

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Одеська національна академія зв'язку ім. О.С. Попова

Кафедра комп'ютерно-інтегрованих технологічних процесів та виробництв

О.М. Воробйова, М.П. Савицька, Ю.В. Флейта

ЦИФРОВІ ПРИСТРОЇ

Навчальний посібник

Частина 2

Одеса – 2016

Воробйова О.М. Цифрові пристрої: навч. посіб. – Ч. 2 / О.М. Воробйова, М.П. Савицька, Ю.В. Флейта. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 80 с.

Надано принципи побудови та функціонування інтегральних логічних елементів, логічні та арифметичні основи цифрових пристроїв, методи синтезу пристроїв, компараторів, шифраторів, перетворювачів кодів, мультиплексорів, суматорів, тригерів, регістрів, лічильників, елементарних цифрових автоматів, цифро-аналогових та аналого-цифрових перетворювачів, мікросхем пам'яті.

УХВАЛЕНО

на засіданні кафедри КІТ П і В
і рекомендовано до друку.
Протокол № 7 від 28.01.2016 р.

ЗАТВЕРДЖЕНО

методичною радою академії зв'язку.
Протокол № 6 від 23.02.2016 р.

© Воробйова О.М., Савицька М.П.
Флейта Ю.В., 2016
© ОНАЗ ім. О.С. Попова, 2016

ЗМІСТ

Передмова	5
Список прийнятих скорочень.....	5
1. КОМБІНАЦІЙНІ ПРИСТРОЇ	7
1.1. Загальні відомості	7
1.2. Шифратори.....	7
1.3. Дешифратори	10
1.4. Перетворювачі кодів	14
1.5. Мультиплексори	15
1.6. Демультиплексори	17
1.7. Цифрові компаратори	18
1.8. Програмовані логічні матриці.....	19
1.8.1. Матрична схема.....	19
1.8.2. Дворівнева та триврівнева ПЛМ	23
1.9. Особливості роботи комбінаційних пристроїв	26
Контрольні питання	27
Рекомендована література	27
2. ЦИФРО-АНАЛОГОВІ ТА АНАЛОГО-ЦИФРОВІ ПЕРЕТВОРЮВАЧІ.....	28
2.1. Загальні відомості	28
2.2. Цифро-аналогові перетворювачі	29
2.2.1. Параметри ЦАП	30
2.2.2. Схеми ЦАП	32
2.3. Аналого-цифрові перетворювачі	38
2.3.1. Параметри АЦП	40
2.3.2. АЦП розгортального перетворення	42
2.3.3. Слідкуючий АЦП	44
2.3.4. АЦП паралельного кодування	46
2.3.5. Послідовно-паралельні АЦП	48
Контрольні питання	49
Рекомендована література	49
3. ВСТУП ДО МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ (МТ)	51
3.1. Переваги використання МТ	51
3.2. Структура мікропроцесорної системи (МС)	52
3.3. Узагальнена структура мікропроцесора (МП)	55
3.4. Пристрої мікроконтролерів AVR	57
3.4.1. Процесор	57
3.4.2. Пам'ять.....	58
3.4.3. Периферія.....	60
3.4.4. Переривання (INTERRUPTS)	60
3.4.5. Таймери-лічильники (TIMER-COUNTERS)	60
3.4.6. Аналого-цифровий перетворювач (A/D convertor).....	61
3.4.7. Універсальний послідовний приймач-передавач	

(VART або VSART)	61
3.4.8. Інтерфейс JTAG	61
3.4.9. Стек	61
Контрольні питання	62
4. ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ	63
4.1. Класифікація запам'ятовувальних пристроїв (ЗП)	63
4.2. Параметри ЗП	64
4.3. Оперативні запам'ятовувальні пристрої	64
4.3.1. Статичні запам'ятовувачі ВІС ОЗП	65
4.3.2. Структура ВІС ОЗП	66
4.3.3. Інформаційні та керуючі сигнали ВІС ОЗП	70
4.4. Постійні запам'ятовувальні пристрої (ПЗП)	73
4.4.1. Класифікація ПЗП	73
4.4.2. Структура ВІС ОЗП	74
Контрольні питання	78
Рекомендована література	79

ПЕРЕДМОВА

Радіолокація, автоматика та телемеханіка, супутниковий зв'язок та системи глобального позиціонування, системи відеонагляду, охоронні системи та інші – це галузь широкого застосування різноманітних цифрових пристроїв та комп'ютерів, і пов'язано з тим, що цифрові пристрої не потребують індивідуальних налаштувань, що досить важливо за масового виробництва.

Вони малогабаритні та надійні, економічні та багатофункціональні. Розвиток інфокомунікаційних технологій нерозривно пов'язаний з розвитком цифрової техніки.

СПИСОК ПРИЙНЯТИХ СКОРОЧЕНЬ

A – адреса, адресний вхід або вихід
AD – суматор
C – вхід тактових імпульсів
CD – кодер, шифратор
Ст – Counter – лічильник
CD – Data – дані, вхід даних
DB – двонаправлена шина даних
DC – дешифратор
EEPROM – репрограмований ПЗП з електричним стиранням даних
E – дозвіл, вхід дозволу
I – Input – вхід
MPL – програмований логічний пристрій
MUX – мультиплексор
PROM – програмований ПЗП
R – Reset – скид
RAM – пам'ять з довільною вибіркою (ОЗП)
RD – Read – читання
S – встановлення ВС у стан логічної «1»
T – Trigger – тригер
RG – регістр
WR – запис даних
АЦП – аналого-цифровий перетворювач
ВІС – велика інтегральна схема
ЗП – запам'ятовуючий пристрій
ІС – інтегральна схема
КМОП – комплементарна МОП-структура
ЛЕ – логічний елемент
МК – мікроконтролер
МП – мікропроцесор

МОН – метал–окисел–напівпровідник
МПС – мікропроцесорна система
БЕТ – багатоемітерний транзистор
ОЗП – оперативний ЗП
ПЗП – постійний ЗП
ПЛМ – програмована логічна матриця
РЗП – регістр загального призначення
НВІС – надвелика ІС
Z – третій стан ЛЕ
ТТЛ – транзисторно-транзисторна логіка
ТТЛШ – ТТЛ з діодами Шотткі
ЦАП – цифро-аналоговий перетворювач
ЦП – цифровий пристрій
ША – шина адреси
ШД – шина даних
ШК – шина керування
ШФ – шинний формувач
ЕЗЛ – емітерно-зв'язана логіка

1. КОМБІНАЦІЙНІ ПРИСТРОЇ

1.1. Загальні відомості

Комбінаційні пристрої – це автомати без пам'яті. В комбінаційних пристроях кожний символ на виході (логічний “0” або логічна “1”) визначається станом входів тільки на даний момент і не залежить від того, який рівень діяв раніше. Комбінаційні пристрої не зберігають відомості про минулу роботу пристрою.

До комбінаційних пристроїв належать:

- логічні елементи з різними логічними та функціональними можливостями, починаючи від самих простих до самих складних;
- шифратори;
- дешифратори;
- перетворювачі кодів;
- арифметико-логічні пристрої;
- цифрові суматори;
- мультиплексори;
- демультимплексори;
- цифрові компаратори;
- програмовані логічні матриці;
- драйвери.

1.2. Шифратори

Шифратори виконують перетворення (шифрування) алфавітно-цифрової або символної інформації, яку подано кодом “1 з n ”, наприклад, десятковим у більш компактний код, наприклад, двійковий.

Кодова комбінація на вході шифратора має тільки *один* активний стан однієї змінної x_i вхідного набору $\{x_1, x_2, \dots, x_{n-1}\}$. Вихід шифраторів, як правило, паралельний і складається з m розрядів. Число виходів m однозначно зв'язане з числом входів n . Якщо $n = 2^m$, тобто використаний повний набір вихідних і вихідних комбінацій, то такий шифратор називають *повним*. Наприклад, шифратор 8 – 3 є повним, бо він реалізує повний набір можливих комбінацій змінних x_i ($n = 8$) в повний вихідний набір y_i ($m = 3$), при цьому виконується рівність $2^3 = 8$.

У *неповному* шифраторі число входів n не відповідає числу всіх можливих вихідних комбінацій 2^m , причому завжди $n < 2^m$, що відповідно утворює певне число невикористаних вихідних наборів. Наприклад, шифратор 10 – 4, що використовується для кодування десяткового коду у двійково-десятковий код 8-4-2-1, є неповним, бо з можливого числа комбінацій $2^4 = 16$ використовується лише 10, а саме (0 – 9).

Шифратори поділяються на *прості* та *пріоритетні*.

Прості шифратори реалізують обов'язкову відповідність m -розрядного числа від тільки *одного* активного входу. Вони не допускають одночасної

активізації *декількох* входів, що може мати місце, наприклад, при натисканні декількох клавіш на клавіатурі, яка підключена до входу шифратора. Щоб шифратор реагував тільки на один активний вхід навіть за кількох активних входів, його схему будують за пріоритетним принципом.

Пріоритетний принцип побудови шифратора дозволяє одночасно активізувати декілька входів. Реакція шифратора при цьому може бути різною, бо будуються вони за трьома пріоритетними принципами.

Перші шифратори реагують лише на перший за часом сигнал, який надходить з будь-якого входу раніше за усіх.

Другі шифратори мають програмовані пріоритетні вхідні комбінації. Якщо на вхід такого шифратора надходить група символів серед яких один задовольняє умовам пріоритету, то шифратор буде працювати за алгоритмом, який означений пріоритетним символом. На зайві комбінації шифратор не реагує.

У пріоритетному шифраторі *третього* типу вихідне число завжди відповідає тому активному входу, який має найбільший номер набору. Наприклад, при вхідному числі {0111}, коли рівень логічної «1» одночасно присутній на перших трьох входах $x_1 = 1$, $x_2 = 1$, $x_4 = 1$, на виході пріоритетного шифратора установиться двійкове число {0100}, що відповідає змінній $x_4 = 1$, а активні входи x_1 та x_2 ігноруються.

За принципом побудови схеми прості шифратори поділяються на *лінійні* та *пірамідальні*. Лінійні шифратори будуються за лінійним принципом, коли всі однойменні входи логічних елементів підключають до однієї спільної шини. Тому для реалізації лінійного шифратора потрібно мати багатовходові логічні елементи, число входів яких дорівнює розрядності m . Лінійні шифратори мають досить високу швидкодію.

Розглянемо принцип побудови лінійного повного шифратора 8 - 3, стани входів і виходів якого описує таблиця станів (табл. 1.1).

Таблиця 1.1 – Стани повного лінійного шифратора 8 – 3

Число	Входи								Виходи		
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_3	y_2	y_1
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

З табл. 1.1 видно, що шифратор має бути побудованим на трьох (за кількістю виходів y) схемах логічного додавання АБО. Ці схеми повинні мати стільки входів, скільки логічних “1” у відповідному стовпчику y табл. 1.1.

Оскільки в кожному стовпчику по чотири логічних “1”, то схеми АБО мають бути чотиривхідними.

Для складання схеми знайдемо рівняння кожного виходу y_1, y_2, y_3 за допомогою табл. 1.1 наступним чином. Стан кожного виходу покажемо у вигляді суми станів тих входів x_i , які мають значення 1 у рядку з одиничним виходом y_k .

$$\begin{aligned} y_1 &= x_1 \vee x_3 \vee x_5 \vee x_7 = x_1 + x_3 + x_5 + x_7, \\ y_2 &= x_2 \vee x_3 \vee x_6 \vee x_7 = x_2 + x_3 + x_6 + x_7, \\ y_3 &= x_4 \vee x_5 \vee x_6 \vee x_7 = x_4 + x_5 + x_6 + x_7. \end{aligned} \quad (1.1)$$

Дійсно одиницю містять чотиридвійкові числа

$$001_2 = 1_{10}; \quad 011_2 = 3_{10}; \quad 101_2 = 5_{10}; \quad 111_2 = 7_{10}.$$

Для розробки схеми шифратора використовуємо *матрицю*, тобто систему ортогональних ліній (рис. 1.1). Матрична схема досить наочна, бо позбавлена від перегонів ліній зв'язку.

Входи саме шифратора зображені вертикальними лініями, а входи логічних схем АБО – горизонтальними.

Входи кожної схеми АБО мають бути з'єднані з тими входами шифратора, які є в функціях (1.1).

Роботу схеми шифратора пояснюють таблиця станів (табл. 1.1) і рівняння (1.1).

Схема працює наступним чином.

При вхідному числі 0 одиничним є лише один вхід шифратора $x_0 = 1$. При цьому решта входів нульова. Підставляючи всі $x_i = 0$ у рівняння (5.1), знаходимо, що $y_3 = 0; y_2 = 0; y_1 = 0$, тобто вихідне число становить 000.

При вхідному числі 1 одиничним є вхід шифратора $x_1 = 1$. При цьому решта входів нульова, через що $y_3 = 0; y_2 = 0; y_1 = 1$, тобто вихідне число становить 001.

Як видно з рівнянь (5.1), табл. 5.1 і схеми (рис. 5.1), взагалі на першому виході шифратора $y_1 = 2^0$ сигнал логічної “1” з'являється тоді, коли на один із входів або x_1 , або x_3 , або x_5 , або x_7 подано логічну “1”.

Другий вихід $y_2 = 2^1 = 1$ шифратора буде одиничним тоді, коли логічна «1» присутня на одному з входів або x_2 , або x_4 , або x_6 , або x_7 .

Самий старший вихідний розряд $y_3 = 2^2 = 4$ буде одиничним, якщо на одному з входів шифратора або x_4 , або x_5 , або x_6 , або x_7 діє логічна “1”.

Для лінійних шифраторів характерна незадіяна змінна x_0 . Це означає, що за будь-якого сигналу на вході x_0 , на виході шифратора не буде жодних змін.

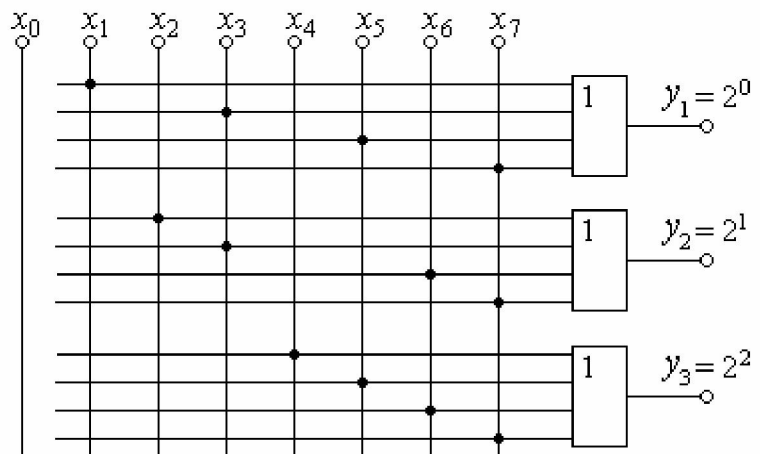


Рисунок 1.1 – Схема повного шифратора 8 - 3

Проте така ситуація на практиці завжди враховується і тому передбачається обов'язкова наявність активного входу x_0 , хоча він не використовується.

Для розглянутого шифратора (рис. 1.1) присутність двох або більшого числа активних входів заборонена і він відноситься до типу простих неперіоритетних лінійних шифраторів.

Шифратори значно використовуються в цифрових системах для перетворення вхідних десяткових чисел у двійкову форму. За допомогою шифраторів кодуються різні символи (в тому числі й літери). Серійні шифратори можна знайти у складі мікросхем багатьох серій. Пріоритетні шифратори найбільш поширені, бо на них, крім основних, можна виконувати функції простих шифраторів.

В інтегральній схемотехніці частіше зустрічаються шифратори, що перетворюють сигнали низького рівня на одному з інформаційних входів \bar{x}_i в обернений двійковий код на виході. Це дає можливість діагностики шифратора в початковому стані, в якому на виходах мають бути тільки логічні «1». Поява на будь-якому виході логічного «0» свідчить про несправність шифратора.

Умовне позначення неповного шифратора показано на рис. 1.2.

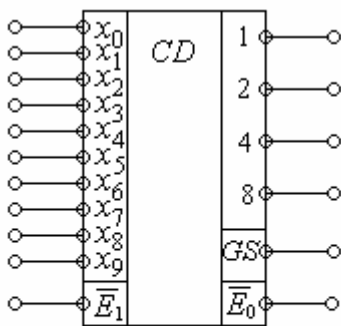


Рисунок 1.2 – Умовне позначення неповного шифратора 9 - 4

Крім інформаційних входів $\bar{x}_0 \dots \bar{x}_9$, шифратор має вхід дозволу \bar{E}_1 . При $\bar{E}_1 = 1$ можна змінити комбінацію на входах без зміни числа на виходах. \bar{E}_0 – вихід сигналу дозволу. При $\bar{E}_0 = 0$ фіксуються високі рівні на всіх інформаційних виходах.

Вихід групового сигналу \overline{GS} свідчить про наявність хоча б одного активного низького рівня на виході. За допомогою виходу \bar{E}_1 та виходу \bar{E}_0 можна нарощувати розрядність шифратора. При каскадному з'єднанні шифраторів слід вихід \bar{E}_0 попереднього каскаду з'єднати зі входом \bar{E}_1 наступного каскаду.

1.3. Дешифратори

Дешифраторами називаються комбінаційні логічні структури, які призначені для перетворення (дешифрації) коду, що надходить до входу в символну або алфавітно-цифрову інформацію, тобто в код «1 з n », в якому сигнал з'являється лише на одному з виходів. Дешифратори за порівнянням з шифраторами виконують обернене перетворення.

Дешифратори як і шифратори, бувають *повними* і *неповними*.

Розглянемо принцип побудови повного лінійного дешифратора 2 – 4 на логічних схемах І. Такий дешифратор має 4 двовходові схеми логічного множення І, які виявляють одиничні стани всіх входів.

Якщо вхідне число подається двійковим кодом, то на входах різних схем І буде комбінація прямих та інверсних вхідних змінних. Через це входи дешифратора повинні мати інвертори.

Знайдемо логічний вислів кожного виходу.

Вхідне число $x_2 = 0$; $x_1 = 0$ схема I може виявити, якщо до її входів підвести інверсні значення \bar{x}_2 та \bar{x}_1 . Тоді $y_0 = \bar{x}_2 \wedge \bar{x}_1 = \bar{x}_2 \cdot \bar{x}_1$. Таким чином, рівняння станів виходів дешифратора є логічними добутками станів входів, у які підставляємо прямий стан входу, якщо він одиничний, та інверсний, якщо він нульовий.

Отже стани дешифратора 2 – 4 на схемах I описуються наступними рівняннями:

$$\begin{aligned} y_0 &= \bar{x}_2 \wedge \bar{x}_1 = \bar{x}_2 \cdot \bar{x}_1, \\ y_1 &= \bar{x}_2 \wedge x_1 = \bar{x}_2 \cdot x_1, \\ y_2 &= x_2 \wedge \bar{x}_1 = x_2 \cdot \bar{x}_1, \\ y_3 &= x_2 \wedge x_1 = x_2 \cdot x_1. \end{aligned} \quad (1.2)$$

Схему дешифратора розробляємо у виді матриці (рис. 1.2). Входи схем 2I D3...D6 з'єднуємо з тими входами дешифратора, які є в рівняннях (1.2).

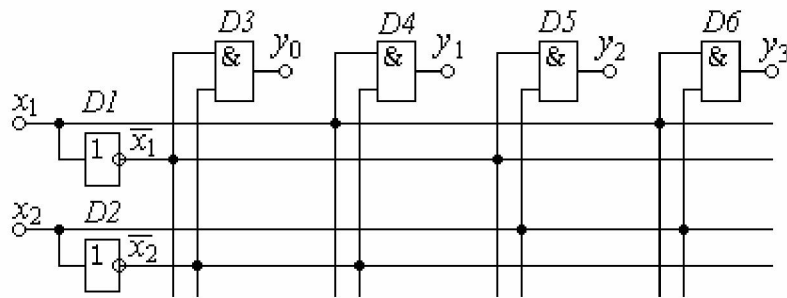


Рисунок 1.3 – Схема лінійного дешифратора 2 - 4

Для пояснення роботи дешифратора надамо таблицю його станів (табл. 1.2).

Дешифратор працює наступним чином.

Таблиця 1.2 – Стани повного дешифратора 2-4

Число	Входи		Виходи			
	x_2	x_1	y_0	y_1	y_2	y_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

Інвертори D1 та D2 здійснюють інверсію \bar{x}_1 та \bar{x}_2 .

Підставляючи стани входів з табл. 1.2 у рівняння (1.2), переконуємось у наступному.

При вхідному числі 0 обидва входи нульові: $x_2 = 0$ та $x_1 = 0$. Підставляючи ці значення в рівняння (1.2), отримуємо, що тільки на виході

схеми D3 вихід одиничний: $y_0 = 1$. Це пояснюється тим, що при $x_1 = 0$ та $x_2 = 0$ їхні інверсні стани одиничні, що й виявляє схема D3. На входах усіх інших схем D4, D5, D6 хоча б один з входів нульовий, через що $y_1 = 0$; $y_2 = 0$; $y_3 = 0$.

При вхідному числі 1 стани входів дешифратора $x_2 = 0$; $x_1 = 1$. Підставляючи ці значення в рівняння (1.2), переконуємось в тому, що тільки на виході схеми D4 вихід одиничний: $y_1 = 1$ і т.д.

Дешифратори можна здійснити і на інших логічних схемах, наприклад І-НЕ. Різниця буде лише в тому, що при обох одиничних входах на виході відповідної схеми буде логічний “0”.

Якщо дешифратор виконати на схемах 2АБО, то прямі та інверсні їхні входи слід змінити на протилежні. Тоді на виході схеми з двома вхідними нулями буде логічний “0”. На виходах інших трьох схем будуть логічні “1”.

Щодо неповних дешифраторів, то в них є певне число невикористаних вхідних наборів. Досить розповсюдженим прикладом є неповний дешифратор 4-10, який виконує перетворення двійково-десятькового коду в десятиковий.

Розглянемо принцип побудови неповного дешифратора 4 - 10 на схемах логічного додавання АБО. Такий дешифратор має 10 чотиривхідних схем логічного додавання АБО, кожна з яких виявляє нульові стани всіх своїх чотирьох входів. Тому рівняння кожного виходу є сумою станів входів, у яку підставляємо прямий стан входу, якщо він нульовий, та інверсний, якщо він одиничний.

Щоб активний рівень виходів дешифратора був одиничним, слід кожен згадану суму проінвертувати, для чого використовуємо логічні схеми АБО-НЕ.

Отже, логічний вислів станів дешифратора наступний:

$$\begin{aligned}
 y_0 &= \overline{x_4 \vee x_3 \vee x_2 \vee x_1} = \overline{x_4 + x_3 + x_2 + x_1}, \\
 y_1 &= \overline{x_4 \vee x_3 \vee x_2 \vee \bar{x}_1} = \overline{x_4 + x_3 + x_2 + \bar{x}_1}, \\
 y_2 &= \overline{x_4 \vee x_3 \vee \bar{x}_2 \vee x_1} = \overline{x_4 + x_3 + \bar{x}_2 + x_1}, \\
 y_3 &= \overline{x_4 \vee x_3 \vee \bar{x}_2 \vee \bar{x}_1} = \overline{x_4 + x_3 + \bar{x}_2 + \bar{x}_1}, \\
 y_4 &= \overline{x_4 \vee \bar{x}_3 \vee x_2 \vee x_1} = \overline{x_4 + \bar{x}_3 + x_2 + x_1}, \\
 y_5 &= \overline{x_4 \vee \bar{x}_3 \vee x_2 \vee \bar{x}_1} = \overline{x_4 + \bar{x}_3 + x_2 + \bar{x}_1}, \\
 y_6 &= \overline{x_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee x_1} = \overline{x_4 + \bar{x}_3 + \bar{x}_2 + x_1}, \\
 y_7 &= \overline{x_4 \vee \bar{x}_3 \vee \bar{x}_2 \vee \bar{x}_1} = \overline{x_4 + \bar{x}_3 + \bar{x}_2 + \bar{x}_1}, \\
 y_8 &= \overline{\bar{x}_4 \vee x_3 \vee x_2 \vee x_1} = \overline{\bar{x}_4 + x_3 + x_2 + x_1}, \\
 y_9 &= \overline{\bar{x}_4 \vee x_3 \vee x_2 \vee \bar{x}_1} = \overline{\bar{x}_4 + x_3 + x_2 + \bar{x}_1}.
 \end{aligned}
 \tag{1.3}$$

Схему дешифратора розробляємо у вигляді матриці (рис. 1.3). Входи схем 4АБО-НЕ, які мають виходи $y_0 \dots y_9$, з'єднуємо відповідно з тими входами дешифратора, які є в рівняннях (1.3).

На рис. 1.4 надане умовне позначення двійково-десятькового лічильника $CT2 - CT10$, двійкові виходи 1, 2, 4, 8 якого підключені відповідно до входів дешифратора x_1, x_2, x_3, x_4 .

Ці входи прямі чи інверсні з'єднані з чотирма входами схем АБО-НЕ згідно з функціями (1.3).

Так, входи елемента АБО-НЕ y_0 з'єднані з усіма прямими входами x_1, x_2, x_3, x_4 дешифратора.

Входи елемента АБО-НЕ y_1 з'єднані з прямими входами дешифратора x_2, x_3, x_4 та інверсним \bar{x}_1 і т.д., щоб сума станів входів для кожного виходу $y_0, \dots y_9$ дорівнювала б нулю.

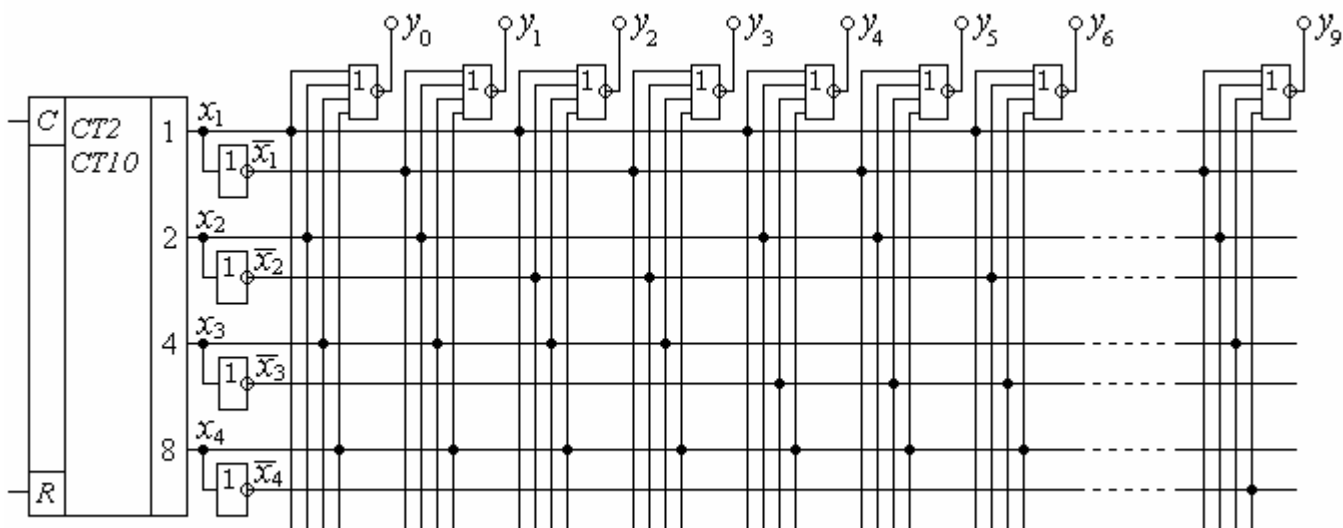


Рисунок 1.4 – Схема неповного дешифратора 4 - 10

Для пояснення роботи дешифратора надамо таблицю його станів (табл. 1.3).

Таблиця 1.3 – Стани неповного лінійного дешифратора 4 - 10

Число	Входи				Виходи									
	x_4	x_3	x_2	x_1	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9
0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	1	0	0	0	0	0	0
4	0	1	0	0	0	0	0	0	1	0	0	0	0	0
5	0	1	0	1	0	0	0	0	0	1	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0	1	0	0	0
7	0	1	1	1	0	0	0	0	0	0	0	1	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	1	0
9	1	0	0	1	0	0	0	0	0	0	0	0	0	1

Дешифратор працює наступним чином.

Як бачимо з табл. 1.3, при скинутому лічильнику на його виходах нулі: $x_1 = 0$; $x_2 = 0$; $x_3 = 0$; $x_4 = 0$. Підставляючи ці значення та їхні інверсії в усі рівняння (1.3), переконуємося в тому, що тільки $y_0 = 1$. Це пояснюється тим, що схема АБО-НЕ виявляє тільки нульові стани входів. Тому при $x_1 = 0$; $x_2 = 0$; $x_3 = 0$; $x_4 = 0$ під нулями тільки входи схеми y_0 , через що $y_0 = 1$. Решта усіх інших виходів нульова, бо хоча б один із входів інших схем АБО-НЕ одиничний.

Якщо лічильник містить число 1, то нульовими будуть усі входи схеми АБО-НЕ y_1 , через що тільки цей вихід буде одиничним: $y_1 = 1$ і т.д. Будь-якому вмісту лічильника в межах $0 \dots 9$ відповідає тільки одна одиниця на виходах y_0, \dots, y_9 .

Дешифратори можна здійснити і на інших логічних схемах, наприклад І-НЕ. Різниця буде лише в тому, що при всіх одиничних входах на виході відповідної схеми буде логічний "0".

Промисловість випускає дешифратори як повні (рис. 1.5,а), так і неповні (рис. 1.5,б).

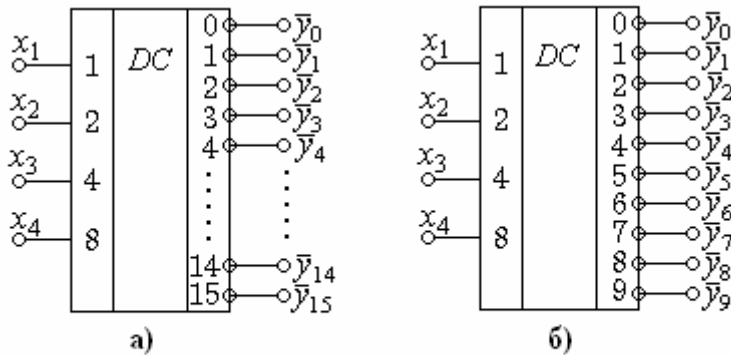


Рисунок 1.5 – Умовне позначення дешифраторів:
а) повного 4 - 16; б) неповного 4 - 10

Існування неповного дешифратора виправдано, наприклад, перетворенням двійково-десятькового коду в десятковий (рис. 1.5,б).

Слід зауважити, що в інтегральних дешифраторах формування інверсій входних змінних виконується у самому дешифраторі і тому додаткові інвертори на їхніх входах ставити не треба.

1.4. Перетворювачі кодів

Перетворювачі кодів призначені для перетворення кодової комбінації одного коду в кодову комбінацію іншого коду.

Виходом перетворювачів є довільний код, а не код "1 з n", який має місце у дешифраторах. На відміну від шифраторів на входи перетворювача коду теж може надходити будь-який набір. Різновидів цифрових кодів дуже багато і тому й перетворювачів кодів існує довільна множина.

Прикладом може бути перетворювач коду для керування цифровим індикатором.

Досить поширеним цифровим індикатором є семисегментний (рис. 1.6).

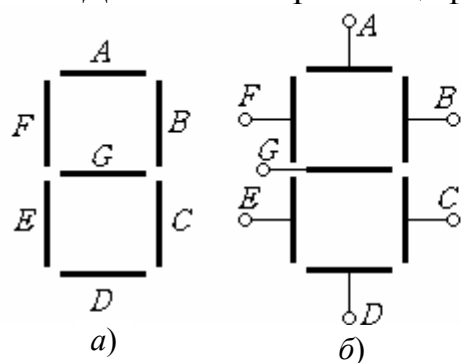


Рисунок 1.6 – Семисегментний індикатор: а) конфігурація; б) схема

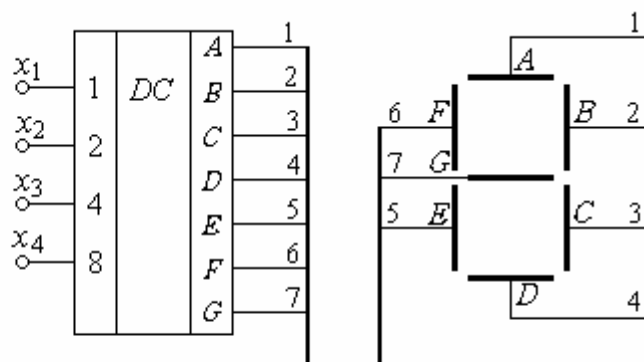


Рисунок 1.7 – Схема з'єднання перетворювача коду з цифровим індикатором

Кожний сегмент такого індикатора A, B, C, D, E, F, G є світлодіодом і може світитися в залежності від комбінації логічних змінних на виході перетворювача. Викликаючи висвітлення сегментів у певних комбінаціях, можна отримати зображення десяткових цифр 0, 1, ..., 9.

Для керування семисегментними індикаторами в інтегральній схемотехніці існують промислові перетворювачі *DC* (рис. 1.7), які перетворюють двійковий код 8-4-2-1 у семирозрядний, що спеціально призначений для керування семисегментним індикатором. На рис. 1.7 з'єднання перетворювача *DC* з семисегментним індикатором показано скруткою.

Роботу перетворювача видно з таблиці станів (табл. 1.4), яка складена за умови, що активним сигналом є логічна "1".

Перетворювач *DC* з індикатором працюють наступним чином.

Таблиця 1.4 – Стани перетворювача коду 4 - 7

Число	Входи				Виходи						
	x_4	x_3	x_2	x_1	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>	<i>E</i>	<i>F</i>	<i>G</i>
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Якщо на входи перетворювача надходить число 0000, то індикатор має висвітлити цифру 0. Тому при числі 0 (див. табл. 1.4) усі сегменти одиничні, крім $G = 0$.

При вхідному числі 1, тобто 0001 мають світитися тільки два сегменти: *B* і *C*. Тому $B = 1$ і $C = 1$. Усі інші сегменти нульові і т.д.

Якщо подане число 8, тобто 1000, то мають світитися усі сегменти, для чого вони всі одиничні.

Число 9 (1001) викликає одиницю на всіх сегментах, крім $E = 0$.

Промисловістю випускаються серійні перетворювачі також для інших типів індикаторів, наприклад, матриці 7×5 точок або матриці 7×4 точки; лінійних індикаторів амплітуди зі шкалами у вигляді світлового стовпця, рухомої точки, пари точок тощо.

За традиціями перетворювачі цієї групи іноді називають дешифраторами і мікросхеми, які реалізують ці функції, позначають символами ІД, як звичайні дешифратори.

Все більшого поширення набувають перетворювачі двійково-десятькового коду в коди латинської та російської абетки, а також у код додаткових символів-знаків.

Такий інтерес до цих мікросхем пояснюється розвитком виробництва комп'ютерів, де найбільш часто використовуються розглянуті перетворювачі.

1.5. Мультиплексори

Мультиплексори належать до пристроїв комутування цифрової інформації. Вони здійснюють комутацію одного з декількох інформаційних входів x_i до одного виходу y . Мультиплексори мають декілька інформаційних входів, адресні входи, вхід дозволу мультиплексування (стробуючий вхід) та один вихід.

Кожному з інформаційних входів мультиплексора відповідає номер, який називається адресою і подається двійковим числом до адресних входів.

На рис. 1.8 представлений варіант схеми мультиплексора 4 - 1, який виконує комутацію від чотирьох джерел сигналу x_0, x_1, x_2, x_3 , до одного виходу y .

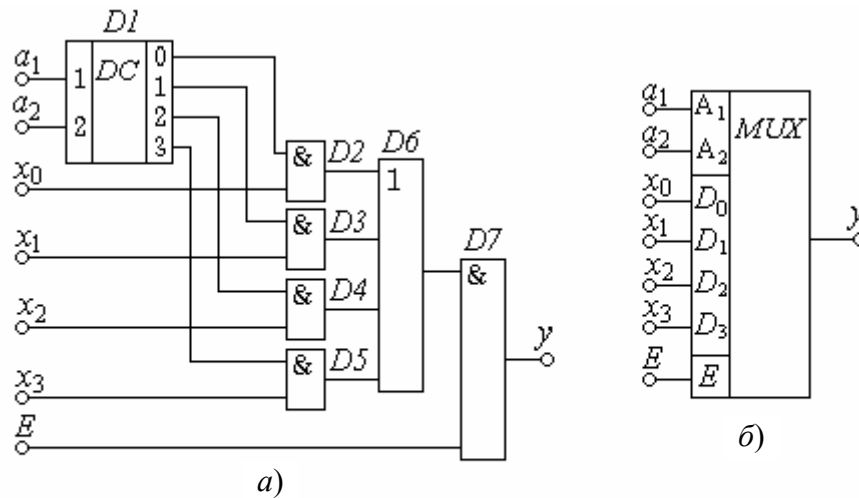


Рисунок 1.8 – Мультиплексор 4 - 1: а) схема; б) умовне позначення

Очевидно, що число інформаційних входів $n_{\text{інф}}$ і число адресних входів $n_{\text{адр}}$ зв'язані співвідношенням: $n_{\text{адр}} = 2^{n_{\text{інф}}}$.

Комутація єдиного з усіх входів до виходу здійснюється тим, що кон'юнктори $D2 \dots D5$ виконують роль ключів. Такий ключ буде відкритий якщо обидва його входи одиничні. Оскільки з усіх виходів дешифратора адреси $D1$ тільки один одиничний, за будь-якої адреси відкрита тільки одна зі схем $D2 \dots D5$.

Виходячи з цього, отримуємо логічний вислів мультиплексора

$$y = x_0 \cdot \bar{a}_2 \cdot \bar{a}_1 \vee x_1 \cdot a_2 \cdot \bar{a}_1 \vee x_2 \cdot \bar{a}_2 \cdot a_1 \vee x_3 \cdot a_2 \cdot a_1. \quad (1.4)$$

Для пояснення роботи мультиплексора надамо таблицю його станів (табл. 1.5).

Таблиця 1.5 – Стани мультиплексора 4 – 1

Адресні входи		Керуючий вхід E	Вихід y
a_2	a_1		
X	X	0	0
0	0	1	x_0
0	1	1	x_1
1	0	1	x_2
1	1	1	x_3

Мультиплексор працює наступним чином.

Дозвіл на спрацьовування мультиплексора в цілому визначається станом керуючого входу E . При $E = 0$, незалежно від стану входів адреси a_2 та a_1 (в табл. 1.5 позначено X), мультиплексор не спрацьовує. Якщо ж $E = 1$, то робота мультиплексора дозволяється. Тому надалі розглядаються тільки чотири нижні рядки табл. 1.5.

При адресному числі 00 стан адресних входів $a_2 = 0$ та $a_1 = 0$. Підставляючи ці значення в (1.4), отримуємо $y = x_0$, тобто до виходу передається тільки інформація x_0 . Це пояснюється тим, що рівень логічної "1" установлюється тільки на виході 0 дешифратора $D1$, а на всіх інших виходах $D1$

він нульовий. Тому лише логічний елемент $D2$ має дозвіл на спрацьовування. При цьому на виході мультиплексора буде інформація $y = x_0$ (або 0, або 1).

При адресному числі 01 $a_2 = 0$ та $a_1 = 1$. Підставляючи ці значення в (1.4), отримуємо $y = x_1$, тобто до виходу передається тільки інформація x_1 . Це пояснюється тим, що рівень логічної "1" устанавлюється тільки на виході 1 дешифратора $D1$, а на всіх інших виходах $D1$ він нульовий. Тому лише логічний елемент $D3$ має дозвіл на спрацьовування. При цьому на виході мультиплексора буде інформація $y = x_1$ (або 0, або 1) і т.д.

При комутації багаторозрядних слів використовують декілька мультиплексорів, виходи яких з'єднуються за схемою АБО. Для цієї мети випускаються декілька однотипних мультиплексорів в одному корпусі.

1.6. Демультимплексори

Демультимплексори як і мультиплексори відносяться до пристроїв комутації цифрової інформації. Вони здійснюють комутацію одного інформаційного входу до одного з декількох виходів, адреса якого задана. Демультимплексори мають один інформаційний вхід, декілька виходів та адресні входи.

Таким чином, на приймальному кінці мультиплексованої магістралі потрібно виконати зворотну операцію – демультимплексування.

Демультимплексор можна реалізувати за тією ж схемою, що й для мультиплексора (рис. 1.8), якщо всі інформаційні входи x_0, x_1, x_2, x_3 з'єднати в один вхід x (рис. 1.9).

Варіант побудови схеми демультимплексора 1 - 4 на базі дешифратора $D1$ та логічних елементів $D2 \dots D5$ показаний на рис. 5.9.

Входи дешифратора a_1, a_2 є адресними. Тому в залежності від адресного числа лише на одному з виходів дешифратора з'являється логічна "1", яка дає дозвіл до спрацьовування лише одного з чотирьох кон'юнкторів $D2 \dots D5$. На другі входи кожного кон'юнктора надходить шина вхідного сигналу x .

Вхідна інформація відтворюється на виході одного з чотирьох логічних елементів $D2 \dots D5$, який отримав дозвіл відповідно адресному входу.

Виходячи з цього, отримуємо логічний вислів демультимплексора:

$$\begin{aligned} y_0 &= x \bar{a}_2 \bar{a}_1, \\ y_1 &= x a_2 \bar{a}_1, \\ y_2 &= x \bar{a}_2 a_1, \\ y_3 &= x a_2 a_1. \end{aligned} \quad (1.5)$$

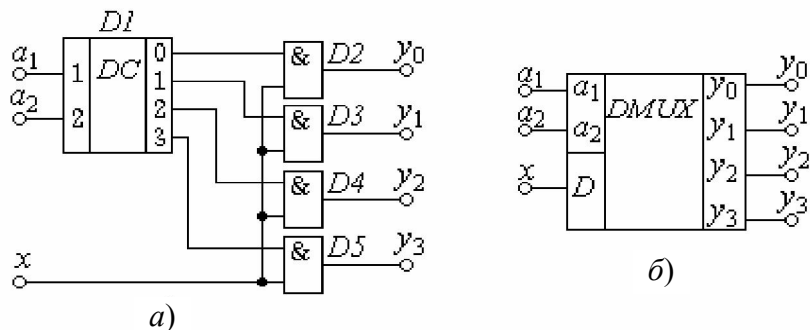


Рисунок 1.9 – Демультимплексор 1 - 4: а) схема; б) умовне позначення

Для пояснення роботи демультиплексора надамо таблицю його станів (табл. 1.6)

Таблиця 1.6 – Стани демультиплексора 1 - 4

Адресні входи		Виходи			
a_2	a_1	y_0	y_1	y_3	y_4
0	0	x	0	0	0
0	1	0	x	0	0
1	0	0	0	x	0
1	1	0	0	0	x

Демультиплексор працює наступним чином.

Підставляючи в (1.5) з табл. 1.6 адресу $a_2 = 0; a_1 = 0$, переконуємося в тому що тільки $y_0 = x$, тобто вхідний сигнал x діє тільки на одному виході y_0 . Це пояснюється тим, що при адресі $a_2 = 0; a_1 = 0$ тільки вихід 0 дешифратора $D1$ одиничний. Тому для вхідного сигналу x відкритий тільки кон'юнктор $D2$.

При адресі $a_2 = 0; a_1 = 1$ таким самим чином одержуємо $y_1 = x$ і т.д.

Стани виходів для інших адрес видно з табл. 1.6.

1.7. Цифрові компаратори

Цифрові компаратори – це пристрої, що призначені для порівняння двох чисел.

Розглянемо два цілих двійкових числа A і B . При їхньому порівнянні можуть бути три варіанти: $A = B; A > B$ і $A < B$.

Перший випадок ($A = B$) реалізується в найпростіших компараторах, які іноді носять назву нуль-органів. Такий компаратор виявляє лише факт рівності або нерівності двох поданих на його вхід чисел A і B і формує на виході сигнал рівності (логічну "1"), або нерівності – логічний "0". Функцію, яку реалізує компаратор у цьому випадку можна визначити формулою

$$y(A = B) = \begin{cases} 1, & \text{коли } A = B, \\ 0, & \text{коли } A \neq B. \end{cases} \quad (1.6)$$

Числа A і B можуть мати n розрядів і подаватися відповідно значенням змінних a_1, \dots, a_n та b_1, \dots, b_n . Компаратор рівності порівнює окремі розряди за формулою (1.6), яка записана для варіанта $n = 2$

$$y(A = B) = \bar{a}_1 \bar{a}_2 \bar{b}_1 \bar{b}_2 \vee \bar{a}_1 a_2 \bar{b}_1 b_2 \vee a_1 \bar{a}_2 b_1 \bar{b}_2 \vee a_1 a_2 b_1 b_2. \quad (1.7)$$

Найпростіший компаратор рівності реалізується за допомогою логічних схем "Виняткове АБО" (елементи $D1$ і $D2$) та багатовхідного кон'юнктора $D3$ (рис. 1.10).

На виході $D1$ і $D2$ (виняткове АБО з інверсією виходу) рівень логічної "1" з'являється тоді, коли $a_1 = b_1$ та $a_2 = b_2$, незалежно від порівняння логічних "1" або логічних "0".

Тому на виході y компаратора логічна "1" присутня лише у випадку, коли набули рівності обидва розряди двійкових чисел A і B .

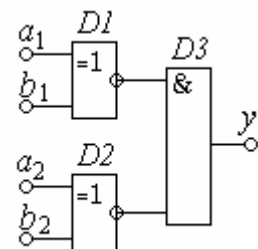


Рисунок 1.10 – Компаратор рівності

Функціонування схеми (рис. 1.10) пояснюється таблицею станів (табл. 1.7).

У другому випадку компаратори можуть відрізнити не тільки рівність чисел A і B , але й нерівність, коли $A > B$ або $A < B$. Такі компаратори будуються за більш складною схемою.

Вони визначають стани нерівностей і описуються системою нерівностей

$$y_1 = \begin{cases} 1, & \text{коли } A < B; \\ 0, & \text{коли } A \geq B; \end{cases} \quad (1.8)$$

$$y_2 = \begin{cases} 1, & \text{коли } A > B; \\ 0, & \text{коли } A \leq B. \end{cases}$$

Нерівнісні компаратори мають два виходи: y_1 та y_2 . Рівень логічної "1" з'являється на виході y_1 при $A < B$, а на виході y_2 – при $A > B$. Порівняння виконується порозрядно за формулою (1.9), яка записана для випадку $n = 2$.

$$y_1(A < B) = \bar{a}_1 \bar{a}_2 \bar{b}_1 b_2 \vee \bar{a}_1 \bar{a}_2 b_1 \bar{b}_2 \vee \bar{a}_1 \bar{a}_2 b_1 b_2 \vee \bar{a}_1 a_2 \bar{b}_1 \bar{b}_2 \vee \bar{a}_1 a_2 b_1 \bar{b}_2 \vee \bar{a}_1 a_2 b_1 b_2;$$

$$y_2(A > B) = \bar{a}_1 a_2 \bar{b}_1 \bar{b}_2 \vee \bar{a}_1 a_2 \bar{b}_1 b_2 \vee \bar{a}_1 a_2 b_1 \bar{b}_2 \vee \bar{a}_1 a_2 b_1 b_2 \vee a_1 \bar{a}_2 \bar{b}_1 \bar{b}_2 \vee a_1 \bar{a}_2 \bar{b}_1 b_2 \vee a_1 \bar{a}_2 b_1 \bar{b}_2 \vee a_1 \bar{a}_2 b_1 b_2;$$
(1.9)

Промисловість випускає цифрові повні багаторозрядні компаратори, які мають і вихід рівності $A = B$, і виходи нерівностей $A < B$ та $A > B$.

Умовне позначення такого компаратора наведено на рис. 1.11. Входи аналізу призначені для каскадування компараторів, тобто для збільшення розрядності. При каскадному з'єднанні компараторів входи аналізу з'єднуються з однойменними виходами попереднього компаратора.

1.8. Програмовані логічні матриці

1.8.1. Матрична схема

Матрична схема або логічна матриця являє собою сітку ортогональних провідників, у перетинах яких можуть бути установлені напівпровідникові елементи: діоди або транзистори, що реалізують необхідну схему.

Для можливості програмування матриці ці діоди або транзистори підключаються до відповідних провідників матриці через легкоплавкі перемички. Під час програмування ці перемички або перепалюють, або

Таблиця 1.7 – Стани компаратора рівності

Входи				Вихід
a_1	a_2	b_1	b_2	y
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	0
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

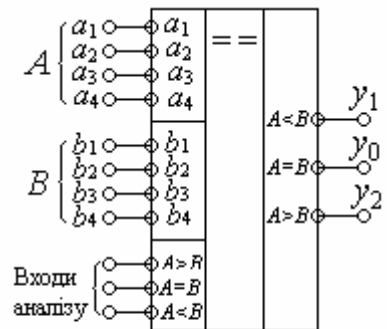


Рисунок 1.11 – Умовне позначення повного компаратора

залишають в залежності від схеми, яку треба реалізувати за допомогою матриці. Матричні схеми орієнтовані на реалізацію булевих функцій вузлів великих інтегральних схем.

Отже, з матричної структури шляхом її програмування отримують заданий комбінаційний пристрій. Тому такі структури називаються *комбінаційними програмованими логічними матрицями* (ПЛМ).

ПЛМ, як правило, мають *два схемотехнічних рівні*: на *першому* з них утворюються потрібні *кон'юнкції*, а на *другому* – *диз'юнкції*. Іноді одна з матриць може бути фіксованою. Обидві матриці з'єднуються каскадно.

Існують і *послідовні* ПЛМ, які у своєму складі містять певне число вбудованих елементів пам'яті. Такі ПЛМ характеризуються розрядністю регістра пам'яті.

Розглянемо програмовану логічну матрицю, що показана на рис. 1.12 і містить матриці M1 та M2.

Матриця M1 має три входи x_1, x_2, x_3 , шість горизонтальних і чотири вертикальних шин. Горизонтальні шини x_1, x_2, x_3 в M1 є вхідними, а вертикальні $P_1 - P_4$ – вихідними.

Виходи P носять назву *ліній терм*.

Кожна вхідна шина x_1, x_2, x_3 зв'язана з однією горизонтальною шиною безпосередньо, а з іншою – через інвертор (шини $\bar{x}_1, \bar{x}_2, \bar{x}_3$).

Спосіб включення напівпровідникових елементів (наприклад, діодів) у перетинах, які позначені хрестиками, дозволяє реалізувати на будь-якому з виходів, будь-яку *кон'юнкцію* її вхідних змінних, тобто функцію I.

Матриця M1 працює як *дешифратор*, виходами якого є кон'юнктивні терми $P_1 \dots P_4$.

Дійсно, при появі одиниці (високого рівня напруги) на одному з входів x_i , на виході відповідного інвертора з'являється \bar{x}_i (рівень логічного "0"). Якщо перемички на обох горизонтальних лініях x_i та \bar{x}_i присутні, то високий потенціал лінії x_i закриє "свій" діод і на відповідній вертикальній лінії терм буде високий рівень напруги $\approx +E$, якщо на всіх інших горизонтальних лініях буде також рівень логічної "1".

Низький рівень лінії \bar{x}_i відкриє "свій" діод і практично весь струм вертикальної шини потече через горизонтальну лінію, залишаючи вертикальну шину без струму, що рівнозначно логічному "0".

Зберегти високий рівень логічної "1" на вертикальній шині P_N можна лише тоді, коли всі діоди на цій лінії будуть закриті, що можливо тільки при високих рівнях логічної "1" на всіх горизонтальних лініях, до яких підключена вертикальна шина P_N .

Таким чином, на вертикальній лінії P_N реалізується кон'юнкція змінних. Матриця кон'юнкцій M1 має перемички в місцях, позначених хрестиками, і реалізує функції кон'юнкції за формулами:

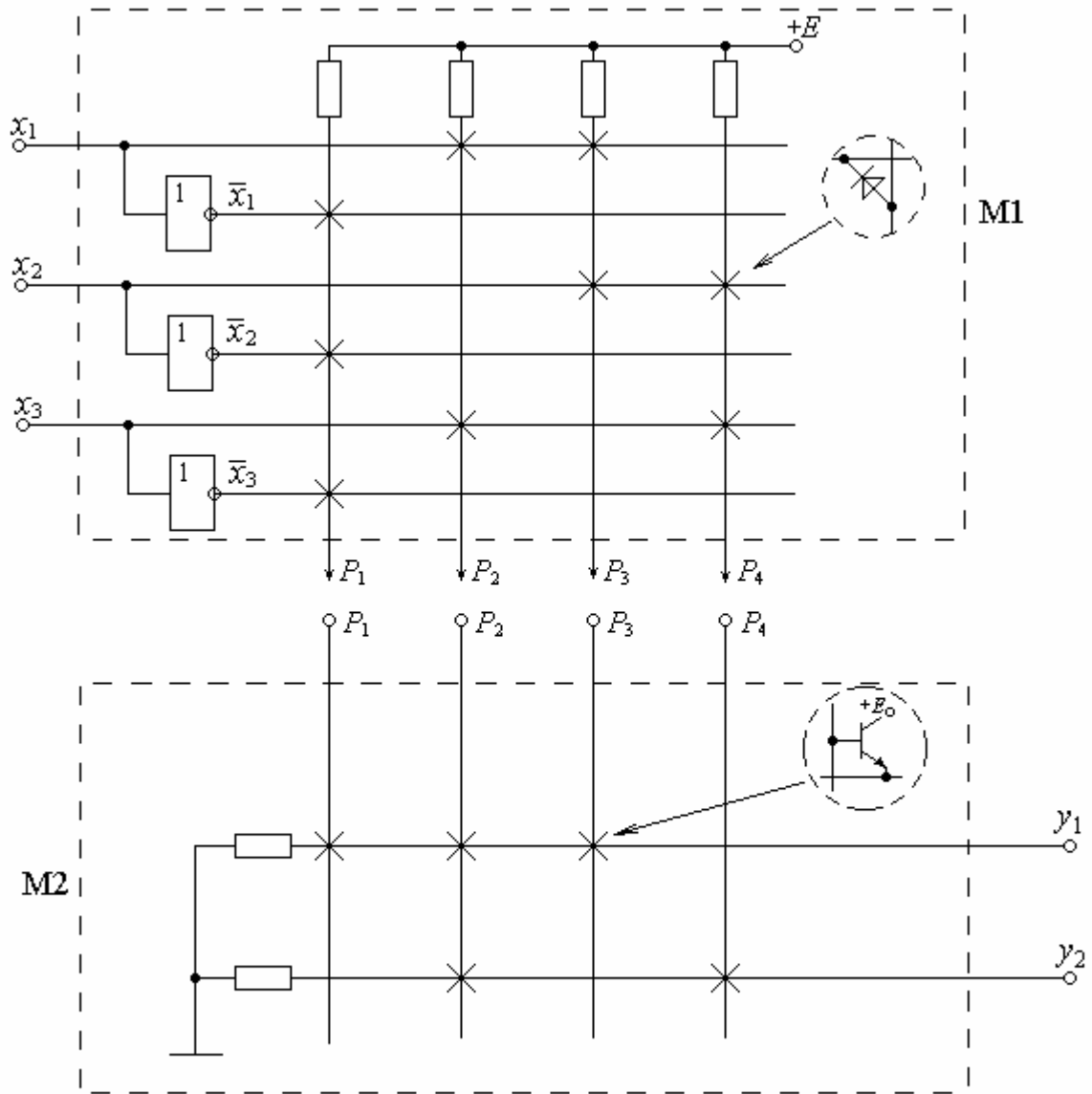


Рисунок 1.12 – Програмована логічна матриця: M1 – матриця кон'юнкцій;
M2 – матриця диз'юнкцій

$$\begin{aligned}
 P_1 &= \bar{x}_1 \bar{x}_2 \bar{x}_3; \\
 P_2 &= x_1 x_3; \\
 P_3 &= x_1 x_2; \\
 P_4 &= x_2 x_3.
 \end{aligned}
 \tag{1.10}$$

Матриця M2 має чотири вертикальні $P_1 - P_4$ шини і дві горизонтальні. Спосіб включення транзисторів у перетинах шин дозволяє реалізувати на будь-якому з її виходів будь-яку диз'юнкцію (функцію АБО) вхідних змінних. У матриці M2 вхідними є вертикальні шини, а вихідними – горизонтальні, на якій показаний приклад реалізації елементарних диз'юнкцій, що описується математичним виразом

$$\begin{aligned}
 y_1 &= P_1 \vee P_2 \vee P_3; \\
 y_2 &= P_2 \vee P_4.
 \end{aligned}
 \tag{1.11}$$

Дійсно, на матриці диз'юнкцій M2 транзистор на позначеному хрестиком перетині ортогональних ліній буде відкритим лише за наявності струму у вертикальній лінії, а за відсутності струму – закритий. Для матриці M2 характерно те, що на її горизонтальній лінії досить хоча б одного відкритого транзистора, щоб на виході y_j був високий рівень логічної “1”. Тільки відсутність струмів на всіх горизонтальних лініях забезпечує на ній логічний “0”, що відповідає операції диз'юнкції.

Отже, в наведеному прикладі (рис. 1.11) реалізується функція, яка описується формулами (5.11).

Якщо з'єднати обидві матриці M1 та M2, як показано на рис. 1.13, то отримана схема буде реалізувати систему булевих функцій

$$\begin{aligned} y_1 &= P_1 \vee P_2 \vee P_3 = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee x_1 x_3 \vee x_1 x_2 \\ y_2 &= P_2 \vee P_4 = x_1 x_3 \vee x_2 x_3. \end{aligned} \quad (1.12)$$

Побудова схем з матричною структурою зводиться до визначення точок перетину шин, де повинні бути включені напівпровідникові прилади.

Реалізації системи функцій матрицями M1 та M2 можна зобразити у вигляді таблиці програмування (табл. 1.8).

У табл. 1.8, стовпці якої відзначені змінними x_1, \dots, x_L та функціями y_1, \dots, y_N , кожній проміжній шині P_1, \dots, P_B ставиться у відповідність рядок таблиці (B – кількість вертикалів).

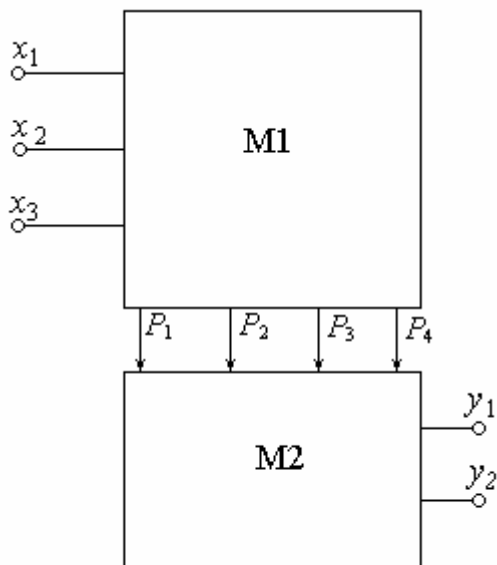


Рисунок 1.13 – Дворівнева матриця

Таблиця 1.8 – Таблиця програмування ПЛМ

Входи			Виходи	
x_1	x_2	x_3	y_1	y_2
0	0	0	1	.
1	–	1	1	1
1	1	–	1	.
–	1	1	.	1

На перетині j -го рядка і стовпця x_l ($l = \overline{1, L}$) записуються:

- 1, якщо змінна x_e входить в j -ту елементарну кон'юнкцію без інверсії;
- 0, якщо змінна x_e входить в j -ту елементарну кон'юнкцію з інверсією;
- риска (–), якщо змінна x_e не входить в j -ту елементарну кон'юнкцію.

На перетині j -го рядка і стовпця y_n ($n = \overline{1, N}$) записуються:

- 1, якщо j -та елементарна кон'юнкція входить в диз'юнкцію y_n ;
- крапка (.), якщо j -та елементарна кон'юнкція не входить в диз'юнкцію

y_n .

Відповідним чином будь-яка система булевих функцій y_1, \dots, y_N вхідних змінних x_1, \dots, x_L може бути реалізована дворівневою матричною схемою, на першому рівні якої утворюються різні елементарні кон'юнкції P_1, \dots, P_B , а на другому – диз'юнкції y_1, \dots, y_N відповідних кон'юнкцій. В результаті побудова схеми з матричною структурою зводиться до визначення точок перетину шин, де повинні бути включені діоди або транзистори.

Складність матричної реалізації оцінюється сумарною ємністю (площею) матриць. У загальному випадку у схемі, що має L входів, N виходів та B вертикалей площа становить

$$S(M) = S(M_1) + S(M_2) = 2LB + BN. \quad (1.13)$$

1.8.2. Дворівневі та тривірневі ПЛМ

На рис. 1.14 показана дворівнева ПЛМ, яка складається з двох матриць: M_1 та M_2 .

Матриця M_1 має S входів і q виходів. Вона дозволяє реалізувати q елементарних кон'юнкцій P_1, \dots, P_q змінних x_1, \dots, x_S , які надходять на її входи.

Матриця M_2 має q входів і t виходів. Вона дозволяє реалізувати t елементарних диз'юнкцій y_1, \dots, y_t змінних P_1, \dots, P_q , які надходять на її входи з виходів матриці M_1 .

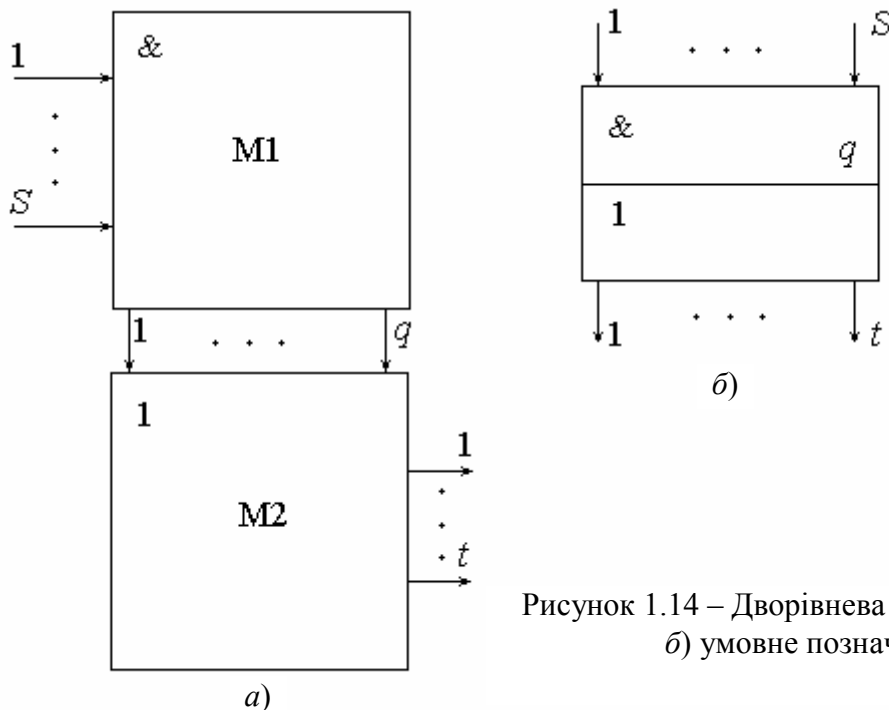


Рисунок 1.14 – Дворівнева ПЛМ: а) схема; б) умовне позначення

Виходи матриці M_1 з'єднані з входами матриці M_2 й утворюють проміжні шини $1-q$ ПЛМ. ПЛМ, що має S входів, t та q проміжних шин називається ПЛМ (S, t, q) .

Різновидом ПЛМ (S, t, q) є їхня удосконалена модель: ПЛМ (Z, q) . В ПЛМ (Z, q) фіксується лише два параметри: підсумоване число входів і виходів $Z = S + t$ та число проміжних шин q . Параметри S і t можуть набувати будь-яких конкретних значень при програмуванні.

Трирівнева ПЛМ комбінаційного типу (рис. 1.15) на відміну від дворівневої (рис. 1.14) має додатковий S -вхідний блок Д.

Число виходів блока Д дорівнює числу h горизонтальних шин у матриці М1.

Кожний i -й вихід з'єднаний з i -ю горизонтальною шиною цієї матриці ($i = \overline{1, h}$).

Блок Д може мати різноманітну внутрішню структуру. Наприклад, він може відігравати роль матриці M_0 , яка дозволяє утворювати h елементарних диз'юнкцій змінних x_1, x_2, \dots , які надходять на її входи і може використовуватися або з інверсією, або без інверсії. Найбільш часто блок Д складений з набору $S/2$ двохвхідних повних дешифраторів. Такі матриці позначаються ПЛМД (S, t, q) . Для ПЛМД (S, t, q) блок Д має S входів і $2S$ виходів, а число горизонтальних шин матриці М1 кратне чотирьом.

Входи кожного дешифратора – це входи блока Д, а чотири виходи кожного i -го з них з'єднані з відповідними чотирма горизонтальними шинами i -ї групи матриці. Використання ПЛМД на відміну від ПЛМ дозволяє значно зменшити число проміжних шин для реалізації заданої системи булевих функцій, тобто реалізувати більш складні системи. Площа матриць М1 і М2 за цього використовується значно раціональніше.

Подальша робота, яка направлена на збільшення ефективності використання площі матриць ПЛМ привела до створення структури, яка має наступні чотири особливості:

- матриця поділена на дві частини – M'_1 і M''_1 . Матриця M'_1 розміщена над матрицею M_2 , а матриця M''_1 – під матрицею M_2 . Це дозволяє за необхідності розрізати проміжні шини в матриці M_2 та реалізувати на верхній та нижній частинах однієї проміжної шини різні елементарні кон'юнкції вхідних змінних;

- входи матриць M'_1 та M''_1 розміщені з двох боків ліворуч та праворуч; кожна горизонтальна шина M'_1 або M''_1 розрізається в одному місці і на одну її частину подається змінна з лівого боку матриці, а на другу – з правого;

- виходи матриці M_2 розміщені з двох боків (зліва та справа). Кожна горизонтальна шина M_2 розрізається в одному місці і на першій її частині формується значення функції для лівого виходу M_2 , а на другій – для правого;

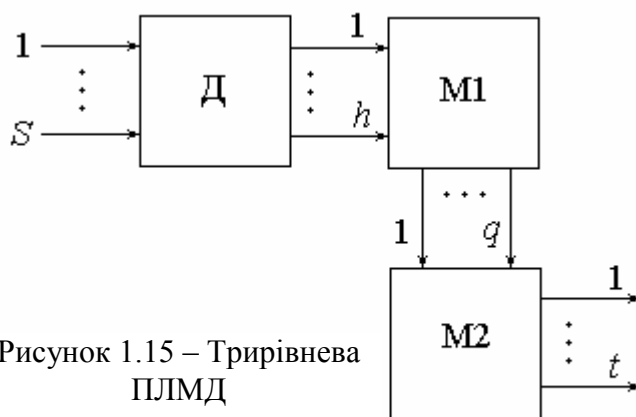


Рисунок 1.15 – Трирівнева ПЛМД

– на кристалі ВІС ПЛМ передбачена спеціальна система шин, яка дозволяє з'єднувати виходи першої матриці з входами другої.

Розріз шин і організація необхідних зв'язків між входами та виходами різних матриць виконується на етапі налагоджування ПЛМ на виробництві.

Завдяки значним успіхам інтегральної технології з'явилася можливість реалізації ВІС з “жорсткою” та “гнучкою” структурами. Особливий практичний інтерес виявляється до програмованих гнучких структур. Серед них розрізняють ВІС, які придатні до *програмування на етапі виготовлення*, і ВІС, які можуть бути *програмовані користувачем*.

Перші з них – це мікросхеми, які називають незавершеними логічними матрицями. Програмують такі ПЛМ *програмою замовника на виробництві* на стадії занесення програми в матрицю в процесі її виготовлення шляхом металізації ділянок матриці через спеціальну маску-шаблон. Після виготовлення інтегральної схеми з використанням маскових технологій перепрограмування неможливе.

Другі ПЛМ поділяють на дві групи: ВІС, які можуть бути *одноразово запрограмовані замовником або користувачем* – це прості програмовані логічні матриці, а ВІС з *багаторазовим перепрограмуванням* – це *перепрограмовані* логічні матриці.

Більш зручними є перепрограмовані ЛМ (ППЛМ). Це стандартні готові мікросхеми – напівфабрикати, в яких активні елементи на початковій стадії включені на всіх перетинах матриць через ніхромові перемички. Такі ВІС програмують самі користувачі за допомогою спеціальних програматорів шляхом електродного випалювання перемичок імпульсом струму. При цьому, якщо змінна x_i входить у терм P_i в прямій формі, перепалюють перемичку, що з'єднує терм P_i з її інверсією \bar{x}_i , і навпаки. Якщо змінна x_i та її інверсія \bar{x}_i не входять у терм P_i , перепалюють обидві перемички.

На рис. 1.16 показане умовне позначення мікросхеми ППЛМ, де вхід FE використовується тільки при програмуванні і називається *вхід дозволу перепалювання*, а вхід \overline{CE} – *дозвіл вибірки*.

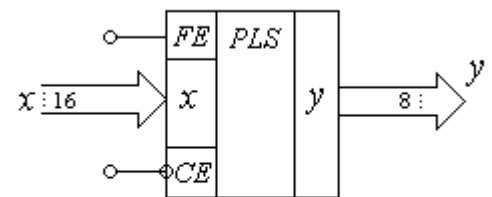


Рисунок 1.16 – Умовне позначення ППЛМ

Якщо на вході цієї ВІС ПЛМ повне число можливих комбінацій складає $2^{16} = 65536$, то на виході буде лише 48 різних вихідних комбінацій. Тому з усіх вхідних комбінацій невикористаних буде $65536 - 48 = 65488$. Недоліком ПЛМ є обмежені функціональні можливості.

На базі ПЛМ можна будувати комбінаційні цифрові пристрої на досить простих принципах.

Основним етапом розробки є складання таблиці програмування ПЛМ, яка є інструкцією процедури перепалювань непотрібних перемичок у матрицях.

З метою розширення функціональних можливостей використання ПЛМ промисловістю випускаються її спрощені варіанти, наприклад, програмовані матриці вентилів, програмовані матриці логіки, програмовані мультиплектори і т.д.

Будь-яка система Q з N булевих функцій L змінних, що описує поведінку комбінаційної схеми, може бути тривіально реалізована на одній ПЛМ (s, t, q) , якщо виконуються вимоги $L < s, N < t, B < q$.

1.9. Особливості роботи комбінаційних пристроїв

Однією з особливостей роботи комбінаційних пристроїв є затримка ними вихідних сигналів відносно вхідних.

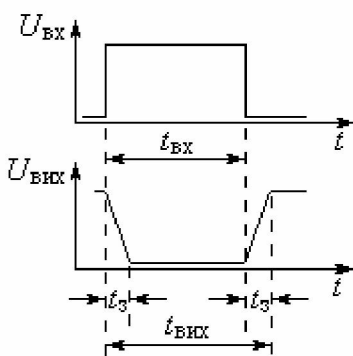


Рисунок 1.17 – Вхідний та вихідний сигнали інвертора

Дійсно, кожний логічний елемент, з яких складається будь-яка схема пристрою, має власну скінчену затримку. Тому сигнали на виході комбінаційного пристрою з'являються тільки через певний інтервал часу після подачі сигналів на вхід.

На рис. 1.17 показана часова діаграма роботи інвертора, з якої видно, що вихідний вхідний сигнал $U_{\text{ВЫХ}}$ зазнає часової затримки t_3 . Через неї перш за все знижується швидкодія, бо тривалість вихідного сигналу більше за вхідного: $t_{\text{ВЫХ}} > t_{\text{ВХ}}$, а по-друге, створюються хибні сигнали.

Поява хибних сигналів з'ясовується наступним фактором. Якщо на входи комбінаційного пристрою подаються одночасно кілька сигналів, то внаслідок часових затримок утворюються *логічні змагання* або "гонки". На виході такого комбінаційного пристрою можуть з'являтися нерегламентовані сигнали. Хоча ці сигнали, як правило, короткочасні, поява їх на виході призводить до збою працездатності схем, підключених до виходу цих приладів, наприклад, тригера, який зафіксує новий стан від хибного сигналу.

На рис. 1.18 наведені часові діаграми роботи логічної схеми 2І без затримки сигналів (рис. 1.18,а) та із затримкою (рис. 1.18,б).

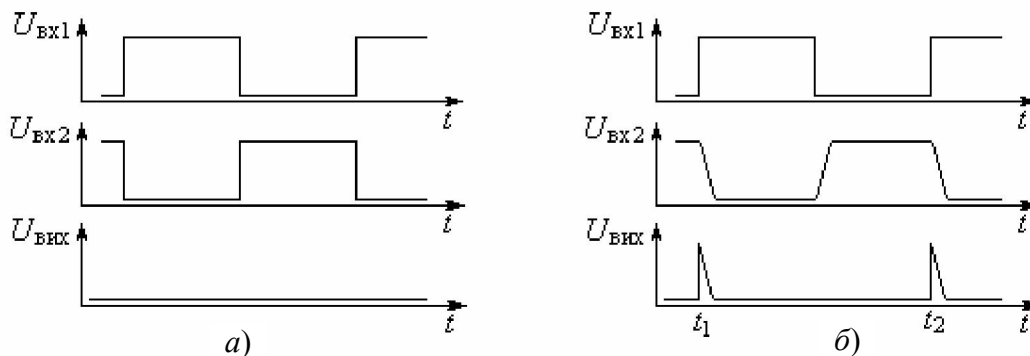


Рисунок 1.18 – Часова діаграма роботи логічної схеми 2І:
а) без часової затримки; б) із часовою затримкою

З діаграми видно, що коли сигнали позбавлені часової затримки (рис. 1.18,*а*), то добуток цих сигналів $U_{\text{вих}} = U_{\text{вх1}} * U_{\text{вх2}} = 0$, тобто вихідний рівень завжди нульовий.

Коли ж один із сигналів $U_{\text{вх2}}$ зазнав часової затримки (рис. 1.18,*б*), то виникають так звані *логічні змагання* або “гонки”.

Через це на виході замість нульового рівня в моменти t_1 та t_2 з’являються *короткочасні хибні сигнали*, які називаються “голками”.

“Голки” вважаються найбільш небезпечними для працездатності асинхронних цифрових пристроїв.

Позбутися цих завад можна різноманітними способами, наприклад, у коло одного з сигналів увести штучну лінію затримки, яка виконана з кола потрібного числа буферів-повторювачів. “Голки” на виході комбінаційного пристрою можна зняти за допомогою включення конденсаторів малої ємності, але при цьому змінюються вихідні параметри схеми, що треба врахувати при подальшій роботі, особливо тоді, коли комбінаційний прилад навантажується послідовнісним приладом.

Контрольні питання

- 1.1. Що таке шифратор?
- 1.2. Що таке дешифратор?
- 1.3. Які шифратори (дешифратори) називаються повними, а які – неповними?
- 1.4. Що таке мультиплексор?
- 1.5. Що таке демультимплексор?
- 1.6. Які функції можуть виконувати компаратори?
- 1.7. Що таке програмовані логічні матриці ПЛМ?
- 1.8. Що таке дворівнева ПЛМ? Які функції виконує кожний її рівень?
- 1.9. Що таке перепрограмована логічна матриця ППЛМ?

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко В.Д. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. – С. С. 276 – 296.
2. Титце У. Полупроводниковая схемотехника: Справочное руководство / У. Титце, К. Шенк. – М.: Мир, 1982. – С. 318 – 331.
3. Хоровиц П. Искусство схемотехніки / П. Хоровиц, У. Хилл. – М.: Мир, 1983. – Т. 1. – С. 528 – 539.
4. Рицар Б.Є. Цифрова техніка / Рицар Б.Є. – К.: НМК ВО, 1990. – 371 с.
5. Баранов С.И. Цифровые устройства на программируемых БИС с матричной структурой / С.И. Баранов, В.А. Склярков. – М.: Радио и связь, 1986. – 269 с.

2. ЦИФРО-АНАЛОГОВІ ТА АНАЛОГО-ЦИФРОВІ ПЕРЕТВОРЮВАЧІ

2.1. Загальні відомості

У більшості випадків сигнал, який надходить від джерела інформації, має аналогову форму та описується неперервною функцією із миттєвими значеннями, які розміщені в деякому інтервалі часу. Це, наприклад, речовий сигнал у телефонії та радіомовленні, телевізійний сигнал тощо.

Передача та обробка таких сигналів може виконуватись двома способами: *аналоговим* та *цифровим*. Останнім часом найбільшого розповсюдження набули *цифрові системи передачі* (ЦСП).

На передавальному кінці ЦСП аналогові сигнали перетворюються в цифрові, а на приймальному кінці виконується обернене перетворення сигналів з цифрової форми в аналогову.

Основні переваги цифрових методів обробки та передачі порівняно з аналоговими полягають у високій завадостійкості, принциповій можливості виключення апаратурної похибки при обчисленнях, стабільності параметрів (незалежно від часу та змін температури), оперативності зміни алгоритму обробки, підвищенні швидкодії тощо.

Перетворення сигналів з аналогової форми в цифрову виконується в приладах, які називаються *аналого-цифровими перетворювачами* (АЦП).

Прилади, які виконують обернене перетворення сигналів з цифрової форми в аналогову, називаються *цифро-аналоговими перетворювачами* (ЦАП).

АЦП і ЦАП є основними функціональними вузлами цифрової техніки та техніки зв'язку, які визначають точність, швидкодію та конструктивні параметри цифрових радіотехнічних систем. За останні роки досягнуті значні успіхи в побудові систем обробки неперервної інформації завдяки появі аналогових мікропроцесорів, де поєднуються перетворення форми зображення та цифрова обробка сигналів.

Параметри інтегральних схем ЦАП і АЦП можна поділити на дві групи: *статичні* та *динамічні*.

Статичні параметри, у свою чергу, розподіляються на дві підгрупи.

До першої підгрупи статичних параметрів відносяться параметри, загально прийняті для усіх типів ІС, які визначають енергетичні показники, наприклад, струм споживання, напругу живлення і т.д.

До другої підгрупи відносяться параметри, які характерні лише для перетворювачів. Це характеристика перетворення (ХП), розрядність, діапазон та рівні вхідних і вихідних сигналів, нелінійність, диференціальна нелінійність, абсолютна похибка перетворення, напруга відхилення нуля на виході ЦАП або АЦП.

До *динамічних параметрів* відносяться час перетворення і частота перетворення, які визначають швидкодію ЦАП і АЦП.

2.2. Цифро-аналогові перетворювачі

Цифро-аналоговими перетворювачами (ЦАП) називають пристрої, які перетворюють вхідну цифрову комбінацію на вихідну аналогову величину. Наприклад, вхідна комбінація Σx_i (рис. 2.1) перетворюється на вихідну напругу $U_{\text{вих}}$.

До входу E_0 подається стабілізована так звана опорна напруга $U_{\text{оп}}$, з якої утворюється аналогова копія $U_{\text{вих}}$ вхідного числа Σx_i .

Цифро-аналогове (ЦА) перетворення складається із підсумовування еталонних величин, що відповідають розрядам вхідного числа.

Застосовуються, як правило, два методи ЦА перетворення: підсумовування однакових еталонних величин та підсумовування еталонних величин, вагові коефіцієнти яких відрізняються.

У першому випадку при формуванні вихідного аналогового сигналу використовується тільки одна еталонна величина вагою в один квант.

У другому методі використовуються еталонні величини з вагами, залежними від номера розряду, а в підсумовуванні беруть участь лише ті еталонні величини, для яких у відповідних розрядах вхідного числа встановлюється рівень логічної "1". При цьому в більшості випадків використовуються двійковий або двійково-десятковий коди.

У випадку двійкового коду робота ЦАП описується рівнянням:

$$U_{\text{вих}} = U_{\text{оп}} (x_1 2^{-1} + x_2 2^{-2} + \dots + x_n 2^{-n}), \quad (2.1)$$

де $U_{\text{вих}}$ – вихідна аналогова напруга;

$U_{\text{оп}}$ – опорна напруга;

x_i – вагові коефіцієнти відповідних двійкових розрядів, які набувають значення або одиниці, або нуля;

n – число розрядів.

Опорною може бути напруга постійного або змінного струму. З неї формуються окремі еталонні величини, що відповідають значенням розрядів вхідного числа, які підсумовуються й утворюють дискретні значення вихідної напруги (аналогової величини).

Класифікація ЦАП може бути подана за різноманітними ознаками:

– за способом формування вихідного сигналу (підсумовування напруг, ділення напруг, підсумовування струмів);

– за видом вихідного сигналу (зі струмовим виходом, з виходом за напругою);

– за полярністю вихідного сигналу (уніполярні, біполярні);

– за характером джерела опорної напруги (постійної чи змінної);

– за типом елементів (резистивні, ємнісні, оптоелектронні тощо).

Схеми ЦАП виготовляють здебільшого в інтегральному виконанні. Серед різних способів ЦА перетворення значного використання набули ЦАП,

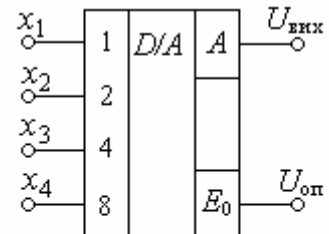


Рисунок 2.1 – Умовне позначення двійкового чотирирозрядного ЦАП

спільною ознакою яких є наявність матриці резисторів R з вихідним аналоговим суматором на перетворювачі струму в напругу.

Матриця резисторів призначена для “зважування” цифрового сигналу, який подано паралельним кодом. Для реалізації ЦА перетворення переважно використовують два типи матриць резисторів:

- складеної з двійково-зважених резисторів;
- на основі резисторів зі співвідношенням $R-2R$.

2.2.1. Параметри ЦАП

Параметри, що нормуються для даного типу ЦАП, визначаються їхнім призначенням або областю їхнього використання. Основні параметри інтегральних мікросхем ЦАП регламентуються стандартом. Номенклатура нормованих параметрів розбивається на дві групи: *статичні* та *динамічні*.

Статичні параметри описують роботу перетворювачів у статичному режимі і містять наступне.

Число розрядів n визначається як двійковий логарифм максимального числа кодових комбінацій на вході ЦАП

$$n = \log_2 b, \quad (2.2)$$

де b – число можливих кодових комбінацій на вході ЦАП.

Сукупність значень вихідної аналогової величини x_i (найчастіше це

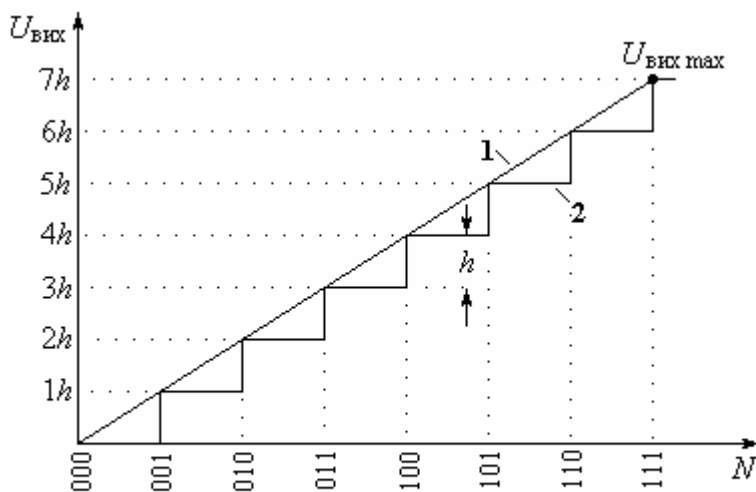


Рисунок 2.2 – Характеристика перетворення трирозрядного двійкового ЦАП

напруга $U_{\text{вих}}$ і тому надалі у формулах будемо використовувати тільки цей параметр), в залежності від вхідного числа N_i називають *номінальною функцією* або *характеристикою перетворення* (ХП). Така сукупність може надаватися у вигляді графіка (рис. 2.2), формули, таблиці.

Значення дискретної зміни вихідної аналогової величини при зміні вхідного числа на одиницю називається *кроком квантування h* .

Кількісний зв’язок між вхідним числом та його аналоговим еквівалентом установлює формула

$$U_{\text{вих } i} = hN \pm \delta U, \quad (2.3)$$

де h – крок квантування;

δU – похибка перетворення;

N – вхідне число.

Двійкове число N складається з певної кількості розрядів $2^i x_i$, які можуть набувати значень або 0, або 1.

$$N = 2^0 x_1 + 2^1 x_2 + \dots + 2^{n-1} x_n = \sum_{i=1}^n 2^i x_i, \quad (2.4)$$

$$x_i = \epsilon \{x_1, x_2, \dots, x_n\},$$

де n – кількість двійкових розрядів.

З характеристики перетворення трирозрядного ЦАП (рис. 2.2), яка побудована за таблицею станів ЦАП (табл. 2.1), видно, що при зміні вхідного числа N на одиницю вихідна аналогова напруга $U_{\text{вих}}$ дискретно змінюється на крок квантування h .

Якщо на вхід ЦАП подано максимальне для даного пристрою двійкове число, то на виході отримуємо максимальне значення вихідної аналогової величини $U_{\text{вих max}}$.

Щодо мінімальної вихідної напруги, то вона може набувати будь-яких значень у межах $0 \leq U_{\text{вих min}} < U_{\text{вих max}}$.

Інтервал значень вихідної напруги від $U_{\text{вих min}}$ до $U_{\text{вих max}}$ називають *діапазоном змінення вихідної величини*.

Крок квантування та діапазон змінення вихідної величини (без урахування похибки перетворення) зв'язані співвідношенням

$$h = \frac{U_{\text{вих max}}}{2^n}. \quad (2.5)$$

При зменшенні кроку квантування (рис. 2.2) характеристика перетворення наближається до прямої лінії 1.

У випадку *ідеального* лінійного ЦАП усі кроки квантування однакові. В *реальних* ЦАП кроки квантування в різних точках відрізняються одне від одного, що зумовлює похибку перетворення. Тому використовують середнє значення кроку квантування $h_{\text{сер}}$.

Найменше значення зміни вихідної аналогової величини від кроку квантування h , яку можна відрізнити, називається *роздільною здатністю*. Роздільна здатність може розраховуватися або в одиницях вихідної аналогової величини, або у відсотках.

Реальні характеристики перетворення відрізняються від ідеальної неоднаковим значенням кроків квантування на всій довжині ХП. Різниця між ідеальною ХП та реальною визначає *похибку перетворення*. Похибка, яка виникає в реальному випадку, характеризується рядом відхилень та визначається відповідними параметрами: нелінійність, диференційна нелінійність, відхилення аналогової величини від номінального значення, напруга зсуву нуля.

Нелінійність δ_L – це максимальне відхилення точки реальної ХП від ідеальної. Нелінійність може бути приведеною до виходу і тоді вона визначається у відносних одиницях формулою

$$\delta_L = \frac{\delta_{\text{max}}}{U_{\text{вих max}}}, \quad (2.6)$$

де δ_{max} – максимальне відхилення будь-якої точки реальної ХП від ідеальної.

Диференційна нелінійність δ_{LD} – це відхилення дійсного кроку квантування h від його середнього значення $h_{\text{сер}}$, яке визначається формулою

Таблиця 2.1 –
Стани ЦАП

x_3	x_2	x_1	$U_{\text{вих}}$
0	0	0	0
0	0	1	$1h$
0	1	0	$2h$
0	1	1	$3h$
1	0	0	$4h$
1	0	1	$5h$
1	1	0	$6h$
1	1	1	$7h$

$$\delta_{LD} = \frac{h - h_{\text{сеп}}}{U_{\text{max}}}. \quad (2.7)$$

Враховують лише максимальне значення диференційної нелінійності за абсолютною величиною.

Абсолютна похибка перетворення в кінцевій точці шкали δ_{FS} відраховується як відхилення реального значення кінцевої точки характеристики перетворення від ідеального за формулою

$$\delta_{FS} = U_{\text{вих max } i} - U_{\text{вих max}}, \quad (2.8)$$

де $U_{\text{вих max } i}$ і $U_{\text{вих max}}$ – значення вихідної напруги в кінцевій точці, відповідно реальної та ідеальної ХП.

Напруга зсуву нуля вихідної аналогової величини – це дійсне значення вихідної величини при нульовому вхідному числі. Цей параметр визначається як величина, на яку паралельно ідеальній зсувається реальна ХП відносно початку координат.

Динамічні параметри ЦАП зумовлені тим, що в реальних умовах експлуатації ЦАП працюють при безперервному змінненні значень вхідного числа. Відлік напруги на виході ЦАП повинен відбуватися тільки після закінчення всіх перехідних процесів у ньому. Тому *швидкодія* ЦАП визначається часом його перехідних процесів, а динамічні параметри обумовлюють швидкодію та похибку оброблення інформації.

Найважливішим динамічним параметром є *час установлення вихідного сигналу* або *час перетворення* t_s . Він визначається як час від початку зміни цифрового сигналу на вході ЦАП до появи на виході відносно сталого аналогового сигналу. В залежності від типу ЦАП це може бути час установлення вихідного струму t_{si} , або час установлення вихідної напруги t_{su} . Найчастіше в довідниках називають час установлення при стрибкоподібній зміні вхідного числа від мінімуму до максимуму, або навпаки.

Другий важливий динамічний параметр – це *час затримки розповсюдження* t_r . Він визначається тривалістю часу до моменту досягнення вихідною аналоговою величиною половини максимального вихідного значення $U_{\text{вих max}}$.

2.2.2. Схеми ЦАП

Схеми ЦАП у мікросхемному виконанні є резистивними матрицями, які іноді називають *кантувачами*.

Найпростішими кантувачами вважаються матриці на *двійково-зважених резисторах*, бо містять найменшу кількість резисторів – лише по *одному* резистору на кожний двійковий розряд.

Номінали резисторів, які використовуються для ЦАП, визначаються з ряду $2^0R; 2^1R; 2^2R; \dots; 2^{n-1}R$, де n – число двійкових розрядів.

З цього ряду видно недолік матриці на двійково-зважених резисторах – значна кількість номіналів резисторів: n . Якщо врахувати прецизійне виготовлення кожного резистора, то згаданий недолік є суттєвим. Тому в більшості випадків матриці на двійково-зважених резисторах виготовляють

тетрадними, тобто чотирирозрядними. Тоді в кожній тетradі лише чотири резистори. Щодо збільшення розрядності, то її нарощують кількістю тетрад.

Схема ЦАП на матриці двійково-зважених резисторів з підсумовуванням зважених струмів показана на (рис. 2.3),

Тут:

$R, 2R, 4R, 8R$ – матриця двійково-зважених резисторів;

E_0 – джерело стабілізованої опорної напруги;

$A1$ – комутатор;

$A2$ – суматор зважених струмів (перетворювач струму в напругу).

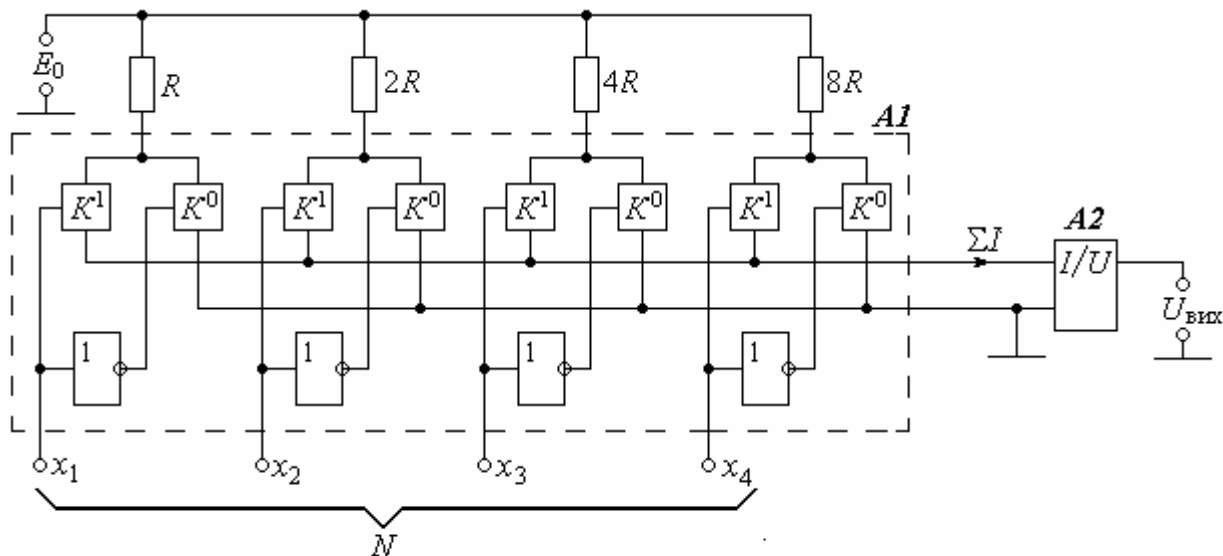


Рисунок 6.3 – Схема ЦАП на двійково-зважених резисторах

Входи резисторів матриці $R, 2R, 4R, 8R$ живляться від джерела опорної напруги E_0 . Виходи цих резисторів підключені до одиничних K^1 та нульових K^0 ключів комутатора $A1$. До входів ключів K^1 підключені двійкові розряди $x_1; x_2; x_3; x_4$ вхідного числа N безпосередньо, а до входів ключів K^0 – через інвертори. Тому за будь-якого значення розряду один із ключів відкритий, а інший закритий.

Щодо наявності в кожному розряді двох ключів, а не одного, то це пояснюється наступним.

Для уникнення впливу внутрішнього опору джерела E_0 на точність перетворювання слід мати навантаження E_0 незмінним за будь-якого стану ключів. З цієї вимоги слід, що за будь-якого стану входів виходи резисторів $R, 2R, 4R, 8R$ мають підключатися до однієї і тієї самої напруги. Такою напругою тут вибрана нульова. Тоді при нульовому стані входів $x_1; x_2; x_3; x_4$ ключі K^0 підключають виходи резисторів до нульової шини, а при одиничному – до інвертуючого входу приймача струму $A2$, який є віртуальною землею, тобто теж до нульової напруги. Отже, для незмінного навантаження E_0 слід мати парні перекидні ключі K^0 і K^1 .

Через одиничні ключі K^1 усі резистори матриці зв'язані зі входом суматора струмів $A2$, а через K^0 – з нульовою шиною. Тому виходи резисторів матриці підключаються або до входу суматора струмів $A2$, або до нуля.

Перетворення вхідного числа N в напругу $U_{\text{вих}}$ здійснюється наступним чином.

В залежності від вхідного числа N на входах ключів K^1 установлюється відповідна комбінація логічних “0” та логічних “1”. Ключі K^1 , на входах яких напруга дорівнює логічному “0”, закриваються і струм через них не тече, тобто вони не підключають резистори до входу суматора струмів $A2$.

Так, при нульовому вхідному числі $N = \{0000\}$ всі ключі K^1 закриті, а K^0 відкриті. Тому виходи резисторів відключені від входу суматора ΣI і підключені до нуля. Через це сума струмів на вході суматора теж нульова $\Sigma I = 0$. Тому вихідна напруга дорівнює нулю: $U_{\text{вих}} = Q \Sigma I = 0$, де Q – коефіцієнт перетворення струму в напругу.

Коли ж усі розряди підведеного числа одиничні $N = \{1111\}$, то стани ключів змінюються на протилежні: ключі K^0 закриваються, відключаючи резистори від нуля, а ключі K^1 відкриваються, підключаючи резистори до входу суматора $A2$. Через це сума струмів на вході суматора максимальна $\Sigma I = I_{\text{max}}$, через що вихідна напруга теж максимальна: $U_{\text{вих max}} = Q I_{\text{max}}$.

Проміжні значення вхідного числа $\{0000\} < N < \{1111\}$ викликають відповідні проміжні пропорційні значення вихідної напруги $0 < U_{\text{вих}} < U_{\text{вих max}}$.

Так, розряд $x_4 = 1$ забезпечить на вході $A2$ градацію струму $I_1 = E_0/8R$, розряд $x_3 = 1$ забезпечить дві таких градації і т.д. Сума струмів на вході суматора становитиме

$$\Sigma I = \frac{E_0}{R} x_1 + \frac{E_0}{2R} x_2 + \frac{E_0}{4R} x_3 + \frac{E_0}{8R} x_4 = 15 \frac{E_0}{8R}. \quad (2.9)$$

Вихідна напруга визначиться як

$$U_{\text{вих}} = Q \Sigma I. \quad (2.10)$$

Приймаючи в (2.8) усі значення $x_i = 1$, знаходимо максимальну вихідну напругу

$$U_{\text{вих max}} = 15 Q \frac{E_0}{8R}. \quad (2.11)$$

Крок квантування дорівнює градації вихідної напруги від наймолодшого розряду ($x_4 = 1$):

$$h = Q \frac{E_0}{8R}. \quad (2.12)$$

Діленням (2.11) на (2.12) знаходимо, що кількість кроків квантування чотирирозрядного двійкового ЦАП становить 15.

В залежності від вхідного числа N проміжні значення вихідної напруги набувають значення в межах $0 < U_{\text{вих}} \leq 15h$.

Наперед слід сказати, що для сучасної техніки (про це йдеться нижче) 15 кроків квантування – це дуже мало. Як правило, потрібно $10^2 \dots 10^4$ кроків. Тому виникає задача нарощування розрядності ЦАП.



Рисунок 2.4 – Схема нарощування розрядності двійково-десятькового ЦАП

Цю задачу вирішує каскадне з'єднання тетрад ЦАП (рис. 2.4) через масштабуючі резистори з опорами

$$R_m = 8,1R_i, \quad (2.13)$$

де R_i – внутрішній опір тетради. Його провідність визначається сумою провідностей всіх резисторів

$$\frac{1}{R_i} = \frac{1}{8R} + \frac{1}{4R} + \frac{1}{2R} + \frac{1}{R} = \frac{15}{8R}, \quad (2.14)$$

звідки внутрішній опір тетради становить

$$R_i = \frac{8}{15}R. \quad (2.15)$$

За формулами (6.13) та (6.15) можна розрахувати опір масштабувальних резисторів R_m .

Схема працює наступним чином.

Старша тетрада “× 1000” живиться *повною* напругою $U_{оп}$, а кожна наступна тетрада – напругою, зменшеною в 10 разів через масштабуючі резистори $R_m = 8,1R_i$. Так, надана схема шістьма номіналами резисторів забезпечує 9999 десяткових градацій вихідної напруги, тобто 9999 кроків квантування.

Кількість каскадів для подальшого можливого зменшення кроку квантування не обмежується.

Недоліком ЦАП на матриці двійково-зважених резисторів є досить значна кількість номіналів: шість на тетраду.

Цей недолік усунений в ЦАП на матриці типу $R-2R$.

Перевага матриці типу $R-2R$ полягає в простоті її виготовлення, бо вона має лише два номінали резисторів R та $2R$ проти багатьох номіналів матриці на двійково-зважених резисторах.

Схема ЦАП на матриці $R2R$ показана на рис. 2.5.

Вона відрізняється від матриці на двійково-зважених резисторах (рис. 2.3) лише тим, що замість них використані резистори з опорами $2R$, а між ними включені резистори з опорами R . Тут x_1 – самий старший розряд, а x_n – наймолодший.

Підрахуємо кількість кроків квантування, яку забезпечує n -розрядний двійковий ЦАП.

Для цього визначимо опір між кожними з точок B, C, D, \dots, F і нульовою шиною (корпусом) та потенціали цих точок (рис. 2.5).

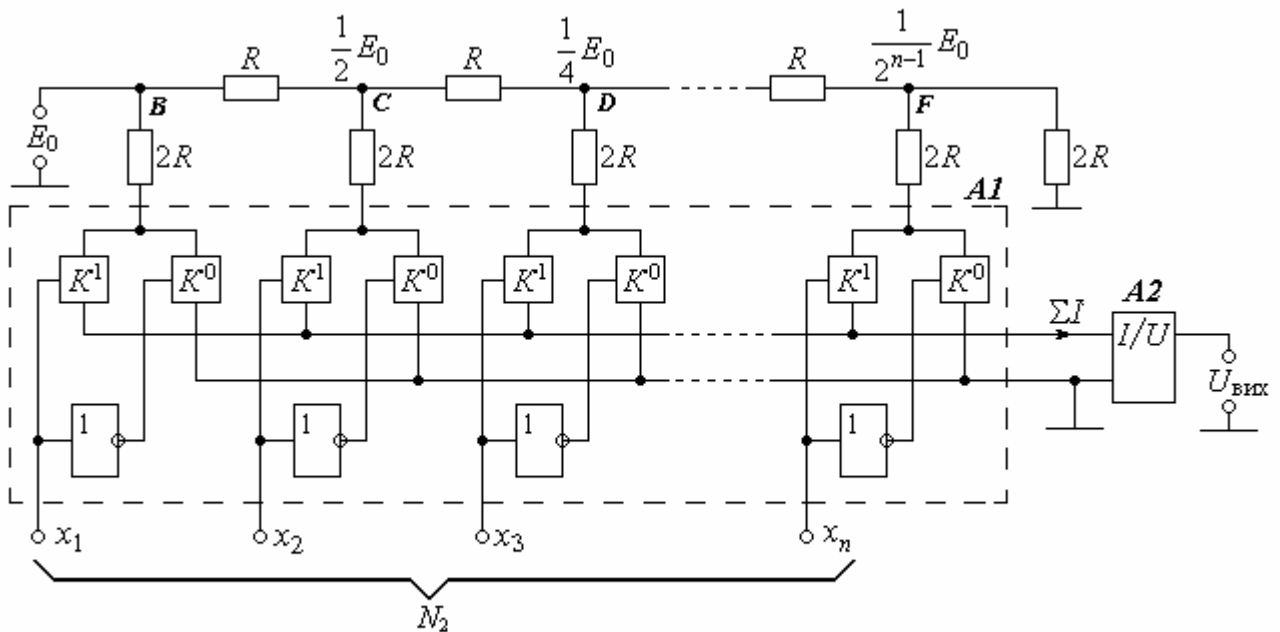


Рисунок 2.5 – Схема ЦАП на матриці типу $R-2R$

Між точкою F і корпусом включені два паралельно з'єднані резистори з опором $2R$. Результуючий опір цього з'єднання становить $R_F = R$. (Щодо опорів ключів K^0 і K^1 , то вони нехтовно малі, через що їхнього впливу на опір R_F практично немає).

Тоді точка F є виходом подільника напруги з опором R в кожному плечі і тому потенціал точки F буде результатом ділення на два вхідної напруги цього подільника.

Отже, потенціал точки D буде результатом ділення на два потенціалу точки C . Таким же чином переконаємося в тому, що між точкою D і корпусом увімкнений еквівалентний опір $R_D = R$. Тоді потенціал точки D також буде результатом ділення на два, але потенціалу точки C .

Таким чином, від розряду до розряду потенціал ділиться на два. Тому потенціали точок B, C, D, \dots , відповідно становлять $E_0; (1/2)E_0; (1/4)E_0; \dots$. З цього слідує, що розряд $x_1 = 1$ забезпечить на вході перетворювача $A2$ протікання струму $I_1 = \frac{E_0}{2R}$, розряд $x_2 = 1$ забезпечить $I_2 = \frac{1}{2} \frac{E_0}{2R}$ і т.д. Сумарний струм на вході перетворювача струму в напругу $A2$ при вхідному двійковому числі N становитиме

$$\Sigma I = \frac{E_0}{2R} x_1 + \frac{1}{2} \frac{E_0}{2R} x_2 + \frac{1}{4} \frac{E_0}{2R} x_3 + \dots + \frac{1}{2^{n-1}} \frac{E_0}{2R} x_n = \frac{E_0}{2R} \sum_{i=1}^n \frac{1}{2^{i-1}} x_i, \quad (2.16)$$

де x_i – коефіцієнт розряду, який набуває значення 0 або 1 в залежності від двійкового вхідного числа N .

Вихідна напруга визначиться як $U_{\text{вих}} = Q \Sigma I$, тобто

$$U_{\text{вих}} = E_0 \frac{Q}{2R} x_1 + \frac{1}{2} E_0 \frac{Q}{2R} x_2 + \frac{1}{4} E_0 \frac{Q}{2R} x_3 + \dots + \frac{1}{2^{n-1}} E_0 \frac{Q}{2R} x_n =$$

$$= E_0 \frac{Q}{2R} \sum_{i=1}^n \frac{1}{2^{i-1}} x_i, \quad (2.17)$$

де Q – коефіцієнт перетворення струму в напругу.

Приймаючи в (2.17) $x_i = 1$ та $i = n$, знаходимо максимальну вихідну напругу

$$U_{\text{вих max}} = E_0 \frac{Q}{2R} \frac{1}{2^{n-1}} (n^2 - 1). \quad (2.18)$$

Крок квантування знайдемо як різницю між будь-якими двома суміжними членами (2.18):

$$h = \frac{1}{2^{n-1}} E_0 \frac{Q}{2R}. \quad (2.19)$$

Діленням (2.18) на (2.19) знаходимо кількість кроків квантування в залежності від числа n двійкових розрядів

$$q = (n^2 - 1). \quad (2.20)$$

З формул (2.10) ... (2.12) та (2.18) ... (2.20) можна зробити наступні висновки:

- вихідна напруга ЦАП прямо пропорційна напрузі живлення E_0 і вхідному двійковому числу $N = \{x_1, x_2, x_3, \dots, x_n\}$;
- крок квантування не залежить від максимальної вихідної напруги і за інших рівних умов визначається тільки кількістю розрядів вхідного числа;
- кількість кроків квантування за будь-яких умов визначається тільки кількістю розрядів вхідного числа;
- чим більше вхідне число, тим менше крок квантування.

Щодо серійних мікросхем ЦАП, то вони містять усі елементи, які позначені на рис. 2.5, крім перетворювача струму в напругу $A2$ (рис. 2.6,а). Тому його підключають до ЦАП зовнішньо (рис. 2.6,б). Як перетворювач струму в напругу здебільшого використовують операційний підсилювач $A2$.

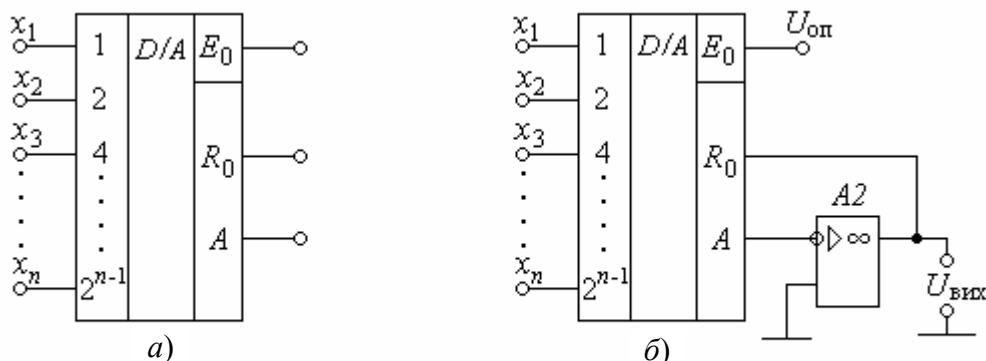


Рисунок 2.6 – Двійковий ЦАП: а) умовне позначення; б) схема включення

2.3. Аналого-цифрові перетворювачі

Аналого-цифровими перетворювачами (АЦП) називаються пристрої, які перетворюють аналогову величину в пропорційне число, тобто створюють цифрову копію аналогової величини і таким чином перетворюють аналог у код.

Аналого-цифрове перетворення містить наступні операції: дискретизацію, квантування та кодування.

Дискретизація складається з вибору із неперервного за часом аналогового сигналу U_c окремих миттєвих його значень (рис. 2.7), які надходять через визначений часовий проміжок $\Delta t = (t_{i+1} - t_i)$.

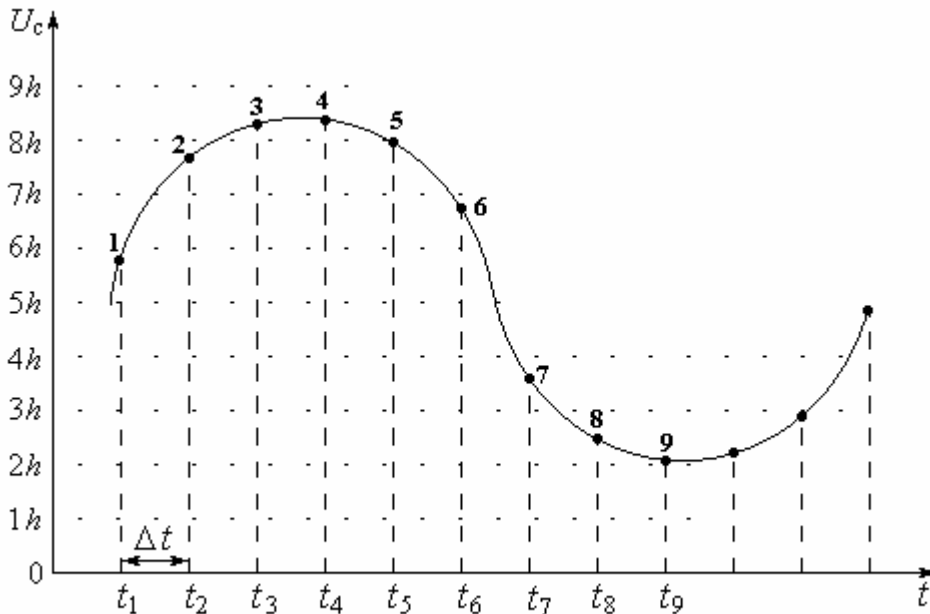


Рисунок 2.7 – Дискретизація та квантування неперервного сигналу

Моменти $t_1; t_2; \dots; t_{i-1}; t_i$, в яких визначаються миттєві значення, називаються *тактовими моментами часу*, а різниця між суміжними моментами $\Delta t = (t_{i+1} - t_i)$ – *тактовим інтервалом часу*. Дискретні значення сигналу слід відраховувати з таким малим інтервалом Δt , щоб можна було б установити сигнал в аналоговій формі з можливо малою похибкою.

Згідно з теоремою Котельникова, якщо сигнал має обмежений спектр, тобто всі його спектральні складові мають частоти нижче, ніж деяка частота F_{\max} , то для установлення аналогового сигналу з послідовності його дискретних значень тактовий інтервал має задовольняти умові

$$\Delta t \leq \frac{1}{2F_{\max}}. \quad (2.21)$$

Наприклад, при дискретизації речових сигналів, що мають спектр частот 300...3400 Гц, вибирається тактова частота $F \geq 2F_{\max} = 6800$ Гц. При цьому тактовий інтервал становить $\Delta t \leq 1/F = 147$ мкс.

Операція квантування складається з утворення сітки так званих рівнів квантування (рис. 2.7), які зсунені один відносно одного на крок квантування h . При цьому кожний рівень має свій порядковий номер (0; 1; 2; ...).

Кодування здійснюється тим, що отримані в процесі дискретизації значення аналогового сигналу U_c замінюють ближчими до них номерами рівнів квантування.

Так, значення напруги в момент t_1 замінюється номером 6 ближчого до нього рівня квантування $6h$, в тактові моменти t_2 ; t_3 ; t_4 та t_5 значення напруги U_c замінюються номером 8 ближчого до них рівня квантування $8h$, моменту t_8 відповідає або номер 3, або номер 2 і т.д.

Отже, квантування – це визначення для кожної точки дискретизації сигналу ближчого номера рівня з існуючих.

Процес квантування вносить похибку $\delta_{\text{кв}}$, яка називається *шумом квантування* і визначається в межах $-h/2 \leq \delta_{\text{кв}} \leq +h/2$. Зменшення шуму квантування досягається шляхом зменшення кроку квантування h , але це призводить до збільшення числа рівнів, через що ускладнюється АЦП. Щоб запобігти невиправданого нарощування апаратних засобів, крок квантування вибирають таким, що дорівнює припустимій абсолютній похибці δ :

$$h = \delta. \quad (2.22)$$

Тоді при завжди відомій максимальній вхідній напрузі сигналу $U_{\text{сmax}}$ кількість кроків квантування становить

$$K = \frac{U_{\text{сmax}}}{h}. \quad (2.23)$$

Чим більше число рівнів квантування, тим точніше відтворюється цифрова копія аналогового сигналу U_c , але це збільшення обмежується ускладненням пристрою, бо кількість кроків квантування визначає число двійкових розрядів відповідних вузлів апаратури.

Тому число рівнів квантування для кожного конкретного випадку вибирається за умов отримання заданих параметрів перетворення.

Так, наприклад, для телефонного зв'язку припустима похибка перетворення становить (0,5 ... 1,0) % від максимальної аналогової величини, тобто аналоговий сигнал має відраховуватися з дискретністю не менше (0,5 ... 1,0) % . Таким самим має бути і крок квантування. Тоді з формули (2.23) маємо, що кількість кроків квантування повинна бути не менше 100. Найближче значення двійкового розряду дорівнює 128. Тому для телефонного зв'язку використовується 128 ... 256 кроків квантування.

Таким чином, в процесі квантування послідовність визначених при дискретизації значень аналогового сигналу замінюється послідовністю відповідних чисел (номерів рівнів квантування).

Нарешті, в процесі кодування числа послідовності номерів рівнів квантування зображуються у визначеній системі числення, наприклад, у двійковій.

Таким чином, при аналого-цифровому (АЦ) перетворенні, який складається з процесів квантування за рівнем дискретизованого в часі аналогового сигналу U_c та подальшого його кодування, на виході формується двійкове n -розрядне число $N = \{x_1, x_2, x_3, \dots, x_n\}$.

Класифікація АЦП, якою тепер користуються, показує, як у часі розгортається процес перетворення. За такою класифікацією всі АЦП можна поділити на три типи: *послідовні, паралельні та паралельно-послідовні*.

До *послідовних* АЦП відносяться такі, де перетворення відбувається послідовно за часом, тобто крок за кроком. Такими АЦП є, наприклад, перетворювачі розгортаючої дії, слідкуючі АЦП, перетворювачі напруги в частоту, АЦП інтегруючого типу тощо. Усі ці АЦП дозволяють отримувати високу розрядність, але мають низьку швидкодію.

Паралельні АЦП побудовані за принципом одноразового перетворення сигналу шляхом його кодування за допомогою набору компараторів. Такі АЦП виявляються більш швидкодіючими за інші і досягають частот перетворення 100 ... 400 МГц. До недоліків паралельних АЦП відноситься збільшення числа компонентів схеми при зростанні розрядності вихідного числа.

Паралельно-послідовні АЦП засновані на двоступінчатому алгоритмі перетворення: спочатку відбувається визначення старших розрядів вихідного числа за допомогою першого малорозрядного паралельного АЦП, а далі виділяється залишок аналогового сигналу, який визначається молодшими розрядами за допомогою другого малорозрядного паралельного АЦП. Зв'язок між цими двома паралельними АЦП відбувається за послідовним алгоритмом.

2.3.1. Параметри АЦП

Усі параметри АЦП, так само як і ЦАП, можна поділити на дві групи: *статичні та динамічні*.

Найважливішими *статичними* параметрами АЦП є такі, що характеризують похибку перетворення.

Число розрядів або розрядність – це найближче ціле двійкового логарифму номінального значення вихідного числа N :

$$q = \log_2 N. \quad (2.24)$$

При подачі на вхід АЦП лінійної змінної напруги на виході спостерігається послідовна зміна чисел.

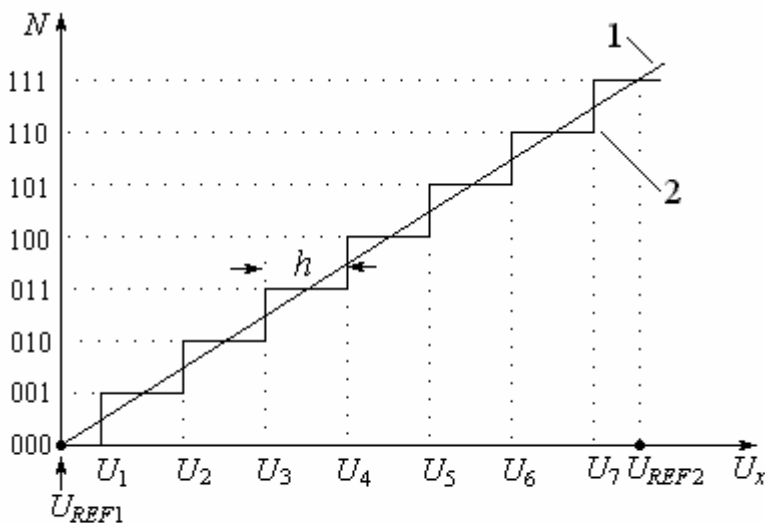


Рисунок 2.8 – Номінальна характеристика перетворення двійкового трирозрядного ЦАП

Залежність між вхідною аналоговою напругою U_x , яка перетворюється, і вихідним числом N називається *характеристикою перетворення* (ХП) (рис. 2.8).

Характеристика перетворення ХП АЦП, як і для ЦАП, може надаватися у вигляді таблиць, графіків або формул. ХП описується рядом параметрів. Точки на осі абсцис відповідають напрузі міжчислових переходів.

Під напругою міжчислового переходу розуміється така вхідна напруга, статистичні ймовірності перетворення якої в заданому і попередньому значеннях вихідного числа рівні.

Для ідеального АЦП напруги міжчислових переходів відповідають опорним напругам, які формуються ЦАП.

На рис. 2.8 показана лінійна ХП (1), яка утворюється при необмеженому зменшенні кроку квантування h . Початкова опорна напруга реальної ХП (2) становить $U_{REF1} = 0$.

Для отримання ХП ідеального АЦП необхідно, щоб напруга першого міжчислового переходу U_1 розміщувалась на відстані $h/2$ від початкової опорної напруги $U_{REF1} = 0$. Аналогічно, напруга останнього міжчислового переходу (U_{i+1}) повинна відрізнятись від максимальної опорної напруги U_{REF2} на половину кроку квантування ХП. Для ідеального АЦП ширина сходинки дорівнює кроку квантування

$$h = \Delta U = U_{i+1} - U_i = \frac{U_{REF2} - U_{REF1}}{2^{n-1}}. \quad (2.25)$$

Реальна ХП АЦП може значно відрізнятись від ідеальної, завдяки неідентичності кроків квантування. Відхилення реальної ХП від ідеальної визначає *похибку* АЦП і характеризується наступними параметрами: нелінійність; напруга зсуву нуля; диференційна нелінійність.

Дійсне значення вхідної напруги в нульовій точці ХП, відносно номінального нульового значення цієї напруги, визначає *напругу зсуву нуля*. Фізично ця напруга показує паралельний зсув ХП уздовж осі абсцис.

Нелінійність АЦП – це відхилення вхідної напруги в будь-якій точці ХП від величини, визначеної за ідеальною ХП у цій самій точці. Цей параметр характеризує відхилення центрів сходинок дійсної ХП від прямої лінії, що апроксимує номінальну ХП.

Диференційна нелінійність АЦП – це відхилення дійсних значень кроків квантування ХП від їхнього середнього значення. Диференційна нелінійність визначає, наскільки відрізняється реальна сходинка між сусідніми числами від ідеальної.

Діапазон перетворень – це різниця між максимальним та мінімальним значеннями вхідної напруги

$$A = U_{\text{вх max}} - U_{\text{вх min}}. \quad (2.26)$$

Для розрахунку кількості кроків квантування використовують поняття *відносної похибки* АЦ перетворення $\delta = 2^{-n}$, де n – число розрядів.

Якщо відносна похибка АЦ перетворення задана, то її приймають за відносний крок квантування і тоді кількість кроків квантування становить

$$K = 1/\delta. \quad (2.27)$$

Наприклад, припустима відносна похибка перетворення становить $\delta = 10^{-3}$. Тоді кількість кроків квантування дорівнює 1000. Це число забезпечують 10 двійкових розрядів АЦП: $2^{10} = 1024$.

До динамічних параметрів належать *швидкодія* та *час перетворення*.

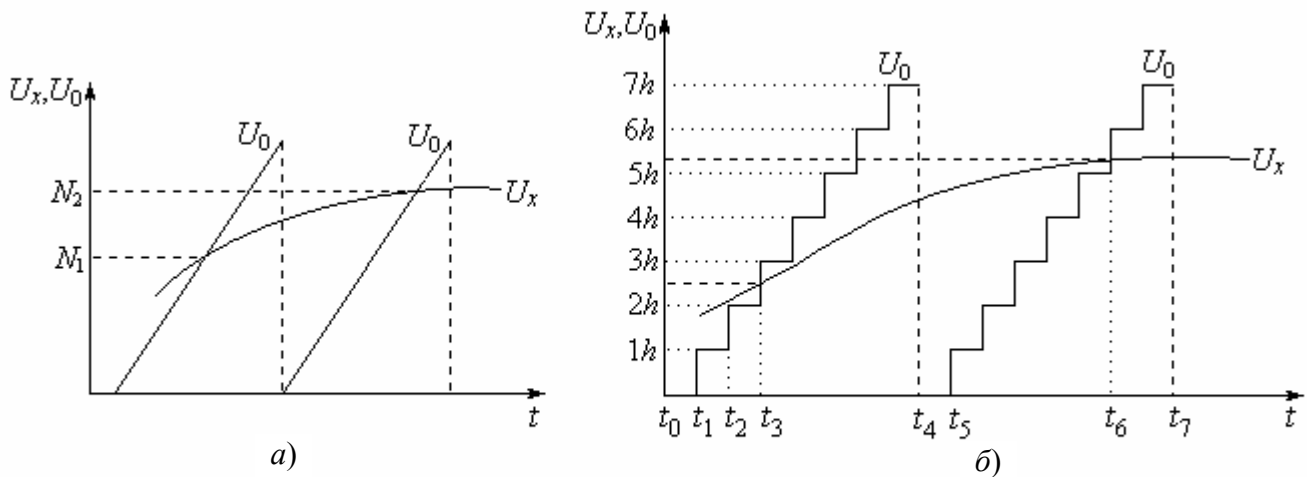


Рисунок 2.9 – Часова діаграма роботи ЦАП розгортального перетворення:
а) принцип; б) реалізація

Швидкодія АЦП характеризується часом перетворення або частотою перетворення.

Час перетворення t_c – це інтервал часу між моментами появи двох сусідніх вихідних чисел.

2.3.2. АЦП розгортального перетворення

Відомо багато способів АЦ перетворення і кожний з них має свої переваги і недоліки для конкретних практичних випадків.

Серед схем АЦП найчастіше зустрічаються АЦП *розгортального перетворення*, *слідкуючі АЦП* та АЦП *паралельного кодування*.

Принцип роботи АЦП *розгортального перетворення* полягає в порівнянні вхідної напруги U_x (рис. 2.9,а) з еталонною U_0 , послідовному нарощуванні еталонної напруги U_0 та формуванні чисел N_1, N_2, \dots, N_i , пропорційних аналоговому сигналу, коли еталонна напруга досягає вхідної: $U_0 = U_x$.

Структурна схема АЦП розгортального перетворення (урівноваження) показана на рис. 2.10.

До складу АЦП розгортального перетворення мають входити наступні пристрої:

- генератор прямокутних імпульсів $D1$;
- лічильник імпульсів $D2$;
- ЦАП $D5$;
- формувач одиничного імпульсу $D3, D4$ для запуску регістра $D6$;
- компаратор $A1$;
- відтворювальний пристрій $A2$, яким може бути цифровий індикатор, інтерфейс передавання тощо.

АЦП працює наступним чином.

З кожним імпульсом генератора $D1$ вміст лічильника $D2$ збільшується на одиницю, через що еталонна напруга U_0 на виході ЦАП $D5$ збільшується на один крок квантування h (див. рис. 2.9,б). Так, у момент t_1 в лічильник $D2$

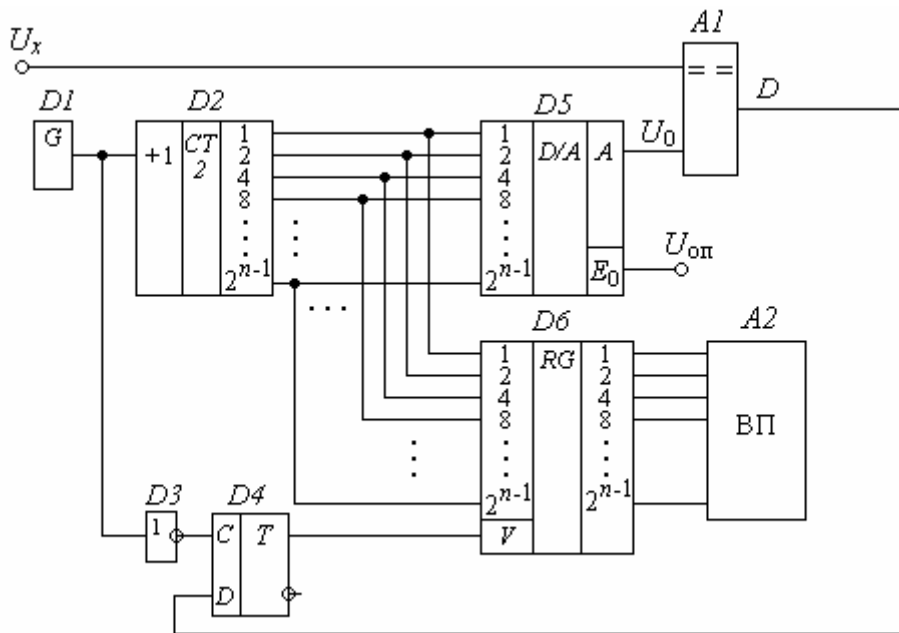


Рисунок 2.10 – АЦП розгортального перетворення

потрапляє один імпульс і тому $U_0 = 1h$. У момент t_2 вміст лічильника дорівнює 2, через що $U_0 = 2h$. У момент t_3 еталонна напруга становить $U_0 = 3h$ і т.д. Таким чином, вмістом лічильника $D2$ є номер рівня еталонної напруги U_0 .

Після кожного імпульсу генератора $D1$ через напівперіод на виході інвертора $D3$ з'являється фронт для запуску тригера $D4$, щоб записати в регістр $D6$ вміст лічильника $D2$. Однак запуск тригера $D4$ здійснюється не завжди, а тільки при $U_0 \geq U_x$. (Слід зауважити на те, що інвертор $D3$ і тригер $D4$ виконують функцію формувача одиничного імпульсу, схема якого не має значення і може бути іншою. Тут наданий один найпростіший з багатьох інших можливих варіантів).

До моменту t_3 сигнал U_x недокомпенсований ($U_0 < U_x$) і тому $D = 0$. Тригер $D4$ залишається в нульовому стані, через що вміст лічильника $D2$ в регістр $D6$ не записується.

У момент t_3 на третьому імпульсі в лічильнику $D2$ сигнал перекомпенсований ($U_0 > U_x$) еталонною напругою $U_0 = 3h$, через що на виході компаратора $D = 1$. Тригер $D4$ підготовлений до встановлення. Через напівперіод після моменту t_3 на вході C тригера з'явиться фронт і тригер $D4$ встановлюється, записуючи вміст лічильника $D2$ в регістр $D6$, тобто 3. Так, у момент t_3 значення вхідної напруги U_x перетворилося на номер рівня $3h$, тобто на число 3.

Незважаючи на велику чи малу вхідну напругу U_x , лічильник $D2$ після перекомпенсації продовжує накопичувати імпульси до повної місткості. При цьому тригер $D4$ залишається в установленому стані. Фронту на його виході немає і тому вміст регістра $D6$ не поновлюється.

У момент t_4 лічильник $D2$ переповнюється, скидається в нульовий стан. Сигнал U_x знову стає недокомпенсованим ($U_0 < U_x$) і тому $D = 0$. Наступний фронт інвертора $D3$ скидає тригер $D4$ у нульовий стан. Його вихідний зріз при цьому не поновлює запис у регістр $D6$, залишаючи в ньому вміст 3.

З моменту t_5 процес заповнення лічильника $D2$ та його результати повторюються. До моменту t_6 сигнал U_x недокомпенсований ($U_0 < U_x$) і тому $D = 0$. Тригер $D4$ залишається в нульовому стані, через що вміст лічильника $D2$ в регістр $D6$ не записується.

У момент t_6 в лічильник $D2$ надійшло 6 імпульсів. При цьому еталонна напруга $U_0 = 6h$ перевищує вхідну U_x , тобто настає перекомпенсація сигналу ($U_0 > U_x$), через що $D = 1$. Тригер $D4$, встановлюючись, записує в регістр $D6$ новий номер рівня еталонної напруги $U_0 = 6h$, тобто 6. Так, у момент t_6 значення вхідної напруги U_x перетворилося на номер рівня $6h$, тобто число 6 і т.д.

Отже, з кожним фактом перекомпенсації сигналу ($U_0 > U_x$) поновлюється вміст регістра $D6$, який є результатом аналого-цифрового перетворення. Ці результати потрапляють на відтворювальний пристрій ВП $A2$, призначення і схема якого для кожної окремої задачі є своїми. ВП може бути дешифратором цифрового індикатора, перетворювачем коду, інтерфейсом системи передачі даних тощо. Тому його схема тут не розкривається.

Перевагою АЦП розгортального перетворення (див. рис. 2.10) є висока роздільна здатність, яка зумовлена відсутністю зворотного зв'язку (зв'язок $A1 - D4$ не є зворотним, бо не впливає на роботу елементів тракту сигналу: $D2, D5$ та ін.). Через це можна без появи *самозбудження* збільшувати коефіцієнт підсилення компаратора $A1$, підвищуючи саме тим роздільну здатність.

Значним недоліком АЦП розгортального перетворення є дуже мала швидкодія, яка зумовлена тим, що незалежно від рівня вхідної напруги U_x (див. рис. 2.9,б), лічильник імпульсів $D2$ (див. рис. 2.10) починає свою роботу з нульового стану (моменти t_0, t_4, t_7 і т. ін.) для кожного перетворення (моменти t_3, t_6, \dots). При цьому втрачається значний час $t_3 \dots t_4$.

Цей недолік усунений в сліdkуючому АЦП.

2.3.3. Сліdkуючий АЦП

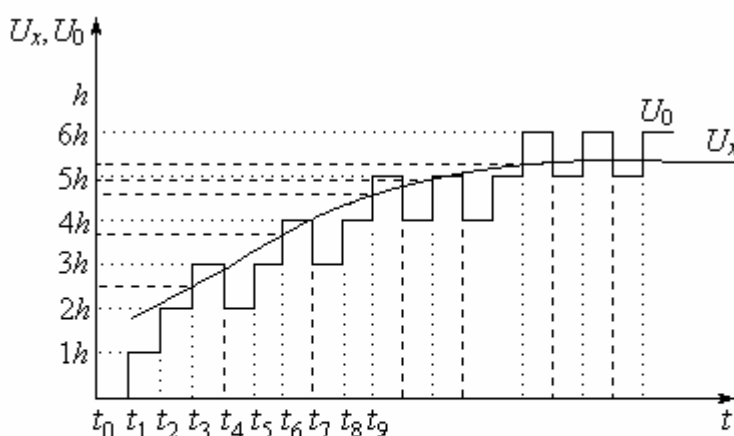


Рисунок 2.11 – Сліdkуюче аналого-цифрове перетворення:

- моменти підсумовування імпульсів;
- моменти віднімання імпульсів

Принцип дії сліdkуючого аналого-цифрового перетворення полягає в тому, що після кожної перекомпенсації еталонна напруга U_0 (рис. 2.11) зменшується не на всю величину, а тільки, щоб викликати недокомпенсацію. Так еталонна напруга U_0 “сліdkує” за вхідною напругою U_x , відхиляючись від неї лише на малу величину. На це відхилення потрібно мало часу, через що підвищується швидкодія.

Схема сліdkуючого АЦП показана на рис. 2.12.

момент t_6 вміст лічильника збільшується до 4. Оскільки при цьому настає перекомпенсація ($U_0 > U_x$), то вміст 4 заноситься в регістр $D8$.

Далі цей процес продовжується. Так у момент t_7 еталонна напруга U_0 зменшиться на h , у моменти t_8 і t_9 збільшиться відповідно на h і в момент t_9 перекомпенсує сигнал. В лічильнику $D6$ і відповідно в регістрі $D8$ буде число 5 і т.д.

Отже, еталонна напруга U_0 змінюється не на весь свій діапазон, як в АЦП розгортального перетворення, а лише на декілька кроків квантування, “слідкуючи” за вхідною напругою U_x . Це й підвищує швидкодію.

Щодо недоліків слідкуючого АЦП, то головний з них полягає в тому, що АЦП має зворотний зв'язок ($A1 - D6$). Через це слідкуючі АЦП мають схильність до самозбудження. Тому підсилення $A1$ має бути менше за АЦП розгортального перетворення, через що й роздільна здатність менше.

Наступним недоліком є те, що час перетворення хоча і менше за АЦП послідовного наближення, але може займати декілька тактових інтервалів.

Цей недолік усунуто в АЦП паралельного кодування, де час перетворення завжди дорівнює лише одному тактовому інтервалу.

2.3.4. АЦП паралельного кодування

АЦП паралельного кодування – це найбільш швидкодіючі перетворювачі. Висока швидкодія цього типу АЦП забезпечується за рахунок одночасного порівняння вхідного сигналу з багатьма N квантами опорної напруги на N компараторах.

Найпростіша функціональна схема семирозрядного АЦП паралельного кодування (далі – паралельного АЦП) показана на рис. 2.13.

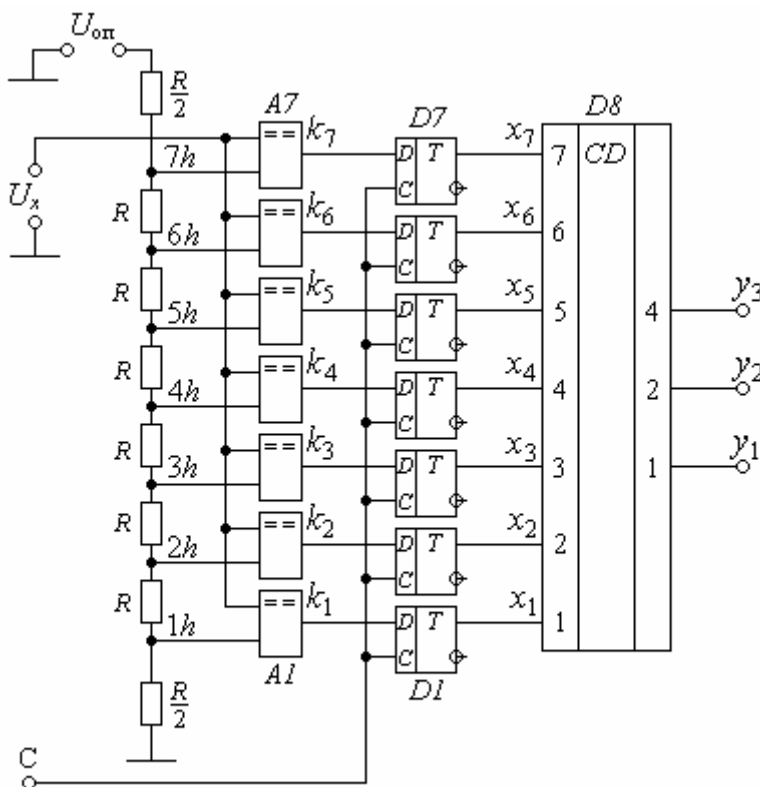


Рисунок 2.13 – ЦАП паралельного кодування

АЦП містить подільник опорних напруг на резисторах $R/2, R, R, \dots, R/2$; компаратори $A1 - A7$; тригери $D1 - D7$ та пріоритетний шифратор $D8$.

Компаратори порівнюють вхідну напругу U_x з усіма опорними напругами $1h, 2h, \dots, 7h$, кількість яких дорівнює числу розрядів.

Тригери запам'ятовують номери рівнів опорної напруги, до якого наближена вхідна напруга U_x . Ці тригери необхідні для того, щоб гонки між виходами компараторів не впливали на роботу шифратора.

Подільник опорних напруг є послідовним з'єднанням високоточних резисторів, які утворюють шину ділення стабілізованого джерела опорної напруги $U_{оп}$ з виводами через кожний крок квантування. Кількість виводів дорівнює числу розрядів в коді "1 із n ".

Усі виходи подільника з'єднанні з першими входами компараторів $A1 - A7$, на другі входи яких подана вхідна напруга U_x , що підлягає АЦ перетворенню.

Виходи компараторів $k_1 \dots k_7$ з'єднанні відповідно з D -входами тригерів $D1 - D7$, прямі виходи яких підключені до входів $x_1 \dots x_7$ шифратора $D8$.

Паралельний АЦП працює наступним чином.

Вхідна напруга U_x одночасно порівнюється з опорними рівнями, які отримані з подільника. В залежності від неї виходи компараторів набувають значень, які надані в таблиці станів (табл. 2.2).

Таблиця 2.2 – Стани елементів паралельного АЦП

Вхідна напруга U_x	Стани компараторів							Стани шифратора			Десятькове число
	k_7	k_6	k_5	k_4	k_3	k_2	k_1	y_3	y_2	y_1	
0	0	0	0	0	0	0	0	0	0	0	0
$1h$	0	0	0	0	0	0	1	0	0	1	1
$2h$	0	0	0	0	0	1	1	0	1	0	2
$3h$	0	0	0	0	1	1	1	0	1	1	3
$4h$	0	0	0	1	1	1	1	1	0	0	4
$5h$	0	0	1	1	1	1	1	1	0	1	5
$6h$	0	1	1	1	1	1	1	1	1	0	6
$7h$	1	1	1	1	1	1	1	1	1	1	7

Стани компараторів $k_1 \dots k_7$ за командою C запам'ятовуються відповідно тригерами $D1 - D7$. Завдяки цим тригерам входи шифратора $x_1 \dots x_7$ позбавлені аналогових змін напруг, які діють на виходах компараторів. Тому входи шифратора $D8$ не зазнають гонок, що викликають його хибні спрацьовування. Щодо станів $x_1 \dots x_7$, то вони збігаються зі станами $k_1 \dots k_7$.

Зі схеми (рис. 2.13) і табл. 2.2 видно, що на входи $x_1 \dots x_7$ шифратора $D8$ надходить номер того рівня опорної напруги, якому дорівнює вхідна напруга U_x з похибкою $\pm h/2$, де h – крок квантування.

Отже, будь-яка вхідна напруга за будь-якої кількості кроків квантування перетворюється на число за лише один такт синхроімпульсу C . Тому швидкодія паралельного АЦП найбільша.

Умовне позначення АЦП показано на рис. 2.14.

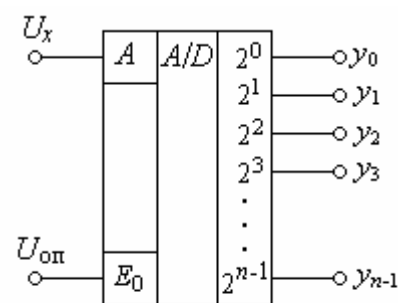


Рисунок 2.14 – Умовне позначення АЦП

Недоліком АЦП паралельного кодування є значні апаратні витрати, які пропорційні розрядності перетворювача. Наприклад, при заданій припустимій похибці перетворення 1%, тобто кроку квантування 1%, згідно з формулою

(2.27), потрібно 100 кроків квантування. Для цього необхідно мати понад 100 резисторів, 100 компараторів, 100 тригерів і 100-входовий шифратор.

Реалізувати ці вимоги складно.

Зазначений недолік частково усунений в *послідовно-паралельних* АЦП, але за рахунок зменшення швидкодії вдвічі.

2.3.5. Послідовно-паралельні АЦП

В *паралельно-послідовних* АЦП використовують два АЦ перетворення: спочатку вхідну напругу перетворюють на число з великим кроком квантування, а потім залишену частину вхідної напруги в межах до одного великого кроку перетворюють дрібними кроками квантування.

Нехай припустима похибка АЦ перетворення становить $\delta = 1\%$. Тоді крок квантування $h = \delta$ також дорівнюватиме 1%, а кількість кроків досягне $1/0,01 = 100$. Як зазначалося раніше, для цього необхідне обладнання 100 унітарних розрядів.

Використаємо два АЦП по 10 кроків квантування в кожному. Кроки квантування дорівнюватимуть відповідно 10% першого АЦП та 1% – другого. При цьому перший АЦП компенсуватиме вхідний сигнал з похибкою $\pm 5\%$, а залишок $< 10\%$ компенсуватиме другий АЦП з похибкою $\pm 0,5\%$. Тоді крок квантування, як і в паралельному АЦП становить 1%, але необхідні лише 20 унітарних розрядів, а не 100.

Схема такого паралельно-послідовного АЦП показана на рис. 2.15.

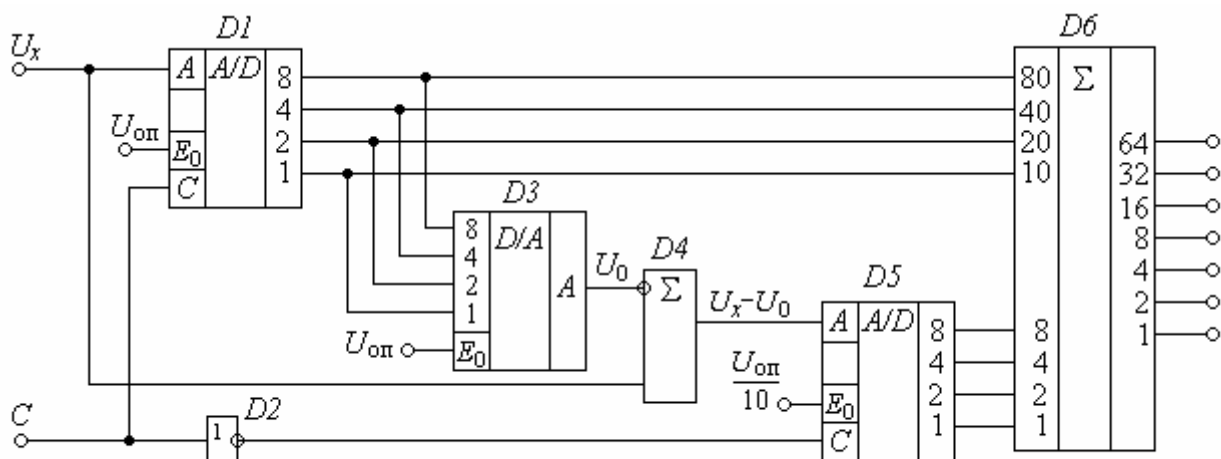


Рисунок 2.15 – Паралельно-послідовний АЦП

Схема містить два АЦП $D1$ і $D5$. АЦП $D1$ має опорну напругу $U_{оп}$, а АЦП $D5$ – у 10 разів менше. Тому крок квантування АЦП $D1$ становить $h_1 = U_{оп}/10 = 0,1U_{оп}$ (тобто 10%), а АЦП $D5$ має крок квантування $h_2 = 0,1U_{оп}/10 = 0,01U_{оп}$ (тобто 1%).

АЦП працює наступним чином.

Завдяки інвертору $D2$ перетворювачі $D1$ і $D5$ спрацьовують по черзі: спочатку фронт $C = 1$ виставляє число на виходах $D1$, а через напівперіод при $C = 0$ фронт $\bar{C} = 1$ з'явиться на виході $D2$ і виставить число на виходах $D5$.

Число АЦП $D1$ за допомогою ЦАП $D3$ перетворюється на аналогову напругу U_0 . Ця напруга, яка отримана з точністю $h_1 = 10\%$, віднімається віднімачем $D4$, на виході якого буде неперетворена частина $U_x - U_0$ вхідної напруги U_x .

Залишок неперетвореної частини $U_x - U_0$ потрапляє на другий АЦП $D5$, в якому вона перетворюється з точністю $h_2 = 1\%$.

Результати перетворення з виходів АЦП $D1$ потрапляють до входів старших розрядів цифрового суматора $D6$, а з виходів АЦП $D2$ – до входів молодших розрядів. Вагові коефіцієнти цих розрядів мають відрізнятись в нашому випадку в 10 разів.

Нехай вхідна напруга, яку треба перетворити на число, дорівнює $U_x = 99h_2$. Тоді число на виході суматора $D6$ має дорівнювати 99. Це здійснюється наступним чином. На виходах АЦП $D1$ буде число $8 + 1 = 9$. Суматором $D6$ воно сприймається як 90.

На виходах АЦП $D2$ буде число $8 + 1 = 9$. Разом $90 + 9 = 99$. При цьому на виходах суматора $D6$ буде двійкове число 1100011, десятковий еквівалент якого становить $64 + 32 + 3 = 99$.

Так, число 99 отримане не 99 унітарними розрядами, а лише двадцятьма.

Недоліком послідовно-паралельних АЦП є подвійний час перетворення порівняно з паралельним АЦП та досить значна кількість елементів, але кращого швидкодіючого АЦП сьогодні немає.

Контрольні питання

- 2.1. Які функції виконують аналого-цифрові (АЦП) та цифро-аналогові перетворювачі (ЦАП)?
- 2.2. Що таке матриця на двійково-зважених резисторах?
- 2.3. Які вузли входять до складу схеми ЦАП на двійково-зважених резисторах?
- 2.4. Які переваги та недоліки мають ЦАП на основі матриці резисторів $R-2R$?
- 2.5. У чому полягає принцип АЦ перетворення? З яких процесів він складається?
- 2.6. Що таке АЦП розгортальної дії? Які переваги та недоліки мають такі АЦП?
- 2.7. Що таке слідкуючі АЦП?
- 2.8. Що таке АЦП паралельного кодування? Які переваги та недоліки мають такі АЦП?
- 2.9. Що таке послідовно-паралельні АЦП? Які переваги та недоліки мають такі АЦП?

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. –

С. 318 – 343.

2. Титце У. Полупроводниковая схемотехника: Справочное руководство / У. Титце, К. Шенк. – М.: Мир, 1982. – С. 444 – 464.

3. Орнатский П.П. Автоматические измерения и приборы (аналоговые и цифровые) / Орнатский П.П. – К.: Вища шк., 1986. – 504 с.

4. Авт. свид. СССР № 932272, кл. G 01 G 23/36. Аналого-цифровой преобразователь для тензометрических весов/ В.Д. Иванченко, И.П. Панфилов, Ю.В. Флейта и др.; Бюл. № 20, 1982

5. Авт. свид. СССР № 1101684, кл. G 01 G 23/36. Цифровой прибор тензометрических весов/ Е.М., Воробьева, В.Д., Иванченко, И.П. Панфилов и др.; Бюл. № 25, 1984.

6. Авт. свид. СССР № 1541525, кл. G 01 R 3/147. Цифровой вольтметр для тензометрических весов/ Е.М. Воробьева, В.Д. Иванченко, И.П. Панфилов и др.; Бюл. № 5, 1990.

3. ВСТУП ДО МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ (МТ)

3.1. Переваги використання МТ

Сьогодні неможливо уявити сферу діяльності людини, де б не використовувалися мікропроцесори та мікроконтролери.

Більшість сфер людського життя при використанні мікропроцесорної техніки змінилися у кращий бік. Переваги мікропроцесорних пристроїв (малі габарити, економічність енергоспоживання, велика швидкодія) дозволили прискорити наукові дослідження, автоматизувати виробничі процеси, збільшити продуктивність праці. Для мікропроцесорних систем характерні висока гнучкість та можливість перенастроювання для розв'язання поточних завдань при застосуванні алгоритму роботи. Універсальність мікропроцесорних систем і порівняно низька ціна забезпечує їх доступність широкому загалу користувачів.

На основі мікропроцесорів (МП) створюються сучасні комп'ютери та складні радіотехнічні системи, пристрої зв'язку та «розумна» побутова техніка. Сучасний комп'ютер може обробляти числову, текстову, графічну та звукову інформацію. Інформація для оброблення повинна бути подана у вигляді, що сприймається комп'ютером. Структурна схема перетворення інформації в мікропроцесорній системі зображена на рис. 3.1.

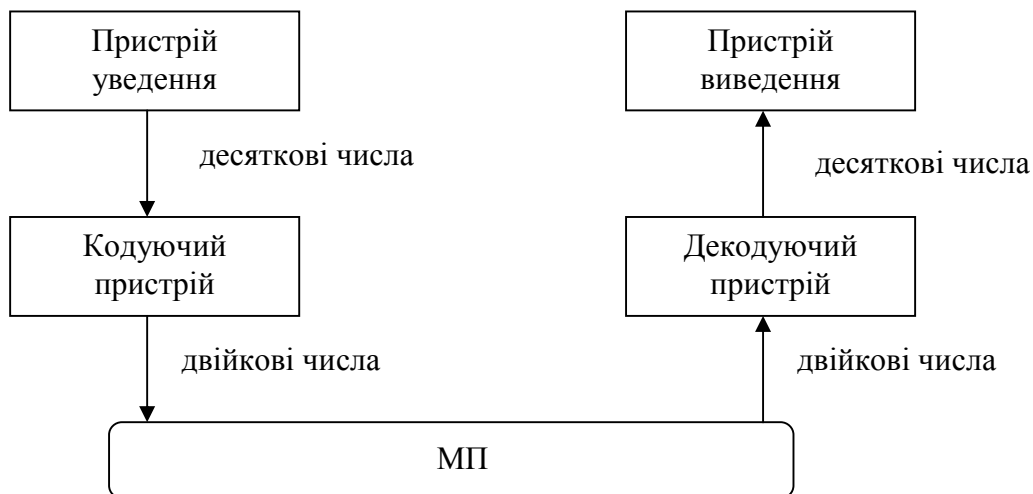


Рисунок 3.1 – Структурна схема перетворення інформації в процесорній системі

Інформація уводиться, звичайно, з низькочастотних датчиків та пристроїв керування, з клавіатури комп'ютера, що різко знижує загальну швидкість системи.

Комп'ютер обробляє інформацію, подану у вигляді двійкових чисел. Інформація може уводитися в комп'ютер у більшості випадків з клавіатури датчиків і т.ін. у десятковому коді. Тому інформація від джерела має бути перетворена у двійковий код. В якості такого перетворювача може бути використаний шифратор – перетворювач десяткового коду у двійковий.

Процесор – електронний блок (або інтегральна схема – мікропроцесор), який виконує машинні коди програми, є головною частиною апаратного забезпечення комп'ютера або програмованого логічного контролера.

Після оброблення інформація з процесора надходить на перетворювач двійкового коду у десятковий. Таким перетворювачем слугує дешифратор. Далі інформація надходить на пристрій виведення, наприклад, на екран монітора, на принтер, апарат передавання даних і т.ін.

3.2. Структура мікропроцесорної системи (МС)

Принцип дії мікропроцесорної системи аналогічний дії будь-якої електронної обчислювальної машини (ЕОМ), які виконують обчислення на основі алгоритмів. Алгоритм – це будь-який точний припис, який задає обчислювальний процес, що починається з довільних вихідних даних та спрямований на отримання повністю визначеного цими вихідними даними результату.

Кожний алгоритм характеризують:

- сукупність можливих вихідних даних;
- сукупність можливих проміжних та кінцевих результатів;
- правилами початку, перероблення, закінчення та витягнення результату.

Важливими властивостями алгоритмів, які найбільш суттєво впливають на організацію МС, є:

- дискретність інформації, якою оперують алгоритми;
- кінцевість та елементарність набору операцій, які виконуються при реалізації алгоритмів;
- детермінованість обчислювальних процесів, що породжуються алгоритмами.

Елемент даних, які беруть участь в операціях як деякі величини, називають *операндом*.

На МС можна автоматично розв'язати будь-яку задачу, якщо вона подана у вигляді певних дій над операндами, що описуються системою формальних правил та умов, названих *алгоритмами розв'язання задач*.

Алгоритм, поданий сукупністю команд, називається *програмою*. Фактично – це запис машинною мовою послідовності виконання операцій (команд), необхідних для розв'язання задачі. Мова, якою людина формулює послідовність операцій для розв'язання задачі, а МС сприймає її, – називається *машинною мовою*.

Вперше спосіб реалізації програмного керування був запропонований Дж. фон Нейманом у 1945 р. Цей спосіб називається нейманівським принципом програмного управління і використовується нині як основний принцип побудови МС.

Для цього принципа характерне наступне:

- різнотипні слова інформації різняться за способом використання, але не способами кодування;

- слова інформації розміщуються у комітках пам'яті, які нумеруються. Номера комірок називаються *адресою слова*.

Найпростіша нумерація комірок – лінійна. Це нумерація послідовними номерами. Будь-які інші структури, якими оперує програміст, повинні перетворюватися лише з цієї структури.

Машинна мова надана словом у двійковому коді. Для усіх способів надання даних до МС характерна неподільність слів інформації: слово оброблюється повністю як самостійний об'єкт – машинний елемент інформації.

Керуючі слова – команди – також являють собою двійковий код, який складається з кількох частин (полів) – коду операції та адрес операндів. Код операції задає машинну операцію, наприклад, операцію множення. МС реалізує набір функцій, який повинен володіти властивістю повноти, тобто бути достатній для описання алгоритму.

Дані та команди записані до комірок пам'яті. Місця даних та команд у пам'яті МС позначаються за допомогою адрес, що являють собою номер комірки. Читання та запис інформації при використанні пам'яті виконується з обов'язковим передаванням до пам'яті адреси комірки. При читанні інформації з комірки її зміст не руйнується і може бути прочитаний багато разів. При записуванні інформації до комірки попередній стан змінюється на новий.

Така операція заміни значення називається *операцією присвоєння* та позначається як =.

У команді мають бути зазначені адреси операндів. Використання у команді адрес замість самих операндів надає програмі універсальність формули: одна й та сама програма буде працювати при записуванні до комірки різних даних. Кількість адрес у команді залежить від кількості її операндів.

Для бінарних операцій (додавання, віднімання) необхідно два операнди, а у команді треба мати три адреси: дві – для зазначення операндів; одна – для адреси комірки, куди повинен бути розміщений результат.

Всі ці дані розташовані у команді у певному (для кожної МС) порядку, що задається форматом команди, тобто схемою розташування двійкових цифр у команді, що дозволяє розрізняти її складові частини та визначати їх функції.

Розглянемо спрощену структуру МС, що призначена для оброблення даних або управління деяким процесом, яка показана на рис. 3.2.

Структура МС – це сукупність елементів і зв'язків між ними. Усі МС мають наступні функціональні блоки:

- процесор (ЦПЕ), що складається з арифметико-логічного пристрою (АЛП) та пристрою керування (ПК);
- пам'ять (внутрішню та зовнішню);
- пристрій уведення та пристрій виведення інформації.

Об'єднання функціональних вузлів та блоків МС здійснюється за допомогою наступних шин:

- шина даних, якими здійснюється обмін інформацією між блоками;
- шина адреси, що використовується для передавання адрес, якими здійснюється звертання до різних пристроїв МС;

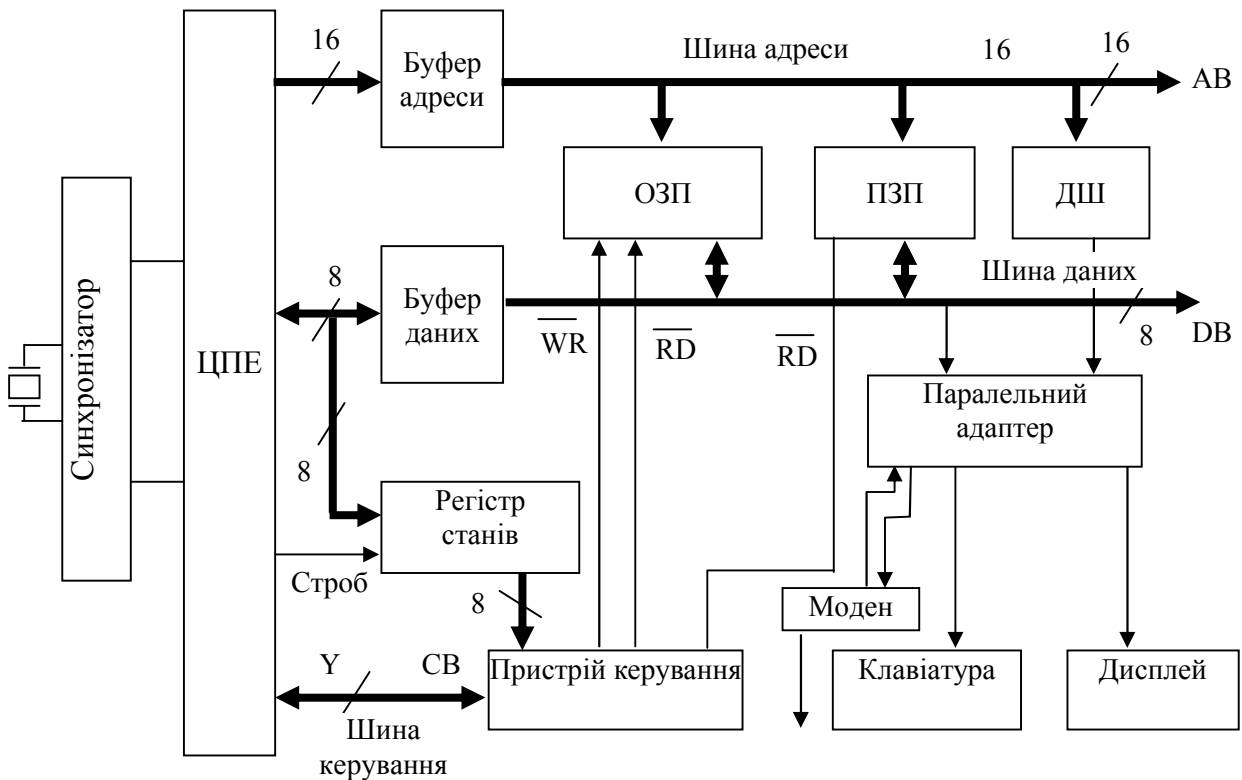


Рисунок 3.2 – Структура МС

– шина керування для передавання керуючих сигналів.

Протоколи (порядок обміну даними, структура шин та швидкодія) при обміні можуть бути оптимальними.

У цій системі центральне місце займає процесор, який виконує арифметичні та логічні операції над даними, здійснює програмне керування пристроєм оброблення інформації, організує взаємодію усіх пристроїв, що входять до системи. Робота процесора здійснюється під дією сигналів схеми синхронізації й таймера, який дуже часто виконується у вигляді окремого кристалу.

Структурна (рис. 3.2) схема відображає магістрально-модульний принцип організації мікропроцесорних пристроїв і систем. Окремі блоки є функціонально закінченими модулями з вбудованими схемами керування, виконаними у вигляді одного або кількох кристалів ВІС, укладених у корпуси з відповідною кількістю виводів.

Міжмодульний обмін інформацією здійснюється за допомогою колективних шин (магістралей), до яких мають доступ всі основні модулі системи. У будь-який момент часу можливий обмін лише між двома модулями системи. Таким чином, обмін інформацією здійснюється шляхом розподілу у часі модулями системи шин (магістралей). Це є магістральний принцип побудови поєднання модулів (інтерфейсу) мікропроцесорної системи МС. Цей принцип припускає наявність інформаційно-логічної сумісності модулів, яка реалізується шляхом використання єдиних способів подання інформації, алгоритму керування обміном, форматів команд керування обміном та способу синхронізації.

Для мікропроцесора характерна тришинна структура, яка містить шину адреси ША, двоспрямовану шину даних ШД і шину керування ШК. Типова структура МП-системи припускає наявність поєднання (загального або єдиного) інтерфейсу для модулів пам'яті постійних (ПЗП) та оперативних (ОЗП), периферійних пристроїв (зовнішніх ЗП) та пристроїв введення – виведення (ПВВ).

Периферійний пристрій (ПП) приєднується до шин інтерфейсу (шина МП) не безпосередньо, а через програмований периферійний адаптер (ППА) і програмований зв'язний адаптер (ПЗА), які обслуговують ПП відповідно передаванню інформації паралельним та послідовним кодом. Наявність програмно-вбудованих адаптерів робить значно гнучкішою та функціонально багатою систему введення-виведення інформації до МС.

Наявність єдиного інтерфейсу для модулів пам'яті та периферійних пристроїв, крім того, пояснюється обмеженою кількістю зовнішніх виводів корпусу МП.

До структури МП інтерфейс є вузьким місцем через обмежену кількість виводів, тому необхідно використовувати двоспрямовану лінію передавання інформації. Це ускладнює схеми буферів і вимагає використання часового мультиплексування шин. За такого методу шинами передається різноманітна інформація, однак треба мати спеціальні лінії для ідентифікації передаваної інформації. Швидкість передавання через інтерфейс при цьому знижується.

Обмежена пропускна здатність інтерфейсу МП знижує продуктивність МС і призводить до того, що МС працює з невеликою кількістю джерел та споживачів інформації.

3.3. Узагальнена структура мікропроцесора (МП)

Процесор (ЦПП) – це функціональний блок обчислювального пристрою, призначений для реалізації оброблення цифрових даних та управління ходом цього оброблення. Його іноді називають *мікропроцесором* або просто *процесором*.

Головними характеристиками ЦПП є тактова частота, продуктивність, енергоспоживання, норми метографічного процесу, використовуваного при виробництві, та архітектура.

В узагальненому вигляді функціонування процесора може бути подано, як циклічне чергування двох етапів: 1) вибірка команд з пам'яті та їх дешифрування; 2) виконання команд.

Вибірка команд – це автоматичний процес, який здійснюється під впливом імпульсів від генератора тактових імпульсів (ГТІ). Механізм реалізації залежить лише від апаратної структури процесора.

При дешифруванні команд формується послідовність керуючих сигналів для усіх вузлів процесора та інших блоків на основі коду, що міститься у команді.

Дії, які виконуються у відповідності з командою, можуть являти собою арифметичне або логічне оброблення, пересилання даних, формування адреси

наступної команди або зміну режиму роботи МП. Ці дії задає програміст у межах тих команд, які властиві даному МП.

Після виконання команди, процесор автоматично переходить до вибірки наступної команди з пам'яті. Сучасні мікропроцесори суттєво різняться набором функціональних блоків та зв'язками між ними. Тим не менш, у структурі будь-якого процесора можна виокремити основні елементи, що визначають специфіку процесора як управляючого центру обчислювача. Перш за все, мова йде про два блоки: пристрій керування та операційний пристрій. Узагальнена структура мікропроцесора показана на рис. 3.3.

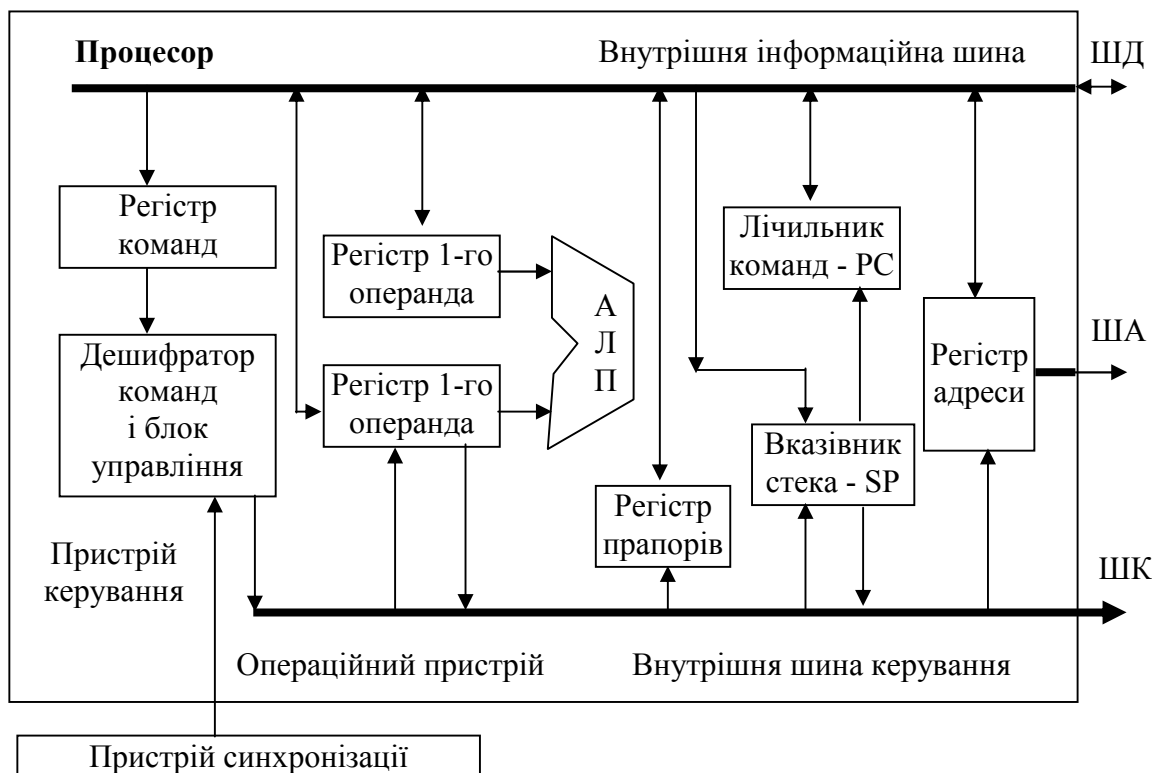


Рисунок 3.3 – Узагальнена структура мікропроцесора

Пристрій керування (ПК) призначений для реалізації вибірки команд, їх дешифрування, й на основі цього – для управління обміном та обробленням інформації шляхом генерування послідовності керуючих сигналів.

Операційний пристрій (ОП) слугує для оброблення цифрової інформації (арифметичні або логічні дії, зміщення, аналіз чисел і т.ін.).

Основним елементом для зберігання інформації всередині процесора є регістри (РГ1, РГ2), які виконують функцію надоперативного ОЗП з мінімальним часом записування й зчитування.

Регістр команд (РГК) (англ. instruction register) використовується для фіксації кода команди після зчитування її з пам'яті. У цьому регістрі фіксується лише код операції (КОП) – частина коду команди, що визначає виконувану дію та спосіб адресації операндів.

Регістри операндів слугують для зберігання даних у процесі їх оброблення та дозволяють уникати постійних звертань до пам'яті. У сучасних процесорах кількість регістрів операндів може досягати 10...15. По суті, вони

утворюють внутрішню оперативну пам'ять процесора. В однокристальних мікроконтролерах кількість регістрів операндів доведена до кількох десятків. Тому їх іноді називають *регістровими файлами*. Деякі регістри використовуються для зберігання або формування адрес інших операндів, тобто на їх основі реалізується механізм непрямой адресації.

Лічильник команд (англ. PC-programmingcounter) – регістр, в якому при вибірці або виконанні поточної команди формується адреса наступної команди. Модифікації змісту регістра PC – це засіб управління послідовністю вибірки команд з пам'яті і, отже, управління ходом обчислювального процесу.

Вказівник стека (англ. SP-stackpointer) – регістр, в якому при виконанні програми зберігається адреса межі тієї області пам'яті, для якої програміст використовує принцип послідовного доступу до даних (так званий протокол роботи зі стеком).

Регістр адреси – регістр, в якому формується адреса будь-якого пристрою, зовнішнього по відношенню до МП (комірка пам'яті або порту введення/виведення), перед звертанням до цього пристрою. Даний регістр необхідний, тому що джерелами адресної інформації можуть бути різні регістри процесора. Регістр адреси відіграє роль накопичувального буфера, з якого адресна інформація видається на зовнішню шину адреси.

Результат операції, що виконується в АЛП, заноситься до одного з регістрів або пересилається до пам'яті (залежно від команди). В регістрі ознак автоматично формуються ознаки, що характеризують цей результат.

Регістр ознак (англ. F-flags) – це елемент внутрішньої пам'яті, в якому у вигляді окремих біт фіксуються ознаки, що характеризують результат операції, виконаної в АЛП (нульовий результат, переповнення стека і т.ін.).

Арифметично-логічний пристрій (АЛП) – функціональний блок МП, призначений для реалізації дій з оброблення даних.

3.4. Пристрій мікроконтролерів AVR

Мікроконтролер AVR містить швидкий RISC-процесор, два типи енергозалежної пам'яті (Flash – пам'ять програм і пам'ять даних EEPROM), оперативну пам'ять RAM, порти введення/виведення та різноманітні інтерфейсні схеми.

3.4.1. Процесор

Серце мікроконтролера AVR – це 8-бітне мікропроцесорне ядро, побудоване на принципах RISC-архітектури. Основою цього блока є арифметико-логічний пристрій (АЛП). За системним тактовим сигналом з пам'яті програм відповідно змісту лічильника команд (Program Counter – PC) обирається чергова команда та виконується в АЛП. Під час вибору команди з пам'яті програм відбувається виконання попередньої вибраної команди, що дозволяє досягти швидкодії 1 MIPS на 1 МГц.

Структурну модель мікроконтролера AVR показано на рис. 3.4.

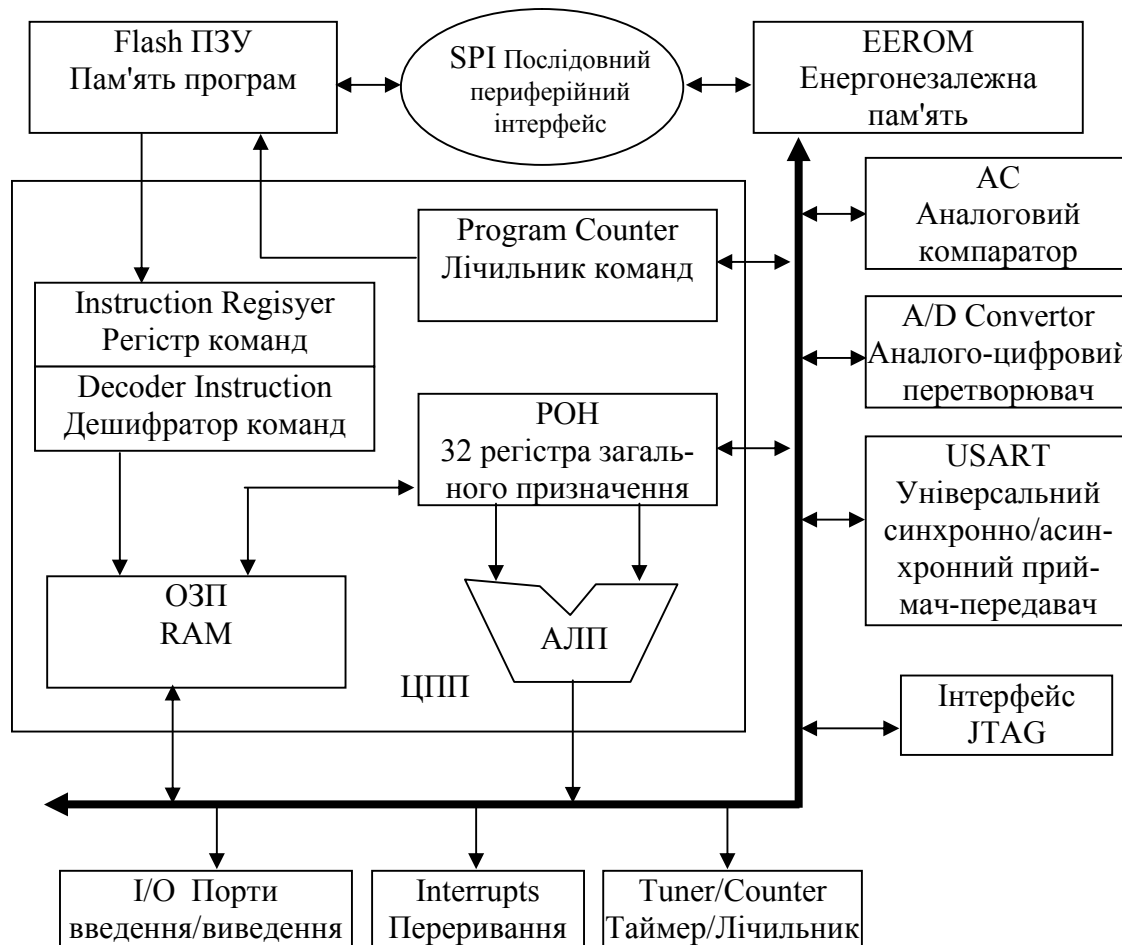


Рисунок 3.4 – Структурна модель мікроконтролера AVR

АЛП відімкнений до регістрів загального призначення РОН (General Purpose Registers – GPR). Регістрів загального призначення всього 32, вони мають байтів формат, тобто кожен з них складається з восьми біт. РОН знаходиться на початку адресного простору оперативної пам'яті, але фізично не є її частиною. Тому до них можна звертатися двома способами (як до регістрів, так і до пам'яті). Таке рішення є особливістю AVR та підвищує ефективність роботи і продуктивність мікроконтролера.

Відмінність між регістрами та оперативною пам'яттю полягає у тому, що з регістрами можна виконувати будь-які операції (арифметичні, логічні, бітові), а до оперативної пам'яті можна лише записувати дані з регістрів.

3.4.2. Пам'ять

У мікроконтролерах AVR реалізована Гарвардська архітектура (Harvard architecture), відповідно з якою розділені не лише простори пам'яті програм та пам'яті даних, а й шини доступу до них. Кожна з областей пам'яті даних (оперативна пам'ять та EEPROM) також розташовані у своєму адресному просторі.

Пам'ять програм (Flash ROM або Flash ПЗП)

Пам'ять програм призначена для зберігання послідовності команд, управляючих функціонуванням мікроконтролера, й має 16-бітну організацію.

Усі AVR мають Flash-пам'ять програм, яка може мати розмір від 1 до 256 кбайт. Її головна перевага полягає у тому, що вона побудована на принципі електричного перепрограмування, тобто припускає багаторазове записування та стирання інформації. Програма заноситься до Flash-пам'яті AVR або за допомогою звичайного програматора, або за допомогою SPI-інтерфейсу, у тому числі безпосередньо на власній платі. Усі мікропроцесори сімейства Mega мають можливість самопрограмування, тобто самостійно змінювати зміст своєї пам'яті програм. Ця особливість дозволяє створювати на їх основі досить гнучкі системи, алгоритм роботи яких буде змінюватися самим мікроконтролером залежно від якихось внутрішніх умов або зовнішніх подій.

Гарантована кількість циклів Flash-пам'яті у мікроконтролерів AVR другого покоління складає не менше 10 тисяч циклів при типовому значенні 100 тисяч циклів.

Пам'ять даних

Пам'ять даних розподілена на три частини: регістрова пам'ять, оперативна пам'ять (ОЗП) та енергонезалежна пам'ять (ЕСПЗП або EEPROM).

Регістрова пам'ять (РОН та РВВ)

Регістрова пам'ять містить 32 регістри загального призначення (РОН або GPR), об'єднаних у файл, та службові регістри введення/виведення (РВВ). І одні, й інші розташовані в адресному просторі ОЗП, але не є його частиною.

В області регістра введення/виведення розташовані службові регістри (регістр керування мікроконтролером, регістри станів і т.ін.), а також регістри керування периферійними пристроями, що входять до складу мікроконтролера. По суті, управління мікроконтролером полягає в керуванні цими регістрами.

Енергонезалежна пам'ять даних (EEPROM)

Для довготривалого зберігання різноманітної інформації, яка може змінюватися в процесі функціонування мікроконтролерної системи, використовується EEPROM-пам'ять. Усі AVR мають блок енергонезалежної електрично перезаписуваної пам'яті даних EEPROM від 64 байт до 4 кбайт. Цей тип пам'яті доступний програмі мікроконтролера безпосередньо у ході її виконання, зручний для зберігання проміжних даних, різних констант, коефіцієнтів і т.п. EEPROM може бути завантажена як ззовні через SPI інтерфейси, так і за допомогою звичайного програматора. Кількість циклів записування/стирання не менше 100 тисяч.

Оперативна пам'ять (ОЗП або RAM)

Внутрішня оперативна статична пам'ять Static RAM (SRAM) має байтів формат та використовується для оперативного зберігання даних.

Розмір оперативної пам'яті може варіюватися у різних чипів від 64 байт до 4 кбайт. Кількість циклів читання та записування до RAM не обмежена, але при відключенні живлячої напруги уся інформація втрачається.

Для деяких мікроконтролерів можлива організація відімкнення зовнішнього статичного ОЗП об'ємом до 64 К.

3.4.3. Периферія

Периферія мікроконтролерів AVR містить:

- порти (від 3 до 48 ліній введення та виведення);
- підтримку зовнішніх переривань;
- таймери-лічильники;
- очікуваний таймер;
- аналогові компаратори;
- 10-розрядний 8-канальний АЦП;
- інтерфейси UART, JTAG та SPL;
- пристрої скидання при зниженому живленні;
- широтно-імпульсні модулятори.

3.4.4. Переривання (INTERRUPTS)

Система переривань – одна з важливих частин мікроконтролера. Усі мікроконтролери AVR мають багаторівневу систему переривань. Переривання припиняє нормальний хід програми для виконання пріоритетної задачі, що визначається внутрішньою або зовнішньою подією.

Для кожної такої події розробляється окрема програма, яку називають *підпрограмою оброблення запиту на переривання* та розміщують у пам'яті програм.

При виникненні події, що викликає переривання, мікроконтролер зберігає зміст лічильника команд, перериває виконання центральним процесором поточної програми й переходить до виконання підпрограми оброблення переривання.

Після виконання підпрограми переривання здійснюється відновлення попередньо збереженого лічильника команд та процесор повертається до виконання перерваної програми.

Для кожної події може бути встановлений пріоритет. Поняття пріоритет означає, що виконувана підпрограма переривання може бути перервана іншою подією лише за умови, що вона має більш високий пріоритет, ніж поточна. В іншому випадку центральний процесор перейде до оброблення нової події лише після закінчення оброблення попередньої.

3.4.5. Таймери-лічильники (TIMER-COUNTERS)

Мікроконтролери AVR мають у своєму складі від 1 до 4 таймерів-лічильників з розрядністю 8 або 16 біт, які можуть працювати і як таймери від внутрішнього джерела тактової частоти, і як лічильники зовнішніх подій.

Їх можна використовувати для точного формування часових інтервалів, підрахунку імпульсів на виводах мікроконтролера, формування послідовності імпульсів, трактування приймально-передавального послідовного каналу зв'язку. У режимі ШІМ таймер-лічильник може являти собою широтно-імпульсний модулятор та використовуватися для генерування сигналу з програмованою частотою та щільністю. Таймери-лічильники здатні виробляти запити переривань, перемикаючи процесор на їх обслуговування за подіями та звільняючи його від необхідності періодичного запиту стану таймерів. Оскільки

основне застосування мікроконтролери знаходять у системах реального часу, таймери-лічильники є одним із найбільш вагомих елементів.

3.4.6. Аналого-цифровий перетворювач (A/D convertor)

Аналого-цифровий перетворювач (АЦП) слугує для отримання числового значення напруги, поданої на його вхід. Цей результат зберігається у регістрі даних АЦП. Який з виводів мікроконтролера є входом АЦП, визначається числом, занесеним у відповідний регістр.

3.4.7. Універсальний послідовний приймач-передавач (UART або USART)

Універсальний асинхронний або універсальний синхронний прийомопередавач – зручний та простий послідовний інтерфейс для організації інформаційного каналу обміну мікроконтролера із зовнішнім світом. Здатен працювати у дуплексному режимі (одночасне передавання та приймання даних). Він підтримує протокол стандарту RS-232, що забезпечує можливість організації зв'язку з персональним комп'ютером. Для стикування мікроконтролера та комп'ютера обов'язково необхідна схема поєднання рівнів сигналу. Для цього існують спеціальні мікросхеми, наприклад, MAX232.

3.4.8. Інтерфейс JTAG

Інтерфейс JTAG був розроблений групою провідних спеціалістів з проблем тестування електронних компонентів та був зареєстрований як промисловий стандарт (IEEE). Чотиридротовий інтерфейс JTAG використовується для тестування друкованих плат, внутрішньосхемного налагодження, програмування мікроконтролерів.

Багато мікроконтролерів сімейства MEGA мають сумісний з IEEE інтерфейс JTAG або debug WIRE для вбудованого налагодження. Крім того, всі мікроконтролери MEGA з флеш-пам'яттю ємністю 16 кбайт і більше можуть програмуватися через інтерфейс JTAG.

3.4.9. Стек

У комірках оперативної пам'яті організується системний стек, який використовується автоматично для зберігання адрес повернення при виконанні підпрограм, а також може використовуватися програмістом для тимчасового зберігання змісту оперативних регістрів (команди PUSH і POP). Мікроконтролери, що не мають SRAM, містять трирівневий апаратний стек.

Слід мати на увазі, що якщо стек розміщений на зовнішній SRAM, то виклики підпрограм та повернення з них вимагають двох додаткових циклів, якщо біт SRW не встановлений, та чотирьох, якщо встановлений.

Розмір стека, що організується в оперативній пам'яті, обмежений лише розмірами цієї пам'яті. Якщо мікроконтролер містить на кристалі 28 байт внутрішньої SRAM та не має можливості відімкнення зовнішньої SRAM, то в якості вказівника вершини стека використовується регістр введення/виведення SPL. Якщо є можливість відімкнення зовнішньої пам'яті або внутрішня пам'ять

має розміри 256 байт і більше, то вказівник стека складається з двох регістрів введення/виведення SPL і SPH.

При занесенні числа до стека автоматично виконуються наступні дії:

- число записується до комірки пам'яті за адресою, що зберігається у вказівнику стека. При цьому $(SPH : SPL) < \text{число}$;
- вміст вказівника стека зменшується на одиницю: $SPH : SPL = SPH : SPL - 1$.

Обернені дії виконуються при витягненні числа зі стека:

- вміст вказівника збільшується на одиницю: $SPH : SPL = SPH : SPL + 1$;
- число витягується з комірки пам'яті з адресою, що зберігається у вказівнику стека: $(SPH : SPL) > \text{число}$.

Таким чином, стек зростає від старших адрес до молодших, тому, враховуючи, що початкове значення вказівника стека після скидання дорівнює нулю, програміст AVR обов'язково повинен в ініціалізованій частині програми потурбуватися про встановлення вказівника стека, якщо він припускає використати хоча б одну підпрограму.

Контрольні питання

1. Накресліть структурну схему перетворення інформації в МС.
2. Що таке програма?
3. Що таке найманівський принцип програмного управління?
4. Що таке лінійна нумерація комірок?
5. Куди записуються дані і команди?
6. За допомогою яких шин здійснюється об'єднання функціональних блоків МС?
7. Що таке модульний принцип організації мікропроцесорних пристроїв?
8. Що таке процесор?
9. Назвіть основні блоки структури мікропроцесора.
10. Які види пам'яті використовуються у МС?
11. Що таке периферія?
12. Що таке таймер-лічильник та де він використовується?
13. Навіщо у МК вмикають універсальний приймач-передавач?
14. Що таке стек?

4. ЗАПАМ'ЯТОВУВАЛЬНІ ПРИСТРОЇ

4.1. Класифікація запам'ятовувальних пристроїв

Для зберігання малих масивів кодових слів можуть використовуватися регістри пам'яті. Однак за необхідності зберігання навіть десятка слів застосування регістрів призводить до невиправдано значних апаратних витрат. Тому для запам'ятовування великих масивів даних побудовані спеціальні мікросхеми, які здатні зберігати інформацію обсягом до тисяч біт. Такі мікросхеми називаються *запам'ятовувальними пристроями* (ЗП) і служать для *фіксування (запису), зберігання та зчитування (обміну) інформації*.

Основою будь-якого ЗП є *комірка пам'яті*, що здатна запам'ятовувати один біт інформації, тобто один двійковий розряд. Такою коміркою пам'яті може бути або тригер, або конденсатор з транзисторним ключем.

ЗП, що будуються на базі тригерів, належать до *статичних* ЗП. Записана інформація, тобто стан тригера не руйнується з часом експлуатації, а змінюється лише при наданні зовнішніх керуючих сигналів.

ЗП, що використовують у ролі запам'ятовувачів конденсатори, належать до *динамічних* ЗП. Через неминучі струми витікання конденсатора останній після запису розряджається, а при зчитуванні зарядженого стану частина заряду розтікається по шині зчитування. Тому динамічну пам'ять час від часу треба поновлювати.

ЗП будують, як правило, за матричним принципом. На відміну від ПЛМ, на перетинах шин включають запам'ятовувальні комірки або статичні, або динамічні, тому й ЗП носять назву або статичних, або динамічних.

За функціональною ознакою інтегральні ЗП поділяють на два класи: *оперативні та постійні*.

Оперативні запам'ятовувальні пристрої (ОЗП) призначені для тимчасового зберігання двійкової інформації. ОЗП виконують операції запису й зчитування за наявності напруги живлення. Щодо зберігання інформації, то в одних ОЗП воно здійснюється під дією напруги, а в інших – енергонезалежно.

Постійні запам'ятовувальні пристрої (ПЗП) призначені для тривалого зберігання записаної інформації. Зміст запису в ПЗП не змінюється під час експлуатації і за відсутності напруги живлення не руйнується. ПЗП бувають двох типів: *одноразового запису* та такі, що допускають *поновлення* занесеної інформації *декілька разів*. Цей процес реалізують за допомогою спеціальних пристроїв – програматорів.

За способом звернення ЗП класифікують на *адресні та безадресні*. Матриця ЗП має всього дві шини – запису і зчитування. Оскільки запис або зчитування може здійснюватись тільки для одного запам'ятовувача, то для цього організують адресну вибірку, згідно з якою шукану комірку знаходять за номером стовпчика і рядка, тобто за адресою, яка має вигляд комбінації *n*-розрядного двійкового числа.

Адресні ЗП мають спеціальні адресні входи і можуть бути з довільним або з послідовним звертаннями. У перших пошук інформації здійснюється шляхом

безпосереднього звернення до перетину шин, адрес яких задається двійковим числом. У других – за допомогою послідовної вибірки при збільшенні або при зменшенні адресного числа.

У *безадресних ЗП* звернення виконується незалежно від координат запам'ятовувачів, тобто не за адресою, а за певними ознаками самої інформації, що міститься в запам'ятовувачі ЗП.

За технологією виконання ЗП підрозділяються на біполярні ЗП, де використовується схемотехніка ТТЛ і ЕЗЛ; ЗП МОН – структур; інжекційні ЗП.

4.2. Параметри ЗП

Основні параметри ЗП залежать від технології виготовлення і визначаються так само, як і для інших мікросхем. Це наступні параметри: навантажувальна здатність, споживана потужність, завадостійкість, числова величина логічних рівнів.

До найважливіших параметрів ЗП належать місткість і швидкодія.

Кількість інформації, що може зберігатися у ЗП, визначає його *місткість*. Складаються ЗП з декілька комірок пам'яті N , кожна з яких може зберігати слово з визначеним числом розрядів n . Отже, місткість задається або добутком числа запам'ятовувачів N на розрядність n слів: $M = N \cdot n$, або у розкритій формі M . Наприклад, для першого випадку позначення ЗП у формі 32×8 біт визначає, що ЗП здатний зберігати 32 слова по 8 розрядів, тобто 32 байти або 256 біт. Для другого випадку місткість ЗП позначається лише у вигляді 32 байти або 256 біт, не розкриваючи при цьому кількість слів та кількість розрядів.

Швидкодія ЗП характеризується часом повного циклу звернення і часом вибірки або часом доступу до пам'яті. Період звернення – це мінімально припустимий час між двома черговими зверненнями до ЗП. Цей параметр залежить від характеристики і властивостей ЗП відновлюватися після попереднього звернення до нього.

Час вибірки – це інтервал часу між моментом подачі сигналу вибірки до появи інформації на виході ЗП. Для ВІС ЗП час вибірки наближається до пікосекундного діапазону.

Кожна мікросхема ЗП характеризується *потужністю споживання*, набором *напруг живлення*, типом корпусу. Мікросхеми ПЗП мають додаткові параметри: часову тривалість зберігання інформації, по закінченню якої інформація може самовільно змінитися, та максимальну кількість циклів перезапису, після чого мікросхема визначається нездатною до використання.

4.3. Оперативні запам'ятовувальні пристрої

Оперативні ЗП (ОЗП) – це невіддільні елементи цифрових приладів, які використовуються для тимчасового зберігання інформації і без яких не може бути жодної ЕОМ. У цифрових системах зв'язку на ОЗП зберігають дані, які

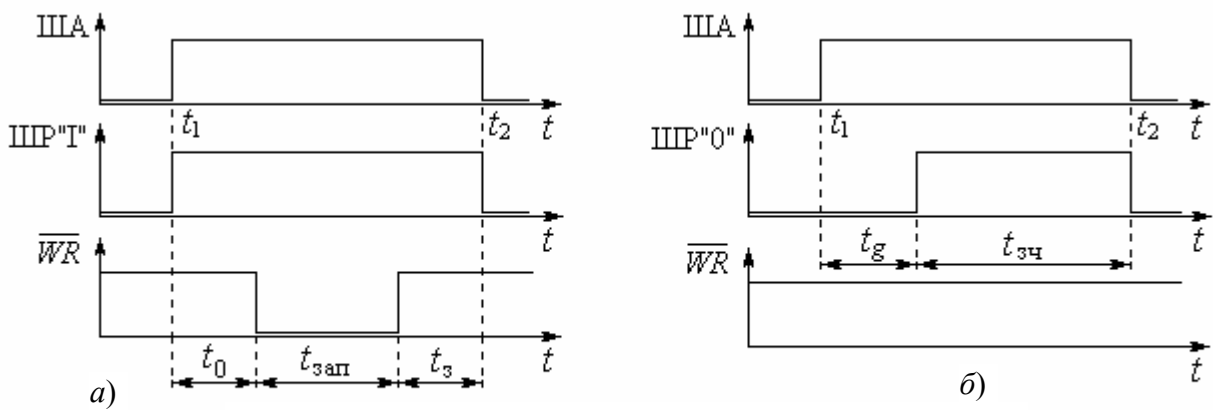


Рисунок 4.2 – Часова діаграма запам'ятовувача: а) в режимі запису; б) в режимі зчитування

Зчитування при $ША = 0$ також неможливе, бо завдяки ключу $D4$ вхід дозволу драйвера $D6$ нульовий ($Z = 0$), через що на виході $D5$ високий опір, тобто розрядна шина вихідних даних ШП "0" від'єднується від прямого виходу Q тригера $D3$.

Так здійснюється режим зберігання.

Режими запису та зчитування досягаються при одиничному стані шини адреси ($ША = 1$). При цьому через ключ $D2$ дається дозвіл на запис, а через інвертор $D4$ – дозвіл на зчитування. Послідовність подачі сигналів у режимах запису та зчитування має бути такою, що показана на рис. 4.2.

Запис досягається в активному стані запам'ятовувача ($ША = 1$) при $\overline{WR} = 0$ протягом $t_{\text{зап}}$. За час t_0 до запису треба виставити дані ШП "Г". Протягом $t_{\text{зап}}$ обидва входи ключа $D2$ одиничні. Тому $C = 1$ і саме тим стан розрядної шини вхідних даних ШП "Г" через вхід тригера D записується в тригер. Запис слід припинити за час затримки t_3 до моменту t_2 .

Зчитування досягається в активному стані запам'ятовувача ($ША = 1$) при $\overline{WR} = 1$. Цей рівень треба встановити за час t_g до зчитування. При цьому ключ $D2$ запирається, припиняючи запис, а обидва входи ключа $D5$ нульові. Тому вхід дозволу $Z = 0$ драйвера $D6$ здійснює з'єднання прямого виходу Q тригера $D3$ з розрядною шиною вихідних даних ШП "0".

4.3.2. Структура ВІС ОЗП

У структурі ВІС ОЗП запам'ятовувачі компонуються у прямокутну матрицю, яку називають *накопичувачем ЗП*. Кожний рядок накопичувача – це адресна шина, а стовпець – дві розрядні шини: шина запису ШП "Г" та шина зчитування ШП "0".

У кожному перетині матриці ставиться запам'ятовувач, який може зберігати один біт інформації. Запам'ятовувачі, що згруповані по горизонталях, здатні запам'ятовувати n біт (розрядів) одного слова, бо з'єднані однією адресою. Кількість слів, яку може запам'ятовувати матриця, визначається числом горизонтальних рядків.

Кожне слово вибирається шляхом ініціації відповідної адресної шини.

З метою мінімізації кількості адресних входів ОЗП інформація щодо адреси надходить у вигляді двійкового числа і подається на дешифратор адреси, який входить до складу ОЗП. На виході дешифратора адреси ініціюється тільки одна шина, яка вибирає для запису або зчитування лише один рядок, тобто одне слово.

Матриця N слів, кожне з яких має розрядність n , визначається основним вузлом ЗП, який називають *накопичувачем інформації*. Залежно від способу звернення (\overline{WR}/RD) до запам'ятовувачів розрізняють два найбільш поширених типи організації накопичувачів:

- з однокоординатною або послівною вибіркою;
- з двокоординатною вибіркою.

Накопичувач з однокоординатною або послівною вибіркою характеризується тим, що має лише одну координату звернення до запам'ятовувачів, а саме – номер *рядка* накопичувача і тому має лише один дешифратор адреси.

Накопичувач з двокоординатною вибіркою має дві адресні шини: горизонтальні по рядках (ШАХ) і вертикальні по стовпцях (ШАУ).

Структурна схема накопичувача з однокоординатною вибіркою показана на рис. 4.3.

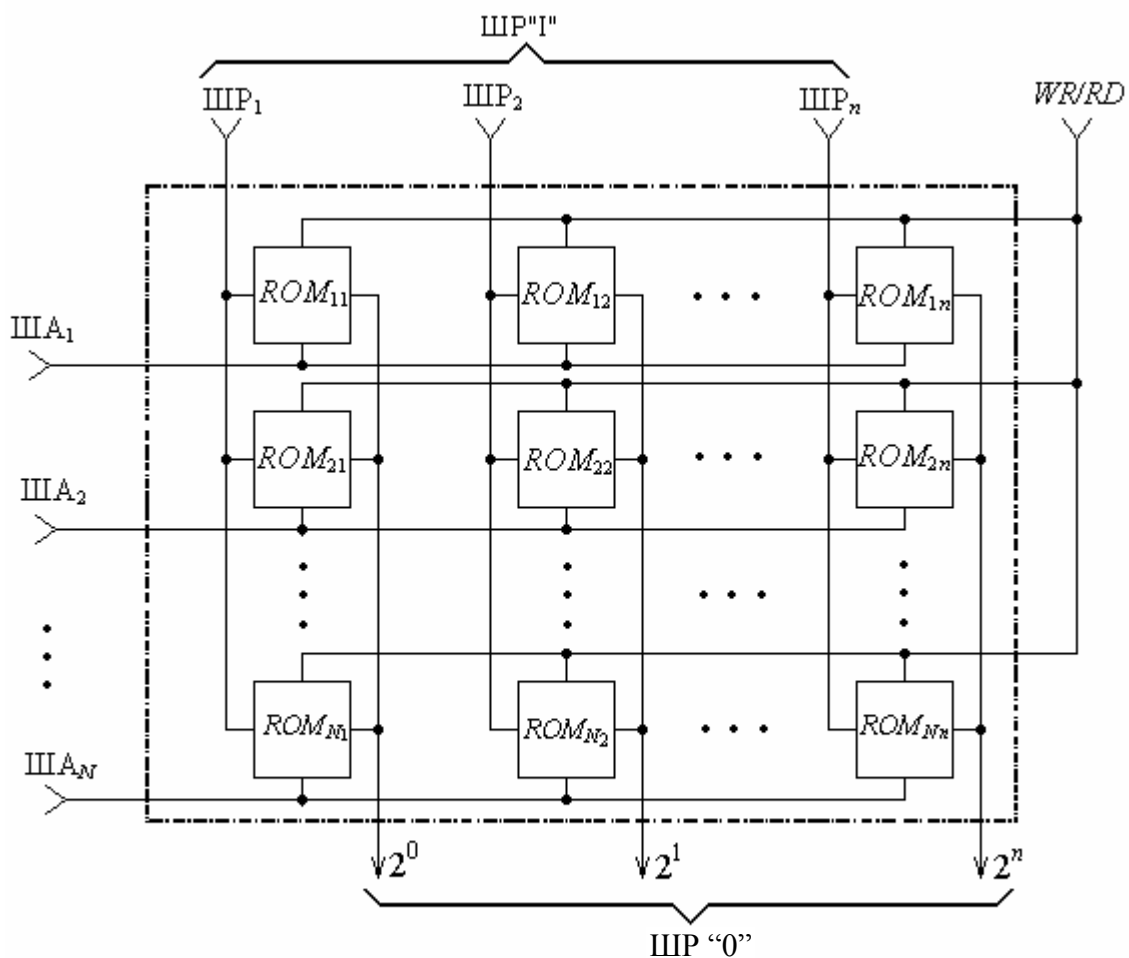


Рисунок 4.3 – Схема ОЗП з однокоординатною вибіркою

ОЗП має число $N \cdot n$ запам'ятовувальних елементів ROM_{Nn} , які розміщені по N рядках та n стовпцях. Кожний рядок утворює слово з номером

N . При однокоординатній вибірці пошук кожного слова з номером i здійснюється відповідною однією шиною адреси $ША_i$, бо ОЗП має шини адреси $ША_1, ША_2, \dots, ША_i, \dots, ША_N$, що з'єднані відповідно з кожним запам'ятовувачем ROM_{ij} однойменного i -го слова.

Розрядні вхідні шини $ШР_j$ з'єднують входи запису кожного розряду усіх N слів.

Вихідними є шини $2^0, 2^1, \dots, 2^n$, які з'єднують виходи зчитування кожного розряду всіх N слів.

Для усіх $N \cdot n$ запам'ятовувачів ROM_{ij} існує загальна шина звернення \overline{WR}/RD , сигнал на якій визначає режим запису або зчитування.

Схема працює наступним чином.

Сигнал вибірки адреси в кожний момент часу може з'явитися лише на одній з адресних шин $ША_i$.

Запис або зчитування відбувається за наявності на вході \overline{WR}/RD необхідного рівня (1 або 0). Зчитування RD (від англ. *Read* – читання) відбувається при надходженні сигналу одиничного рівня, а запис WR (від англ. *Writ* – писати), коли на шині встановлюється нульовий рівень.

Для запису слова в комірки запам'ятовувачів ROM_{ij} слід активізувати i -й рядок накопичувача, тобто адресну шину $ША_i$, а на шину \overline{WR}/RD подати сигнал дозволу на запис інформації (лог. "0"), через що всі шини $ШР_1, ШР_2, \dots, ШР_n$ будуть підключені до входів запису i -го рядка.

Для зчитування слова з комірок запам'ятовувачів ROM_{ij} слід активізувати i -й рядок накопичувача, тобто адресну шину $ША_i$, а на шину \overline{WR}/RD подати сигнал дозволу на зчитування інформації (лог. "1"), через що на всіх шинах $2^0, 2^1, \dots, 2^{n-1}$ встановляться відповідні значення логічного "0" або логічної "1" зчитаного N -го слова.

Розглянутий тип накопичувача інформації має лише одну координату звернення до запам'ятовувачів, а саме – номер рядка накопичувача і тому має лише один дешифратор адреси. Такий накопичувач з однокоординатною вибіркою називають ще двовимірним (типу $2D$), бо запам'ятовувачі в ньому розміщені на площині.

Типова структурна схема ВІС ОЗП з однокоординатною вибіркою показана на рис. 4.4.

До складу ВІС ОЗП, крім основного вузла – накопичувача, входять регістр адреси RGA , дешифратор адресних шин DC , схема керування, пристрій запису даних DI і пристрій зчитування даних DO .

Інформаційні сигнали, які треба записувати, надходять по шині DI . Зчитування інформації відбувається за допомогою пристроїв, які комутуються до шин DO . Регістр адреси RGA може бути відсутнім. Тоді адресний вхід мікросхеми ВІС ОЗП безпосередньо зв'язаний з входом дешифратора DC .

Накопичувач з двокоординатною вибіркою має розрядну шину $ШР$ і *дві адресні шини*: горизонтальні (по рядках) $ШАX$ і вертикальні (по стовпцях) $ШАY$. Накопичувач з такою структурою має тривимірну будову і складається з n накопичувачів розрядів, кожний з яких містить N запам'ятовувачів.

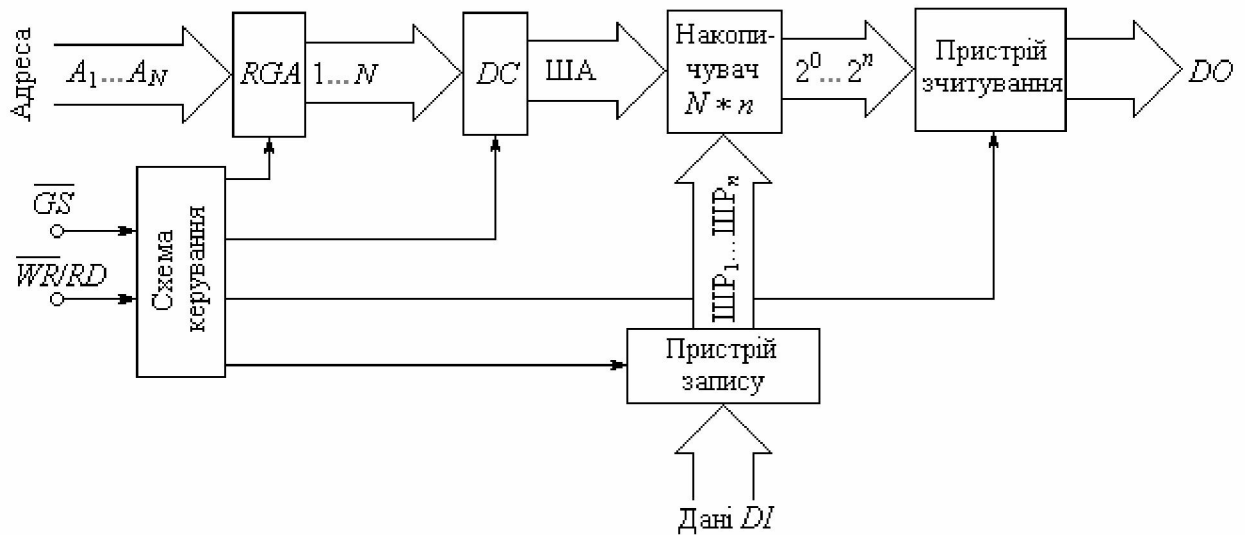


Рисунок 4.4 – Структурна схема ВІС ОЗП з однокоординатною вибіркою ,

Отже, кожний накопичувач відповідає тільки за один однойменний розряд усіх N слів, а кожне слово записується, зберігається і зчитується в усіх n накопичувачах за ідентичною двокоординатною адресою x_i, y_j .

Структурна схема ВІС ОЗП з двокоординатною вибіркою показана на рис. 4.5.

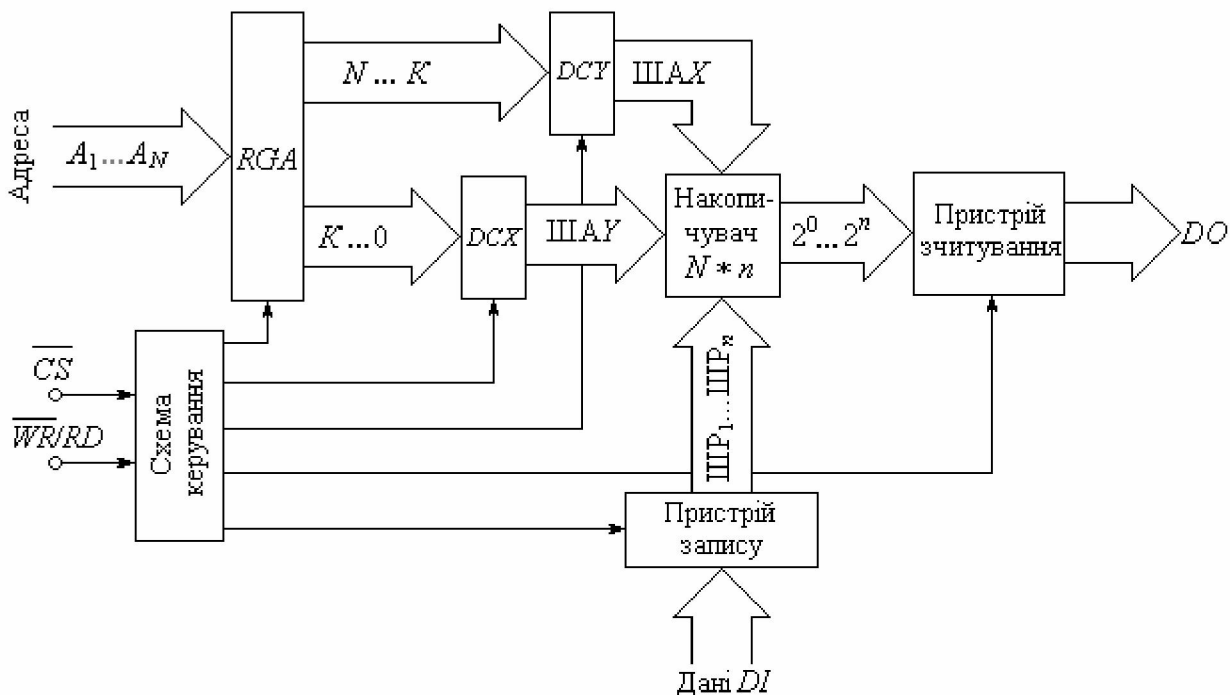


Рисунок 4.5 – Структурна схема ВІС ОЗП з двокоординатною вибіркою

Обидві структурні схеми відрізняються лише наявністю одного (рис. 4.4) дешифратора адреси, або двох (рис. 4.5), що носять назви: DCX – дешифратор адресних шин строк ШАХ; DCY – дешифратор адресних шин стовпців ШАУ.

Для запису слова у запам'ятовувач за адресою x_i, y_j на вході адресних дешифраторів ШАУ і ШАХ (рис. 4.5) подаються відповідні адреси.

Інформаційне двійкове слово надходить на спільні розрядні шини ШР через пристрій запису, а зчитування записаного слова здійснюється через пристрій зчитування.

Структурна схема (рис. 4.5) має два керуючих входи: запис/зчитування \overline{WR}/RD та додатковий – \overline{CS} .

Керуючий вхід \overline{CS} , що носить назву “вибір мікросхеми” – це вхід дозволу проходження сигналу.

Вхідні схеми ВІС ОЗП – це формувачі на логічних елементах, що забезпечують спряження накопичувача з вхідними зовнішніми пристроями за струмом та напругою.

Вихідні каскади ВІС ОЗП – це логічні елементи з відкритим колектором або тристанові буфери, які забезпечують каскадування ВІС у системах з шинною організацією передавання даних.

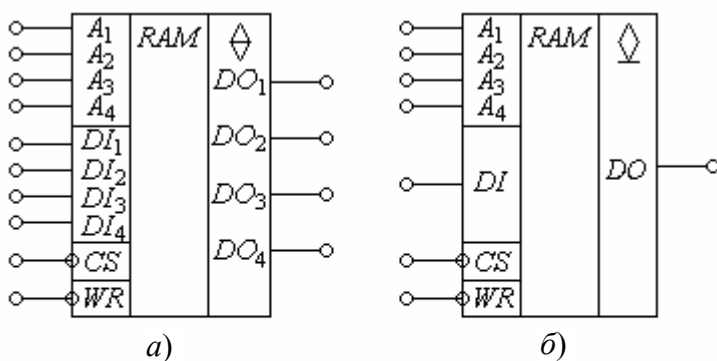


Рисунок 4.6 – Умовне позначення ВІС ОЗП:
 а) з паралельними входами і виходами;
 б) з послідовними входом і виходом

На рис. 4.6 показані умовні позначення статичних ВІС ОЗП з паралельними входами й виходами (рис. 4.6,а) та послідовним входом і виходом (рис. 4.6,б).

Мікросхема статичного чотирирозрядного ОЗП (рис. 4.6,а) має паралельні входи DI_j і виходи DO_j з тристановим буфером і з організацією пам’яті місткістю (16×4) біт,

тобто 64 біт.

Мікросхема статичного чотирирозрядного ОЗП (рис. 4.6,б) має послідовні вхід DI і вихід DO даних з відкритим колектором.

4.3.3. Інформаційні та керуючі сигнали ВІС ОЗП

Обмін інформацією з ОЗП здійснюється по шинах або магістралях, які поділяються на три групи: шина даних, адресна шина, шина керуючих сигналів.

Шина даних – це набір сигнальних ліній, за допомогою яких передається паралельно двійкова інформація. Число ліній, що утворюють шину, визначають максимальну довжину числа, яке можна передавати в двійковій формі. Шини даних бувають 8-, 16- і 32-розрядні.

Символами DI позначається шина запису даних, а DO – шина зчитування даних.

Шина даних може бути *двонаправленою*, тобто для взаємного обміну даними різні пристрої підключаються до єдиного проводу – шини, як показано на рис. 4.7. По цій шині відбувається обмін інформацією між *робочими блоками* пристроїв. Робочий блок – це будь-який вузол, до якого надходять дані $EI_1 \dots \dots EI_N$, наприклад, ОЗП.

Виходи робочих блоків через елементи $\&$ з відкритим колектором підключені до числової шини D . До інших входів елементів $\&$ надходять керуючі сигнали (сигнали звернення) $EO_1 \dots EO_N$.

При $EO_i = 0$ вихідний транзистор елемента $\&$ з відкритим колектором закритий і вихід блока від'єднується від шини. Якщо на керуючому вході EO_i присутня логічна "1", то вихідний сигнал відповідного i -го робочого блока пропускається на шину D .

У кожний відрізок часу одиничне значення сигналу $EO_i = 1$ має подаватися лише на один пристрій. Якщо одиничні рівні $EO_i = 1$ на керуючих входах змінювати по чергово, то на шині D вихідні сигнали пристроїв будуть відбиватися також по чергово.

Керуючі сигнали $EI_1 \dots EI_N$ визначають режими приймання чи передавання. Завдяки цьому до шини D можна підключати не тільки виходи, а й входи робочих блоків з метою утворення *двонаправленого режиму* роботи.

Якщо на деякий час подати $EO_i = 0$, а $EI_i = 1$, то робочий блок буде приймати сигнали з шини. Якщо на один з пристроїв подавати сигнал $EO_i = 1$, а на інший $EI_j = 1$, то можна сигнал від i -го пристрою передавати на j -й пристрій, тобто виконувати за необхідності обмін даними між двома робочими блоками.

Одержану числову магістраль D на базі елементів з відкритим колектором називають *двонаправленою шиною*. Таке з'єднання елементів дозволяє зменшити кількість виводів пристроїв і кількість провідників магістралі.

Коли усі пристрої відключені від шини, тобто шина знаходиться у стані інформаційного нуля, то на ній утримується високий рівень напруги, завдяки джерелу $+E$ та резистору R_n . Тоді обмін по шині з відкритим колектором здійснюється за правилами негативної логіки, коли активний рівень низький. Це забезпечує меншу споживану потужність у проміжках між операціями обміну.

Суттєвим недоліком таких шин є мала швидкодія та низька завадостійкість. Це пояснюється тим, що шини передавання інформації, як правило, довгі, мають значні паразитні ємності і тому, як наслідок, вони чутливі до завад. Крім того, активний вихід (навіть високоомний) незручний і досить шкідливий в тих випадках, коли потрібно вести обмін даними одночасно з кількома робочими блоками або вузлами, як це має місце у мікропроцесорній техніці.

Сумісну роботу декількох блоків на одній лінії інформаційної шини успішно забезпечує логічний елемент з трьома вихідними станами або тристанний драйвер, який був розроблений спеціально для використання в ролі

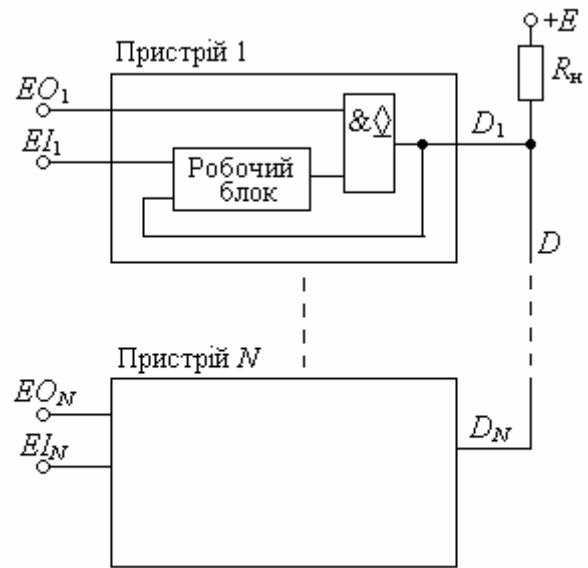


Рисунок 4.7 – Організація шини даних

вихідного буфера для підключення цифрових блоків до магістралі або шини. Буфери з трьома станами називають *шинними драйверами*.

Адресна шина – це група сигнальних ліній, за допомогою яких визначається прилад, в який чи з якого повинна бути передана або зчитана інформація. Кожне двійкове число, що подане на адресну шину, може визначити лише конкретну область пам'яті. Восьмирозрядні процесори мають 16 адресних ліній; 16-розрядні мають, як правило, 20-розрядні адресні шини. Адресні шини, звичайно, однонаправлені і позначаються символами A_i .

Шина керуючих сигналів – це набір сигнальних ліній. Склад таких сигналів може бути досить різноманітним у залежності від типу запам'ятовувача.

Нормальне функціонування ОЗП залежить від організації часових співвідношень між сигналами. Тому всі згадані сигнали мають бути розподілені за часом.

Розглянемо процеси запису й зчитування інформації в мікросхемі (рис. 4.6,*а*). Ці процеси ілюстровані часовими діаграмами на рис. 4.8,*а* та 4.8,*б*.

Мікросхема має наступні шини керуючих сигналів: \overline{CS} – вибір мікросхеми; \overline{WR}/RD – керування режимами запису та зчитування.

Послідовність подачі сигналів у режимах запису та зчитування показана на рис. 4.8, що обов'язково треба виконувати для правильної роботи мікросхеми.

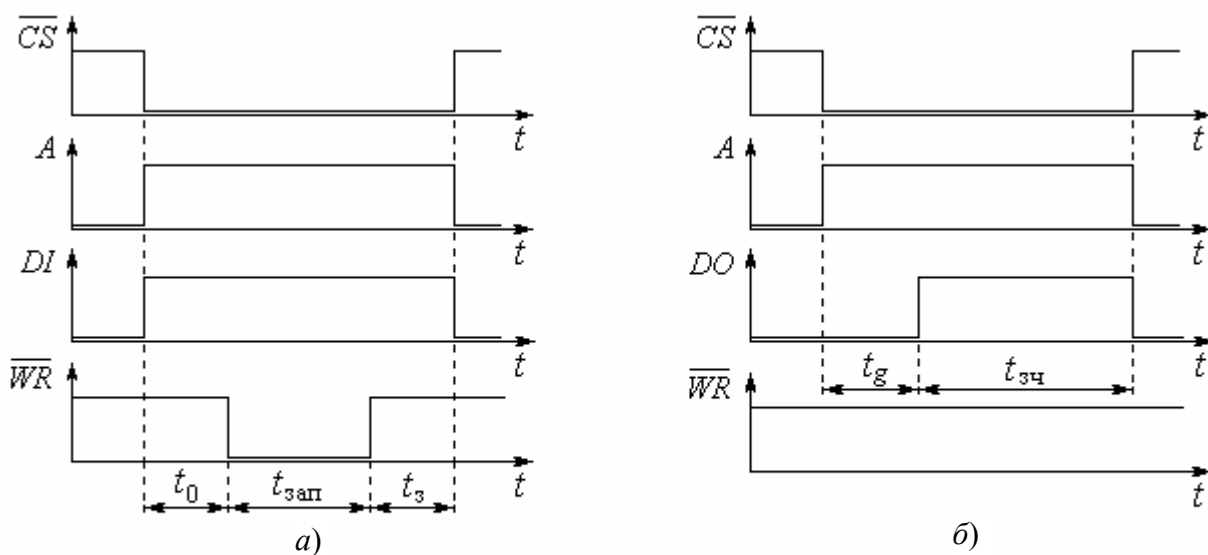


Рисунок 4.8 – Часові діаграми роботи ВІС ОЗП:
а) у режимі запису; *б*) у режимі зчитування

У режимах запису й зчитування тривалість імпульсу \overline{CS} визначає час звернення до даної мікросхеми, за який потрібно закінчити процеси звернення.

У режимі запису інформації рис. 4.8,*а* дозвіл на запис $\overline{WR} = 0$ подається (за часом) після установки адреси. У нашому випадку адреса задається сигналами одиничного рівня, тобто $A = 1$. Часова затримка приходу сигналу \overline{WR} за порівнянням з $A = 1$ на тривалість t_0 запобігає помилок, які пов'язані із заборонаю починати запис інформації до закінчення часу дешифрування

адреси. Для забезпечення надійного запису у вибрану комірку ОЗП сигнал дозволу на запис \overline{WR} знімається раніше за всі інші сигнали DI , A і \overline{CS} . Отже останні мають припинити свою дію лише з певною затримкою t_3 . Сума часових інтервалів називається *часом циклу запису*

$$T_3 = t_0 + t_{\text{зап}} + t_3. \quad (4.1)$$

У режимі зчитування інформації рис. 8.8,б після установалення активного рівня адреси $A = 1$ дані на виході DO з'являються тільки після певного проміжку часу t_g , який називають *часом доступу* при зчитуванні. Сума часів називається *часом циклу зчитування*

$$T_{\text{зч}} = t_g + t_{\text{зч}}. \quad (4.2)$$

ВІС ОЗП допускають нарощування місткості пам'яті збільшенням розрядності та числа запам'ятовувачів.

4.4. Постійні запам'ятовувальні пристрої (ПЗП)

4.4.1. Класифікація ПЗП

Постійні запам'ятовувальні пристрої (ПЗП) – це функціональні вузли, які мають лише два режими роботи: зберігання та зчитування. Запис нових даних у ПЗП можна здійснювати тільки до включення ПЗП в роботу. Цей процес носить назву “Програмування ПЗП” і здійснюється як заводом-виготовувачем, так і користувачем-програмувачем.

Існує три типи ПЗП, що підрозділяються за способом програмування.

ПЗП з *масковим програмуванням* (МПЗП) – це пристрої, в які інформація записана раз і назавжди.

Зміст пам'яті такого МПЗП залишається постійним на весь час його роботи. Маскові ПЗП виготовляються тільки на заводі, де за картою програмування, яка по суті є таблицею істинності, устанавлюються ділянки металізації, які потрібні для кодування тої чи іншої інформації.

За такою технологією виготовляються ПЗП для перетворення, наприклад, двійкового коду в коди символів (російських, латинських літер, цифр тощо).

Програмовані ПЗП (ППЗП) на відміну від МПЗП мають напівпровідникові діоди або транзистори, які з'єднані з усіма точками перетину шин матриці плавкими перемичками, тобто при виготовленні за всіма адресами ППЗП записується число $\{11 \dots 11\}$.

Режим програмування складається з послідовної подачі адреси слів, після чого імпульсами струму руйнуються перемички в місцях, де вони непотрібні.

Для перепалювання перемичок використовується спеціально призначений для цього пристрій – програматор.

Якщо при програмуванні була допущена помилка, то вона не може виправитися. При цьому ППЗП буде неприцездатним.

Перепрограмовані або *репрограмовані* ПЗП (РПЗП) дозволяють виконувати запис та стирання інформації. Організація РПЗП відрізняється від ППЗП тим, що між лініями рядків і стовпців включені не діоди чи біполярні транзистори з плавкими перемичками, а спеціальні МОН-транзистори з так званим плаваючим заслоном.

Оскільки ПЗП існують тільки в інтегральному виконанні, то вони відомі під назвою *великих інтегральних схем* (ВІС).

ВІС (РПЗП) схожі на ОЗП, в яких цикл запису в декілька тисяч разів більше за цикл зчитування. В РПЗП можна повернути в початковий стан будь-який окремих МОН-транзистор.

Недоліком РПЗП та МПЗП порівняно з ППЗП є більший час вибірки: сотні секунд.

За способом зчитування ВІС ПЗП поділяють на *асинхронні* та *синхронні*.

Зчитування інформації з асинхронних ПЗП відбувається в будь-який час при зверненні до даного ПЗП, а із синхронних ПЗП – лише за наявності на спеціальному вході ВІС ПЗП синхроімпульсу заданої тривалості.

За технологією виготовлення ВІС ПЗП розрізняють за типом запам'ятовувачів: діодні, біполярні та польові.

4.4.2. Структура ВІС ПЗП

ПЗП можна розглядати як дворівневу ПЛМ (див. рис. 5.13). Програмується лише матриця М2 (диз'юнкцій), а матриця М1 (кон'юнкцій) налагоджена на реалізацію функцій повного дешифратора усіх $q = 2^n$ вихідних від n вхідних адресних кодових комбінацій. Реалізація ПЗП доводить, що схему повного дешифратора (матрицю М1) програмувати неможливо, через що параметр $q = 2^n$ зафіксований.

На рис. 4.9 показана узагальнена структурна схема ВІС ПЗП.

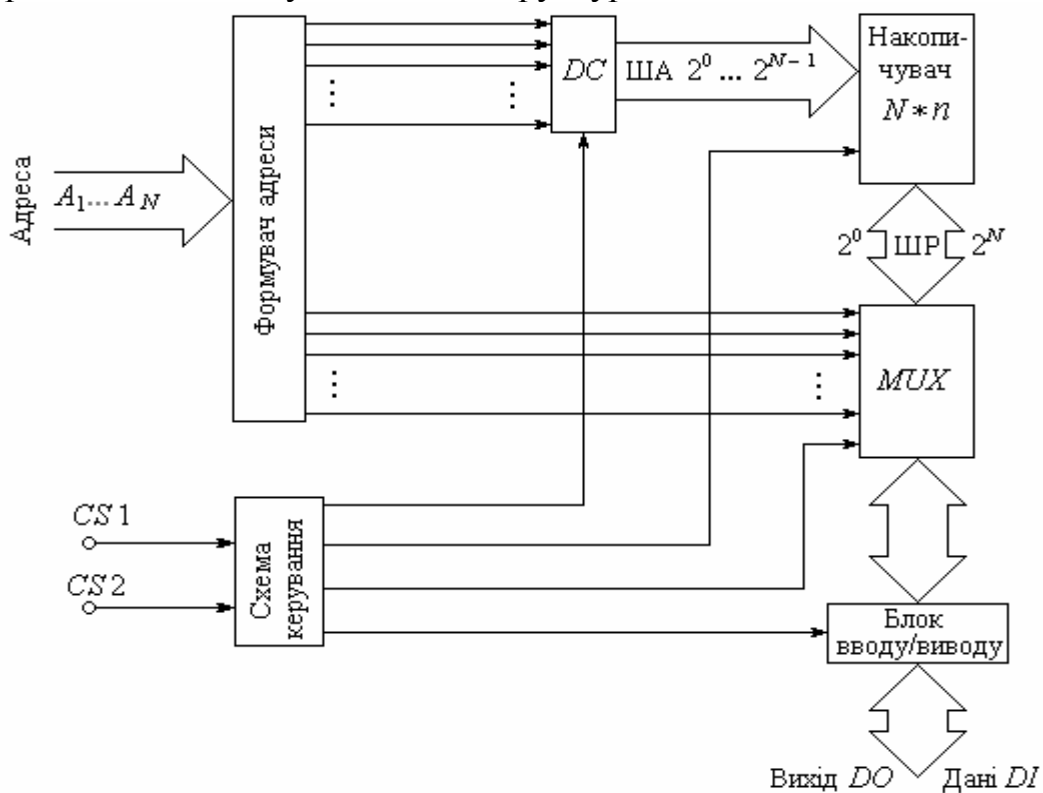


Рисунок 4.9 – Узагальнена структурна схема ВІС ПЗП

Основою структурної схеми є *накопичувач інформації* (надалі – накопичувач), який з'єднаний через ША з дешифратором адреси DC, а через

ШР – з блоком мультиплексорів *MUX*. Паралельний код адреси *A* подається на формувач адрес, протифазні сигнали якого надходять на *DC* і блок *MUX*, які активізують одну з горизонтальних (адресних) шин ША. Зчитування записаної у запам'ятовувачах інформації відбувається по всіх вертикальних шинах ШР через блок мультиплексорів і пристрій вводу-виводу.

Схема керування синхронізує роботу *DC* і блока *MUX*. Пристрій вводу-виводу виконується або за схемами з відкритим колектором, або з використанням тристанових драйверів.

Схема керування визначає режими запису, зчитування та стирання інформації. На відміну від ОЗП при зчитуванні накопичувача ПЗП видається вміст цілого рядка запам'ятовувачів, причому такий рядок може містити навіть декілька слів.

З вибраного дешифратором рядка за допомогою мультиплексорів *MUX* виділяється і передається на вихід потрібне слово.

Щодо програмування, то воно починається зі складання таблиці істинності, де кожній комбінації повного набору адреси A_1, \dots, A_n відповідає кодова комбінація даних адреси $D_0, \dots, D_{(m-1)}$.

Для наглядного прикладу задамося таблицею істинності, що має вигляд табл. 4.1.

Таблиця 4.1 – Стани запам'ятовувача ПЗП

Слово ША	Дані ПЗП							
	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8
A_1	0	1	1	0	0	1	1	1
A_2	1	0	1	0	0	0	1	0
A_3	1	1	0	0	1	0	0	0
A_4	0	1	1	1	0	0	0	0

Таблиця істинності описує стани матриці накопичувача для побудови ПЗП, яка має місткість 32 біти, що розбиті на чотири слова ($A_1 \dots A_4$) по вісім розрядів ($D_8 \dots D_1$) у кожному з них.

Отже, такий ПЗП здатний запам'ятовувати чотири однобайтових слова ($4 \times 8 = 32$ біти).

Діодна матриця накопичувача ПЗП, яка відповідає табл. 4.1, показана на рис. 4.10.

Для вибору потрібного слова на одну з чотирьох адресних шин подається одиничний рівень напруги. Тоді на тих шинах даних ШР, на перетинах яких з обраною шиною адреси ША присутній діод, з'являється одиничний рівень, а якщо діод відсутній – нульовий. Наприклад, при активізації шини адреси A_1 кодова комбінація даних на шині ШР є словом: 01100111 (якщо вважати D_1 молодшим розрядом).

Основним недоліком ВІС ПЗП з діодним накопичувачем є відносно велике струмоспоживання по входах $A_1 \dots A_4$, що навантажує джерело сигналу.

Цей недолік усунений в ПЗП з *транзисторним* накопичувачем.

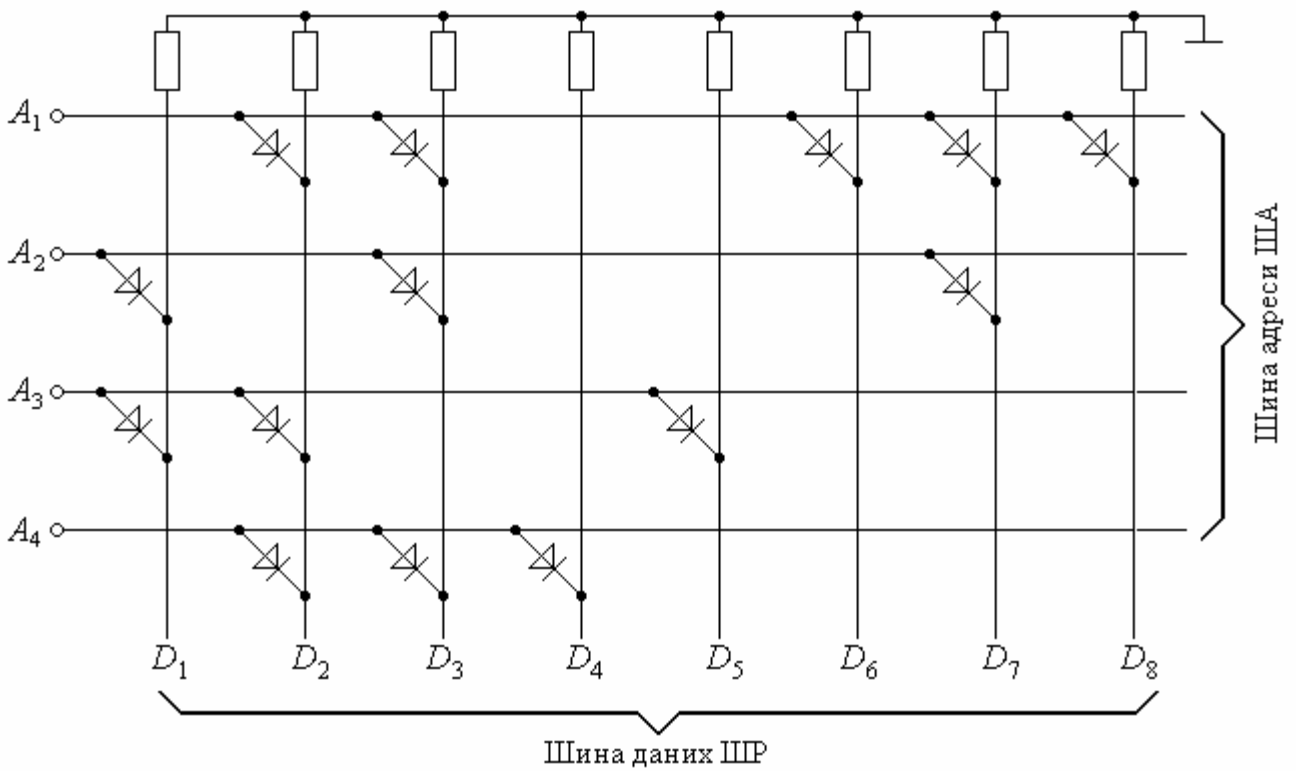


Рисунок 4.10 – Діодна матриця накопичувача ПЗП

Схема біполярної матриці накопичувача, що відповідає табл. 4.1, показана на рис. 4.11.

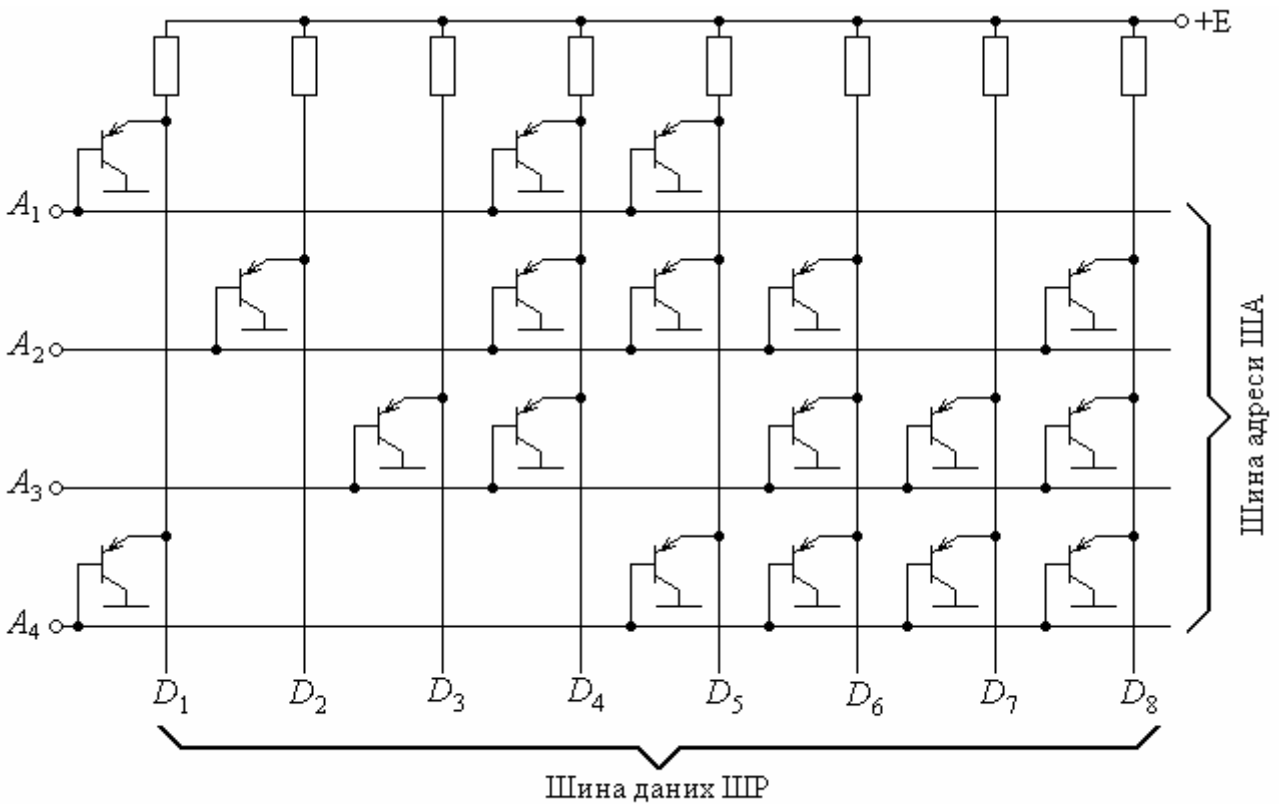


Рисунок 4.11 – Біполярна матриця накопичувача ПЗП

Активним для матриці адресним потенціалом є рівень логічного “0”. В координатному полі накопичувача транзистори розміщують у точках перетину, де повинні зберігатися біти, що мають значення логічного “0”.

При подачі на одну з адресних шин рівня логічного “0”, відкриваються транзистори, які підключені до цієї адресної шини. Тоді на тих шинах даних ШР, на перетинах яких з обраною ША присутній транзистор, формується нуль, а на решті ШР через резистори надходить рівень логічної “1” і від джерела $+E$.

Щодо МОН-накопичувачів, то вони мають нижчу швидкодію, ніж біполярні, але потужність розсіювання їх значно нижча. З метою реалізації польової матриці-накопичувача на перетинах адресних шин і шин даних за відповідною таблицею істинності ПЗП включаються польові транзистори. Так само, як і у біполярного накопичувача, наявність або відсутність транзистора у точці ортогональних ліній відповідає стану логічної “1”, або логічного “0” запам’ятовувача.

Структура програмованих ПЗП (ППЗП) аналогічна структурі маскових ПЗП. Різниця, по-перше, у схемі запам’ятовувача, а по-друге, в кількості напівпровідникових елементів, які перепалюються при програмуванні.

Запам’ятовувачі таких ППЗП бувають різні. Найчастіше зустрічаються запам’ятовувачі, що виготовляються з плавкими перемичками (рис. 4.12,а), або зі зворотно включеними діодами (рис. 4.12,б).

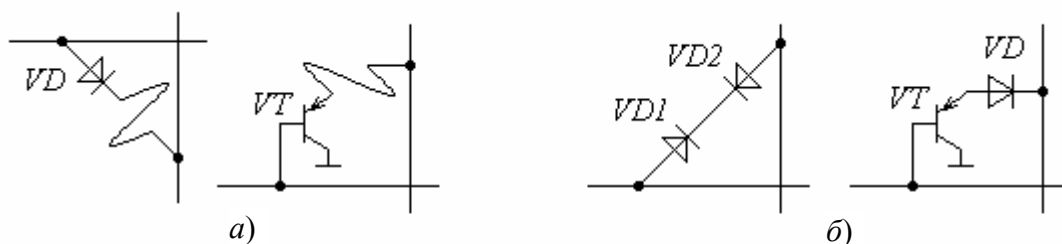


Рисунок 4.12 – Типи запам’ятовувачів програмованих ПЗП: а) з плавкими перемичками; б) зі зворотно включеними діодами

У процесі програмування перемички перепалюються електричним струмом. Якщо перемичка перепалюється, запам’ятовувач від’єднується від точки перетину.

Другий тип запам’ятовувача, що використовується у програмованих ПЗП, має зворотно включені діоди VD (рис. 4.12,б). При такій схемі у початковому стані до програмування ПЗП жодна точка перетину не має запам’ятовувачів.

У процесі програмування зворотно включені діоди $VD1$ (VD) закорочуються, тобто залишається або один діод $VD2$, або транзистор VT , що відповідає присутності запам’ятовувача у точці перетину.

Програмування ПЗП із запам’ятовувачами першого типу виконується таким чином, щоб перепалити перемички в тих точках перетину шин, де за таблицею істинності запам’ятовувач повинен бути відсутнім.

Програмування ПЗП із запам’ятовувачами другого типу, навпаки, зводиться до закорочування зворотних діодів у точках, де треба поставити запам’ятовувач.

Режими програмування в різних мікросхемах програмованих ПЗП різні, бо визначаються не тільки технологією виготовлення даної ВІС, але й типом мікросхеми ПЗП.

Структура репрограмованих РПЗП аналогічна масковим та програмованим ПЗП, але відрізняється тим, що в них можна багаторазово стирати записану інформацію та записувати нову. Накопичувачі репрограмованих РПЗП виконуються за спеціальними технологіями з використанням спеціальних типів транзисторних структур.

Наприклад, на основі МОН-структур будують так звані транзистори з плаваючим ізольованим заслоном. Ці транзистори змінюють під час програмування свої характеристики. Такі зміни є ознакою стану інформації, що зберігається в ПЗП. У процесі стирання всі транзистори під дією, наприклад, ультрафіолетового випромінювання закриваються, що відповідає запису в усіх запам'ятовувачах логічної "1". Потім за допомогою шин адреси і даних вибираються ті транзистори, в які потрібно занести логічний "0", і відбувається запис нової інформації.

На рис. 4.13 показані умовні позначення трьох типів ПЗП: маскові МПЗП (рис. 4.13,а), програмовані ППЗП (рис. 4.13,б) та репрограмовані РПЗП (рис. 4.13,в).

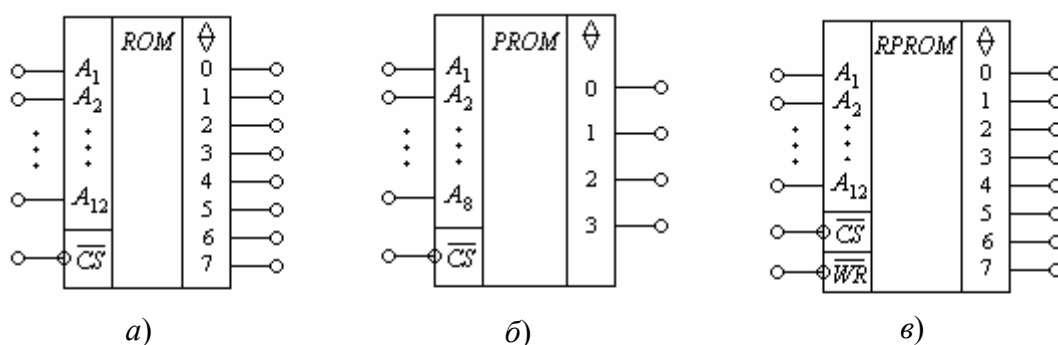


Рисунок 4.13 – Умовне позначення ПЗП:

а) маскові (МПЗП); б) програмовані (ППЗП); в) ре програмовані (РПЗП)

Маскові ПЗП більш придатні для зберігання констант, стандартних таблиць, символів, підпрограм і нагадують "книжку для читання".

Програмовані та репрограмовані ПЗП дозволяють ще реалізувати широкий клас функціональних вузлів і нагадують "блокнот з чистими сторінками". ПЗП можуть бути використані для реалізації булевих функцій, побудови цифрових автоматів, перетворювачів кодів, арифметичних пристроїв та пристроїв оброблення інформації.

У ряді випадків застосування ВІС ПЗП можна отримати значний вигреш у швидкодії, габаритних розмірах та вартості проєктованих пристроїв.

Контрольні питання

- 4.1. Що таке ОЗП?
- 4.2. Що таке запам'ятовувач?

- 4.3. Які вузли входять до типової структурної схеми ОЗП і які функції виконує кожен з них?
- 4.4. Як відбуваються запис та зчитування у накопичувач ОЗП?
- 4.5. Які часові співвідношення між вхідними сигналами ВІС ОЗП?
- 4.6. Що таке ПЗП? Які функції вони виконують?
- 4.7. Яка існує класифікація ПЗП за способом програмування?
- 4.8. Які принципи побудови ВІС ПЗП?
- 4.9. Які типи запам'ятовувачів використовуються в ВІС ПЗП?
- 4.10. Яка різниця між МЗП, ППЗП та РПЗП?

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова О.М., В.Д. Іванченко. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. – С. 397 – 317.
2. Рицар Б.Є. Цифрова техніка / Рицар Б.Є. – К.: НМК ВО, 1990. – 371 с.
3. Большие интегральные схемы запоминающих устройств: Справочник; под ред. Ю.А. Гордонова, Ю.Н. Дьякова. – М.: Радио и связь, 1990. – 420 с.
4. Букреев И.Н. Микро–электронные схемы цифровых устройств / Букреев И.Н., Горячев В.И., Мансуров Б.М.. – М.: Радио и связь, 1990. – 414 с.
5. Зубчук В.И. Справочник по цифровой технике / Зубчук В.И., Сигорский В.П., Шкуро А.Н. – К.: Техника, 1990. – 145 с.

Навчальне видання

Воробйова Олена Михайлівна
Савицька Маргарита Павлівна
Флейта Юрій Вікторович

ЦИФРОВІ ПРИСТРОЇ

Навчальний посібник

Частина 2

Редактор Кодрул Л.А.

Верстання Гардиман Ж.А.