні пристрої мають шинну структуру, за якою звертання до комірки ЗП відбувається за номером, що називається **адресою**.

6.2 Запам'ятовувальні пристрої для зберігання оперативної інформації (ОЗП)

Такі ЗП **RAM -типу** (Random Acces Memory – ОЗП) поділяються на статичні і динамічні.

У **статичних ЗП – SRAM** (Static RAM) – за ЗЕ використовуються тригери з ланцюгами установки і встановлення нуля. За К-МОН-технології тригер складається з шести транзисторів. Тому такі ЗП доволі дорогі і займають багато місця на кристалі але мають велику швидкодію.

В **динамічних ЗП – DRAM** (Dynamic RAM) – дані зберігаються у вигляді зарядів ємностей МОН-структур. Основою ЗЕ тут є просто конденсатор невеликої ємності. Такий ЗЕ значно простіший тригерного, що дозволяє розмістити на кристалі значно більше ЗЕ. Оскільки конденсатор з часом втрачає свій заряд, то зберігання даних вимагає їхньої періодичної регенерації (через кожні декілька мікросекунд) за допомогою спеціальних підсилювачів-регенераторів. Зараз ємність динамічних ЗП становить до 128 Мбіт, у стані розробки ЗП на 256 Мбіт (з числом транзисторів на кристалі до 300 мільйонів). Динамічні ЗП у 4-5 разів дешевші статичних і у стільки ж разів мають більшу інформаційну ємність.

Обидва ці типи пам'яті є **енергозалежними** – при вимкненні джерела живлення ІМС інформація безповоротно губиться.

6.3 Запам'ятовувальні пристрої для зберігання постійної інформації (ПЗП)

Є декілька типів ЗП для зберігання постійної інформації – **ROM-типу** (Read On Memory – ПЗП). Як ЗЕ в них використовують перемички, діоди, біполярні і МОН-транзистори.

У **масочні ЗП типу ROM(M)** інформація записується при виготовленні ІМС за допомогою спеціального шаблону – маски. Матриця діодного ЗП такого типу становить координатну сітку з горизонтальних ліній вибірки слів і вертикальних ліній зчитування. Код слова визначається наявністю діода (що відповідає одиниці) або його відсутності (відповідає нулю) у вузлах координатної сітки.

Такі ЗП компактні і дешеві. Їх застосовують для зберігання інформації масового призначення – кодів літер і цифр, таблиць типових функцій, стандартного програмного забезпечення і т. п. Користувач змінити інформацію, що зберігається, не може.

Мікросхеми ЗП типу PROM (Programmable ROM – програмовані ПЗП) програмують одноразово видаленням або створенням перемичок у вузлах координатної сітки. У вихідній заготовці наявні (або відсутні) всі перемички. При програмуванні користувачем за допомогою спеціального програмуючого пристрою залишаються (або видаляються) тільки необхідні. Перемички можуть становити у найпростішому випадку плавкі елементи ввімкнені послідовно з діодом, або два зустрічно увімкнених діоди, один з яких пробивається при програмуванні.

ЗП типу EPROM дозволяють не тільки записувати в них інформацію, а й стирати її та замінювати на нову – вони є **репрограмовуваними**. Як ЗЕ в них використано **транзистори ЛІЗМОН-типу** (МОН-транзистори з лавинною

інжекцією заряду). Такі транзистори мають так званий плаваючий затвор – обмежену з усіх боків діелектриком провідну зону. Введений у неї як у пастку в результаті лавинного пробоя під дією імпульсу напруги у 20–25 В заряд зберігається дуже тривалий час. Цей заряд забезпечує закритий стан транзистора. Стирання інформації відбувається під дією ультрафіолетового опромінення, для чого корпус ІМС має спеціальне прозоре віконце. Виникнення фото- і теплових струмів дозволяє заряду покинути плаваючий затвор. Стирання триває декілька хвилин, одразу стирається вся інформація. Опромінення веде до змін властивостей матеріалів транзисторів, тому число циклів перепрограмування складає лише 10–100 разів.

Новітні ЗП типу **EEPROM** також є репрограмовуваними. Їхньою основою є МНОН-транзистори, що, на відміну від звичайних МОН-транзисторів мають двошаровий підзатворний діелектрик – окрім тонкого шару SiO_2 є ще більш товстий шар нітриду кремнію Si_3N_4 (звідси літера Н в аббревіатурі). Під дією електричного поля достатньо високої напруженості носії заряду проходять через тонкий шар і скопичуються на межі розділу шарів. Після зняття поля заряд залишається у приграничному шарі нітриду кремнію і вже не розсмоктується, що і забезпечує зберігання інформації на протязі десятків років. Для стирання інформації необхідно видалити носії заряду з приграничного шару, для чого подається напруга, що створює електричне поле протилежної направленості. При цьому інформацію можна стирати не зі всього кристалу, а вибірково. Тривалість процесу достатньо коротка. Кількість циклів перепрограмування складає 10^4 – 10^6 разів.

Найвищим досягненням у розробці напівпровідникових програмованих ЗП є створення **флеш-пам'яті**. Її ЗЕ подібні елементам пам'яті типу EEPROM, але в схемах флеш-пам'яті не передбачене стирання окремих слів. Інформація може стиратися або вся одразу, або достатньо великими блоками за єдиним сигналом, миттєво (flash – спалах). По своїй дії флеш-пам'ять вже подібна до пам'яті ОЗУ але є **енергонезалежною**. Пристрої флеш-пам'яті зі збільшенням кількості запам'ятовуваної інформації (з розвитком технології) поступово

витісняють такі накопичувачі, як жорсткі магнітні диски, оптичні диски та інші носії. Вони у сотні разів скорочують споживану потужність, мають велику надійність, невеликі розміри і вагу та у декілька разів збільшують швидкодію пристроїв пам'яті. Однією з найважливіших властивостей є також відсутність в ЗП цього типу рухомих механічних частин, зокрема таких, що обертаються, і повна технологічна сумісність з ІМС. Флеш-пам'ять може бути частиною багатьох типів ІМС мікроконтролерів (що забезпечує універсальність їхнього застосування завдяки можливості багатократного перепрограмування) та ін.

Тут можна зазначити, що це є одним з прикладів поступової відмови людства від колеса у багатьох застосуваннях.

У 1971 році було створено теорію **мемристора** (memo – пам'ять, resistor – електричний опір). Це четвертий після резистора, конденсатора та котушки індуктивності пасивний елемент. Він змінює опір у залежності від заряду, що через нього протікає і після зняття напруги зберігає свій стан. В 2008 році колективом вчених фірми «Hewlett-Packard» було створено лабораторний зразок подібного елемента, що значно менший за розмірами від елемента флеш-пам'яті, менш енергоємніший, має більшу швидкодію. Далі було анонсовано вихід запам'ятовувальних пристроїв на основі мемристорів у 2013 році, який потім перенесли на 2014 рік. Тоді ж було опубліковано проєкт суперкомп'ютера з мемристорною пам'яттю. Його комерціалізація очікується до кінця 2020-х років.

7 ЗАВДАННЯ І ВИКОНАННЯ ПРОГРАМ МІКРОПРОЦЕСОРНОГО ПРИСТРОЮ

1. Алгоритм. Способи представлення алгоритму.
2. Завдання і виконання програм МПП.
3. Шістнадцяткова система числення.
4. Асемблер.

ОСНОВНІ ПОНЯТТЯ

Алгоритм; представлення алгоритму; ініціалізація виконання програми; машинна програма; шістнадцяткова система числення; асемблер; програма-ретранслятор; мнемоніка; мікрокоманда; мікрооперація; внутрішній ПЗП; система команд; вихідна програма; об'єктна програма; потужність МП.

7.1 Алгоритм. Способи представлення алгоритму

Алгоритм (від латинської трансляції імені математика аль-Хорезмі) – це спосіб (програма) розв'язання обчислювальних та інших задач, що точно вказує, як і в якій послідовності отримати результат, однозначно визначений вихідними даними.

Отже послідовність виконання дій пристрою керування і є алгоритмом. Задати алгоритм можна різними способами.

Перш за все – це **описове представлення алгоритму** на «людській» мові. Така форма представлення прийнятна навіть для неспеціаліста у пристроях керування, але громіздка, допускає тлумачення. А отже, не годиться для створення більш-менш складного пристрою керування, як з жорсткою логікою, так і програмованого.

Більш впорядкованим є **представлення алгоритму у вигляді таблиці**.

Доволі популярним є **графічне представлення алгоритму** – у вигляді граф-схеми. Воно забезпечує наочність, оперативність огляду.

На основі алгоритму складають схему електричну принципову пристрою керування з жорсткою логікою або програму роботи програмованого. Це врешті теж є способами представлення алгоритму.

7.2 Завдання і виконання програм МПП

Для забезпечення виконання МПП необхідних функцій у його ПЗП (іноді у ОЗП) записується набір програм, (стандартних, що реалізують, наприклад, арифметичні і логічні операції, обчислення тригонометричних функцій і т. п., а також і спеціальних, що реалізують, наприклад, алгоритми керування конкретним об'єктом).

При складанні програм програміст повинен дати МП детальний опис його дій. МП може працювати з великою швидкістю і точністю протягом тривалого часу, але тільки за чітким завданням, що не допускає тлумачень. МПП може реагувати на зміну умов, якщо в пам'яті у нього є програма, що повідомляє, як йому поводитись у нових обставинах. МПП діє логічно, але не має творчих здібностей (принаймні, у певний час). Удавана розумність різних МПП є результатом великого числа програм, закладених у них. Хоча варто зазначити, що в наш час інтенсивно ведуться роботи зі створення так званого штучного інтелекту (з точки зору філософії, набору програм, що імітують інтелектуальну діяльність людини). І на цьому шляху, як відомо, здобуто чимало досягнень. Вважається, що створення «розумних» машин буде у першому наближенні завершено тоді, коли вони навчатися створювати собі подібних.

Для ініціалізації виконання програми в МП заноситься її початкова адреса (адреса комірки ПЗП чи ОЗП) і подається команда ПУСК.

При роботі МП забезпечує таку послідовність дій:

- 1) витяг даних (коду команди, числа) з пам'яті (ПЗП чи ОЗП);
- 2) розшифрування команди;
- 3) виконання команди тощо знову.

При цьому, наприклад, щоб переслати число з ПЗП до ОЗП, МП, як тактований (синхронний) пристрій, у найближчому такті роботи за сигналами тактового генератора задає згідно з програмою на ША код адреси комірки пам'яті ПЗП. Після завершення перехідних процесів, пов'язаних з зарядом паразитних ємностей провідників (ліній) ША (від чого, до речі, у значній мірі залежить швидкодія пристрою), у другому такті роботи МП по ШК дає команду на видачу коду з ПЗП на ШД (читання), а у наступному такті – команду по ШК на прийом коду у свій внутрішній регістр пам'яті, що називається акумулятором.

Надалі, у подальших тактах роботи, відповідно задається на ША номер комірки ОЗП та подається по ШК команда запису і число, нарешті, потрапляє на визначене програмою місце.

З опису цієї послідовності дій зрозуміло, чому програмні пристрої обчислення і керування значною мірою поступаються у швидкодії порівняно з пристроями з жорсткою логікою, де код з одного вузла до іншого зазвичай передається за один такт по окремо виділених лініях зв'язку.

На робочому рівні МП реагує на список команд, представлений, як уже зазначалося, у вигляді набору двійкових кодів, що називається машинною програмою.

Таке представлення програм є винятково важким для оперативного сприйняття людиною.

7.3 Шістнадцяткова система числення

Деяке спрощення при написанні і читанні команд дає використання **шістнадцяткової системи числення**. Вона містить 16 символів, кожному з яких відповідає чотирирозрядне число у двійковому коді: число представляється як сума ступенів числа 16, помножених відповідно на 0, 1, 2, ..., 9, A, B, C, D, E, F

$$N_{16} = a_1 \cdot 16^0 + a_2 \cdot 16^1 + a_3 \cdot 16^2 + \dots + a_{n+1} \cdot 16^n, \quad (7.1)$$

де $a_i = \{0;1;2;3;\dots;9; A; B; C; D; E; F\}$.

Таблиця 7.1 показує відповідність запису чисел у десятковому, двійковому і шістнадцятковому кодах.

Таблиця 7.1 – Запис чисел у різних кодах (системах числення)

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

Перетворити двійковий код у шістнадцятковий дуже легко: двійкове число поділяється на групи по чотири розряди, починаючи з молодшого – на тетради. Кожна група замінюється відповідним шістнадцятковим символом. При зворотному перетворенні кожен символ замінюється двійковою тетрадою.

Наприклад, $0101\ 0111\ 1101\ 1111\ 0001_2 = 57DF1_{16}$.

Таким чином, шістнадцятковий код, з точки зору застосування для запису програм для МПП, можна вважати більш компактним записом двійкового.

7.4 Асемблер

Програма, записана у шістнадцятковому коді, усе-таки залишається складною для розуміння.

Виходом є використання мови простого програмування – **асемблера**. Відповідна **програма-ретранслятор** переводить слова і фрази програми, представлені у вигляді **мнемонік** (англійських скорочень назв) асемблера, у машинний код, який потім буде занесено у ІМС ПЗП, що підключаються для цього до спеціального адаптера машини-програматора.

Приклади мнемонік асемблера та їхня відповідність у шістнадцятковому коді наведено у таблиці 7.2.

Таблиця 7.2 – Приклади команд асемблера

Команда	Мнемоніка	Шістнадцятковий код
1. Читання даних з об'єкта з адресою F у внутрішній регістр пам'яті A.	INF F	DB FA
2. Порозрядне логічне множення чисел, що знаходяться у внутрішніх регістрах пам'яті A і B.	ANA B	A0
3. Додавання числа, що зберігається у внутрішньому регістрі C до числа, що зберігається у внутрішньому регістрі A.	ADD C	81
4. Вивід числа, що зберігається у внутрішньому регістрі A до об'єкта, адреса якого знаходиться у внутрішніх регістрах пам'яті B і C	STAX B	02

Відзначимо також, що кожній команді асемблера відповідає **мікрокоманда МП** – набір дій МП (**мікрооперацій**), запрограмованих при

виробництві МП у його **внутрішньому ПЗП**. Ініціює виконання мікропрограми пристрій керування МП по отриманні коду команди.

При розробці програм для МПП, що виконує функції керування, складають узагальнений алгоритм виконання програми – послідовність дій з опитування вхідних даних, їхньої обробки та формування команд керування на відповідних виходах.

Далі алгоритм розбивають на окремі фрагменти і деталізують дії у кожному з них до рівня, що відповідає **системі команд** вибраного типу МП (зазвичай – від декількох десятків і десь до півтори сотень команд).

Програма спочатку складається у вигляді **списку мнемонік**, що називається **вихідною програмою**. Кожному коду присвоюється номер комірки пам'яті. Потім вручну або зазвичай за допомогою ЕОМ програма переводиться на машинну мову. Тепер вона називається об'єктною програмою і заноситься в ПЗП чи ОЗП МПП.

Використання алгоритмічної мови високого рівня (BASIC, FORTRAN або ін.) часто недоцільне, бо веде до втрати програмістом контролю за використанням ресурсів пам'яті. Це може призвести після її транслявання у машинні коди до значного перевищення оптимально можливого об'єму ПЗП та втрати швидкодії. А це також веде до збільшення апаратних витрат (числа ІМС ПЗП) та загрожує втратою можливості керуючого МПП працювати у **реальному масштабі часу**, коли реакція МПП на зміну ситуації у об'єкті керування повинна відбуватися зі швидкістю не меншою за ту, з якою відбуваються процеси в об'єкті.

При використанні МПП для керування можливість зміни програми ПЗП заміною ІМС чи їхнім перепрограмуванням (у залежності від типу) забезпечує використання даного МПП у різних застосуваннях для вирішення різних задач без будь-яких схемних змін в апаратурі – функція МПП задається його програмуванням.

Поліпшення техніко-економічних характеристик МП дозволило надзвичайно ефективно використовувати їх у різних пристроях цифрової

обробки інформації і керування (як у центральних процесорах, так і у периферійних пристроях ЕОМ, у мікрокалькуляторах, у системах керування і регулювання поточковими лініями, верстатами-автоматами, роботами-маніпуляторами, стабілізованими джерелами живлення, в електронних пристроях – телевізорах, радіостанціях, вимірювальних приладах, цифрових фотоапаратах, іграшках і т. п.).

Це підняло на новий рівень ефективність роботи технічних пристроїв і систем.

Завдяки використанню МПП ЕОМ у наш час перетворилися у предмет побутової техніки. Взагалі МПП немає різниці чим керувати: ракетою, світлофором чи пральною машиною. Але зазвичай для конкретних застосувань вибирають з широкої номенклатури МП необхідної інформаційної потужності і будують МПП у відповідному конструктивному та кліматичному виконанні.

Потужність МП визначається його здатністю обробляти дані і оцінюється трьома параметрами: довжиною слова даних (кількість розрядів ШД), довжиною слова пам'яті (кількість розрядів ША – визначає кількість об'єктів, до яких може звертатися МП), швидкістю виконання команд.

8 ПРОГРАМОВАНІ ЛОГІЧНІ КОНТРОЛЕРИ. МІКРОКОНТРОЛЕРИ

1. Програмовані логічні контролери.

2. Мікроконтролери.

3. Структура мікроконтролера.

4. Засоби програмування для мікроконтролерів.

5. Співставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих.

ОСНОВНІ ПОНЯТТЯ

Програмований логічний контролер; багатоплатний ПЛК; одноплатний ПЛК; мікропроцесор; контролер; мікроконтролер; мікрокомп'ютер; центральний процесорний елемент (ЦПЕ); шина адрес / даних; паралельні і послідовні порти вводу / виводу; запити на переривання; таймер-лічильник.

8.1 Програмовані логічні контролери

Зрозуміло, що продуктивність устаткування та точність виконання технологічних операцій значною мірою залежать від ступеню автоматизації. Також зрозуміло, що у наш час найкраще забезпечують це електронні пристрої керування, особливо цифрові.

Тут варто зазначити, що цифрові пристрої керування реалізуються як автомати з жорсткою логікою або програмовані.

Апаратні методи побудови цифрових пристроїв керування з жорсткою логікою близькі до методів побудови широко розповсюджених раніше аналогових систем керування: для виконання кожної функції використовується окремий елемент або група елементів і введення нової функції вимагає введення нових елементів. Системи керування, реалізовані такими методами, мають високу швидкодію та надійність, традиційні принципи їхньої побудови не викликають труднощів при розробці та експлуатації (з точки зору

застосовуваного при регламентних та ремонтних роботах обладнання, приладів та з огляду на рівень кваліфікації персоналу). В той же час, вони складаються з великої кількості електронних компонентів (переважно інтегральних мікросхем – ІМС – середнього ступеню інтеграції), громіздкі, мають малу гнучкість при необхідності змін алгоритму керування (або частіше взагалі не дозволяють цього робити) та дорого коштують. Вони вимагають індивідуальної розробки, причому, кількість типів функціональних модулів, що створюють функціонально закінчену систему, яка слугує для побудови різних систем керування, виявляється значною. Отже використання пристроїв керування з жорсткою логікою вимагає у кожному окремому випадку розробки спеціального унікального пристрою.

Для створення МПП керування технічними пристроями зручно використовувати програмовані логічні мікроконтролери, що випускаються промисловістю.

Програмовані пристрої будуються на основі процесорів зазвичай у мікрвиконанні – мікропроцесорів.

Варто зазначити, що побудова пристрою керування класу, що нами розглядається, на основі мікропроцесорного набору (власне процесор, оперативний та постійний запам'ятовувальні пристрої, інтерфейс та ін.) за величиною апаратних затрат мало відрізняється від його реалізації на ІМС середнього ступеня інтеграції. Вартість розробки та реалізації такого пристрою виявляється навіть більшою за вартість пристрою з жорсткою логікою на ІМС середнього ступеня інтеграції. Крім того, мікропроцесорні пристрої висувають зовсім інші вимоги до кваліфікації як розробників апаратури, так і до тих, що забезпечують експлуатацію. Але гнучкість програмованих пристроїв керування при необхідності змін алгоритму, що має величезне значення як на етапах розробки так і при експлуатації одиночно або малосерійно продукowanego технологічного устаткування, робить їхнє застосування вкрай перспективним.

Тут могли б придатися програмовані логічні контролери.

Зазначимо, що раніше контролерами називали багатопозиційні перемикачі для комутації електричних кіл (наприклад, як ті, що застосовуються для керування електроприводом трамваю).

Програмований логічний контролер – це програмований логічний пристрій керування з мовою програмування, доступною для неспеціалістів з інформатики (спочатку, наприклад, мали мову драбинчастих діаграм і призначалися для заміни релейних шаф керування послідовними логічними процесами в умовах промисловості у реальному масштабі часу).

Отже, малогабаритні і гнучкі у програмуванні ПЛК спочатку використовувалися для заміни логічних пристроїв керування, виконаних на електромагнітних реле, що займали цілі шафи. Надалі, з появою мікроелектронних ЦАП і АЦП, їх почали застосовувати і для розв’язання задач, пов’язаних з обчисленням та порівнянням аналогових сигналів.

ПЛК монтуються на одній або декількох платах (у вигляді одного або декількох модулів) і доповнюються пристроями програмування і індикації з клавіатурою та цифровим дисплеєм.

Більшу універсальність мають **багатоплатні** ПЛК, що становлять набір модулів: процесора, ПЗП, ОЗП, ЦАП, АЦП, вводу та виводу даних, підсилення потужності, джерела живлення та ін. З цих модулів створюється необхідна у конкретному випадку конфігурація ПЛК, для чого необхідний їхній набір встановлюється у так звану корзину – конструкцію з основою, на якій розміщено рознімачі для підмикання модулів до системної магістралі і джерела живлення. Підмикання до джерел сигналів та до виконуючих пристроїв об’єкта керування здійснюється окремими кабелями.

Таким чином, багатоплатний ПЛК є універсальним конструктором для створення програмованих пристроїв керування.

Такі ПЛК широко застосовувались до тих пір, доки з розвитком мікросхемотехніки стало можливим усі вузли (до 95–98 % слабкострумівих елементів) контролера розміщувати у одній ВІС – отримали **одноплатний**

ПЛК. Силові елементи і джерело живлення при цьому розміщують на окремих платах.

Одноплатні ПЛК мають значно менші габаритні розміри, ніж багатоплатні, простіші у виготовленні і дешевші. Вони більш надійні, бо не мають корзини з великою кількістю дорогих і малонадійних рознімачів. Але вони не є універсальними. Тому фірми-продукувачі зазвичай випускають набори таких ПЛК з різними можливостями для забезпечення керування типовими для деякої галузі техніки об'єктами.

Виходячи з недоліків чи переваг одно- і багатоплатних ПЛК у конкретних застосуваннях використовують як одні, так і інші.

Побудовані на мікропроцесорах контролери, мають спрощену систему команд, випускалися серійно і були достатньо дешеві, порівняно з вартістю розробки та виготовлення пристрою керування з жорсткою логікою. Але як пристрої керування технологічного устаткування програмовані логічні контролери широкого розповсюдження не знайшли, бо були спочатку громіздкими, все ж таки достатньо дорогими і для багатьох можливих застосувань (наприклад, для керування електронними ключами джерел живлення технологічного устаткування) повільно діючими. Також важливим чинником у обмеженні їхнього розповсюдження були обмежена доступність для пересічного споживача як власне контролерів, так і інформації про користування ними.

8.2 Мікроконтролери

Починаючи десь з середини 80-х років минулого сторіччя на світовому ринку електронної продукції з'являються ІМС мікроконтролерів. Наприклад, перші восьмирозрядні мікроконтролери були створені фірмою «Intel» (США) у 1982 році. Зараз такі мікроконтролери у світі продукують більш як тридцять фірм (деяку незручність створює несумісність мікроконтролерів різних фірм-виробників).

До речі, подібні ІМС з'явилися у кінці 80-х років минулого сторіччя і називалися однокристальними електронними обчислювальними машинами (серія KP1816). Знову ж таки, вони були дорогими, малодоступними, бракувало інформації про їхнє застосування.

На територію країн СНГ мікроконтролери поставляються переважно фірм «Atmel» та «Microchip Technology Inc.» (США), виготовлені у Південно-Східній Азії. Ці мікроконтролери мають особливу популярність також і у розробників електронної техніки малої та середньої складності всього світу. Номенклатура їхня надзвичайно велика. Наприклад, тільки фірма «Atmel» у кінці 90-х років минулого сторіччя випустила 140 моделей на протязі одного року (різної інформаційної потужності, у корпусах для різного типу монтажу та ін.).

Наведемо сучасне розуміння термінів у цій галузі.

Мікропроцесор – є центральним обчислювальним ядром комп'ютерної системи.

Контролер – закінчений електронний пристрій, зазвичай виконаний на друкованій платі і призначений для прийому та обробки сигналів від датчиків, а також для керування зовнішніми пристроями на основі результатів обробки прийнятих сигналів.

Мікроконтролер – програмно керована ІМС, що застосовується для побудови контролерів. Мікроконтролер окрім ядра має оперативний та постійний запам'ятовувальні пристрої, таймери, лічильники, канали вводу та виводу інформації, інші пристрої – це самостійна комп'ютерна система, що вміщує процесор, допоміжні схеми та пристрої вводу-виводу даних, розміщені у спільному корпусі.

Мікрокомп'ютер має ще й відео- та звуковий процесори.

Нагадаємо, традиційні великі комп'ютери, що будувались ще з сорокових років минулого століття спочатку на лампах, а потім на ІМС середнього ступеня інтеграції, мали так звану прінстонську архітектуру (запропоновану у 1949 році фон Нейманом, що працював у Прінстонському

університеті), за якої як для програм, так і для даних використовувався єдиний простір пам'яті. Це, внаслідок зменшення апаратних затрат, підвищувало надійність машин, але шляхом зниження швидкодії.

Мікроконтролери, у яких завдяки інтегральній технології виконання відпала проблема впливу кількості складових мікроелементів на надійність виробу, мають гарвардську архітектуру (запропоновану тоді ж таки Гарвардським університетом, але відхилену із зрозумілих міркувань), за якої для даних і команд використано окремі області пам'яті та шини. Це забезпечує високу швидкодію завдяки можливості доступу за один цикл як до пам'яті, так і до даних за одночасного виконання побіжної команди і вибірки наступної.

Мікроконтролери мають низьке споживання (струм живлення складає кілька міліампер завдяки побудови на К-МОН-структурах), мінімальні габарити (випускаються залежно від інформаційної потужності у 8-, 18-, 20-, 28-, 40-, 44- та 64-вивідних корпусах, та найчастіше застосовують контролери у перших чотирьох типах корпусів), потребують мінімум (буквально декілька) зовнішніх дискретних компонентів.

Пам'ять даних у них складає від 36 байт до 368 байт, а пам'ять програм від 0,5 кілобайтів до 128 кілобайтів.

Кількість ліній вводу-виводу, що можуть за вибором передавати інформацію у одному з напрямків, становить від 6 до 53.

Робоча частота становить 12–24 МГц.

Напруга живлення може знаходитись у межах 1,8–6 В (номінальне значення 5 В).

Вони можуть мати у своєму складі аналогові компаратори (наприклад, два), аналого-цифрові перетворювачі (наприклад, у деяких мікроконтролерів AVR фірми «Atmel» від 4 до 11), таймери (1–2), широтно-імпульсні модулятори (1-8), різні інтерфейси зв'язку з зовнішніми пристроями.

У мікроконтролерів PIC (Peripheral Interface Controller – периферійний контролер інтерфейсу) фірми «Microchip» є можливість захисту коду програми від несанкціонованих змін або копіювання (один або два біти захисту).

За технологією виконання пам'яті програм є три типи мікроконтролерів:

1) масочно-програмовані – мають найменшу вартість, але застосовуються лише при масовому виробництві, бо програмуються на заводі-виробнику ІМС у процесі виготовлення кристалу;

2) з ультрафіолетовим стиранням – дозволяють перепрограмування, але процес цей доволі тривалий і після ультрафіолетового опромінення можлива нечітка робота контролера, особливо за зниженої напруги живлення;

3) з флеш-пам'яттю – дозволяють багаторазове електричне стирання (дозволяють здійснювати розробку і відпрацьовування програм на готовому пристрої), але у декілька разів дорожчі від контролерів першого типу.

Та все ж вартість масово застосовуваних мікроконтролерів останнього типу сягає лише чотирьох доларів США. Саме такі мікроконтролери є найприйнятнішими для застосування при розробці пристроїв керування технологічного устаткування, виходячи з таких їхніх особливостей:

1) практично необмежена кількість циклів перепрограмування, що в умовах одиничного та малосерійного виробництва забезпечує виконання налагоджувальних і довідних робіт, а також оперативну зміну алгоритму керування при експлуатації устаткування;

2) достатньо високі робочі частоти;

3) наявність простих, дешевих і доступних для самостійної реалізації моделей програматорів;

4) мінімальні габарити, вартість, висока надійність;

5) доступність і вкрай прийнятна вартість;

6) наявність моделей з вбудованими аналоговими компараторами та аналого-цифровими перетворювачами (з розвитком інтегральної технології з'явилась можливість розміщення на одному кристалі як цифрових так і аналогових елементів);

7) можливість захисту програм від несанкціонованих змін та копіювання;

8) можливість отримання інформації від сучасних інтегральних датчиків та передачі інформації комп'ютеру у відповідних протоколах інтерфейсу;

9) велика кількість доступної довідкової та іншої технічної інформації, програмного забезпечення (для складання програм, їхнього відлагодження, занесення у пам'ять мікроконтролера), інформації з типових застосувань та досвіду використання як у друкованому вигляді, так і у інтернет-джерелах.

На жаль, сучасні мікроконтролери втратили початкову особливість програмованих логічних контролерів – доступність мови програмування для неспеціалістів з інформатики. Хоч процесори мікроконтролерів, на відміну від CISC-процесорів (Complex Instruction Set Computer) EOM з складною системою команд, відносять до RISC (Reduced Instruct Set Computers) процесорів, у яких набір виконуваних команд скорочено до мінімуму (залежно від типу мікроконтролера його асемблер має від 33 до 133 команд), їхнє програмування на асемблері особливо складне і доступне власне спеціалістам з мікроконтролерів. Дещо простіше програмування на алгоритмічній мові C+ або на спеціальній версії Basic, але також вимагає впевненого знання комп'ютера та веде до збільшення об'єму програм до тридцяти відсотків.

Це висуває вимоги високої кваліфікації персоналу (вміння користування комп'ютером, програмування), використання при обслуговуванні й ремонті пристроїв керування на основі мікроконтролерів достатньо дорогого комп'ютерного устаткування.

Але сучасний рівень виробництва все більше висуває саме такі вимоги як до персоналу, так і до власне устаткування.

Окрім того, зрештою, конечно зниження строків виробництва устаткування, зниження його вартості, збільшення надійності роботи, зручності експлуатації того варті.

8.3 Структура мікроконтролера

Отже, перші мікроконтролери з'явилися у 1976 році, коли у одній ВІС стало можливим сумістити процесор, ОЗП, ПЗП і елементи інтерфейсу. На відміну від універсальних мікроЕОМ в мікроконтролерах невелика за розміром пам'ять (декілька сотень байт ОЗП та декілька десятків кілобайт ПЗП) і простий інтерфейс зв'язку з зовнішніми пристроями. Це витікає із специфіки використання мікроконтролерів. Їх не застосовують в універсальних обчислювальних системах. Мікроконтролери призначені для створення високоефективних і дешевих систем керування і регулювання. Вони реалізують відносно нескладні алгоритми, тому потребують об'ємів пам'яті на декілька порядків менших, ніж ЕОМ універсального призначення. Їх застосовують у системах керування маніпуляторів, вимірювальних приладів, технологічного устаткування, станків, автомобілів, побутової техніки та ін.

Типову структуру мікроконтролера наведено на рисунку 8.1.

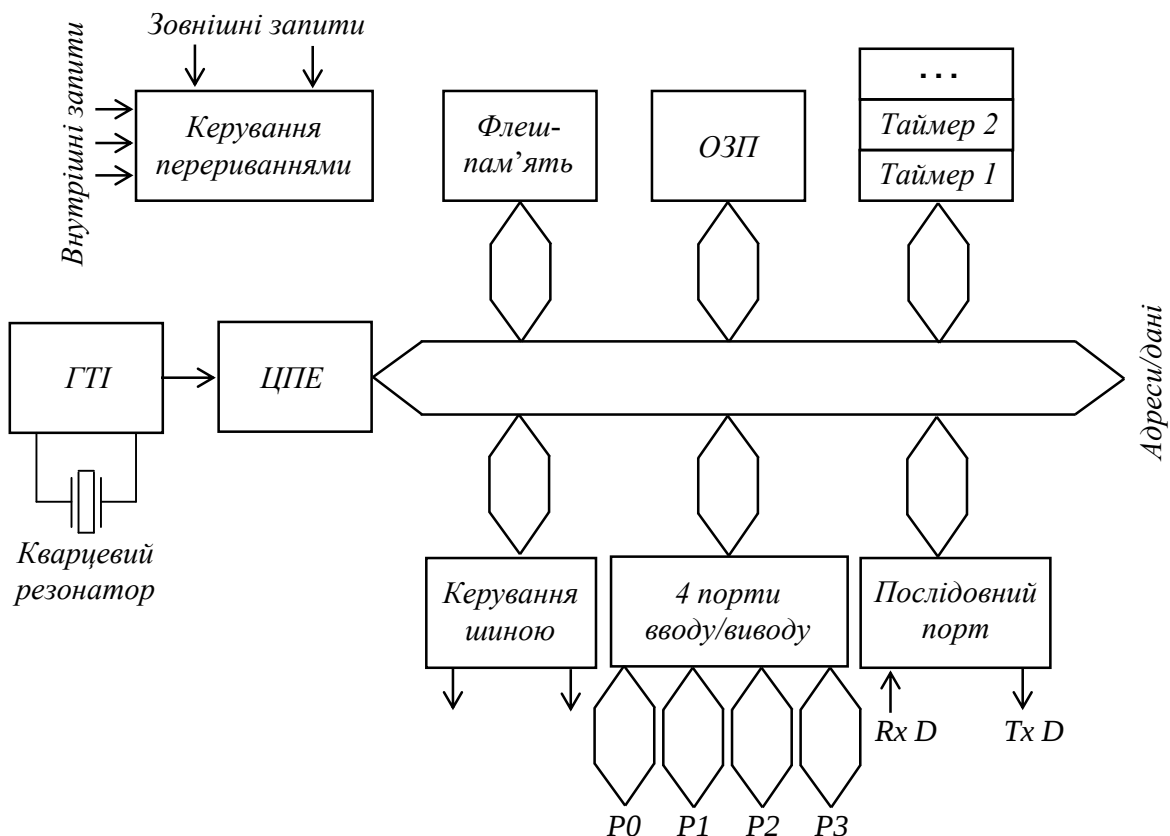


Рисунок 8.1 – Типова структура мікроконтролера

Ємності ОЗП у декілька сотень байт цілком достатньо для тимчасового зберігання даних та проміжних результатів при виконанні програм керування. Програми зберігаються у флеш-пам'яті ємністю у десятки кілобайт, чого зазвичай також вповні вистачає. За необхідності, передбачено можливість нарощування об'ємів пам'яті шляхом підключення зовнішніх ВІС оперативної та постійної пам'яті.

Зрозуміло, що роботою всіх вузлів керує **центральный процесорний елемент ЦПЕ**. Синхронізують його роботу імпульси ГТІ з частотою 12–24 МГц, що задається зовнішнім кварцевим резонатором. Це забезпечує високу стабільність частоти, що необхідно при відпрацюванні проміжків часу.

Адреси і дані передаються по **шині адрес / даних** з розрядністю слів, що обробляються мікроконтролером.

Ввід і вивід інформації здійснюється, наприклад, через **чотири паралельних і один послідовний порти вводу / виводу**. Через порти *P0-P3* обмін з зовнішніми пристроями відбувається по 8-розрядних шинах. Зараз також випускають 16- і 32-розрядні мікроконтролери. Кожен з портів забезпечує взаємодію з одним пристроєм. Функції ліній шин в різних режимах можуть мати різне значення (ввід або вивід).

Через вхід *Rx D* послідовного порту відбувається прийом даних у послідовному коді, через вхід *Tx D* – передача.

Мікроконтролер може обслуговувати декілька як внутрішніх так і зовнішніх **запитів на переривання** програми для переходу до виконання більш пріоритетних підпрограм.

Таймери-лічильники призначені для відпрацювання інтервалів часу або підрахунку кількості імпульсів.

Систему команд мікроконтролера орієнтовано на виконання задач керування, тому поряд зі звичайними командами, характерними для всіх мікропроцесорів, є й специфічні.

Оскільки всі вузли мікропроцесорної системи тут знаходяться всередині однієї ВІС, то непотрібні звертання до зовнішніх пристроїв. Тому більшість

команд короткі з часом виконання, наприклад, біля 1 мкс за тактової частоти 12 МГц.

8.4 Засоби програмування для мікроконтролерів

Засоби програмування для мікроконтролерів AVR охоплюють як апаратні, так і програмні інструменти, що забезпечують повний цикл розробки – від написання коду до прошивки й налагодження пристрою.

Основною мовою програмування AVR-контролерів є мова C, що використовується в середовищі AVR-GCC – стандартному компіляторі сімейства GNU. У складі цього компілятора доступна також бібліотека AVR Libc, яка містить макроси, функції та опис регістрів, що спрощують роботу з периферією мікроконтролера (таймери, UART, SPI, I2C, АЦП тощо). Для більш просунутих розробок використовується C++.

Найбільш популярним інтегрованим середовищем розробки є Microchip Studio (раніше Atmel Studio). Це безкоштовне середовище з підтримкою C/C++, симуляції, налагодження через JTAG або DebugWIRE, інтегрованим компілятором AVR-GCC та засобами автоматичної генерації коду через фреймворки ASF або START. Microchip Studio працює лише в середовищі Windows.

Інтерфейс ISP (In-System Programming) – це один із найпоширеніших способів програмування мікроконтролерів AVR без вилучення їх із плати. Він дозволяє записувати прошивку безпосередньо в чип через SPI-шину, використовуючи кілька сигнальних ліній. Цей метод широко підтримується в більшості мікроконтролерів сімейства AVR (рис. 8.2).

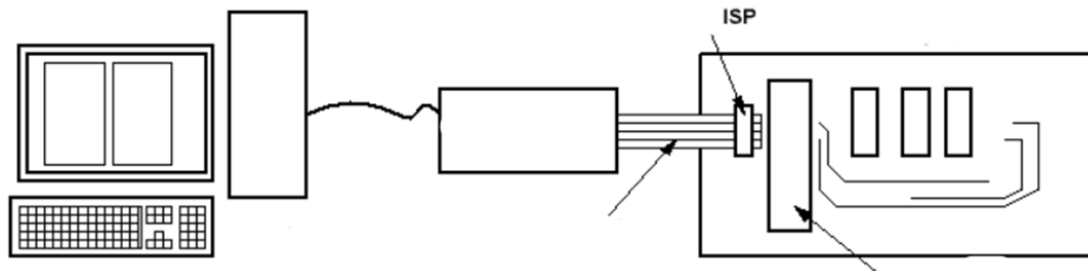


Рисунок 8.2 – Типова схема підключення мікроконтролера до програматора

Під час програмування через ISP до мікроконтролера підключаються такі сигнали (рис. 8.3):

- MOSI (Master Out, Slave In) – передача даних від програматора до МК;
- MISO (Master In, Slave Out) – передача даних від МК до програматора;
- SCK (Serial Clock) – синхронізуючий сигнал;
- RESET – перезавантаження мікроконтролера для переходу в режим програмування;
- GND – загальна земля;
- VCC – живлення (не завжди обов'язкове).

У режимі ISP контролер переходить у спеціальний стан, де може приймати інструкції для запису чи зчитування даних у флеш-пам'ять, EEPROM або fuse-біти (біти конфігурації).

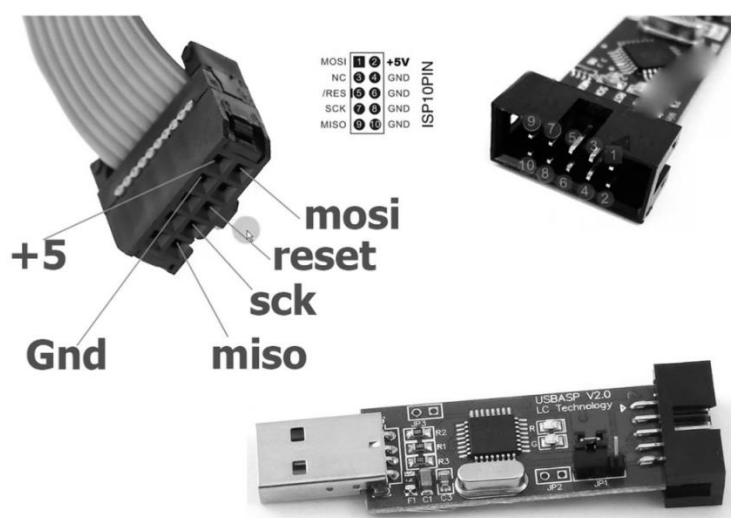


Рисунок 8.3 – Розпіновка інтерфейсу ISP програматора

Інформація щодо пінів для програмування до яких можна підключити програматор відображається у спеціальних інформаційних листах (даташитах). В цих цифрових документах зазвичай міститься вся необхідна інформація щодо роботи мікроконтролера включаючи інформацію щодо програмування. Приклад «цокольовки» мікроконтролерів наведено на рисунку 8.4.

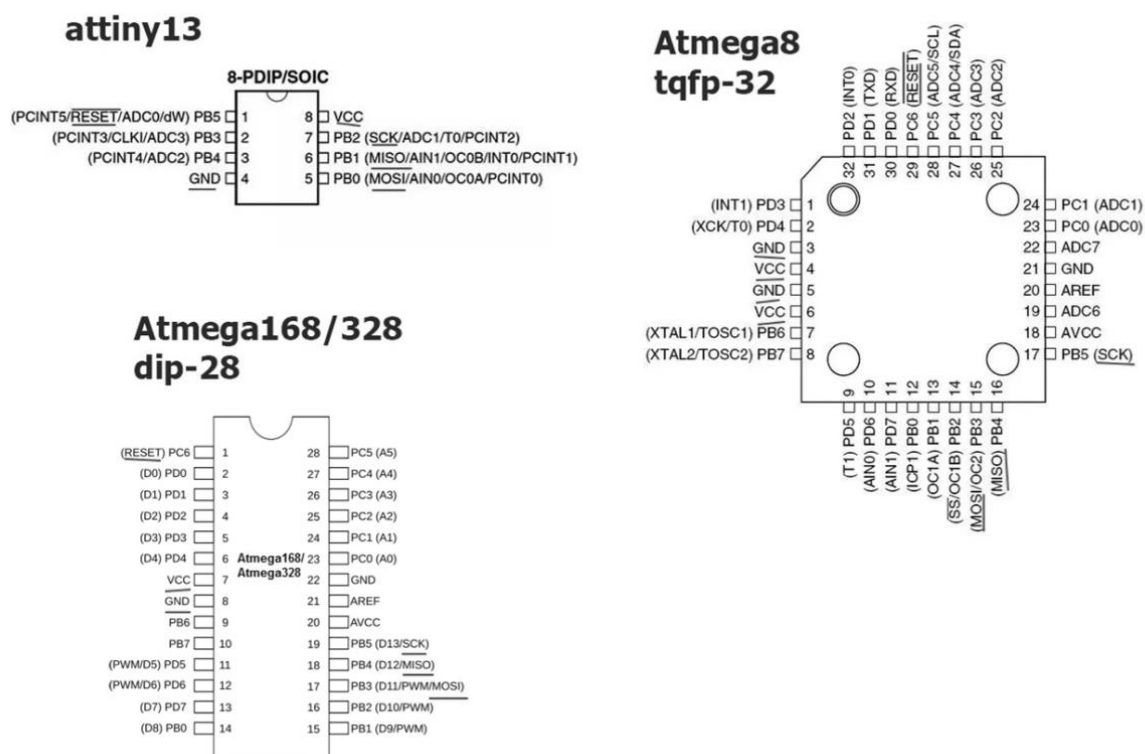


Рисунок 8.4 – «Цокольовка» пінів поширених мікроконтролерів сімейства AVR

Для запису прошивки в мікроконтролер зазвичай використовується утиліта AVRDUDE. Вона підтримує різноманітні апаратні інтерфейси програмування, зокрема ISP, PDI, TPI, JTAG. Одними з найпоширеніших апаратних програматорів для AVR є USBasp, USBtinyISP, AVRISP mkII. Вони з'єднуються з мікроконтролером через інтерфейс SPI та дозволяють завантажити файл прошивки у форматі HEX або ELF.

AVRDUDE (AVR Downloader-Uploader) – кроссплатформенна вільна консольна програма, призначена для прошивки мікроконтролерів фірми

«Atmel» серії AVR. Користується значною популярністю серед розробників та підтримує понад 80 програматорів та понад 150 мікроконтролерів. Крім «рідних» пристроїв від Atmel сюди входять саморобні аматорські пристрої, які не підтримуються AVR Studio, наприклад популярний програматор USBAsp.

У зв'язку зі зростаючим інтересом до портування ПЗ на інші операційні системи, розробник Брайан Дін вирішив зробити проєкт публічним. Також він змінив назву програми на Avrdude, щоб усунути неоднозначність з AVRProg – утиліти, що розповсюджується Atmel разом зі своїм програмним забезпеченням AVR Studio.

Для початківців зручною альтернативою є середовище Arduino IDE. Воно дозволяє писати код на спрощеному C++ і використовує власну екосистему бібліотек. Платформи на основі AVR (наприклад, Arduino Uno) мають вбудований завантажувач, тому прошивка здійснюється через USB, без необхідності зовнішнього програматора.

Для моделювання й тестування програмної логіки доцільним є використання симуляторів, таких як Proteus. Він дозволяє створювати віртуальні схеми з AVR-контролерами та запускати в них прошивку. Інші менш поширені емулятори включають SimulAVR та VMLAB.

У процесі розробки проєктів часто використовують програмні бібліотеки, такі як ASF (Atmel Software Framework) — набір готових драйверів для периферії, що інтегруються у Microchip Studio, або ручну роботу з регістрами за допомогою AVR Libc.

8.5 Співставлення основних етапів створення пристроїв керування на основі цифрових автоматів з жорсткою логікою і програмованих

У таблиці 8.1 наведено співставлення основних етапів створення пристроїв керування з жорсткою логікою і програмованих. З неї видно, що при застосуванні типових, таких, що можна придбати готовими, програмованих пристроїв керування, у циклі створення готового пристрою відсутні процедури,

пов'язані з розробкою конструкторської і технологічної документації та власне виробництвом. Усі процедури фактично пов'язано винятково з розробкою та відлагодженням програми.

Таблиця 8.1 – Співставлення основних етапів створення пристроїв керування

Пристрій керування з жорсткою логікою	Програмований пристрій керування
1. Розробка алгоритму керування	
2. Розробка електричної принципової схеми пристрою керування на основі обраного типу ІМС	2. Розробка програми роботи пристрою керування на основі обраного типу МПП (наприклад, ПЛК)
3. Створення макету пристрою керування	3. Занесення програми в ОЗУ МПП
4. Відлагодження макету пристрою керування і відповідне коригування його електричної принципової схеми	4. Відлагодження і відповідне коригування програми
5. Розробка конструкторської і технологічної документації для виробництва пристрою керування	5. Придбання типового МПП (наприклад, ПЛК)
6. Виробництво пристрою керування	6. Занесення програми в пам'ять мікроконтролера або в ІМС постійної пам'яті і встановлення їх в ПЗУ МПП
7. Поставка пристрою керування замовнику	

На завершення зазначимо, що специфіка роботи з МПП відносить їх скоріш до галузі інформатики, бо у більш-менш серйозних застосуваннях вартість розробки програмного забезпечення складає до 80 % вартості всього пристрою керування. Виходячи з цього ми тут обмежились лише наданням загальних понять, особливо у тому, що відноситься до питань програмування МПП.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. Мікропроцесорна техніка : підручник / Ю. І. Якименко, Т. О. Терещенко, Є. І. Сокол та ін. ; за ред. Т. О. Терещенко. – 2-ге вид., – Київ : ІВЦ «Видавництво «Політехніка»; Кондор, 2004. – 416 с.
2. Колонтаєвський Ю. П. Електроніка і мікросхемотехніка : підручник / Ю. П. Колонтаєвський, А. Г. Сосков ; за ред. А. Г. Соскова. – Київ : Каравела, 2006. – 384 с.
3. Сосков А. Г. Промислова електроніка : підручник / А. Г. Сосков, Ю. П. Колонтаєвський ; за ред. А. Г. Соскова. – Київ : Каравела, 2015 – 520 с.
4. Костенко І. М. Мікропроцесорна техніка : підручник / І. М. Костенко. – Київ : Кондор, 2018. – 392 с. ISBN 978-617-673-808-7.
5. Тесленко В. І. Мікропроцесори та мікроконтролери : навч. посіб / В. І. Тесленко. – Київ : КНУБА, 2020. – 264 с. ISBN 978-966-303-345-2.
6. Пащенко В. О. Мікропроцесорна техніка: Теорія та практика / В. О. Пащенко, Л. П. Малюта. – Харків : Факт, 2017. – 316 с.

Електронне навчальне видання

КОСТЕНКО Іван Олександрович,

ШКУРПЕЛА Олександр Олександрович,

КУЗНЕЦОВ Анатолій Іванович

МІКРОПРОЦЕСОРНА ТЕХНІКА

КОНСПЕКТ ЛЕКЦІЙ

*(для здобувачів першого (бакалаврського) рівня вищої освіти всіх форм
навчання зі спеціальності*

141 – Електроенергетика, електротехніка та електромеханіка)

Відповідальний за випуск *Я. Б. Форкун*

Редактор *О. В. Михаленко*

Комп'ютерне верстання *І. О. Костенко*

План 2025, поз. 168Л

Підп. до друку 14.11.2025. Формат 60 × 84/16.

Ум. друк. арк. 5,0.

Видавець і виготовлювач:

Харківський національний університет
міського господарства імені О. М. Бекетова,
вул. Чорноглазівська, 17, Харків, 61002.

Електронна адреса: office@kname.edu.ua

Свідоцтво суб'єкта видавничої справи:

ДК № 8386 від 14.07.2025.