

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОКОРИСТУВАННЯ УКРАЇНИ**

Кулініч О.М., Касаткін Д.Ю., Шкарупило В.В

НАВЧАЛЬНИЙ ПОСІБНИК

**КОМП'ЮТЕРНА СХЕМОТЕХНІКА І АРХІТЕКТУРА
КОМП'ЮТЕРІВ**

(Частина 1. КОМП'ЮТЕРНА СХЕМОТЕХНІКА)

Київ – 2024

УДК 389 Д 439

Рецензенти:

Голуша С.В. доктор технічних наук, професор, Київського національного університету ім. Т. Г. Шевченко

Лакно В.А. доктор технічних наук, професор, Національного університету біоресурсів і природокористування

Складанний П.М. кандидат технічних наук, доцент, Київського столичного університету імені Бориса Грінченка

Рекомендовано Вченою радою НУБіП в якості навчального посібника для студентів за спеціальностями 122, 123, 125 .

Кулініч О.М., Касаткін Д.Ю., Шкарупило В.В. Комп'ютерна схемотехніка і архітектура комп'ютерів (частина 1. Комп'ютерна схемотехніка) Навчальний посібник. – К.: НУБіП, 2024 – 320 с.

У навчальному посібнику розглянуто аналіз та синтез цифрових елементів і пристроїв, які є основою сучасної елементної бази електронної апаратури різного призначення інфокомунікаційних систем, обчислювальної та вимірювальної техніки, систем керування та автоматики, мікропроцесорної техніки.

Викладено основи математичного, логічного проектування та схемотехніку цифрових елементів і пристроїв: системи числення, формати та коди чисел, алгебра логіки, синтез цифрових автоматів, принцип роботи базових логічних елементів, цифрових вузлів комбінаційного та накопичувального типу, позначення, параметри та характеристики, реалізація на їх основі сучасних інтегральних мікросхем.

Навчальний посібник призначений для використання в навчальному процесі Національного університету біоресурсів і природокористування а також може застосовуватися в інших вищих та середніх спеціальних навчальних закладах за фахом кібербезпека, комп'ютерні науки та ін.

ISBN 966-7714-82-9

УДК 389 Д439
НУБіП, 2024

ЗМІСТ

СПИСОК СКОРОЧЕНЬ	6
ВСТУП	7
РОЗДІЛ 1. МАТЕМАТИКО-ЛОГІЧНІ ОСНОВИ ПОБУДОВИ ЦИФРОВИХ ПРИСТРОЇВ	9
1.1 Системи числення та кодування чисел	10
1.1.1. Системи числення.....	10
1.1.2 Правила переведення чисел у задану систему числення	14
1.1.3. Формати цілих і дійсних чисел.....	19
1.1.4. Кодування номерів.....	23
1.1.5. Арифметичні операції над кодами.....	25
1.2 Основи логічної алгебри та логічних функцій	27
1.2.1 Основні логічні функції алгебри логіки	27
1.2.2 Функції перемикачів	30
1.3 Мінімізація комутаційних функцій	41
1.3.1. Метод алгебраїчної мінімізації	41
1.3.2. Мінімізація булевих функцій за допомогою карт термінів	43
1.3.3. Інтегральні цифрові мікросхеми є логічними елементами	48
1.3.4. Синтез схеми процесора на основі заданої логічної функції	50
1.3.5 Методи спрощення логічних задач.....	51
РОЗДІЛ 2. ЛОГІЧНІ ЕЛЕМЕНТИ ЦИФРОВОЇ ТЕХНІКИ	54
2.1 Поняття комбінаційної схеми та цифрової машини	54
2.2 Класифікація та характеристика елементної бази цифрових пристроїв	59
2.3 Діодні логічні елементи (DL)	68
2.4 Діодно-транзисторні логічні елементи (ДТЛ)	71
2.5 Транзисторно-транзисторна логіка (ТТЛ)	75
2.5.1 Схема і принцип дії ТТЛ- елемент.....	75
2.5.2 Елементи ТТЛ з відкритим колектором.....	78
2.5.3 Елементи ТТЛ на транзисторах Шотткі	80
2.5.4. ТТЛ елементи з трьома вихідними станами.....	81
2.5.5. Особливості практичних схем елементів ТТЛ.....	84
2.6 Логічні елементи з емітерним зв'язком (ЕЗ)	86
2.6.1. Спрощена схема і принцип роботи елементів ЕЗ	86
2.6.2. Специфіка схем реальних елементів ЕЗЛ	90
2.6.3. Поєднання елементів ЕЗЛ з елементами ТТЛ.....	94
Рис. 2.23. Координаційний елемент ЕЗЛ-ТТЛ	95
2.7 Логічні елементи на польових транзисторах	95
2.7.1. Логічні елементи на <i>n</i> - канальних транзисторах МДН.....	95
Рис. 2.24. Манулювання на транзисторах МДН з нелінійним навантаженням.....	96
2.7.2. Логічні елементи на комплементарних парах транзисторів МДН	99
2.7.3 Логічний елемент КМДН з трьома вихідними станами.....	102
2.7.4. Особливості схем ЛЕ на польових транзисторах та їх узгодження	104
елементи ТТЛ.....	104
2.7.5. Концепція логічних елементів з потужністю інжекції.....	106
РОЗДІЛ 3. КОМПЛЕКСНІ ЦИФРОВІ ПРИСТРОЇ КОМБІНАЦІЙНОГО ТИПУ	109
3.1 Загальні відомості про комбіновані пристрої	109
3.2.1 Принцип побудови однорозрядного суматора.....	111
3.2.2. Принцип побудови розрядного суматора	115
3.3. Цифрові компаратори	119
3.4. Дешифратори	123
3.5. Шифратори	129

3.6. Перетворювач двійково-десятькового коду індикатора в семисегментний....	130
3.7 Мультиплектори та демюльтиплектори.....	132
3.7.1. Загальна характеристика мультиплекторів.....	132
3.7.2. Логічні функції мультиплекторів.....	135
3.7.4. Побудова демюльтиплекторів.....	137
3.7.5. Шинне демюльтиплексування.....	140
РОЗДІЛ 4. ТИПОВИЙ ЦИФРОВИЙ ВУЗОЛ З ПАМ'ЯТЮ.....	142
4.1. Загальні відомості про цифрові запам'ятовуючі пристрої.....	142
4.2 Тригери та їх характеристики.....	144
4.3 Регістри.....	159
4.3.1. Принцип роботи регістрів пам'яті.....	159
4.3.2 Регістри зсуву.....	161
4.4 Лічильники.....	166
4.4.1. Призначення, класифікація та характеристика лічильників.....	166
4.4.2. Лічильники з послідовним переносом.....	167
4.4.3. Лічильники з прохідною передачею.....	171
4.4.4. Лічильники з паралельним перенесенням.....	173
4.4.5. Лічильники з коефіцієнтом перерахунку, відмінним від 2^p	175
4.5. Дільники.....	177
4.6. Генератори та системи формування імпульсів.....	181
ГЛАВА 5. ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ.....	186
5.1. Види запам'ятовуючих пристроїв.....	186
5.2. Основні параметри запам'ятовуючих пристроїв.....	189
5.3. Підтримка пристроїв пам'яті.....	192
5.4 Постійні складські приміщення.....	197
5.5. Пристрої енергонезалежної пам'яті були перепрограмовані.....	203
РОЗДІЛ 6. СИСТЕМИ ВВЕДЕННЯ ТА ВІДОБРАЖЕННЯ ЦИФРОВИЙ ІНФОРМАЦІЯ.....	207
6.1. Пристрої цифрового введення.....	207
6.2. Елементна основа світлофорних пристроїв.....	212
6.3. Вказівні пристрої.....	215
РОЗДІЛ 7. ЦИФРО-АНАЛОГОВІ ПЕРЕТВОРЮВАЧІ.....	218
7.1. Загальні відомості.....	218
7.2. Основні параметри та класифікація ЦАП перетворювачів.....	220
7.2.1. Статичні та динамічні параметри ЦАП.....	220
7.2.2. Класифікація ПКН.....	223
7.3. ПКН на резисторній матриці з ваговими резисторами.....	224
7.4. ПКН на матриці резисторів R-2R.....	227
7.5. Перетворювачі коду в часові рамки.....	229
7.5.1 ПКВ на основі двійкового лічильника.....	230
7.5.2. ПКВ на основі лічильника та цифрового компаратора.....	232
7.6 Особливості застосування ДАС.....	233
РОЗДІЛ 8. АНАЛОГО-ЦИФРОВІ ПЕРЕТВОРЮВАЧІ.....	242
8.1. Основні параметри та класифікація АЦП.....	242
8.2. Перетворювачі напруги в послідовний код підрахунку.....	247
8.2.1. ПНК з поступово розвивається напругою.....	247
8.2.2. Тип відстеження ПНК.....	250
8.3. Побітові перетворювачі напруги в код.....	251
8.4. АЦП паралельного типу.....	255
РОЗДІЛ 9. ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ.....	262
9.1 Класифікація процесорних пристроїв.....	262
9.2. Мікропроцесори у вимірювальних пристроях.....	268
9.3. Поліпшення метрологічних властивостей приладів.....	275
9. 4. Похибки вимірювання.....	279
9.5. Архітектура процесора.....	281
9.6. Типи пам'яті мікроконтролерів.....	288

9.7 . Регістри мікроконтролера. Простір введення-виведення.....	294
9.8. Зовнішня пам'ять	296
РОЗДІЛ 10. АВТОМАТИЧНЕ ПРОЕКТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ У САД РСAD 2004	297
10.1 Загальні відомості про технологію виготовлення друкованих плат.....	297
Система автоматичного проектування САД 2004.....	299
10.2.1. Схематичний графічний редактор	299
10.2.2 Графічний редактор Schematic.....	302
10.2.3 Робота з графічним редактором РСВ.....	307
Список літератури.....	317

СПИСОК СКОРОЧЕНЬ

АЦП - аналого-цифровий перетворювач
ВАХ - вольт-амперна характеристика
ВІС - велика інтегральна схема
ДДНФ - досконала диз'юнктивна нормальна форма
ДЗКД - дільник зі змінним коефіцієнтом ділення
ДКНФ - досконала кон'юнктивна нормальна форма
ДЛ - діодна логіка
ДНФ - диз'юнктивна нормальна форма
ДТЛ - діодно-транзисторна логіка
ЕЗЛ - емітерно-зв'язана логіка
ЕРС - електрорушійна сила
ІЛ - інтегральна інжекційна логіка
ІМС - інтегральна мікросхема
ІШ - інформаційна шина
КМДН - комплементарний МДН - транзистор
КНФ - кон'юнктивна нормальна форма
ЛЕ - логічний елемент
МДН - метал-діелектрик-напівпровідник
МДНЛ - логіка на МДН - транзисторах
МР - молодший розряд
МП - мікропроцесор
ОП - операційний підсилювач
ПЕОМ – персональна електронно-обчислювальна машина
ПЗЗ - позитивний зворотний зв'язок
ПЗп - підсилювач запису
ПЗч - підсилювач зчитування
ПК - пристрій комутації
ПП - пристрій пам'яті
ППП - пристрій постійної пам'яті
ПСТЛ - перемикач струму транзисторної логіки
РВВ - регістр уведення-виведення

РПП - репрограмована постійна пам'ять
РШ - розрядна шина
СР - старший розряд
СШ - струмова сигнальна шина
ТІ - тактовий імпульс
ТТЛ - транзисторно-транзисторна логіка
ТТЛШ - ТТЛ на діодах Шотткі
УГП – умовно графічне позначення
ЦА - цифровий автомат
ЦАП - цифро-аналоговий перетворювач
ЦВ - цифровий вузол

ВСТУП

Інформатизація суспільства — важливе завдання, що вимагає інтенсивного розвитку комп'ютерної техніки та інших методів обробки інформації. Всі різноманітні пристрої цифрової техніки: комп'ютери, мікропроцесорні системи вимірювання й автоматизації технологічних процесів, цифровий зв'язок і телебачення тощо створені на загальній елементній базі, що включає мікросхеми різного ступеня складності - від логічних елементів, що виконують найпростіші операції, до найскладніші програмовані кристали, що містять мільйони логічних елементів.

Цифрова техніка і мікроелектроніка є основою сучасної обчислювальної техніки та техніки управління, тому широко використовується ряд нових класів електронних пристроїв - мікропроцесори, мікроконтролери, мікроперетворювачі, сигнальні процесори. Реалізація сучасних технологій передачі інформації неможлива без використання цифрових систем, функціонування яких базується на процесах створення, передачі, прийому та обробки цифрових сигналів.

Комп'ютери як цифрові пристрої входять до складу систем передачі інформації, управління технологічними процесами, автоматизації, медичних комплексів і науково-технічних досліджень. Сучасні інформаційно-телекомунікаційні системи, що входять до складу цілісних систем передачі інформації, ви-

користовують цифрові сигнали, створення та обробка яких здійснюється за допомогою цифрових технологій. Схемні рішення функціональних вузлів обладнання реалізовано на цифрових інтегральних схемах. Вони мають широкі функціональні можливості, що дозволяють проектувати електронну техніку з високими технічними параметрами.

У підручнику висвітлено теоретичні основи побудови цифрових елементів – системи числення, формати цілих і дійсних чисел, кодування чисел, арифметичні дії над кодами, логічну алгебру та її закони, булеві функції та їх властивості, методи мінімізації булевих функцій та принципи аналіз і синтез цифрових елементів, вузлів і пристроїв на основі логічних елементів. Розглянемо цифрові вузли комбінаційного типу, принципи побудови та реалізації схем суматорів, компараторів, декодерів, кодерів, мультиплексорів та демультіплексори, надано рекомендації щодо каскадування та практичного застосування. Розглянуто роботу послідовних цифрових вузлів: тригерів, лічильників, подільників, регістрів, елементів відображення цифрової інформації, цифрових запам'ятовуючих пристроїв, аналого-цифрових і цифро-аналогових перетворювачів.

При проектуванні сучасних цифрових пристроїв широко використовуються САПР, які забезпечують оптимальне схемне рішення, дозволяють аналізувати технічні характеристики, компоувати компоненти та створювати друковані плати. У посібнику розглянути можливості САПР P-CAD2004 при проектуванні схем цифрових пристроїв, вивченні основних параметрів та розробці друкованих плат.

Підручник призначений для використання в навчальному процесі Державного університету інформаційно-комунікаційних технологій для студентів спеціальності 050901 – Радіотехніка, також може бути корисним аспірантам та інженерно-технічним працівникам, які займаються розробкою та дослідженням мікроелектроніки, цифрових систем, його можна використовувати в інших спеціальних вищих навчальних закладах і середніх, що спеціалізуються на радіоелектроніці та телекомунікаціях.

РОЗДІЛ 1. МАТЕМАТИКО-ЛОГІЧНІ ОСНОВИ ПОБУДОВИ

ЦИФРОВИХ ПРИСТРОЇВ

Під поняттям інформації в широкому розумінні зазвичай розуміють різноманітну інформацію про події суспільного життя, явища, процеси, що відбуваються в технологічних пристроях. Він зустрічається в рідній мові, в текстах книг і газет, в показниках контрольних приладів і відображає різноманіття, властиве предметам і явищам реального світу. Інформація, яка втілена і зафіксована в певній матеріальній формі, називається повідомленням і передається з використанням сигналів. Природа майже всіх фізичних величин така, що вони можуть відображати будь-які значення в будь-якому діапазоні (температура, прискорення, сила тощо). Сигнал, який відображає дану інформацію і з'являється на виході відповідного вимірювального пристрою, може мати нескінченну кількість значень у будь-якому заданому інтервалі часу. Оскільки в цьому випадку безперервний сигнал змінюється подібно до вихідної інформації, його в більшості називають аналоговим, а засоби, в яких діють дані сигнали, — аналоговими. Існують і дискретні сигнали, параметри яких містять фіксований набір індивідуальних показників. А оскільки ця множина скінченна, то кількість інформації в таких повідомленнях також скінченна.

На практиці безперервні повідомлення можуть бути представлені в дискретній формі. Безперервність повідомлень за розміром не може бути забезпечена через помилки джерел і одержувачів інформації. Таким чином, квантування рівня та часу може бути застосоване до безперервних сигналів, що представляють повідомлення. При квантуванні рівня набір можливих значень напруги або струму замінюється кінцевим набором дискретних значень у цьому інтервалі. Квантування часу передбачає заміну безперервного сигналу послідовністю імпульсів, які слідує за певними інтервалами часу, які називаються тактовими сигналами. Якщо тактові інтервали вибрані належним чином, втрати інформації не буде. У разі використання поділу за часом і рівнем амплітуда кожного зразка приймає ближче дозволене значення з вибраного набору значень. Сума всіх виборок створить дискретизований або оцифрований сигнал. Любе значення дискретизованого сигналу може бути представлено числами. У цифровій схемотехніці такий процес називається кодуванням, а сукупність отриманих чисел – сигнальним кодом. Замість перетворення або передачі певних сигналів, ці операції в цифрових пристроях можна виконувати над їх кодами. При цьому можлива робота з аналоговими імпульсами, які перетворюються в цифрові при використанні аналого-цифрового перетворювача (АЦП).

1.1 Системи числення та кодування чисел

1.1.1. Системи числення

Цифрове повідомлення містить набір цифр і символів. Числа можуть бути представлені в різних системах числення. Форма запису чисел у них істотно відрізняється одна від одної, тому, перш ніж переходити до особливостей подання чисел у цифрових пристроях, розглянемо способи їх запису в різних системах рахунку.

Система числення означає спосіб представлення будь-якого числа за допомогою певного набору цифр (символів), кількість яких дорівнює основі системи.

Розрізняють *позиційну* і *непозиційну* системи рахунку. У позиційних системах сутність числа залежить від позиції, яку воно займає в числі, в непозиційних системах - такого співвідношення немає.

Наприклад, у десятковому числі 129 сутність одиниці (1) у розряді сотень перевищує сутність дев'ятки (9) у розряді одиниць; у непозиційній системі, такій як римська, у числі XXX усі десятки рівнозначні, хоча й знаходяться в різних позиціях.

При постановці задач, введення в процесор певних числових співвідношень, вигідно використовувати десяткову систему числення, тому повинні бути засоби відображення чисел у цій системі.

Використання в процесорі двійкової системи, яка має тільки два символи «0» і «1», пояснюється тим, що будь-який двійковий розряд можна представити за допомогою пристрою, який має тільки два стійких стану, наприклад: «на », "вимкнено". Такими пристроями можуть бути перемикачі, реле, тригери, електронні ключі. Недоліком двійкової системи є те, що числа набагато довші, ніж у десятковій. Це компенсується простотою пристроїв, на яких вони реалізовані, їх уніфікованістю та меншими експлуатаційними витратами навіть на досить складні пристрої.

Центральний процесор також використовує допоміжні системи - вісімкову і шістнадцяткову, що пояснюється кратністю їх основ у двійковій системі ($8 = 2^3$, $16 = 2^4$) і коротшим числовим записом порівняно з двійковою системою. Десяткова, двійкова, вісімкова і шістнадцяткова системи числення характеризуються такими даними.

Десяткове число: символи $a = 0, \dots, 2, 3, \dots, 5, 6, 7, 9$; основа $a = 10$.

Двійковий: $a = 0, 1$; $a = 2$.

Вісімкове число: $a = 0, 1, 2, 3, 4, 5, 6, 7$; $a = 8$.

Шістнадцяткове число: $a = 0, \dots, A, B, C, D, E, F$; $a = 16$.

У шістнадцятковій системі, щоб уникнути написання символів над «9», що складаються з двох цифр, використовуються відповідні літери

10 - A,

11 - B,

12 - C,

13 - D,

14 - E,

15 - F

Утворення чисел у будь-якій системі числення відбувається так: позиції, які називаються цифрами, фіксуються. Кожному рангу присвоюється власна вага h_i (де i номер рангу); $h_i = p^i$ (p - основа системи); цифри розміщуються всередині цифр a_i . Тоді будь-яке число A можна виразити у вигляді:

$$A = \sum_{i=-m}^{n-1} a_i h_i \quad (1.1)$$

Послідовність цифр $a_{n-1}, a_{n-2}, \dots, a_1, a_0, a_{-1}, a_{-2}, \dots, a_{-m}$, можна розглядати як код числа в даній системі числення:

$$N = a_n p^n + a_{n-1} p^{n-1} + \dots + a_0 p^0 + a_{-1} p^{-1} + a_{-m} p^{-m} \quad (1.2)$$

ціла частина є дробовою частиною числа

де: $-i_i$ - будь-який символ із повного набору обраної системи числення;

- ($i = 1, 2, 3, \dots, n$) або ($i = -1, -2, -3, \dots, -m$);

- n і m для a_i — порядковий номер цілої (n) і дробової (m) частин числа;

- n і m для p — показники степеня основи цілої (n) і дробової (m) частин числа.

Приклад. Десяткове число $N_{(10)} = 162,75$ у двійковій, вісімковій і шістнадцятковій системах буде представлено таким чином:

$$N_{(2)} = 10100010,11;$$

$$N_{(8)} = 242,6$$

$$N_{(16)} = A2, C.$$

З наведеного прикладу видно, що запис одного й того ж числа в двійковій системі містить найбільшу кількість цифр, а найменшу — у шістнадцятковій (табл. 1.1).

Таблиця 1.1 – Форма запису чисел у різних системах числення

Позначення			
Десятьковий	Двійкова	Вісімкова	Шістнадцяткова
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D

14	1110	16	E
15	1111	17	F
16	10000	20	10

1.1.2 Правила переведення чисел у задану систему числення

Щоб перевести число з будь-якої системи числення в десяткову, необхідно розкласти його в ряд за виглядом (1.2) в десяткову систему і додати умови розкладання, виключаючи нульові добутки.

Переклад двійкових чисел у вісімкове та шістнадцяткове здійснюється шляхом ділення цілої та дробової частин відповідно ліворуч та праворуч від десяткової коми на тріади при перетворенні у вісімкову та на тетради при перетворенні у шістнадцяткову. При цьому пропущені цифри в останніх тріадах (тетрадах) заповнюються нулями.

Тріади і тетради позначають три- і чотирирозрядні двійкові числа відповідно.

Приклад. Переведемо двійкове число $N_{(2)} = 1110011,0110101$ у вісімкове. Цілу і дробову частини чисел розбиваємо на тріади:

ціла частина - 1 6 3

001 110 011

дробова частина - 3 2 4

011 010 100

В останніх тріадах цілої та дробової частин додаються нулі. Отже, число $N_{(8)} = 163,324$ має вигляд

$N_{(2)} = 1110011,0110101$.

Це ж число переведемо в шістнадцяткову систему, розділивши його цілу і дробову частини на тетради:

7 3 6 A

0111 0011 0110 1010.

Тут також додаються нулі в останні тетради. Зверніть увагу, що вони не змінюють значення числа, оскільки додаються на початку цілого числа та в кінці дробової частини числа. Шістнадцяткове число $N_{(16)} = 73,6$ А має вигляд $N_{(2)} = 1110011,0110101$.

Розглянемо переклад вісімкових і шістнадцяткових чисел у двійкові. Для перекладу іменовані числа необхідно представити порозрядно (цілі і дробові частини) у вигляді двійкових тріад і тетрад і записати на них відповідні двійкові числа.

Приклад . Вісімкове число $N_{(8)} = 1675,324$.

Запишемо порядок тріад для кожного розряду цілої та дробової частин -

001 110 111 101, 011 010 100.

Двійкове число буде $N_{(2)} = 1\ 110111101,0110101$.

приклад. Шістнадцяткове число $N_{(16)} = \text{FAB6,3E5}$.

Запишемо порядок тетрад для кожного розряду цілої та дробової частин -

1111 1010 1011 0110, 0011 1110 0101.

Тоді двійкове число $N_{(2)} = 1111101010110110,001111100101$.

Сформулюємо правила перекладу.

1. Щоб перетворити двійкове число у вісімкове (шістнадцяткове), розділіть його цілу та дробову частини справа наліво та зліва направо від коми відповідно на тріади (тетради), дописуючи пропущені цифри в останні тріади (тетради).) з нулями. Вісімкове (шістнадцяткове) число записується тріадами (тетрадами) зліва направо, від цілої частини до дробової.

2. Для переведення вісімкового (шістнадцяткового) числа в двійкове кожену цифру цілого числа і дробову частину числа потрібно записати трійками (тетрадами). Двійкове число записується комбінуванням тріад (тетрад) цілої і дробової частин числа, виключаючи нулі на початку цілого числа і в кінці дробових частин.

При складанні програм на алгоритмічних мовах в основному використовують десяткову систему числення. Проте всі арифметичні дії МП виконуються в двійковій системі числення, і для переходу до неї необхідна проміжна т.зв. двійково-десятковий код, який передбачає запис кожної цифри дробової частини та цілої десятикової частини у двійкові тетради.

Наприклад, число $N_{(10)} = 286,74$ у двійковій десятиковій системі записується як

$$N_{(2-10)} = 0010\ 1000\ 0110,0111\ 0100.$$

Переклад цілої та дробової частин десятикових чисел у двійкові, вісімкові та шістнадцяткові числа здійснюється за різними правилами, які можна сформулювати таким чином.

Щоб перевести десяткове число в двійкову систему (вісімкову, шістнадцяткову), потрібно послідовно ділити число на основу системи числення 2 (8 або 16), фіксуючи остачу і частоту при кожному діленні, поки не вийде число в останній дріб мінус основа нової системи числення. Число в двійковій системі (8-бітна, 16-бітна) записується шляхом взяття останньої частки та залишку від ділення на кожному кроці, починаючи з останньої частки, зліва направо.

Приклади переведення всього десятикового числа $N_{(10)} = 327$ в двійкову, вісімкову і шістнадцяткову системи наведені у вигляді стовпчиків, в яких частки і остачі, отримані при діленні числа 327 за системою числення $a = 2, 8$ і 16 відповідно записують запис порядкового номера в системі з новою основою (2, 8, 16), позначеною у колонках стрілками (рис. 1.1).

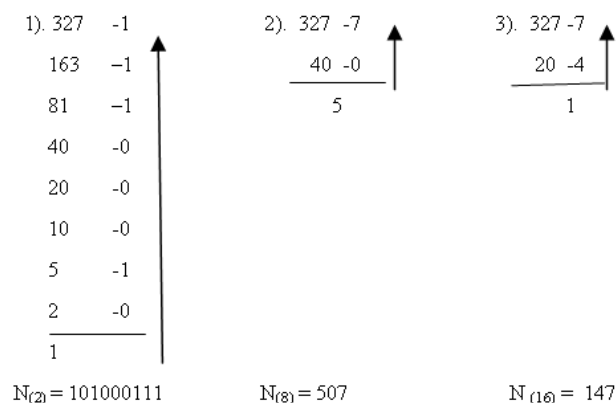


Рис. 1.1 Приклади перекладу десятикового числа N

Щоб правильно перевести десяткову або дробову частину змішаного числа в двійкову систему числення (вісімкову, шістнадцяткову), помножьте дробову частину вихідного числа й отримані неповні добутки на 2 (8 або 16), щоб отримати вказане число. десяткових цифр у новій системі числення. Запис числа в новій системі здійснюється за допомогою цілих частин неповних добутків, починаючи з першої, зліва направо. Зверніть увагу, що кількість цифр у новій системі визначає точність перекладу чисел з однієї системи в іншу. У машинному перекладі кількість бітів визначається розрядною сіткою елементів пам'яті.

Приклади переведення десяткового дробу $N_{(2)} = 0,841$ у двійковий, вісімковий і шістнадцятковий показані у відповідних стовпцях послідовних множень на 2, 8 і 16 із зазначенням часткових добутків, отриманих на кожному кроці множення. Стрілками вказано порядок запису чисел у новій системі числення з основами 2, 8 і 16 (рис. 1.2).

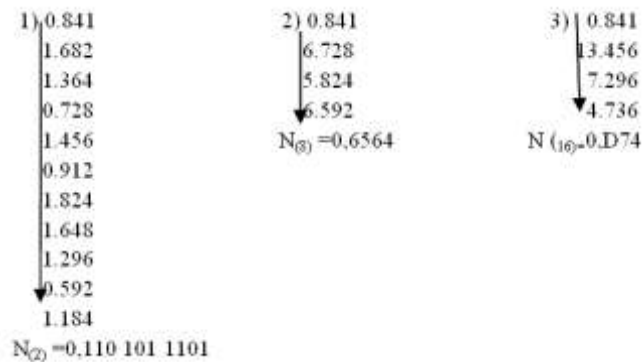


Рис. 1.2 Приклади перекладу десяткового дробу в двійковій, вісімковій і шістнадцятковій системах

Перевірка правильності переведення десяткового дробу $N_{(10)} = 0,841$ в системі числення з основами 2, 8 і 16:

приклад 1 2 8 9 10

$$N_{(2)} = 0,1101011101 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-4} + 1 \cdot 2^{-6} + 1 \cdot 2^{-7} + 1 \cdot 2^{-8} + 1 \cdot 2^{-10} = 0,8408203125;$$

приклад 1 2 3 4

$$N_{(8)} = 0,6564 = 6 \cdot 8^{-1} + 5 \cdot 8^{-2} + 6 \cdot 8^{-3} + 4 \cdot 8^{-4} = 0,8408203125;$$

приклад 1 2 3

$$N_{(16)} = 0.D74 = 13 \cdot 16^{-1} + 7 \cdot 16^{-2} + 4 \cdot 16^{-3} = 0,8408203125.$$

Продовжуючи множення, ви можете досягти високої точності перекладу.

Розв'язування алгебраїчних і диференціальних рівнянь різного порядку на процесорі в кінцевому рахунку зводиться до двох елементарних арифметичних операцій - додавання і віднімання. Навіть операції множення та ділення є просто похідними від додавання та віднімання, доповнених операцією зсуву. Таке спрощення було досягнуто завдяки широкому розвитку чисельних методів вирішення складних математичних співвідношень, а також обмеження операцій над числами в різних обчислювальних системах двійковою арифметикою.

Правила проведення арифметичних операцій над двійковими числами подано в таблицях додавання, віднімання і множення:

додавання віднімання множення

$$0 + 0 = 0 \quad 0 - 0 = 0 \quad 0 * 0 = 0$$

$$0 + 1 = 1 \quad 1 - 0 = 1 \quad 0 * 1 = 0$$

$$1 + 0 = 1 \quad 1 - 1 = 0 \quad 1 * 0 = 0$$

$$1 + 1 = 10 \quad 10 - 1 = 1 \quad 1 * 1 = 1$$

Слід звернути увагу:

1) одиниця від результату останньої суми в таблиці додавання переходить у старший розряд;

2) в останній різниці розрахункової таблиці одиниця запозичується з вищого порядку, перетворюючись на дві одиниці нижчого порядку.

Особливістю операції множення двійкових чисел є те, що кількість нулів у множнику відповідає числу зсувів добутку вліво в стовпчику суми, а дода-

ються числа, які завжди дорівнюють множнику, оскільки вони перемножуються по одному. Операція ділення заснована на операціях віднімання та зсуву.

1.1.3. Формати цілих і дійсних чисел

У процесорі числа представлені у двох формах: з фіксованою комою (звичайна форма) і з плаваючою комою (напівлогарифмічна форма). При поданні чисел із фіксованою крапкою мається на увазі, що ця крапка фіксується або перед старшою цифрою дробового числа, або після молодшої цифри цілого числа. На рис. 1.3 надає графічне представлення елементів пам'яті для цілих і дробових чисел з фіксованою комою різної розрядності: а) невелике число у форматі машинного слова (4 байти); б) ціле число у форматі машинного слова; в) ціле в машинному форматі півслова (2 байти).

Байт - це восьмирозрядне двійкове число, прийняте як одиниця представлення даних в ПК, тому кількість біт в комірках пам'яті завжди кратно восьми, і дані відповідно мають байтову структуру, тобто складаються з певну кількість байтів.

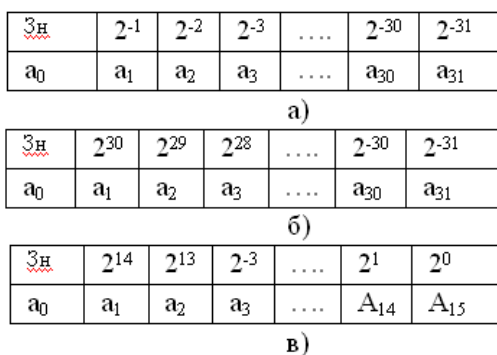


Рис. 1.3 Будова комірок пам'яті при записі двійкових чисел

Знакова цифра елемента пам'яті дорівнює «0» означає додатне число, а «1» означає від'ємне число, що зберігається в місці пам'яті. Найбільше число

у формі з фіксованою комою відповідає заповненню всіх розрядів комірок пам'яті одиницями, а найменше - нулями (крім молодшого розряду), відповідно, вони рівні.

Числа більше зазначених максимальних значень не можна вводити в комірки пам'яті - бітова сітка буде переповнена. Ви можете запобігти переповненню бітової сітки, масштабуючи дані перед введенням їх у процесор. Формат даних з фіксованою комою в процесорі в основному використовується для представлення адрес чисел, констант і даних, які не вимагають високої пропускну здатності. Арифметичні операції над числами, представленими у формі фіксованої коми, виконуються з вищою швидкістю порівняно з числами, представленими у формі з плаваючою комою. Однак через можливе переповнення розрядної сітки використання звичайної форми в обчисленнях обмежене.

Для вирішення широкого кола обчислювальних завдань найважливішим є представлення чисел у формі з плаваючою комою, яка, незважаючи на меншу швидкість виконання операцій, не потребує масштабування даних.

Зображення з плаваючою комою будь-якого числа N зазвичай описується формулою

$$N = \pm ma \pm p,$$

де m – мантиса (дробова частина) числа; a - основа системи рахунку; p - рядок (ціле).

Наприклад, десяткове число 238,745 і двійкове число 11001,1011 можна записати в декількох варіантах відповідно за формулою:

$$0,238745 * 10^3; 0,0238745 * 10^4; 0,000238745 * 10^6;$$

$$0,110011011 * 10^1; 0,00110011011 * 10^{11}$$

У першому варіанті запису чисел після коми стоїть значуща цифра (2 в десяткових зображеннях і 1 в двійкових). Це так звана *нормалізована* форма чисел з плаваючою комою, останні записи не нормалізуються, оскільки після коми є нулі. МР автоматично нормалізує числа. Цифри мантиси, їх порядок і знаки зберігаються в комірках пам'яті.

У двійковому представленні чисел для модуля порядку в комірці пам'яті відведено шість двійкових цифр (рис. 1.4), що відповідає $p_{max} = 63$ або $p_{max} = 111111$. У випадку модуля мантиси 23 цифри є виділені для $m_{max} = 1-2^{-24}$.

Знак мантиси	Знак порядку	Модуль порядку					Модуль мантиси (a=2)				
		2	3	4	7	8	9	30	31
0	1	b ₁	b ₂	b ₃	b ₆	a ₁	a ₂	a ₂₃	a ₂₄

Рис. 1.4. Будова комірок пам'яті щодо двійкових чисел з плаваючою комою у форматі слова (4 байти)

Таким чином, діапазон нормалізованих чисел, представлених у двійковій формі, буде в діапазоні

$$N_{min} = (0,100...0) \cdot 2^{-63} = 2^{-64}$$

$$N_{max} = (0,111...1) \cdot 2^{63} = (1 - 2^{-24}) \cdot 2^{63}$$

Враховуючи знак мантиси, діапазони від'ємних і додатних чисел рівні відповідно

$$\text{від } -(1 - 2^{-24}) \cdot 2^{63} \text{ до } -2^{-64},$$

$$\text{від } +2^{-64} \text{ до } +(1 - 2^{-24}) \cdot 2^{63}.$$

Наведені десяткові обмеження відповідають N значень $\cong 10^{+/-19}$, які набагато перевищують діапазон чисел з фіксованою комою, представлених у 32 бітах елемента пам'яті.

Шістнадцяткове представлення чисел (формат слова) в елементі пам'яті має дві модифікації (рис. 1.5), які відрізняються способом представлення знака порядку.

Знак мантиси	Знак порядку	Модуль порядку					Модуль мантиси (a=16)				
		2	4	5	7	8-11	12-15	16-23	24-27	28-31
0	1	b ₁	b ₂	b ₃	b ₆	h ₁	h ₂	h ₅	h ₆

Рис. 1.5 Структура комірок пам'яті для 16-розрядних чисел з плаваючою комою у форматі слова (4 байти)

Для зручності зберігання 16-розрядних чисел з плаваючою комою у форматі word використано т.зв. *зміщений порядок*, у якому цифри мантиси поділені на півбайти (4 цифри). Кожен бітл представляє одну шістнадцяткову цифру в тетрадах (0000 - 1111) - загалом шість бітлів (Малюнок 1.6). Ступені рядка є двійковими, максимальне значення рядка $p_{max} = 63$. Суть зміщеного рядка полягає в тому, що його кінцеве значення p_{cm} розглядається як сума $p_{cm} = p_{cm} + M$, де p - порядок числа, введеного в пам'ять, який може приймати значення від $p_{max} = 63$ до $p_{min} = -64$; M - зсув порядку, $M = 2^k$, k - кількість двійкових цифр, що використовуються в модулі замовлення.

Зміщений порядок зберігається в семи бітах елемента пам'яті без знакового біта. Отже, максимальне значення зміщеного порядку $p_{cm\ max}$ може дорівнювати $2^7 - 1 = 127$. Діапазон зміни зміщеного порядку при зміні порядку введених у пам'ять чисел від $p = -64$ до $p = 63$ буде в діапазоні від $p_{cm} = 0$ до $p_{cm} = 127$. При цьому інформація про знак рядка введеного числа вкладається в значення r_{cm} без додаткової цифри знака.

Знак мантиси	Модуль зміщеного порядку					Модуль мантиси (a=16)				
	1	2	3	7	8-11	12-15	16-23	24-27	28-31
a_0	b_1	b_2	b_3	b_6	h_1	h_2	h_5	h_6

Рис. 1.6 Структура комірок пам'яті для 16-розрядних чисел з плаваючою точкою у форматі слова (4 байти) із порядком зсуву

Використання зміщеного порядку полегшує операції над рядками під час виконання арифметичних операцій над числами, представленими у формі з плаваючою комою, зводячи їх до операцій над додатними цілими числами.

У форматі слова для шістнадцяткового представлення ми отримуємо діапазони нормалізованих чисел, включаючи знак мантиси:

$$\text{від } -(1 - 16^{-6}) * 16^{63} \text{ до } -16^{-64}$$

$$\text{від } 16^{-64} \text{ до } (1-16^{-6}) * 16^{63} .$$

Це в десятковому вигляді відповідає значенню $N = 10^{-77} \dots 10^{+76}$.

Щоб зберігати шістнадцяткові числа у двійковому форматі слова (8 байт), встановіть довжину мантиси на 14 бітів, порядок зсуву – на -7 бітів, для загальної довжини 64 біти (Малюнок 1.7).

Знак мантиси	Модуль зміщеного порядку					Модуль мантиси (a=16)				
	1	2	3	...	7	8-11	12-15	16-55	56-59	60-63
a ₀	b ₁	b ₂	b ₃	...	b ₆	h ₁	h ₂	...	h ₁₃	h ₁₄

Рис. 1.7 Структура комірок пам'яті для 16-розрядних чисел з плаваючою комою у форматі подвійного слова (8 байт) зі зміщенням

Діапазон подання чисел не змінюється, а точність подання підвищується завдяки мантисі. Використання шістнадцяткового формату з плаваючою комою економить бітовий простір і забезпечує зручність запису двійкових чисел.

1.1.4. Кодування номерів

Для спрощення арифметичних операцій фіксовані числа і числа з плаваючою комою представлені за допомогою спеціальних кодів. Це дозволяє замінити операцію віднімання операцією додавання в інверсних і додаткових кодах, підвищити надійність розпізнавання переповнення бітової сітки та спростити пристрої ЦП, що виконують арифметичні операції.

Для кодування чисел використовуються коди: прямий, зворотний, додатковий, видозмінений (зворотний і додатковий).

Числа в кодах представлені у вигляді правильних дробів, знакові розряди яких включають нуль - якщо число додатне, і одиницю - якщо число від'ємне.

При цьому додатні числа в прямому, оберненому та додатковому кодах представлені однаково, тому можна записати, що при $x > 0$

$$x_{i\bar{0}} = \tilde{0}_{\bar{c}\bar{a}} = \tilde{0}_{\bar{a}\bar{i}\bar{a}} = \tilde{0}$$

Закодовані позитивні числа ($x > 0$) загалом можна представити формулою

$$x = 0.x_1x_2x_3\dots x_n$$

де: 0 - додатний знак числа (знакова цифра);

$x_1\dots x_n$ - цифри його мантиси.

Зображення від'ємних чисел у прямому, оберненому та додатковому кодах описуються відповідно загальними формулами

$$x_{np} = 1.x_1x_2x_3\dots x_n,$$

$$\tilde{\delta}_{\dot{c}\dot{a}} = 1.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots\tilde{\delta}_i$$

$$\tilde{\delta}_{\ddot{a}\ddot{a}} = 1.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots(\tilde{\delta}_i + 1),$$

де 1 — від'ємний знак числа, знаковий розряд;

\bar{x}_i - перетворення (інверсія) цифрового числа (0 або 1), тобто перехід одиниці в нуль і нуля в одиницю;

($\bar{x}_n + 1$) - додавання числа в зворотному коді на одну одиницю в останньому (молодшому) розряді.

На основі загальних виразів можна сформулювати правила перетворення чисел, заданих у вигляді власних дробів, у прямі, обернені та додаткові коди.

1. Переклад додатних чисел у прямий, обернений і додатковий коди не змінює образу цих чисел.
2. При перекладі від'ємного числа в код знак «мінус» перед числом пропускають по всій частині, а замість нуля вставляють одиницю. У розрядах мантиси в прямому коді послідовність нулів і одиниць не відрізняється від вихідної, в зворотному коді - інверсія нулів і одиниць, в додатковому коді після інверсії нулів і одиниць - одиниця. додається до наймолодшої цифри мантиси.

Приклад . Дане число є від'ємним числом $x = -0,101101$.

Його зображення в кодах: $x_{\dot{c}\dot{a}} = 1,101101$ $\tilde{\delta}_{\dot{c}\dot{a}} = 1,010010$ $\tilde{\delta}_{\ddot{a}\ddot{a}} = 1,010011$

Переклад чисел у модифікований зворотний і додатковий коди відрізняється від наведеного лише тим, що для додатних і від'ємних чисел замість однієї ставляться дві знакові цифри за загальними зразками:

Коли $x > 0$

$$x_{\dot{c}\dot{a}_i} = 00.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots\tilde{\delta}_i,$$

$$\tilde{\delta}_{\bar{a}\bar{a}_i} = 00.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots\tilde{\delta}_i.$$

Коли $x > 0$

$$\tilde{\delta}_{\bar{c}\bar{a}_i} = 11.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots\tilde{\delta}_i,$$

$$\tilde{\delta}_{\bar{a}\bar{a}_i} = 11.\tilde{\delta}_1\tilde{\delta}_2\tilde{\delta}_3\dots(\tilde{\delta}_i + 1)$$

Представлення чисел у модифікованих кодах полегшує розпізнавання ситуацій переповнення бітової сітки під час виконання арифметичних операцій. Це пов'язано з появою в знаку цифр результатів заборонених операцій кодових комбінацій 10 або 01, які не відповідають очікуваним 00 для позитивних результатів і 11 для негативних результатів.

Приклад. Після додавання чисел $A = 00.110101$ і $B = 00.101001$ і $C = 11.001100$ і $D = 11.100110$, записаних у модифікованому додатковому коді, бітова сітка переповнюється:

$$A = 00,110101 \quad C = 11,001100$$

++

$$B = 00,101001 \quad D = 11,100110$$

$$A+B=01,011110 \quad C+D=1 \quad \leftarrow 10,110010$$

Як бачимо, результати підсумовування знакових розрядів містять недопустимі комбінації 01 і 10 (одиниця в третьому розряді суми C і D при додаванні чисел у зміненому додатковому коді відкидається). Це означає, що результати підсумовування більші за одиницю і шляхом введення цих чисел необхідно було масштабувати додаток. Розпізнавання заборонених комбінацій при переповненні здійснюється за допомогою спеціальних логічних пристроїв.

1.1.5. Арифметичні операції над кодами

Розглянемо порядок виконання операцій при додаванні чисел з плаваючою комою:

- вирівнюючи порядок додатків, менший порядок призводить до більшої операції зсуву мантиси вправо;

- мантиси термінів перетворюються на модифікований зворотний (або додатковий) код;
- коди mantis виконуються побітово, включаючи знакові цифри,
- коли в модифікованому додатковому коді в цифрах знака результату з'являється додаткова одиниця, ця одиниця відкидається, у модифікованому зворотному коді одиниця додається до молодшої цифри мантиси;
- мантиса, отримана в зворотному (або додатковому) коді, перетворюється в прямий код;
- після отримання ненормалізованої мантиси суми виконується операція нормалізації (зсув вліво) і перевпорядкування.

Приклад . Додайте двійкові числа

$$x_{(2)} = 0.1011101 \cdot 10^{-100} \quad y_{(2)} = -0.1101101 \cdot 10^{-101}$$

Подання цих чисел у прямому коді з урахуванням порядків буде.

$$x_{i\bar{0}} = 0.1011101 \cdot 10^{-100} \quad \acute{o}_{i\bar{0}} = 1.1101101 \cdot 10^{-101}$$

Порівнюємо замовлення:

$$\tilde{\delta}_{i\bar{0}} = 0.1011101 \quad \acute{o}_{i\bar{0}} = 1.01101101$$

Представимо мантиси в модифікованих (зворотних і додаткових) кодах і додамо x і y :

$$\tilde{\delta}_{\acute{c}\acute{a}_i} = 00.10111010 \quad \tilde{\delta}_{\acute{a}\acute{a}_i} = 00.10111010$$

$$\acute{o}_{\acute{c}\acute{a}_i} = 11.10010010 \quad \acute{o}_{\acute{a}\acute{a}_i} = 11.10010011$$

$$100.01001100 \quad 100.01001101$$

| _____ → 1 ← _____ |

цикл. відхилити передачу

$$x + y = 00,01001101 \quad x + y = 00,01001101$$

Результати підсумовування, отримані в модифікованих кодах, позитивні та ненормовані. Для нормалізації необхідно зрушити мантису вліво на одну цифру і зменшити порядок на одиницю, тоді кінцевий результат буде $x + y = 00,1001101 \cdot 10^{-101}$. Немає необхідності перетворювати результат у прямий код, оскільки додатні числа представлені однаково в усіх кодах.

Сучасні процесори обробляють не тільки числову інформацію, а й текстову інформацію, тобто буквено-цифрову інформацію, що включає цифри, букви, знаки пунктуації, математичні символи та інші спеціальні символи.

1.2 Основи логічної алгебри та логічних функцій

1.2.1 Основні логічні функції алгебри логіки

В алгебрі логіки ми розглядаємо змінні, які можуть приймати тільки два значення: 0 і 1. Далі змінні будемо позначати латинськими літерами x, y, z, \dots

В алгебрі логіки еквівалентність визначено відношення ($=$) і три операції:

- диз'юнкція (операція АБО), позначена знаком \vee ;
- сполучник (операція And), позначений крапкою, яку можна опустити (наприклад, $x \cdot y = xy$);
- заперечення (інверсія, операція НЕ), позначене тире над змінними або елементами 0 і 1 (наприклад $\bar{x}, \bar{0}, \bar{1}$).

Відношення еквівалентності задовольняє такі властивості:

- $x = x$ - рефлексивність ;
- якщо $x = y$, то $y = x$ - симетрія ;
- якщо $x = y$ так $y = z$, то $x = z$ - транзитивність .

Принцип підстановки впливає з відношення еквівалентності: якщо $x = y$, то в будь-якій формулі, що містить x , y можна замінити на x і вийде еквівалентна формула.

Алгебра логіки визначається такою системою аксіом:

$$\left. \begin{array}{l} x = 0, \text{ якщо } x \neq 1, \\ x = 1, \text{ якщо } x \neq 0; \end{array} \right\} \quad (1.3)$$

$$\left. \begin{array}{l} 1 \vee 1 = 1, \\ 0 \cdot 0 = 0; \end{array} \right\} \quad (1.4)$$

$$\left. \begin{array}{l} 0 \vee 0 = 0, \\ 1 \cdot 1 = 1; \end{array} \right\} \quad (1.5)$$

$$\left. \begin{array}{l} 0 \vee 1 = 1 \vee 0 = 1, \\ 1 \cdot 0 = 0 \cdot 1 = 0; \end{array} \right\} \quad (1,6)$$

$$\left. \begin{array}{l} \bar{0} = 1, \\ \bar{1} = 0. \end{array} \right\} \quad (1,7)$$

Аксиома (1.3) стверджує, що в логічній алгебрі розглядаються лише бінарні змінні, аксіоми (1.4) - (1.6) визначають операцію диз'юнкції та кон'юнкції, а аксіома (1.7) визначає операцію заперечення.

Якщо в наведених попарно аксіомах (1.4)-(1.7) виконати взаємну заміну операцій диз'юнкції та кон'юнкції та елементів 0 і 1, то отримаємо ще одну з однієї пари аксіом. Ця властивість називається *принципом двоїстості*.

Багато теорем і тотожностей можна довести за допомогою аксіом логічної алгебри. Одним із ефективних методів доведення теорем є *метод сортування за всіма значеннями змінних*. Таким чином, можна легко перевірити справедливність тверджень, зведених у таблиці 1.2, використовуючи метод сортування.

Якщо в логічному виразі присутні операції диз'юнкції та кон'юнкції, необхідно дотримуватися порядку виконання операцій: спочатку виконується операція кон'юнкції, а потім — диз'юнкція. У складних логічних виразах круглі дужки використовуються для визначення порядку операцій.

Деякі теореми та тотожності логічної алгебри мають особливе значення, оскільки вони дозволяють спрощувати логічні вирази. Тотожності (1.17)-(1.20) особливо часто використовуються для перетворення логічних виразів.

Таблиця 1.2 – Закони логічної алгебри

Назва закону	Формули	
Ідемпотент	$\left. \begin{array}{l} x \vee x = x \\ x \cdot x = x \end{array} \right\}$	(1,8)
Комутативні	$\left. \begin{array}{l} x \vee y = y \vee x \\ x \cdot y = y \cdot x \end{array} \right\}$	(1,9)
Асоціатив- ний	$\left. \begin{array}{l} (x \vee y) \vee z = x \vee (y \vee z) \\ (x \cdot y) \cdot z = x \cdot (y \cdot z) \end{array} \right\}$	(1.10)

Дистрибутивний	$\left. \begin{aligned} x \cdot (y \vee z) &= x \cdot y \vee x \cdot z \\ x \vee y \cdot z &= (x \vee y) \cdot (x \vee z) \end{aligned} \right\}$	(1,11)
Із запереченням	$\left. \begin{aligned} x \vee \bar{x} &= 1 \\ x \cdot \bar{x} &= 0 \end{aligned} \right\}, \left. \begin{aligned} 0 \vee x &= x \\ 1 \cdot x &= x \end{aligned} \right\}, \left. \begin{aligned} 1 \vee x &= 1 \\ 0 \cdot x &= 0 \end{aligned} \right\}$	(1,12-14)
Подвійності (де Моргана)	$\left. \begin{aligned} \overline{x \vee y} &= \bar{x} \cdot \bar{y} \\ \overline{x \cdot y} &= \bar{x} \vee \bar{y} \end{aligned} \right\}$	(1,15)
Двомісний відмова	$\overline{\bar{x}} = x$	(1,16)
Поглинання	$\left. \begin{aligned} x \vee x \cdot y &= x \\ x \cdot (x \vee y) &= x \end{aligned} \right\}$	(1,17)
Операції склеювання	$\left. \begin{aligned} x \cdot y \vee x \cdot \bar{y} &= x \\ (x \vee y) \cdot (x \vee \bar{y}) &= x \end{aligned} \right\}$	(1,18)
Операції узагальнене склеювання	$\left. \begin{aligned} x \cdot y \vee \bar{x} \cdot z \vee y \cdot z &= x \cdot y \vee \bar{x} \cdot z \\ (x \vee y) \cdot (\bar{x} \vee z) \cdot (y \vee z) &= (x \vee y)(\bar{x} \vee z) \end{aligned} \right\}$	(1,19)
	$\left. \begin{aligned} x \vee \bar{x} \cdot y &= x \vee y \\ x \cdot (\bar{x} \vee y) &= x \cdot y \end{aligned} \right\}$	(1,20)

Операція «сума по модулю два» (виключаюче АБО, логічна нерівність) позначена символом \oplus і визначається співвідношенням

$$x \oplus y = \bar{x}y \vee x\bar{y} = (\bar{x} \vee \bar{y}) \cdot (x \vee y). \quad (1,21)$$

Таблиця істинності для (1.3) має вигляд:

Таблиця 1.3 – Таблиця істинності для суми по модулю два

експресія	Значення
$0 \oplus 0$	0
$0 \oplus 1$	1
$1 \oplus 0$	1
$1 \oplus 1$	0

Операція підсумовування за модулем два є комутативною, кумулятивною та розподільною відносно операції кон'юнкції, тобто

$$\left. \begin{aligned} x \oplus y &= y \oplus x, \\ x \oplus (y \oplus z) &= (x \oplus y) \oplus z, \\ x \cdot (y \oplus z) &= x \cdot y \oplus x \cdot z \end{aligned} \right\} \quad (1,22)$$

Для неї також вірні такі ідентичності:

$$\left. \begin{aligned} x \oplus 0 &= x; \\ x \oplus 1 &= \bar{x}; \\ x \oplus \bar{x} &= 0; \\ \overline{x \oplus y} &= \bar{x} \cdot \bar{y} \vee x \cdot y = (\bar{x} \vee y) \cdot (x \vee \bar{y}) = \bar{x} \oplus y = x \oplus \bar{y} \end{aligned} \right\} \quad (1,23)$$

1.2.2 Функції перемикання

Любий логічний вираз, який складається з n змінних, x_1, \dots, x_n можна розглядати як функція від n перемінних. Відповідно до аксіом (1.3) - (1.7) функція в змозі приймати лише два значення: 0 і 1. Отже, ці функції придатні для опису, аналізу та синтезу комутаційних схем, сигналами, яких характеризується лише 2-ма рівнями напруги: високим (1) і низьким (0). Тому такі функції називають функціями переключення.

Властивості функцій переключення. Для функцій від n змінних x_1, \dots, x_n застосовується загальне позначення $f(v) = f(x_1, \dots, x_n)$, де $v = (x_1, \dots, x_n)$ це набір перемінних, що розглядається як n -вимірний вектор. Люба змінна x_p ($p = 1 \dots n$) має можливість приймати два значення: 0 і 1.

Визначене значення змінної x_1, \dots, x_n (0 або 1) позначається e_p .

Щоб призначити функцію, $f(v)$ необхідно вказати її значення в усіх точках області визначення, тобто встановити значення $f(v_i) = 0$ або 1, де $i = 0, 1, \dots, 2^n - 1$. Кожній конкретній функції з n змінних може бути присвоєно 2^n бітове число, що складається зі значень $f(v_i) = 0$ або 1 ($i = 0, 1, \dots, 2^n - 1$), які вона приймає в 2^n точках області визначення.

Функції n змінних можуть не залежати від усіх змінних x_1, \dots, x_n . Такі функції називаються *виродженими*. Великий інтерес становлять невироджені функції двох змінних x_1, x_2 , імена яких даються за допомогою операцій логічної алгебри, використаних для їх створення:

$$f(x_1, x_2) = x_1 \vee x_2 - \text{диз'юнкція (АБО)};$$

$$f(x_1, x_2) = x_1 \cdot x_2 - \text{сполучник (І)};$$

$$f(x_1, x_2) = \overline{x_1 \cdot x_2} - \text{функція І-НІ};$$

$$f(x_1, x_2) = \overline{x_1 \vee x_2} - \text{функція АБО-НІ};$$

$$f(x_1, x_2) = x_1 \oplus x_2 - \text{сума по модулю 2}.$$

Область функції n перемінних може визначена масивом значень, які вона приймає в точках v_i . Такі таблиці називають *таблицями істинності*.

Таблиця 1.4, підготовлена відповідно до аксіом (1.3)-(1.7) для конкретних функцій, є таблицею істинності, що визначає ці функції.

Табл. 1.4 – Таблиця істинності для двох змінних LF x_1, x_2

x_1, x_2	$x_1 \vee x_2$	$x_1 \cdot x_2$	$\overline{x_1 \cdot x_2}$	$\overline{x_1 \vee x_2}$	$x_1 \oplus x_2$
0 0	0	0	1	1	0
0 1	1	0	1	0	1
1 0	1	0	1	0	1
1 1	1	1	0	0	0

Функція n змінних $f(v)$ називається *повністю визначеною*, якщо її значення дорівнюють 0 або 1 у всіх точках області визначення. $f(v_i) =$ Якщо значення функції не визначено хоча б в одній точці v_i , то вона називається *неповністю визначеною*.

Принцип і закони подвійності.

Дана алгебра має властивість, яка називається *принципами подвійності*: якщо тотожність існує, $f(v, 0, 1 / \vee, \&) = g(v, 0, 1 / \vee, \&)$ то тотожність також дійсна

$f(v,1,0/\&,v) = g(v,1,0/\&,v)$, тобто якщо в будь-якій тотожності символи 0 і 1 поміняні місцями та виконуються операції диз'юнкції та кон'юнкції, то тотожність буде також отримати.

Дві ідентичності, поєднані таким чином, подвійні. Істинність самого принципу двоїстості не доведена, оскільки цей принцип є внутрішньою властивістю алгебри логіки (полягає в її аксіомах).

Закони подвійності (теореми де Моргана) (1.13) встановлюють метод знаходження обернених функцій, які є альтернативою та кон'юнкцією двох змінних.

Клод Шеннон запропонував загальна теорема, яка дозволяє знайти обернену до будь-якої функції $f(v)$.

Закон подвійності Клода Шеннона має вигляд:

$$\overline{f(v/\vee,\&)} = f(\overline{v}/\&,v) \quad (1,24)$$

тобто інверсію будь-якої функції можливо отримати взаємною підстановкою змінних x_p ($p=1..n$) і операціями диз'юнкції та кон'юнкції.

Розглянемо кілька прикладів.

Дозволити

$$f(v) = \overline{x_2} \cdot x_1 \vee x_2 \cdot \overline{x_1}, \quad \text{Тоді} \quad \overline{f(v)} = (x_2 \vee \overline{x_1}) \cdot (\overline{x_2} \vee x_1)$$

Дозволити

$$f(v) = [(\overline{x_2} \cdot x_1 \vee x_3 \cdot x_2) \cdot x_3 \cdot x_1 \vee x_3 \cdot \overline{x_1}] \cdot (x_2 \cdot \overline{x_1} \vee x_3) \vee x_4, \quad \text{Тоді}$$

$$\overline{f(v)} = \{ [(x_2 \cdot \overline{x_1}) \cdot (\overline{x_3} \vee \overline{x_2}) \vee \overline{x_3} \vee \overline{x_1}] \cdot (\overline{x_3} \vee x_1) \vee (\overline{x_2} \vee x_1) \cdot \overline{x_3} \} \cdot x_4$$

На підставі закону подвійності це легко продемонструвати

$$\overline{\bigvee_{p=1}^n x_p} = \prod_{p=1}^n \overline{x_p}; \quad \prod_{p=1}^n \overline{x_p} = \overline{\bigvee_{p=1}^n x_p}.$$

Теорема про розкладання та пов'язані з нею тотожності.

Теорема розкладання має особливе значення : кожна функція $f(v)$ може бути розкладена на змінну x_p виду

$$f(x_1, \dots, x_p, \dots, x_n) = x_p \cdot f(x_1, \dots, 0, \dots, x_n) \vee x_p \cdot f(x_1, \dots, 1, \dots, x_n) \quad (1,25)$$

Цю теорему легко довести розрахунком:

$$\text{I) } x_p = 0 \Rightarrow f(x_1, \dots, \bar{0}, \dots, x_n) = \bar{0} \cdot f(x_1, \dots, 0, \dots, x_n) \vee 0 \cdot f(x_1, \dots, 1, \dots, x_n) = f(x_1, \dots, 0, \dots, x_n)$$

тобто коли $x_p = 0$ пропозиція дійсна незалежно від значень інших змінних;

$$\text{B) } x_p = 1 \Rightarrow f(x_1, \dots, 1, \dots, x_n) = \bar{1} \cdot f(x_1, \dots, 0, \dots, x_n) \vee 1 \cdot f(x_1, \dots, 1, \dots, x_n) = f(x_1, \dots, 1, \dots, x_n),$$

тобто при $x_p = 1$ теорема справедлива незалежно від значень інших змінних, що означає, що теорема справедлива для будь-яких значень всіх змінних, які вимагали доведення.

Згідно з принципом двоїстості, ми отримуємо *теорему подвійного розкладання* :

$$f(x_1, \dots, x_p, \dots, x_n) = [\bar{x}_p \vee f(x_1, \dots, 1, \dots, x_n)] \cdot [x_p \vee f(x_1, \dots, 0, \dots, x_n)] \quad (1,26)$$

Теорема розширення являється зручним методом для перетворення логічних виразів, що містять операцію підсумовування по модулю 2, оскільки в багатьох випадках вона може звести цю операцію над функціями для простих операцій, приклад:

$$\begin{aligned} & \overline{x_2 \cdot x_1 \oplus (x_3 \vee x_1) \oplus x_3 \cdot x_1 \oplus (x_2 \vee x_1)} = \\ & = \bar{x}_1 \cdot [\overline{x_2 \cdot 0 \oplus (x_3 \vee 0) \oplus x_3 \cdot 0 \oplus (x_2 \vee 0)}] \vee x_1 \cdot [\overline{x_2 \cdot 1 \oplus (x_3 \vee 1) \oplus x_3 \cdot 1 \oplus (x_2 \vee 1)}] = \\ & = \bar{x}_1 \cdot (\overline{0 \oplus 1 \oplus 0 \oplus 1}) \vee x_1 \cdot (\overline{x_2 \oplus x_3 \oplus x_3 \oplus x_2}) = \bar{x}_1 \cdot 1 \vee x_1 \cdot 1 = \bar{x}_1 \vee x_1 = 1. \end{aligned}$$

Тотожності пов'язані з теоремою про розкладання:

$$\left. \begin{aligned} & \overline{x_p \cdot f(x_1, \dots, x_p, \dots, x_n)} = \bar{x}_p \cdot f(x_1, \dots, 0, \dots, x_n) \\ & x_p \cdot f(x_1, \dots, x_p, \dots, x_n) = x_p \cdot f(x_1, \dots, 1, \dots, x_n) \end{aligned} \right\} \quad (1,27)$$

За принципом подвійності цим тотожностям відповідають подвійні тотожності:

$$\left. \begin{aligned} & \bar{x}_p \vee f(x_1, \dots, x_p, \dots, x_n) = \bar{x}_p \vee f(x_1, \dots, 1, \dots, x_n) \\ & x_p \vee f(x_1, \dots, x_p, \dots, x_n) = x_p \vee f(x_1, \dots, 0, \dots, x_n) \end{aligned} \right\} \quad (1,28)$$

Ідентифікаційні дані є потужним інструментом для спрощення логічних виразів.

Нехай функція спрощена

$$f(v) = \overline{\overline{x_2 x_1} \oplus \overline{x_3 x_2} \oplus \overline{x_1 \vee x_3 x_2} \cdot \overline{x_2}}$$

Використовуючи першу тотожність (1.3.25) для x_2 , ми отримуємо:

$$f(v) = \overline{\overline{0 \cdot x_1} \oplus \overline{x_3 \cdot 0} \oplus \overline{x_1 \vee x_3} \cdot \overline{0 \cdot x_2}} = \overline{\overline{x_3} \oplus \overline{x_1 \vee x_3} \cdot \overline{x_2}}$$

для спрощення виразу $\overline{x_3} \oplus \overline{x_1 \vee x_3}$ ми можемо використати другу тотожність (1.28).

$$f(v) = \overline{\overline{0} \oplus \overline{x_1 \vee x_3} \cdot \overline{x_2}} = \overline{\overline{x_1 \vee x_3} \cdot \overline{x_2}} = \overline{\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}}$$

Первісні терміки. Змінні x_p та їх інверсії $\overline{x_p}$ називаються основними термінами, для яких використовується символічне позначення

$$x_p^{e_p} = \overline{e_p x_p} \vee e_p x_p = \overline{e_p} \oplus x_p \quad (1,29)$$

де $e_p = 0$ або 1 .

Це символічне позначення поєднує $x_p^{e_p}$ як базове x_p , так і $\overline{x_p}$. Дійсно, підставляючи значення 1 в (1.29), отримуємо $e_p = 0$

$$x_p^{e_p} = \begin{cases} \overline{x_p}, & \text{при } e_p = 0, \\ x_p, & \text{при } e_p = 1. \end{cases}$$

Лише шляхом введення цього символічного позначення можна формалізувати виведення загальних залежностей для функцій перемикавання. Очевидно, що два основні доданки $x_p^{e_p}$ і $x_p^{e'_p}$ рівні лише тоді, коли $e_p = e'_p$. До первинних термінів застосовуються такі залежності:

$$x_p^0 = \overline{x_p^1} = \overline{x_p};$$

$$\overline{x_p^{e_p}} = x_p^{\overline{e_p}} = \overline{x_p^{e_p}}; \quad (1,30)$$

$$\begin{aligned} x_p^{e_p} \cdot x_p^{\overline{e_p}} &= 0 \\ x_p^{e_p} \vee x_p^{\overline{e_p}} &= 1; \end{aligned} \quad (1,31)$$

$$x_p^{e_p} = \begin{cases} 0, & \text{якщо } x_p = \overline{e_p}, \\ 1, & \text{якщо } x_p = e_p. \end{cases} \quad (1,32)$$

Істинність цих співвідношень елементарно перевіряється на основі визначення первинних термінів (1.27).

Мінтерми та макстерми. Функція n змінних називається *мінтермом* (одиничним терміном).

$$K_i(v) = x_1^{e_1} \dots x_n^{e_n} = \prod_{p=1}^n x_p^{e_p} \quad (1,33)$$

де $v = (x_1, \dots, x_n)$, $e_p = 0$ або 1 , $i = e_1, \dots, e_n$. Це визначення означає, що існують 2^n різні мінтерми n змінних, оскільки існують 2^n різні n -розрядні двійкові числа $i = 0, 1, \dots, 2^n - 1$

Мінтерми мають такі властивості:

$$K_i(v) = \begin{cases} 1, & \text{якщо } v = v_i, \\ 0, & \text{якщо } v = v_j \neq v_i; \end{cases} \quad (1,34)$$

$$K_i(v) \cdot K_j(v) \equiv 0, \text{ якщо } i \neq j \quad (1,35)$$

$$\bigvee_{i=0}^{2^n-1} K_i(v) \equiv 1. \quad (1,36)$$

Властивість мінтермів (1.34), що будь-який мінтерм $K_i(v)$ дорівнює 1 лише в одному місці v_i області, що складається з 2^n крапок, можна легко довести за допомогою властивостей примітивних термів (1.32). Властивості (1.35) і (1.36) зтверджуються на основі властивості (1.34).

Давайте занотуємо всі мінімальні терми двох змінних x_1, x_2 :

$$K_0(v) = x_1^0 x_2^0 = \overline{x_1} \cdot \overline{x_2}; \quad K_1(v) = x_1^1 x_2^1 = x_1 \cdot x_2;$$

$$K_2(v) = x_1^0 x_2^1 = \overline{x_1} \cdot x_2; \quad K_3(v) = x_1^1 x_2^0 = x_1 \cdot \overline{x_2};$$

$v = (x_1, x_2)$ де Таким же чином ви можете написати будь-який мінтерм $K_i(v)$ з більшою кількістю змінних. Тож нехай $n = 4, i = 13$ напр.

$$K_{13}(v) = x_1^1 x_2^0 x_3^1 x_4^1 = x_1 \cdot \overline{x_2} \cdot x_3 \cdot x_4$$

Функція n змінних називається *maxterm* (нульовий член).

$$M_i(v) = \overline{K_i(v)} = \overline{\prod_{p=1}^n x_p^{e_p}} = \bigvee_{p=1}^n \overline{x_p^{e_p}}.$$

Відповідно до властивості примітивних доданків (1.30) можна записати

$$M_i(v) = \bigvee_{p=1}^n \overline{x_p^{e_p}}, \quad (1,37)$$

Де $v = (x_1, \dots, x_n), i = (e_1, \dots, e_n)$

Maxterms мають такі властивості:

$$M_i(v) = \begin{cases} 0, & \text{при } v = v_i, \\ 1, & \text{при } v = v_j \neq v_i; \end{cases}$$

$$M_i(v) \vee M_j(v) \equiv 0, \text{ при } i \neq j$$

$$\prod_{i=1}^{2^n-1} M_i(v) \equiv 0.$$

Для ясності в таблиці 1.5 (таблиці істинності) наведено всі міні- та максимальні терміни двох змінних x_1 і x_2 .

Таблиця 1.5 – Максимальний і мінімальний терміни двох змінних

I	x_1	x_2	K_0	K_1	K_2	K_3	M_0	M_1	M_2	M_3
0	0	0	1	0	0	0	0	1	1	1
1	1	0	0	1	0	0	1	0	1	1
2	0	1	0	0	1	0	1	1	0	1
3	1	1	0	0	0	1	1	1	1	0

Мінтерми (maxterms) - це функції, які приймають мінімальне (максимальне) значення зі значень своїх примітивних доданків $x_p^{e_p}$, тобто якщо хоча б один з примітивних доданків $x_p^{e_p}$ дорівнює 0(1), то мінтерм (maxterm) також дорівнює до 0(1).

Досконала диз'юнктивна нормальна форма (ДДНФ). Теорему розкладання (1.23) для функцій n змінних можна застосувати n разів, тобто функцію

можна розкласти на всі n змінних x_p . Як приклад розглянемо розкладання функції $f(x_1, x_2)$ двох змінних x_1 і x_2 .

Відповідно до теореми про розкладання (1.25) отримуємо

$$f(x_1, x_2) = \overline{x_2} \cdot f(x_1, 0) \vee x_2 \cdot f(x_1, 1).$$

Тоді кожну з функцій $f(x_1, 0)$ і $f(x_1, 0) \vee x_2 \cdot f(x_1, 1)$ можна розкласти за допомогою змінної x_1

$$\begin{aligned} f(x_1, x_2) &= \overline{x_2} \cdot [\overline{x_1} \cdot f(0, 0) \vee x_1 \cdot f(1, 0)] \vee x_2 \cdot [\overline{x_1} \cdot f(0, 1) \vee x_1 \cdot f(1, 1)] = \\ &= \overline{x_1} \cdot \overline{x_2} \cdot f(0, 0) \vee x_1 \cdot \overline{x_2} \cdot f(1, 0) \vee \overline{x_1} \cdot x_2 \cdot f(0, 1) \vee x_1 \cdot x_2 \cdot f(1, 1) = \\ &= \bigvee_{i=0}^3 x_1^{e_i} x_2^{e_2} \cdot f(e_1, e_2) = \bigvee_{i=0}^3 f(v_i) K_i(v), \end{aligned}$$

де $v = (x_1, x_2)$, $v_i = (e_1, e_2)$, $i = e_1 e_2$, $K_i(v) = x_1^{e_1} x_2^{e_2}$ - мінтерми двох змінних x_1, x_2 .

Оскільки $f(v_i) = a_i = 0$ або 1 (значення функції в точці v_i , то $f(v) = \bigvee_{i=0}^3 a_i K_i(v)$.

Такий вид представлення функції двох перемінних називається ДДНФ. Розподіл функцій n змінних представлятиме альтернативу 2^n елементів форми

$$x_1^{e_1} \dots x_p^{e_p} \dots x_n^{e_n} f(e_1, \dots, e_p, \dots, e_n) = f(v_i) K_i(v) = a_i K_i(v)$$

$$f(v) = \bigvee_{i=0}^{2^n-1} a_i K_i(v) \quad (1,38)$$

Вираз (1.38) показує функцію ДДНФ n змінних. Тому значення функції $a_i = 0$ або 1 , то $a_i K_i(v) = 0$, якщо $a_i = 0$, і $a_i K_i(v) = K_i(v)$, якщо $a_i = 1$. Отже функцію ДДНФ можна представити у вигляді

$$f(v) = \bigvee_{i_s} K_{i_s}(v) \quad (1,39)$$

де i_s номери вибраних точок, де функція $f(v)$ прирівняна 1 .

Як примір розглянемо $f(v)$ функцію 3 змінних, подану в таблиці істинності (табл. 1.6).

Таблиця 1.6 – Таблиця істинності функцій трьох змінних

I	x_1	x_2	x_3	$f(v)$
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Ця таблиця показує це

$$a_0 = a_3 = a_4 = a_6 = 0, \quad a_1 = a_2 = a_5 = a_7 = 1$$

отже, згідно (1.39)

$$f(v) = K_1(v) \vee K_2(v) \vee K_5(v) \vee K_7(v) = \bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee \bar{x}_1 \cdot x_2 \cdot \bar{x}_3 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \vee x_1 \cdot x_2 \cdot x_3.$$

Це досконала кон'юнктивно нормальна форма (ДКНФ) цієї функції. Такий вигляд функції n перемінних можна отримати, виходячи з теореми подвійного розподілу (1.26). Однак ДКНФ можливо отримати простіше, написавши обернену функцію ДКНФ $\overline{f(v)}$. Інвертування функції в будь-якій точці v_i повинна мати зворотні значення у відповідності до значення a_i самої функції, тобто $\overline{f(v_i)} = \bar{a}_i$ якщо $f(v_i) = a_i$.

На основі (1.38) запишемо ДДНФ оберненої функції

$$\overline{f(v)} = \bigvee_{i=0}^{2^n-1} a_i K_i(v).$$

З цього співвідношення випливає закон подвійності

$$f(v) = \overline{\bigvee_{i=0}^{2^n-1} a_i K_i(v)} = \prod_{i=0}^{2^n-1} \overline{a_i K_i(v)} = \prod_{i=0}^{2^n-1} [a_i \vee \overline{K_i(v)}].$$

З визначення макстермів випливає, що

$$f(v) = \prod_{i=0}^{2^n-1} [a_i \vee M_i(v)]. \quad (1,40)$$

Така форма представлення функції 2-х змінних називаємо ДКНФ.

Тому, що значення функції $a_i = 0$ дорівнює 1, то $a_i \vee M_i = M_i(v)$, якщо $a_i = 0$ і $a_i \vee M_i = 1$, якщо $a_i = 1$. Тому ДКНФ можна подати на формі

$$f(v) = \prod_{i_s} M_{i_s}(v), \quad (1,41)$$

де i_s номери точок, де функція $f(v)$ дорівнює 0.

визначення значення функції на основі (1.41): $a_0 = a_3 = a_4 = a_6 = 0$

$$f(v) = M_0 \cdot M_3 \cdot M_4 \cdot M_6 = (x_1 \vee x_2 \vee x_3) \cdot (\overline{x_1} \vee x_2 \vee x_3) \cdot (x_1 \vee \overline{x_2} \vee x_3) \cdot (x_1 \vee x_2 \vee \overline{x_3}).$$

Це функція ДКНФ.

Ідеальні нормалізовані форми в основах І-НІ та АБО-НІ. Набір елементарних функцій, при допомозі, яких можна записати любую функцію, $f(v)$ називають *функціонально повною системою функцій або базисом*. З виразів (1.39) і (1.41) випливає, що $f(v)$ для представлення будь-якої функції в ДДНФ і ДКНФ можна використовувати тільки функції (операції) І, АБО і НЕ, тобто сукупність цих функцій є основою.

Перетворимо функцію ДДНФ (1.38) при допомозі закону подвійного заперечення та закону двоїстості

$$f(v) = \overline{\bigvee_{i=0}^{2^n-1} a_i K_i(v)} = \overline{\prod_{i=0}^{2^n-1} a_i K_i(v)}. \quad (1,42)$$

Ця форма подання функції називається *ідеальною нормальною формою* (PNF) у базі даних І-НІ, оскільки вона вимагає використання лише функцій (операцій) І-НІ.

Перетворимо тепер функцію ДКНФ (1.40), використовуючи закон подвійного заперечення та закон подвійності

$$f(v) = \overline{\prod_{i=0}^{2^n-1} [a_i \vee M_i(v)]} = \overline{\bigvee_{i=0}^{2^n-1} a_i \vee M_i(v)}. \quad (1.43)$$

Ця форма подання функції є ДНФ на основі АБО-НІ, тому, що вона вимагає використання лише функцій АБО-НІ.

На основі (1.42) і (1.43) з ДДНФ і ДКНФ функції, наведеної в таблиці 1.6, можна отримати DFNF цієї функції в базах даних І-НІ та АБО-НІ:

$$f(v) = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_1 \cdot x_2 \cdot x_3}$$

$$f(v) = \overline{x_1 \vee x_2 \vee x_3 \vee x_1 \vee x_2 \vee x_3 \vee x_1 \vee x_2 \vee x_3 \vee x_1 \vee x_2 \vee x_3}$$

Ці дві форми є основними в логічній базі.

Прикладами інших функціонально завершених базових систем НЧ є:

Базис Жегалкіна: інверсія, сума за модулем 2, константа 1;

Основа заборони: константа 1, дія «заборона», інверсія;

База імплікації: константа 0, операція імплікації, інверсія ;

Мажоритарний базис: інверсія, мажоритарні операції, константи 0 і 1.

Основи еквівалентності: інверсія, альтернатива, константа 0, еквівалентність;

Основа Пірса: операція Пірса;

База даних Шеффера: Операція Шеффер

Останні два базиси, кожен з яких складається з однієї функції, називаються універсальними базисами. Застосовуються коефіцієнти:

$$\overline{x_2 \cdot x_1} = x_2 / x_1; \quad \overline{x_2 \vee x_1} = x_2 \downarrow x_1;$$

$$\overline{x_2 \vee x_1} = \overline{x_2} \vee \overline{x_1}; \quad \overline{x_2 \vee x_1} = \overline{x_2} \cdot \overline{x_1}.$$

За основу взято логічні бази МДФН і МКНФ. Від будь-якої мінімальної форми можна перейти до універсальної основи Шиффера і Пірса.

Оптимальний перехід вважається наступним:

Основа МДФН Шеффера, →

База даних Peirce МКНФ. →

Вибір переходу залежить від типу функції. Алгоритм переходу заснований на використанні правил де Моргана та подвійної інверсії (рис. 1.8).

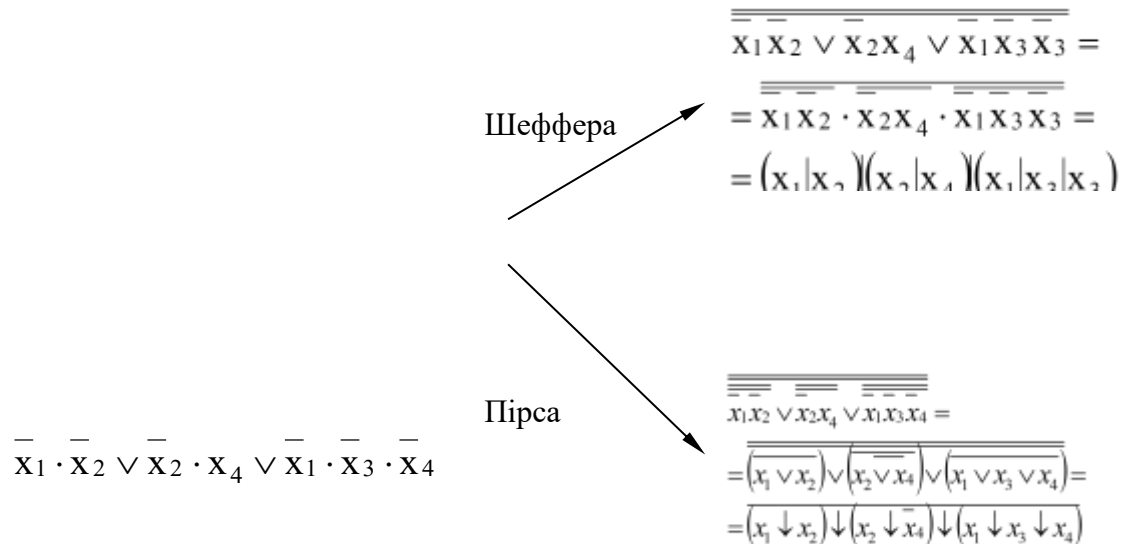


Рис. 1.8 Приклад переходу до універсального базису Шеффера і Пірса

1.3 Мінімізація комутаційних функцій

1.3.1. Метод алгебраїчної мінімізації

Побудова логічних діаграм на основі логічних функцій, представлених у ДНФ або ДКНФ, складених безпосередньо з таблиць істинності, у більшості випадків недоцільна. Після складання структурної формули є можливість перевести її в спрощений вигляд, що призводить до меншого споживання логічних елементів на схемі. При проектуванні обладнання намагаються реалізовувати конструктивні схеми, що забезпечують мінімальний знос обладнання та мінімальну вартість за умови забезпечення певного рівня надійності.

Мінімізація логічної функції найчастіше розуміється як знаходження її найпростішого представлення з мінімальною кількістю входів у нього змінні. Після мінімізації, зрозумілої таким чином, можна виконувати інші

перетворення функції, спрямовані на зменшення кількості типів елементів, приведення функції до вигляду, зручного для реалізації на заданих елементах, наприклад, на елементах І-НІ тощо.

В алгебраїчному методі мінімізації булевих функцій виконуються такі операції. У ДДНФ представлена булева функція, здійснюється пошук суміжних компонентів (мінтермів) і виконується їх склеювання. Для зручності склеювання при необхідності додайте компоненти, які вже присутні в оригінальному конструктивному малюнку. Після будь-якого склеювання робиться спроба видалити надлишок інгредієнтів за допомогою закону поглинання. Результат не є ні надмірністю, ні глухим кутом. Варто зазначити, що виключення термів може здійснюватися різними способами, в результаті чого булева функція може мати кілька форм тупикових ситуацій. Мінімальні форми також є тупиковими. Тому процес пошуку мінімальних форм зводиться до отримання всіх сліпих форм заданої функції та вибору серед них форм із мінімальною кількістю букв. Потім виконуються перетворення на основі теореми де Морган і стане генералом змінні в дужках.

приклад. Знайти сліпі форми функцій $F = \bar{A}BC + A\bar{B}C + ABC$. У цій функції до мінтерм можна застосувати закон склеювання $\bar{A}BC$ і $A\bar{B}C$ (склеювання зі змінною B) або $\bar{A}BC$ і $A\bar{B}C$ (склеювання зі змінною A). У першому випадку ми отримуємо це

$$F = \bar{A}BC + AC(\bar{B} + B) = \bar{A}BC + AC = C(\bar{A}B + A)$$

у другому випадку

$$F = BC(\bar{A} + A) + A\bar{B}C = BC + A\bar{B}C = C(B + A\bar{B}).$$

Ви можете використати властивість 1.6 булевих функцій (див. таблицю 1.3) і додати термін ABC до вихідного виразу функції F.

$$F = \bar{A}BC + ABC + A\bar{B}C + ABC$$

Згрупувавши доданки, отримуємо

$$F = (\overline{ABC} + ABC) + (A\overline{BC} + ABC) = BC(\overline{A} + A) + AC(\overline{B} + B)$$

Після склеювання

$$F = C(B + A)$$

все три форми логічної функції є тупиковими. Остання форма логічної функції є найпростішою з двох попередні були отримані виключно шляхом склеювання мінтерм . У цій формі виконуються тільки дві операції - логічне додавання і множення, при цьому в обох формах логічних операцій більше.

Розглянуті алгоритми сліпого пошуку форми досить трудомісткі , оскільки при виконанні операції склеювання необхідно порівняти всі можливі пари членів вихідного виразу . Існують методи спрощення пошуку сліпих форм булевих функцій . Для логічного проектування мікроелектронних пристроїв зручний метод карт Мінтерма – Вейха і Карно .

1.3.2. Мінімізація булевих функцій за допомогою карт термінів

Однією з основних проблем, що виникають при створенні комбінаційних схем, є мінімізація комутаційних функцій, які виконують ці комбінаційні схеми. Чим простіше логічний вираз, тим простіше і дешевше комбінаційна схема, яка його реалізує.

Функції перемикавання можна спростити за допомогою законів і тотожностей логічної алгебри. Такий метод мінімізації є трудомістким, тому в більшості випадків можна використовуються графічні засоби мінімізації (діаграми Віча, діаграми Венна, карти Карно). Перевагою даних методів є прозорість і простота використання при малій кількості перемінних ($n \leq 4$).

У 1953 р. М Карно публікує статтю про створену ним систему для графічного представлення та упрощення функцій переключання. Карта Карно зображена на рис. 1.9. Тут і надалі приймемо положення $f(v) = y$.

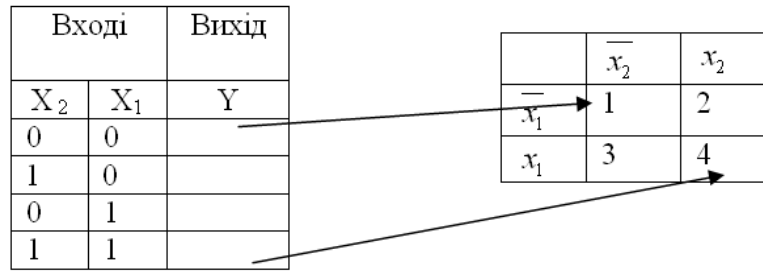


Рис. 1.9 Позначення квадратів на карті Карно

4-ри квадрати (1, 2, 3, 4) відповідає чотирьом комбінаціям x_1, x_2 у таблиці істинності з двома змінними. На цьому малюнку квадрат 1 відповідає продукту $\overline{x_1} \cdot \overline{x_2}$, квадрат 2 - $\overline{x_1} \cdot x_2$ і так далі.

Тепер припустімо, що вам потрібно створити карту Карно для функції переключення, що зберігається в ДДНФ: $\overline{x_1} \cdot x_2 \vee x_1 \cdot \overline{x_2} \vee x_1 \cdot x_2 = y$.

Розмістимо логічні «1» у всіх квадратах, що відповідають добутку вихідної функції переключення на рис. 1.10 Готова таким чином карта Карно готова до побудови. Суміжні агрегати об'єднуються в один контур у групи по два, чотири або вісім агрегатів.

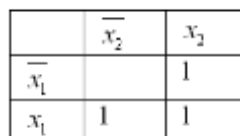


Рис. 1.10 Нанесення одиниць на карту Карно

Побудова контурів продовжується, доки всі одиниці не розмістяться всередині контуру. Кожний контур є новим членом спрощеної залежності перемикачів. Зверніть увагу, що на рис. 1.10 отримано лише два контури. Це значить, що нова проста функція перемикачів складатиметься лише з двох елементів АБО. Давайте тепер спростимо функцію перемикачів, розглянувши дві схеми на рис. 1.11.

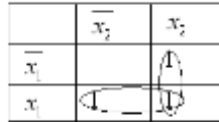


Рис. 1.11. З'єднання агрегатів у групи в одне коло.

Беручи нижній контур, ми помічаємо, що x_1 він з'являється тут у поєднанні з x_2 і $\overline{x_2}$. Згідно з правилами алгебри логіки x_2 , і $\overline{x_2}$ є додатковими один до одного і можуть бути опущені.

Отже в контурі нижче залишається лише термін x_1 . Подібним чином вертикально розміщений контур містить x_1 і $\overline{x_1}$, які також можна опустити, залишивши лише x_2 . Решта елементів x_1 і x_2 поєднується з функцією АБО, що дає спрощену функцію перемикачів $x_1 \vee x_2 = y$.

Алгоритми мінімізації перемикачів функцій записується як:

1. Перенесення функції перемикачів на ДДНФ.
2. Зображення одиниць на картах Карно.
3. З'єднання суміжних блоків, що охоплюють два, чотири чи вісім квадратів.
4. Спрощення з виключенням додаткових смуг всередині контуру.
5. Підключення решти елементів (по одному в кожному ланцюзі) за допомогою функції АБО.
6. Збережіть мінімізовану функцію перемикачів в ДДНФ.

Карти Карно з 3-ма змінними.

Подивимося на функцію перемикача

$$x_1 \cdot \overline{x_2} \cdot x_3 \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \vee x_1 \cdot x_2 \cdot x_3 = y.$$

Карту Карно для цієї функції показано на рисунку 1.12.

Нижній контур містить x_2 і $\overline{x_2}$, їх можна опустити. Тоді в складі нижнього контуру залишається тільки член $x_1 \cdot \overline{x_3}$. Верхній контур містить x_3 і $\overline{x_3}$, які також

видаляються, залишаючи лише термін $\overline{x_1} \cdot \overline{x_2}$. Остаточна функція перемикання має вигляд $\overline{x_1} \cdot \overline{x_3} \vee \overline{x_1} \cdot \overline{x_2} = y$.

Важливо, щоб карта Карно була намальована точно так, як показано на малюнку 1.12. Зауважте, що коли ви рухаєтесь ліворуч вниз по карті, на кожному кроці змінюється лише одна змінна. Товар пишеться у верхньому лівому куті $\overline{x_1} \cdot \overline{x_2}$ та в рядку нижче $\overline{x_1} \cdot x_2$ (замінник $\overline{x_2}$) x_2 . Потім при русі $\overline{x_1} \cdot x_2$ вниз $x_1 \cdot x_2$ вона змінюється на x_1 тощо. Якщо картку Карно взято неправильно, вона не дасть очікуваного результату $\overline{x_1}$.

	$\overline{x_3}$	x_3
$\overline{x_1} \cdot \overline{x_2}$	1	1
$\overline{x_1} \cdot x_2$		
$x_1 \cdot x_2$	1	
$x_1 \cdot \overline{x_2}$	1	

	$\overline{x_3}$	x_3
$\overline{x_1} \cdot \overline{x_2}$	1	1
$\overline{x_1} \cdot x_2$		
$x_1 \cdot x_2$	1	
$x_1 \cdot \overline{x_2}$	1	

Рис. 1.12 Спрощення функції перемикання на основі карти Карно

Карты Карно з чотирма змінними. Таблиця істинності для чотирьох змінних містить 16 можливих комбінацій. Розглянемо функцію перемикача

$$\overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 = y.$$

Карта Карно для функції перемикання з чотирма змінними допускає 16 можливих комбінацій x_1, x_2, x_3, x_4 (рис. 1.13).

	$\overline{x_3} \overline{x_4}$	$\overline{x_3} x_4$	$x_3 \overline{x_4}$	$x_3 x_4$
$\overline{x_1} \overline{x_2}$		1	1	
$\overline{x_1} x_2$		1	1	
$x_1 \overline{x_2}$				
$x_1 x_2$	1	1		

	$\overline{x_3} \overline{x_4}$	$\overline{x_3} x_4$	$x_3 \overline{x_4}$	$x_3 x_4$
$\overline{x_1} \overline{x_2}$		1	1	
$\overline{x_1} x_2$		1	1	
$x_1 \overline{x_2}$				
$x_1 x_2$	1	1		

Рис. 1.13. Мінімізація функції перемикання чотирьох змінних

Ці комбінації представлені 16 квадратами карти відповідно. Помістимо на карту 6 одиниць, що відповідають 6 елементам даної функції перемикання. Групи з двох і чотирьох одиниць з'єднані контурами. Нижній контур двох блоків дозволяє спускатися x_4 та $\overline{x_4}$. Тоді учасник залишається в ньому $x_1 \cdot \overline{x_2} \cdot \overline{x_3}$. Крім того, у верхньому контурі чотири одиниці, x_3 і $\overline{x_3}$, x_4 і $\overline{x_4}$, відкидаються парами, так що верхній контур призводить до члена $\overline{x_1} \cdot x_4$. Спрощена функція перемикання в DDNF має вигляд $x_1 \cdot \overline{x_2} \cdot \overline{x_3} \vee \overline{x_1} \cdot x_4 = y$.

Інші типи карт Карно.

Розглянемо функцію перемикача

$$x_1 \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} \vee x_1 \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} = y.$$

Його чотири члени представлені на карті Карно чотирма одиницями (рис. 1.14).

	$\overline{x_3} \overline{x_4}$	$\overline{x_3} x_4$	$x_3 \overline{x_4}$	$x_3 x_4$
$\overline{x_1} \overline{x_2}$				
$\overline{x_1} x_2$	1			1
$x_1 \overline{x_2}$	1			1
$x_1 x_2$				

Рис. 1.14. Спрощення функції перемикання на основі карти Карно

На цьому ж малюнку показано правильний спосіб побудови контуру. Зауважте, що в цьому випадку доданки $x_1 \bar{x}_3$ і $x_1 x_3$ попарно опущені \bar{x}_3 . Спрощена функція перемикання виглядає так $x_2 \cdot \bar{x}_4 = y$.

Інші способи побудови схеми ми покажемо на прикладі комутаційних функцій (рис. 1.15 і рис. 1.16 відповідно):

$$\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \cdot x_4 = y,$$

$$\bar{x}_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee x_1 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot \bar{x}_4 \vee \bar{x}_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 \vee x_1 \cdot \bar{x}_2 \cdot x_3 \cdot \bar{x}_4 = y$$

Тому мінімізовані функції виглядатимуть так:

$$\bar{x}_2 \cdot \bar{x}_3 = y,$$

$$\bar{x}_2 \cdot \bar{x}_4 = y.$$

	$\bar{x}_3 \bar{x}_4$	$\bar{x}_3 x_4$	$x_3 x_4$	$x_3 \bar{x}_4$
$\bar{x}_1 \bar{x}_2$	1	1		
$\bar{x}_1 x_2$				
$x_1 \bar{x}_2$				
$x_1 x_2$	1	1		

	$\bar{x}_3 \bar{x}_4$	$\bar{x}_3 x_4$	$x_3 x_4$	$x_3 \bar{x}_4$
$\bar{x}_1 \bar{x}_2$	1			1
$\bar{x}_1 x_2$				
$x_1 \bar{x}_2$				
$x_1 x_2$	1			1

Рис. 1.15. Перший варіант

Рис. 1.16. Другий варіант спрощення

1. 3.3. Інтегральні цифрові мікросхеми є логічними елементами

Логічний елемент — електронний пристрій, що виконує одну з логічних функцій. Серії розглянутих мікросхем включають велику кількість логічних елементів. На схемі логічний елемент зображується прямокутником, усередині якого розміщено зображення покажчика функції. Лінії зліва від прямокутника

показують входи, а праворуч — вихід елемента. На малюнку 1.17 показані основні логічні елементи, які використовуються в цифрових пристроях:

Елемент І (роз'єм): $y = x_1 \cdot x_2$ (а);

Елемент АБО (непересічний): $y = x_1 \vee x_2$ (б);

елемент NO (інвертор 1): $y = \bar{x}$ (в).

Крім перерахованих вище, існує ряд логічних елементів, які виконують більш складні логічні перетворення. Ці перетворення являють собою комбінації найпростіших логічних операцій. До таких елементів відносяться:

Елемент І-НІ: $y = \overline{x_1 \cdot x_2}$;

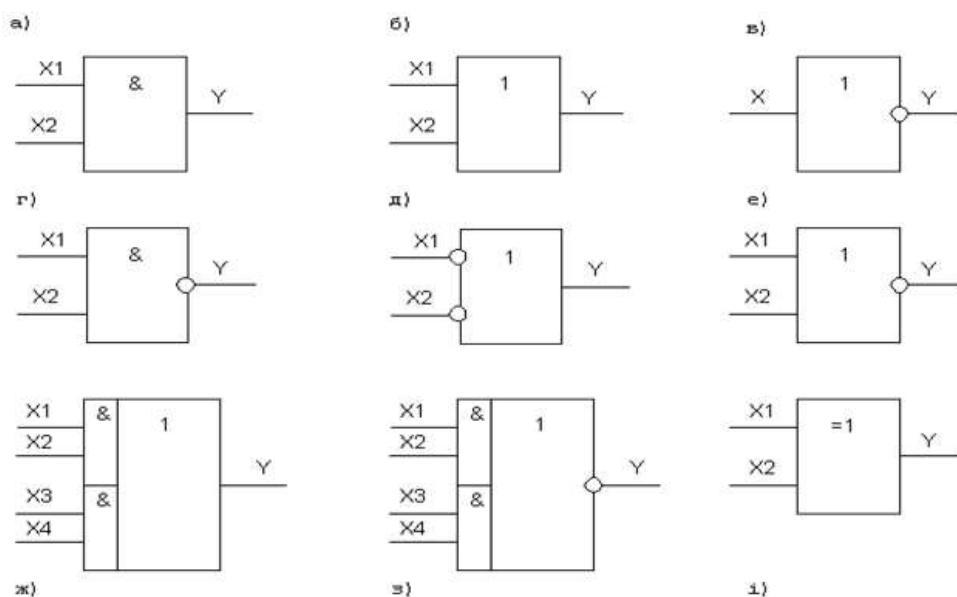
елемент АБО-НІ: $y = \overline{x_1 \vee x_2}$;

Елемент І-АБО: $y = x_1 \cdot x_2 \vee x_3 \cdot x_4$;

Елемент І-АБО-НІ: $y = \overline{x_1 \cdot x_2 \vee x_3 \cdot x_4}$;

додавання за модулем 2: $y = \overline{x_1 \cdot x_2} \vee \overline{x_1 \cdot x_2}$.

Кількість входів до логічних елементів для різних цілей може відрізнятися, але входи кожного елемента еквівалентні. Деякі з них можуть не використовуватися при роботі на певних пристроях. Входи, які не використовуються в схемах І, І-НІ, підключаються до +U_j , а на діаграмах АБО, АБО-НІ суматор по модулю 2 - із загальним проводом (0 В).



1.3.4. Синтез схеми процесора на основі заданої логічної функції

Основні етапи синтезу цифрових пристроїв.

Перший етап:

- а) Усний виклад і формулювання задачі синтезу.
- б) Дано основу, на якій має бути побудована діаграма.
- с) Встановлені умови експлуатації та обмеження.

Етап другий: Отримання таблиці (або таблиць) істинності - таблиці роботи пристрою.

Третій етап: Запис логічної функції в аналітичній формі завжди базується на принципі Буля,

мінімізація відомої функції за допомогою діаграм Вейча або карт Карно.

Четвертий етап: Перехід до заданого базису та побудова діаграми.

При побудові діаграми враховуються обмеження деяких параметрів:

- кількість входів логічних елементів,
- глибина контуру (впливає на затримку сигналу),
- уникнення шаблонних перегонів та інших.

В інженерній практиці кожен етап іноді вимагає творчого підходу і тому може бути доповнений допоміжними етапами (підетапами).

Може статися, наприклад, що логічні функції залежать від змінних, які мають більше 4. Діаграми Вейча не дуже допомагають отримати мінімальну форму. Тут може бути корисним етап декомпозиції: поділ пристроїв на окремі менш складні вузли. Іноді може бути корисно трохи ускладнити схему, щоб можна було виділити спільні вузли, для яких схеми вже відомі.

На основі отриманої логічної формули можна побудувати комбінаційну схему з елементів АБО, І, НЕ, яка матиме задані властивості, тобто реалізуватиме задану булеву функцію. Логічні елементи, призначені для виконання операцій, заданих у формулі, розміщуються на схемі, починаючи з входів, у

тому ж порядку, в якому виконуються логічні операції. Спочатку показано інвертори для отримання вхідних змінних, потім елементи І, а потім елемент АБО. Відповідно до формули до елементів додаються або вхідні сигнали, або результати логічних операцій над цими сигналами з виходів інших елементів.

Приклад . Для булевої функції $F = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$ синтезуємо схему, зображену на рис. 1/18.

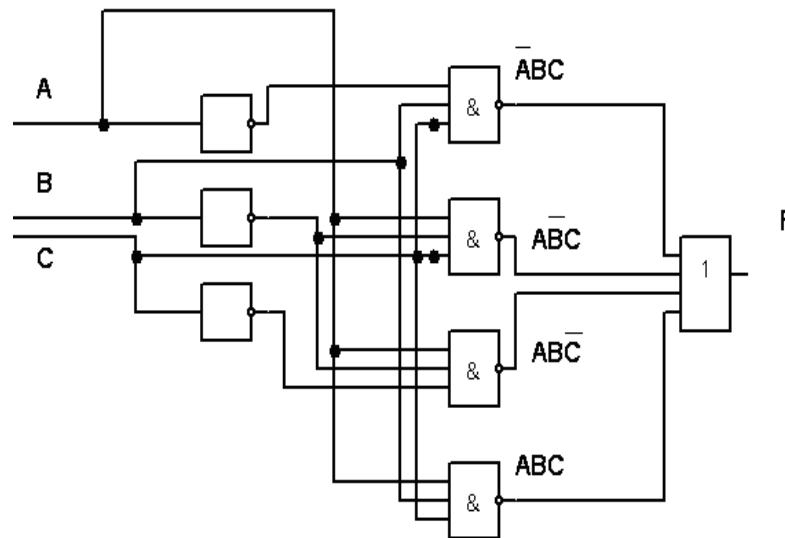


Рис. 1.18 Схема процесора, синтезованого за допомогою булевої функції

1.3.5 Методи спрощення логічних задач

Виробники інтегральних схем полегшили вирішення багатьох проблем, пов'язаних з побудовою комбінаційних логічних схем, створивши т.зв. *селектори даних* , використання яких часто дозволяє вирішити складну логічну задачу за допомогою лише однієї інтегральної схеми. Селектор даних складається з великої кількості логічних елементів, розміщених в одній інтегральній схемі.

Селектор даних «1 з 8» показаний на малюнку 1.19. Зверніть увагу на наявність восьми входів інформації з лівого боку селектора , пронумерованих від 0 до 7, і трьох входів селектора в нижній частині селектора даних, позначених А, В і С.

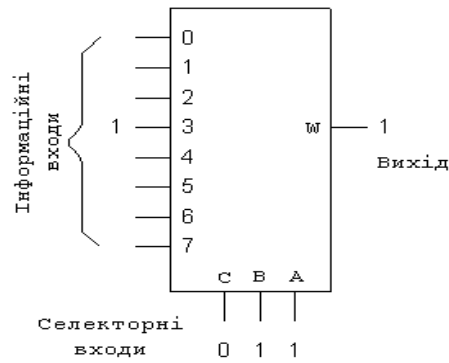


Рис. 1.19. Конвенція селектора даних "1 із 8".

Основним завданням селектора даних є передача даних з певного входу на вихід W вибір входу, з якого передаються дані, визначається двійковим кодом, що подається на входи селектора (рис. 1.20). Селектор даних працює за тим же принципом, що і поворотний перемикач. Для зміни входу передачі даних змінюється двійковий код на входах селектора.

Наведемо приклад побудови логічних схем. Наведемо спрощену функцію перемикання:

$$y = \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \cdot x_4 \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot \overline{x_2} \cdot x_3 \cdot x_4 \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot \overline{x_3} \cdot x_4 \vee \overline{x_1} \cdot x_2 \cdot x_3 \cdot \overline{x_4} \vee \overline{x_1} \cdot x_2 \cdot x_3 \cdot x_4.$$

Покажемо на рис. 1.21 діаграма, що відповідає цій функції перемикання. Якщо для побудови цієї функції використовувалися стандартні мікросхеми, їх знадобилося б 6-9. Таке рішення було б дуже марнотратним. При меншій вартості це завдання вирішується за допомогою селектора даних. Для пояснення показана на рис. 2.38 функція перемикання записана у вигляді таблиці істинності. До цієї таблиці додано селектор даних "1 із 16". Зверніть увагу, що логічні 0 і 1 застосовуються до шістнадцяти інформаційних входів селектора відповідно до стовпця вихідних значень у таблиці істинності. Ці зв'язки залишаються постійними для даної таблиці істинності. На входи селектора (D, C, B і A) подаються двійкові числа, що відповідають різним комбінаціям входів у таблиці істинності. Наприклад, якщо двійкове число 0000 застосовано до входів селектора D, C, B, A, тоді логічна 1 буде призначена виходу W селектора даних.

Перший рядок таблиці істинності вимагає, щоб на виході W з'явилася логічна одиниця, коли всі змінні D, C, B і A приймають значення 0. Якщо на вході селектора D надходить двійкове число 0001, B і A, C , то відповідно до вимог таблиці істинності результатом W буде логічний 0. Будь-яка комбінація вхідних змінних D, C, B і A дасть необхідний результат відповідно до таблиці істинності.

Використання селектора даних є простим, зручним і ефективним методом вирішення задач побудови комбінаційних логічних схем. Селектори даних також називають *мультиплексорами*.

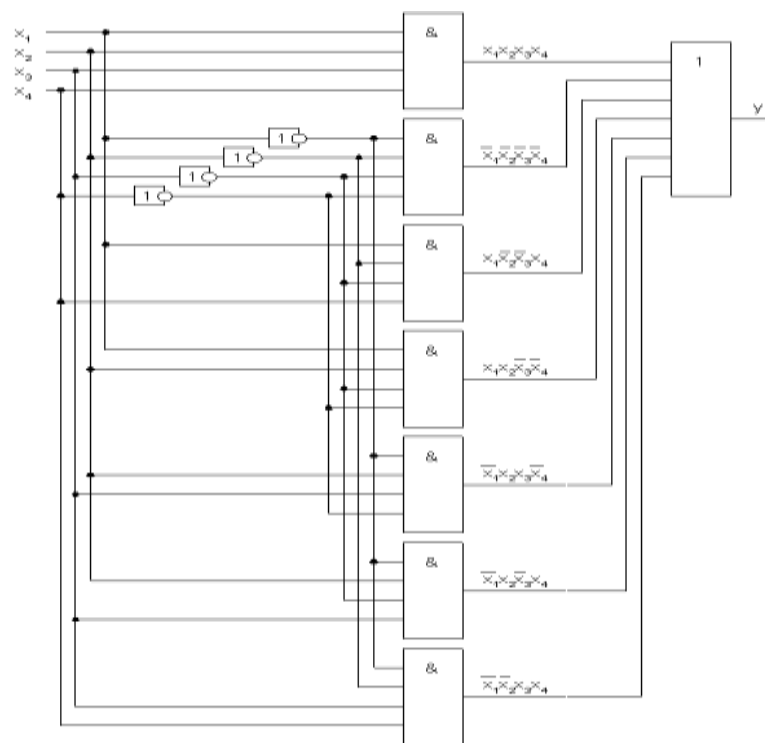


Рис. 1.20. Логічна схема функції перемикання

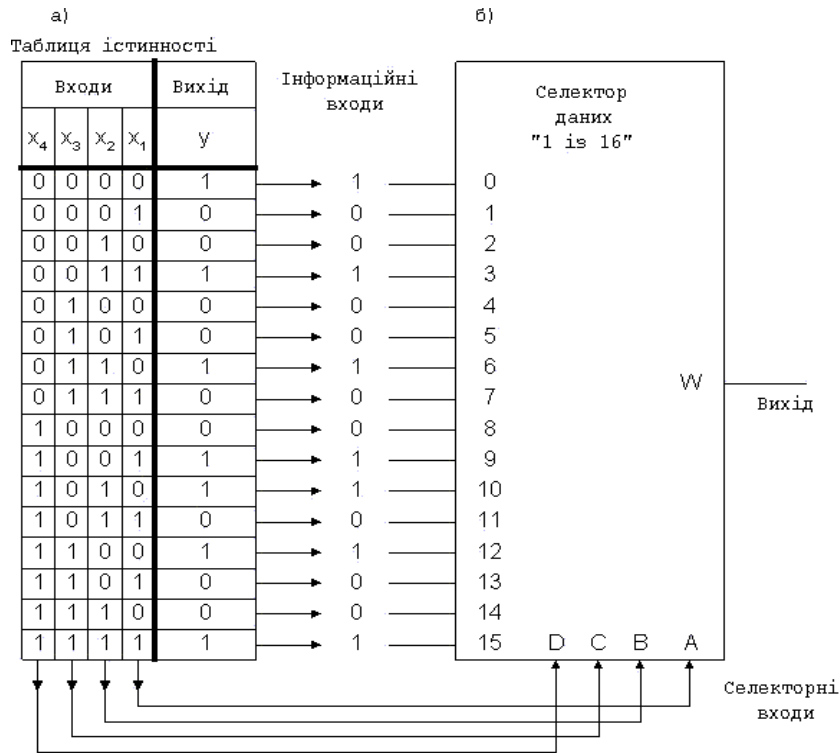


Рис. 1.21. Розв'язування логічної задачі
за допомогою селектора даних

РОЗДІЛ 2. ЛОГІЧНІ ЕЛЕМЕНТИ ЦИФРОВОЇ ТЕХНІКИ

2.1 Поняття комбінаційної схеми та цифрової машини

Кожен інформаційний символ в цифрових пристроях кодується двійковим алфавітом, тому сигнали можуть приймати тільки два значення: високий або низький рівень напруги, наявність або відсутність імпульсу напруги. Умовою цього є здатність елементів цифрової схеми достовірно розпізнавати два значення сигналу, що відповідають символам 0 і 1, в умовах зміни температури навколишнього середовища, напруги джерела живлення і впливу інших дестабілізуючих факторів.

Ми вже говорили, що значення рівня сигналу (U_c) приймаються елементами цифрових пристроїв не безперервно, а в дискретні моменти часу, інтервал яких називається робочим циклом T . Як правило, одне елементарне пере-

творення отримані кодові слова відбуваються за один робочий цикл на цифровому вході пристроїв. Часова дискретність забезпечується спеціальними керуючими пристроями, які генерують *імпульси синхронізації* (SI). У дискретних пристроях використовується два способи подання інформації: потенціальний і імпульсний. У потенціальному методі значення логічного 0 і логічної 1 відповідають низьким і високим напругам. Якщо напруга низького рівня відповідає логічному нулю, а напруга високого рівня відповідає логічній одиниці, то така логіка називається позитивною, і навпаки, якщо напруга високого рівня приймається за логічний нуль, а напруга низького рівня приймається за логічну одиницю, то така логіка називається негативною. Надалі ми будемо розглядати лише пристрої з позитивною логікою.

Потім в основному використовуються терміни «висока і низька напруга» (сигнали високого і низького рівня), що відповідають рівням логічної 1 і логічного 0.

Інформація в цифрових пристроях може бути представлена в послідовному і паралельному кодах. При використанні послідовного коду кожен годинник відповідає одній цифрі двійкового коду. Кількість цифр визначається числом годинника, який відраховується від годинника, що збігається з початком подання коду.

Графіки, зображені на рис. 2.1 проілюструємо код послідовності двійкового байтового числа 10011011 з потенційним та імпульсним способами подання інформації. У першому способі (рис. 2.1, а) сигнал зберігає низький або високий рівень протягом одного або кількох циклів. Коли сигнал переходить з одного рівня на інший, його значення є невизначеним. При імпульсному способі подання цифрової інформації (рис. 2.1, б) наявність або відсутність імпульсу кінцевої тривалості відповідає одиничному і нульовому значенню двійкової змінної. Завдяки порядковому номерному коду всі його цифри можна помістити в один елемент і відправити по одному каналу передачі інформації. Для передачі всього числа потрібно вісім циклів (рис. 2.1, в).

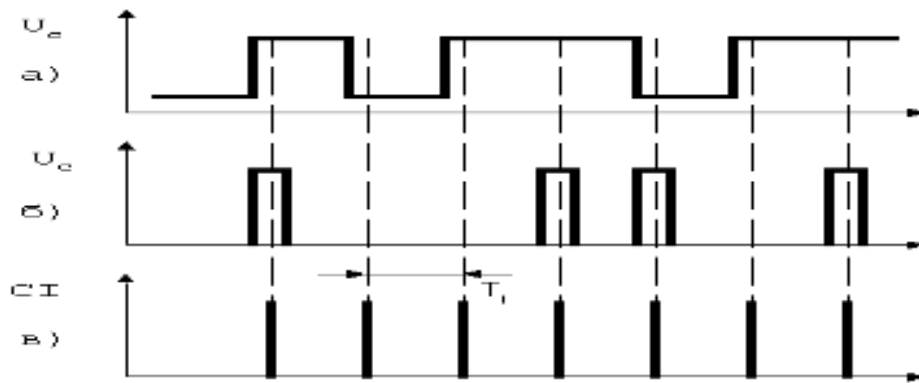


Рис. 2.1. Послідовний код двійкового числа при потенціалі і імпульсивні способи подачі інформації

Паралельний код дозволяє істотно скоротити час обробки і передачі інформації. Наприклад, на рис. 2.2 зображено паралельний код семизначного числа 1101101. У цьому випадку як при імпульсному (рис. 2.2, а), так і при потенційному (рис. 2.2, б) способах подання інформації всі розряди двійковий код представлений в одному часовому циклі і може визначатися окремими елементами і передаватися по окремих каналах (бітових шинах). Цифрові пристрої, які обробляють і перетворюють інформацію, що надходить, у вхідні дані, називають цифровими машинами.

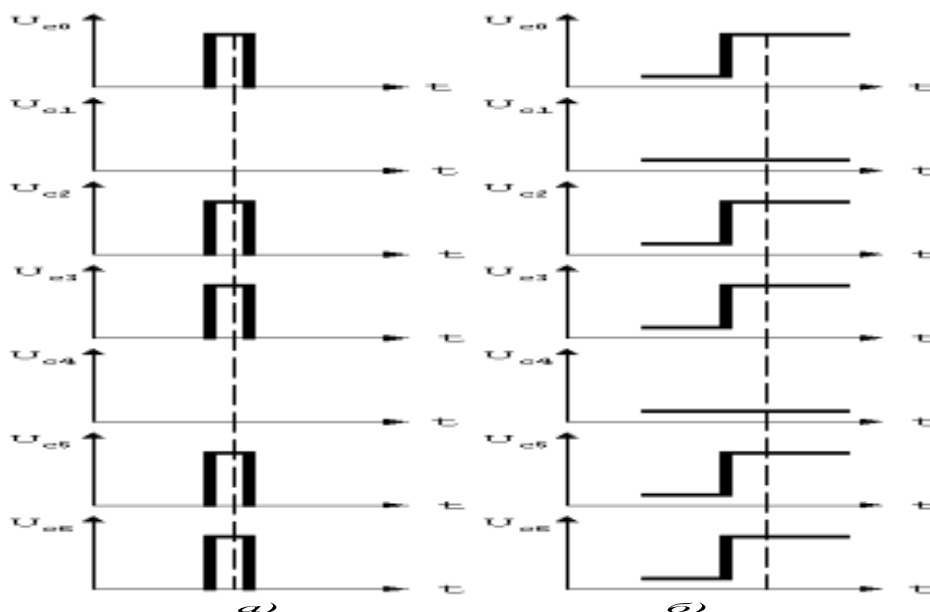


Рис. 2.2. Паралельний двійковий числовий код з потенціалом і імпульсом способи подання інформації

Умовне графічне зображення найпростішого цифрового автомата показано на рис. 2.3. Комбінація двійкових змінних X_1, X_2, \dots, X_n , подається на входи машини, а комбінація двійкових змінних видається з виходу Y_1, Y_2, \dots, Y_m . Логічні сигнали 0 і 1, звані двійковими, діють на входи і виходи цифрової машини. Завдання побудови цифрової машини, що виконує певні дії над двійковими сигналами, полягає у виборі елементів і способу їх з'єднання, які забезпечують задане перетворення. Ці задачі вирішуються за допомогою математичної логіки або логічної алгебри.

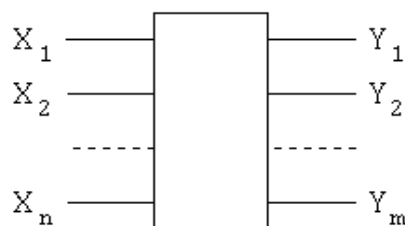


Рис. 2.3. Умовне позначення найпростішого цифрового автомата

Пристрої, які створюють функції логічної алгебри, називаються *логічними* або *цифровими* та класифікуються за різними характеристиками. За характером інформації на входах і виходах цифрові пристрої поділяються на послідовні, паралельні та змішані.

Символи кодів слів надходять на послідовні входи пристрою не одночасно, а один за одним (рис. 2.4). Вихідне кодове слово створюється в тому ж порядку.

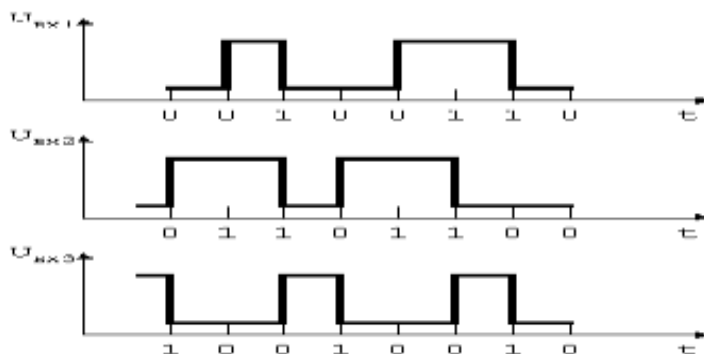


Рис. 2.4. Подання інформації в цифровому пристрої з послідовним функціонуванням

Розглянутий пристрій формує на виході сигнал $U_{\text{вих}}$, код якого дорівнює сумі кодів вхідних сигналів $U_{\text{вх1}}$ і $U_{\text{вх2}}$. Для реалізації паралельного пристрою, який виконує аналогічну функцію, потрібні дві групи входів з вісьмома бітами в кожній групі та вісьмома виходами (відповідно до розрядності вихідного слова). Існують також пристрої змішаного типу, в яких, наприклад, вхідне слово подається паралельно, а вихідне — послідовно (це перетворювачі коду).

За схемним рішенням і характером зв'язків між вхідними і вихідними величинами з урахуванням їх зміни шпаруватості виділяють комбінаційні і послідовні цифрові пристрої.

У комбінованих пристроях набір вихідних сигналів у будь-який конкретний момент часу повністю визначається вхідними сигналами, що діють на його входи в цей час. Якщо функції входу та виходу в n - циклі позначити як X_n і Y_n , то зв'язок між ними визначатиметься виразом

$$Y^n = l(X^n),$$

де l – знак логічного перетворення, виконаного пристроєм.

Цифрові пристрої, на відміну від аналогових, допускають практично будь-які види перетворення, але комбіновані пристрої не мають пам'яті. У цифрових пристроях послідовного типу (або машинах з пам'яттю) значення вихідних змінних Y_n в n - такті визначаються не тільки значеннями X_n активних в даний момент часу вхідних змінних, а й залежать від внутрішні стани пристрою C_n . У свою чергу, внутрішні стани пристрою C_n залежать від значень змінних, що діють на вхід в попередніх циклах. Таким чином, послідовні цифрові пристрої зберігають інформацію про попередні цикли пристрою і, отже, мають пам'ять. Роботу послідовного пристрою можна аналітично записати у вигляді

$$\begin{aligned} Y_n &= J(X^n, C^n); \\ C_n &= F(X^{n-1}, C^{n-1}), \end{aligned}$$

де X^{n-1} і C^{n-1} – набір вхідних змінних і внутрішніх станів пристрою в попередньому циклі відповідно.

Прикладом послідовного пристрою може бути лічильник імпульсів, вихідні стани якого залежать від загальної кількості імпульсів, що надходять на його вхід. Набір змінних на виході лічильника в n -такті залежить від наявності імпульсу на вході лічильника в такті, а також від станів лічильника, які визначаються загальною кількістю імпульсів, що надійшли на його вхід в такт $1, 2, \dots, n-1$.

2.2 Класифікація та характеристика елементної бази цифрових пристроїв

Майже всі логічні елементи (ЛЕ) виготовляються у вигляді інтегральних мікросхем (ІС) на основі напівпровідникових елементів (транзисторів і діодів). Класифікація ЛЕ базується в основному на типі активних інгредієнтів, які використовуються в їх конструкції, і враховує взаємозв'язки між цими інгредієнтами

У процесі розробки мікросхемотехніки було виділено кілька типів базових логічних елементів (БЛЕ), які мають досить високі значення параметрів і зручні для комплексної реалізації. Їх класифікація наведена на рис. 2.5.

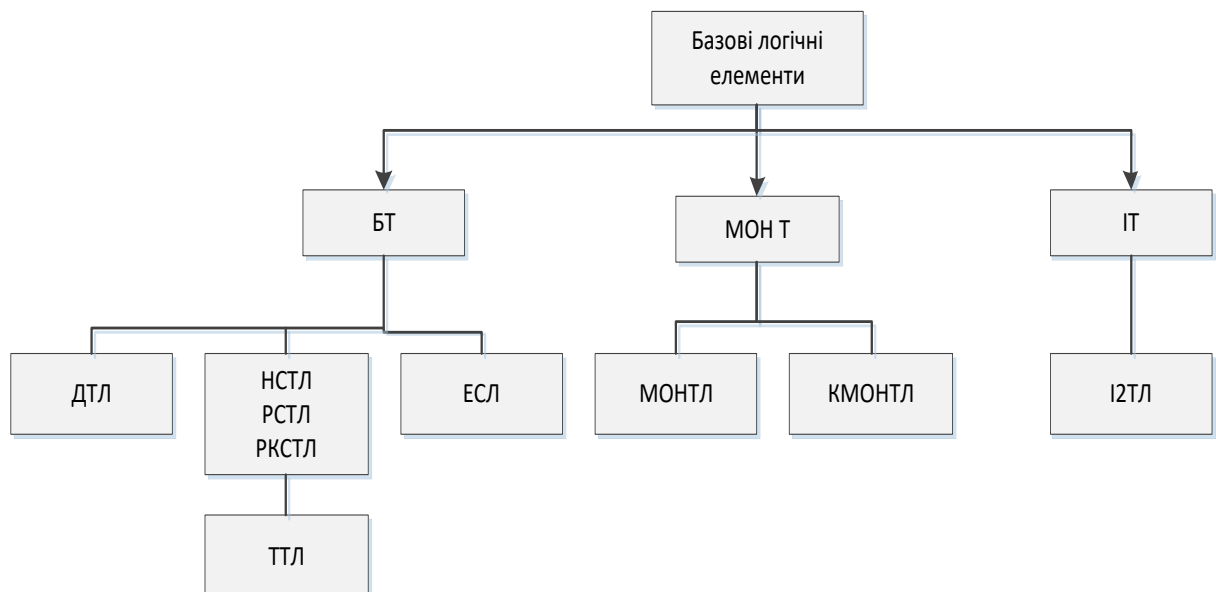


Рис. 2.5 Класифікація типових логічних елементів

Активними пристроями є біполярні транзистори (BT), транзистори MDP (MDPT) і інжекційні транзистори (IT).

Залежно від типу активних пристроїв, за допомогою яких здійснюються операції введення і виведення, базові логічні елементи і цифрові інтегральні схеми на їх основі прийнято поділяти на такі типи логіки:

- світлодіоди (DL);
- діодно-транзисторний ЛЕ (ДТЛ);
- транзисторно-транзисторний ЛЕ (ТТЛ);
- ЛЕ з'єднаний з емітером (EL);
- світлодіод на польових транзисторах (FET);
- ЛЕ з інжекційним джерелом живлення (ІЛ).

Для порівняльної оцінки властивостей ЛЕ введено ряд характеристик, найважливішими з яких є: статичні характеристики редуктора, швидкість, кількість входів, навантажувальна здатність, стійкість до перешкод і споживана потужність.

Розробка електронних пристроїв почалася близько 50 років тому. Залежно від використовуваних компонентів і технології їх виробництва цифрові пристрої поділяють на те чи інше покоління. Основою першого покоління елементів цифрової техніки були електронні лампи, електромагнітні реле, резистори, конденсатори та інші дискретні компоненти. Швидкість комп'ютера на основі цих пристроїв становила кілька десятків тисяч операцій в секунду. Основними недоліками, які перешкоджали їх широкому застосуванню в різних сферах народного господарства, були трудомісткість, низька швидкість і значна енергоємність. У наступні роки з появою транзисторів і мікросхем ситуація істотно змінилася. Основу компонентів ЕОМ другого покоління склали напівпровідникові прилади і мініатюрні дискретні деталі, а в якості пристроїв пам'яті використовували феритно-транзисторні комірки. Продуктивність твердотільних комп'ютерних компонентів зростає більш ніж на порядок порівняно з ламповими компонентами.

Мікросхеми з низьким і середнім ступенем інтеграції стали основними базовими елементами ЕОМ третього покоління. Після появи мікрочіпів фізики

та інженери спрямували свої зусилля на розробку вдосконалених технологічних процесів, які дозволили б більш компактно розмістити компоненти на монокристалі. У 1972 році почалася інтенсивна розробка ЕОМ четвертого покоління. Застосування технологій тонкого шару та методів осадження тонкого шару у вакуумі у виробництві оптичних методів призвело до створення великих інтегральних схем (ІС), які містять десятки й сотні тисяч елементів і з'єднань. З кожним роком кількість елементів на одному кристалі подвоювалася, в результаті чого в 1980 році рівень інтеграції сягнув порядку сотень тисяч елементів на кристалі - це вже надвеликі інтегральні мікросхеми.

Залежно від технології виготовлення інтегральні схеми поділяються на серії (сімейства), які відрізняються фізичними параметрами основних елементів, а також кількістю мікросхем, що входять до них. В даний час розроблено кілька десятків технологій виробництва інтегральних схем (ІС). Найпоширенішими застосуваннями є інтегральні схеми, виготовлені за такими технологіями: *транзисторно-транзисторна логіка (TTL)*; *комплементарний металооксидний напівпровідник (CMON)*; *n-MON*. Кожна технологія постійно вдосконалюється, щоб збільшити швидкість мікросхеми, зменшити споживання електроенергії та збільшити порядок інтеграції – кількість елементів, розміщених на кристалі із заданою площею поверхні.

TTL Є. У таблиці 2.1 наведено основні параметри базових елементів різних серій іноземних інтегральних схем, виготовлених за технологією TTL, а в таблиці 2.2 — сумісність іноземних інтегральних схем з вітчизняними.

Крім серії SN74, іноземні фірми випускають серії SN54 і SN84, які відрізняються лише діапазоном температур і допустимим відхиленням напруги живлення від номінального значення (табл. 2.3).

Найважливішим параметром, що характеризує якість серії ІС, є операція перемикання - добуток середнього часу затримки сигналу в вентилі $t_{з,сеп}$ на споживану потужність вентиля P (табл. 2.1).

Решта параметрів мають такі позначення:

- $I_{вх}^0$ - вхідний струм при подачі на вхід рівня логічного нуля;

- $I_{вх}^1$ - вхідний струм при подачі на вхід на рівні логічної одиниці;
- F_{max} - максимальна частота перемикання;
- $I_{вих}^0$ - вихідний струм на рівні логічного нуля на виході;
- $I_{вих}^1$ - вихідний струм на рівні логічної одиниці на виході;
- n - вантажопідйомність (коефіцієнт $I_{вих}^0 / I_{вх}^0$).

Таблиця 2.1 - Основні параметри технології ТТЛ ІМС

Серія ІС	$m_{з.сир}, НС$	$P, мВт/вен$	$I_{вих}^1, мкА$	Вхід $I^0, мА$	$F_{max}, кГц$	$I_{вих}^1, мкА$	Вихід $I^0, мА$	n	$m_{з.сир}, НС$
SN74	10	10	40	-1,6	35		16	10	100
SN74L	33	1	10	-0,18	3	-400	3.6	10	33
SN74H	6	22	50	-2	50	-200	20	10	132
SN74LS	9.5	2	20	-0,36	45	-500	8	20	19
SN74S	3	19	50	-2	125	-400	20	10	57
SN74AL	4	1	10	-0,2	50	1000	8	40	4
SN74AS	1.5	22	-	-	200	-400	20	100	33
SN74F	2	4	20	-0,6	130	-1000	20	33	8

Таблиця 2.2 – Сумісність іноземних МЕ з вітчизняними

Серія IS		Серія IS	
Батьківщина	іноземні	Батьківщина	іноземні
155	SN74	133	SN54
158	SN74L	136	SN54L
131	SN74H	130	SN54H
555	SN74LS	533	SN54LS
531	SN74S	530	SN54S
KP1533	SN74ALS	1533	SN54ALS
KP1531	SN74F	1531	SN54F

Серія IS розшифровується наступним чином:

SN74/SN54 - стандарт - 1963;

SN74H/SN54H - висока швидкість - швидка робота - 1967;

SN74L/SN54L - Low power - малопотужний - 1967;

SN74S/SN54S - з використанням діодів Шоттки - 1969 р.;

SN74LS/SN54LS - малопотужний Шоттки - малопотужний на діодах Шоттки - 1971;

SN74AS/SN54AS - Advanced Schottky - вдосконалений діодами Шоттки - 1982;

SN74ALS/SN54ALS - Advanced Low power Schottky - вдосконалена низька потужність з діодами Шоттки - 1980;

SN74F/SN54F - удосконалений TTL Шоттки Fairchild - покращений діодами Шоттки Fairchild - 1979;

Діоди Шоттки не допускають глибокого насичення транзисторів, що значно збільшує швидкість перемикання затвора.

Усі серії сумісні щодо входів і виходів: вихідна напруга логічного нуля $U_{вих}^0 \leq 0.4\text{В}$; вхідна напруга логічного нуля $U_{вх}^0 \leq 0.4\text{В}$; вихідна напруга логічної одиниці $U_{вих}^1 \geq 2.4\text{В}$; вхідна напруга логічного блоку $U_{вх}^1 \geq 2.4\text{В}$.

Таблиця 2.3 - Відмінності між параметрами серій іноземних інтегральних схем

Серія IS	Діапазон робочих температур, °C	Напруга живлення, В
SN54	-55...+125	4,5 ... 5,5
SN74	0...+70	4,75 ... 5,25
SN84	-25...+85	4,75 ... 5,25

Системи КМОН характеризуються низьким енергоспоживанням у статичному режимі та вищою захищеністю порівняно з системами TTL. У таблиці 2.4 наведено сумісність між вітчизняними та іноземними серіями ІС.

У системах КМОН швидкість роботи значною мірою залежить від напруги джерела живлення (з підвищенням напруги живлення зростає швидкість роботи).

Порівняно з TTL ІС слід відзначити наступні переваги КМОН ІС:

- низьке енергоспоживання в діапазоні частот до 2 МГц;
- широкий діапазон напруг живлення (3...15 В);
- можна використовувати нестабілізоване джерело живлення;
- дуже високий вхідний опір ($10^3 \dots 10^6$ МОм);
- висока навантажувальна здатність ($n=50$; $n=1000$ на частотах до 10 кГц);
- мала залежність характеристик від температури.

Недоліки КМОН ІС серій 561 і 1561:

- надмірний вхідний опір (0,5 ... 1 кОм);
- вплив потужності навантаження та напруги живлення на час затримки, тривалість фронту та споживану потужність;
- тривалі періоди затримок і фронтів.

Таблиця 2.4 - Відповідність між вітчизняними та закордонними ІС КМОН

Серія IS		Надійний	Напруга живлення, В
Батьківщина	іноземні		
164, 176	CD4000	RCA	9 і 3...15

564, 561	CD4000A	RCA	3...15
-	MC14000A	Motorola	-
KP1561	CD4000B	RCA	3...18
-	MC14000B	Motorola	-
1561	54HC	National Semiconductor Corp. Motorola	2...6

При побудові схем ІС часто використовуються не всі їх входи. Залежно від логіки схеми, ці входи повинні подаватись на логічний рівень 0 або 1. Логічний рівень 0 у схемах TTL і КМОН живиться шляхом підключення невикористаного входу до шасі (0 В). Логічний рівень 1 застосовується до невикористаних входів, підключаючи їх до джерела напруги живлення, однак рекомендується підключати входи TTL ІС до джерела через резистор обмеження струму, щоб захистити від перенапруг, що виникають, наприклад, коли живлення ввімкнено. Якщо вхід TTL мікросхеми не підключено ні до шасі, ні до джерела живлення, мікросхема працюватиме так, ніби до неї було застосовано логічний рівень 1, але надійність роботи буде низькою (імпульсний шум, спричинений перемиканням суміжних входів і виходів, може призвести до неочікуваної операції ІС). Жоден невикористаний вхід не може залишатися непідключеним у КМОН ІС).

Статичною характеристикою передачі є співвідношення $U_{вих} = f(U_{вх})$, де $U_{вх}$ - напруга на одному з входів. Після видалення цієї характеристики постійний рівень, що відповідає 0 (для елементів OR, OR-NOT) або 1 (для елементів AND, AND-NOT), застосовується до решти входів. Типова форма характеристики інвертувального елемента наведена на рис. 2.6. Тут позначені його основні параметри:

логічні рівні вихідної напруги U^0_{out} і U^1_{out} , які відповідають значенням 0 і 1; однакові рівні U^0_{in} і U^1_{in} також показані на вході;

$$\text{значення логічної різниці } U_m = U^1_{вих} - U^0_{вих};$$

порогові рівні $U^1_{пор}$ і $U^2_{пор}$, які умовно визначаються в тих точках характеристики, де дотична до неї проходить під кутом 45° до горизонту;

ширина активної області $\Delta U = U^2_{\text{пор}} - U^1_{\text{пор}}$;

середній коефіцієнт пропускання (підсилення) в активній зоні $K_{\text{ср}} = U_m / \Delta U$.

Логічні можливості елемента характеризуються логічною функцією, яку він виконує, і такими двома параметрами:

$K_{\text{об}}$ — вхідний коефіцієнт зв'язку, тобто максимальна кількість входів, обмежена в основному кількістю виходів з корпусу МК; як правило, коливається від 2 до 8;

$K_{\text{разг}}$ — коефіцієнт розгалуження на виході, тобто максимальна кількість однотипних елементів, які можна підключити паралельно до виходу без порушення нормальної роботи цього елемента. Залежно від типу елемента цей параметр, що виражає навантажувальну здатність, може коліватися від 3 до 100. Збільшення $K_{\text{об}}$ і $K_{\text{разг}}$ дозволяє зменшити загальну кількість елементів, необхідних для побудови логічної схеми.

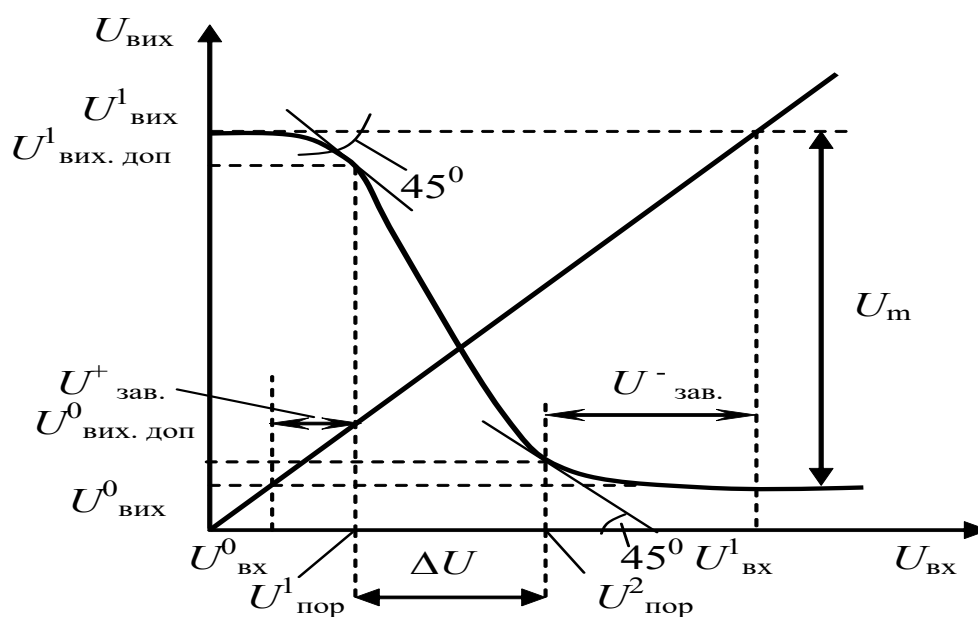


Рис. 2.6. Типові статичні характеристики інвертуючого логічного елемента

Опір визначається на основі допустимого значення напруги перешкод $U_{\text{зав}}$, при якому не відбувається помилкового спрацьовування елемента (перехід

від 0 до 1 або навпаки). Значення і $U_{зав}^- = U_{вх}^1 - U_{пор}^2$ використовуються для наближеної оцінки опору $U_{зав}^+ = U_{пор}^1 - U_{вх}^0$ (рис. 2.6).

Допустимі динамічні (імпульсні) збурення можуть перевищувати ці значення і бути тим більше, чим коротше збурюючий імпульс.

Швидкість ЛЕ є його основною динамічною характеристикою. Це визначається затримкою поширення падіння напруги на елементі $t_{зр}$, вимірним при половині значення падіння (рис. 2.7). Як правило, $t_{зр}^{01} \neq t_{зр}^{10}$, а швидкість характеризується середнім запізненням $t_{зр.ср} = (t_{зр}^{01} + t_{зр}^{10}) / 2$.

Для різних типів елементів це значення коливається від часток до сотень наносекунд.

Переварений потужність в статичному режимі $P_{сп}$ залежить від стану елемента 0 або 1, тому що струм, що витікає з джерела живлення, може бути різним. При великій кількості елементів на діаграмі приблизно половина з них перебуватиме в стані 0, інша половина – у стані 1. Тому доцільно використовувати середні значення потужності.

$$P_{сп.ср} = (P_{сп}^0 + P_{сп}^1) / 2.$$

У динамічному режимі споживання електроенергії зростає зі збільшенням частоти перемикавання ЛЕ.

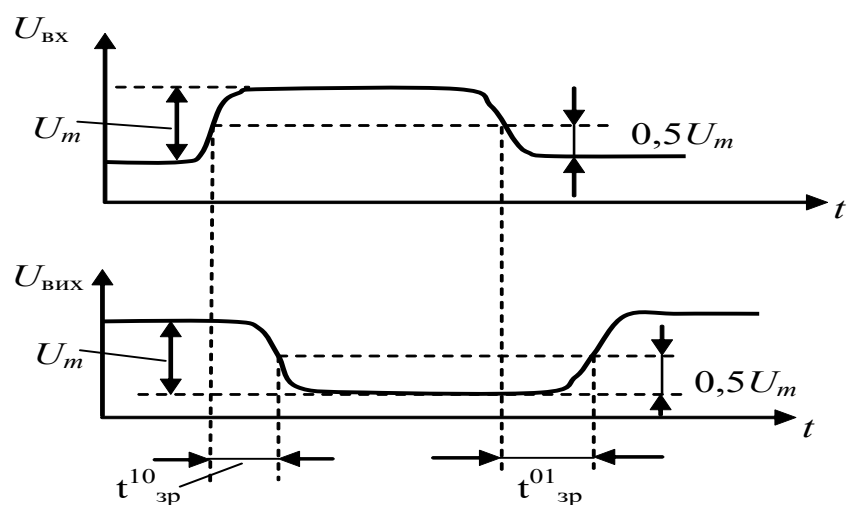


Рис. 2.7. Вимірювання часу затримки логічного елемента

У разі узагальнених характеристик елемента $A = P_{\text{сп.сп}} \cdot t_{\text{зр.сп}}$ часто використовують величину, яку називають роботою перемикання. Існує тісний зв'язок між $P_{\text{сп.сп}}$ і $t_{\text{сп.сп}}$. Тому перевага віддається таким технологіям виробництва ЛЕ, при яких A зменшується, тобто досягається збільшення швидкості без збільшення енерговитрат. Сучасні ЛЕ можуть мати значення A менше 1 пДж (10^{-12} Дж).

2.3 Діодні логічні елементи (DL)

Основою діодів ЛЕ є діодний ключ - найпростіша схема, яка складається з діода VD і резистора R (рис. 2.8). Активний елемент схеми - діод працює в ключовому режимі.

Якщо напруга на вході ключа нижче порогу відкриття діода: $U_1 < U_{dv}$, то згідно з характеристиками діода $I_d = f(U_d)$ діод замкнутий ($I_d = 0$), напруга на резисторі R : $U_R = I_d R = 0$. При $U_1 > U_{\text{діод}}$ відкритий, опір діода у відкритому стані R_d мало — (50-100) Ом, а якщо вибрати $R \gg R_d$, то без урахування падіння напруги на діода U_d напруга U_R можна вважати наближено рівним U_1 .

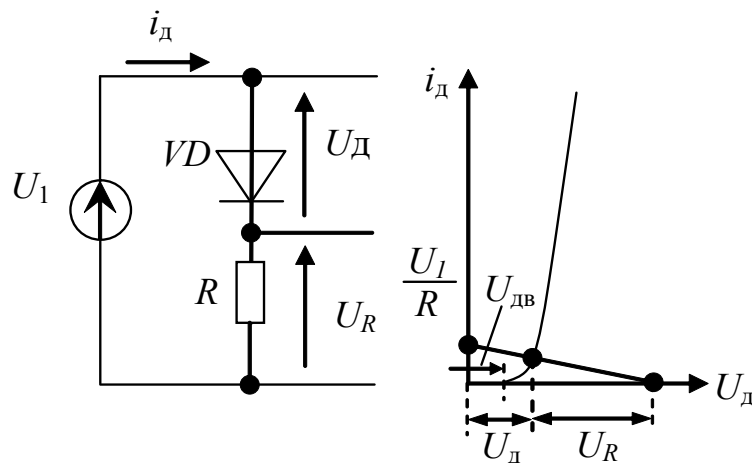


Рис. 2.8. Діодний перемикач та його статичні характеристики

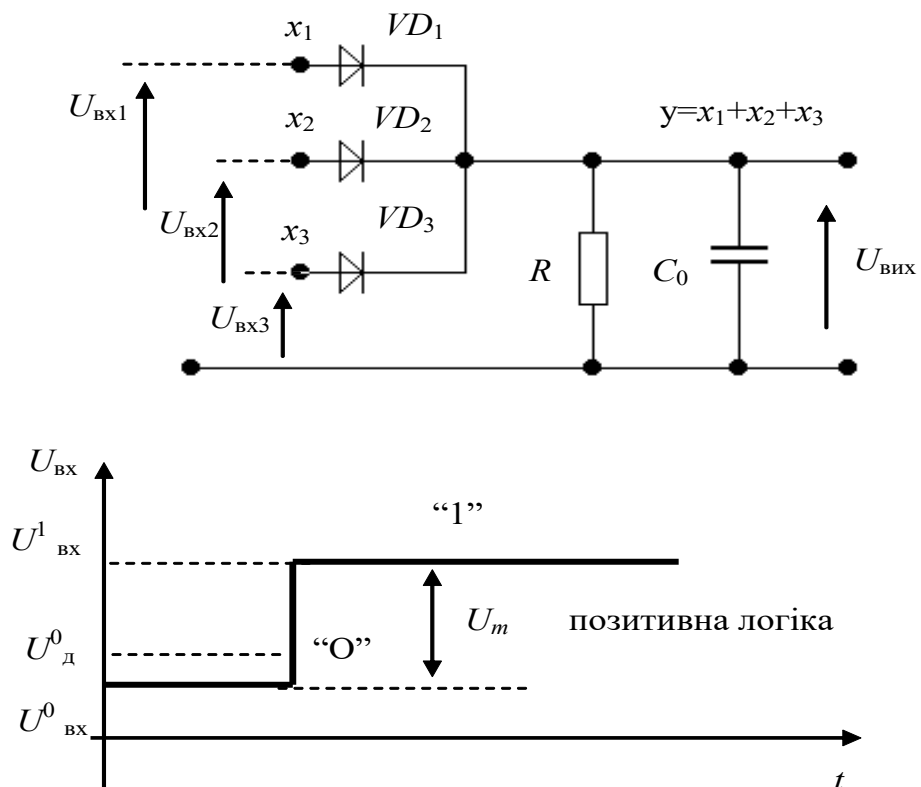


Рис. 2.9 . діодний елемент АБО та рівні його вхідної напруги

логічний елемент АБО. Схема LE та рівні вхідної керуючої напруги, що відповідають позитивній логіці, показані на рис. 2.9. Після подачі 0 на всі входи ($U^0_w < U_{dv}$) діоди замикаються і вихідна напруга $U_{out}=0$ ($y=0$). Достатньо подати 1 лише на один із входів ($U^1_w > U_{dv}$), щоб відповідний діод відкрився, і напруга на виході $U_{out}=U^1_w - U_d$ приблизно ($U_d \neq 0$) буде напруга на вході ($y=1$). При цьому діоди, входи яких живляться зі значенням 0, залишаються замкнутими, тобто вихідна напруга з одного з входів не доходить до інших входів - таким чином LE забезпечує роздільну здатність у вхідних ланцюгах. Як впливає з розглянутого принципу роботи, елемент виконує операцію АБО: $y=x_1+x_2+x_3$.

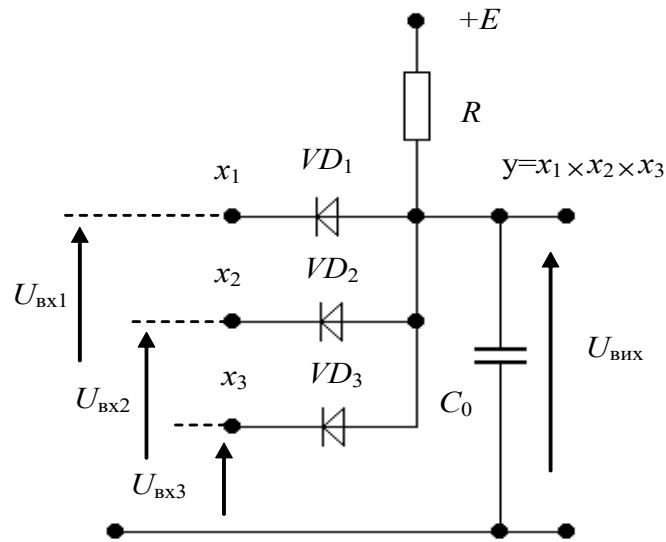


Рис. 2/10. Діодний елемент I

Логічний елемент I. Схема LE наведена на рис. 2/10. У ньому, на відміну від схеми АБО, вихідна напруга береться не з резистора R , а з діодів, з'єднаних паралельно один одному. Отже, якщо хоча б на один із входів подати 0, то відповідний діод під дією електромагнітного поля джерела живлення E відкриється і напруга на виході $U_{\text{вих}} = U_{\text{вх}}^0 + U_d$ дещо відрізнятиметься від вхідного U^0 , тобто на виході буде логічний 0 ($y=0$).

Тільки коли значення 1 застосовується до всіх входів, діоди замикаються, і на виході виникає високий рівень E ($y = 1$). Будь ласка, пам'ятайте, що якщо $E > U^1_{\text{вх}}$, діоди не замкнуті накоротко, але через них на вихід подається високий потенціал $U^1_{\text{вх}}$, що не змінює логіку роботи елемента. Тому він виконує логічну операцію I: $y = x_1 \cdot x_2 \cdot x_3$.

Розв'язка вхідних кіл тут забезпечується тим, що вхідний сигнал 1, який діє на один із входів, закриває відповідний діод, і висока напруга з цього входу не досягає інших входів.

Швидкодія діодів LE обмежена часом перемикання, що виникає внаслідок інерції діодів і, що особливо важливо, процесами встановлення, пов'язаними із зарядкою та розрядкою ємності C_0 , яка включає в себе несучу та паразитну ємності par . Швидкість роботи залежить від кількості входів елемента

та його навантажувальної здатності ($K_{\text{відкритий}}$). Збільшення числа входів (кількості діодів) призводить до збільшення загальної вихідної ємності C_0 за рахунок паразитних ємностей закритих діодів і до зниження швидкодії. Крім того, це зменшує еквівалентне навантаження закритих діодів, з'єднаних паралельно, що спричиняє шунтування виходу й таким чином зменшує значення падіння вихідної напруги.

вплив шунтування наступних елементів на попередній, тим більше що вхідне навантаження елемента приблизно дорівнює опору резистора R . При великій кількості елементів навантаження умова $R \gg R_d$ може не виконуватися. Практично значення K_{xg} не перевищує 2–3. З цих причин діоди ЛЕ в чистому вигляді, без підсилювальних транзисторних ключів, використовуються вкрай рідко.

2.4 Діодно-транзисторні логічні елементи (ДТЛ)

У діодно-транзисторних ЛЕ в якості активних елементів поряд з діодами використовуються біполярні транзистори в ключовому режимі, при якому транзистор може перебувати в одному з двох станів - закритому або відкритому і насиченому.

Для забезпечення надійного блокування транзистора, як видно з його характеристики (рис. 2.11), напруга на базі має бути менше напруги блокування E_{bo} , а струм колектора дорівнює тепловому струму I_{ko} . Залишкова напруга на колекторі в насиченому стані U_{kn} близька до нуля.

Цей режим може бути забезпечений подачею напруги зсуву від окремого джерела до базового кола. Щоб не ускладнювати схему, обмежуються режимом умовного блокування, домагаючись, щоб напруга на базі була менше порогового значення U , оскільки. При цьому колекторний струм I_K не перевищує значення $(1 + \beta)I_{ko}$ і ним можна знехтувати.

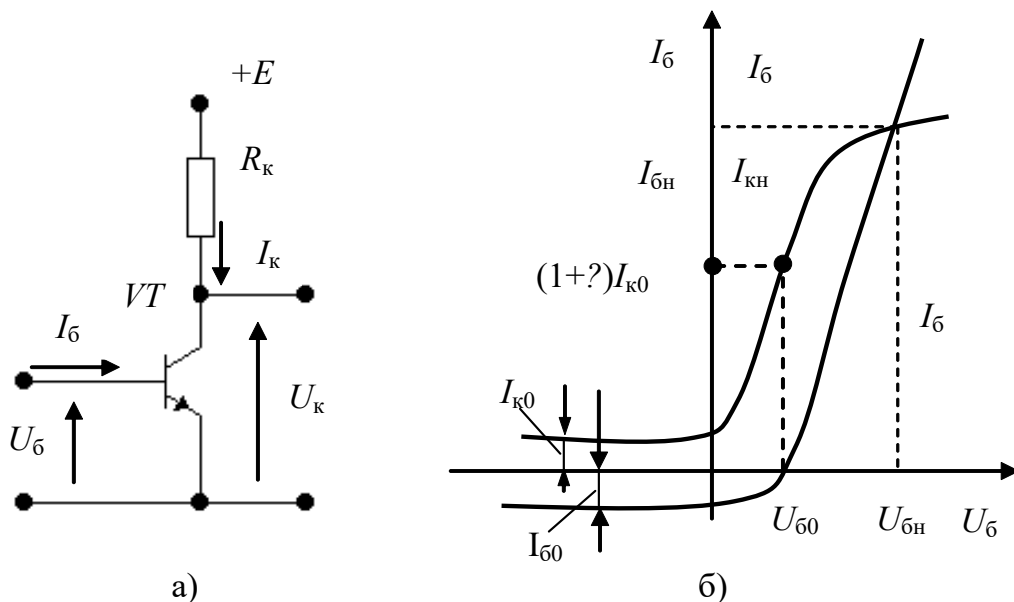


Рис. 2/11. Транзисторний ключ (а) і його статичні характеристики (б)

Щоб наситити транзистор, необхідно створити базовий струм, який перевищує базовий струм насичення:

$$I_{\text{б}} > I_{\text{бн}} = I_{\text{кн}} / B,$$

де $I_{\text{кн}}$ — струм колектора в режимі насичення;

— базовий коефіцієнт передачі струму в режимі великого сигналу.

Найпростіша схема елемента ДТЛ з трьома входами наведена на рис. 2.12, а. Виконує операцію І-НІ. Дійсно, діоди $VD_1 \dots VD_3$ разом з резистором R_0 і джерелом E_0 є елементом І (рис. 2.9), VT - інвертор, що виконує операцію НЕ, а діоди VD_{z1}, VD_{z2} разом з резистором R — коло зв'язку. Розглянемо роботу елемента докладніше.

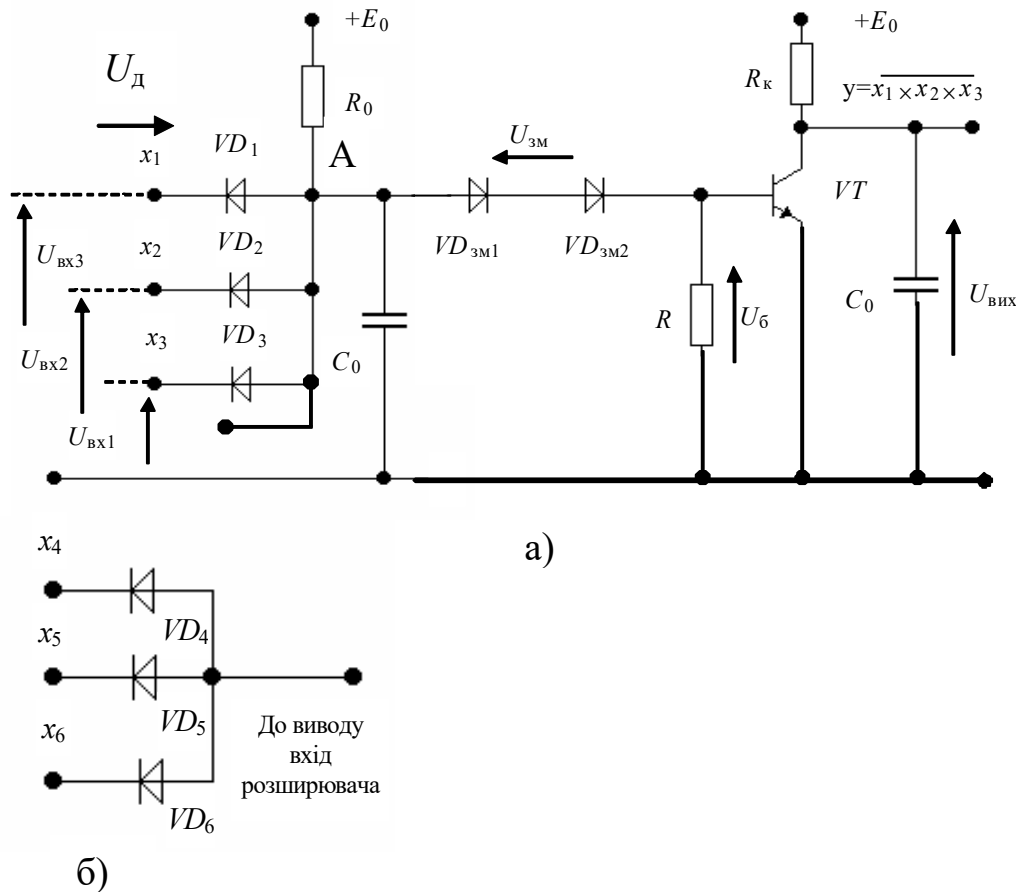


Рис. 2/12. Діодно-транзисторні ЛЕ а) – принципова схема; б) – вхідний розширювач

Коли принаймні до одного входу подається логічний нуль $U_{\text{вх}}^0$, напруга в точці А дорівнює $U_A = U_{\text{вх}}^0 + U_{\text{д}}$:

Якщо врахувати, що $U_{\text{вх}}^0$ це напруга на колекторі насиченого транзистора $U_{\text{кн}}$ аналогічного елемента і воно невелике, а напруга на відкритому діоді становить 0,5-0,7 В, то U_A становить близько 1 В. Д.о. безпосередньо на базі не забезпечується блокування транзистора VT . Діоди зміщення VD_{zm1} і VD_{zm2} разом з резистором R утворюють дільник, в якому напруга на плечі R дорівнює $U_R = U_{\text{б}} = U_A - U_{\text{zm}} < U_{\text{б0}}$. При цьому транзистор VT стає умовно закритим, а $K \approx 0$ і напруга $U_{\text{вих}} = U_K \approx E_K$. Це означає, що вихід ЛЕ буде 1, тобто рівень $U_{\text{вих}}^1$.

Коли значення 1 подається на всі входи ($x_1 = x_2 = x_3 = 1$), діоди вимкнуться, а джерело електромагнітного поля E_0 генеруватиме струм у базовому колі.

$$I_{\bar{o}} = \frac{E_0 - U_{zm} - U_{bn} - \frac{U_{bn}}{R}}{R_0},$$

і якщо цей струм перевищить I_{bn} , то транзистор VT насититься, напруга $U_{вих} = U_K \approx 0$ — на виході ЛЕ буде 0, $U_{вих}^0$ тобто.

Слід підкреслити, що струм колектора в насиченому стані визначається не тільки струмом I_{Rk} , а й залежить від вхідного струму навантаження: $I_{кн} = I_{Rk} + I_{вх} K_{розг}$. При постійному базовому струмі I_b збільшення навантаження може викликати перенасичення транзистора, збільшиться $U_{вих}$ — порушиться умова нормальної роботи ЛЕ. Для збільшення навантажувальної здатності (збільшення $K_{відкритий}$) замість транзисторного ключа встановлюють складний інвертор. Подібний складний інвертор використовується в транзисторно-транзисторних елементах.

Отже, якщо хоча б один вхід розглянутого елемента має значення 0, вихід буде 1, і лише коли всі входи мають значення 1, вихід буде встановлено в 0. Іншими словами, цей елемент реалізує І-НЕ функція, $y = \overline{x_1 \cdot x_2 \cdot x_3}$ тобто

Для збільшення числа входів елемента в мікросхемі створюється висновок «Вхід розширення», до якого можна підключити діодну збірку (рис. 2.12, б), тобто виконати «Розширення по І», Потім $y = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6}$.

Швидкість елемента обмежена інерцією транзистора (час накопичення заряду в базі до граничного значення при включеному транзисторі і час розрядки заряду від бази при його вимкненні), т.к. а також процеси встановлення, пов'язані із заряджанням і розряджанням паразитних конденсаторів і конденсаторів навантаження. Заряд ємності C_0 при закритому VT ($y = 1$) відбувається через резистор R_k , а розряд ($y = 0$) - через малий опір відкритого транзистора.

2.5 Транзисторно-транзисторна логіка (TTL)

2.5.1 Схема і принцип дії TTL- елемент

Транзисторно-транзисторні логічні елементи випускаються виключно з так званим складним інвертором. Мета його використання - зменшити вихідний опір елемента R_{ne} тільки в стані 0 (як у простого інвертора), але і в стані 1.

Один із варіантів схеми елемента TTL наведено на рис. 2/13. На вході схеми встановлений мультиемітерний транзистор VT_b , призначений спеціально для цього застосування. Кількість емітерів цього транзистора визначає кількість входів на логічний елемент. Транзистори VT_1 , VT_2 , VT_3 і діод зміщення VD_{zm} утворюють складний інвертор.

Якщо емітерні переходи VT вважати діодами, а колекторний перехід вважати діодом зміщення, то разом з транзистором VT_1 ця частина схеми стає схожою на схему елемента DTL, тобто забезпечує виконання логічного І- НЕ операція. Однак загальна база, що з'єднує емітерний і колекторний переходи VT_b , а також транзистори VT_2 і VT_3 створюють особливості в роботі схеми.

Розглянемо роботу елемента з урахуванням цих особливостей, маючи на увазі роботу сигналів у позитивній логіці. Коли на всі входи транзистора U_{ex}^1 VT подається 1 (високий рівень напруги), емітерні переходи закриті, а колекторний перехід відкритий, то багатоемітерний транзистор працює в активному реверсному режимі: він діє як емітер через колектор, а емітери виконують роль колекторів. Струм бази транзистора VT_1 визначається струмом колектора транзистора з оберненим зв'язком VT_b :

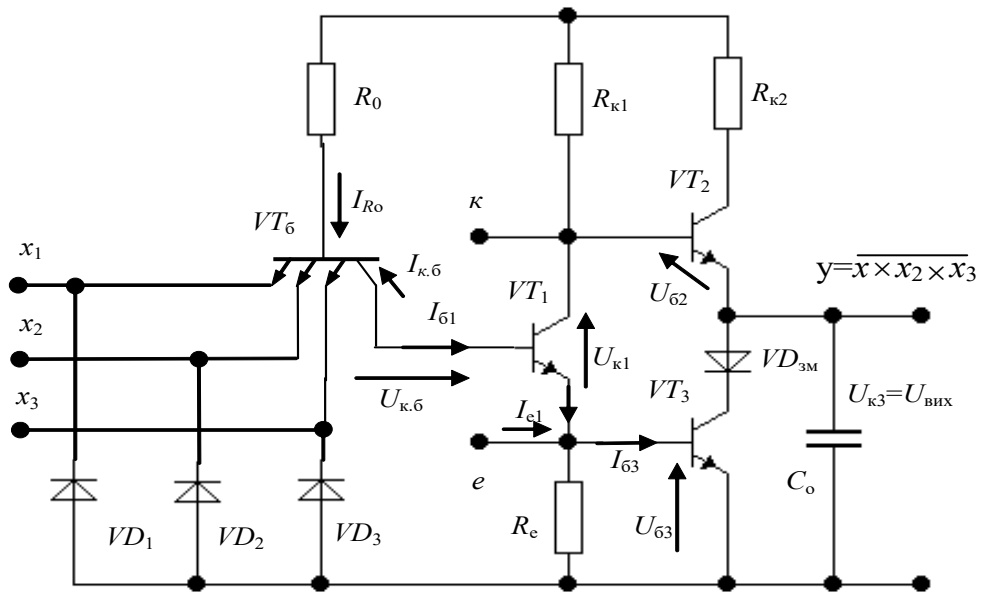


Рис . 2/13. Транзистор-транзистор ЛЕ зі складним інвертором

$$I_{\text{б1}} = -I_{\text{кб}} = I_{\text{Ro}} + \sum_{i=1}^3 I_{\text{вх}} \cdot$$

Коефіцієнт передачі основного струму VT_6 в реверсивному режимі β_{inv} малий. Вхідні струми («ті, що входять в схему») $I_{\text{вх}} = \beta_{\text{inv}} \cdot I_{\text{Ro}}$ малі, ними можна знехтувати, а якщо $I_{\text{б1}} \approx I_{\text{Ro}} > I_{\text{бн1}}$, то транзистор VT_1 буде насиченим. Базовий струм транзистора VT_3 $I_{\text{б3}} \approx I_{\text{е1}} \gg I_{\text{бн3}}$ і тому VT_3 також насичений. При цьому транзистор VT_2 буде закритий. Це забезпечується діодом зсубу VD . Дійсно, як видно зі схеми, у розглянутому випадку $U_{\text{б2}} = U_{\text{кн1}} + U_{\text{бн3}} - U_{\text{кн3}} - U_{\text{3м}}$. Вважаючи, що напруги на колекторах насичених транзисторів рівні: $U_{\text{кн1}} = U_{\text{кн3}}$, неважко помітити, що умова блокування VT_2 може $U_{\text{б2}} \approx U_{\text{бн3}} - U_{\text{3м}} < U_{\text{б02}}$ бути забезпечена лише напругою $U_{\text{3м}}$ на діоді $VD_{\text{3м}}$.

Напруга на виході ЛЕ при $x_1 = x_2 = x_3 = 1$ $U_{\text{вих}} = U_{\text{кн3}}$ — мала, що відповідає $U_{\text{вих}}^0$, тобто $y = 0$.

Якщо 0 — низький рівень напруги прикладається до одного або кількох входів $U_{\text{вх}}^0$, тоді відповідні емітерні переходи VT_6 будуть відкриті. Мультиемітерний транзистор буде в режимі прямого перемикавання - струм бази $I_{\text{бм}}$ і струм

колектора I_{km} течуть у бік відкритих переходів (вхідні струми змінюють напрям - «виходять з кола»). Струм колектора VT буде обмежений малим тепловим струмом бази транзистора VT_1 . Тому при заданому струмі $I_{бм} = I_{Ro} \approx E_{\kappa} / R_o$ мультиемітерний транзистор глибоко насичений, напруга на його колекторі відносно корпусу мала: $U_{\kappa} = U_{\kappa бн} + U_{\epsilon x}^0 < U_{\delta 01} + U_{\delta 03}$, транзистори VT_1 і VT_3 закриті. Високий потенціал колектора VT_1 по відношенню до корпусу забезпечить розблокування VT_2 . У ланцюзі емітера VT_2 є великий опір закритого транзистора VT_3 , тобто транзистор VT_2 працює в режимі емітерного підсилювача (опір резистора $R_{\kappa 2}$ невеликий і обмежує лише скачки струму колектора з транзистор розблокований). Коефіцієнт пропускання емітерного підсилювача близький до 1. Тому вихідна напруга приблизно повторює рівень високої напруги на колекторі VT_1 : $U_{\text{вих}} = U_{\text{вих}}^1$, тобто $y = 1$.

Швидкість елемента обмежена інерцією транзисторів і процесами зарядки та розрядки, навантажувальною здатністю та паразитними ємностями. Колекторний струм мультиемітерного транзистора сприяє прискоренню процесу зняття заряду бази при вимкнених транзисторах VT_1 і VT_3 . Заряд і розряд конденсаторів навантаження в елементі ТТЛ здійснюється малим вихідним опором комплексного інвертора як у стані 1, так і в стані 0. У першому випадку це визначається малим вихідним опором емітерного підсилювача, у другому - малим опором відкритого транзистора VT_3 .

Швидкодія елементів ТТЛ характеризується величиною t_{zr} . в середньому від 10 до 50 нс. Складний інвертор також забезпечує високу навантажувальну здатність. Звичайно, $K_{ореп} = 10$ гарантовано.

Розширення через АБО. Для розширення логічних можливостей елемента в деяких типах мікросхем (МС) додаткові колекторні (К) і емітерні (Е) виводи транзистора VT_1 виконуються з корпусу (рис. 2.14).

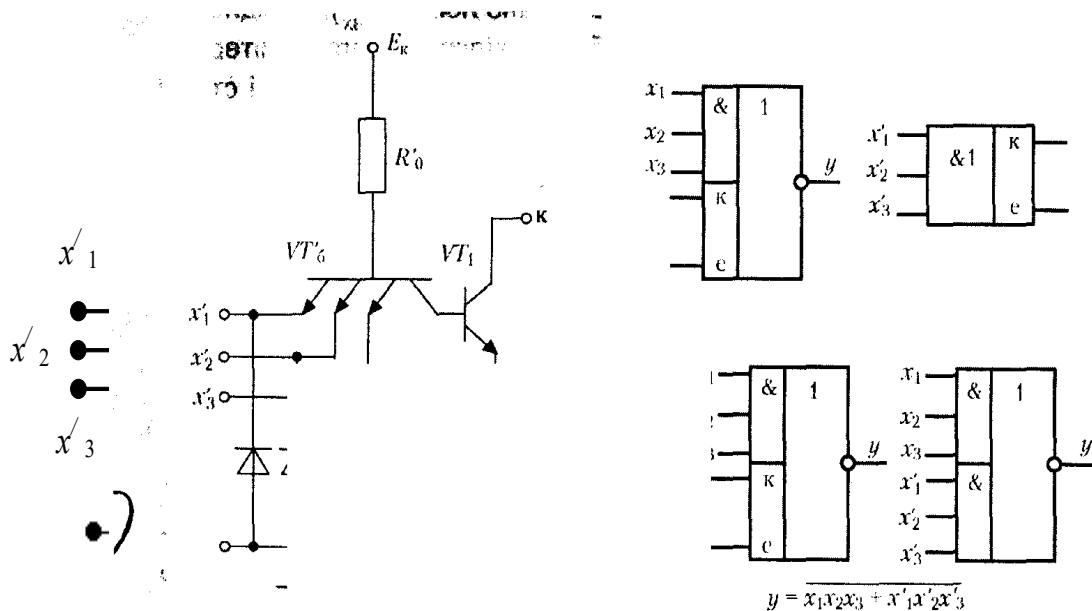


Рис. 2/14. Розширювач АБО для TTL - елемент:

а) – принципова схема; б) – позначки на функціональних схемах

Наявність цих клем дозволяє підключити «OR Extender», схема якого наведена на рис. 2.13, о. Стан транзистора VT_1 розширювача визначається комбінацією входних змінних x'_1, x'_2, x'_3 . Наявність у складному інверторі двох паралельно з'єднаних транзисторів $VT_{вих1}$ і VT_1 забезпечить низький рівень вихідної напруги, якщо хоча б один з транзисторів VT_1 або $VT_{вих1}$ відкритий, а для цього необхідно $x_1 = x_2 = x_3 = 1$ або $x'_1 = x'_2 = x'_3 = 1$. Лише коли обидва ці транзистори замкнуті накоротко ($x_i = 0$ і $x'_i = 0$), висока напруга з'явиться на вихід інвертора $U_{вих}^1$. Отже, ЛЕ з розглянутим розширювачем буде виконувати логічну операцію І-АБО-НІ.

2.5.2 Елементи TTL з відкритим колектором

Іноді необхідно використовувати ЛЕ для керування навантаженням, яке має окреме джерело живлення та споживає великий струм. Таким навантаженням може бути обмотка реле, світловий індикатор і т. д. Використовуються

елементи, в яких колектор вихідного транзистора залишається вільним (відкритим).

Спрощена схема ЛЕ з відкритим (вільним) колектором наведена на рис. 2.15, а.

Для нормальної роботи такої схеми колектор вихідного транзистора VT_2 необхідно підключити, як видно на малюнку, до джерела живлення через зовнішнє навантаження. У той же час підключені зовнішні пристрої можуть працювати від інших джерел живлення *вищої* напруги. ЛЕ з відкритим колектором дозволяють підключати декілька введів паралельно до загального навантаження (рис. 2.15, б). При такому підключенні, якщо на виході елементів є низький потенціал $U_{вих}^0$, то на виході всієї системи буде 0. Щоб забезпечити високий рівень потенціалу $U_{вих}^1$ на загальному виході, необхідно закрити вихідні транзистори усіх ЛЕ, тобто встановити для них стан 1.

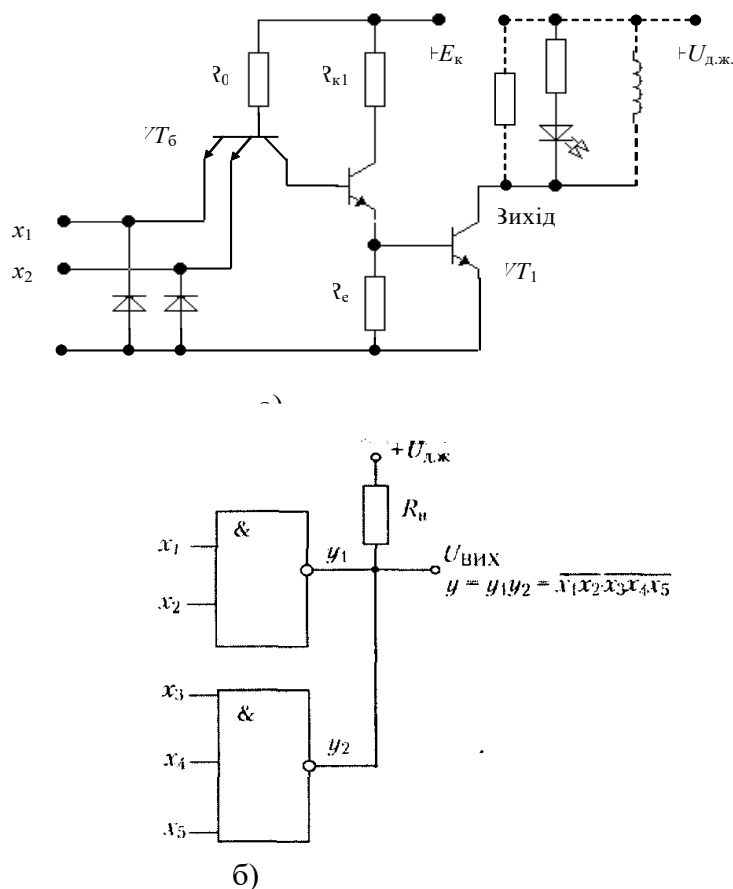


Рис. 2.15. Елемент TTL з відкритим колектором:

а) – принципова схема; б) – з'єднання елементів збірки І

Таким чином створюється загальне навантаження шляхом паралельного з'єднання декількох ЛЕ з відкритим колектором

система, що виконує операцію І: $F = y_1 \cdot y_2 = \overline{x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5}$. Це з'єднання ЛЕ з відкритим колектором зазвичай називається "I-mount".

2.5.3 Елементи ТТЛ на транзисторах Шотткі

Особливістю транзистора Шотткі є, як відомо, те, що при ключовому режимі струм колектора обмежується зверху без входу в область насичення, тобто без відключення колекторного переходу. Тому ніяких додаткових дій для прискорення зняття надлишкового заряду з бази при вимкненому транзисторі робити не потрібно, тому що такий заряд не створюється. Використання транзисторів Шотткі в ТТЛ-елементах дозволяє підвищити їх швидкодію без збільшення споживаної потужності або зменшити споживану потужність за рахунок збільшення опору всіх резисторів без зниження швидкодії. Такі ЛЕ називають елементами ТТЛШ.

Принципова схема одного з варіантів елементів ТТЛШ наведена на рис. 2.16. За будовою та загальним принципом дії вона не відрізняється від схеми на рис. 2.13.

Особливість полягає в тому, що роль діода зміщення тут виконує емітерний перехід транзистора VT_4 . Більше того, пара $VT_2 - VT_4$ утворює так званий складений транзистор (схема Дарлінгтона). Величина його коефіцієнта пропускання по базовому струму близька до добутку коефіцієнтів пропускання $V_2 \cdot V_4$ транзисторів VT_2 і VT_4 . Це призводить до зменшення вихідного опору емітерного підсилювача при закритому VT_3 і, як наслідок, до прискорення процесу зарядки навантаження при переході елемента зі стану 0 у стан 1. В елементах ТТЛШ можлива для реалізації значення затримки $t_{зг}$ в середньому близько 5 нс.

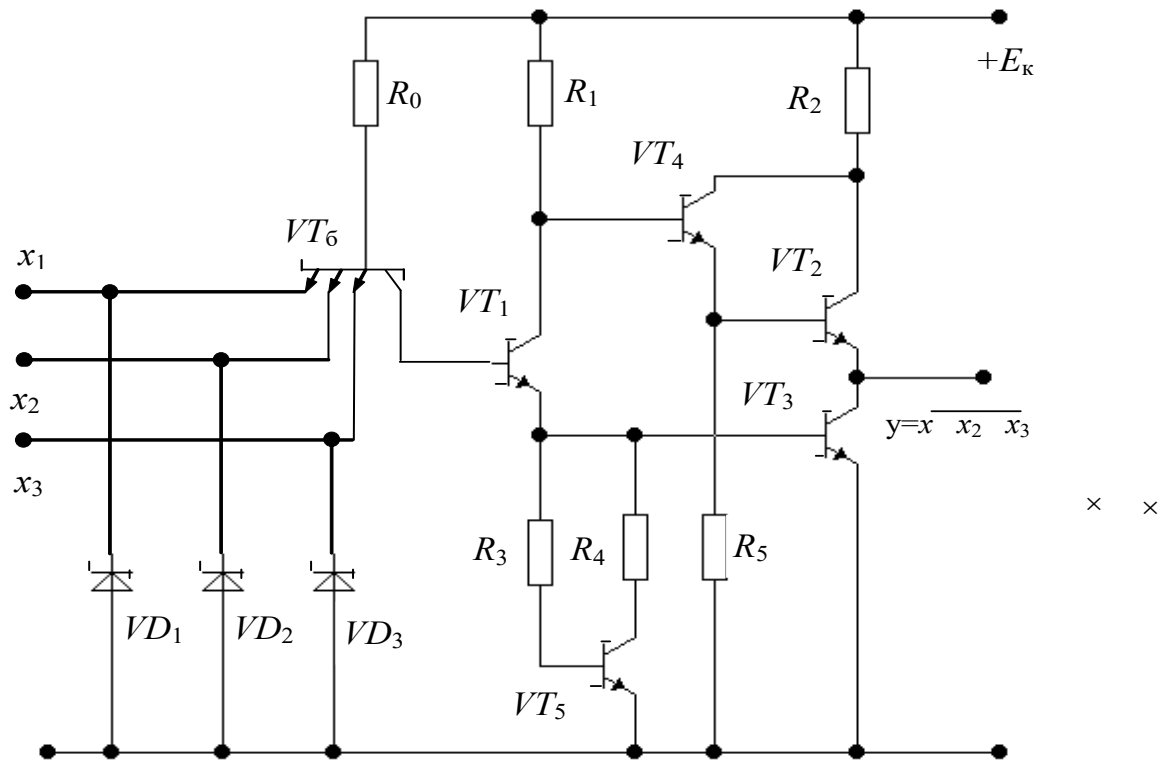


Рис. 2.16. Елемент ТТТ на транзисторах Шоттки і

2.5.4. TTL елементи з трьома вихідними станами

Як відомо, розглянуті ЛЕ в залежності від комбінації вхідних сигналів можуть перебувати в одному з двох станів - 0 або 1. У кожному з цих станів на виході елемента встановлюється відповідний рівень напруги - $U_{\text{вих}}^0$ або $U_{\text{вих}}^1$, і елемент має питомий вихідний опір. У стані 0 вихідний опір малий порівняно із загальною шиною (транзистор відкритий і насичений). У стані 1 - відносно шини живлення (вихідний опір емітерного підсилювача). Такі властивості елементів виключають можливість об'єднання їх виходів для роботи на спільне навантаження.

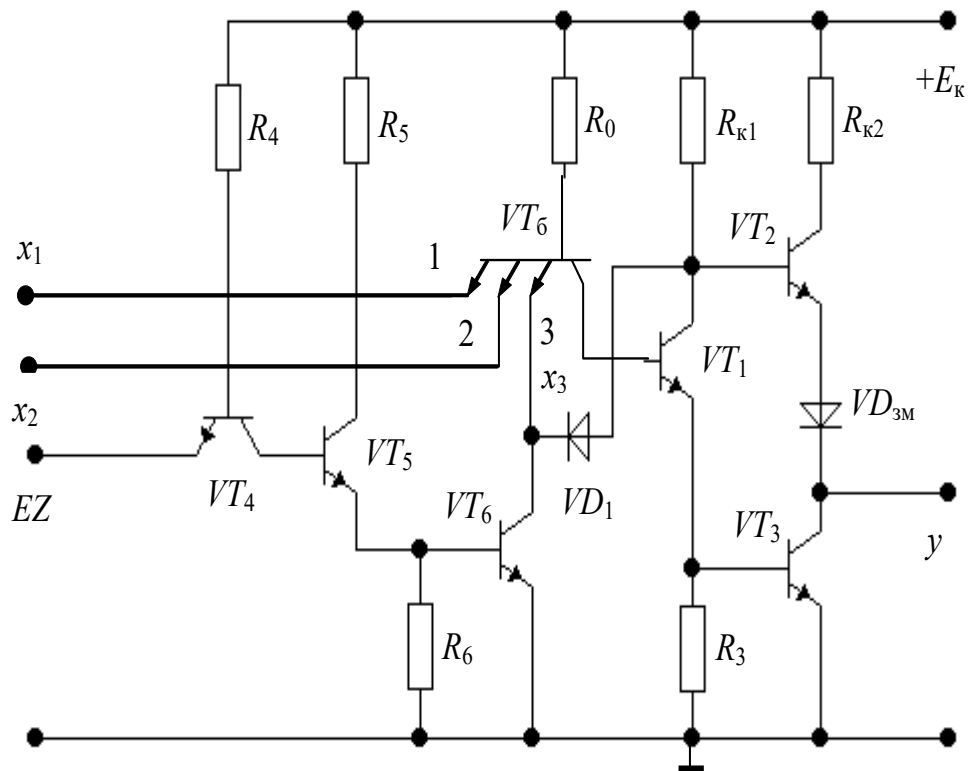
Існують мікросхеми різного функціонального призначення, виготовлені за різними технологіями, здатні після подачі відповідного керуючого сигналу переходити в третій стан, в якому вихідний опір стає дуже значним як по відношенню до загальної шини, так і по відношенню до шини живлення. Цей стан

еквівалентно відключенню елемента від навантаження і називається високим імпедансом.

При попереми́нній роботі таких пристроїв їх виходи можна об'єднати і об'єднати в загальну мережу, зокрема підключивши до звичайних міських автобусів. Спрощена принципова схема елемента І-НІ з трьома початковими станами наведена на рис. 2.17.

Основною відмінністю від схеми базового елемента (рис. 2.13) є наявність діода VD_1 , який забезпечує односторонній зв'язок одного з емітерів транзистора VT (третього) з базою транзистора VT_2 .

Якщо $x_3 = 1$, тобто шлях для струму через цей емітер і через діод VD_1 відсутній, то стан елемента визначається сигналами x_1 і x_2 і він виконує свою характерну функцію $y = \overline{x_1 \cdot x_2}$. Якщо $x_3 = 0$, тобто в цей момент є низький потенціал, то за принципом дії елемента І-НІ транзистор VT_3 закритий і вихід повинен бути в стані 1. Але при цьому сигнал 0 через діод VD_1 закриває транзистор VT_2 . Таким чином, вихід елемента відключається як від загальної шини, так і від шини живлення. Високоімпедансний стан Z встановлюється при $x_3 = 0$, сумарний струм третього емітера і діода VD_1 досить великий. Щоб не навантажувати ним джерело керуючого сигналу, цей сигнал EZ подається через спеціальний інвертор на транзисторах VT_4, VT_5, VT_6 . Якщо $EZ=0$, VT_4 глибоко насичений, VT_5 і VT_6 закриті, це означає, що $x_3 = 1$.



а)

x_1	x_2	EZ	y
0	0	0	1
0	1	0	1
1	0	0	1
1	1	0	0
x	x	1	Z

б)

Рис. 2.17. TTL — елемент із трьома станами: а) схема; б) таблиця стану

Коли $EZ=1$, VT_4 закривається, а VT_5 і VT_6 відкриваються і переходять у насичений стан. Струм емітера і діода VD_1 протікає через насичений транзистор VT_6 . Елемент переходить у третій стан. Таблиця стану елемента показана на рис. 2.17, нар.

2.5.5. Особливості практичних схем елементів ТТЛ

Елементи ТТЛ мають досить високу швидкодію, особливо елементи ТТЛШ. Напруги і струми викликають різкі стрибки, причому струми на входах під час стрибків навіть змінюють напрямок. У результаті в коливальних контурах, створених паразитними ємностями та індуктивностями, виникають затухаючі коливання, які часто називають дзвінками.

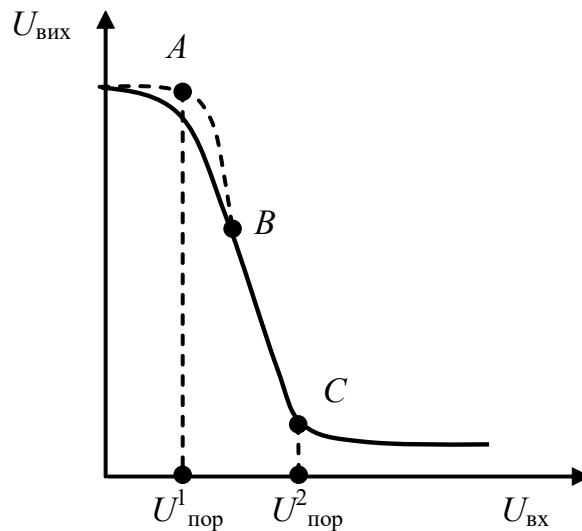


Рис. 2.18. Передавальні характеристики елемента ТТЛ

Особливу небезпеку становлять перепади напруги на входах. Наприклад, після позитивного провалу, який закриває емітерний перехід транзистора VT_6 , негативний напівперіод коливальних може призвести до відкриття переходу та помилкового спрацювання елемента. Для усунення цього ефекту входи з'єднані з корпусом за допомогою реверсивно перемиканих діодів проти дзвінка, які шунтують їх до негативної напруги і тим самим гасять коливання (рис. 2.13-2.16).

У випускається в даний час серії ТТЛ-елементів в ланцюзі емітера транзистора VT_1 замість резистора R_6 (рис. 2.13) розміщена резисторно-транзисторна ланцюг $VT_5 - R_3 - R_4$ (див. рис. 2.16). Це призначено для покращення форми статичних характеристик передачі. Справа в тому, що за схемою на рис. 2.13, характеристики передачі відрізняються від наведених на рис. 2.13. 2.18.

Ще до того, як вхідна напруга досягне порогового рівня $U^1_{\text{пор}}$, відбувається досить швидке падіння вихідної напруги U_{out} на ділянці АВ (рис. 2.18).

Це пояснюється тим, що коли транзистор VT_1 відкривається і через нього починають протікати струми колектора і емітера (при $U_{\text{вх}} = 0,6 - 0,65$ В), збільшення $U_{\text{вх}}$ призводить до зниження напруги на колекторі VT_1 . Це падіння рівня напруги передається через емітерний повторювач на транзисторі VT_2 на вихід (ділянка АВ). У той же час транзистор VT_3 все ще закритий, оскільки напруга на $R_{\text{недостатня}}$ через струм емітера VT_1 , що протікає через нього, щоб відкрити VT_3 .

Коли вхідна напруга U становить близько 1,4 В, напруга на базі VT_3 досягає рівня відкриття, коефіцієнт передачі (коефіцієнт підсилення) значно зростає, а характеристика падає різкіше в секції VS . На діаграмі на рис. 2.16 Емітерний і колекторний струми транзистора VT_1 не можуть виникнути до відкриття транзистора VT_5 . І він відкривається приблизно за тієї ж вхідної напруги, що й VT_3 . Отже, немає похилого перерізу АВ (штрихова лінія на рис. 2.17). Такі покращені характеристики пристрою гарантують більшу стійкість до перешкод елементів. Дійсно, тепер $U_{\text{вх}} < U^1_{\text{пор}}$ позитивна напруга перешкоди не передається на вихід.

У практичному застосуванні ІМС деякі входи елементів можуть виявитися непотрібними для цієї логічної схеми. Відповідно до принципу дії елемента І-НЕ, на ці входи слід подавати постійний сигнал 1. Це можна зробити різними способами: підключити резервний вхід до одного з використовуваних, подати напругу від джерела живлення (. зазвичай +5 В) до цього входу через резистор 2 - 3 кОм або залиште його вільним, не підключеним до жодної схеми. Бажано підключення до джерела живлення. При цьому не збільшується вхідний струм і знижується рівень перешкод, викликаних індукцією від зовнішніх електромагнітних полів. Елементи TTL охоплюють широкий діапазон значень швидкості та енергоспоживання. У той же час, чим вище передбачувана робоча частота, тим більше споживана потужність від джерела. Елементи серій К134 і К158 найбільш економічні, але розраховані на роботу на тактових

частотах лише до 3 МГц. Швидкісні елементи серії 130 або К131 можуть працювати на частотах до 30 МГц, але споживають набагато більше енергії. Елементи ТТЛШ, що мають підвищену швидкість, ефективно витісняють елементи ТТЛ. При такому ж або навіть меншому енергоспоживанні їх можна використовувати в цифрових вузлах з тактовою частотою до 50 МГц (серії 531, 1531). Всі елементи ТТЛ мають відносно високий опір; Допускаю статичну напругу перешкод до 0,5 В.

2.6 Логічні елементи з емітерним зв'язком (ЕЗ)

2.6.1. Спрощена схема і принцип роботи елементів ЕЗ

Елементи засновані на транзисторному ключі струму. Тому такі елементи іноді називають елементами ПСТЛ-елементами (транзисторна логіка на перемикачі струму). Схема комутатора струму (рис. 2.19,а) містить два транзистори VT_1 і VT_2 з резисторами R_{k1} і R_{k2} , які виконують роль колекторних навантажень.

Емітери транзисторів з'єднані разом, і в їх ланцюзі присутній резистор R_e , який забезпечує глибокий негативний зворотний зв'язок по струму. Зазвичай схема симетрична ($R_{k1} = R_{k2}$) і транзистори мають однакові параметри, наскільки це можливо.

якщо на бази транзисторів і U_{x2} подати вхідну напругу U_{w1} , а між колекторами підключити навантаження, то це буде т.з. Диференціальний підсилювач. Знак і величина напруги на навантаженні залежать від знака і величини різниці $U_{vx1} - U_{vx2}$. Подібна схема використовується, наприклад, на вході операційних підсилювачів.

Але в даному випадку ця схема працює в ключовому режимі, коли або через транзистор VT_1 (VT_2 закритий), або через транзистор VT_2 (VT_1 закритий) протікає приблизно однаковий струм. Розглянемо роботу вимикача струму докладніше.

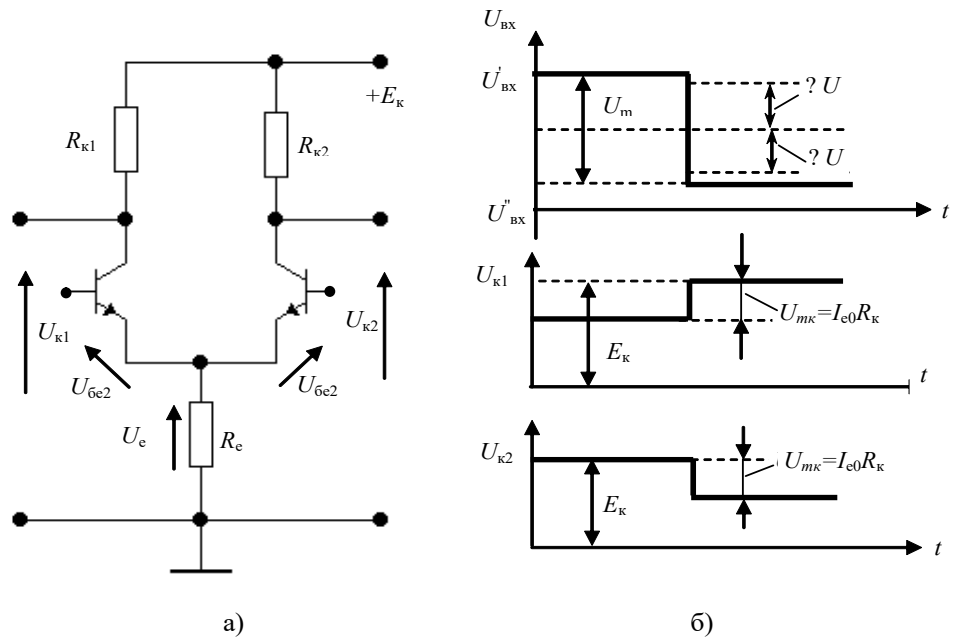


Рис. 2.19. Перемикач струму транзистора:
 а) – спрощена принципова схема; б) - часові діаграми

Транзистор VT_1 називається контролером, на базу якого подається вхідна керуюча напруга U . Транзистор VT_2 - опорний, на основі його роботи виникає постійна опорна напруга $U_{оп}$.

При $U_{вх} = U_{оп}$ емітерні струми транзисторів I_{e1} і I_{e2} будуть однаковими і дорівнюють половині сумарного струму I_{e0} , що протікає через резистор Re :

$$I_{e1} = I_{e2} = I_{e0} / 2.$$

Напруги на колекторах $U_{к1}$ і $U_{к2}$ також будуть однаковими. Але згідно із законом Ома $I_{e1} = (U_{вх} - U_{бе1}) / R_e$; $I_{e2} = (U_{вх} - U_{бе2}) / R_e$ і тому рівність $I_{e1} = I_{e2}$ можлива лише до тих пір, поки $U_{вх} = U_{оп}$.

Якщо $U_{вх}$ збільшується, то напруга на емітері, повторюючи $U_{вх}$, зростатиме. При постійному базовому потенціалі VT_2 , що дорівнює $U_{оп}$, це зменшить напругу $U_{бе2}$ на емітерному переході та зменшить струм колектора $I_{к2}$. Струм I_{e0} більше не буде рівномірно розподілятися між транзисторами, тому що більша його

частина тепер буде проходити через VT_1 . Коли внаслідок збільшення U_{kh} напруга U_e перевищить U_{op} , то при переході U_{be2} стане менше порогового U_{bo2} , тоді VT_2 закриється, весь струм I_{e0} потече через VT_1 і напруга U_{k2} будуть дорівнювати E_k . І навпаки, зменшення U відносно U_{op} призведе до зменшення U_{be1} (напруга U_e повторює найбільший потенціал на базах транзисторів) і, як наслідок, до зменшення колекторного струму I_{k1} перший транзистор. Велика частина струму I_{e0} тепер він буде протікати через VT_2 . Коли $U_{be1} < U_{bo1}$, транзистор VT_1 закриється і весь струм I_{e0} буде протікати через транзистор VT_2 , $U_{k1} = E_k$. Подальше зменшення вхідного значення U не впливає на струми транзистора.

Для перемикання струму з одного транзистора на інший необхідно дуже мале відхилення ΔU вхідної напруги U від значення U_{op} — десяті частки вольт. Падіння керуючої напруги $U_m \geq 2\Delta U$, що забезпечує комутацію струмів, не перевищує 1В. Зверніть увагу, що значне перевищення верхнього порогу на цю напругу неприпустимо, оскільки це може призвести до насичення транзистора VT_1 .

Величина падіння напруги на колекторах

$$U_{mk} = \alpha I_{e0} R_k \approx I_{e0} R_k$$

Вона перевищує $U_m = 2\Delta U$, що забезпечить керування наступним елементом за вихідною напругою попереднього при послідовному з'єднанні. Ця умова виконується підбором опорів резисторів R_{k1} , R_{k2} .

Графіки часу, що ілюструють рівні вхідної та вихідної напруги автоматичного вимикача, показані на рис. 2/19 р.н Верхній рівень вхідної напруги приймається за 1, нижній — за 0.

Спрощена схема логічного елемента EZL наведена на рис. 2.20, о.

На схемі присутній перемикач струму і замість одного керуючого транзистора VT_1 паралельно з'єднані три транзистора - $VT_1^{(1)}$, $VT_1^{(2)}$, $VT_1^{(3)}$, до якого вводяться керуючі сигнали x_1 , x_2 , x_3 , а також емітерні підсилювачі на транзисторах VT_3 і VT_4 .

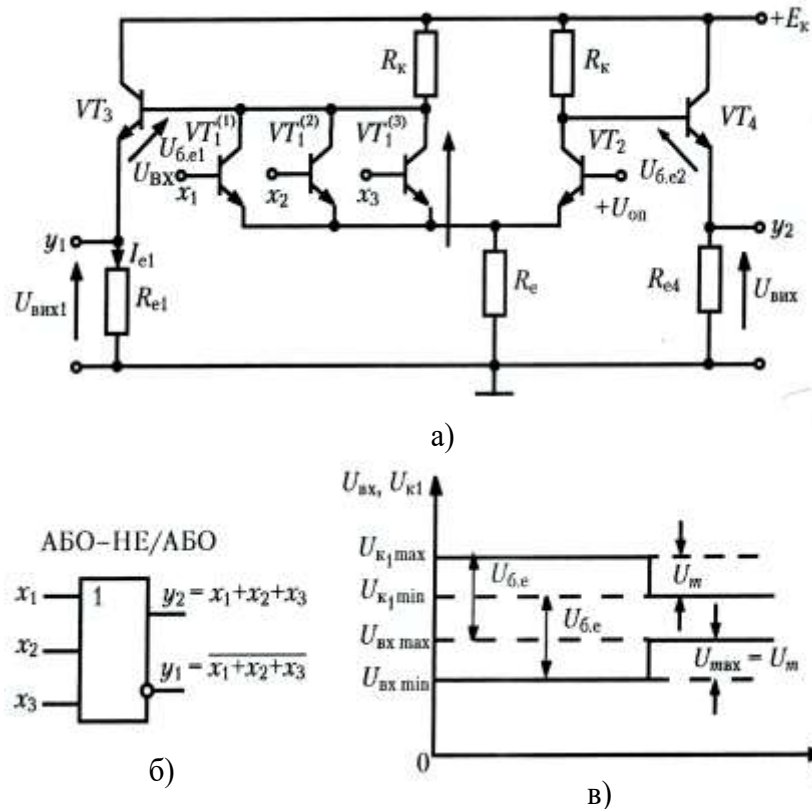


Рис. 2.20. EZL – елемент: а) – спрощена принципова схема;

б) – умовне позначення; в) – часові діаграми вхідної та колекторної напруг

Струм емітера I_e протікає через ліве або праве плече перемикача і визначається в основному значенням опору R_e , вибрані досить великі. Отже, значення струму I_e при проходженні через ліве плече мало залежить від того, відкритий один з транзисторів VT_1 , два з них або всі три. Тому вам просто потрібно подати сигнал $\log.1$ на один із входів, наприклад $x_1 = 1$, коли ліве плече відкривається і транзистор VT_2 . У цьому випадку вихідні сигнали будуть $y_1 = 0$, $y_2 = 1$. І тільки коли $x_1 = x_2 = x_3 = 0$, струм з лівого плеча перемикається на правий і виходи будуть $y_1 = 1$, $y_2 = 0$.

Отже, цей LE з трьома входами виконує операцію АБО-НІ на одному виході

$$y_1 = \overline{x_1 + x_2 + x_3},$$

а з іншого - АБО хірургічне втручання

$$y_2 = x_1 + x_2 + x_3.$$

Йому присвоєно назву елемента АБО-НІ/АБО та позначено на функціональних діаграмах, як показано на рис. 2.20, нар.

Емітерні підсилювачі в LE виконують подвійну функцію. По-перше, маючи низький вихідний опір, вони забезпечують швидкий заряд і розряд вихідних конденсаторів, збільшуючи тим самим навантажувальну здатність без зниження швидкості, по-друге, знижують рівень напруги колектора на величину U_{be} транзисторів VT_3 і VT_4 , що необхідно для узгодження їх з необхідними рівнями вхідної керуючої напруги наступних елементів. Якби вихідні напруги знімалися безпосередньо з колекторів, то, як показано на рис. 2.120, б, їх рівні на частку перевищували б вхідні. При цьому транзистор VT_1 керуючого елемента міг би не закриватися і навіть пішов би в насичення, що неприпустимо.

Низький вихідний опір емітерних підсилювачів, робота транзисторів без переходу в режим насичення, мале значення необхідної логічної різниці $U_m = I_e R_k$ дозволяє використовувати резистори $R_{для}$ малих опорів, що забезпечує високу швидкодію елементів EZL. Найкращі елементи EZL, такі як серія 1500, мають затримку поширення падіння t_{szr} менше 1 нс. Однак вони мають невелику логічну різницю, яка зменшує опір елемента, а значний струм I_{e0} , який протікає в елементі незалежно від його стану, викликає високі витрати енергії, що є істотним недоліком.

2.6.2. Специфіка схем реальних елементів EZL

Принципова схема одного з варіантів реального базового елемента EZL наведена на рис. 2.21а. Заземлення позитивного полюса джерела живлення колектора принципово не змінює роботу елемента. Змінюється лише потенційний опорний рівень (рис. 2.21б). Логічні рівні напруги U_{k1} і U_{k2} , що відповідають закритим станам транзисторів VT_1 і VT_2 , стають близькими до 0 при включенні живлення, і нестабільність напруги живлення мало впливає на їх рівні значень.

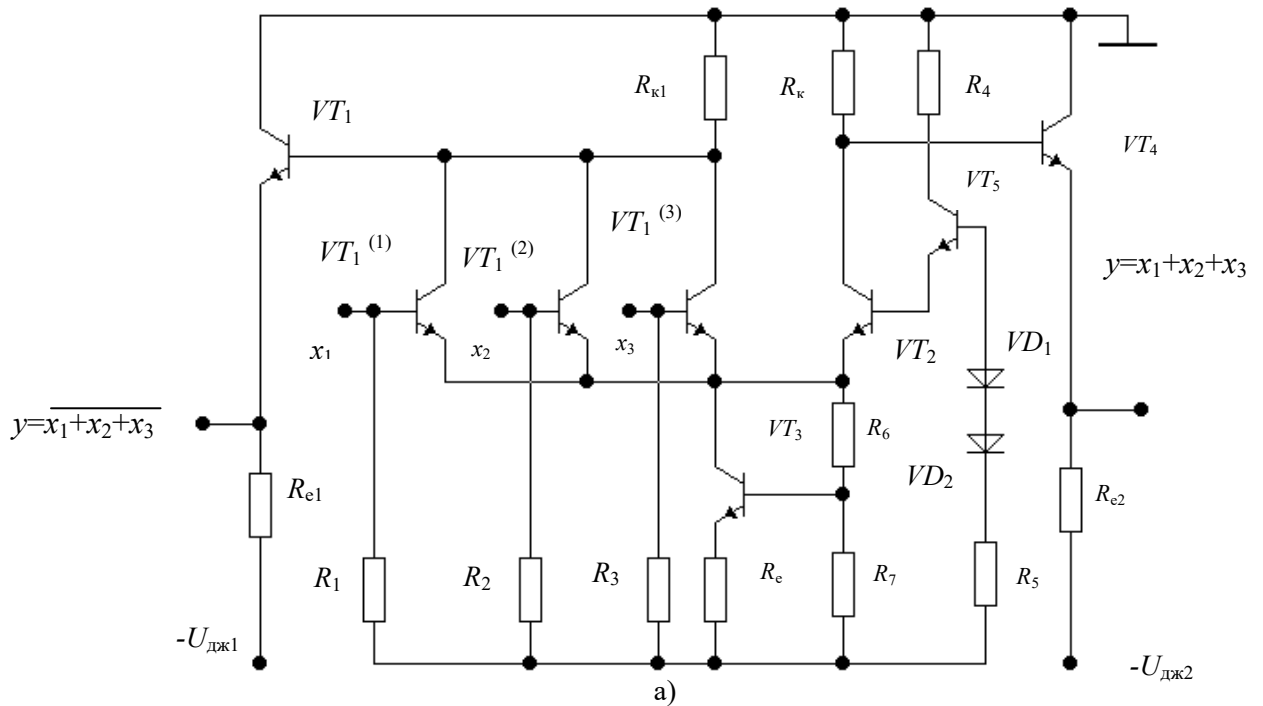


Рис. 2.21. Принципова схема фактичного базового елемента ESL (а) та рівні його вихідної напруги (б)

Опорна напруга $U_{оп}$ на основі транзистора VT_2 живиться від емітерного повторювача (обведеного пунктирною лінією), вбудованого в мікросхему (МС) і може керувати не одним, а кількома компонентами, що містяться в цій МС.

Діоди VD_1, VD_2 в базовому дільнику емітерного підсилювача забезпечують термостабілізацію опорної напруги. Підвищення температури викликає

зниження напруги на діодах, при цьому потенціал бази VT 5 зменшується, що забезпечує стабілізацію її колекторного струму, який зростає з підвищенням температури.

6 служить емітерним навантаженням перемикача струму, покритого глибоким негативним зворотним зв'язком по струму, викликаним резистором R_e . Постійна напруга подається на базу VT 6 від дільника R_6 - R_7 . Низький опір біполярного пристрою постійного струму забезпечує відповідно високе значення струму I_{e0} . При цьому двополюсний вимикач має дуже високий диференціальний опір. Отже, для керування транзисторами VT 1 і VT 2 необхідно змінювати напругу U_e за рахунок дуже малих приростів струму I_{e0} .

R_1 , R_2 , R_3 встановлені у входних колах керуючих транзисторів VT 1 (і), що дозволяє залишати входи невикористаними. Опір цих резисторів становить близько 50 кОм, і непідключений вхід еквівалентний подачі на нього логічного нуля. Резистори у вихідних емітерних підсилювачах з напругами VT 3, VT 4 всередині МК не встановлюються - схема з вільними емітерами. Можливість підключення резисторів R_{e1} , R_{e2} поза МК знижує потужність розсіювання всередині МК і зменшує його нагрів. До невикористаних виходів МС не підключаються зовнішні резистори, що знижує енергоспоживання.

R_{e2} на 50 Ом підключаються до повільних емітерів і підключається друге джерело живлення зниженої напруги U_{d2} , що зменшує споживання *електроенергії*. При цьому вихідні сигнали можна подавати безпосередньо на кабель з хвильовим опором 50 Ом.

З'єднуючи послідовно однотипні елементи, можна розмістити на виході високоомні резистори, підключивши їх до загального джерела живлення $U_{дж}$ 1.

Вільні випромінювачі розширюють логічні можливості елементів. Вони дозволяють з'єднати виходи кількох однойменних елементів (прямих або зворотних) і підключити їх до загального резистора R_e .

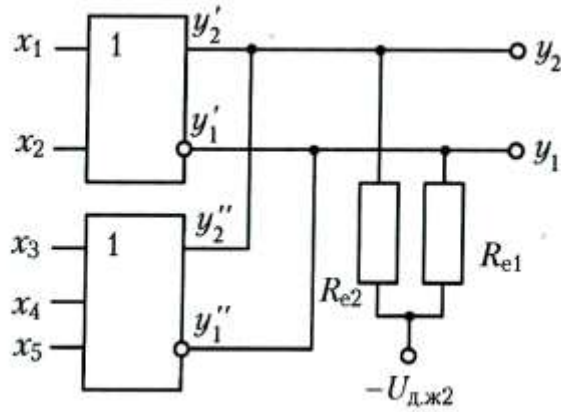


Рис. 2.22. Об'єднання елементів ЕЗЛ в OR та AND

На рис. 2.22 показує це співвідношення двох елементів. Якщо на виході обох елементів повинні бути однакові сигнали, вони будуть на загальному виході. Якщо один вихід дорівнює 1, а інший – 0, вихідний транзистор останнього замкнутий, оскільки різниця рівнів $U_{\text{вих}}^1$ і $U_{\text{вих}}^0$ перевищує напругу блокування U_0 . Таким чином, загальний результат буде 1. Це еквівалентно операції АБО. Комбінуючи прямі продукти, ви отримаєте:

$$y_2 = y'_2 + y''_2 = x_1 + x_2 + x_3 + x_4 + x_5 \text{ ("Збірка АБО")}$$

Об'єднавши зворотні виходи, отримаємо ("збірка І")

$$y_1 = y'_1 + y''_1 = \overline{x_1 + x_2 + x_3 + x_4 + x_5} = \overline{(x_1 + x_2)(x_3 + x_4 + x_5)}$$

Зверніть увагу, що таке підключення змінює рівні вихідних сигналів (зменшує різницю), тому кількість елементів не повинна перевищувати чотирьох. Найзручніше це робити в межах однієї держави-члена з вищим ступенем інтеграції.

2.6.3. Поєднання елементів EZL з елементами TTL

Одним з недоліків елементів ESL є те, що вони мають дуже малу, менше 1 В, різницю між рівнями вихідної напруги, що відповідають станам 0 і 1. Це особливо незручно при роботі з цифровими вузлами або пристроями кінцевих приводів у вигляді світлових індикаторів, обмоток реле і т. д. Як вже було сказано, для цього більше підходять TTL елементи, в тому числі з відкритим колектором. Тому МС, що містять логічні схеми на елементах EZL, мають на своїх виходах елементи кондиціонування перехідних процесів для перетворення рівнів сигналу для підключення до них елементів TTL.

Як приклад на рис. 2.23 показана спрощена схема елемента АБО-НІ з двома входами, який має входні рівні EZL і вихідні рівні TTL. Він складається з двовходового елемента EZL на транзисторах $VT^{(1)}_1$, $VT^{(2)}_1$, VT_2 , з негативним джерелом живлення $-U_{\bar{a}}$ і ключа на транзисторі VT_3 з позитивним джерелом живлення $+E_k$. Відмінність входної частини від звичайної схеми полягає в тому, що на транзисторі VT_2 є колекторна навантаження у вигляді діода VD.

Коли транзистор VT_2 відкритий ($x_1 = x_2 = 0$), на його колекторі виникає невелика негативна напруга, яка падає на відкритий діод і закриває транзистор VT_3 . Вихід має високий потенціал, тобто в системі TTL $y = 1$. Коли транзистор VT_2 закритий (сигнал 1 на одному або обох входах), струм від джерела E_k може протікати тільки через резистор R_6 і на базу транзистора VT_3 , яка стає насиченою (співвідношення R_6 і R_k вибрано таким чином), і тому $y = 0$. Тому конвертер виконує логічну операцію АБО-НІ:

$$y = \overline{x_1 + x_2} .$$

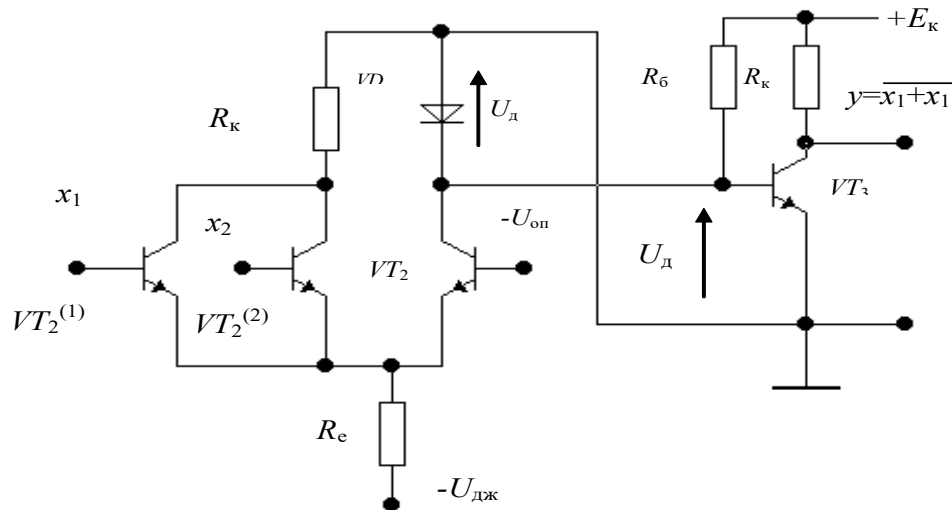


Рис. 2.23. Координаційний елемент ЕЗЛ-ТТЛ

При цьому, як показано, на входи подаються логічні рівні напруги, характерні для елементів ЕЗЛ, а на виході отримують рівні напруги, характерні для елементів ТТЛ.

2.7 Логічні елементи на польових транзисторах

транзистори МДН з наведеними каналами *n*- або *p*- типу, в яких під дією вхідної керуючої напруги канал індукується (індукується) тільки після досягнення напругою певного порогового значення. Ключові каскади на таких транзисторах можна з'єднувати безпосередньо, з'єднуючи вихід одного каскаду з входом іншого без будь-яких елементів зв'язку.

2.7.1. Логічні елементи на *n*- канальних транзисторах МДН

В основі ЛЕ лежить транзисторний ключ із загальним витоком і нелінійним навантаженням (рис. 2.24, а). Перемикачі транзисторні VT_1 . Транзистор VT_2 є транзистором навантаження, його затвор з'єднаний зі стоком, завдяки

чому він перетворюється в біполярний транзистор з нелінійною вольтамперною характеристикою. Коли напруга на вході ключа менше порогового значення U_{z0} , як це видно з вхідної характеристики (рис. 2.24, б), перемикаючий транзистор закритий.

Струм стоку через транзистори залежить тільки від теплових струмів, а транзистор VT_2 близький до відкриття. Це пояснюється тим, що транзистор VT_2 не може бути ні відкритим, тому що шлях струму джерела перекриває закритий транзистор VT_1 , ні закритим, тому що тоді напруга на ньому перевищить значення U_{z0} і він відкриється. Тому падіння напруги на ньому становить $U_n = U_{zv} \approx U_{z0}$, тобто воно знаходиться на межі відкриття та на виході ключа $U_{вих} = U_{дж} - U_{z0}$. На практиці $U_j = 9$ В, а U_z має значення 1,5–2,0 В. Якщо на вхід подати напругу U_{max} , цього достатньо для переміщення VT_1 у крутий (тріодний) регіон вихідної характеристики (Рис. 2.24, в), то його опір буде менше опору двополюсного навантаження і можна отримати $U_{вих.мін} < U_{z0}$.

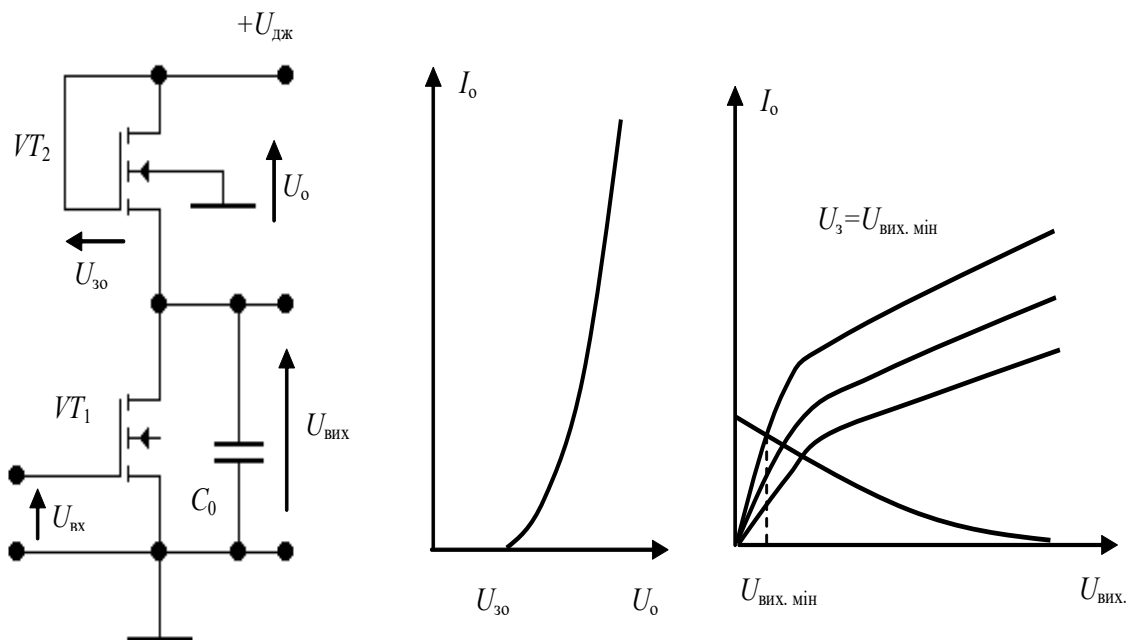


Рис. 2.24. Манулювання на транзисторах МДН з нелінійним навантаженням

а) – принципова схема; б) – вхідні характеристики;

в) – вихідні характеристики

Зверніть увагу, що для забезпечення необхідного співвідношення опорів канал транзистора VT_2 вужчий, ніж канал транзистора VT_1 . Високий рівень напруги на виході замкнутого ключа приймається за рівень логічної 1 ($U_{вих.мах} = U_{вих}^1$), низький рівень напруги на виході відкритого ключа приймається за рівень логічного 0 ($U_{вих.мін} = U_{вих}^0$). Умова $U_{вих}^0 < U_{зо}$ гарантує, що один ключ керує іншим ключем, коли вони безпосередньо з'єднані послідовно.

Логічний елемент АБО-НІ створюється паралельним увімкненням двох або більше перемикаючих транзисторів, які працюють на спільне навантаження. На рис. 2.25, схема двовхідного елемента. Перемикаючі транзистори ми будемо вважати ключами: транзистор відкритий - ключ закритий, транзистор закритий - ключ відкритий. Якщо на обидва входи подається низький логічний рівень 0 ($x_1 = x_2 = 0$), транзистори $VT_1^{(1)}$ і $VT_1^{(2)}$ закриті, обидва ключа відкриті (рис. 2.24,а) і на виході високий рівень $U_{вих}^1$, тобто $y = 1$.

Достатньо хоча б на один із входів подати високий рівень 1, наприклад $x_2 = 1$, коли транзистор $VT_1^{(2)}$ відкривається, тобто ключ закривається, і залишкова напруга на виході стає низькою: $U_{out}^0 < U_{зо}$, тобто $y = 0$.

Логічний елемент І-НЕ утворений двома або кількома перемикаючими транзисторами, з'єднаними послідовно із спільним навантаженням (рис. 2.25, б). У цій схемі обидва ключі перемикача будуть замкнуті, створюючи шлях для струму і забезпечуючи низький рівень вихідної напруги $U_{вих}^0$, тільки коли $x_1 = x_2 = 1$.

Якщо на один із входів подати 0, наприклад $x_2 = 0$, то перемикаючий транзистор $VT_1^{(2)}$ закриється, струм в ланцюзі припиниться, а на виході з'явиться високий рівень напруги $U_{вих}^1$, іншими словами вихідний сигнал y виходить із співвідношення $y = x_1 x_2$.

Коефіцієнт зв'язку на вході $K_{об}$ не великий, особливо в елементах І-НЕ: при більш ніж двох послідовно з'єднаних транзисторах залишкова напруга $U_{вих.мін}$ може виявитися дуже великим, більше $U_{зо}$.

Навантажувальна здатність всіх типів елементів MDN висока, хоча їх вихідний струм низький. Це пояснюється дуже високим входним опором транзисторів MDN (10^{12} Ом і більше). K_{timesg} досягає 20. Пам'ятайте, що збільшення кількості елементів навантаження, а також збільшення кількості входних транзисторів, з'єднаних паралельно, призводить до збільшення паразитних ємностей і, як наслідок, зниження швидкодії. При оцінці швидкості роботи інерційністю транзисторів МДН можна знехтувати, оскільки вони не накопичують і не розсіюють заряди. Швидкість роботи залежить від процесу зарядки і розрядки паразитних конденсаторів через значний опір транзисторів. Особливо це стосується навантажувального транзистора, де, як уже було сказано, канал стає більш вузьким. Тому заряд вихідного конденсатора C_0 до VT_2 протікає повільніше, ніж розряд через VT_1 .

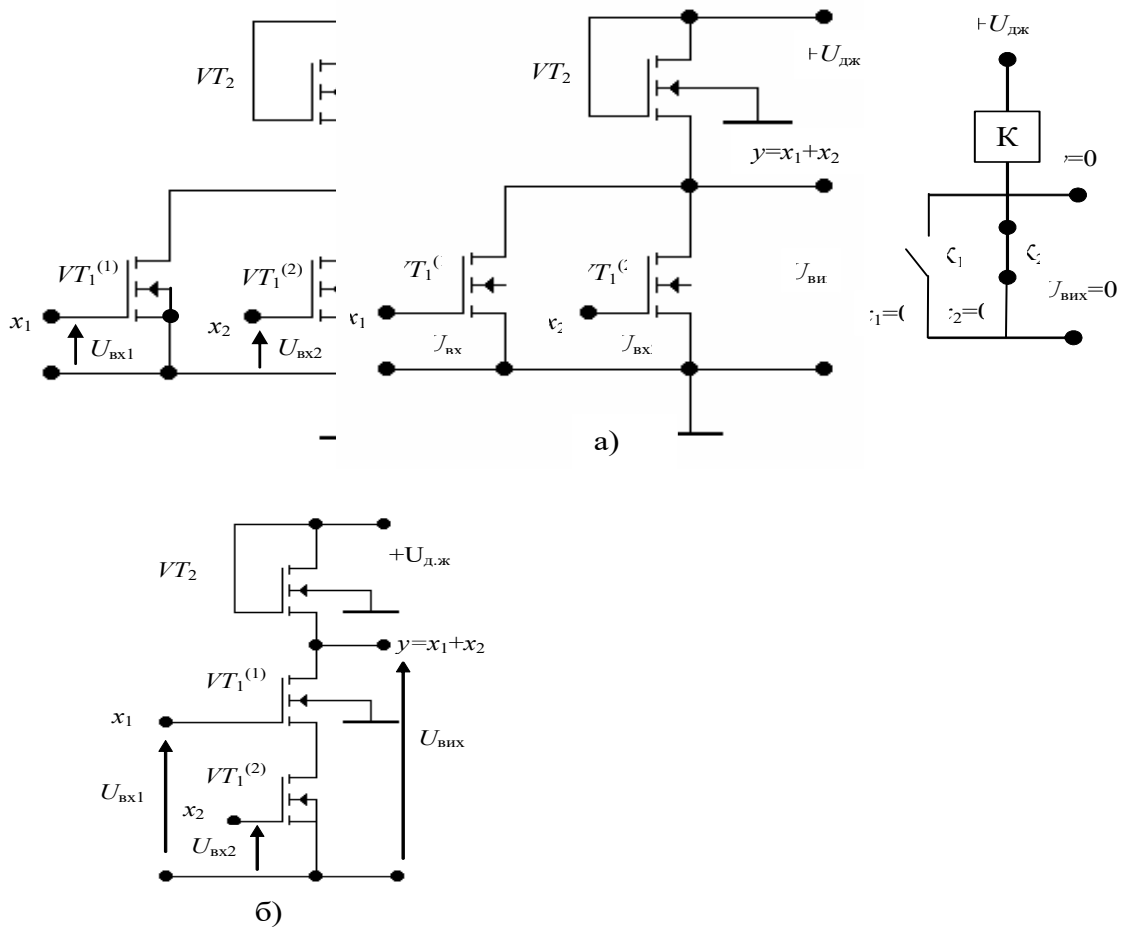


Рис. 2.25. Елемент MDN з нелінійним навантаженням:

а) типу АБО-ПІ ; б) напишіть І-ПІ

2.7.2. Логічні елементи на комплементарних парах транзисторів МДН

Основою елементів КМДН є транзисторний ключ, утворений двома транзисторами МДН з каналами різного типу провідності (рис. 2.26,а). Транзистори з'єднані стоками. Вихід n -канального транзистора VT_1 з'єднаний із загальною шиною, «+» джерела живлення - з виходом p -канального VT_2 . При такій схемі напруга витоку затвора $U_{зв1}$ транзистора VT_1 дорівнює вхідній напрузі $U_{вх}$, а напруга витоку затвора $U_{зв2}$ транзистора VT_2 відрізняється величиною напруги джерела живлення U_j , тобто

$$U_{зв1} = U_{вх}, U_{зв2} = U_{вх} - U_{дж}.$$

Якщо вхідна напруга U_{in} надходить від виходу подібного елемента, вона змінюється між рівнем $U_{вх.мах} \approx U_{дж}$, який приймається за 1, і рівнем $U_{вх.мін} \approx 0$, який приймається за 0.

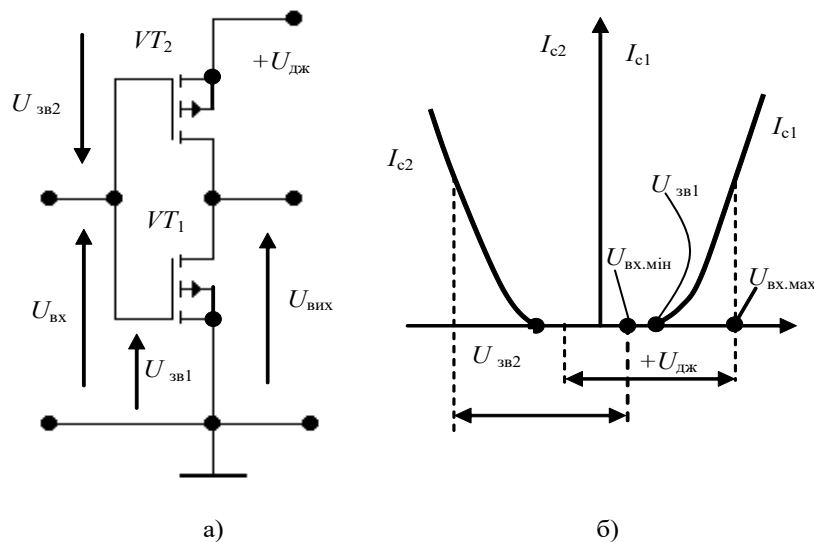


Рис. 2.26. Ключ у вигляді комплементарної пари транзисторів МДН:

а – принципова схема; б - характеристики транзисторів

При подачі на вхід $U_{вх.мах}$ VT_1 відкритий, а VT_2 закритий, тому що його напруга $U_{зв2}$, хоча й негативна, не досягає порогу відкриття $U_{зв2}$. Коли

$U_{\text{вх}} = U_{\text{вх.мін}} < U_{\text{зол}}$ VT₁ закривається і негативна напруга на затворі VT₂ велика, він відкривається. Це проілюстровано на рис. 1. 2.26, б, де показані статкові характеристики обох транзисторів і показані рівні вхідної напруги. Таким чином, в будь-якому стані один з транзисторів закритий, наскрізний струм в ланцюзі живлення відсутній. Напруга на виході ключа U змінюється від моменту $U_{\text{вих.мін}} \approx 0$ відкриття $U_{\text{вих.мак}} \approx U_{\text{дж}}$ транзистора VT₁ до закриття VT₁. Падіння вихідної напруги близьке до U_j . Щоб уникнути стану, в якому обидва транзистори були б закорочені під час процесу перемикавання, значення напруги живлення повинно відповідати умові

$$U_{\text{д}} > U_{\text{від 01}} + | U_{\text{від 02}} | .$$

Логічний елемент АБО-НІ реалізується за допомогою двох і більше ключів КМДН шляхом паралельного перемикавання *n*-канальних транзисторів і послідовного (рівневого) перемикавання *p*-канальних транзисторів. На рис. 2.26 показана схема двовходового елемента АБО-НІ. Якщо на входи подаються низькі рівні, тобто $x_1 = x_2 = 0$, то транзистори VT₁⁽¹⁾ і VT₁⁽²⁾ закриті (ключі відкриті), а транзистори VT₂⁽¹⁾ і VT₂⁽²⁾ відкриті (ключі закриті).

Сила струму в ланцюзі живлення визначається виключно струмами витоку. Напруга на виході $U_{\text{вих}} \approx U_j$, тобто $y = 1$. При подачі хоча б на один із входів, наприклад $x_1 = 1$, транзистор VT₁⁽¹⁾ відкривається і VT₂⁽¹⁾ переходить у закритий стан (див. рис. 2.27).

Струм у ланцюзі живлення ще дуже малий, а вихідна напруга $U_v \approx 0$, тобто $y = 0$.

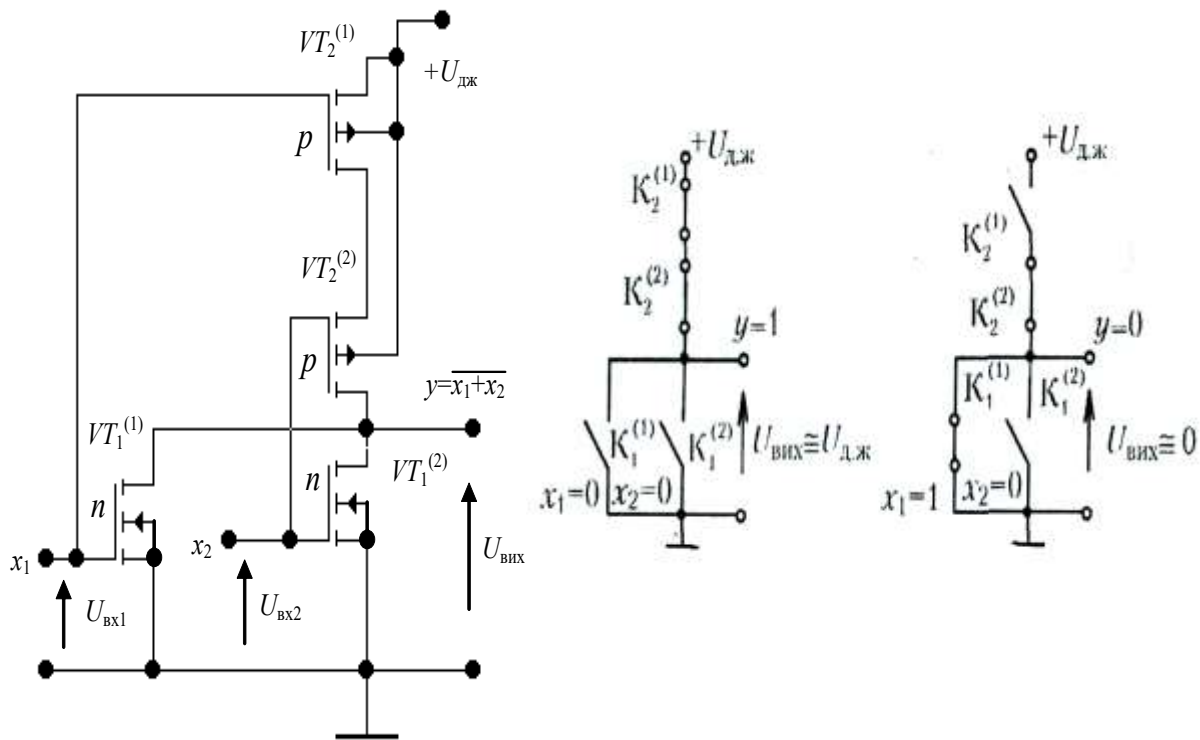


Рис. 2.27. Елемент КМДН АБО-НІ: принцип і еквіваленти схеми ключової схеми

p - каналні транзистори включені паралельно, а n - каналні транзистори - послідовно (рис. 2.28).

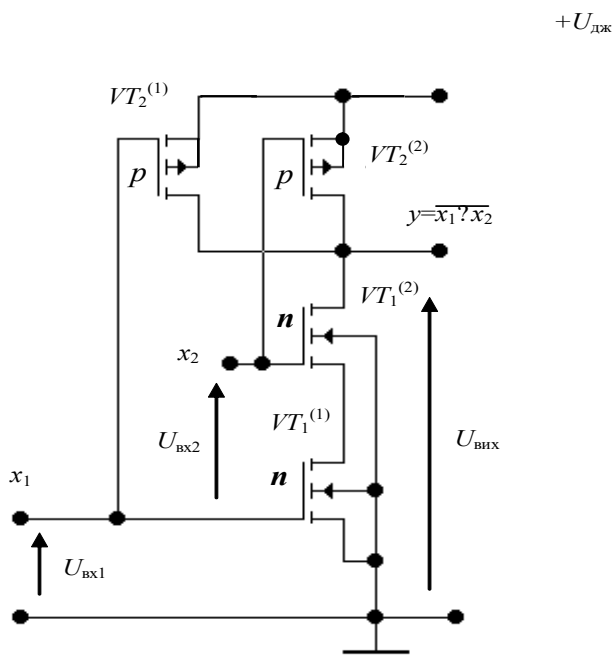


Рис. 2.28. Тип елемента КМДН АБО-НІ

Якщо $x_1 = x_2 = 1$, то транзистори $VT_1^{(1)}$, $VT_1^{(2)}$ відкриті, а $VT_2^{(1)}$, $VT_2^{(2)}$ закриті. Вихідна напруга близька до 0, тобто $y = 0$. Якщо принаймні один вхід дорівнює 0, наприклад $x_1 = 0$, транзистор $VT_1^{(1)}$ закривається і $VT_2^{(1)}$ переходить у відкритий стан. При цьому $U_{\text{вих}} = U_{\text{zh}}$, тобто $y = 1$.

Основна перевага елементів КМДН полягає в тому, що в обох статичних станах струм від джерела живлення практично відсутній, а значить споживана потужність дуже мала. Однак під час роботи елемента струм йде на заряд паразитних конденсаторів, тому динамічна споживана потужність пропорційна частоті комутації і може на кілька порядків перевищувати статичну потужність. Швидкість роботи ЛЕ на структурах КМДН вище, ніж ЛЕ на транзисторах МДН з однотипними каналами, тому що тут як заряд, так і розряд вихідної ємності відбувається через відкриті транзистори VT_1 і VT_2 , які мають однаковий низький опір. Але і ті, і інші відносяться до класу тихохідних елементів. Допустима частота комутації не перевищує 5 МГц.

Кількість входів ($K_{об}$) і вантажопідйомність (K_{open}) тут також обмежені в основному впливом цих показників на швидкість. Паралельно включені транзистори збільшують паразитну ємність, а послідовно збільшують опір, через який ємність заряджається або розряджається. Деякі серії елементів КМДН, наприклад серія 564, допускають використання джерел живлення з напругою від 3 до 15 В. Це визначає швидкість роботи і енергоспоживання. При $U_j = 5$ В такі елементи сигнально сумісні з елементами TTL.

Важливою перевагою елементів КМДН є їх висока стійкість до перешкод. Для тієї ж серії 564 допускається статичний шум 2,5 В.

2.7.3 Логічний елемент КМДН з трьома вихідними станами

Принципова схема елемента наведена на рис. 2.29, а. Транзистори VT_1 і VT_2 утворюють інвертуючий ключ КМДН. Він відрізняється від зображеного на рис. 2.26 полягає в тому, що за допомогою додаткових транзисторів VT_3 з

n -канал і VT_4 з p -каналом, його можна відключити від джерела живлення. Взаємно інверсні керуючі сигнали EZ і \overline{EZ} .

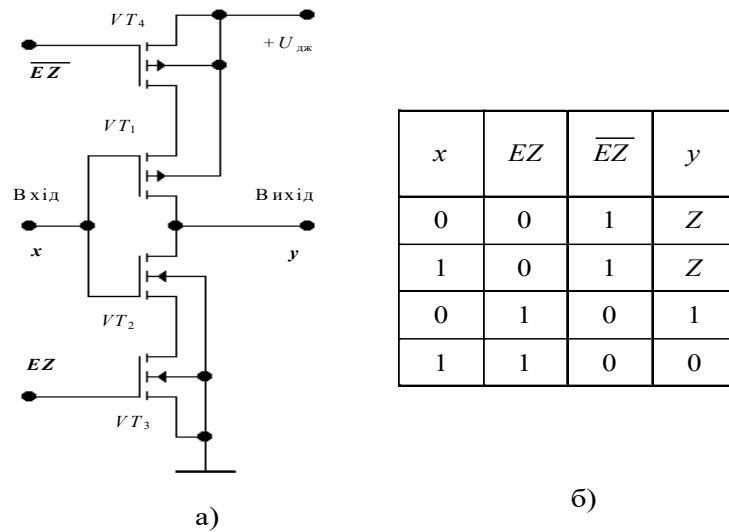


Рис. 2.29. Елемент КМДН з трьома початковими станами:

а – принципова схема; б - таблиця стану

Якщо вхід EZ низький 0, а вхідний сигнал \overline{EZ} високий 1, транзистори VT_3 і VT_4 закриті, на ключ не подається живлення, а вихідний контакт має дуже високий опір як шині живлення, так і спільній землі. залізниця.

Вхідний сигнал не проходить через елемент. Після подачі логічного рівня 1 на вхід EZ і \overline{EZ} рівня 0 на вхід транзистори VT_3 і VT_4 відкриваються, ключ отримує живлення U_a , і напруга на його виході залежить від вхідного сигналу. При цьому вихідний опір елемента в будь-якому зі станів буде визначатися опором двох відкритих транзисторів, з'єднаних послідовно. Таблиця станів елементів в залежності від комбінації керуючих сигналів наведена на рис. 2.29, нар.

2.7.4. Особливості схем ЛЕ на польових транзисторах та їх узгодження елементи TTL

Логічні елементи, побудовані на транзисторах МДН, у порівнянні з ЛЕ інших типів мають певні особливості, що зумовлені властивостями цих транзисторів. Затвор транзистора МДН і підкладка, розділені шаром діелектрика, утворюють конденсатор. Величезний опір витoku цього конденсатора, близько 10^{12} Ом, створює сприятливі умови для накопичення на ньому статичних зарядів, що може призвести до незворотного пошкодження шару діелектрика. При цьому затвор транзистора не повинен залишатися вільним, щоб не було гальванічного зв'язку з загальною рейкою або з шиною живлення. Транзистори, встановлені на входах логічної схеми, найбільш схильні до діелектричного пробою, оскільки їх затвори з'єднані з висновками МС.

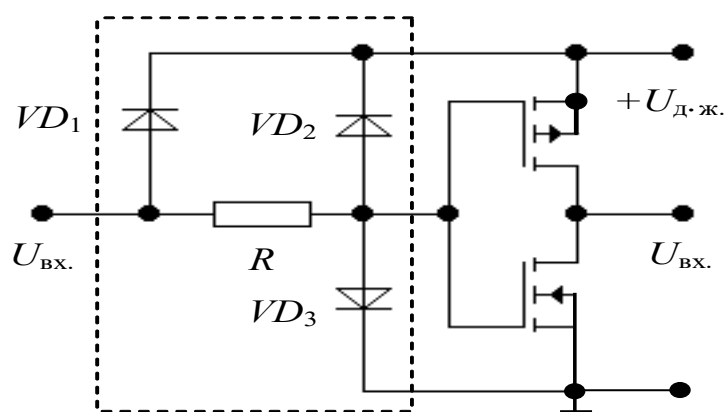


Рис. 2.30. Схема захисту вхідного транзистора MDN

Для захисту таких транзисторів від пошкодження високою напругою в єдиному технологічному процесі виробництва ЛЕ кожен його вхід оснащений діодно-резистивною схемою захисту - рис. 2.30 (на схемах ЛЕ, як правило, схеми не показані). Залежно від значення і полярності напруги перевантаження на вході діоди або проводять в прямому напрямку, або переходять в режим лавинного пробою, що відбувається при зворотній напрузі 30-35 В. Діоди лавинного пробою мають протилежну природу. і не впливають на роботу

ЛЕ. Діоди замикають підвищену вхідну напругу або на джерело живлення U_j , або на загальну шину. У нормальних умовах експлуатації, коли значення вхідної напруги знаходиться в межах норми - $0,7V < U_{вх\text{ид}} < (U_d + 0,7V)$, діоди не відкриваються.

Під час зберігання наконечники МС повинні бути з'єднані між собою - загорнуті в металеву фольгу.

підключаються безпосередньо до виходу ТТЛ (ТТЛШ) - елемента з однаковою напругою живлення $U_j = 5V$. Важливо лише, щоб вихід елемента ТТЛ не навантажувався одночасно входами інших елементів ТТЛ, оскільки їх вхідний струм може знизити напругу на виході попереднього елемента до рівня, недостатнього для відкриття польових транзисторів. Для узгодження вихідної потужності елемента ТТЛ з входами елементів КМДН при подачі на останній підвищеної напруги використовується елемент ТТЛ з повільним колектором, подавши на його вихідний транзистор через резистор напругу джерела живлення. елементів КМДН (рис. 2.31).

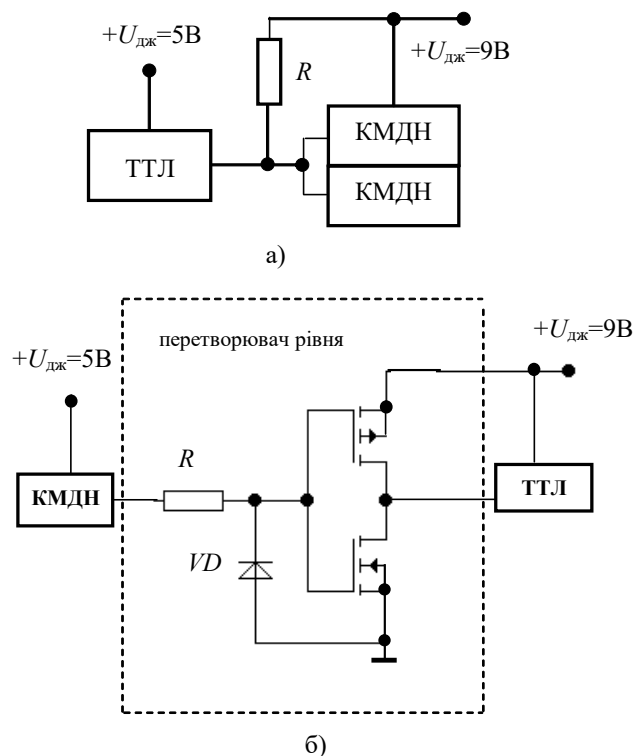


Рис. 2.31 . Підключення елемента ТТЛ до виходу елемента КМДН при різних напругах живлення

Якщо сигнали з виходу елемента КМДН повинні контролюватися елементами TTL, то при їх живленні від одного джерела $U_j = 5В$ достатньо забезпечити узгодження струмів між елементами. Вхідний струм елемента TTL при 0 на вході $I_{вх}^0$ зазвичай перевищує пропускну здатність елемента КМДН. В якості узгоджувального елемента з підвищеною навантажувальною здатністю в стані 0 може бути використаний елемент АБО-НІ, в якому з'єднані всі входи. Як видно з рис. 2.27, навантажувальна здатність такого елемента визначається кількістю паралельно з'єднаних *n*-канальних транзисторів, тобто в даному випадку вона подвоюється.

Якщо $U_{ж.КМДН} > U_{ж.TTL}$, необхідно узгодити і напругу, і струм.

Узгодження напруги забезпечується вмиканням перетворювача рівня, який живиться від спільного джерела з елементом TTL $U_j = 5В$ (рис. 2.31). Відмінність перетворювача від інвертора в тому, що в ланцюзі захисту між затвором і шиною живлення відсутні діоди. Завдяки цьому, не порушуючи роботу перетворювача, можна отримати на його вході напругу, що перевищує напругу живлення. Максимальне навантаження такого перетворювача становить не більше одного входу елемента TTL.

2.7.5. Концепція логічних елементів з потужністю інжекції

ЛЕ з потужністю інжекції (integral injection logic ІІЛ, І²Л) побудовані на основі т.з. поздовжні та поперечні структури транзисторів, виготовлені методами інтегральної технології. Не вдаючись у подробиці електронних процесів у такій структурі, зобразимо її елементарну ланку у вигляді двох транзисторів VT₁ і VT₂ (рис. 2.32, а). Джерело живлення U_j з низькою напругою, достатньою лише для відкриття емітерного переходу, підключено до емітера VT₁, а струм інжекції I постійно протікає через резистор R . Транзистор VT₁ можна розглядати як генератор струму (інжектор), що керує роботою транзистора VT₂. Схема заміщення клітин наведена на рис. 2.32 р.н Вхідний ланцюг може мати два стани: розрив або замикання на корпус (показано пунктиром). Перший стан приймаємо за 1, другий за 0.

Якщо ланцюг розривається на вході (1), то струм інжектора I_i , що надходить на базу транзистора VT, насичує його. Між точками KE буде дуже малий опір, тобто коротке замикання (0). Якщо вхідний ланцюг замкнутий (0), через нього протікає струм інжектора і закритий транзистор VT, то вихідний ланцюг KE буде відкритий (1). Отже, ця комірка є інвертором.

При послідовному з'єднанні комірок вхідним станом є стан попереднього транзистора: закритий (відкритий) або насичений (коротке замикання). Об'єднуючи такі інвертори в різні схеми з прямим підключенням, можна отримати логічні елементи із заданою функціональністю.

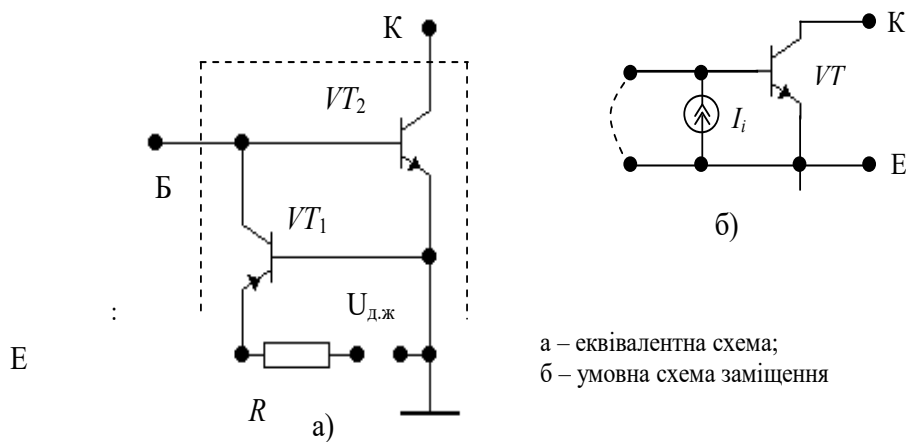


Рис. 2.32. Елементарна комірка з потужністю інжекції

Якщо інвертори включені паралельно (рис. 2.33а), то достатньо подати 1 хоча б на один із входів (розірвати вхідну ланцюг), щоб відповідний транзистор пішов у насичення, тобто вихід закрився. (0). Тому цей елемент виконує операцію OR $y = \overline{x_1 + x_2}$. При послідовному з'єднанні з виходом інвертора (рис. 2.33, б) створюється елемент АБО: $y = x_1 + x_2$.

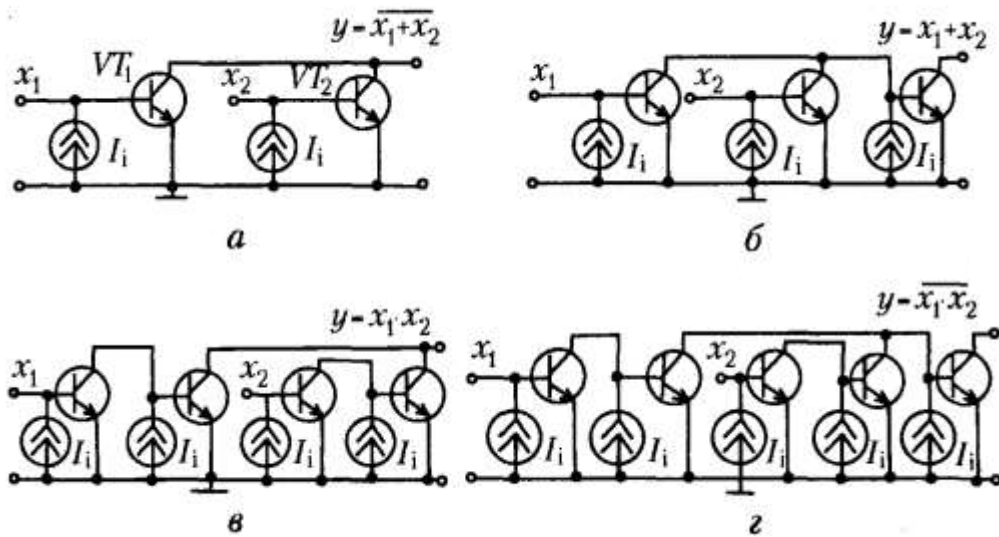


Рис. 2.32. Логічні елементи типу ІІІ:

а – елемент АБО-НІ ; б – елемент АБО ; в – елемент І ; г – елементом І-НЕ

Паралельне з'єднання інверторів, входи яких живляться не безпосередньо, а через аналогічні інвертори (рис. 2.33, в), створює елемент, що реалізує операцію І. Це є результатом перетворення за теоремою де Моргана:

$$y = \overline{\overline{x_1} + \overline{x_2}} = x_1 \cdot x_2$$

Підключивши до виходу інший інвертор, ми отримаємо елемент І-НЕ:
 $y = \overline{x_1 \cdot x_2}$ (рис. 2.33, г).

Розглянуті елементи працюють без додаткових джерел живлення і без навантажувальних резисторів в ланцюгах колектора. Тому вони відрізняються дуже низьким енергоспоживанням, і МК на їх основі можна виготовляти з дуже високим ступенем інтеграції. Використовуються переважно в мікропроцесорних установках (серії 582, 583, 584). На входах і виходах типу МС ІІІ вони мають спеціальні перетворювачі для узгодження з елементами ТТЛ, в яких сигнали 1 і 0 відображаються як рівні напруги і мають відповідно високу вихідну потужність.

РОЗДІЛ 3. КОМПЛЕКСНІ ЦИФРОВІ ПРИСТРОЇ

КОМБІНАЦІЙНОГО ТИПУ

3.1 Загальні відомості про комбіновані пристрої

Логічні пристрої можна класифікувати за різними ознаками. За способом введення/виведення інформації логічні пристрої поділяються на послідовні, паралельні та послідовно-паралельні. Пристрій, в якому вхідні змінні подаються на вхід, а вихідні змінні знімаються з виходу не одночасно, а послідовно, крок за кроком, називається послідовним пристроєм. Паралельним пристроєм називається пристрій, в якому всі розряди вхідних змінних подаються на вхід і всі розряди вихідних змінних одночасно видаляються з виходу. У послідовно-паралельних пристроях вхідні та вихідні змінні представлені в різних формах. Або змінні послідовно подаються на вхід символ за символом і одночасно видаляються з виходу, або навпаки.

За принципом дії всі логічні пристрої поділяються на два класи: комбінаційні та послідовні.

Комбінаційні пристрої або автомати без пам'яті - це логічні пристрої, вихідні сигнали яких однозначно визначаються тільки комбінацією змінних, що діють на вхід в даний момент часу, і не залежать від міток змінних, які раніше діяли на вхід.

Послідовні пристрої, або автомати з пам'яттю, - це логічні пристрої, вихідні сигнали яких визначаються не тільки комбінацією змінних, що діють на вхід в даний момент, але і послідовністю вхідних змінних, які були активними в попередні моменти.

Комбінаційну схему можна представити у вигляді полярного елемента m - k (рис. 3.1, а). Вхідне слово (вхідний алфавіт) комбінаційної схеми задається набором символів $M = (m_1, m_2, \dots, m_i)$, а вихідне слово (вихідний алфавіт) приймає значення вихідних символів $K = (k_1, k_2 \dots k_j)$. У дискретний момент набір вихідних сигналів однозначно визначається набором вхідних сигналів,

які надійшли на вхід в той самий момент. Комбінаційну схему можна охарактеризувати: кількістю вхідних сигналів, кількістю вихідних сигналів, логічною формулою або таблицею істинності. Для керування роботою комбінованої схеми вводяться керуючі тактові сигнали, щоб після завершення переходів з'являвся новий набір вхідних сигналів.

Якщо на вихідні сигнали не впливає внутрішній стан комбінаційної схеми, а лише зміна вхідних сигналів, то таку схему називають автоматом без пам'яті або примітивним автоматом.

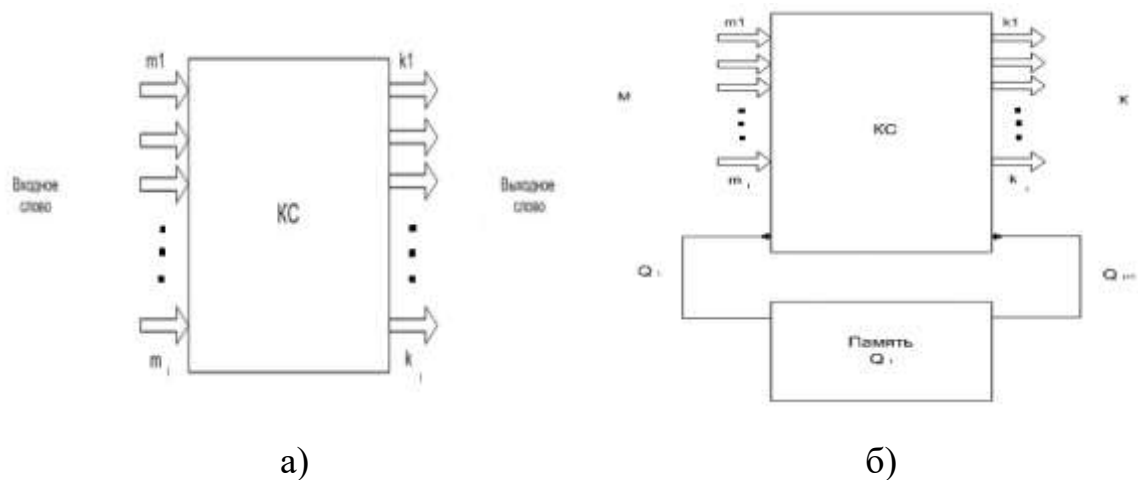


Рис. 3.1 Комбінована схема у вигляді полюсного елемента $m - k$

Якщо на сукупність вихідних сигналів K впливає не тільки сукупність вхідних сигналів, але і внутрішній стан комбінаційної схеми, то таку схему називають автоматом з пам'яттю або повним автоматом (рис. 3.1, б). Автомат з пам'яттю визначається трьома наборами змінних: M, K, Q , де Q - набір змінних, що відображає внутрішній стан схеми.

У більшості схем повних автоматів комбінація комбінаторних елементів дозволяє створити пристрій, вихідні сигнали якого не будуть прямо залежати від вхідних сигналів, а лише від комбінації вхідних сигналів і внутрішнього стану автомата в момент надходження вхідних сигналів. Тому для опису роботи машини в таблиці істинності вказується набір вхідних сигналів, набір відповідних внутрішніх станів і набір вихідних сигналів. Крім того, умова роз-

ділення всіх наборів сигналів за часовими інтервалами є обов'язковою. Внутрішній стан машини до надходження вхідного сигналу Q_0 , внутрішній стан машини, що змінюється з надходженням вхідного сигналу Q_i , і внутрішній стан машини після того, як вхідний сигнал Q_{i+1} знову змінюється, є спеціально обговорюється.

Існує два види машин. Автомат, в якому вихідна змінна в будь-якому циклі t_i залежить від внутрішнього стану, а вхідна змінна називається автоматом Міллі і виражається рівнянням: $K = f(Q_i, M_i)$.

Автомат, вихідна змінна якого в момент часу t_i залежить тільки від внутрішнього стану в t_{i-1} -му циклі, називається автоматом Мура і задається рівнянням $K_j = f(Q_i)$.

3.2. Суматори

3.2.1 Принцип побудови однорозрядного суматора

Додавання використовуються для виконання додавання багаторозрядних чисел. Зазвичай числа подають у прямій, доповнювальній або оберненій двійковій формі, рідше – у двійково-десятковій формі.

Надбудови можуть бути накопичувального та комбінованого типу.

Накопичувальні суматори можуть працювати з декількома (двома або більше) числами, які надходять на їхні входи в різний час, комбінаційні суматори працюють лише з двома числами, які надаються одночасно. Рис. 3.2 наведено умовний графічний запис комбінаційного суматора для двох трицифрових чисел $A\{a_2, a_1, a_0\}$ і $B\{b_2, b_1, b_0\}$. Його виходи створюють результат додавання S і сигнал перенесення P .

Високорозрядні суматори побудовані на основі однорозрядних суматорів, з'єднаних ланцюгами передачі. Залежно від типу ланцюгів передачі розрізняють суматори з послідовними і прискореними передачами.

Половина суматора. Найпростішим елементом додавання є напівсуматор (рис. 3.3). Він має два входи для одноцифрових чисел, що складаються з А і В, і два виходи: С (сума) і Р (перенесення).

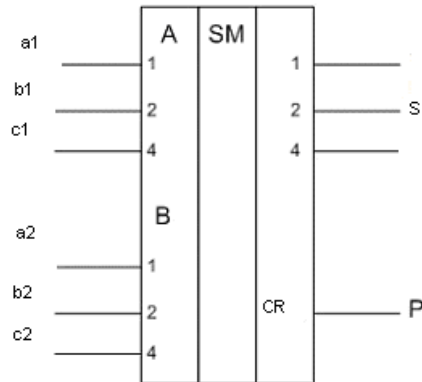


Рис. 3.2 Умовне графічне позначення

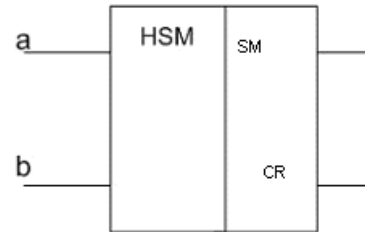


Рис. 3.3 трирозрядний суматор і напівсуматор

Його функціональне призначення відображено в мнемосхемі HSM (половина суми), а робота – у таблиці істинності 3.1.

Таблиця 3.1

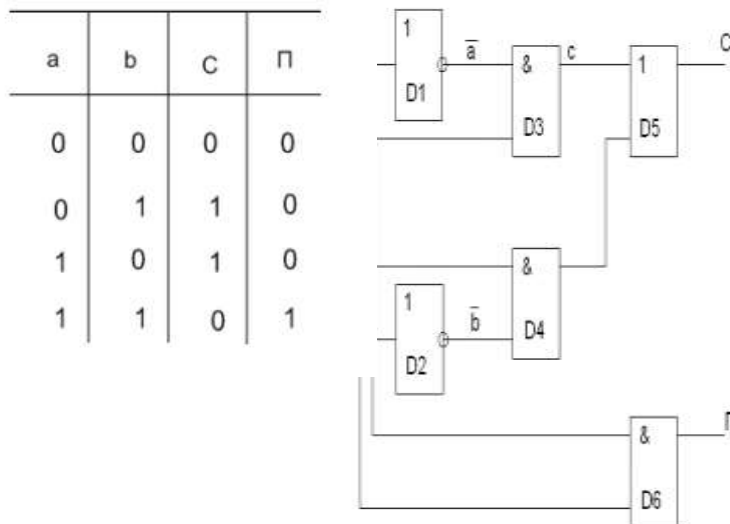


Рис. 3.4. Напівсуматор, заснований на простих логічних операціях

Відповідно до відповідної таблиці запис про аналітичну роботу в СДНФ має такий вигляд:

$$C = a\bar{b} \vee \bar{a}b = a \oplus b \quad (3.1)$$

$$П = ab \quad (3.2)$$

Рис. 3.4 ілюструє реалізацію цих шаблонів.

Наприклад, якщо ви хочете побудувати половинний суматор на інших елементах І-АБО-НІ, вихідні вирази мають бути відповідно перетворені:

$$C = \overline{\overline{ab}} \vee \overline{\overline{ab}} = (a \vee b) (\overline{a} \vee \overline{b}) = ab \vee \overline{a}\overline{b}, \quad (3.3)$$

$$П = \overline{ab} = \overline{a} \vee \overline{b}. \quad (3.4)$$

Функціональна схема такого напівсуматора наведена на рисунку 3.5.

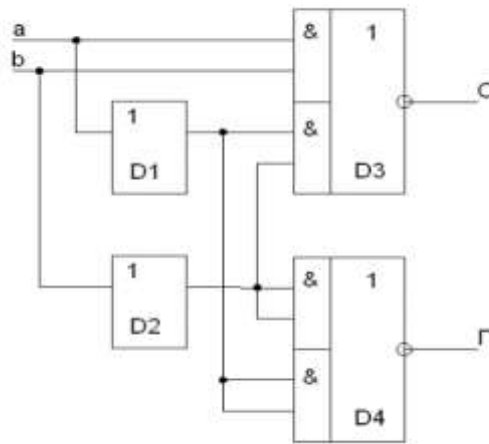


Рис. 3.5. Напівсуматор на основі елементів І-АБО

Оскільки сума C набуває значення 1, коли $a \neq b$, операція виду (3.5) також називається операцією нерівності або «яка виключає АБО». Пристрій, який виконує тільки цю операцію, має своє графічне позначення та оформлення. Тому напівсуматор можна представити як комбінацію однорозрядного вузла нерівності та генератора сигналу переносу (рис. 3.6).

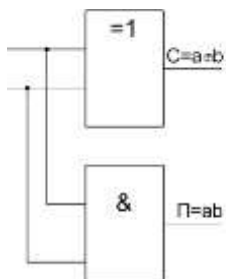


Рис. 3.6 Напівсуматор

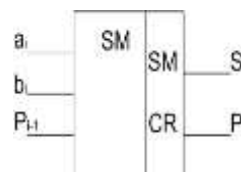


Рис. 3.7. Умовні графічні позначення однорозрядне додавання

Однорозрядне додавання. Напівсуматор підходить лише для використання в нижньому розряді багаторозрядного суматора. Для другого і наступних розрядів потрібні однорозрядні суматори з трьома входами (рис. 3.7): два для однойменних розрядів, що складаються з $a_i b_i$ чисел А, В і один для сигналу передачі P_{i-1} від попередньої цифри.

виходів S_i та P_i в СДНФ

$$S_i = \bar{a}_i \bar{b}_i P_{i-1} \vee \bar{a}_i b_i \bar{P}_{i-1} \vee a_i \bar{b}_i \bar{P}_{i-1} \vee a_i b_i P_{i-1}, \quad (3,5)$$

$$P_i = \bar{a}_i b_i P_{i-1} \vee a_i \bar{b}_i P_{i-1} \vee a_i b_i \bar{P}_{i-1} \vee a_i b_i P_{i-1}. \quad (3,6)$$

Для побудови суматора з цих виразів необхідно мати сім елементів І на три входи, два елементи АБО на чотири входи і три інвертори.

Його реалізація на двох напівсуматорах дещо економічніша. Структурну схему такого вузла можна отримати, виразивши формули (3.5) і (3.6) в (3.1) і (3.2). Для цього згрупуємо середній і крайній члени формули (3.5) і винесемо загальні коефіцієнти в дужки:

$$S_i = \bar{P}_{i-1}(\bar{a}_i \bar{b}_i \vee \bar{a}_i b_i) \vee P_{i-1}(a_i \bar{b}_i \vee a_i b_i).$$

Вираз у дужках є сумою C_i чисел a_i, b_i та її оберненим.

Дійсно,

$$\bar{C}_i = \overline{a_i b_i \vee a_i \bar{b}_i} = (\bar{a}_i \vee \bar{b}_i)(a_i \vee b_i) = \bar{a}_i b_i \vee a_i \bar{b}_i. \quad (3,7)$$

тому

$$S_i = \bar{C}_i \bar{P}_{i-1} \vee C_i P_{i-1} = C_i \oplus P_{i-1}. \quad (3,8)$$

Подібним чином вираз (3.6) перетвориться:

$$P_i = (a_i b_i \bar{P}_{i-1} \vee a_i b_i P_{i-1}) \vee (\bar{a}_i b_i P_{i-1} \vee a_i \bar{b}_i P_{i-1}) = a_i b_i \vee P_{i-1} C_i = \Pi_{1i} \vee \Pi_{2i}. \quad (3,9)$$

Схема однорозрядного суматора наведена на рисунку 3.8. У ньому перший напівсуматор додає два числа (a_i, b_i), що належать i -му розряду, і дає власну суму C_i і переносить P_i .

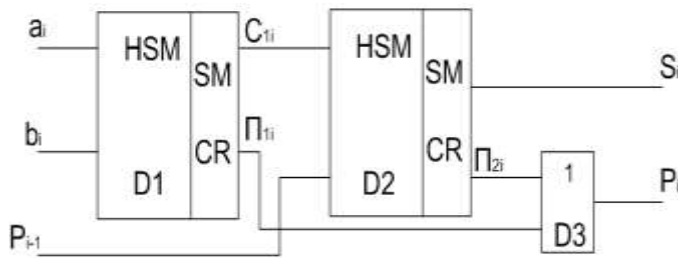


Рис. 3.8. Однорозрядний суматор

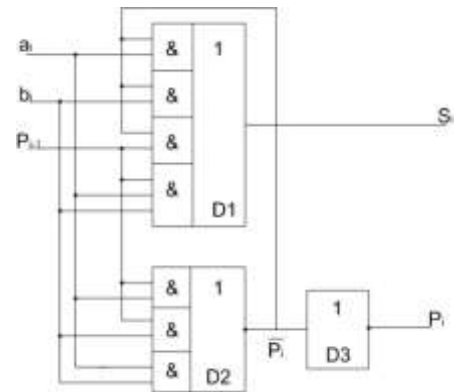


Рис. 3.9. Економний однорозрядний суматор на двох напівсуматорах

Другий напівсуматор переноситься з попереднього біта P_{i-1} із сумою C_{i-1} . На його виходах формується повна сума S_i і друга частина сумарної транзитної складової передачі P_{2i} , яка разом з P_{1i} на виході LE OR утворить сумарну передачу P_i .

Наступний варіант побудови однорозрядного суматора наведено на рис. 3.9. Воно одержується мінімізацією виразів (3.5) і (3.6) і їх зведенням до вигляду:

$$\begin{aligned} S_i &= a_i \bar{P}_i \vee b_i \bar{P}_i \vee P_{i-1} \bar{P}_i \vee a_i b_i P_{i-1} \\ P_i &= b_i P_{i-1} \vee a_i P_{i-1} \vee a_i b_i. \end{aligned} \quad (3.10)$$

Цей варіант більш економічний, тому що вимагає найменшої кількості вихідних каскадів, які визначають енергоспоживання всього пристрою. Тому повні суматори переважно створювати у вигляді окремих мікросхем.

3.2.2. Принцип побудови розрядного суматора

Найпростішим є суматор послідовного перенесення. Він складається з однорозрядних суматорів, кількість яких відповідає розрядам A і B з урахуванням знакового розряду (рис. 3.10). Суматори з'єднані послідовними ланцюжками переносу від молодших до старших розрядів. Крім того, молодша цифра

пов'язана з символічним ланцюгом передачі SM_{n+1} лише за допомогою кодів повернення.

Надбудова працює наступним чином. Усі змінні, що складаються з А і В, одночасно застосовуються до відповідних входів. Кожна цифра створює власну суму та сигнали суми. Останні за рахунок часу затримки t_n досягають входів сусідніх однорозрядних суматорів і викликають транзитні передачі. Відбувається остаточна корекція результату додавання. Максимальний час підсумовування досягається, коли перенос передається по всіх бітах. Завжди числа А і В повинні залишатися незмінними у вхідних даних, а сума не повинна зчитуватися. Отже, час затримки встановлення суматора $T_{zd\ sm} = n \cdot t_n + t_s$.

Як позначаються числа А і В у додатковому та інверсному коді:

$$T_{zd\ sm} = (n+1)t_n + t_s$$

де: n – швидкість передачі програми, $t_n + t_s$ – час затримки формування сигналу передачі та сигналу суми в одному біті.

Недоліком тренажерів з послідовним передаванням є низька швидкодія. Для збільшення швидкості роботи необхідно прискорити процес формування та поширення переказу.

Найбільшу швидкість мають суматори паралельного перенесення. У них сигнал переносу для кожного розряду виробляється з урахуванням аналізу всіх попередніх молодших розрядів, що складаються з n .

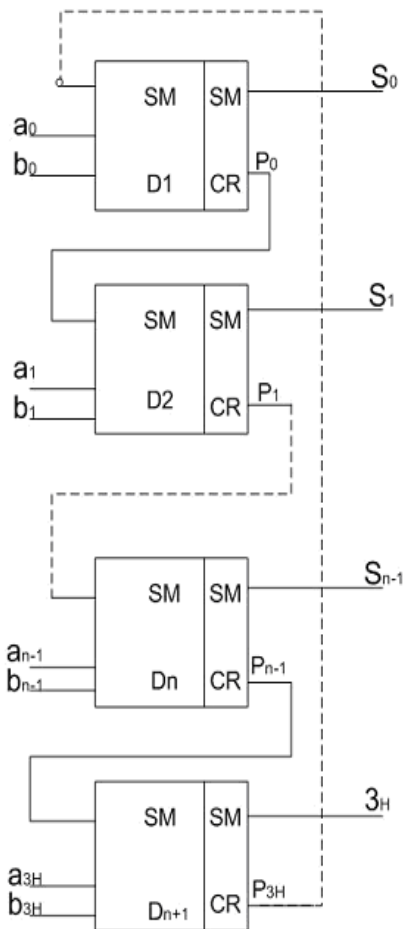


Рис. 3.10 Багаторозрядний суматор із послідовним переносом

Для трирозрядного суматора, побудованого на основі напівсуматорів, використовуючи вираз (3.9), запишемо переноси для кожного біта

$$\begin{aligned}
 P_0 &= \Pi_0, \\
 P_1 &= \Pi_1 \vee P_0 C_1 \\
 P_2 &= \Pi_2 \vee P_1 C_2
 \end{aligned}$$

Виразивши їх через відповідні перенесення та відповідні суми C_i , отримаємо:

$$\begin{aligned}
 P_0 &= \Pi_0, \\
 P_1 &= \Pi_1 \vee \Pi_0 C_1, \\
 P_2 &= \Pi_2 \vee \Pi_1 C_2 \vee \Pi_0 C_1 C_2
 \end{aligned} \quad (3.11)$$

Можна побачити, що для створення сигналів P AND потрібні спеціальні вузли передачі (CRU), які можуть бути реалізовані за допомогою I-OR LE (рис. 3.11). Час затримки створення переказів визначається виключно часом затримки перемикування цих елементів і не залежить від кількості цифр.

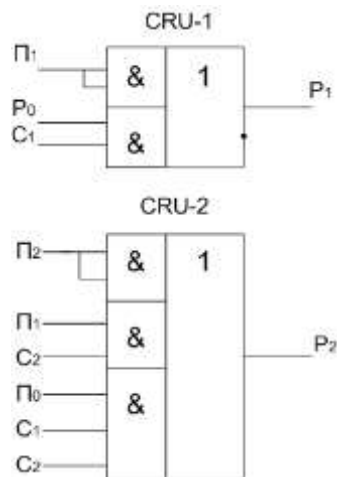


Рис. 3.11 Схема блоку передачі (CRU) на I-OR типу LE

Схема суматора з паралельним перенесенням показана на рисунку 3.12.

Час очікування монтажу:

$$T_{zd\ sm} = 2t_{zd\ HSM} + t_{zd\ вгору},$$

де $t_{zd\ HSM}$ і $t_{zd\ УП}$ – час затримки формування вихідних сигналів модуля напівсуматора та модуля передачі.

Паралельна передача дозволяє істотно збільшити швидкість суматора, але призводить до збільшення апаратного обсягу за рахунок впровадження UE, які стають складнішими зі збільшенням розрядності. Тому на практиці для отримання високої швидкості при помірній кількості обладнання використовується комбінована передача. Її суть полягає в тому, що однорозрядні суматори об'єднуються в групи. Одночасні переїзди організуються всередині групи, а послідовні – між групами. Іншим поширеним методом організації прискореної передачі в багаторозрядних суматорах є метод, при якому створюється послідовна передача в групах і одночасна передача між групами. При цьому, порівняно з попереднім способом, спостерігається збільшення швидкості, якщо кількість груп перевищує кількість цифр у групах.

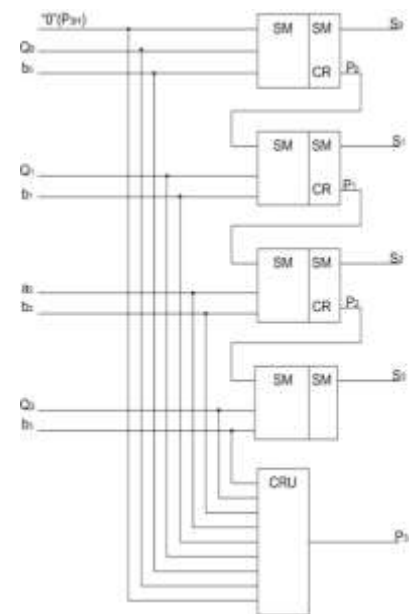
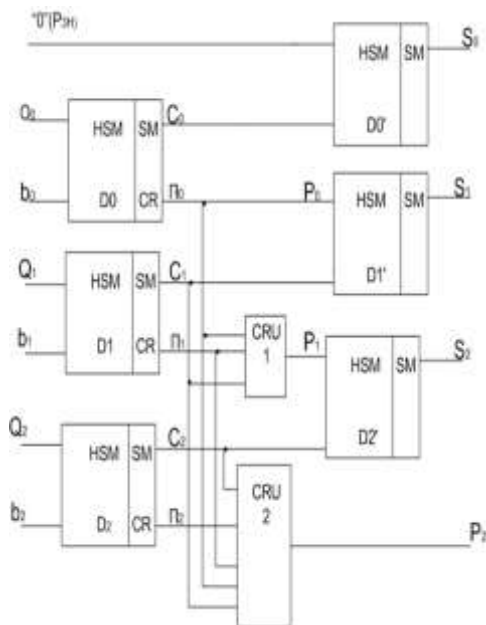


Рис. 3.12. Трирозрядний суматор з паралельним перенесенням
 рис. 3.13. Логічна структура суматора 564ИМ1

Прикладом складу однієї групи розрядів, в якому передача сигналу останнього розряду виробляється за допомогою спеціального вузла, може служити мікросхема 564 ИМ1, структура якої наведена на рис. 3.13.

У таких пристроях однорозрядні суматори зазвичай реалізуються за схемою на рис. 3.9, а вузол передачі поряд з елементами І-АБО включає елементи нерівності для отримання відповідних сум S_i .

3.3. Цифрові компаратори

Цифрові компаратори належать до арифметичних пристроїв комбінаційного типу. Вони призначені для порівняння двох n -розрядних двійкових чисел. Залежно від конструкції компаратора результати порівняння можуть бути такими: $A = B$, $A > B$ і $A < B$.

Інтегральні компаратори функціонально завершені, тобто виконують усі три операції порівняння і мають три виходи. На кожному з них отримують результат однієї конкретної операції у вигляді певного логічного рівня. Приклад умовного графічного зображення чотирирозрядного цифрового компаратора наведено на рисунку 3.14. Вихідні сигнали L , F і N відображаються як рівень

логічної одиниці, коли $A=B$, $A>B$ і $A<B$ відповідно. Операція рівності F є запереченням операції нерівності. Операції порівняння є побітовими. Умови роботи однорозрядного функціонально повного компаратора наведені в таблиці 3.2.

Таблиця 3.2

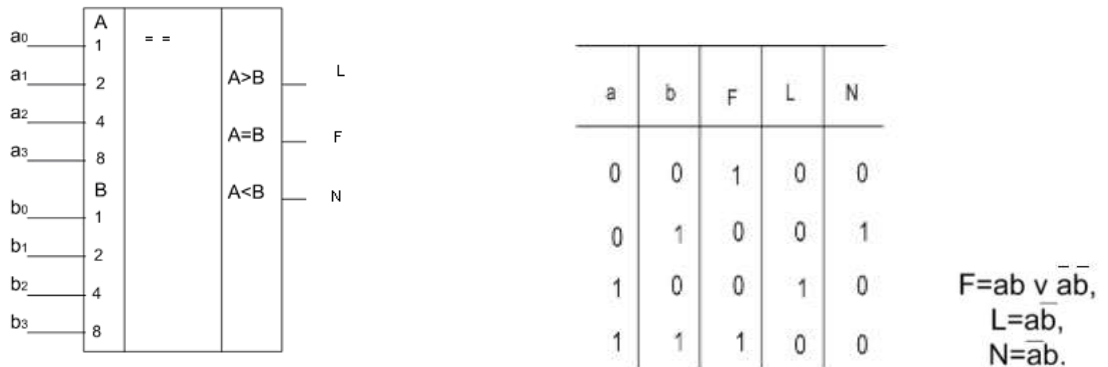


Рис. 3.14. Умовні графічні позначення чотирирозрядний цифровий компаратор

Для реалізації операцій F на виході можна використовувати вузол нерівності, а для операцій L і N - два з'єднувачі $D5$, $D7$ (рис. 3.15).

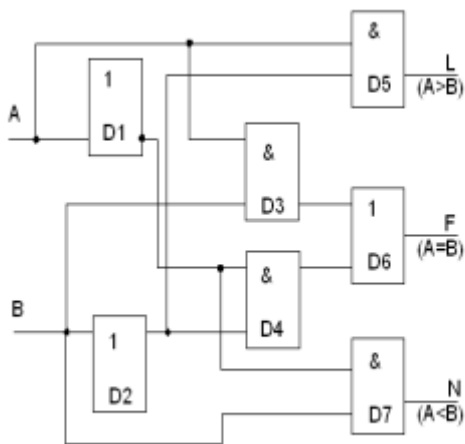


Рис. 3.15 Схема повного компаратора

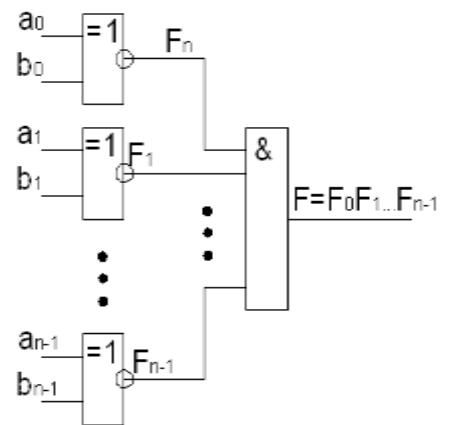


Рис. 3.16. Структурна схема багаторозрядної схеми компаратор еквівалентності

У багаторозрядному компараторі рівність чисел A і B встановлюється у разі рівності пар усіх розрядів з однаковою назвою цих чисел (a_i , b_i). Звідси

впливає, що структура багаторозрядного компаратора еквівалентності повинна бути побудована з відносно однакових однорозрядних вузлів, виходи яких з'єднані логічним елементом І (рис. 3.16).

У компараторах парності бітові вузли залежать один від одного. Під час порівняння бітів старші біти мають пріоритет над нижчими. Наприклад, у трирозрядному компараторі нерівність $A > B$ забезпечується в таких випадках: коли $a_2 > b_2$ або $a_2 = b_2$ і $a_1 > b_1$ або $a_2 = b_2$ і $a_1 = b_1$ і $a_0 > b_0$.

У DDNF це виглядає так:

$$L = a_2 \bar{b}_2 \vee (a_2 b_2 \vee \bar{a}_2 \bar{b}_2) a_1 \bar{b}_1 \vee (a_2 b_2 \vee \bar{a}_2 \bar{b}_2) (a_1 b_1 \vee \bar{a}_1 \bar{b}_1) a_0 \bar{b}_0 = a_2 \bar{b}_2 \vee F_2 a_1 \bar{b}_1 \vee F_2 F_1 a_0 \bar{b}_0.$$

Синтезований за цим виразом компаратор наведено на рис. 3.17. На малюнку далі показано можливість отримання $A = B$ за допомогою молодшого вузла еквівалентності D6 і з'єднувача D11. Зрозуміло, для реалізації двох нерівностей ($A > B$ і $A < B$) компаратор необхідно доповнити трьома інверторами для змінних a_i , трьома з'єднувачами і з'єднувачем, аналогічним D7...D10. Типовим прикладом є чотирирозрядний компаратор К555СП1 (рис. 3.18). Він знаходиться на рівні з чотирма парами входів, які приймають два чотирибітних числа A і B для аналізу, він має три входи $E_X(A > B)$, $E_X(A = B)$, $E_X(A < B)$, призначений для компонентів компаратора з розрядним розширенням.

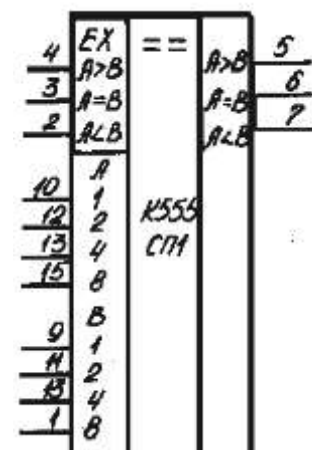
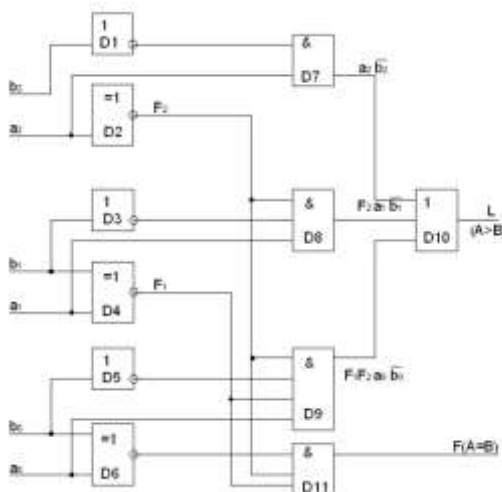


Рис. 3.17. Трирозрядний компаратор

Рис. 3.18.

Відображення умовної графіки
операції $A = B$ і $A > B$

компаратора К555СП1

Додаткові входи перевіряють стан виходів L, F і N лише тоді, коли $A = B$ згідно з таблицею 3.3. Якщо $A \neq B$, то $F=0$ і $L \neq N$ незалежно від змінних на входах EX.

Таблиця 3.3

Входи порівняння EX			Виходи		
$A > B$	$A = B$	$A < B$	$A > B$	$A = B$	$A < B$
1	0	0	1	0	0
0	0	1	0	0	1
×	1	×	0	1	0
1	0	1	0	0	0
0	0	0	1	0	1

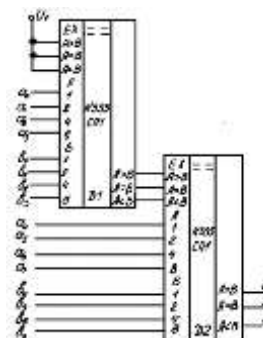


Рис. 3.19. Послідовне з'єднання компараторів К555СП1

Для збільшення пропускної здатності компаратори можна з'єднувати послідовно і пірамідально. При послідовному розширенні (рис. 3.19) виходи $A > B$, $A = B$, $A < B$ мікросхеми, що аналізують молодші розряди чисел A і B , необхідно підключити до входів розширення з однаковими назва EX мікросхеми, яка аналізує наступні розряди. Для входу розширення $EX(A = B)$ першої ІС необхідно використовувати логічну одиницю. Таким чином $4n$ -розрядні числа можна порівнювати за допомогою n мікросхем. Основним недоліком послідовного методу збільшення пропускної здатності є зниження продуктивності.

На рис. 3.20 показаний приклад структури компаратора з пірамідальним методом збільшення пропускної здатності. Компаратор змонтований на шести мікросхемах К555СП1. П'ять із них утворюють перший етап і порівнюють числа A і B до 24 бітів включно. Їх вихідні дані $A > B$ і $A < B$ створюють п'ять пар змінних, які можна представити у вигляді двох п'ятирозрядних чисел x і y , придатних для порівняння другого ступеня. На першому етапі молодша цифра ІС служить чотирирозрядним компаратором, а решта чотири - п'ятибітним компаратором (входи $EX(A > B)$ і $EX(A < B)$ служать п'ятим для забезпечення цього режиму, на входах $EX(A = B)$ подається напруга логічного нуля (табл. 3.3). У пірамідальному методі побудови перехідні процеси виникають раніше,

ніж у послідовному випадку, і в багатьох випадках потрібна велика кількість обладнання.

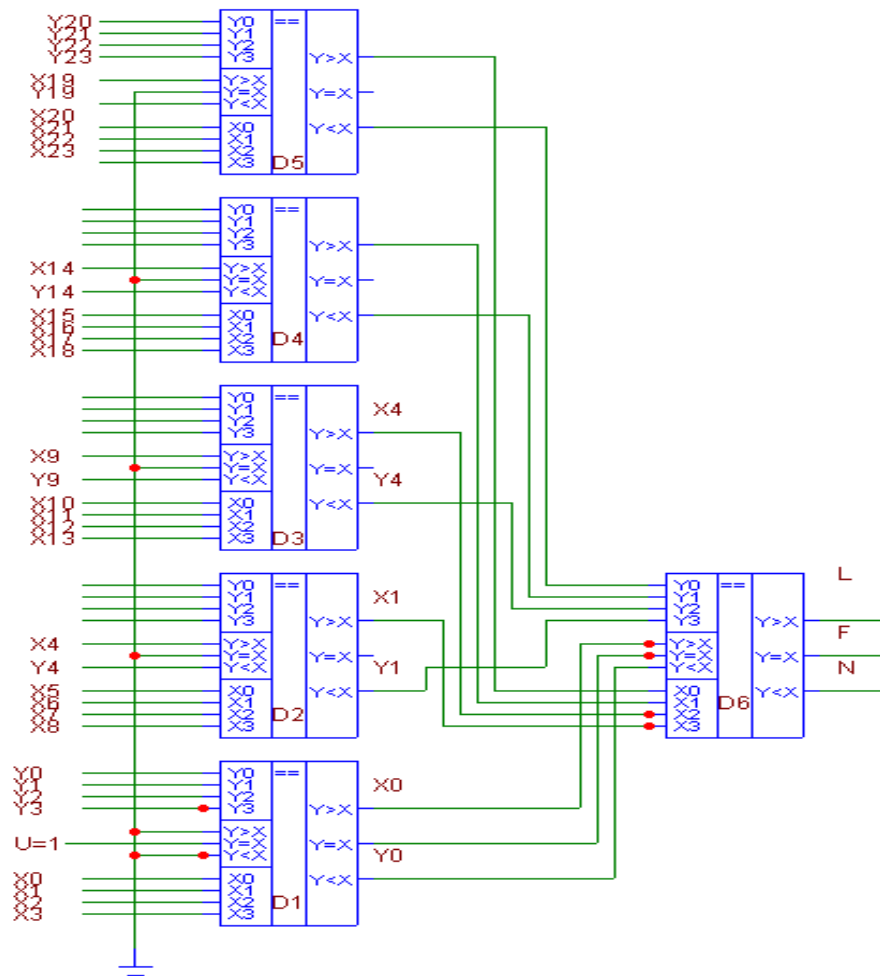


Рис. 3.20. Пірамідальне з'єднання компараторів

3.4. Дешифратори

Для перетворення інформації та форми її представлення в цифрових системах використовуються декодери та кодери. Декодери використовуються для перетворення двійкового коду в інший код. Вони мають n входів і $M \leq 2^n$ виходів. Якщо кількість виходів $M=2^n$, дешифратор називається повним, якщо менше – неповним.

У повних дешифраторах кожна комбінація значень вхідних змінних залежить від напруги логічної одиниці (нуля) на одному з її виходів. Неповні декодери мають перебільшені комбінації, до яких виходи «нечутливі» або їхній стан є невизначеним.

Неповні коди включають, наприклад, декодер двійково-десятькового коду в десятковий. Має чотири входи для двійкового коду і десять виходів, на яких встановлюється логічна одиниця (нуль) тільки для десяти значень вхідного двійкового числа: 0000...1001. Для інших традиційних комбінацій вхідних змінних стан декодера не визначено. Робота дешифратора наведена в таблиці 3.4 і описується такими виразами.

Таблиця 3.4 Таблиця істинності трирозрядного декодера

Вихід	Вхід	ДДНФ
$X_1 X_2$ X_3	$T_0 T_1 T_2 T_3 T_4 T_5 T_6 T_7$	
0 0 0	1 0 0 0 0 0 0 0	$Y_0 = \overline{X_1} \overline{X_2} \overline{X_3}$
0 0 1	0 1 0 0 0 0 0 0	$Y_1 = \overline{X_1} \overline{X_2} X_3$
0 1 0	0 0 1 0 0 0 0 0	.
0 1 1	0 0 0 1 0 0 0 0	.
1 0 0	0 0 0 0 1 0 0 0	.
1 0 1	0 0 0 0 0 1 0 0	.
1 1 0	0 0 0 0 0 0 1 0	$Y_7 = X_1 X_2 X_3$
1 1 1	0 0 0 0 0 0 0 1	

З таблиці та виразів видно, що логічна одиниця встановлюється на виході Y_0 , коли вхідний код дорівнює 000, коли Y_1 дорівнює 001, коли Y_7 дорівнює 111 тощо. Залежно від того, як вихідні функції Y_i реалізовані, розрізняють одноступінчасті декодери (лінійні) і багатоступінчасті. Останні, в свою чергу, діляться на пірамідальні і прямокутні.

При синтезі лінійних дешифраторів кожен логічний вираз y_i реалізується незалежно (рис. 3.21). Лінійні декодери прості, економічні та високошвидкісні. При високій швидкості вхідного коду, що перевищує максимальну кількість входів LE, схема декодера стає значно складнішою. У цьому випадку перевагу надають багатоступеневим декодерам. Приклад умовного графічного представлення повного дешифратора для трьох вхідних змінних наведено на рис. 3.22.

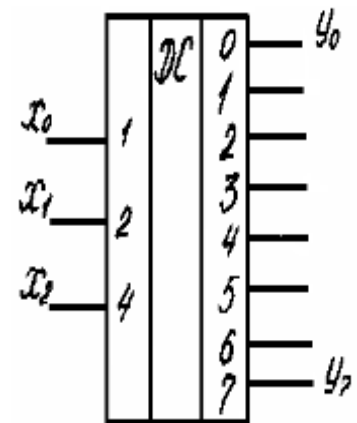
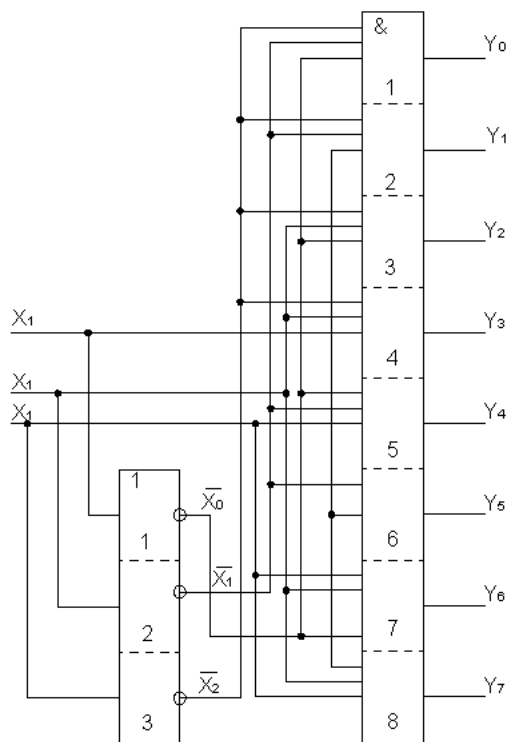


Рис. 3.21. Лінійний 3-дешифратор Рис. 3.22. Умовне графічне зображення

У пірамідних декодерах об'єднання змінних X виконується не одночасно, а послідовно в кілька кроків.

Наприклад, необхідно синтезувати декодер на основі двовходових роз'ємів для вхідного коду з розрядністю $n=4$.

Для цього вихідні функції поділяються наступним чином:

$$\begin{aligned}
 Y_0 &= [(X_3 \overline{X_2}) \overline{X_1}] \overline{X_0} & Y_{10} &= [(\overline{X_3} X_2) \overline{X_1}] X_0 \\
 Y_1 &= [(\overline{X_3} \overline{X_2}) \overline{X_1}] X_0 & & \dots\dots\dots \\
 Y_2 &= [(\overline{X_3} X_2) \overline{X_1}] X_0 & & \\
 & \dots\dots\dots & Y_{15} &= [(X_3 X_2) X_1] X_0
 \end{aligned}$$

x_3 і x_2 за допомогою першого кроку . Отримані результати використовуються на другому кроці для отримання кон'юнкції трьох змінних - x_3 , x_2 , x_1 . На третьому етапі враховується змінна x_0 . Тому схема пірамідного дешифратора має вигляд, показаний на рис. 3.23. Головною перевагою такого декодера є можливість використання LE (Nle), а отже, і споживана енергія. Тому при використанні двохходових LE $Nle = 2^2 + 2^3 + \dots + 2^n = 2^{n+1} - 4$. Це означає, що при великих n кількість Nle майже в 2 рази більша за кількість лінійних декодерів. Декодері Pyramid також втрачають швидкість. Тому вони не знайшли широкого застосування.

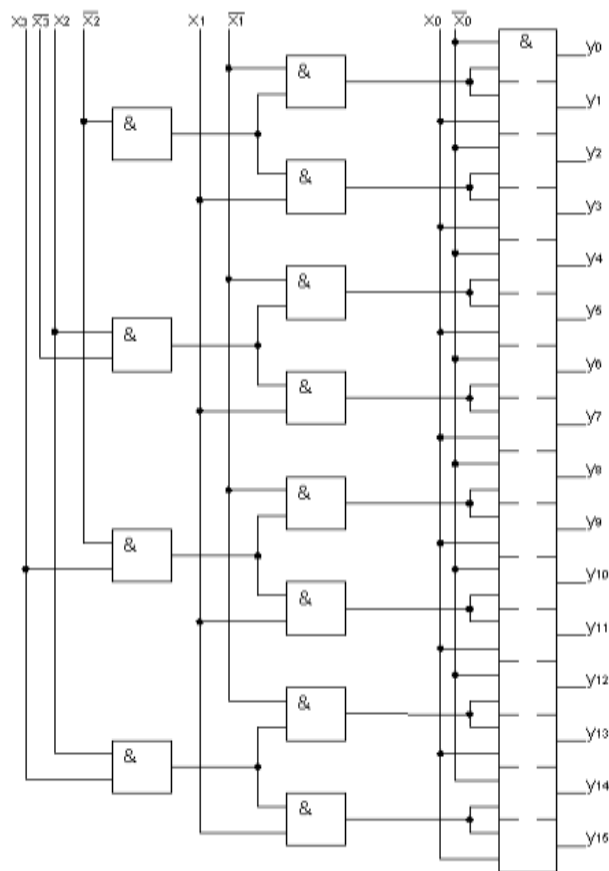


Рис. 3.23. Чотирирозрядний пірамідний декодер

Для вищої швидкості вхідного коду краще використовувати прямокутні декодери. Зазвичай вони двоступеневі. Тому для їх синтезу змінні логічні функції y_i розбивають на дві, по можливості, однакові групи. Кожна група керується своїм лінійним (рідше пірамідальним) декодером першого каскаду, а потім вихідні сигнали цих декодерів попарно об'єднуються за допомогою т.зв. другий каскад матричного декодера. Приклад прямокутного декодера для чотирирозрядного вхідного коду наведено на рис. 3.24.

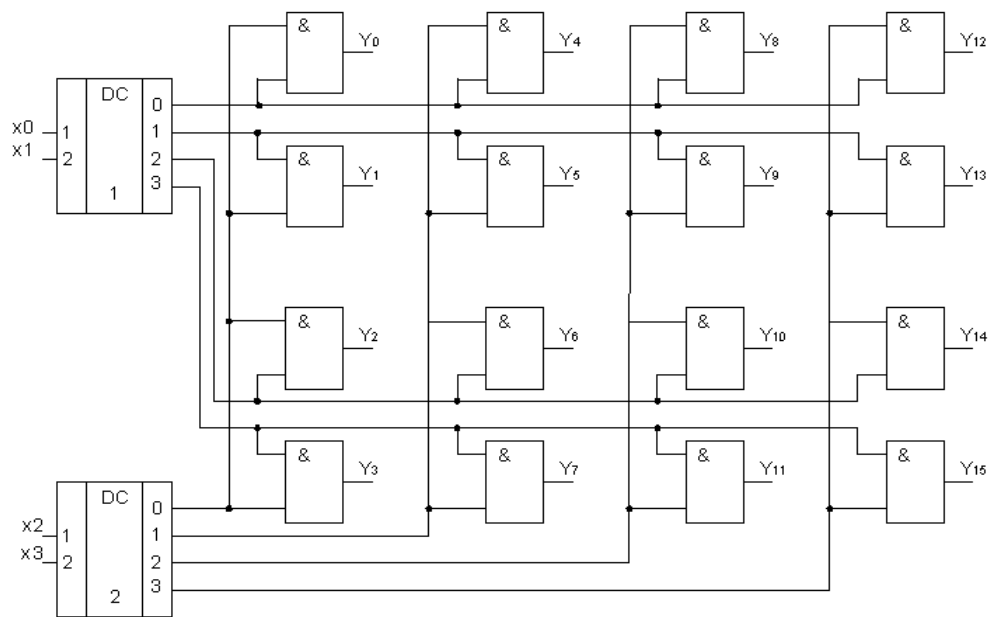


Рис. 3.24. Прямокутний чотирирозрядний двоступеневий декодер

Його збільшену функціональну схему наведено на рис. 3.25. Тут перший ступінь складається з двох лінійних дешифраторів на два входи, а другий ступінь декодера містить набір (матрицю) роз'ємів 4x4. З точки зору структурної глибини, а отже, і швидкості, прямокутні декодери знаходяться між лінійними та пірамідальними декодерами.

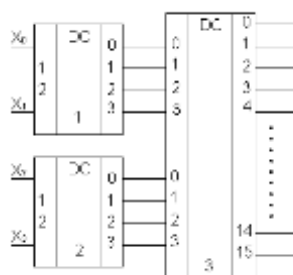


Рис. 3.25. Двоступеневий прямокутний декодер

За апаратним обсягом вони найбільш економічні при $n > 8$. Тому вони використовуються в BIS і SBIS, де необхідно забезпечити доступ до великої кількості (сотні і більше) одержувачів, наприклад, до елементів пам'яті в масі запам'ятовуючі пристрої.

У мікросхемах з низьким і середнім ступенем інтеграції дешифрування виконується лінійним способом. При необхідності збільшення кількості інформаційних входів мікросхеми декодера підключаються особливим чином.

На рис. На рисунку 3.26 зображено схему п'ятирозрядного дешифратора на основі двох чотирирозрядних дешифраторів 133IDZ. У них на збудженому виході встановлюється низький рівень напруги (логічний нуль), а на решті виходів - високий, тобто виходи інвертуються. Ці чотири мікросхеми введення інформації мають два входи зворотного розширення, з'єднані операцією І.

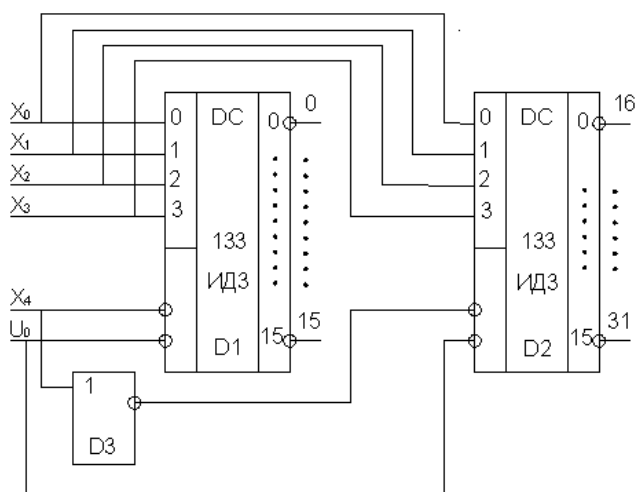


Рис. 3.26. П'ятирозрядний дешифратор на основі двох мікросхем 133ID1

При наявності хоча б на одному вході Е логічної одиниці на всіх виходах мікросхеми, незалежно від значень змінних на інформаційних входах, блок встановлюється, тобто інтегральна схема вимикається. Таким чином, завдяки інвертору, включеному в схему розширення другої ІМС, комбінації вхідних змінних від 0000 до 01111 включно підтримуються першою ІМС і комбінації

10000...11111 - другою ІМС. Також використовується пірамідний метод збільшення швидкості передачі вхідного коду.

3.5. Шифратори

Програми шифрування призначені для виконання операцій, протилежних до операцій декодера. Вони перетворюють різні коди, цифри яких є двійковими змінними, у двійковий код. Таблиця 3.5 ілюструє логіку роботи п'ятирічного шифру. Вихідні функції мають вигляд і реалізуються кодувальником, як на рис. 3.26. Ця схема є пріоритетною. У ньому вихідний код визначається старшим розрядом вхідного коду і не залежить від значень змінних з молодшими розрядами. Шифратори в інтегральному виконанні, крім інформаційних виходів, часто мають допоміжні виходи, що дозволяють розширити функціональні можливості мікросхем.

Наприклад, кодер КМ555ІV1 (рис. 3.27) має 8 входів для десяткового коду, 3 виходи для двійкового коду, а також вхід Е для роботи кодера, виходи G0 і GS, які сигналізують про відносний нуль і ненульова комбінація вхідних змінних при наявності сигналу дозволу АРЕ.

Поєднуючи додаткові виходи двох або більше кодерів певним чином, можна збільшити швидкість передачі перетворених кодів.

Таблиця 3.5 П'ятирічна таблиця істинності шифру

$X_4 X_3 X_2 X_1$ X_0	$T_2 T_1 T_0$	
1 xxxx	1 0 1	$Y_2 = X_4 \vee \overline{X_4} X_3$
0 1 xxx	1 0 0	$Y_1 = \overline{X_4} \overline{X_3} X_2 \vee \overline{X_4} X_3 \overline{X_2} X_1$
0 0 1 xx	0 1 1	$Y_0 = X_4 \vee \overline{X_4} \overline{X_3} X_2 \vee \overline{X_4} X_3 \overline{X_2} \overline{X_1} X_0$
0 0 0 1 x	0 1 0	
0 0 0 0 1	0 0 1	
0 0 0 0 0	0 0 0	

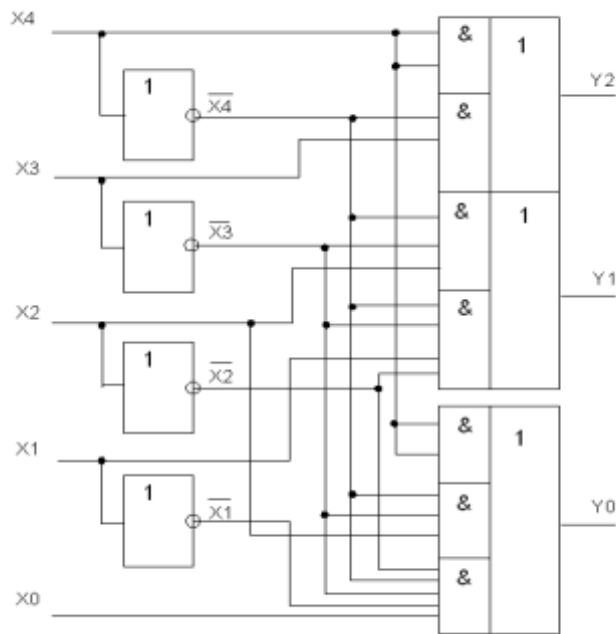


Рис. 3.26. Пріоритетний кодер на п'ять входів

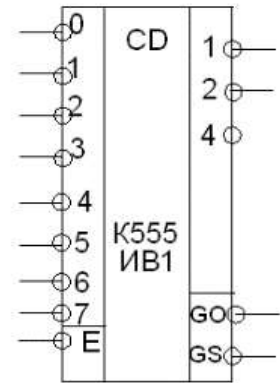


Рис. 3.27. Умовні графічні позначення шифратора КМ555ІВ1

3.6. Перетворювач двійково-десятькового коду індикатора в семисегментний

Числа на табло та пультах зазвичай відображаються в десятковому коді. Для цього можна використовувати дешифратор на мікросхемі К555ІВ1 разом з газорозрядним індикатором. Однак використання таких індикаторів на практиці є небажаним через необхідність використання джерела живлення високої напруги (≈ 200 В). В даний час набули поширення так звані семисегментні світлодіодні індикатори і рідкокристалічні індикатори, які працюють на тій же напрузі, що і мікросхеми. У них індикація здійснюється за допомогою семи елементів (рис. 3.28). Подаючи керуючу напругу на окремі елементи індикатора і викликаючи його світіння (світлодіодні індикатори) або змінюючи його колір (рідкокристалічні індикатори), можна отримати зображення десяткових цифр 0, 1, ..., 9. Деякі мікросхеми - перетворювачі коду 8421 в семисегментні показані на рис. 3.29.

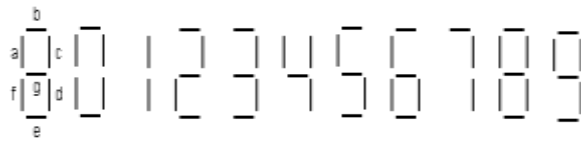


Рис. 3.28. Зображення чисел у семи відрізках

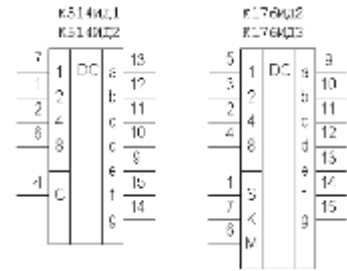


Рис. 3.29. Перетворювачі двійкового і десяткового коду в сім показників сегментовані

Вхідні сигнали рівня TTL надходять на мікросхеми серії K514. Сигнал C використовується для вимкнення сигналізації (низький рівень напруги). Під час нормальної роботи рівень сигналу становить $C=1$. Дешифратор в системі K514ID1 працює зі світлодіодними індикаторами, які мають окремі аноди, а в K514ID2 - з окремими катодами. Дешифратор K514ID2 підключається до індикаторів через струмообмежувальні резистори (200 - 500 Ом) і перший має такі резистори в корпусі.

Мікросхеми K176ИД2 і K176ИД3 є перетворювачами кодів з регістром введення пам'яті. Інформація записується в пам'ять по фронту тактового сигналу, що надходить на вхід S (одночасно сигнал на вхід $K=0$). Якщо сигнал $K=1$, декодер заблоковано. Вихідний код цих декодерів є прямим $M=0$ і зворотним $M=1$. Декодери призначені для роботи з рідкокристалічними і люмінесцентними індикаторами. Також можуть працювати зі світлодіодними показниками повороту при напрузі джерела живлення 9-12 В зі зниженою яскравістю (за рахунок обмеження струму 2-3 мА).

3.7 Мультиплексори та демультиплексори

3.7.1. Загальна характеристика мультиплексорів

Мультиплексор — це процесор, що використовується для комутації інформаційного потоку з одного з n входів на спільний вихід. Номер конкретного вхідного рядка, підключеного до виходу в кожному циклі, визначається кодом адреси A_0, A_1, \dots, A_{m-1} . Співвідношення між кількістю інформаційних входів і адресними входами m визначає співвідношення $n=2^m$. Умовне графічне зображення мультиплексорів продемонстровано на рис. 3.30. Отже, мультиплексор реалізує управляему передачу сигналів від кількох входів до однієї вихідної.

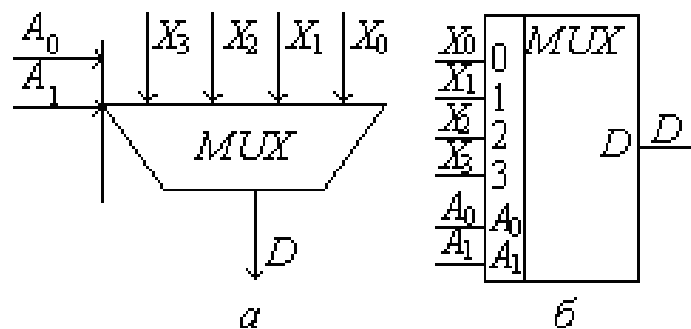


Рис. 3.30 Умовне графічне позначення мультиплексора

Функціонал мультиплексування записується літерами MUX (мультиплексор). Мультиплексорні системи використовуються для таких операцій:

- перемикання як окремих ліній, так і груп ліній (автобусів);
- реалізація логічних функцій;
- побудова діаграм порівняння та генераторів коду.

Логіка чотирьохвходового мультиплексора представлена в таблиці 3.6, де A_0, A_1 – код адреси; F_0, F_1, F_2, F_3 – виходи внутрішнього декодера; X_0, X_1, X_2, X_3 – вхідна інформація; D - вихід загальної інформації.

Таблиця 3.6

1	0	F ₀	F1	F2	F3	Д
0	0	1	0	0	0	F ₀ X ₀
0	1	0	1	0	0	F1X1
1	0	0	0	1	0	F2X2
1	1	0	0	0	1	F3X3

За допомогою табл. 3.6 вихідну функцію D можна представити за допомогою виходів F₀–F₃ внутрішнього дешифратора у вигляді:

$$D = F_0X_0 \vee F_1X_1 \vee F_2X_2 \vee F_3X_3,$$

або з кодом адреси minterms:

$$D = \overline{A_1}\overline{A_0}X_0 \vee \overline{A_1}A_0X_1 \vee A_1\overline{A_0}X_2 \vee A_1A_0X_3$$

Діаграми мультиплексора, що відповідають рівнянням (1) і (2), показані на рис. 3.31.

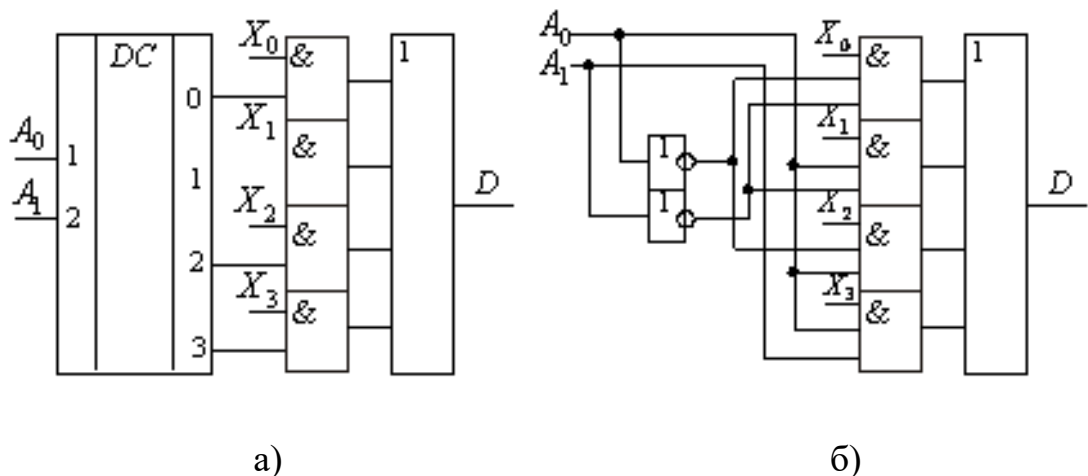


Рис. 3.31. Схеми мультиплексора

При побудові схеми мультиплексора з внутрішнім лінійним дешифратором потрібні логічні елементи з меншою кількістю входів, але при цьому збі-

льшується час встановлення сигналу на виході. Коли декодер вимкнено, продуктивність збільшується, але необхідні схеми узгодження з більшою кількістю вхідних виводів.

У інтегральному виді мультиплексори випускаються на 4-ри, вісім або шістнадцять входів. Каскадування дозволяє комутувати будь-яку кількість вхідних ліній на основі мікросхем послідовного мультиплексора з меншою розрядністю. Приклад побудови мультиплексорної системи з 16 входами на основі типових мультиплексорів із чотирма входами наведено на рис. 3.32. Молодші розряди адреси A_1, A_0 підключені до адресних входів усіх мультиплексорів першого рівня, виходи яких виконують такі функції:

$$D'_0 = F_0 X_0 \vee F_1 X_1 \vee F_2 X_2 \vee F_3 X_3;$$

$$D'_1 = F_0 X_4 \vee F_1 X_5 \vee F_2 X_6 \vee F_3 X_7;$$

$$D'_2 = F_0 X_8 \vee F_1 X_9 \vee F_2 X_{10} \vee F_3 X_{11};$$

$$D'_3 = F_0 X_{12} \vee F_1 X_{13} \vee F_2 X_{14} \vee F_3 X_{15};$$

де $F_0 - F_3$ - виводи дешифраторів:

$$F_0 = \overline{A_1} \overline{A_0}; F_1 = \overline{A_1} A_0; F_2 = A_1 \overline{A_0}; F_3 = A_1 A_0$$

$X_{15} - X_0$ - вхідні перемінні.

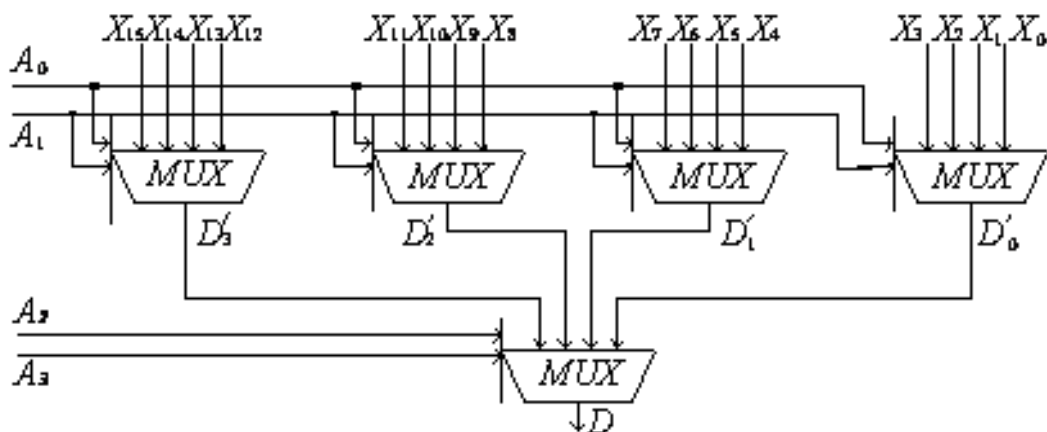


Рис. 3.32. Каскадні мультиплексори

Найстарші розряди адрес A_3, A_2 надходять на адресовані входи мультиплексорів іншого рівня, на виході котрого кінцеве значення створюється

такими мінтермами:

$$F_0 = \overline{A_3} \overline{A_2}; F_1 = \overline{A_3} A_2; F_2 = A_3 \overline{A_2}; F_3 = A_3 A_2$$

Нехай це буде, наприклад, функція

$$D = F_0' D_0' \vee F_1' D_1' \vee F_2' D_2' \vee F_3' D_3'$$

де вихідні дані внутрішнього декодера визначаються значенням адреси $A_3 A_2 A_1 A_0 = 1011$.

перемінна, значення котрої визначається з порівнянь першого рівня, перемикається на вхід другого рівня D_2^1 (оскільки $=1$): F_2'

$$D_2^1 = F_3 X_{11} = A_1 A_0 X_{11} = 1 X_{11} = X_{11}.$$

3.7.2. Логічні функції мультиплексорів

Мультиплексори використовуються для реалізації логічних функцій з числом змінних m , рівним числу розрядів коду адреси. Виконувана функція повинна представлятися в ДНФ. А на адресні входи подаються змінні, а входами конфігурації служать інформаційні виводи - на них подається постійний нуль і одиниця в залежності від виконуваної функції.

Функція трирозрядного мультиплексора з вісьмома входами описується рівнянням:

$$D(A) = \overline{A_2} \overline{A_1} \overline{A_0} X_0 \vee \overline{A_2} \overline{A_1} A_0 X_1 \vee \overline{A_2} A_1 \overline{A_0} X_2 \vee \overline{A_2} A_1 A_0 X_3 \vee A_2 \overline{A_1} \overline{A_0} X_4 \vee A_2 \overline{A_1} A_0 X_5 \vee A_2 A_1 \overline{A_0} X_6 \vee A_2 A_1 A_0 X_7$$

Якщо потрібно взяти логічну функцію з десятичними еквівалентами мінтермів 1, 3, 5 і 7, то до парних входів X_0, X_2, X_4 і X_6 слід застосувати константу "0", а до непарних входів X_1, X_3, X_5 і X_7 – константа «1».

У результаті отримуємо (рис. 3.33):

$$D(A) = \overline{A_2} \overline{A_1} A_0 \vee \overline{A_2} A_1 A_0 \vee A_2 \overline{A_1} A_0 \vee A_2 A_1 A_0$$

При допомозі додаткових логічних перетворень можливо реалізувати логічні функції з декількома змінними $m+1$, що на 1 біт більше адресного вхідного коду мультиплектора.

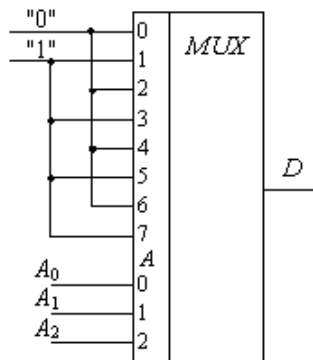


Рис. 3.33

Також можливі другі застосування схем мультиплектора. Мультиплексор функціонує як перетворювач n -розрядного паралельного двійкового коду в послідовний код. Для реалізації потрібно дати паралельний код на інформаційні виводи мультиплектора і потім змінити код на адреси в необхідному порядку.

3.7.3 . Мультиплексування шин

Мультипликація шин — це перемикання ліній (груп ліній) від декількох джерел інформації до одного приймача. Такі мікрооперації реалізуються схемами на основі однолінійних мультиплексорів. При виборі кількості та типу мультиплексорів слід враховувати: кількість комутованих шин $2m$, де m довжина коду адреси;

I -е вхідне число всіх мультиплексорів використовується для об'єднання бітів певної шини. Схема чотиришинного мультиплектора $X(n)$, $Y(n)$, $Z(n)$ і $S(n)$ наведена на рис. 3.34. Його конструкція потребує n двоадресних мультиплексорів із чотирма входами, де n — будь-яка швидкість комутованих шин.

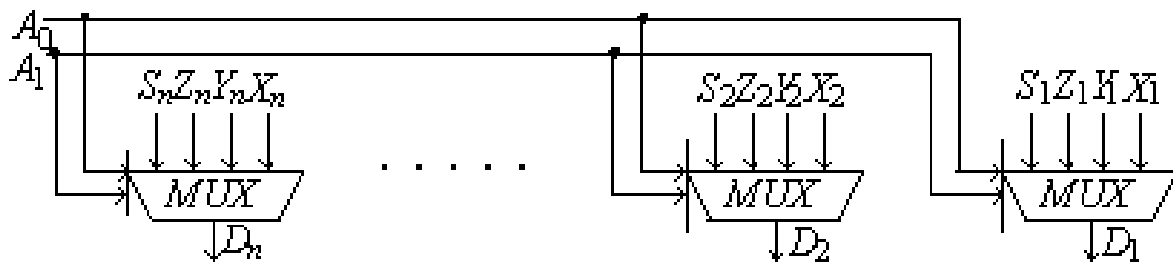


Рис. 3.34. Мультиплексор шини

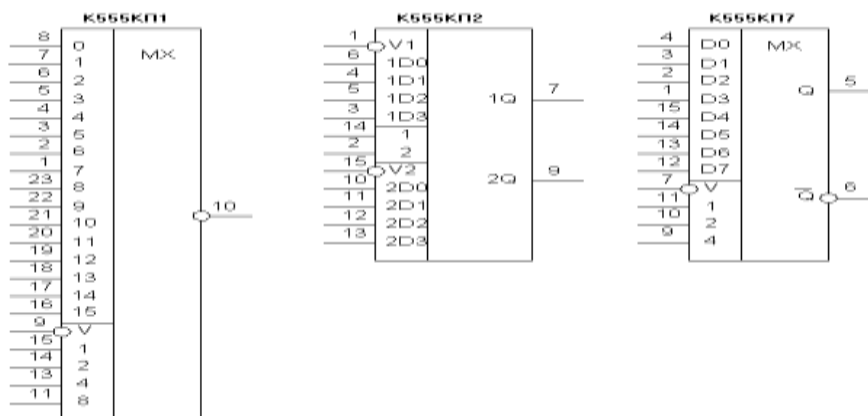


Рис. 3.35. Умовні графічні позначення мультиплексорів

3.7.4. Побудова демультимплексорів

Демультимплексор — функціональний комп'ютерний блок, використовується для комутації імпульсу з одного інформаційного вводу на один з n інформаційних виводів. Порядковий номер виходу, на який передається значення вхідного сигналу в кожному циклі машинного часу, визначається кодом адреси $A_0, A_1, A_2, \dots, A_{m-1}$. Адресні вводи та інформаційні вводи з'єднані з відношенням $n=2^m$ або $m=\log_2 n$. Термін «селектори даних» також використовується для позначення мультиплексорів і демультимплексорів. В умовних графічних позначеннях (рис. 3.36) функціонал демультимплексора визначається літерами DMX.

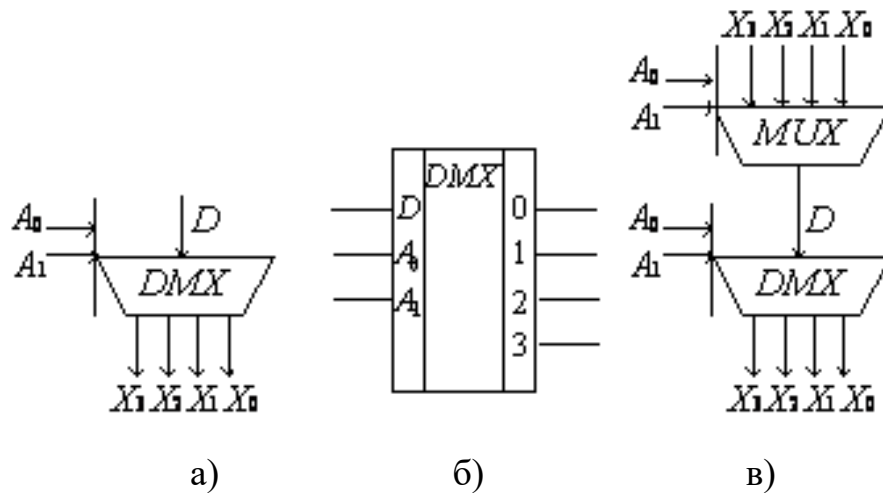


Рис. 3.36. Графічні зображення демультиплексорів:

а – на функціональних схемах; б – на діаграмах; в — типове підключення мультиплексора

Демультиплексори використовуються для виконання наступних операцій: комутації як однорядних, так і багаторозрядних шин; перетворення послідовного коду в паралель; реалізація логічних функцій та інші.

Принцип роботи дво-адресного демультиплексора на мікроопераційній мові наведена в таблиці. 3.7, де D – інформаційний вхід; F_0, \dots, F_3 є виходами внутрішнього декодера адреси.

Таблиця 3.7

1	0	F_0	F_1	F_2	F_3	X_0	X_1	X_2	X_3
0	0	1	0	0	0	$F_0 D$	–	–	–
0	1	0	1	0	0	–	$F_1 D$	–	–
1	0	0	0	1	0	–	–	$F_2 D$	–
1	1	0	0	0	1	–	–	–	$F_3 D$

На підставі даних таблиці 3.7 можна записати декілька рівнянь для інформаційних виводів:

$$X_0 = f_0 d = \bar{A}_1 \bar{A}_0 d; X_1 = f_1 d = \bar{A}_1 A_0 d; X_2 = f_2 d = A_1 \bar{A}_0 d; X_3 = f_3 d = A_1 A_0 d; \quad (3.12)$$

З використанням рівнянь (3.12) будуються схеми демультимплектора з внутрішнім перетворювачем (рис. 3.37, а) в поєднанні адресованих і вхідних перемінних на трьохвхідних І елементах (рис. 3.37, б). Схема демультимплектора з комбінацією адресних і вхідних змінних забезпечує високу швидкість, але вимагає використання логічних елементів з більшою кількістю входів.

Каскадування дозволяє комутувати один вхідний сигнал на будь-яку кількість вихідних ліній на основі низькошвидкісних послідовних мікросхем. Нехай необхідно демультимплексувати вхідний сигнал у n вихідних рядків, визначених m -розрядним адресним кодом, на основі типових « $1n \rightarrow$ » мікросхем менших розмірів. Для цього слід використовувати типові $L = n/n_1$ демультимплектори з кількістю адресованих входів $m_1 = \log_2 n_1$ кожен. Кількість старших розрядів адреси, що рівний різниці $m - m_1$, застосовується допоміжним «ведучим» де-мультимплексором, який розташований на першому рівні каскадної системи. Провідний демультимплексор створює альтернативну активацію одного з L де-мультимплексорів мікросхеми другого рівня. Каскадування \rightarrow демультимплексорів типу « $1 \rightarrow 4$ » для створення \rightarrow комутатора « $1 \rightarrow 16$ » показано на рис. 3.38.

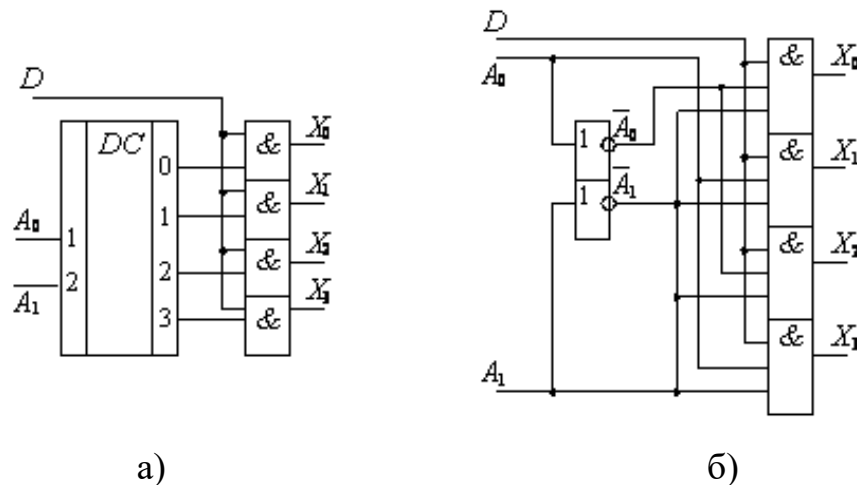


Рис. 3.37. Схема демультимплексорів:

а) – з внутрішнім дешифратором; б) – з комбінацією адресних і вхідних змінних

Нехай схема (рис. 3.37) має код адреси $A_3 A_2 A_1 A_0 = 1010$ і значення вхідного сигналу $D=1$.

Потім $D_2' = A_3 \overline{A_2} D = 1$ на виході провідного демультиплексора і на інших виходах встановлюються нульові значення.

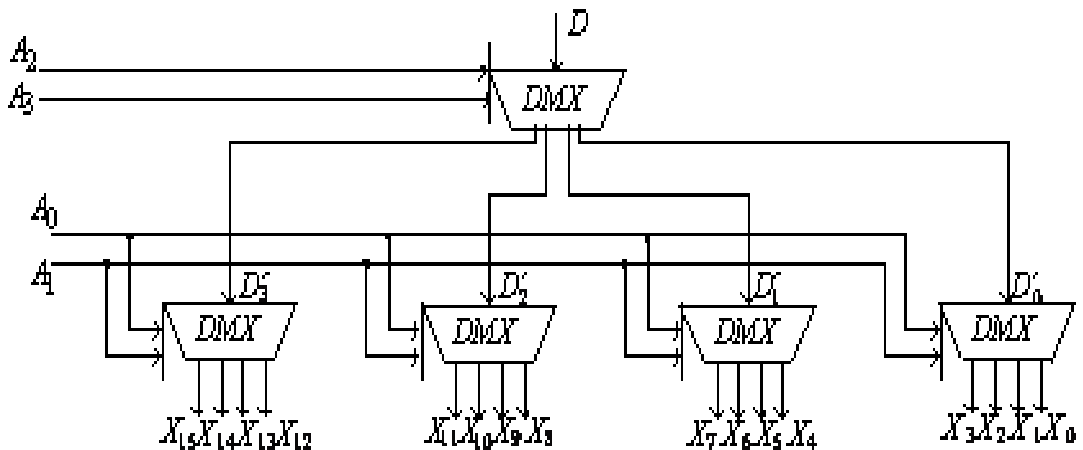


Рис. 3.38. Каскадні демультиплексори

Одиничне значення сигналу D_2' передається на вихід X_{10} підлеглого демультиплексора згідно $X_{10} = A_1 \overline{A_0} D_2 = 1$.

Як самостійні вироби на мікросхемах демультиплексори не виготовляються. Функція демультиплексора зазвичай реалізована в декодерах зі стробуючими входами (демультиплексорні декодери).

3.7.5. Шинне демультиплексування

Демультиплексування шини передбачає по чергове перемикання груп ліній від одного джерела інформації до кількох одержувачів. Ці типи мікрооперацій найчастіше реалізуються на основі однолінійних демультиплексорів. При виборі кількості та типу демультиплексора необхідно враховувати:

- кількість комутованих шин 2^m , де m - довжина коду адреси;
- скільки демультиплексорів, що визначають розрядністю n шин, що демультиплексуються;
- адресовані входи мультиплексорів з'єднані паралельно .

Схеми мультиплексора ввідної шини $D(n)$ для чотирьох вхідних шин $X(n)$, $Y(n)$, $Z(n)$ і $S(n)$ наведена на рис. 3.39.

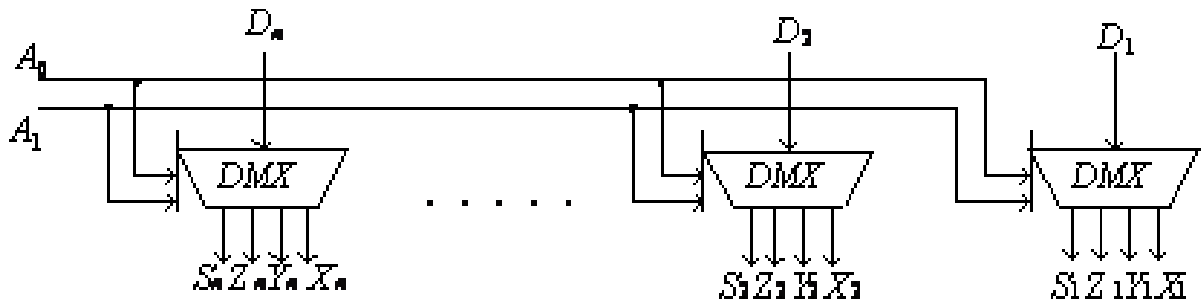


Рис. 3.39. Демультіплексор шини

Часте включення демультіплексорів і мультіплексорів для перемикання ввідних і вивідних шин n -розрядних регістрів A , B , C і D показано на рис. 3.40. В ALP таке перемикання забезпечує використання вмісту будь-якого регістра як першого аргументу суматора і збереження результату операції в будь-якому регістрі, заданому мікропрограмою команди, що виконується.

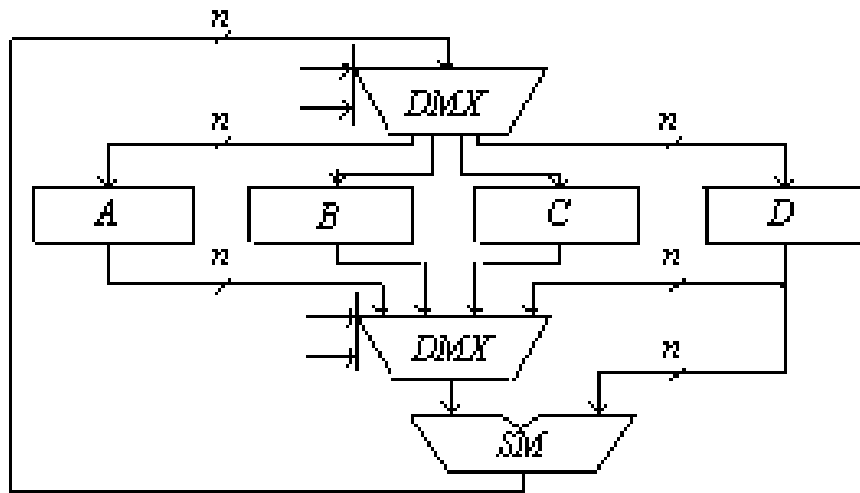


Рис. 3.40. Типове перемикання шин введення та виведення регістра ALP

РОЗДІЛ 4. ТИПОВИЙ ЦИФРОВИЙ ВУЗОЛ З ПАМ'ЯТТЮ

4.1. Загальні відомості про цифрові запам'ятовуючі пристрої

Функціональні Інтегральні схеми та запам'ятовуючі пристрої містять елементи пам'яті, здатні зберігати двійкову інформацію протягом тривалого часу. Завдяки елементам пам'яті, які зазвичай діють як перемикачі, стан запам'ятовуючих схем визначається не тільки конкретною комбінацією вхідних змінних в даний момент, але і станом пристрій перед передачею цієї комбінації, тобто вона визначається на основі послідовності вхідних сигналів. Тому термін « пристрій » використовується разом із терміном « пристрій запам'ятовуючого типу ». " послідовний тип ".

До запам'ятовуючих схем відносяться тригери, лічильники, регістри, запам'ятовуючі пристрої, накопичуючі суматори та ін .

Їх роботу розглядають в безрозмірному дискретизованому часі, для якого реальний час ділиться на пронумеровані інтервали, починаючи з певного моменту. Кожен такий інтервал часу називається тактовим інтервалом. Дана тривалість не впливає на робочі характеристики пристрою. Перехідні затримки, викликані перехідними процесами, зазвичай не враховуються , однак, коли тактова частота висока і порівнянна зі швидкістю пристрою , їх слід враховувати .

Більшість запам'ятовуючих засобів вони є синхронізованими, тобто мають спеціалізовані вводи, на які подаються синхроні (тактові) імпульси, які є свого роду виконавчими командами. Які визначають момент початку роботи пристрою, а значить, момент початку зворотного відліку годинника. Якщо пристрій асинхронний , то початок тактового інтервалу пов'язаний з моментом приходу будь-якого інформаційного сигналу.

Підтримка пристроїв зберігання даних зазвичай описується зв'язком їх станів у двох суміжних циклах t^n і t^{n+1} АБО t^{n-1} і t^n Відповідно до позна-

чення вхідних і вихідних сигналів діяти в межах такту, додавати номер такту. Так, запис $Q^n = I$ означає, що відповідає запиту Q в інтервалі $t^n < t < t^{n+1}$ є сигнал логічної одиниці.

Методи опису роботи запам'ятовуючих пристроїв схожі способи опису роботи комбінованих пристроїв - у вигляді таблиць і виразів отримав назву державних столів і функції переходу. Обидва виражають зв'язок між вихідним і вхідним сигналами в $(n+1)$ циклі, враховуючи стан пристрою в попередньому n -й ударі

При великій кількості входів і виходів таблиці станів заповнені вони громіздкі. Тому на практиці частіше використовують скорочені таблиці з колонками з вихідними сигналами в n -му циклі вимкнені, а стани в $(n+1)$ -му циклі занотовуються в загальному вигляді, наприклад $Q^{n+1} = \underline{Q}^n$. Отже значить вихідний сигнал змінює своє значення на протилежне (від 0 до 1 або від 1 до 0).

Як було сказано вище, запам'ятовуючі пристрої можуть бути синхронними і асинхронними.

Асинхронний перейти в новий стан відразу після надходження на входи певної комбінації інформаційних сигналів.

Синхронні пристрої мають вхід синхронізації і перемикаються тільки при появі на ньому сигналу синхронізації.

Вхід синхронізації може бути статичним або динамічним. У свою чергу обидва є прямими і зворотними.

Статичний вхід - це вхід, через який передається вплив сигналу на пристрій час його існування. Крім того, якщо сигнали на інформаційних входах змінюються під час існування сигналу синхронізації, то пристрій буде реагувати на ці зміни. У цьому контексті статичний вхід синхронізації еквівалентний введенню роздільна здатність в комбінованих пристроях. Статичні входи зазвичай також є входами, до яких подаються інформаційні сигнали.

Динамічний вхід - це вхід, через який ефект передається на пристрій лише тоді, коли вхідний сигнал змінюється з 0 на 1 або навпаки. Властивість

такого входу подібна до властивості диференціального ланцюга ізоляції перепадів напруги.

При необхідності вихідні написи доповнюються літерами та символами, що вказують на функціональне призначення висновки. Зверніть увагу, що для позначення використовуються однакові букви та символи виводів і на структурованих схемах, якщо виводи прямі, і доповніть позначення зворотною характеристикою, R , CR , якщо вони зворотні.

Для уникнення плутанини в таблицях стану батареї засобів, погодимося надавати змінні, що роблять деякі автори, а визначені значення сигналу в плюсовій логіці, тобто цифра 1 буде означати високий рівень напруги, а цифра 0 - низька напруга. Крім того, оскільки входи та вхідні сигнали до них зазвичай позначаються однаковими літерами, щоб розрізнити їх, ми введемо індекси годинника в позначення сигналів (наприклад C^{n+1} , Q^n).

Синтез запам'ятовуючих пристроїв базується на складанні спеціальних таблиць і карт переходів і їх обробці за певними правилами. Основні параметри запам'ятовуючих пристроїв характеризуються потужними властивостями, подібними до комбінованих пристроїв.

4.2 Тригери та їх характеристики

Самим простими цифровим автоматом з пам'яттю є тригери. *Тригер* — це пристрій послідовного типу з 2 стійкими рівноважними станами, призначений для зберігання та запису інформації. Під дією вхідного сигналу тригер може переходити з одного стану до іншого. Напруга на його виводі змінюється скачкоподібно.

Тригер має два виходи - прямий і зворотний. Кількість виводів залежить від структури і функцій тригера. За способом запису інформації тригери поділяються на асинхроннізовані та несинхроннізовані. В несинхронних тригерах

інформація буде записуватися постійно і визначається на основі інформаційних сигналів, що діють на входи в даний момент. Якщо інформація вводиться в тригер тільки тоді, коли т.зв синхронізуючий сигнал, то такий тригер називається *синхронізованим* або *тактовим*. Крім інформаційних входів синхронізовані тригери мають тактовий (синхронізаційний) вхід. У цифровій техніці прийняті наступні позначення тригерних входів і виходів:

- Q - вихід прямого тригера;
- \bar{Q} - вихід реверсного тригера;
- S - окремий вхід пристрою в одиночному стані (високий рівень напруги на прямому виході Q);
- R - окремий вхід установки в нульовому стані (низький рівень напруги на прямому виході Q);
- D - інформаційний вхід;
- C - вхід для синхронізації;
- T – лічильний (тактуючий) вхід.

Велике поширення в цифрових пристроях отримав RS-тригер з двома входами настройки, тактований D-тригер і рахунковий T-тригер. Розглянемо функціональність кожного з них.

Несинхронний тригер RS. Розрізняють тригери RS з прямим і зворотним входами. Їх схеми та позначення наведені на рис. 4.1. Тригери цього типу побудовані на двох логічних елементах: 2АБО-НІ - тригер з прямими входами (рис. 4.1, а), 2І-НІ - тригер зі зворотними входами (рис. 4.1, б).

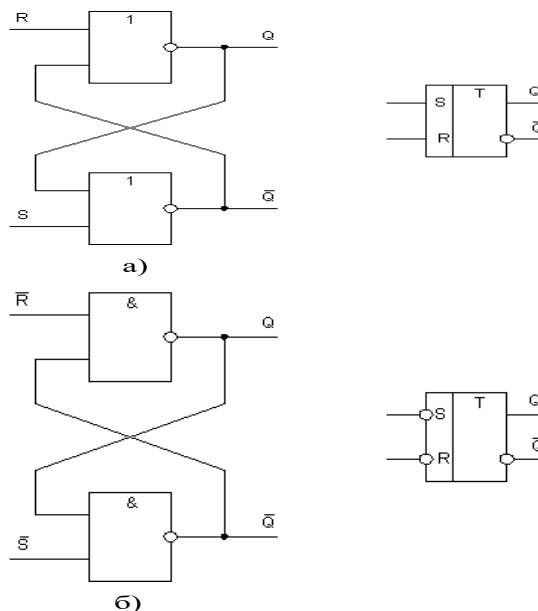


Рис. 4.1. Схеми та маркування тригерів RS

Вихід кожного елемента з'єднаний з одним із входів іншого елемента, забезпечуючи тригеру два стабільних стани. Таблиця істинності показана в таблиці 4.1 для кожного з цих тригерів.

У таблиці 4.1 сигнали Q' та Q'' вказують рівні, що були присутні на виході тригера до того, як активні рівні були застосовані до його входів. Рівень, що присутній на вході логічного елемента, є *активним і визначає який логічний рівень вивідного сигналу*.

Для визначених елементів АБО-НІ діючий рівень вважається високим, а для елементів І-НІ – низьким.

Рівні, які при застосуванні до одного з входів не змінюють логічний рівень на виході елемента, називаються *пасивними*. Рівні Q^{t+1} та Q''^{t+1} представляють логічні рівні на виході тригера після застосування інформації до його входів.

Для тригера з прямими входами $Q^{t+1} = 1$ в $S = 1$ і $R = 0$; $Q^{t+1} = 0$ на $S = 0$ і $R = 1$; $Q^{t+1} = Q'$ о $S = 0$ $R = 0$ і Коли $R = S = 1$ стан тригера буде невизначеним (*), тому що під час роботи інформаційних сигналів логічні рівні на виході тригера однакові $Q^{t+1} = Q''^{t+1}$, і після кінця їх роботи тригер приймає будь-який із стійких станів з рівною ймовірністю. Тому таке поєднання заборонено.

Таблиця 4.1 – Таблиця роботи несинхронних RS тригерів

C	P	Q	\bar{Q}^t	Q^{t+1}	\bar{Q}^{t+1}	\bar{S}	\bar{R}	Q	\bar{Q}^t	Q^{t+1}	\bar{Q}^{t+1}
0	1	0	1	0	1	0	1	0	1	1	0
1	0	0	1	1	0	1	0	0	1	0	1
0	0	0	1	0	1	0	0	0	1	*	*
1	1	0	1	*	*	1	1	0	1	0	1
0	1	1	0	0	1	0	1	1	0	1	0
1	0	1	0	1	0	1	0	1	0	0	1
0	0	1	0	1	0	0	0	1	0	*	*
1	1	1	0	*	*	1	1	1	0	0	1

Режим $S = 1, R = 0$ який називається режимом запису 1 (тому що $Q^{t+1} = 1$); і $R = 1$ mode $S = 0$ - режим реєстрації 0. Режим $S = 0, R = 0$ має називу режим збереження інформації, оскільки інформація на виході залишається незмінною. Для тригера з інверсними входами режим реєстрації логічної 1 реалізовано в $S = 0, R = 1$, режим реєстрації логічного 0 реалізовано в $S = 1, R = 0$. Коли $R = S = 1$ передбачено зберігання інформації. Поєднання $R = S = 0$ заборонено.

Тактовий тригер D. Має інформаційний вхід D і вхід синхронізації C і працює згідно таблиці стану 4.2. Після застосування сигналу синхронізації C тригер встановлюється в стан $Q^{t+1} = D$ і підтримує його до надходження наступного сигналу C . При рівні сигналу на виводі $C = 0$, стан тригера є стабільним і не визначається від рівня сигналу на ввіді інформації. При цьому на входи тригера RS з інверсними входами (елементи 3 і 4) надходять пасивні рівні ($R = S = 1$). Коли до входу застосовується синхронізація $C = 1$, сигнали на основному виході буде повторювати інформацію, що подається на вхід $C = 0$ $Q^{t+1} = Q^t$ і при $C = 1$ $Q^{t+1} = D$. тут Q^t означає логічний рівень на прямому виході до застосування імпульсу синхронізації та Q^{t+1} означає логічний рівень на

цьому виході після застосування імпульсу синхронізації. Тому тригер затримує двійкові сигнали, які надходять на ввід D, на час проходження сигналу синхронізації C. Тому його також називають тригером затримки. Своєю назвою він зобов'язаний слову Delay - відкладати. Одна з можливих конструктивних схем одноктактного тригера D і його умовне позначення наведено на рис. 4.2.

Таблиця 4.2 – Таблиця істинності Clock D-flip

D	Q	Q^{t+1}
0	0	0
0	1	0
1	0	1
1	1	1

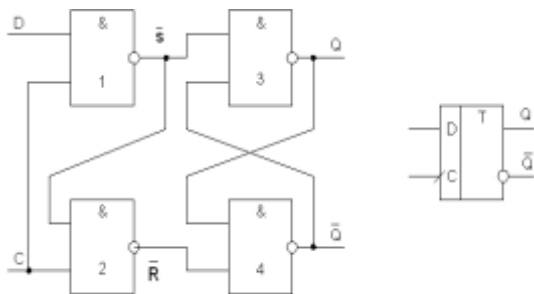


Рис. 4.2. Схема та маркування D

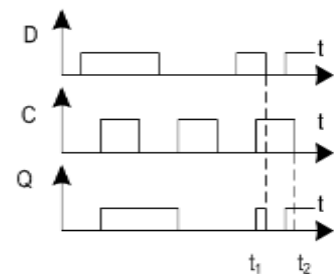


Рис. 4.3 D Часові графіки запуску зі статичним входом синхронізації

У такому тригері вихідний сигнал затримується відносно вхідного на час паузи між сигналами синхронізації. Для стабільної роботи тригера необхідно, щоб інформація на вході не змінювалася протягом імпульсу синхронізації.

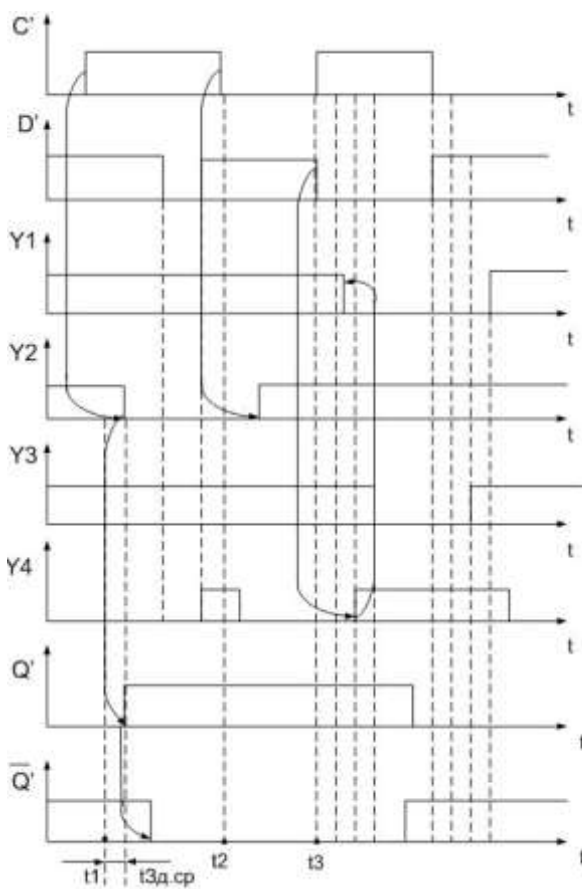
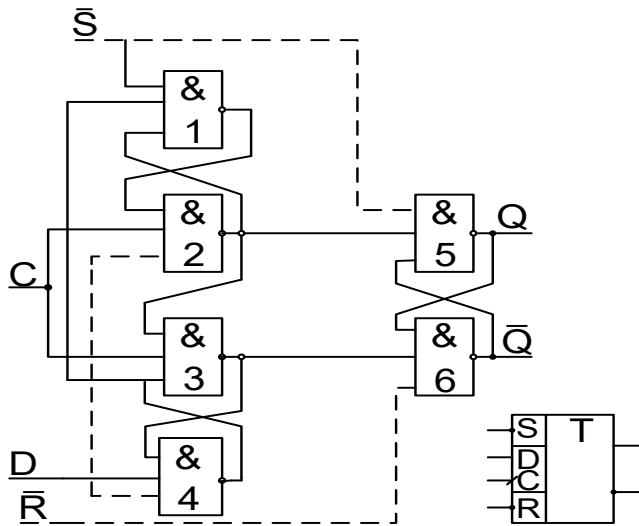
Годинникові тригери типу D можуть бути оснащені входом статичної та динамічної синхронізації. У першому інформація записується при рівні сигналу $C=1$. На малюнку 4.2 показана найпростіша тригерна схема D з прямим

статичним входом С. Коли $C = 0$, логічні елементи LE1 і LE2 вимкнені ($\bar{S} = \bar{R} = 1$). Асинхронний RS-тригер (LE3 і LE4) є осередком пам'яті з реверсивними входами, що працює в режимі зберігання. Робота тригера показана на рис. 2. 4.3. Сигнал $C=1$ дозволяє надсилати інформацію від входу D до комірки пам'яті. Буває так. Якщо $D=0$, то $S = \bar{S} = 1$, $R = \bar{R} = 0$ і $Q^{t+1} = 0$; Якщо $D=1$, то $S = \bar{S} = 0$, $R = \bar{R} = 1$ і $Q^{t+1} = 1$.

Зміни інформації на вході D під час роботи сигналу синхронізації передаються на вихід тригера ($t_1 < t < t_2$). Зазвичай це небажано. Тому, коли необхідно записати інформацію тільки в певний момент, використовуються або дуже короткі С-імпульси, або D-тригер з динамічним С-входом.

У схемах з оперативним керуванням інформація записується тільки під час спаду напруги на ввіді синхронізації. Динамічні входи представлені на діаграмах трикутником або лінією. Якщо верхня частина звернена до мікросхеми, то комірка працює по фронту ввідного імпульсу. У даному тригері вихідна інформація може бути задержана на один цикл в порівнянні до ввідної інформації.

Найбільш популярною схемою запуску D з динамічною синхронізацією є схема на основі трьох асинхронних тригерів RS (рис. 4.4, а). Тригер на LE5 і LE6 служить коміркою пам'яті, а решта два - на LE1, LE2 і LE3, LE4 - виконують допоміжну функцію, утворюючи вузол управління. Крім входів D і С можуть бути також входи асинхронного блоку в одиничному і нульовому станах (S і R). Асинхронність цих вхідних даних підкреслюється в умовному графічному позначенні, відокремлюючи їх від області міток С роздільниками.



а)

б)

Рис. 4.4. D-тригер із входом динамічної синхронізації:

а) діаграма та УГП, б) графіки робочого часу

Принцип дії тригера пояснюється тимчасовими графіками напруги на виходах логічних елементів (рис. 4.4, б). Графіки побудовані з урахуванням часу затри-

мки перемикання $LE\ t_{3d, sr}$, що дозволяє підвищити наочність причинно -наслідкових зв'язків у тригері. Нехай $D=1$, $C=0$ у початковому стані ($t < t_1$). На входи $LE2$ і LEZ подається сигнал $C=0$, тому вони вимикаються ($w_2 = w_3 = 1$), комірка пам'яті ($LE5$, $LE6$) працює в режимі зберігання. Для ясності передбачається, що на входах $LE4$ є дві одиниці, що означає, що $y_4 = 0$, а $LE1$, пов'язаний з його виходом, вимкнено ($y_1 = 1$).

З надходженням сигналу синхронізації ($t = t_1$) в найближчий інтервал часу t одружуватися $LE2$ увімкнено, оскільки його входні дані керуються рівнями одиниць. Елемент LEZ не змінює свого стану. Під час другого періоду дискретизації можна перемикати $LE1$, LEZ , $LE5$, на входи яких надійшов сигнал зі станом $y_2 = 0$. Оскільки $LE1$ і LEZ були вимкнені, їх стан не змінюється. Вимкнеться лише $LE5$ ($Q' = 1$). Під час третього періоду вибірки \bar{Q}' вмикається лише $LE6$ ($= 0$).

На цьому завершуються перехідні процеси в тригері. Подібним чином будуться графіки для інших комбінацій зовнішніх сигналів C і D .

З рис. 4.4b видно, що, по-перше, тригер фактично працює відповідно до функції переходу $Q^{t+1} = D$. По-друге, він перемикається лише тоді, коли логічний перехід 0-1 надходить на вихід C , зміна інформації на вході D як під час сигналу синхронізації ($C = 1$), так і під час паузи ($C = 0$) не змінює стан тригера. Таким чином, вхід синхронізації є прямим і динамічним. Це дозволяє тригеру фіксувати вхідну інформацію тільки по фронту імпульсу C (рис. 4.5).

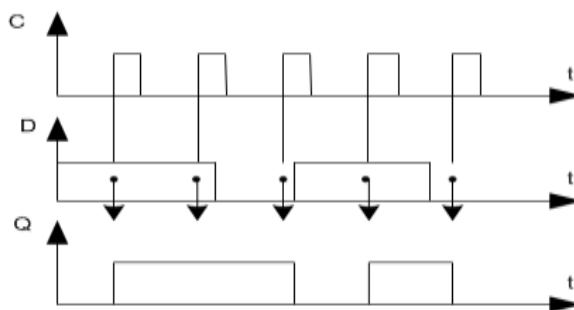


Рис. 4.5. Тимчасові діаграми D-тригера з динамічним входом синхронізації

Динамічна синхронізація дозволяє використовувати D-тригер як режим підрахунку, що досягається підключенням D-входу до виходу \bar{Q} (Рис. 4.6). До моменту надходження на інформаційний вхід \bar{Q} наступного сигналу синхронізації ($C=1$) $D=$
 Q^t , тому в результаті впливу сигналу C стан тригера стане $Q^{t+1} = D = Q^t$. Вираз $Q^{t+1} = Q^t$ є функцією переходу T-тригера.

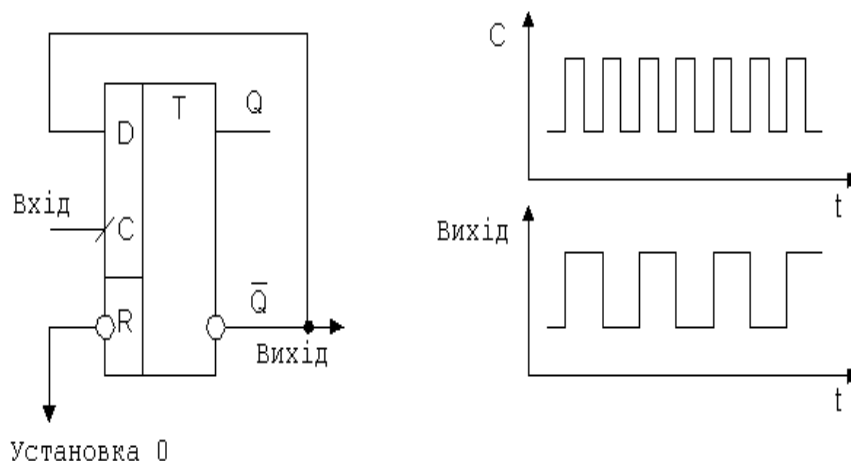


Рис. 4.6. Використання тригера D як тригера T

Швидкість запуску в лічильному режимі залежить від мінімального періоду проходження лічильних імпульсів, який, у свою чергу, відповідає логічній глибині, тобто найдовшому шляху проходження сигналу. З рис. 4.4 логічна глибина (з урахуванням зв'язку D з \bar{Q}) дорівнює п'яти: ЛЕ2-ЛЕ5-ЛЕ:-ЛЕ4-ЛЕ1 і

$$F_{\max} = 1/5 \text{ цд.сп}$$

Мінімальна тривалість імпульсів лічильника дорівнює мінімальній тривалості сигналів керування на входах елемента пам'яті (асинхронного тригера RS) і становить $t_{i,\min} = 2t_{3d.sp}$.

Розчіплювачі типу D також можуть бути двоступеневими. Для цього використовуються D-тригери зі статичною синхронізацією (рис. 4.7).

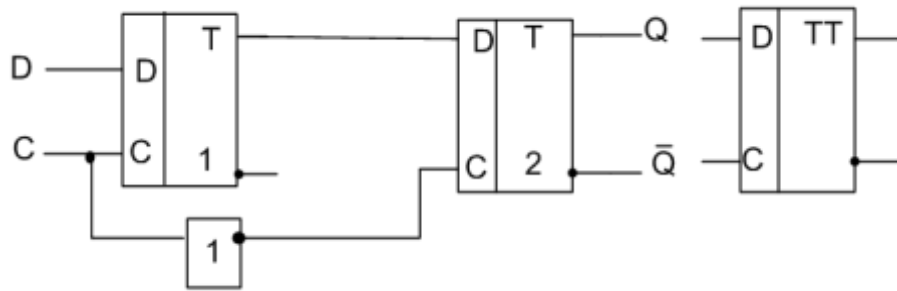


Рис. 4.7. Двоступеневий D-спусковий механізм

Двоступінчастий тригер D працює так само, як і двоступеневий тригер RS. При активному сигналі $C=1$ інформація з входу D записується на тригер першого ступеня, а при $C=0$ - на другий ступінь. При цьому другий каскад фіксує лише стан першого каскаду, який виник на момент зриву імпульсу синхронізації, тобто двоступенева конструкція забезпечує властивість динамічної синхронізації.

Підрахунок T-тригера (підрахунок тригера) має тільки один інформаційний вхід, який позначений +1. Це також називається тригером входу лічильника (рис. 4.8). Стан змінюється кожного разу, коли він отримує один фрагмент інформації (1 або 0), тобто виникає логіка $Q^{t+1} = Q^t$. Якщо вхід тригера прямий, то зміна стану відбувається на початку падіння 0-1, якщо зворотний, то 1-0. Умовне графічне зображення T-тригера з прямим арифметичним введенням і часові діаграми, що ілюструють його роботу, показані на рис.

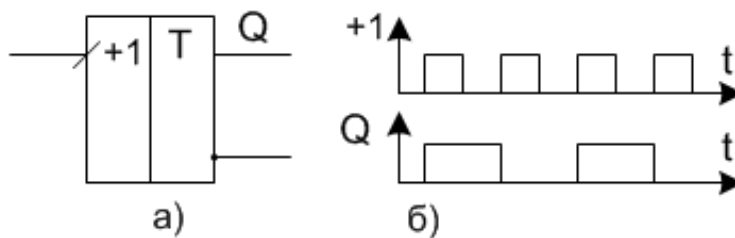


Рис. 4.8. Тригер підрахунку

Вмикання кожного вхідного імпульсу можна інтерпретувати як підрахунок їх за модулем 2 (після кожного непарного імпульсу на виході $Q = 1$, а після парного $Q = 0$). Ця властивість тригера використовується в лічильниках і акумуляторах. Розчіплювачі Т не виготовляються в комплексній конструкції як самостійні вироби, тому що їх функції можуть виконувати розчеплювачі типу Д і ЖК. Т-тригер може бути створений з тактованого D-тригера, якщо його зворотний вихід підключено до інформаційного входу. Як видно на графіку, частота імпульсу на виводі Т-тригера вдвічі менша за частоту імпульсу на ввіді, тому даний тригер можна використовувати як подільник частоти та двійниковий лічильник.

У серію мікросхем, що випускаються, входять також універсальні тригери JK. При правильному підключенні вхідної логіки тригер JK може виконувати функції будь-якого іншого типу тригера.

Тригер

JK.

Тригер JK має два інформаційних входи (JK) і вхід синхронізації. Це працює відповідно до функції переходу $Q^{t+1} = \bar{K}Q^t \vee J\bar{Q}^t$ або відповідної таблиці станів (рис. 4.9).

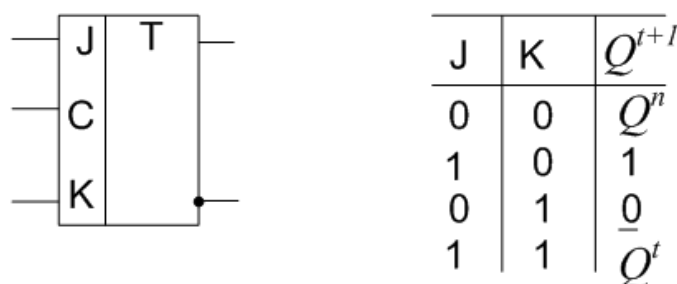


Рис. 4.9. Тригер JK: а) умовне графічне позначення, б) таблиця станів

При надходженні сигналу синхронізації С і наявності на інформаційних входах сигналів $J = K = 1$ тригер переходить у протилежний стан $Q^{t+1} = \bar{Q}^t$, тобто працює в лічильному режимі.

В інших випадках він працює як синхронний RS-тригер із прямими входами, якщо вхід J вважається входом S, а вхід K вважається входом R. Тригер JK також може працювати як тригер D, якщо не враховувати випадки еквівалентності сигналів J і K.

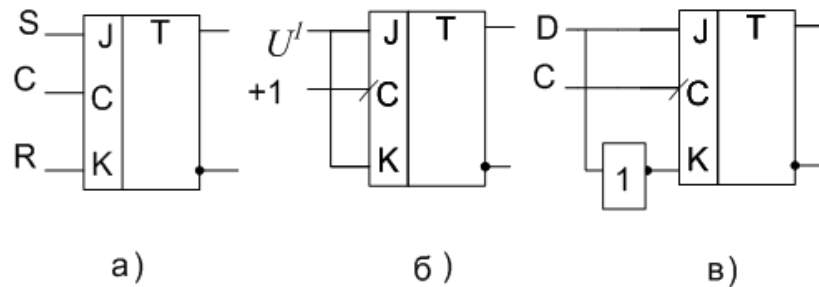


Рис. 4.10 Розчіплювач JK як розчіплювач типу RS (а), Т (б) і D (с).

На рис. 4.10 показані схеми його використання в режимах: а) синхронний RS-тригер з $J = K = 1$, б) Т-тригер, в) D-тригер.

На практиці найчастіше використовують двоступеневі тригери JK. Їх структурна схема та умовні графічні позначення представлені на рис. 4/11. За складом і підключенням схема подібна до двоступеневих тригерів RS. До входів першого каскаду додається тільки зворотний зв'язок з виходів другого каскаду, що виключає невизначений стан з комбінацією $J=K=1$. Для створення зворотного зв'язку тригер першого каскаду містить два входи S і R, з'єднаних кон'юнкцією.

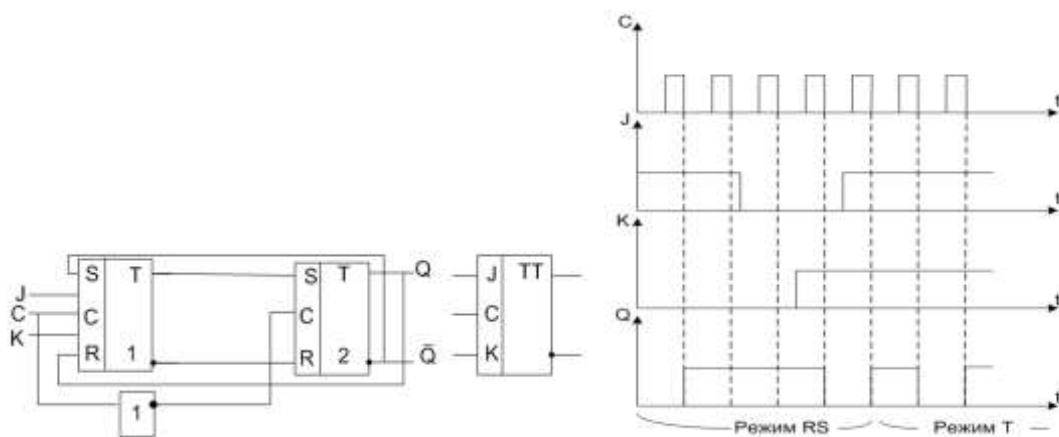


Рис. 4.11 Двоступеневий розчіп JK Рис. 4.12. Тимчасові діаграми, що ілюструють роботу тригера JK,

В цілому організація роботи двокаскадного тригера JK така ж, як і у всіх розглянутих раніше двокаскадних конструкцій: почерговий перехід каскадів з режиму прийому інформації в режим зберігання, що визначається сигналом синхронізації. Однак перехід тригера в той чи інший стан залежить не тільки від інформаційних сигналів J і K, а й від його стану в попередньому циклі. Продемонструємо це на прикладі роботи, коли $J = K = 1$. До появи наступного сигналу $C=1$ на виході $Q^t=0$ обидва входи S першого тригера будуть на одному рівні, а один з R входи будуть нульовими. Тому при надходженні сигналу $C=1$ перший тригер буде встановлено в одиночний стан, а після його завершення ($C=0$) другий тригер буде встановлено в одиничний стан ($Q^t = 1$). Внаслідок осової симетрії діаграми, якщо $Q^t = 1$, то $Q^{t+1} = 0$. Отже, має місце лічильний режим: $Q^{t+1} = \overline{Q^t}$. Часові діаграми, які зображують роботу тригера JK, наведені на рис. 4.12.

Рис. 4.13 наведена функціональна схема універсального тригера ІМС К133ТВ1. Який має 3 входи J і K, з'єднаних логічною операцією I, і асинхронні входи S і R, що дозволяють встановити тригер в одиничний і нульовий стан не залежить від сигналів на інформаційних і синхронізаційних входах. Перший ступінь (провідний) випуску складається з елементів D1 і D2, а другий ступінь (підпорядкований) випуску складається з елементів D3 ... D8. Блок керування тригером RS (D3...D6), крім своєї звичайної ролі, також виконує функцію інвертора сигналу C для другого ступеня (D5, D6).

Багато серій цифрових інтегральних схем разом із двокаскадними тригерами містять однокаскадні динамічно синхронізовані інтегральні схеми. З точки зору функціонування відмінностей між ними немає. Однак це забезпечує різноманітність деяких споживчих властивостей тригерів. Наприклад, тригер, що міститься в ІМС К155ТВ15, має один прямий (J) інформаційний вхід і інший інверсний (K), вхід синхронізації динамічний. Ця функція дозволяє перетворити його на тригер D з прямими входами, просто з'єднавши виходи J і K.

Швидкість тригерів JK (в режимі підрахунку) залежить від конкретної схеми і зазвичай трохи нижче швидкості тригерів D.

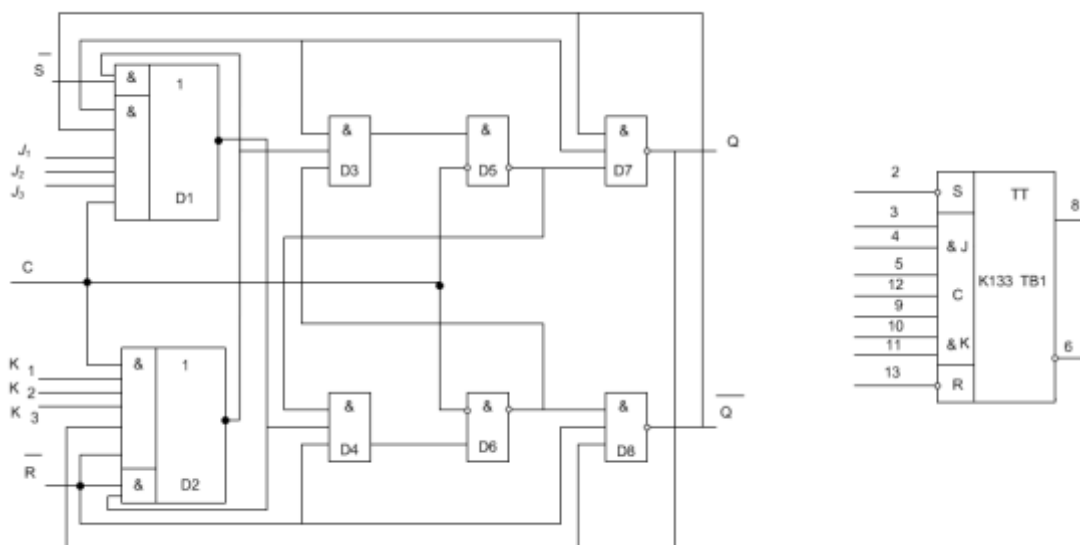


Рис. 4.13. Схема універсальної спускової системи JK K133TV1

Як згадувалося раніше, цифрові мікросхеми CMOS мають ширший набір основних компонентів, ніж мікросхеми будь-якої іншої технології. Це визначило схемотехнічну специфіку пристроїв КМОПТЛ. Зокрема, стало можливим введення у вихідні кола двонаправлених перемикачів і отримання таким чином третього стану (високого імпедансу) на входах. Введення ключів до внутрішніх інформаційних ланцюгів дозволило спростити реалізацію складних тригерів. На рис. На малюнку 4.14 показано конструкцію, що має характеристики тригера D із статичним входом синхронізації та використовується в різних типах двокаскадних тригерів.

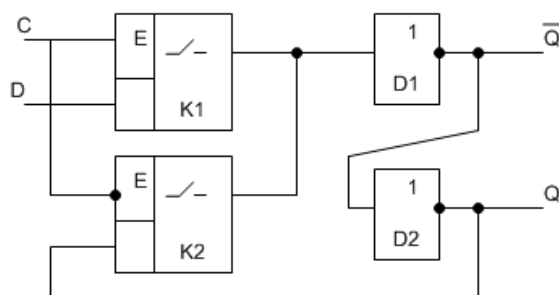


Рис. 4.14. D-тригер на основі двосторонніх клавiш

Його комірка пам'яті створена за допомогою двох інверторів із взаємним зворотним зв'язком (D1, D2). Крім того, один ланцюг зворотного зв'язку (від виходу D2 до входу D1) перемикається двонаправленим ключем К2. Ланцюг замикається, коли подається сигнал $C=0$. Ключ К1 має прямий вхід дозволу Е, тому, коли $C=0$ він відкритий, а комірка пам'яті з інформаційного входу D. Тригер знаходиться в режимі зберігання. Якщо використовується вхід $C=1$, інформація з входу D буде надходити на виходи Q \bar{Q} і статус інверторів 4.15 показаний приклад схеми JK-тригера, в каскадах якого використовується описаний вище D-тригер.

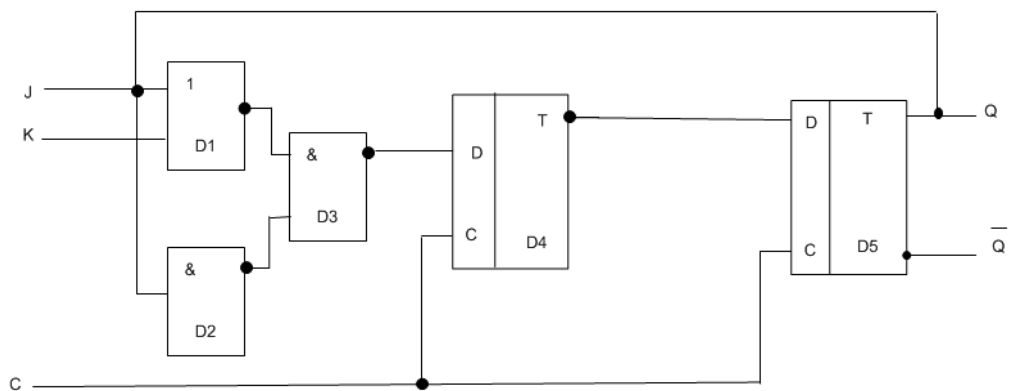


Рис. 4.15. Схеми тригерів JK, в яких використовується тригер D

Відмінною рисою тригерів з двонаправленими ключами є критичність їх роботи по тривалості, фронту (спаду) імпульсу C. Дуже повільна зміна сигналу C може призвести до замикання клавiш K1 і K2 в часі в порівнянні з затримкою час спрацьовування тригера. У результаті тригер буде в невизначеному стані після завершення процесів переходу. Цю особливість необхідно брати до уваги, але практичні умови для дуже поганих синхронізуючих імпульсів дуже рідкісні. Наприклад, для тригера K561ТВ1 допустима тривалість крутизни і спаду імпульсу синхронізації становить 5 мкс, а тривалість спадів на виходах будь-якої мікросхеми серії K561, яка може служити джерелом сигналу C, не перевищує кількох десятків наносекунд.

4.3 Регістри

Регістри — цифрові пристрої, призначені для прийому, зберігання та виведення інформації у двійковому коді (двійкові числа, слова). Вони будуть використовуватися як безадресні запам'ятовуючі пристрої, перетворювачі та генератори коду, пристрої тимчасова затримка цифрової інформації, дільники частоти тощо.

Щоб зареєструвати n - розрядне число, необхідно мати реєстр n тригерів. Кожен з них зберігає один біт інформації. Тригери можуть бути пов'язані разом за допомогою ланцюгів керування та передачі. У звичайних графічних реєстрових зображеннях мітки інформаційних входів і виходів нумеруються в природному порядку.

Усі реєстри залежно від способу отримання та видачі інформації поділяються на такі види:

- з паралельними збір і доставка;
- з послідовним прийомом і видачею;
- з комбінованим самовивозом і доставкою.

Реєстри с паралельний прийом і видача називаються реєстрами пам'яті, а всі інші - реєстрами зсуву. За напрямком руху (передачі) інформації розрізняють односторонні та реверсивні реєстри.

4.3.1. Принцип роботи реєстрів пам'яті

Регістри пам'яті являють собою набір *синхронних тригерів* з незалежними інформаційними входами. Тригери підключаються виключно через керуючі сигнали (синхронізація, дозвіл запису на вихід, високий імпеданс тощо). Введення (запис) і виведення (зчитування) інформації здійснюється одночасно в усіх (розрядів за допомогою паралельного коду. Введення). подається синхронізуючим імпульсом. З приходом чергового такого імпульсу відбувається відновлення збереженої інформації. Якщо вхідна інформація є парафазною, тобто кожна змінна отримані одночасно в прямій і

оберненій формі , будуть використані *Тригери RS або JK* (рис. 4.4, а), якщо однофазні, тригери *D* (рис. 4.16, б). Зчитування може відбуватися в прямому та зворотному кодах, тобто з прямого та зворотного тригерних виходів . У регістрах, призначених для роботи на інформаційній магистралі, як вихідні (буферні) каскади будуть використовуватися елементи з трьома станами , керовані сигналом дозволу на читання ERD .

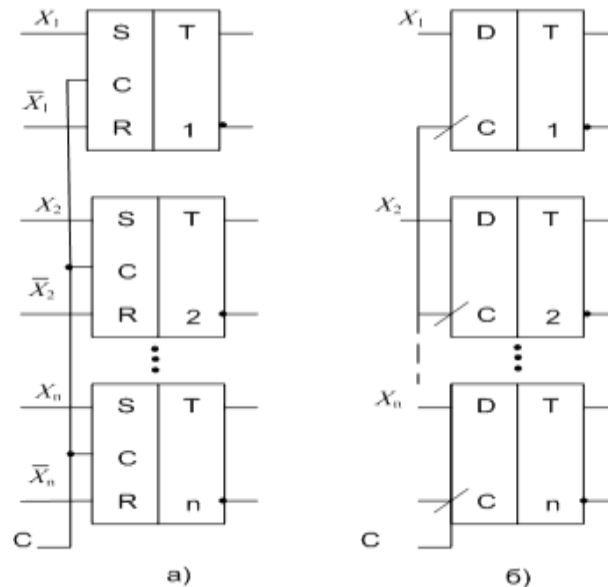


Рис. 4.16. Регістри пам'яті на основі синхронних тригерів *RS-(a)* і *D (b)*

Типовим прикладом може бути мікросхема 555 IR 15 (рис. 4.17). Це чотирирозрядний регістр пам'яті з однофазними інформаційними входами D_1, D_2, D_3, D_4 реєстрація, через яку це відбувається відповідно до нахилу 0-1 сигналу синхронізації C , якщо обидва входи це дозволяють, RE є низьким НАПРУГА .

Застосування одного рівня принаймні до одного входу RE змушує регістр перейти в режим зберігання. Для зчитування інформації, що зберігається в регістрі, низька напруга подається на обидва входи $ERD (1,2)$, інакше виходи вимикаються, що відповідає високоімпедансному входу хвороба

Збільшення швидкості передачі регістру досягається паралельним підключенням керуючих входів кількох мікросхем.

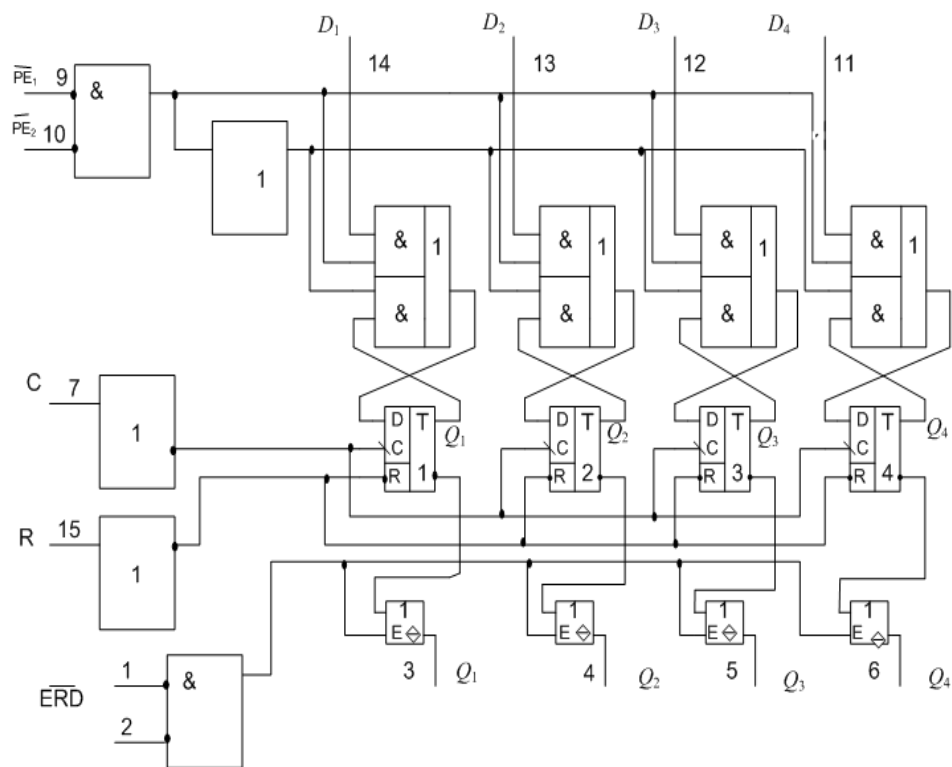


Рис. 4.17. Регістр пам'яті 555 IR 15

4.3.2 Регістри зсуву

Регістри зсуву мають властивість зміщувати збережену в них інформацію ліво (у бік старших розрядів) або вправо (у бік молодших розрядів) на задану кількість розрядів. Тому тригери в них з'єднані послідовно. Суть зсуву полягає в тому, що з надходженням імпульсу синхронізації тригерний вміст кожного розряду перезаписується на сусідній розряд. Число зсуву визначається кількістю отриманих імпульсів синхронізації. Фактичний напрямок зсуву регістра визначається напрямком між з'єднаннями тригера. Однак, з точки зору зсуву, щодо числової бітової сітки, той самий регістр, який має односторонній зв'язок між тригерами, може бути правим регістром зсуву та лівим регістром зсуву, залежно від порядку розміщення інформації.

Регістри побудовані на *тригерах з входом динамічної синхронізації* або з *внутрішньою затримкою*. Ця умова є обов'язковою. В іншому випадку при

надходженні сигналу синхронізації інформація може зрушитися не на один біт, а на стільки разів, скільки тригери встигають переключитися за час дії цього сигналу.

Рис. 4.18 показано схематичне, умовне графічне позначення та часові діаграми роботи трибітового регістра зсуву на тригерах D. Він реалізує послідовний метод прийому та послідовної видачі. Передбачається, що *вхідна інформація надходить у малих прямих бітах*, тобто вхідний тригер є тригером вищого порядку, а вміст регістра зчитується з виходу Q тригера молодшого порядку.

На кожному фронті імпульсу C двійкова змінна, що зберігається в i -му тригері, вводиться в $(i - 1)$ -й тригер. Оскільки входи C є динамічними, наступні зміни рівнів напруги на входах D у межах даного сигналу $C=1$ не призводять до нових тригерних перемикачів. Таким чином, коли надходить один імпульс C , інформація в регістрі буде зміщена на один біт вправо: змінна x із входу 0 буде записана для запуску D_3 . Вміст D_3 передається тригеру D_2 , а попередній вміст D_2 передається в D_1 . Для повного завантаження регістра новою інформацією необхідно використати три (відносно кількості розрядів регістра) імпульси синхронізації. Наступні C імпульси забезпечують висновок записаної інформації з тригерного виходу D_1 .

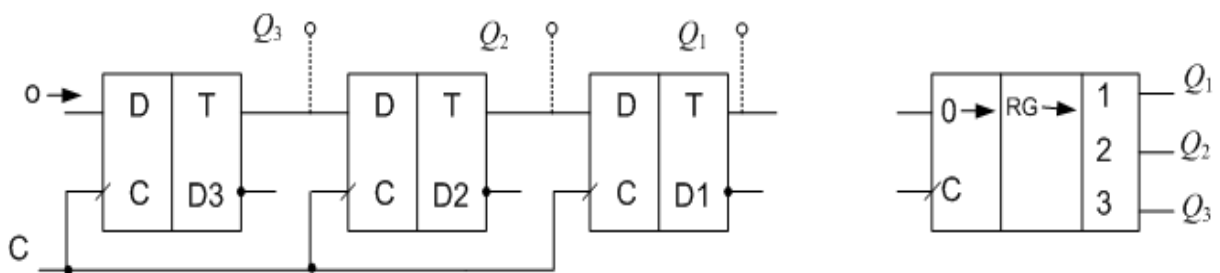


Рис. 4.18 Схема трирозрядного регістра зсуву

Якщо імпульси синхронізації надходять безперервно з періодом T_s , то вихідний сигнал Q повторює вхідний із затримкою nts , а регістр діє як дискретний пристрій затримки цифрової інформації. Виходи Q -тригерів зазвичай мають зовнішні контакти, з яких можна видалити збережену інформацію. Такий регістр можна використовувати як перетворювач послідовного коду в паралельний.

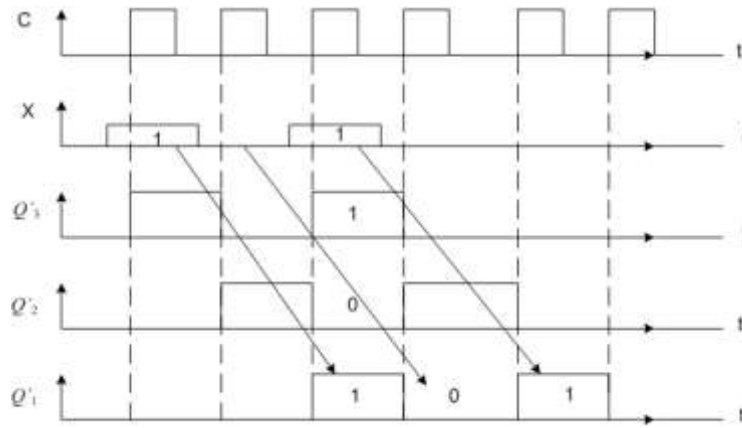


Рис. 4.19 Тимчасові діаграми трирозрядного регістра зсуву

Реверсивні регістри будуть використовуватися для зміни напрямку зсуву без зміни порядку надходження інформаційних сигналів. Принцип їх побудови полягає в переключенні входу i -го тригера на вихід $(i-1)$ або $(i+1)$ -го тригера (рис. 4.20). Тут при керуючому сигналі $E=1$ включається $(i-1)$ -й тригера, а при $E=0$ підключається $(i+1)$ -й тригера відповідно, встановлюється режим зсуву вліво або вправо.

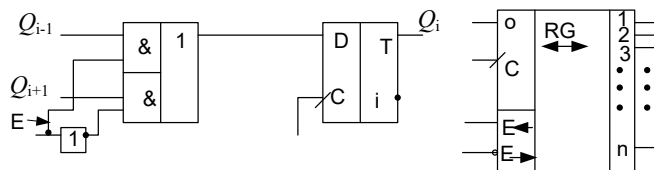


Рис. 4.20. Реверсивні регістри міжтригерні з'єднання

Функціональні можливості реєстраторів значно розширюються після впровадження режиму паралельного запису інформації. Такі регістри можуть додатково функціонувати як регістри пам'яті, а також паралельно перетворювачам послідовного коду. На рис. 4.21 зображено схему такого регістра. Монтується на тригерах JK з асинхронними входами S і R . Паралельна реєстрація двійкової інформації здійснюється за допомогою суматорів, керованих вхідними змінними D . Залежно від значення D_i команда запису C_2 надходить на вхід S або R вхід i -го тригера. У режимі перемикання передач C_1 служить виконавчим сигналом.

Не допускається одночасне надходження сигналів синхронізації C_1 і C_2 . Щоб сигнали C_1 і C_2 не впливали на регістр при їх випадковому збігу, тобто щоб виключити накладення різних режимів роботи, в багатьох практичних схемах вводиться режимний контроль.

Багато сучасних регістрів містять вихідні ключі з високим імпедансом, що полегшує зв'язок з інформаційними магістралями. Споживчі можливості реєстрів ще більші, якщо ключі двонаправлені. Введення ключів також у вхідні схеми реєстру дозволяє істотно спростити організацію двосторонньої передачі інформації між двома магістралями з можливістю їх попередньої обробки. Така обробка може включати зсув двійкового числа n цифр вліво або вправо. Якщо при цьому знаки числа не втрачаються (що можливо, коли розрядність регістра більше розрядності числа), зсув еквівалентний множенню або діленню числа відповідно на 2^m . Прикладом регістра з двонаправленою передачею є IC 564IR6.

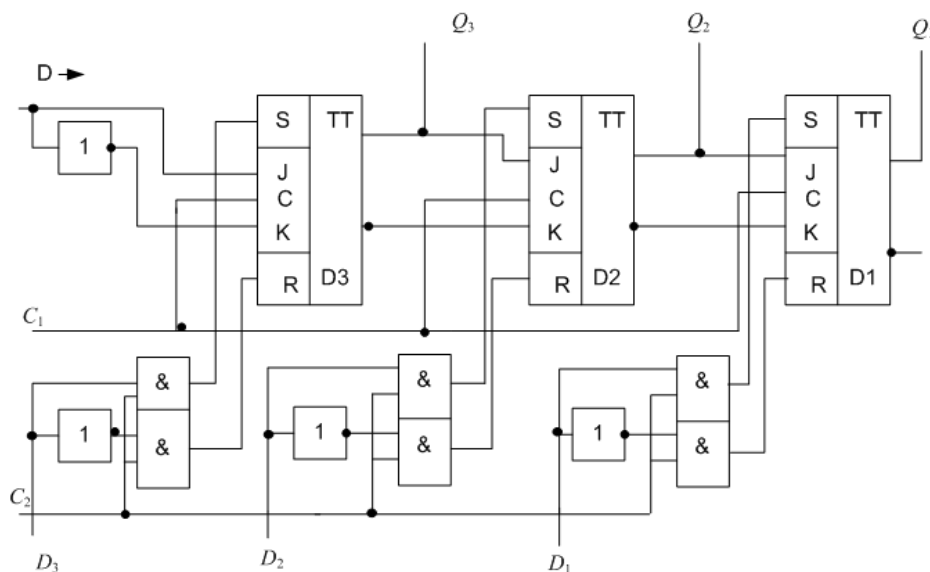


Рис. 4.21. Регістр зсуву з можливістю паралельного запису інформації

Область застосування регістрів зсуву широка. Регістр може діяти як кільцевий лічильник, коли вхід підключено до прямого виходу останнього тригера. Регістр зсуву можна перетворити на кільцевий лічильник (рис. 4.22),

якщо вихід останнього тригера підключити до входу D першого. Перед початком підрахунку в нульовий розряд лічильника (Q_0) записується початковий задаючий імпульс, логічна 1, а в інші розряди - логічний 0. На початку підрахунку кожен із вхідних T імпульсів лічильника перезаписує 1 у наступному тригері, а кількість вхідних імпульсів визначається на основі номера виходу, на якому знаходиться 1. Передостанній ($N-1$) імпульс встановить останній тригер у стан одиниці, а N-й імпульс перенесе цей стан на вихід нульового тригера, і підрахунок почнеться знову. Таким чином, ви можете побудувати кільцевий лічильник з будь-яким коефіцієнтом перетворення (будь-якою базою розрахунку), лише змінюючи кількість тригерів у ланцюжку.

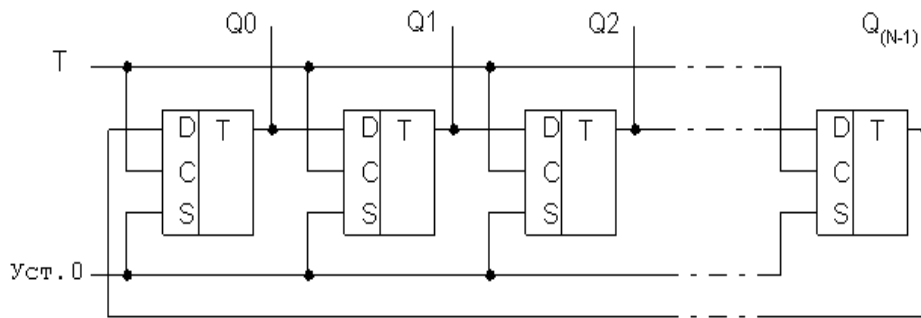


Рис. 4.22. Кільцевий лічильник у регістрі зсуву

Недоліком такого лічильника є велика кількість тригерів, необхідних для його побудови. Лічильники, побудовані на T-тригерах, економічніші і тому більш поширені. Після кожного тактового імпульсу T сигнал на вході D змінюється на протилежний, тому частота вихідних імпульсів становить половину частоти вхідних імпульсів. Зібравши послідовний ланцюжок з n тригерів підрахунку (з'єднавши вихід попереднього тригера з входом C наступного), ми отримаємо частоту $f_{вих} = f_{вх} / 2^n$.

При цьому кожен вхідний імпульс змінює числовий код на виході лічильника на 1 в діапазоні від 0 до $N = 2^n - 1$.

4.4 Лічильники

4.4.1. Призначення, класифікація та характеристика лічильників

Лічильник призначений для підрахунку кількості одиниць інформації (облік імпульси). Прийоми одиниць інформації складається з \mathbb{N} вхідні ефекти лічильник падіння напруга 0-1 при прямому вході або 1-0 при зворотному вході . Тому що один пульс містить Це і інші відпусти це його і ідентифікувати с одиниця не будемо переоцінювати тип інформації будь-які .

Лічильник має *саміт* стабільний держави , усі с чий повторюється після розрахунок *саміт* бухгалтерія імпульси . Інакше кажучи рахунок імпульси здійснюється з коефіцієнтом перетворення (модулем). K_n .

Відповідно до методу кодування числові інформації виділити лічильники

- з позицією (одинарна , двійкова , десяткова тощо)

- непозиційне кодування (наприклад, у кодах Грея) .

У лічильниках з позиційним кодуванням числове вираження поточного стану задається формулою

$$A = \sum_{i=0}^{n-1} Q_i M_i$$

Де n - кількість виділення ; Питання логічне значення розряд ($Q_i=0,1$);

M_i - вага i - т розрядка

\mathbb{N} лічильники с непозиційні кодування не мають ніяких розрядів постійний ваги і числові вираз країна пропонується кожен встановити значення Q_i .

Основа структури двійкова розміщені лічильники n бухгалтерія тригери Для всіх з них враховано \mathbb{N} відповідність поодинці деякі Ваги с набір :

$$2^0, 2^1, \dots, 2^{n-1} .$$

Кількість імпульсів, що прибув представляє себе \mathbb{N} у вигляді суми:

$$A = Q_{n-1} 2^{n-1} + Q_{n-2} 2^{n-2} + \dots + Q_1 2^1 + Q_0 2^0$$

Максимум фактор розрахунок двійковий лічильник $K_n = 2^n$.

Лічильники розділені за іншими класифікаціями знаки За домовленістю розрізняти , підсумовуючи , забрати І оборотний ; відповідно до методу запуск - асинхронний І синхронний ; відповідно до методу організацій передача - від послідовно , від кінця до кінця І паралельний перекази

Основні параметри лічильників вони там розчинний здатність , час заклад І місткість .

Розкладається майстерність - Це мінімум період перехід бухгалтерія імпульси *Стаття* T , в якій зберігається ефективність будь-які . Зворотний розмір $F_{до} = 1 / \text{Мистецтво } T$ характеризує максимальна частота підрахунку .

час установка $T_{вкт}$ є інтервал між початком адміністрування та вступом бухгалтерія імпульс і в кінці найдовшого тимчасовий процес IN Параметри лічильника *Мистецтво* T І $T_{вкт}$ характеризувати швидка дія будь-які .

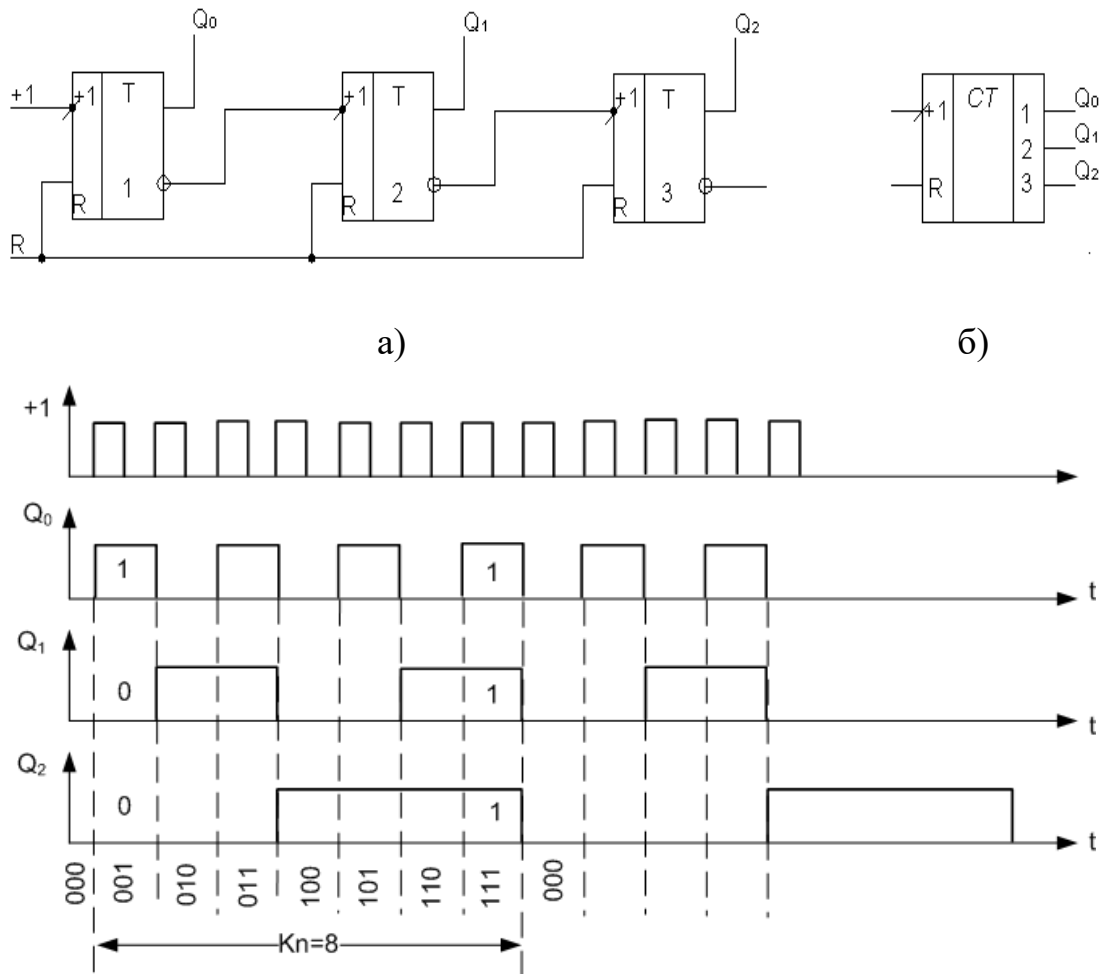
Ємність лічильник визначається максимум за номером імпульси , які можуть пройти номери реєстру місткість дорівнює фактор розрахунок Kp .

4.4.2. Лічильники з послідовним переносом

Підведення підсумків лічильники . Підведення підсумків лічильник повинен функціонувати так , щоб після його отримання Вихід один імпульс записані IN його кількість збільшилася на одиницю . Принцип побудови резюме лічильник заснований на прикладному правилі вниз двійковий чисел одиниць . Відповідно до до цього правила, наприклад , трибітний лічильник повинен послідовно прийняти дає 000, 001, 010, 011,... ,111.

В одному це тригер менше писати перемикачі всіма бухгалтерія імпульсний , тобто вхід лічильника служить Вихід Це Стан тригера другий І третій тригери змінюються всіма другий І четвертий імпульс . Це надається послідовний підключення Тригери змін значення Y t писати відбувається тоді коли вниз додаток регулярні всі одиниці попередні розрядка був одиниць . хто я тригери мати прямий бухгалтерія вхід , то вони з'єднані вниз взаємний Вихід попередній trigger , я не власник взаємний введіть , потім підключіться вниз

прямий Вихід УВІМКНЕНО рис. 4.23 подано трирозрядну схему резюме лічильник с тимчасовий діаграми Робота I умовний малюнок . Лічильник можливо візьміть 8 різних стану , які повторюються за кожні 8 входних імпульси ($K_n = 8$). З найбільшій частота перемикачі тригер менше різні серії . роздільна здатність лічильник визначається іноді затримки перемикання тригер $T_{st} = T_{tan}$. Це положення відноситься до всіх типів двійковий лічильники .



IN)

Рис. 4.23. Діаграма (а), UGP (б) і часові діаграми (в) трирозрядного послідовного лічильника передачі

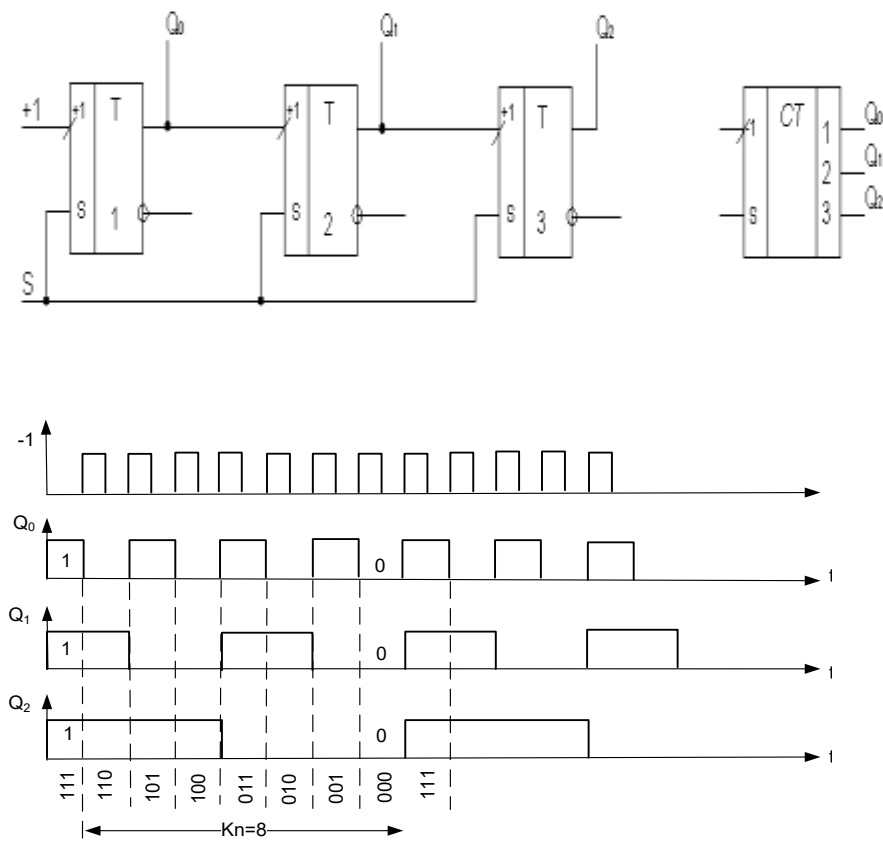
поточний лічильник називається узгодженим чисельником рухатися тому що перемикання тригер тощо писати відбувається IN в результаті послідовний перемикання все попередній тригери менше розрядка , тобто інформації

поширюється по ланцюжку тригери послідовно . час установка лічильник

$$: T_{vst} = pT_{tg} = pT_{st}.$$

При необхідності щоб видалити інформації після все прибуття пульс , період перехід повинен Бути $T > pT_{tg}$. Погіршення швидка робота с зростання цифр є основним недоліком лічильників с послідовний руха- тися

Віднімальні лічилки . При подачі лічильника введення якого забирає один бухгалтерія імпульс раніше записані IN його число зменшується на оди- ницю. правила будівництво цьому протидіє відняти, спираючись на правила розрахунок двійковий чисел I різні с правил будівлі Підводячи підсумок, тільки ці лічильники Со Якщо тригери мати прямий потім введіть + 1 їм по- єднувати вниз прямий Вихідний попередній тригер , Якщо тому вхід проти- лежний поєднувати До зворотного виходу (рис. 4.24). Ось ще один Вихід S дозволяє раніше встановити всі тригери IN уніформа хвороба



а)

б)

Рис. 4.24. Схема і УГП (а) і часові діаграми (б) трирозрядного лічильника, Що віднімає з послідовним переносом

Зауважимо це поточний лічильник Ви можете розглянути як підсумовувальний і підсумовувальний (рис. 4.11) подібний до , який віднімає при інвертуванні вихідні дні Q сигнали Q_i . Або знімний інформації с листя \overline{Q}_i .

Реверсивні лічильники . Вони працюють як в режимі додавання, так і в режимі віднімання. Перемикання режимів здійснюється перемиканням облікових входів усіх тригерів (крім тригера молодшого розряду) на інверсні або прямі виходи попередніх тригерів (рис. 4.25). Перемикання режимів здійснюється сигналом дозволу E. При E=1 прямий вхід i-го тригера перемикається на прямий вихід (i-1)-го тригера, що відповідає режиму віднімання. При E = 0 вхід підключено вниз взаємний вихід , і лічильник стає *Підводячи підсумок*

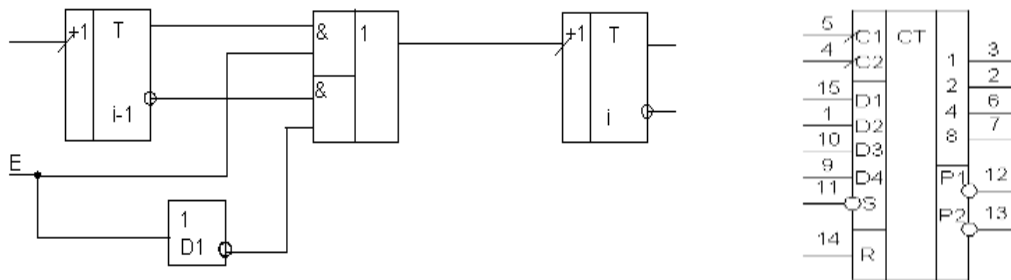


Рис. 4.25. Міжтригерні з'єднання Рис . 4.26. Реверсивний лічильник

Досягається універсальність реверсивного лічильника за ціною запис
Додатково $n - 1$ логічний елементів і погіршення швидкість :

$$T_{vst} = nT_{tan} + (n-1) t_{zt} . \text{ одружуватися}$$

Де $t_{zt, sr}$ середній час затримки перемикання LE I-АБО .

Наприклад, мікросхеми K555IE6 і K555IE7 є реверсивними лічильниками з попереднім записом. Перший з них двійково-десятковий, другий чотирозрядний (рис. 4.14). Вони встановлюються на 0, коли рівень на вході R високий . У лічильнику можна ввести число, двійковий код якого подається на входи $D_1 - D_4$ (у K555IE6 від 0 до 9, у K555IE7 від 0 до 15). . Для цього низький рівень повинен бути поданий на вхід S (високий рівень на входах C1

і $C2$, низький рівень на вході R). Підрахунок почнеться із записаного числа через імпульси низького рівня, подані на вхід $C1$ (у режимі додавання) або $C2$ (у режимі віднімання). інформація на виході змінюється по фронту імпульсу лічильника. При цьому на вході другого лічильника і вході S повинен бути високий рівень, на вході R - *низький*, а стан входів D має бути нейтральним. При цьому на кожному десятому (шістнадцятому) імпульсі на вході $C1$ на виході $P1$ з'являється *вихідний імпульс*, який повторює його і може бути поданий на вхід наступного лічильника. У режимі віднімання з кожним імпульсом на вході $C2$, який змушує лічильник перейти в стан 9 (15), *вихідний імпульс з'являється на виході $P2$* .

Часова діаграма лічильника наведена на рис. 4.27. Число 6 було введено на графік у режимі паралельного запису ($S=0$) (*високий рівень на входах D_2 і D_3*).

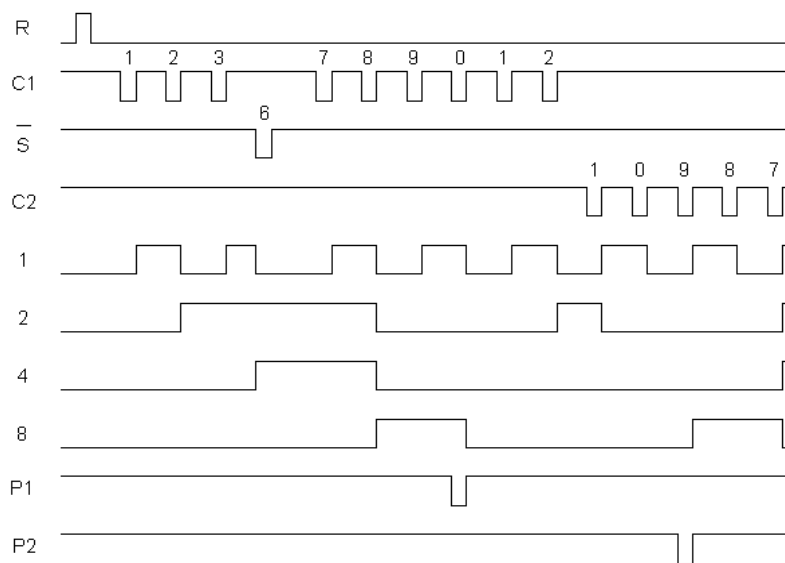
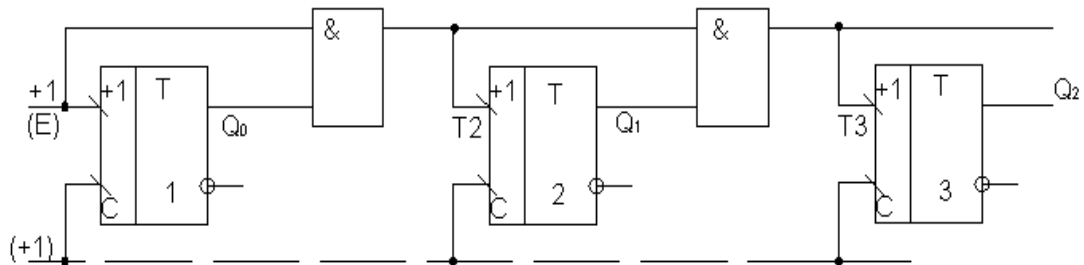


Рис. 4.27. Часова діаграма метрів К555ИЕ6, 555ИЕ7

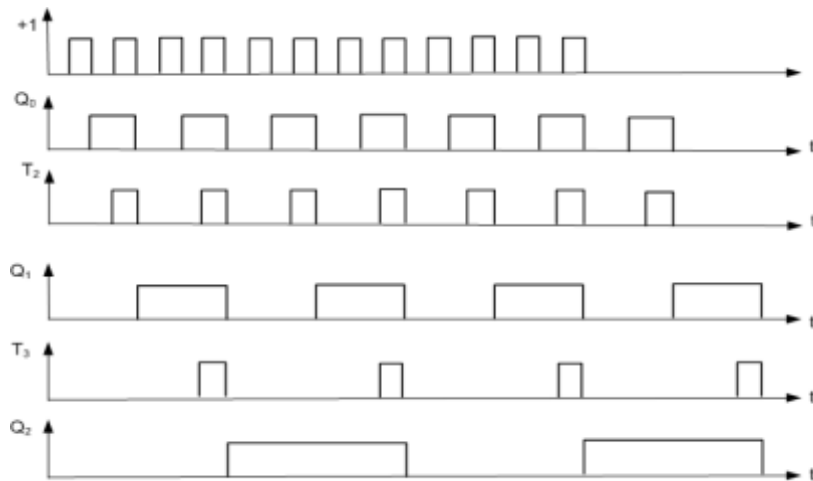
4.4.3. Лічильники з прохідною передачею

як відомо, що i -й тригер писати перемикає, якщо ($i-1$) запускає вниз момент дохід від цього Вихід регулярний бухгалтерія сигнал T_{i-1} був IN неодружений стан, тобто має місце $Q_{i-1} m_{i-1} = 1$.

тому с намір прискорення рухатися можливий вхідний сигнал T_{i-1} Я пропусу вхід я - т тригер за допомогти елемент I проходячи повз $(i-1)$ тригер (рис. 4.28, а). Тригери вони можуть Бути асинхронний I синхронний . Відповідно до до цього лічильники є асинхронними Або синхронний .



а)



б)

Рис. 4.28. Схема (а) і часові діаграми (б) трирозрядного лічильника с за рухатися

Від схеми асинхронного лічильника , зображені на рис. 4.28, б, зрозуміло, що дякую діаграми рухатися до під'їздів другий , третій тощо. тригери надсилаються в унісон кожен другий , четвертий і т.д прибуття імпульси . Відразу перехід останні вхідні імпульси тригер затримується на деякий час перенесення : $T_{per} = (n-1) t_{zt. одружуватися}$

час заклад лічильник : $T_{vst} = T_{tg} + (n-1) t_{zt. cp}$

Беручи на себе Будь ласка, зверніть на це увагу час затримки перехід сигнал за LE I менше , ніж Через тригер , перемога IN швидка дія IN лічильники с за передача в порівнянні с лічильники с послідовний рухатися очевидно , але є ще щось сума обладнання .

На синхронному лічильнику с за перенести запис на рахунок є об'єднані Вихід синхронізація з усіма тригери , дякую чому Вони перемикаються одразу Інформативний Вихід перший тригер стає введенням дозволу режим рахунок E . Коли $E = 0$, усі сигнали $T = 0$ і лічильник залишається IN режим зберігання Коли $E = 1$, встановлюється режим облікового запису .

Час перемикання та час встановлення такі ж, як і для синхронного лічильника. Але оскільки в асинхронному лічильнику відбувається безперервне (із затримкою t_{cp}) перемикання тригерів протягом усього часу T_{set} , то для зняття інформації потрібен додатковий час, тобто подовження періоду проходження імпульсів вхідного сигналу.

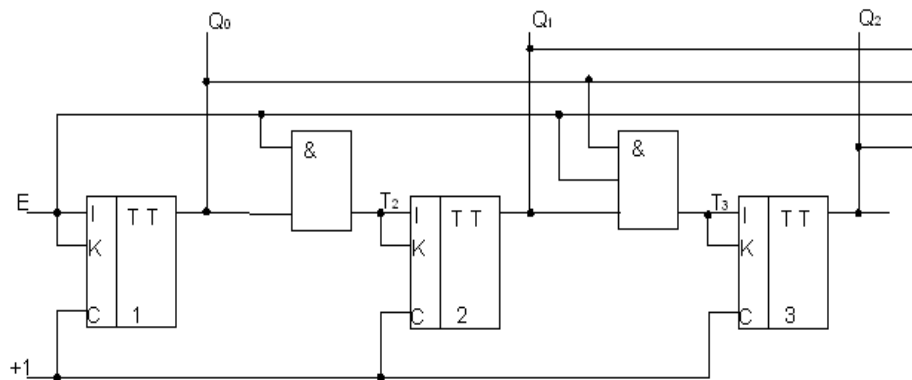
У синхронному лічильнику тригери перемикаються одночасно (за лічильним імпульсом) і лише потім передаються в ланцюжки логічних елементів, тому для видалення інформації не потрібен додатковий час - час T можна використовувати для . Швидкість синхронного лічильника вище, ніж асинхронного.

4.4.4. Лічильники з паралельним перенесенням

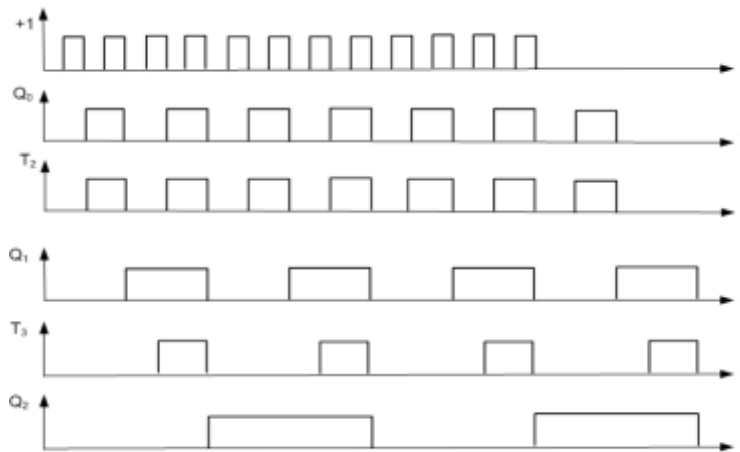
Отримувати паралельний рухатися обставина , що перемикання тригер $Я t$ писати відбувається коли до збору регулярний вхідний імпульс протидіє всім тригерам менше розрядка Якщо IN неодружений стан , тобто: $Q_i .. Q_{i-1} = 1$. Отже, сигнал є інформативним Вихід все тригер Ви можете виробляти за сполучники вихідні дні сигнали попередні тригери Рис . 4.29 наведені принципові та часові графіки лічильника с паралельний рухатися синхронний тип , який буде використовуватися двоступеневий Тригери JK .

Час встановлення лічильника $T_{vst} = T_{tg} + t_{zt. CP T}$. е. практично визначається його роздільною здатністю. Володіючи найвищою швидкодією, він має один істотний недолік - нерівномірність схем передачі. Зі збільшенням ваги розряду

збільшується кількість записів LE. Тому при високих бітових швидкостях вони вдаються до комбінованої передачі - паралельно всередині тригерних груп і послідовно або поперек між групами.



а)



б)

тризначний лічильник с паралельне перенесення

Якщо лічильник побудовано на основі тригерів JK, які мають декілька інформаційних входів, з'єднаних роз'ємом, то його конструкція спрощується – виключаються зовнішні елементи I (рис. 4.30).

За винятком крім того, відмова с запис дозвіл E дозволяє збільшення швидкість передачі лічильник на один без збільшення чисел інформативний входи Z тригери намір збільшення швидкість передачі лічильники Ви можете поєднувати послідовно . Плюс буде використовуватися сигнал переносу $CR' > 15$ як

для наступного лічильник вони там бухгалтерський облік Виробляється по до-
 помоги огорожений прибуття імпульс дешифратора старшого (п'ятнадця-
 того) стану . Позначення «>15» означає, що сигнал перенесення з'являється в
 інтервалі часу від закінчення 15-го лічильного імпульсу до приходу наступ-
 ного - нульового.

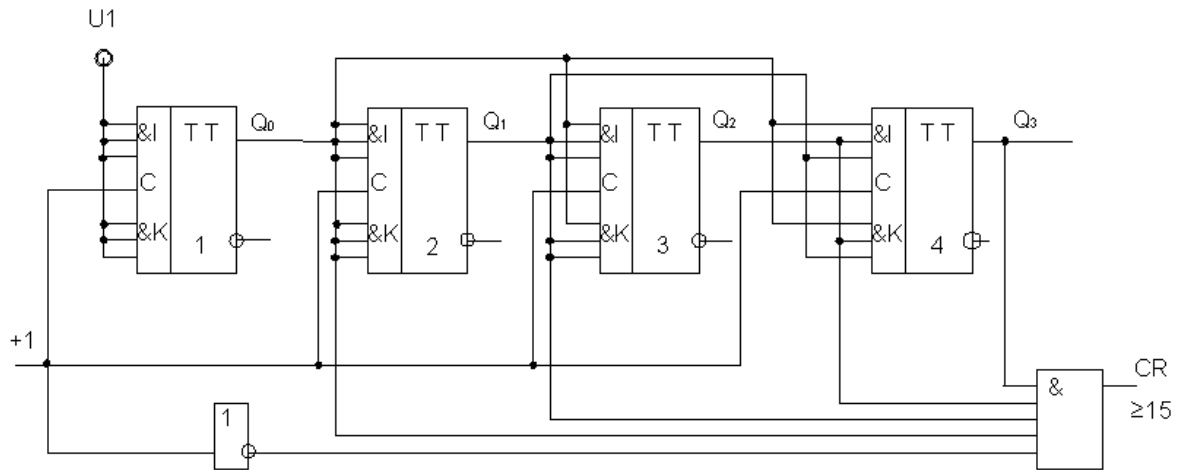


Рис. 4.30. Чотирирозрядний лічильник з передавальним пристроєм форму-
 вання сигналу

Лічильники с паралельний передача , завершено IN у формі індивідуаль-
 ний ЦЕ, звичайно мати додатковий входи для попереднього запису інформації

4.4.5. Лічильники з коефіцієнтом перерахунку, відмінним від 2^p

Великий лічильники поширені і дільники з $саміт \neq 2^n$. Так, ти
 цифровий індикатор пристроїв домінувати двійково-десятковий лічильники
 з $K_n = 10$. Принцип побудови лічильники с $саміт \neq 2n$ зменшується вниз далі вони
 беруть такий числові n запускає , доки не буде виконано хвороба

$$2^{n-1} < K_n < 2^n .$$

Далі схематично за виключити $2^p - K_p$ надлишковий Вони часто опиня-
 ються виключеними старші держави , рідше – молодші Або непрямий Це ро-
 биться за допомогою спеціального декодера для визначення того, що його

власний вихідний сигнал є примусовим встановлює лічильник ІN вихідний з роботи стан , або за допомогою зворотний знайомих між тригери .

На рис. 4.31 подано приклад двійково-десятковий лічильник с виключаючи старших , непотрібних працівників констатує Вступний країна ІN його нуль : $A_0 \{0000\}$. Після прибуття прибуття імпульси результат заходить як ІN звичайний двійковий будь- який Як Просто встановлено стан $A_{10} \{1010\}$, на виході елемента І дешифратора виробляється сигнал $Y = Q_1 Q_3 = 1$ і лічильник сила перекладається ІN попередній країна 1_0

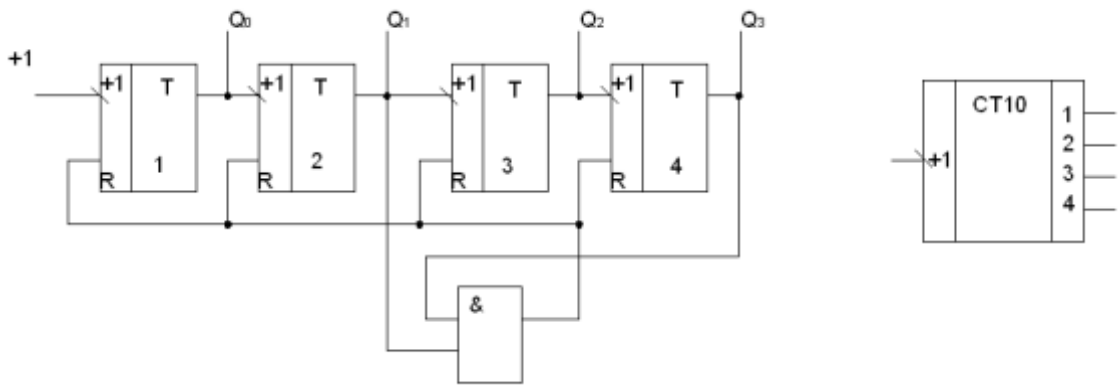
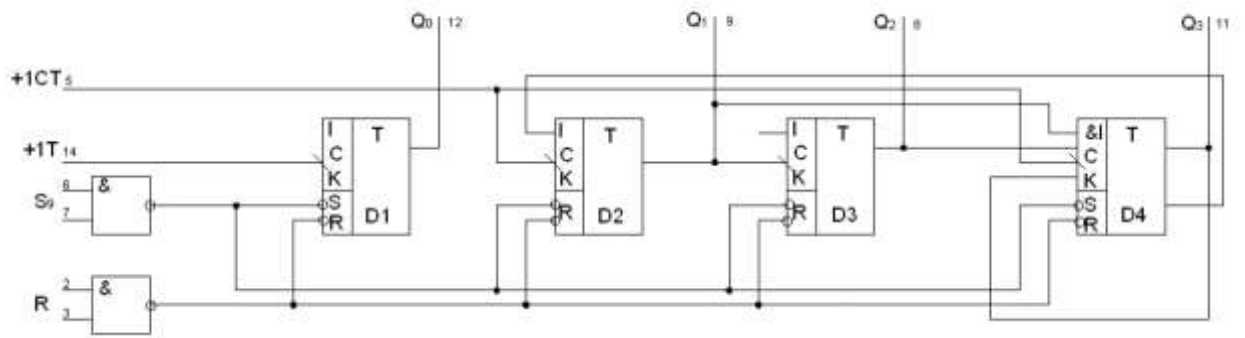


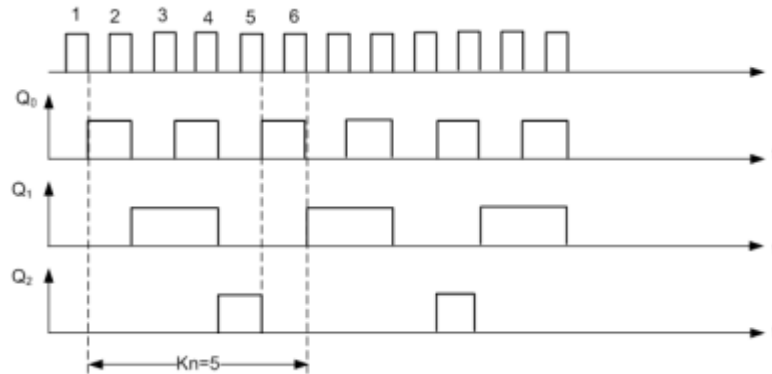
Рис. 4.31. Двійковий - десятковий лічильник с виключаючи старших , непотрібних працівників став

Наприклад використовуючи зворотний з'єднання, які слід виключити надлишковий став можливо служить ІС 133 ІЕ 2 (рис. 4.32, а). вона містить T - тригер D_1 і подвійний - п'ятирічний лічильник на тригерах D_2, D_3, D_4 . дякую навпаки підключення с Вихід $\overline{Q_3}$ тригер D_4 для запуску входу $J D_2$ а також зворотний підключення ІN я тригери D_4 (з Вихід $\overline{Q_3}$ УВІМКНЕНО вхід К). відповідно блокування дії п'ятий бухгалтерія імпульс на курку D_2 і установка Тригер D_4 п'ятий імпульс ІN недійсний стан (рис. 4.32, б). Так після п'ятий імпульс виходить, що $A_0 \{000\}$.

Якщо вихід Q_0 тригер D_1 поєднувати з $+1$ ST входом і розрахунком імпульси застосувати до входу $+1T$, потім до лічильника стає двійково-десятковий с $кн = 10$.



а)



б)

Рис. 4.32 Принципова (а) і часова діаграми (б) чотирирозрядного лічильника з $K_n=5$

4.5. Дільники

Дільники - це лічильники, які мають один вихід на якому після цього з'являється пульсовий сигнал кожен *саміт* бухгалтерія імпульси. Досить часто передбачається можливість Зміна фактор розподіл *саміт* за допомогою спеціальні менеджер код. ІN правил роздільники Ви можете будувати і на основі двійковий метрів шляхом підключення вниз вихід деякого декодера один стан В більшість практичний питання інтегральний перетворений пристроїв зроблено комбіновані - лічильники-дільники. З це намір лічильник доповнюється старішим станом дешифратора, вихідний сигнал якого можливо Бути використовується як сигнал передачі під час нарощування швидкість передачі за допомогою декілька мікросхеми або Як вихідний сигнал дільника.

Принципова схема дільників багато в чому подібні принцип будівлі лічильники. Зазвичай вони є вихід, на якому за інтервал розрахунок з'являється імпульси IN фактор розподіл час менше, ніж вхід ($K_d = N_{in} / N_{out}$). Щоб виділитися ці імпульси за допомогою декодера стану.

Зміна коефіцієнта K_d в подільниках можлива подібно до $K_r \neq 2^n$ метрів, тобто шляхом усунення різної кількості надлишкових станів, але програмним шляхом - за допомогою зовнішніх сигналів управління. Цей спосіб реалізований, наприклад, в IS 564IE15. Це дозволяє отримати $K_d = 3 \dots 21327$ з одиничним кроком.

Метод, у якому коефіцієнт обчислення лічильника, що лежить в основі дільника, не змінюється, але робиться вибір декодерів, налаштованих для виділення різних станів лічильника. Варто зауважити, що якщо вхідна імпульсна послідовність є періодичною, то вихідна послідовність буде періодичною лише в тому випадку, якщо коефіцієнт ділення є цілим числом, в інших випадках вивідні імпульси будуть нерівномірно розподілені в часі. Для підвищення діапазону регулювання коеф. ділення мікросхеми з'єднані послідовно. Імпульси передачі діють як вхідні дані для наступних схем.

Наприклад, мікросхема K555IE5 (рис. 4.33) містить лічильний тригер (вхід $C1$) і дільник на вісім (вхід $C2$), утворений трьома послідовно з'єднаними тригерами. Спрацьовування відбувається після відсікання вхідного імпульсу (при переході від 1 до 0). Якщо ми з'єднаємо всі чотири тригери послідовно, ми отримаємо модульний лічильник $2^4 = 16$. Максимальна кількість, яка зберігається в лічильнику після його повного заповнення одиницями, становить $N = 2^4 - 1 = 15 = (1111)_2$. Такий лічильник працює з коефіцієнтом підрахунку K , який є кратним цілого ступеня 2, і циклічно обчислює $K = 2^n$ стійкі стани. Лічильник має входи для примусової установки на 0.

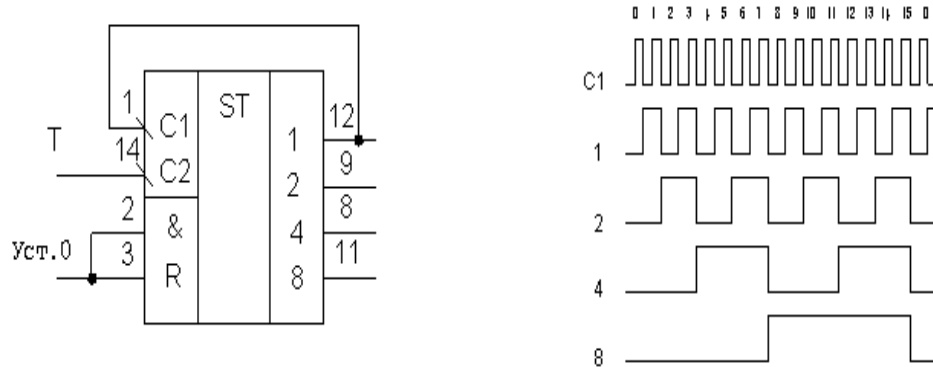


Рис. 4.33. Лічильник з коефіцієнтом перерахунку 16 і його часова діаграма

Мікросхема К555ІЕ1 (рис. 4.34) є дільником на 10. Його тригери встановлюються в 0 шляхом одночасної подачі високого рівня на входи 1 і 2 (елемент І). Лічильні імпульси подаються на вхід 8 або 9 (другий вхід повинен мати одночасно високий рівень) або на обидва входи (елементи) одночасно.

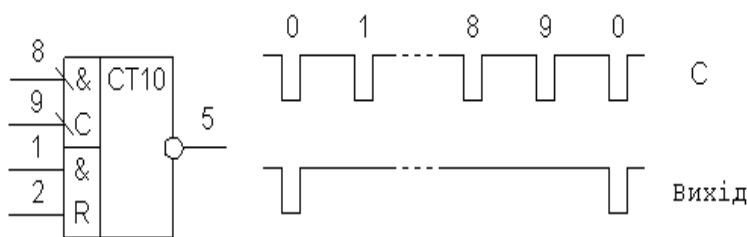


Рис. 4.34. Дільник частоти.

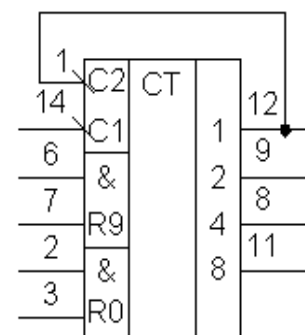


Рис. 4. 35. Двійково-десятковий чис-
лівник

Схема К555ІЕ2 (рис. 4.35) містить тригер з входом лічильника (вхід *C1*) і дільником на 5 (вхід *C2*). Коли вихід тригера лічильника підключено до входу *C2*, буде створено двійково-десятковий лічильник. Підрахунок відбувається на ділянці пульсу. Лічильник має входи, встановлені на 0 (*R0* з логічним І) і входи, встановлені на 9 (*R9* з логічним І).

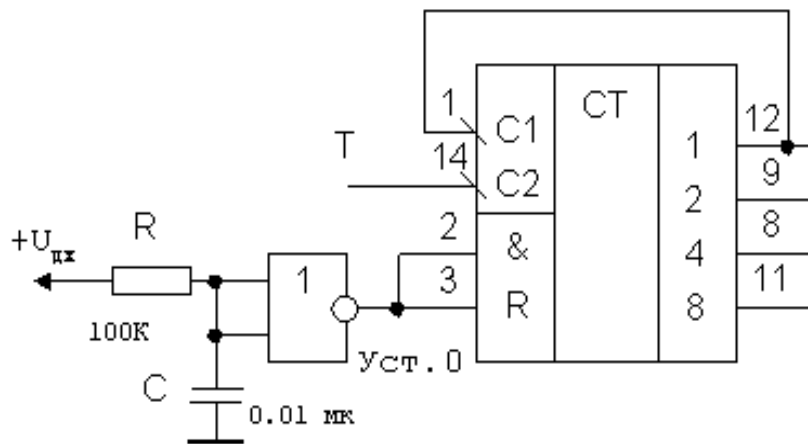


Рис. 4.36. Схема скидання лічильника на 0

Для коректної роботи цих і всіх інших лічильників, виготовлених за технологією КМОН (серії К164, К176, К564, К561), необхідно скинути їх у вихідний стан шляхом подачі імпульсу високого рівня після включення живлення (або після зниження напруги джерела живлення до 3 В) на вхід R

В іншому випадку лічильники можуть працювати з випадковим K_n , імпульс скидання при включенні живлення може надаватися автоматично, якщо введено RC -схему таймера та інвертор (рис. 4.36).

Мікросхеми К176ІЕ3 і К176ІЕ4 (рис. 4.37) являють собою лічильники модуля 6 і 10 з дешифратором, що працює на семисегментний індикатор. Корпуси у них однакові, тільки на місці виходів 2 і 6 (висновки 3 і 2) лічильника К176ІЕ3 десятковий лічильник К176ІЕ4 має виходи 4 і 10. Лічильні імпульси подаються на вхід Т. Напруга на виході може бути $C=0$ прямиий код (при $C=1$), який дозволяє підключати до лічильника індикатори із загальним катодом. У цьому випадку — меандр з частотою Гц подається на вхід С. $f > 50$ При послідовному з'єднанні лічильників сигнал знімається з виходу 6 (К176ІЕ3) або 10 (К176ІЕ4).

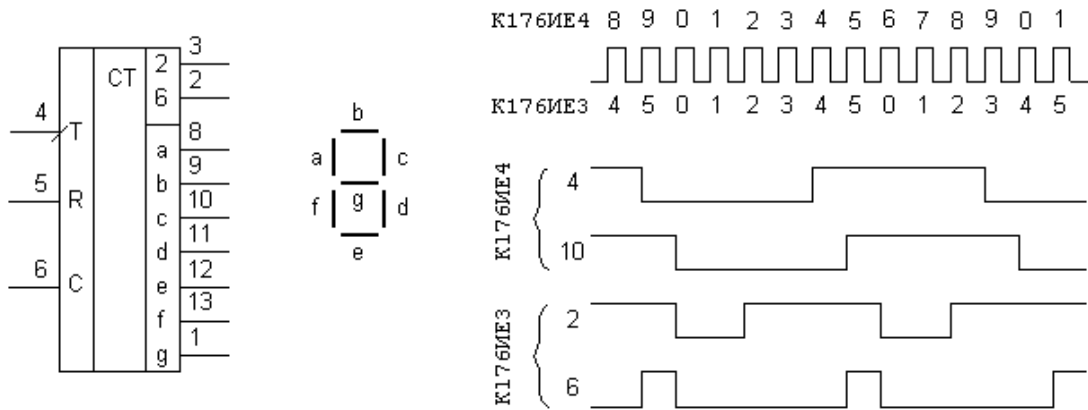


Рис. 4.37. Лічильник з дешифратором і часові діаграми його роботи

4.6. Генератори та системи формування імпульсів

На основі логічних елементів цифрових пристроїв можна побудувати різноманітні генератори імпульсів. Генератор (рис. 4.38) виробляє імпульси в широкому діапазоні частот — від одиниць герц до кількох кілогерц. Залежність частоти f (кГц) від ємності конденсатора $C1$ (пФ) виражається наближеною формулою

$$f \approx 3 \cdot 10^5 / C1.$$

Шифрування імпульсної напруги майже дорівнює 2. При зниженні напруги джерела живлення на 0,5 В частота генерованих імпульсів зменшується на 20%.

У генераторі, зображеному на рис. 4.39 тривалість імпульсів регулюється за допомогою змінного резистора $R2$ (шварованість змінюється від 1,5 до 3), а частота — за допомогою резистора $R1$. Наприклад, в $C1 = 0.1$ генераторі мкФ після видалення резистора $R2$ частоту генерованих імпульсів можна змінити від 8 до 125 кГц тільки за допомогою резистора $R1$. Для отримання іншого частотного діапазону необхідно змінити ємність конденсатора $C1$.

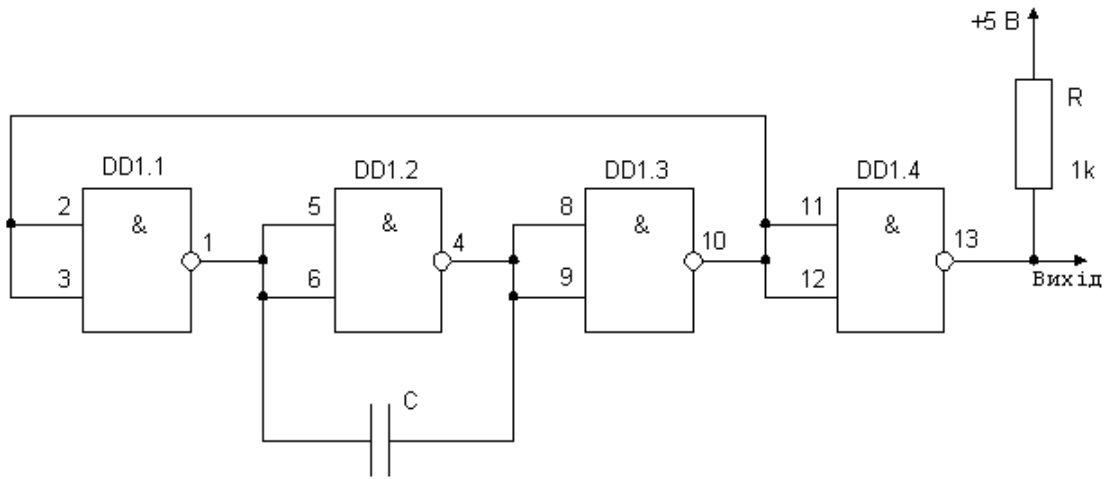


Рис. 4.38. Генератор імпульсів

На рис. На рис. 4.40 показано принципову та часову діаграми схеми формування імпульсу на фронті та зрізі вхідного сигналу. Тривалість кожного генерованого імпульсу $t_{i1} = t_{i2} = n \cdot t^{1,0} + (n+1) \cdot t^{0,1}$ однакова де n – парна кількість елементів, що беруть участь у затримці сигналу; $t^{0,1}, t^{1,0}$ – час затримки елементів при переході від 0 до 1 і від 1 до 0 відповідно.

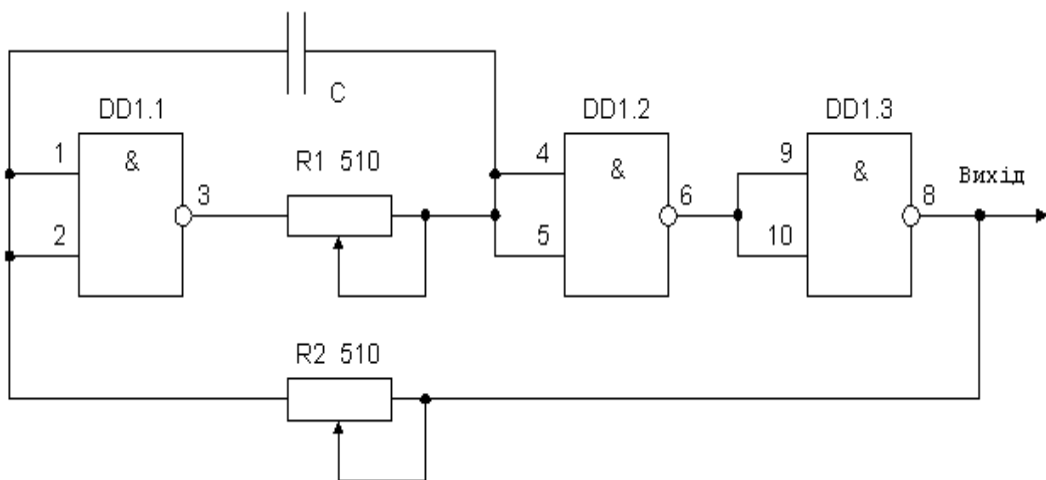


Рис. 4.39. Генератор імпульсів з регульованою тривалістю

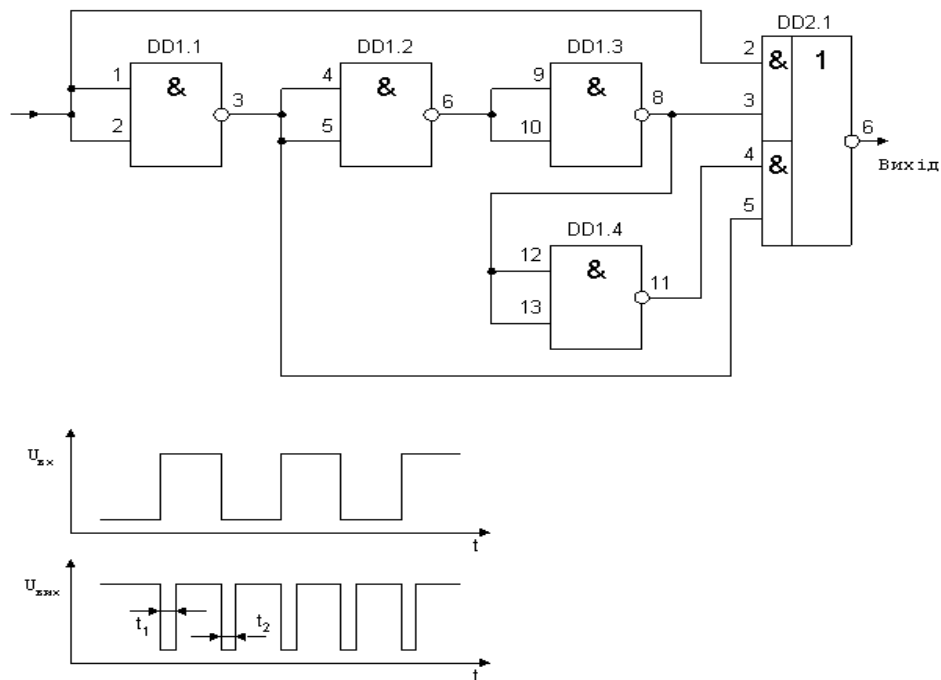


Рис. 4.40. Формування імпульсів по краях і відсікання вхідного сигналу

Принцип роботи цього генератора схожий з принципом роботи описаних раніше генераторів коротких імпульсів.

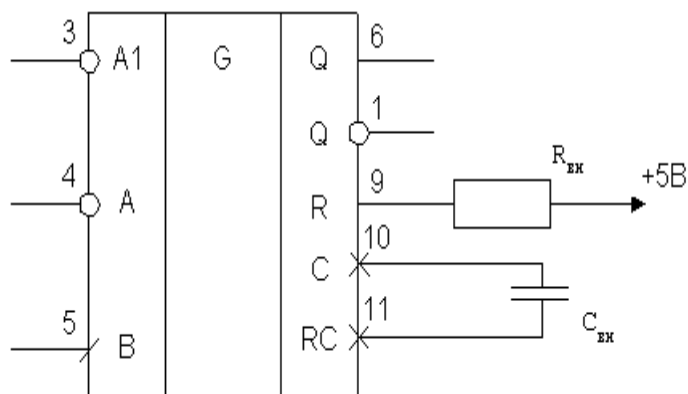


Рис. 4.41. Генератор імпульсів на мікросхемі K555АГ1

Генератори імпульсів входять до складу мікросхем серії K555. Таким чином, мікросхема K555АГ1 являє собою єдиний вібратор з трьома входами, прямим і зворотним виходом і виходами для підключення трьох зовнішніх схем управління (рис. 4.41). Один вібратор може запускатися як позитивними, так і негативними провалами у вхідних сигналах при заданій напрузі, незалежно від

тривалості вхідних імпульсів. Моновібратор перемикається негативним провалом у вхідному сигналі, що подається на один із входів А, коли на вхід В подається напруга високого рівня, або позитивним провалом у вхідному сигналі, що подається на вхід В, якщо напруга низького рівня присутній на одному з входів А або А1.

При максимальному опорі резистора $R_{\text{вн}} = 40 \text{ кОм}$ тривалість вихідного імпульсу не повинна перевищувати $0,9T$, де T - період слідування за вхідними імпульсами.

Тривалість вихідного імпульсу залежить від резистора $R_{\text{вн}} = (0-40) \text{ кОм}$ і $C_{\text{вн}} = (0...1000) \text{ мкФ}$ і визначається як: $t_n = RC \ln 2$.

Ось $R = 2k + R_{\text{вн}}$ опір внутрішнього резистора. $2k$

Серія К555 також включає чіпсет К555АGZ. Містить два окремих вібратора в одному корпусі. Можливості підключення зовнішніх компонентів ГРМ і тимчасова діаграма роботи одного вібратора наведені на рис. 4.42. Моновібратор також запускається або негативною різницею вхідного сигналу на вході А з високим рівнем на входах В і R, або позитивною різницею напруг на вході В з низьким рівнем на вході А і високим рівнем на вході R. тривалість імпульсу t_{i1} визначається постійним часом ланцюга синхронізації, але її можна скоротити шляхом подачі низької напруги на вхід R в $t_{i2} < t_{i1}$.

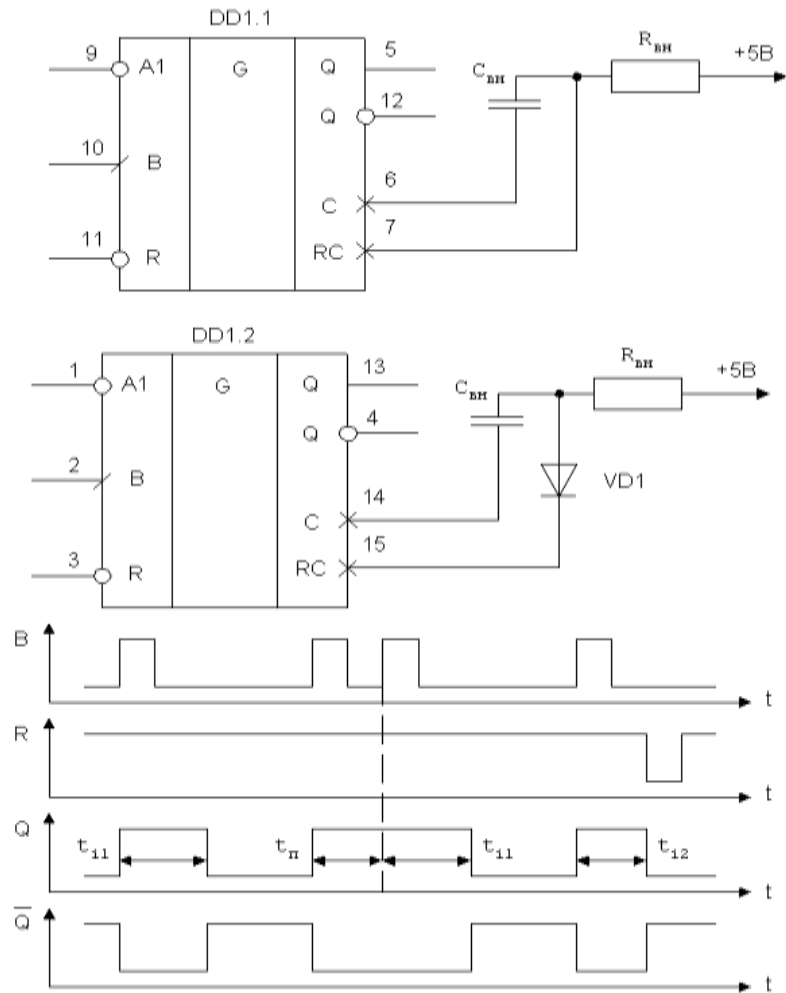


Рис. 4.42. Варіанти схеми формування імпульсу на мікросхемі К555АГЗ

ГЛАВА 5. ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ

5.1. Види запам'ятовуючих пристроїв

Регістри можна використовувати для зберігання невеликих масивів кодів слів. Але навіть якщо необхідно зберігати десятки слів, використання регістрів призводить до невиправдано високих витрат на обладнання. Для зберігання великої кількості слів будуються *запам'ятовуючі пристрої (ЗП) з використанням спеціальних мікросхем, кожна з яких може зберігати великий обсяг інформації*. Різноманітність напівпровідникових ЗУ зумовило їх класифікацію за низкою ознак: функціональним призначенням, способом зберігання інформації, способом доступу (адресації) до осередків пам'яті, технологією виготовлення та ін.

Залежно від виконуваних функцій розрізняють наступні види ЗП:

- оперативна пам'ять (RAM);
- пристрій постійної пам'яті (PLD);
- перепрограмований постійний запам'ятовуючий пристрій (PSSD).

Оперативні ЗП забезпечують не тільки зберігання та видачу інформації, а й її швидке (оперативне) відновлення. Вони використовуються в комп'ютерах або інших комп'ютерах для тимчасового зберігання програм, вихідних даних і результатів обчислень. Інформація, що зберігається в оперативній пам'яті, може зберігатися лише тоді, коли до неї підключено джерело живлення. При вимкненні живлення інформація буде втрачена.

Завдяки цьому оперативна пам'ять забезпечує три режими роботи:

- спосіб зберігання за відсутності посилання на ЗП;
- режим читання письмового слова;
- спосіб написання нових слів.

ПЗП призначений для зберігання певної інформації, яка не знищується при відключенні джерела живлення.

ПЗП передбачає два режими роботи:

- режим зберігання;
- режим швидкісного читання.

Енергонезалежна пам'ять може зчитувати лише раніше збережену інформацію. Інформація в них записується шляхом зміни фізичних властивостей кристалічних ділянок, зайнятих осередком пам'яті, або шляхом зміни провідності електричних кіл. Тому збій живлення не впливає на вміст записаної інформації. ПЗУ використовуються для зберігання довгострокової інформації та підпрограм в комп'ютерах, програм для спеціальних комп'ютерів, що керують і індикують системи, а також різних функціональних пристроїв: дешифраторів, суматорів, перетворювачів коду і т. д. Наприклад, якщо група комірок пам'яті яких адреса задається двійковим числом, буде збережено двійково-десятковий еквівалент, ПЗУ буде виконувати роль перетворювача: після отримання двійкового числа на входах адреси ПЗУ, на його виході в десятковому коді з'явиться це ж число у двійковому форматі.

За способом доступу до комірок пам'яті всі ОЗУ діляться на адресні та асоціативні.

У адресних комірках пам'яті доступ до комірок пам'яті здійснюється за їх фізичними координатами, які визначаються зовнішнім двійковим кодом - адресою. Адресна оперативна пам'ять доступна з довільним доступом, який допускає будь-який порядок передачі адрес, і з послідовним доступом, в якому комірки пам'яті можна вибирати тільки в порядку зростання або спадання адрес. Останні функціонально схожі на регістри зсуву.

В асоціативній пам'яті інформація шукається відповідно до символів, які містяться в збереженій інформації.

За способом зберігання інформації оперативна пам'ять буває статичною і динамічною. Статична оперативна пам'ять покладається на комірки пам'яті, які після ввімкнення можуть залишатися в одному з двох стабільних станів протягом будь-якого часу. Прикладом такої комірки є тригер.]

Конденсатори служать елементами пам'яті в динамічній пам'яті. Коли заряд вводиться в конденсатор, що еквівалентно запису біта даних, він поступово мимовільно розподіляється струмами витоку. Тому така пам'ять потребує періодичного відновлення (регенерації) свого вмісту, що вимагає спеціальних зо-

внішніх регенераторів і ускладнює організацію доступу. Однак завдяки невеликим розмірам і високій ефективності комірок пам'яті динамічні ОЗП вигідно відрізняються від статичних ОЗП великою (десятки тисяч біт і більше) ємністю пам'яті. У сучасній пам'яті регенерація пов'язана з адресацією комірки або групи комірок пам'яті.

Також розробляються ОЗУ з динамічними комірками і вбудованою системою регенерації - квазістатичні ОЗП. Зовнішні керуючі сигнали в цьому випадку такі ж, як і в повністю статичних ОЗП.

Запам'ятовуючі пристрої характеризуються більшим переліком параметрів. Найважливішими з них є *інформаційна ємність*, час дискретизації та *час запису*.

Інформаційна ємність - це максимальна кількість біт даних, які можуть зберігатися в пам'яті. Визначається кількістю комірок пам'яті в мікросхемі. Інформаційна ємність часто виражається як добуток: «кількість слів» на «розмір слова в бітах», наприклад, 64x4, 1024x1.

Час дискретизації - інтервал часу між моментом подачі сигналу дискретизації і появою інформації на виході пам'яті за умови використання всіх інших необхідних сигналів.

Час реєстрації - мінімальний час збіжності керуючих сигналів на входах мікросхеми, що забезпечує запис даних.

ПЗП виконує роль ПЗП у процесі функціонування цифрового пристрою. Вони відрізняються від ПЗП тим, що дозволяють оновлювати записану інформацію, тобто забезпечують режим запису. Однак, на відміну від оперативної пам'яті, збереження інформації вимагає від'єднання карти пам'яті від цифрового пристрою і здійснюється за допомогою спеціальних записуючих пристроїв (програмувальників). Перепрограмовані ПЗУ коштують дорожче ПЗУ і використовуються в процесі налагодження пристрою, що розробляється, після чого їх можна замінити більш дешевими ПЗУ.

5.2. Основні параметри запам'ятовуючих пристроїв

ZP містить певну кількість N комірок, кожна з яких може зберігати слово з певною кількістю цифр n . Комірки нумеруються послідовно двійковими числами. Номер комірки називається адресою. Якщо для представлення адрес використовуються комбінації n -розрядного двійкового коду, то кількість комірок у ZP може бути $N = 2^n$.

Обсяг інформації, який можна зберегти на картці пам'яті, визначається її *ємністю*. Ємність можна виразити як кількість N комірок із зазначенням розміру n слів, що зберігаються в них, у формі $N \cdot n$, або її можна визначити як добуток $M = N \times n$ бітів. Часто швидкість передачі клітинки вибирається як кратне байту (1 байт дорівнює 8 бітам). Тоді ємність зручно вказувати в байтах. Значення великих ємностей виражаються в одиницях $K = 2^{10} = 1024$. Наприклад, $M = 64 \text{ Кбайт}$ означає ємність, що дорівнює $M = 64 \times 1024$ байту, $M = 64 \times 1024 \times 8$ біту.

Швидкість ЗП характеризується двома параметрами:

- час дискретизації t_B , тобто інтервал часу між моментом появи сигналу дискретизації під час запису та моментом дискретизації;
- момент, коли дозволено інший доступ до пам'яті.

ЗП складаються з набору ідентичних мікросхем ЗП з певним з'єднанням. Крім часу відгуку і ємності, кожна мікросхема ЗП характеризується споживаною потужністю, набором напруг живлення і типом корпусу (кількістю клем). Мікросхеми ПЗП додатково характеризуються часом зберігання збереженої в них інформації (після якого інформація, що зберігається в них, може вільно змінюватися), і допустимою кількістю циклів перезапису (після якої мікросхема вважається непридатною до використання).

Перелік та основні ознаки різних типів ПЗП вітчизняного виробництва наведені в таблиці 5.1 - ПЗП, таблиці 5.2 - ПЗП з одноразовим електропрограмуванням, таблиці 5.3 - ПЗП.

Таблиця 5.1 – Основні характеристики пристроїв основної пам'яті

Чіпсет	Інформаційна ємність, біт (організація, слово × цифра)	Час вибі- рки, нс	Напруга живлення, В	Споживана потуж- ність, мВт	Кількість виходів
K155PY5	256 (256×1)	60	5	700	16
KP188PY2A	256 (256×1)	500	5	Pst = 0,05 , Pdyn = 10	16
K500PY410	256 (256×1)	25	-5.2	750	16
K500PY415	1024 (1024×1)	30	-5.2	730	16
KP565PY2A	1024 (1024×1)	450	5	300	16
KP537PY2A	4096 (4096×1)	300	5	50	18
KP541PY1A	4096 (4096×1)	120	5	450	18
KP565PY1A	4096 (4096×1)	200	12; 5; -5	3; 0,25; 0,125	22
KP541PY31	8192 (8192×1)	150	5	550	20
KP541PY3	16384 (16384×1)	150	5	550	20
P581RU4	16384 (16384×1)	200	12; 5; -5	500; 0,05; 2	22

Таблиця 5.2 – Основні характеристики пристроїв енергонезалежної пам'яті

Чіпсет	Інформаційна ємність, біт (організація, слово × цифра)	Час вибірки, нс	Напруга живлення, В	Споживана потужність, мВт	Кількість виходів
K155PE3	256 (256×1)	50	5	550	16
KP500PE149	1024 (256×4)	35	-5.2	730	16
K541PT1	1024 (256×4)	80	5	400	16
K556PT4	1024 (256×4)	70	5	650	16
KP556PT5	4096 (512×8)	70	5	950	24
KP565PT1	4096 (1024×4)	300	-12; 5; -5	Стаття П = 3; 10; 0,5 Rdyn = 130; 1; 95	22

Таблиця 5.3 – Основні характеристики перепрограмованих пристроїв енергонезалежної пам'яті

мікроскоп	Інформаційна ємність, біт (організація, біт слова)	Час вибірки, нс	Час зберігання, год	Кількість циклів перезапису	Напруга живлення, IN	Споживана потужність, мВт	Кількість виходів
<i>ППЗП з багаторазовим електричним перепрограмуванням</i>							
KR505RR4A	512 (256×2)	1200	3000	10 ⁴	-9; 5	350; 200	24
K505RR4	1024 (512×2)	1200	3000	10 ⁴	-9; 5	350; 200	24
KP558PP11	1024 (256×4)	5000	3000	10 ⁴	-12; 5	120; 50	24

KP558PP1	1024 (256x4)	5000	3000	10^4	-12; 5	120; 50	24
<i>ППЗП з ультрафіолетовим стиранням і електрозаписом</i>							
K573RF1	8192 (1024x8)	450	15000	10	12; -5; 5	850; 225; 75	24
K573PФ11	4096 (512x8)	450	15000	10	12; -5; 5	850; 225; 75	24
K573RF13	4096 (1024x4)	450	15000	10	12; -5; 5	850; 225; 75	24
K573RF2	16384 (2048x8)	900	10000	10	5	225	24
K573RF21	8192 (1024x8)	900	10000	10	5	225	24
K573RF23	8192 (2048x4)	900	10000	10	5	225	24

5.3. Підтримка пристроїв пам'яті

На рис. 5.1 показана типова структура чіпсета RAM. Інформація зберігається на диску. Акумулятор являє собою матрицю, що складається з *елементів пам'яті* (ЕП), розташованих уздовж рядків і стовпців. Елемент пам'яті може зберігати 1 біт інформації (логічну 1 або логічний 0). Крім того, він оснащений схемами управління, що дозволяють встановити елемент в один з трьох режимів:

- *режим зберігання* , при якому він відключений від входу і виходу мікросхеми;
- *режим читання* , при якому інформація, що міститься в ЕП, виводиться на вхід мікросхеми;
- *режим запису* , при якому нова інформація, що надходить з входу мікросхеми, записується в ЕП.

Кожному ЕР присвоюється номер, який називається *адресою елемента* . Для пошуку потрібного ЕР вкажіть рядок і стовпець, що відповідають позиції ЕР в приводі. Адреса ЕП у вигляді двійкового числа надходить по адресній шині в адресний регістр. Кількість біт адреси пов'язана з обсягом пам'яті. Кількість рядків і стовпців диска вибирається рівним сумарному ступеню 2. А якщо кількість рядків $N_{\text{ряд}} = 2^{n_1}$ і стовпців $N_{\text{стовп}} = 2^{n_2}$ дорівнює , то загальна кількість ЕП (ємність диска)

$$N = N_{\text{ряд}} \cdot N_{\text{стовп}} = 2^{n_1+n_2} = 2^n ,$$

де $n = n_1 + n_2$ - кількість розрядів адреси, які приймаються в адресний регістр.

Наприклад, контейнери $N = 2^{10} = 1024$ кількість бітів адреси $n = 10$; вибирається одночасно $n_1 = n_2 = n/2 = 5$, у цьому випадку кількість рядків і кількість стовпців диска рівні $2^{n_1} = 2^{n_2} = 32$.

Розряди регістра адреси поділяються на дві групи: одна група з n_1 розрядів визначає двійкове число рядка, в якому знаходиться ЕП в накопичувачі, друга група з n_2 розрядів визначає двійкове число стовпця, в якому вибраний ЕР буде розміщено. Кожна група бітів адреси подається на відповідний декодер: декодер рядків і декодер стовпців. У той же час кожен декодер створює логічний рівень 1 на одному зі своїх вихідних ланцюгів (логічний рівень 0 встановлюється на інших виходах декодера); на обраний ВП впливає рівень логічної одиниці одночасно в колах рядків і стовпців. Під час зчитування вміст ЕР надходить на підсилювач зчитування і далі на вихідний тригер і вихід мікросхеми. Режим запису встановлюється шляхом подачі сигналу на вхід запису. При логічному рівні 0 на вході ДЗ відкривається підсилювач запису і частина інформації з вхідних даних надходить на обраний ЕП і там зберігається.

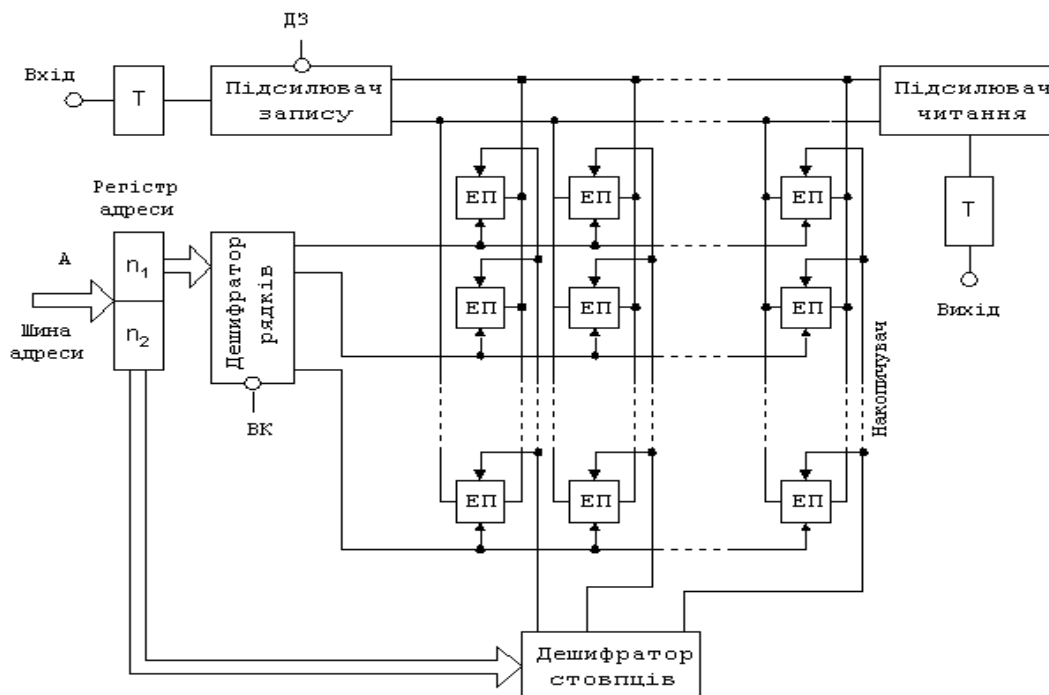


Рис. 5.1. Структура мікросхеми оперативної пам'яті.

Ці процеси відбуваються, коли на вході вибору кристала (VC) є активний рівень логічного 0. На рівні логічного 1 цей вхід встановлює рівень логічного 0 для всіх виходів декодера, і ЗР переходить у режим зберігання. На рис. 5.2 показано умовне графічне позначення мікросхеми ОЗУ.

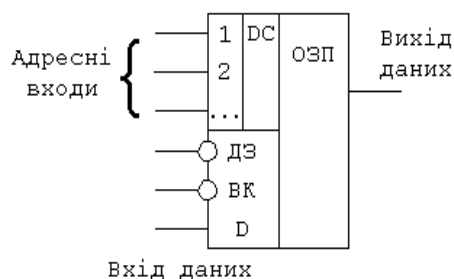


Рис. 5.2. Умовне позначення чіпсета оперативної пам'яті

Розглянемо послідовність сигналів у режимах читання та запису. На рис. На рис. 5.3 а показаний часовий графік сигналів у режимі читання. З деякою затримкою $t_{зам1}$ по відношенню до моменту надсилання адреси і сигналу на ІМС (пов'язаних з процесами декодування адреси і включенням вихідних кіл обраного ЕП) вміст виділеного ЕП мікросхеми з'являється на вихід. У режимі

реєстрації (рис. 5.3б) повинні бути виконані умови, що виключають порушення вмісту комірок, які не адресуються.

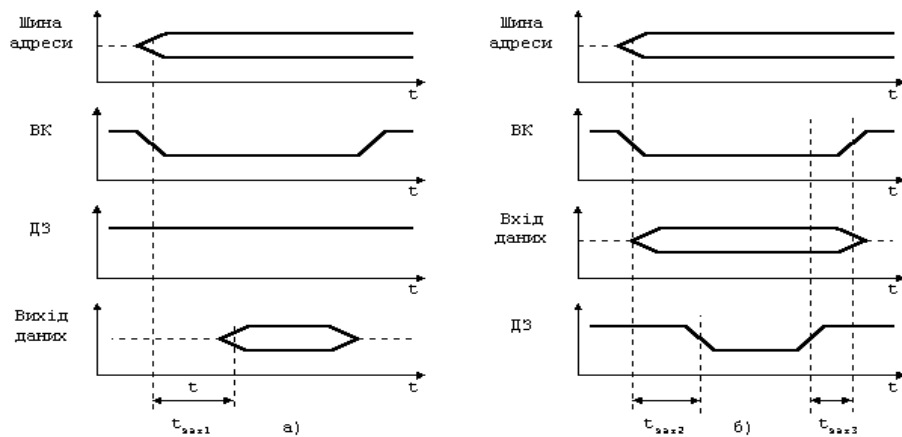


Рис. 5.3. Графіки синхронізації сигналу:
а) у режимі читання; б) в режимі запису

Це забезпечується тим, що сигнал в ланцюг РЗ подається із затримкою $t_{зам2}$ відносно моменту сигналів в ланцюгах адреси, ВК і вхідних даних, а сигнал в ланцюзі ДЗ знімається раніше сигналу в ВК. ланцюг видаляється. В іншому випадку при передчасній подачі сигналу DZ він може записати в комірку з адресою, яка не відповідає інформації на адресних входах мікросхеми.

Мікросхеми оперативної пам'яті дозволяють збільшити ємність пам'яті за рахунок збільшення розміру бітів (і, отже, кількості слів, які можуть зберігатися в них) і збільшення кількості комірок (і, отже, кількості слів, які можуть зберігатися в пам'яті).

Таким чином, використовуючи відповідну кількість мікросхем у певній комбінації, можна побудувати пам'ять з необхідною організацією. Розглянемо схему збільшення швидкості передачі клітинки (рис. 5.4). Для всіх мікросхем дається одна і та ж адреса. При читанні кожна мікросхема видає певну цифру слова, що зчитується. Під час запису вхідне слово крок за кроком записується в EP окремих чіпсетів. Отже, якщо мікросхеми мають організацію $N \times 1$ (N од-нобітових комірок), то для блоку пам'яті з організацією $N \times n$ (N комірок з розміром біта n кожна) Потрібно n мікросхем .

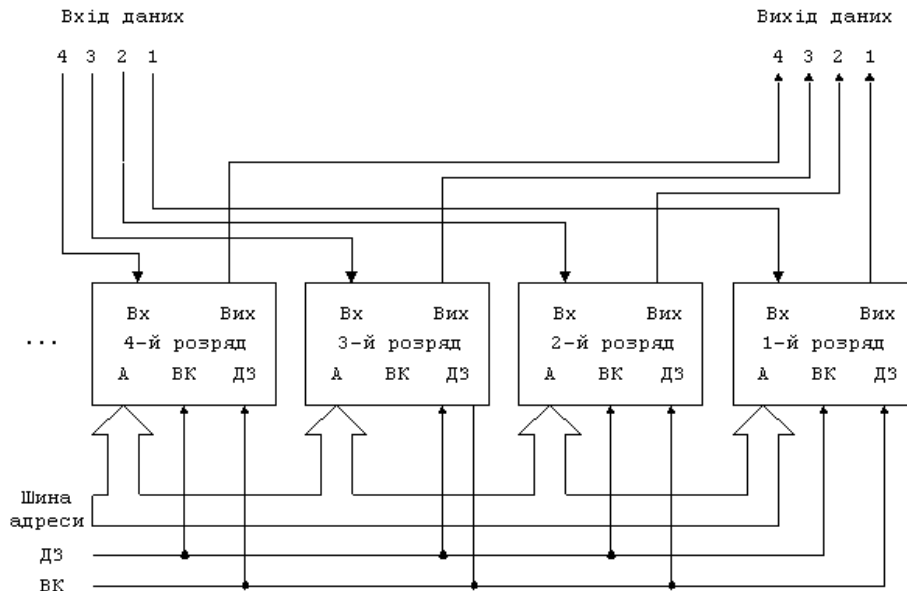


Рис. 5.4. Схема підвищення пропускної здатності осередків ЗП

На рис. 5.5 показана схема збільшення кількості та швидкості пропускання комірок. Блок пам'яті складається з мікросхем, що утворюють окремі лінії (ряди), кожна з яких побудована за схемою збільшення пропускної здатності (рис. 4.4). При цьому розряди адреси блоку пам'яті поділяються на дві групи: A_1 і A_2 . Група цифр A_2 визначає номер рядка, група цифр A_1 - номер клітинки у виділеному рядку. Вибір рядка виконується дешифратором, на вхід якого подається живлення від A_2 , а кожен з виходів підключений до входу ВК певної лінії. Таким чином, в залежності від кодової комбінації, що міститься в A_2 , на виході відповідного дешифратора з'являється логічний рівень 0, що забезпечує вибір конкретної лінійки мікросхем. Логічний рівень 1 надходить з виходу дешифратора на входи ВК інших ліній, а мікросхеми цих ліній переводяться в режим зберігання, в якому вони не відповідають на адресну групу A_1 .

Розглянемо приклад збільшення ємності блоку пам'яті. Припустимо, що на мікросхемах з організацією 1024×1 необхідно побудувати блок пам'яті, тобто організацію 4096×8 , тобто блок пам'яті з 4096 8-розрядними осередками. Збільшення швидкості передачі вимагає в кожному рядку діаграми на рис. 4.5 використовувати 8 мікросхем; щоб збільшити кількість комірок з 1024 до 4096

(в 4 рази), необхідно передбачити 4 лінії мікросхем. Отже, загальна кількість фішок $8 \times 4 = 32$. У такому блоці пам'яті адресна адреса створюється наступним чином. Для виділення рядка в адресі потрібна дворозрядна група A_2 , кожна з чотирьох кодових комбінацій цієї групи (00, 01, 10, 11) буде відповідати певному рядку в блоці пам'яті. Для виділення комірки в лінії мікросхеми необхідна наявність в адресі 10-розрядної групи A_1 (кількість комбінацій 10-розрядної групи $2^{10} = 1024$ дорівнює кількості ЕП в мікросхемі). Тому адреса даного блоку пам'яті повинна мати довжину 12 біт.

У кожному стовпчику матриці мікросхеми на рис. 5.5, виходи всіх мікросхем з'єднані в коло відповідного розряду виходу даних блоку, всі входи даних - в коло відповідного розряду введення даних блоку пам'яті.

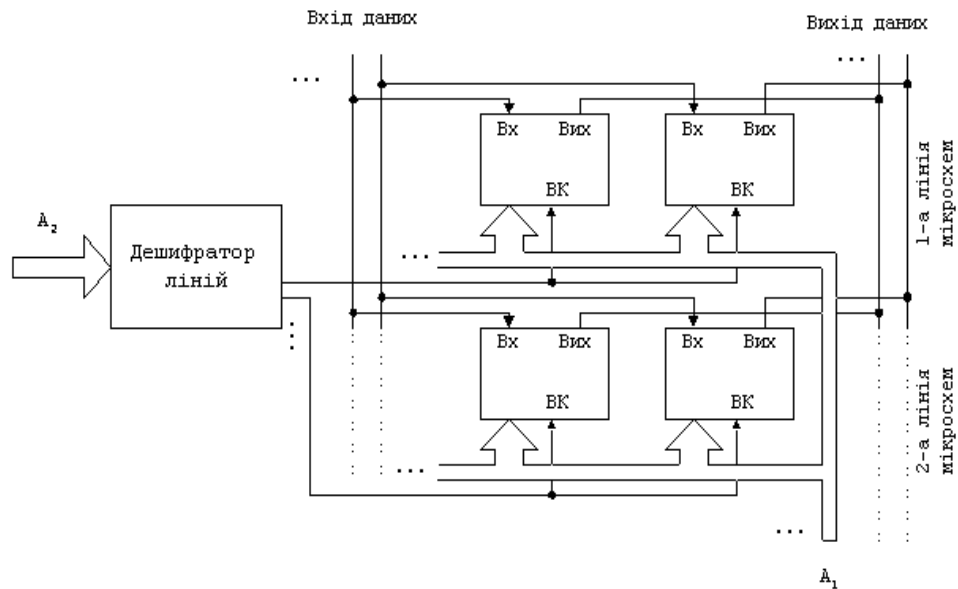


Рис. 5.5. Схема збільшення кількості та швидкості передачі клітин ЗП

5.4 Постійні складські приміщення

Як і оперативна пам'ять, РЗР складається з комірок, до яких можна отримати доступ для відображення їх вмісту. Відмінність від оперативної пам'яті в тому, що інформація записується в комірки один раз, після чого під час операцій використовується тільки режим читання.

За способом введення інформації ПЗП поділяють на два типи: ПЗП, програмований за допомогою маски на підприємстві-виробнику, і ПЗП, програмований користувачем.

У першому відбувається внесення інформації в процесі виробництва мікросхеми за допомогою відповідного фотошаблону. Звичайно, такий спосіб обліку доречний у випадках, коли видається велика партія ПЗП, яка містить однакову інформацію. Промисловість виробляє такі ПЗП, наприклад, для використання в якості перетворювача двійкового коду в конкретний двійково-десятковий код та інших перетворювачів. У них вхідна кодова комбінація служить адресою комірки, а вміст комірки — вихідною кодовою комбінацією (наприклад, двійково-десятькова кодова комбінація).

У ПЗП, що програмується користувачем, запис інформації здійснюється безпосередньо користувачем за допомогою спеціальних пристроїв, які називаються програматорами. Програміст подає відповідну напругу на мікросхему для запису інформації, набраної на клавіатурі. Ці напруги викликають перегорання запобіжників в елементах пам'яті. Звісно, після реєстрації інформації в ПЗП її неможливо змінити надалі. При необхідності зміни змісту ПЗП мікросхеми з раніше збереженою інформацією замінюються на нові, в яких збережені нові дані.

На рис. 5.6 наведено структуру програмованого користувачем ПЗП. Як і оперативна пам'ять, масив пам'яті складається з елементів пам'яті (ЕП), які утворюють рядки та стовпці, але на відміну від оперативної пам'яті при читанні з запам'ятовуючого пристрою відображається вміст усього ряду елементів пам'яті. Цей рядок зазвичай містить кілька слів. За допомогою селектора потрібне слово вибирається з рядка і надсилається на вихід.

Нехай, наприклад, ПЗП має потужність $M = 2^{10}$ біти, розділені на $N = 2^8$ слова $2^2 = 4$ за цифрами в кожному слові. Диск міститиме 2^{10} елементи пам'яті, розташовані вздовж $2^5 = 32$ рядків і $2^5 = 32$ стовпців. При подачі заявки необхідно вказати адресу слова; у цьому прикладі адреса міститиме 8 бітів, розділених на дві групи бітів A_2 і A_1 : 5-бітну групу A_1 та 3-бітову групу A_2 .

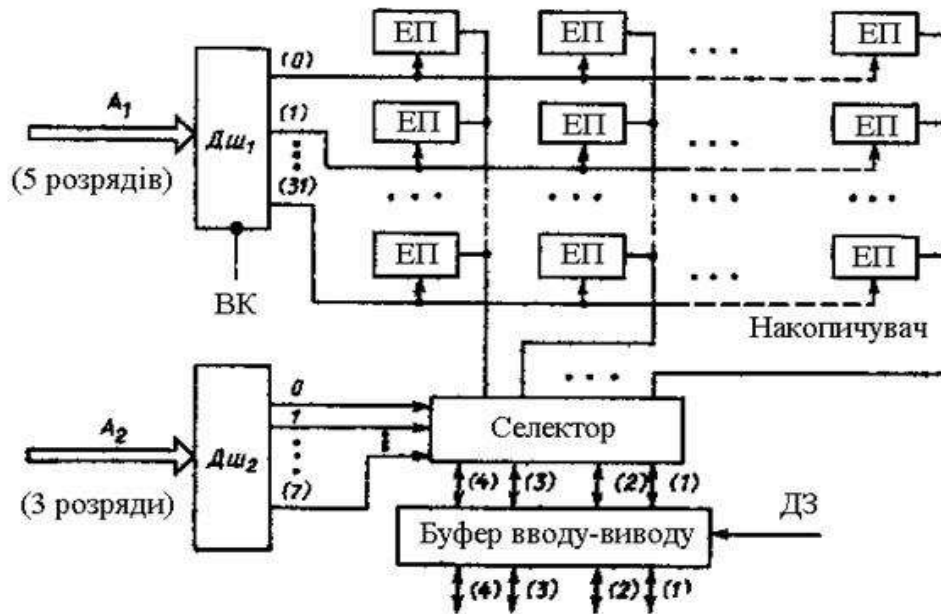


Рис. 5.6. Структура ПЗП програмується користувачем.

Група A_1 подається на дешифратор Dsh_1 , який вибирає одну з $2^5 = 32$ ліній приводу. Вміст рядка складається з 32 бітів або восьми 4-бітних слів. Номер слова в рядку визначається групою A_2 . Декодер Dsh_2 перетворює цю групу адрес у сигнал на одному з восьми своїх виходів. Після цього сигналу з вмісту рядка в селекторі вибирається потрібне слово, яке через буфер введення-виведення надсилається на вихід мікросхеми.

На рис. 5.7 показана принципова схема акумулятора і селектора для цього прикладу. Система побудована на біполярних транзисторах (біполярні транзистори використовуються для побудови швидкодіючих ПЗП).

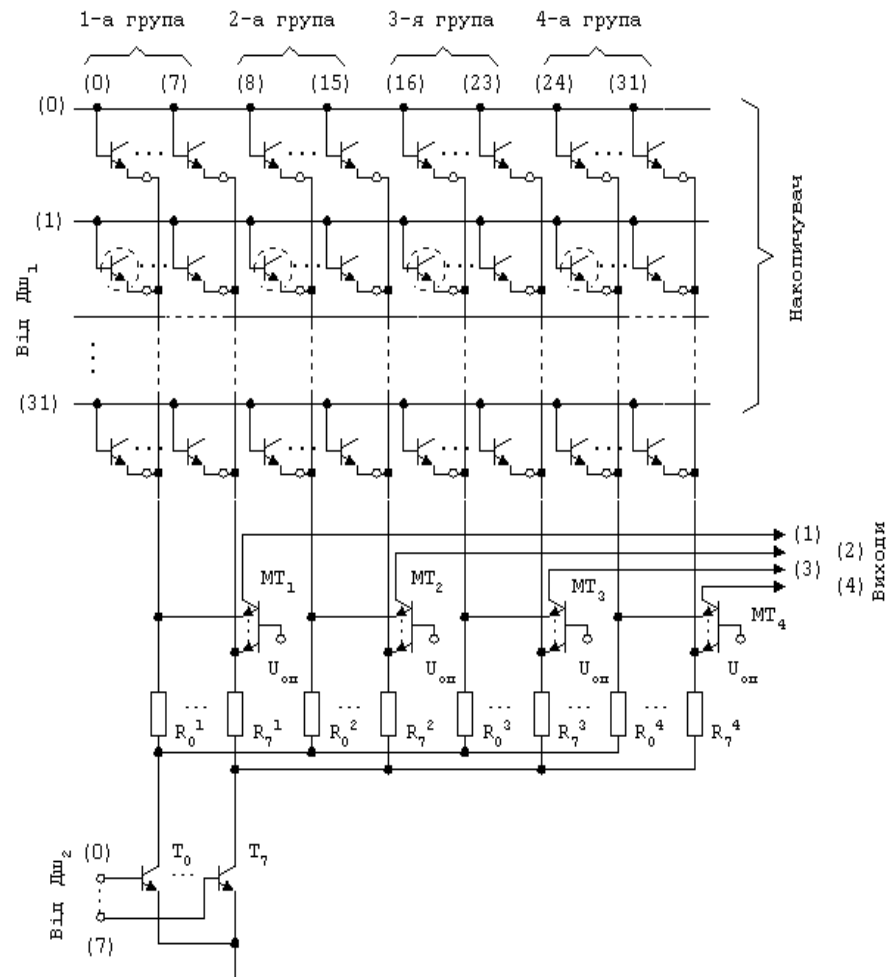


Рис. 5.7. Принципова схема акумулятора і селектора ПЗП

Батарея містить 2^{10} транзистори (елементи пам'яті), які утворюють 32 рядки та 32 стовпці. Колектори накопичувальних транзисторів підключені до джерела живлення (для простоти схема підключення колекторів транзисторів до джерела живлення не показана). У ланцюзі емітера кожного транзистора є плавка вставка (плавкі вставки обведені на малюнку). Перемичка виготовляється з ніхрону, полікремнію або титанату вольфраму і має опір в кілька десятків Ом.

Під час програмування через транзистор достатньо пропустити імпульс струму 20...30 мА тривалістю приблизно 1 мс, щоб спалити перемичку. При роботі в режимі читання струми в транзисторах пам'яті значно менше і не можуть спалити перемички, які не згоріли при програмуванні.

Транзистори VT0...VT7 працюють за селекторною схемою, мультиемітерні транзистори MT1...MT4 - за схемою буфера введення-виведення.

Розглянемо процеси при записі інформації. Групи адрес застосовуються до входів декодерів адреси A_1 , і A_2 логічний рівень 1 буде створено на одному з виходів кожного декодера Нехай логічний рівень 1 з декодера Dsh₁ входить у рядок номер 1, а з декодера Dsh₂ входить логічний рівень 1. обведіть цифру 0 і транзистор в селекторі відкривається VT₀. При цьому в накопичувачі відкриваються транзистори, позначені штриховою лінією (рис. 5.7) (назвемо ці транзистори обраними). Потім ми збільшимо напругу U_{on} та подамо до виходу мікросхеми кодову комбінацію, що зберігається у вибраних чотирьох бітах накопичувача (виходи мікросхеми, які під час читання служать виходами зчитуваного слова, під час запису використовуються як входи для запису письмове слово).

Припустимо, що другий вхід мікросхеми живиться логічним рівнем 1. Одночасно відкритий мультиемітерний транзистор MT₂ у буфері введення-виведення; струм емітера цього транзистора, що протікає через резистор, R_0'' створює напругу, яка закриває другий транзистор у вибраних чотирьох транзисторах запам'ятовуючого пристрою. Отже, стан транзисторів обраної четвірки визначається записаним словом (вибраний транзистор відкритий, якщо відповідний біт записаного слова містить логічний 0, і, навпаки, цей транзистор закритий, якщо біт записаного слова містить логічне 1). Потім збільшимо подачу напруги на колектор транзисторів пам'яті. Через відкриті вибрані транзистори буде протікати великий струм, який згорить перемички в ланцюзі емітера цих транзисторів. Тому перемичка в ланцюзі емітера обраного транзистора згорає, якщо на відповідний вихід поставлений логічний 0. Таким чином необхідна інформація може бути записана на всі елементи пам'яті.

Розглянемо процеси при зчитуванні інформації з ПЗП. При вказівці адреси (групи адрес A_1 і A_2), як обговорювалося вище, вибираються конкретні чотири транзистори приводу. Якщо перемичка в ланцюзі емітера вибраного

транзистора не перегоріла, струм цього транзистора створює напругу на резисторі, яка закриває відповідний мультиемітерний транзистор; якщо перемичка перегоріла, то мультиемітерний транзистор перегорів. Відкритий або закритий стан мультиемітерних транзисторів $MT_1 \dots MT_4$ визначає значення розрядів слова.

Розглянемо тип ПЗП К556РТ4 та його програмування. Мікросхема ПЗП (рис. 5.8 а) має організацію 256×4 . Відповідно, він забезпечує 8 адресних входів (виходи мікросхеми 5,6,7,4,3,2,1,15) і чотири входи/виходи даних (виходи 12,11,10,9), які є виходами для читання та введення в режимі запису. Вивід 16 використовується для підключення джерела живлення, вивід 8 є загальним, вивід 14 (С) є виводом програмування.

Режим зчитування встановлюється подачею напруги 5 В на висновок 16, на висновки 13 (VK) і 14 (С) - напруга логічного рівня 0. Входи даних 12, 11, 10, 9 побудовані за схемою з відкритим колектором. їх необхідно вмикати за схемою, наведеною на рис. 5.8 б.

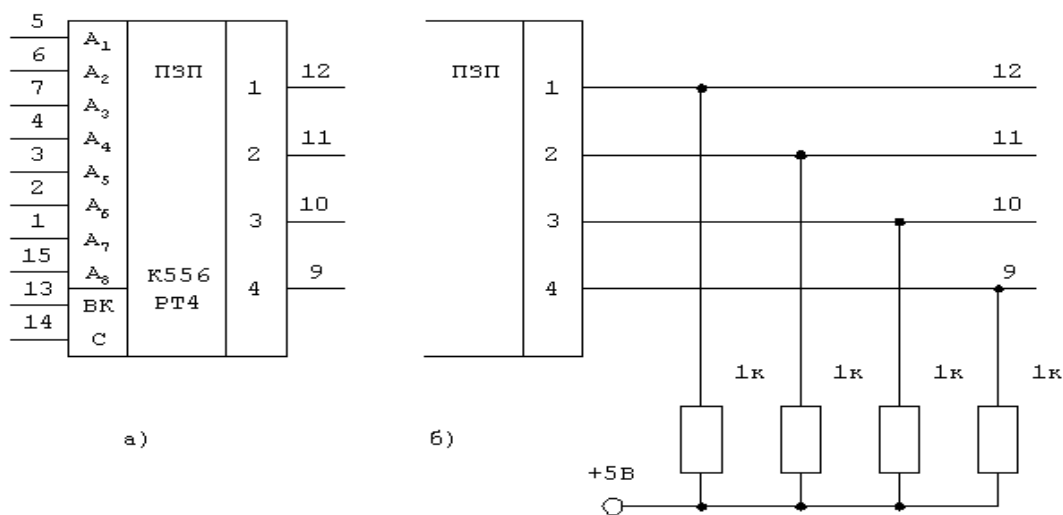


Рис. 5.8. Мікросхема ПЗП К556РТ4:

(а) умовне позначення; б) схема ввімкнення виходів «відкритий колектор».

Під час програмування (в режимі запису) імпульси видаються згідно з часовою діаграмою, наведеною на рис. 2. 5.9:

- 1) на адресних входах задається адреса комірки;

2) напруга живлення (на клемі 16) підвищується з 5 В до 10 В (джерело живлення має бути розраховане на силу струму не менше 400 мА);

3) на вихід програмування 14 подається напруга 15 В (струм джерела повинен бути обмежений 100 мА);

4) 10 В подається на вихід програмування через резистор 300 Ом (під час запису логічної 1). За один цикл можна запрограмувати лише одну цифру.

Збільшення ємності PZP відбувається за тими самими схемами, що й збільшення ємності RAM.

5.5. Пристрої енергонезалежної пам'яті були перепрограмовані

Перепрограмовані PDP мають усі переваги PDP, зберігаючи збережену в них інформацію необмежений час, навіть після вимкнення живлення. При цьому дозволяється видалення збереженої інформації та збереження нової інформації. Однак, якщо читання виконується за частки мікросекунди, то запис займає на багато порядків більше часу.

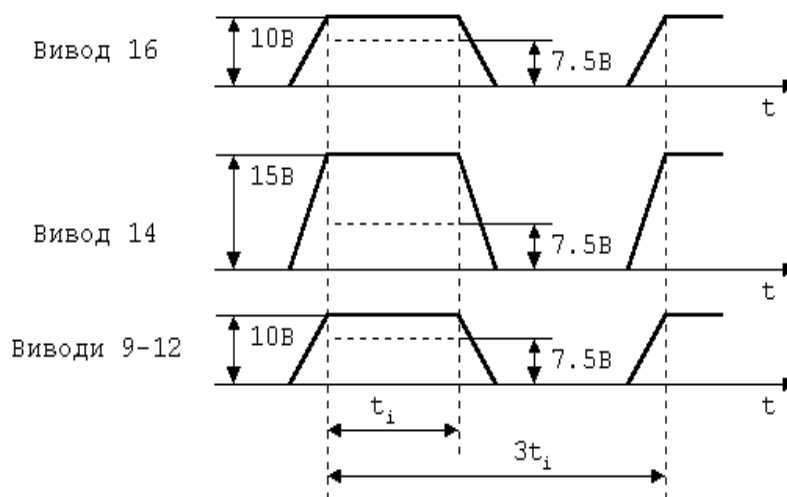


Рис 5.9. Часові діаграми сигналів при запису інформації в ПЗП

Розглянемо принцип дії, зображений на рис. 5.10 *Елемент пам'яті* з електричним записом інформації та стиранням під дією ультрафіолету.

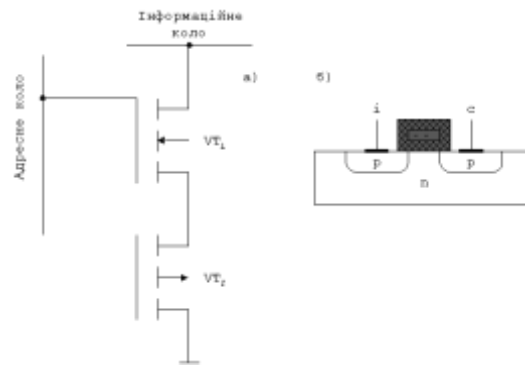


Рис. 5/10. Елемент пам'яті ППЗП: а) схема, б) структура

Транзистор VT_1 використовується для вибору елемента пам'яті. Інформація зберігається в транзисторі VT_2 . Особливість транзистора VT_2 , будова якого наведено на рис. 5.10 б, полягає в тому, що він має ізольований затвор. Коли достатньо висока напруга прикладається до *pn* переходу витоку або стоку, електрони інжектуються в затвор, і цей заряд може залишатися на затворі протягом тривалого часу. Негативний заряд на затворі, притягуючи отвори, створює провідний р-канал між стоком і стоком в області n. Якщо до *pn*-переходу не прикладається підвищена напруга відсутність заряду на затворі, транзистор буде в закритому стані (логічний стан 1).

В одних мікросхемах інформація стирається шляхом подачі відповідних напруг, в інших – ультрафіолетового випромінювання через прозорий кварцовий екран у корпусі мікросхеми. Під дією напруги або світлового випромінювання, що діє протягом приблизно 10 хвилин, знімається заряд із затворів транзисторів і всі транзистори накопичувача переходять у непровідний стан. Звичайне освітлення приміщення практично не впливає на стан транзисторів.

Розглянемо мікросхему ППЗП типу К573РФ1 (рис. 5.11) та її програмування.

8	A ₀	ППЗП		9
7	A ₁			10
6	A ₂		0	11
5	A ₃		1	13
4	A ₄		2	14
3	A ₅		3	15
2	A ₆		4	16
1	A ₇		5	17
23	A ₈		6	
22	A ₉		7	
20	VK/Зп			
18	С			

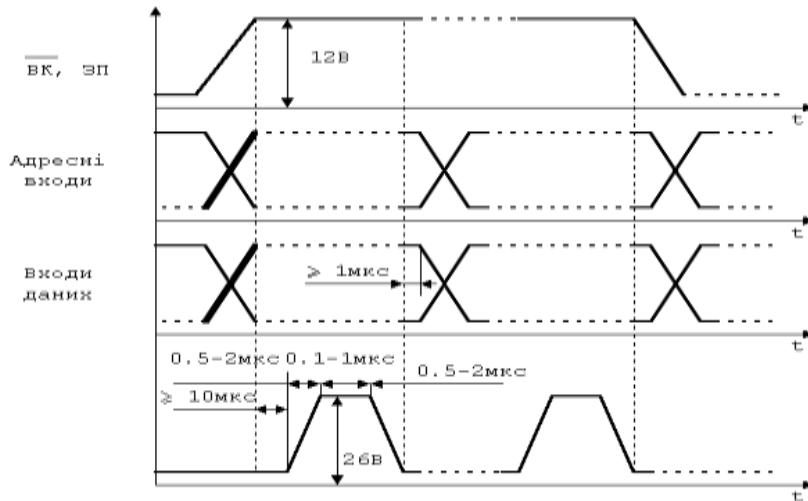


Рис. 5.11 Мікросхема ППЗП К573РФ1 Рис. 5/12 Часові діаграми процесу фіксації інформації в ППЗП

Ця мікросхема має організацію 1024 на 8, забезпечує 10 адресних входів (виводи з номерами 8, 7, 6, 5, 4, 3, 2, 1, 23, 22) і 8 ввідів-виводів даних (виводи 9, 10, 11, 13, 14, 15, 16, 17), сумісний з логікою TTL.

Режим читання інформації. Для мікросхеми потрібно три джерела живлення +12 В (висновок 19), +5 В (висновок 24), -5 В (висновок 21) відносно загального виводу 12. На вході VK/Zp (висновок 20) і вході програмування (18) рівень логічної напруги встановлюється на 0. Виводи 9...11, 13...17 служать виходами даних, на які вводиться кодова комбінація вмісту комірки пам'яті, адресу якої вказано, і адресу з'являються входи мікросхеми.

Рис. 5.11. Умовне маркування мікросхеми К573РФ1.

Режим видалення інформації. Видалення інформації відбувається шляхом подачі на чіп високоінтенсивного випромінювання з довжиною хвилі не

більше 400 мкм через прозорий для випромінювання екран. Як джерело ультрафіолетового випромінювання можна використовувати лампи ДРТ-220 або ДРТ-375. У процесі стирання необхідно замкнути виходи мікросхеми. Після стирання всіх цифр всіх комірок пам'яті на мікросхемі встановлюється логічна 1.

Режим запису інформації. Цикл запису починається з подачі напруги +12 В на вхід VK/Zp (контакт 20). На адресних входах послідовно задаються адреси комірок пам'яті, а інформація в комірки записується у вигляді паралельного 8-розрядного коду. комбінації подаються на висновки 9...11, 13...17, які в режимі запису служать входами для записаних даних. Після введення кодових комбінацій адреси та зареєстрованого номера з заданою затримкою на вхід програмування С (вивід 18) подається імпульс програмування 26 В. Часова діаграма імпульсів у режимі реєстрації показана на рис. 5.12. подача одного імпульсу програмування на кожен адресу визначає цикл запису. Під час кожного циклу запису всі комірки пам'яті повинні бути запрограмовані. Програмування окремих комірок або груп комірок заборонено. Необхідна кількість циклів програмування пам'яті визначається за формулою $N = 100\text{мс} / t_i$, де t_i програмна тривалість імпульсу, мс. Так, якщо $t_i = 0.5\text{мс}$, необхідно точно записувати інформацію $N = 200$ циклів запису.

Перепрограмовані ПЗП коштують дорожче ПЗП і використовуються в процесі налагодження мікропроцесорних пристроїв, коли необхідно уточнити інформацію, яка повинна зберігатися в пам'яті. Після адаптації ПЗП можна замінити на більш дешевий ПЗП.

РОЗДІЛ 6. СИСТЕМИ ВВЕДЕННЯ ТА ВІДОБРАЖЕННЯ ЦИФРОВИЙ ІНФОРМАЦІЯ

6.1. Пристрої цифрового введення

Кожна цифрова система містить засоби введення-виведення інформації. В якості пристроїв введення для генерації керуючих і кодових сигналів можна використовувати кнопкові перемикачі та контактні площадки. В якості пристроїв виведення широко використовуються кнопки, цифрові індикатори, знаки та індикаторні табло на їх основі.

Сигнал за допомогою кнопкових перемикачів створюється замиканням і розмиканням електричного кола з їх допомогою (рис. 6.1 *a*). Іншим методом отримання сигналу є використання т.зв "контактний зонд" (Рис. 6.1 *б*). Тут потенційний сигнал U береться з виходу площини контакту в початковому стані, і коли зонд торкається площини, рівень сигналу досягає нуля. Сигнали, що генеруються парою контактів, супроводжуються деренчанням (рис. 6.1 *в*), тривалість якого становить 8-12 мс. Для усунення деренчання в прийнятому сигналі (рис. 6.1 *г*) на контактному виході встановлені спеціальні формувачі. На рис. 6.2 наведено схеми найпопулярніших форм. У першому на рис. 6.2 *і* принцип прямого налаштування тригера RS за станом перемикача SB1. Форма сигналу з виходу системи формування наведена на рис. 6.1 *ж*. Формувач на малюнку 6.2 *б* є однорозрядним регістром, який тактується з вхідним тактовим періодом T_1^3 10-20 мс. Його роботу ілюструють діаграми на малюнку 6.2 *с*. Перша схема формування проста, але вимагає двох резисторів (R_1, R_2) і двопровідної лінії для підключення до вимикача SB1. Другий формувач вимагає одного резистора R_3 і однодротової лінії, але додатково вимагає джерела живлення тактовими імпульсами. Перевагою другого способу є синхронізація моменту появи вихідного сигналу з внутрішніми процесами пристрою, для якого створюється сигнал, при цьому як тактові імпульси використовуються вже наявні в пристрої сигнали.

Створення кодових сигналів відбувається в кодуючих пристроях. Додатковими функціями кодуючого пристрою є: формування сигналу «Гарячий» (готовність) керувати переписуванням згенерованого коду; блокування роботи при одночасному натисканні декількох клавіш і захист від деренчання.

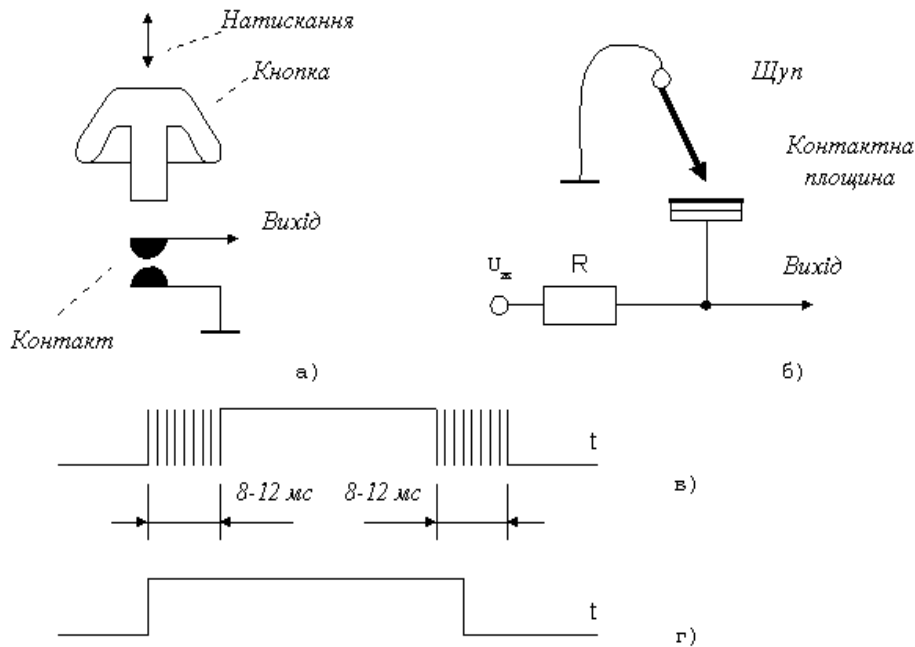


Рис. 6.1. Методи формування електричного контакту в формувачах сигналів (а,б) і сигнали формувача (в,г)

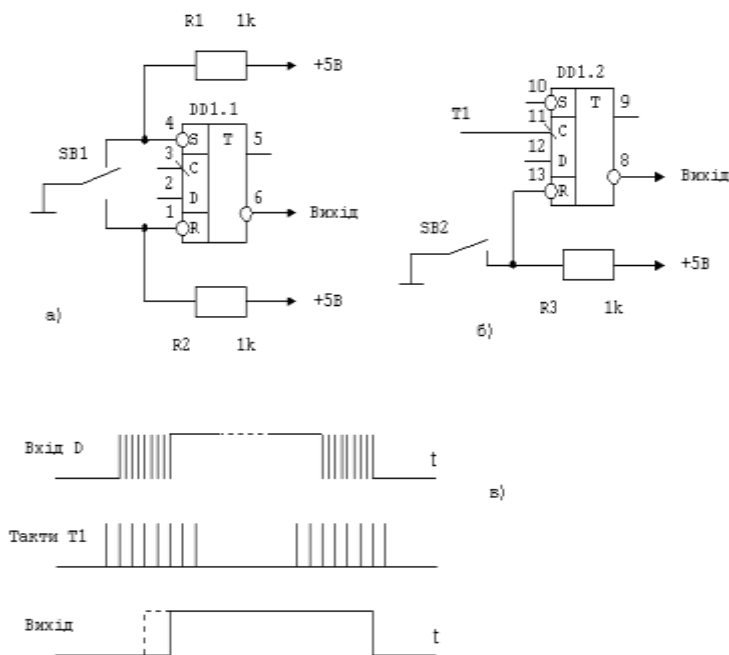


Рис. 6.2. Генератори сигналів (а, б) і часова діаграма генератора (с)

Кодуючий пристрій, структурна схема якого наведена на рисунку 5.3, містить: клавіатуру введення КЛ; Блок кодування КБ; вузол захисту та формування вихідних сигналів ФС (число n на шині означає кількість ліній у цій шині).

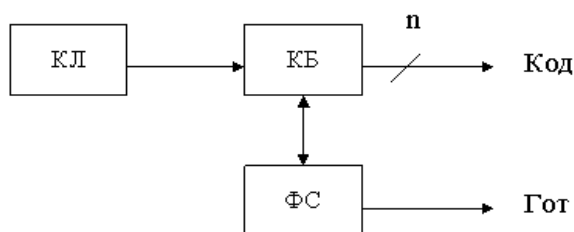


Рис. 6.3. Структурна схема кодуючого пристрою

Найбільшого поширення набули комбіновані кодувальні пристрої та пристрої зі сканованою клавіатурою.

Комбінований кодуючий пристрій (рис. 6.4) виробляє 16 чотирирозрядних кодів. Блок кодування в пристрої складається з чотирьох елементів АБО-НІ на DD1-DD4. Вузол FS пристрою дає можливість п'ятиканальної установки АБО-НІ (DD5.1, DD5.2) і генераторів сигналів на тригерах DD6.1 і DD6.2.

При натисканні будь-якої клавіші на клавіатурі введення вихідні сигнали з елементів DD1-DD4 формують відповідний чотирирозрядний код, на виході тактованого тригера DD 6.2 формується сигнал «Натиснути». Тригер DD6.1 встановлюється в єдиний стан по фронту сигналу і скидається після закінчення циклу на вході R.

Пристрій-кодер на основі сканованої клавіатури (рис. 6.5) також виробляє чотирирозрядний двійковий код і містить: блок сканування (попередньо вмикаються чотирирозрядний двійковий лічильник DD2, шістнадцятковий мультиплексор DD3), генератор сигналу «Готовність». для тригера DD4.1 і елемента I-NO DD1.2. Термін «сканування клавіатури» означає послідовний запит стану всіх елементів клавіатури.

У пристрої 16 виходів вхідної клавіатури з'єднані з 16 входами мультиплектора DD3. В основному режимі лічильник DD2 працює від зовнішніх тактових сигналів, що проходять через елемент DD1.1. Для кожного стану лічильника мультиплексор опитує стан (натиснуто-відпущено) відповідного вхідного контакту клавіатури.

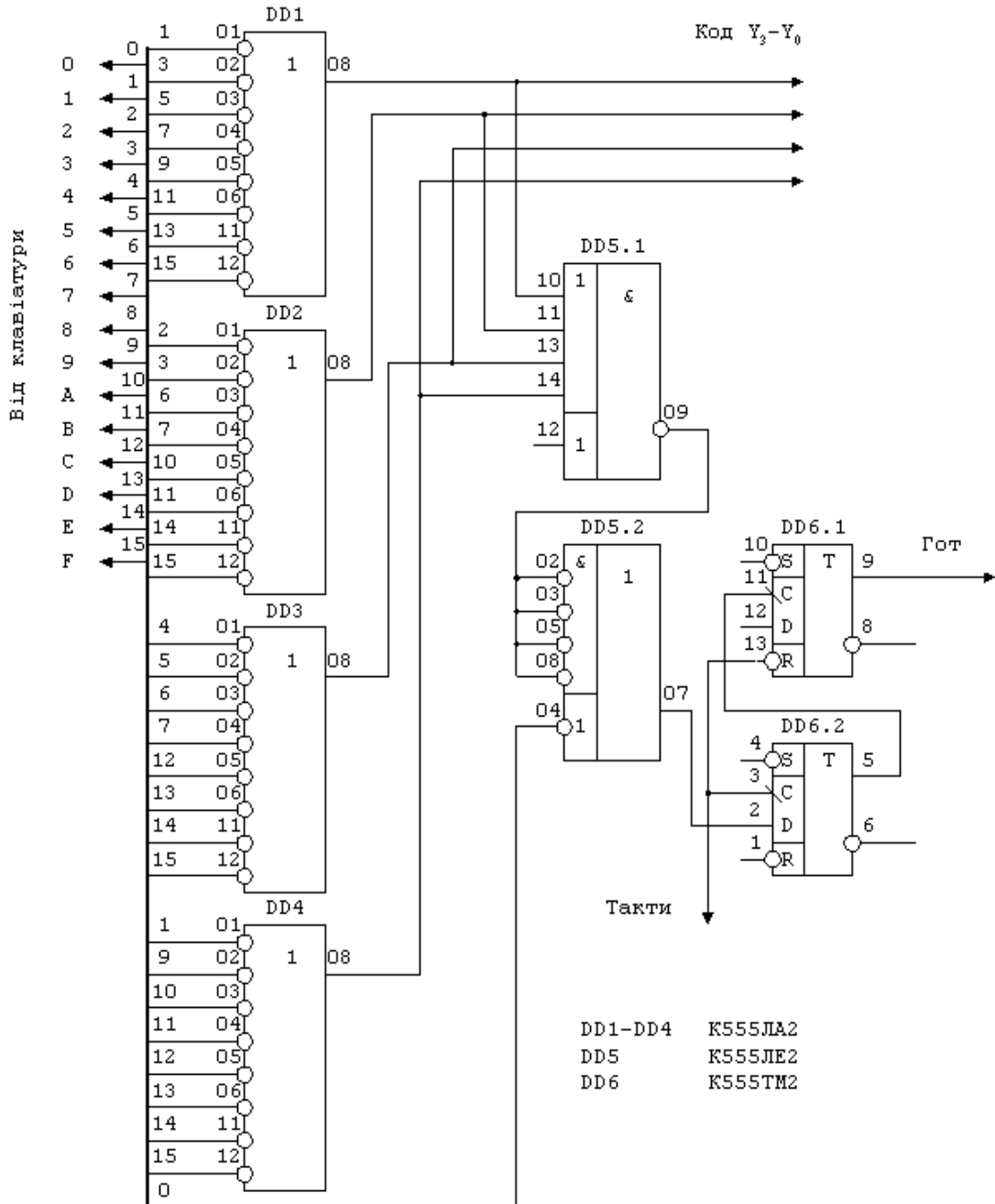


Рис. 6.4. Принципова схема блоку комбінаційного кодування

Перевагами пристрою зі сканованим кодом CL перед комбінованим пристроєм є: менша вартість обладнання, наявність захисту від створення помилкового коду при одночасному натисканні кількох клавiш CL.

6.2. Елементна основа світлофорних пристроїв

Найпростішими пристроями відображення інформації в цифрових пристроях є: світлодіоди, цифрові індикатори та світлові табло.

Напівпровідникові світлодіоди використовують властивість рп-переходу випромінювати світло у видимій частині спектру, коли через них пропускається постійний струм. Світлодіоди доступні в червоному, жовтому, зеленому та помаранчевому, а також у різних кольорах. Важливим параметром світлодіода є інтенсивність світла, яка вимірюється в міліканделях (мКд). Він пропорційний постійному струму, що протікає через світлодіод. Світлодіоди випускаються з спрямованим і розсіяним випромінюванням. Перші з них виконані в металевому корпусі з лінзою, що забезпечує спрямоване випромінювання. Розсіяне випромінювання створюють світлодіоди, розміщені в пластиковому корпусі з прозорого компаунду.

Постійна постійна напруга для світлодіодів різних типів становить 2-3 В, а значення постійного (робочого) струму від 5 до 20 мА. Світлодіоди можна використовувати для індикації вихідних рівнів ТТЛ мікросхем і транзисторних каскадів. Світлодіод може світитися як при низькому, так і при високому рівні сигналу на виході мікросхеми. Схеми включення таких показників наведено на рис. 6.6.

Крім світлодіодів випускають цифрові, символні, лінійні та матричні світлодіодні індикатори та табло. Цифрові і символні індикатори бувають одно-розрядними і багатозначними, їх символи складаються з семи і більше (9, 11, 16, 18, ...) сегментів. Крім відрізків покажчики містять десяткову крапку або кому.

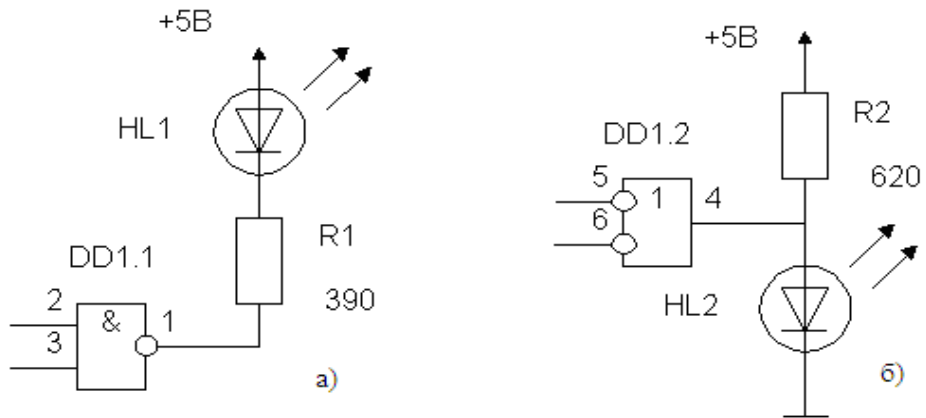


Рис. 6.6. Варіанти ввімкнення світлодіодів для індикації сигналів низького (а) і високого (б) рівнів

Серед цифрових найбільшого поширення набули семисегментні індикатори, у яких стилізоване зображення цифр (і деякого набору літер) складається із семи лінійних сегментів, розташованих у формі вісімки (рис. 6.7). Освітлення виділеного сегмента або групи сегментів після отримання зображення знака забезпечується підключенням їх до контуру протікання струму.



Рис. 6.7. Зображення знаків на основі семисегментного індикатора

На рис. 6.8 наведено зовнішній вигляд і схеми індикаторів ALS324, лінійні шкали на основі світлодіодів - це мікросхеми, сформовані з послідовно з'єднаних сегментів світлодіодів, які вмикаються керуючим пристроєм.

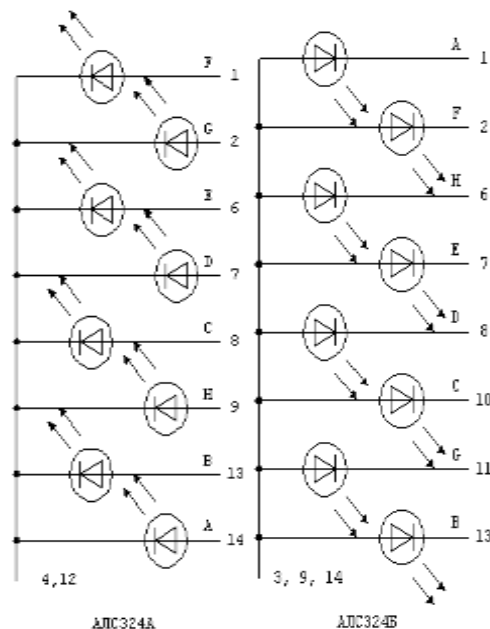


Рис. 6.8. Цифрові світлодіодні індикатори ALS324

Лінійні шкали використовуються для відображення інформації, яка постійно змінюється, наприклад, для вказівки рівня сигналу. Коли увімкнено, лінійна шкала виглядає як лінія, що світиться, довжина якої залежить від кількості задіяних сегментів.

Оскільки рідкокристалічні дисплеї не випромінюють світло, для зчитування цифрової інформації з них потрібне природне освітлення або додаткове джерело світла. Принцип дії такого індикатора заснований на залежності прозорості органічної речовини, що заповнює індикатор, від величини діючого на нього електричного поля. Завдяки цьому змінюється контрастність окремих фрагментів індикатора, з яких створюється необхідне зображення.

Розмір знака і поля індикації покажчика конструктивно не обмежені. Промисловість випускає багато типів рідкокристалічних індикаторів, які широко використовуються в електронних годинниках, мікрокалькуляторах та інших цифрових пристроях відображення. Всі рідкокристалічні індикатори працюють від змінного струму, їх робоча керуюча напруга становить 4-15 В. Струм індикатора, як правило, не перевищує сотень мікроампер, що зумовило їх використання в економічних малогабаритних приладах.

Крім світлодіодних і рідкокристалічних індикаторів, в цифрових пристроях можуть використовуватися також газорозрядні індикатори. Ці індикатори мають один або два аноди і десять катодів, виконаних у вигляді цифр від 0 до 9. Анод газорозрядного індикатора підключений через резистор до джерела постійної або пульсуючої напруги 200-300 В. Середнє значення значення анодного струму для більшості показників становить 1,5 -3 мА. При подачі напруги між анодом і одним з катодів останній починає світитися. В результаті виводиться конкретне число.

6.3. Вказівні пристрої

Великою популярністю користуються пристрої відображення цифрової інформації, які побудовані на основі статичних і динамічних дисплеїв.

Статичний спосіб сигналізації передбачає безперервне горіння індикатора від одного джерела інформації. Наприклад, в електронних годинниках (рис. 6.9) кожен з індикаторів цифрової індикації постійно з'єднаний зі «своєю» декадою лічильника через власний перетворювач кодів (дешифратор). У цьому випадку «втрати» при відображенні n символів становлять $11n$ сполучних проводів у дешифраторах. При використанні комбінованих мікросхем, наприклад, лічильник - дешифратор - індикатор, кількість з'єднувальних проводів значно зменшиться.

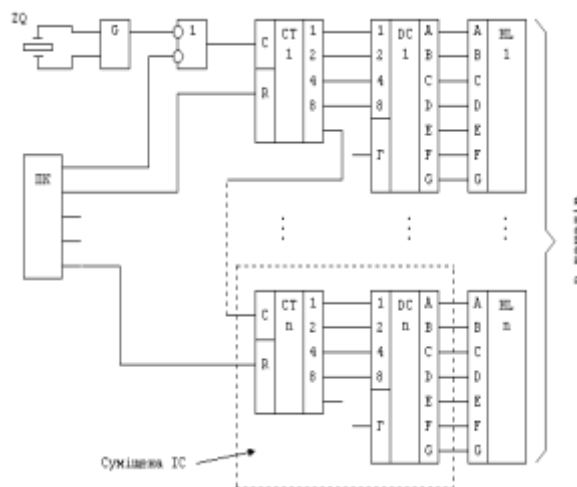


Рис. 6.9. Конструктивна схема пристрою зі статичним дисплеєм

Суть динамічної сигналізації полягає в послідовному, циклічному підключенні кожного індикатора до джерела інформації через загальну схему перетворення коду. Перевагою цього способу є економія на конвертерах кодів і з'єднувальних кабелях. Останнє необхідно при віддаленні дисплея від джерела інформації. Переваги цього методу особливо помітні, коли кількість відображених символів перевищує 4-6.

Принципи побудови цифрового табло, що складається з n символів, пояснюються на прикладі пристрою, схема побудови якого наведена на рис. 5/10. До складу пристрою входять: n -канальний мультиплексор МХ, виконаний за схемою $n \times 4$ ліній по 1, дешифратор DC1 двійково-десятьового коду до семисегментного індикаторного коду, блок керування n станами, що включає двійковий код. лічильник ST і декодер DC2. Лічильник ST працює в режимі постійного підрахунку імпульсів. Коефіцієнт перетворення лічильника $K=n$. Кожен стан лічильника ST гарантує, що відповідний індикатор підготовлений для відображення інформації з декодера DC 1 і що інформація з відповідного джерела надсилається через мультиплексор МН до DC1.

Час свічення одного індикатора протягом циклу роботи лічильника ПП:
 $t_i = \frac{T_y}{n}$, де T_y - час виконання циклу; $t_i = T_G$ дорівнює періоду відстеження імпульсу G . Для усунення мерехтіння зображення на індикаторі встановлюється час циклу в діапазоні від 10 до 15 мс.

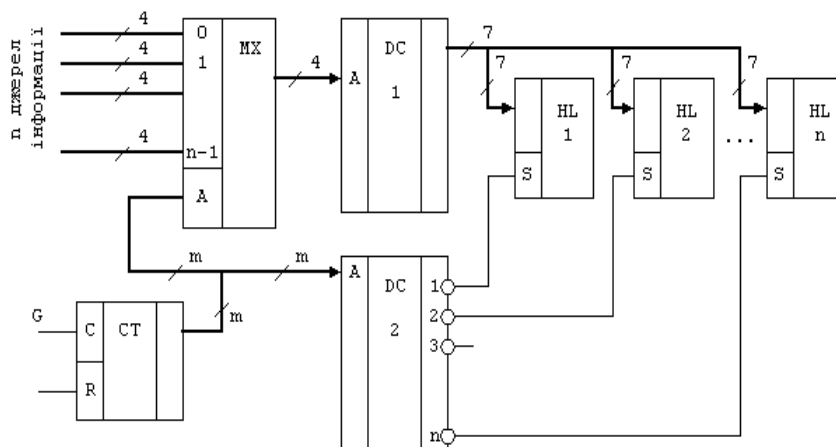


Рис. 6.10. Структурна схема блоку динамічного відображення

Практична схема пристрою, що забезпечує роботу чотирирозрядного цифрового табло, наведена на рис. 6/11.

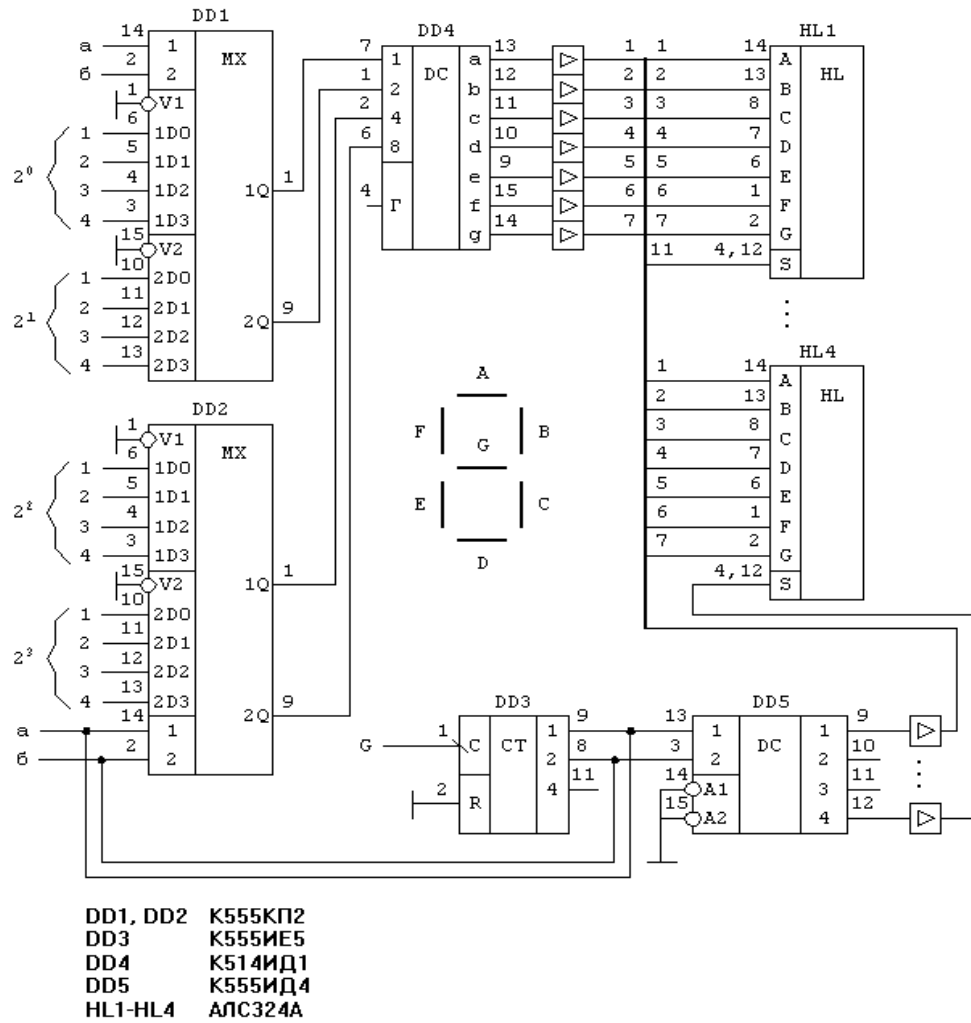


Рис. 6.11. Принципова схема динамічного дисплея

РОЗДІЛ 7. ЦИФРО-АНАЛОГОВІ ПЕРЕТВОРЮВАЧІ

7.1. Загальні відомості

Кожен цифровий пристрій, від найпростішого до найскладнішого, завжди працює за цим принципом: приймає вхідні сигнали, обробляє, передає, зберігає їх і формує вихідні сигнали. При цьому необов'язково, щоб будь-яка зміна вхідних сигналів приводила до негайної і однозначної зміни вихідних сигналів. Реакція пристрою може бути досить складною, відстроченою в часі і неочевидною. Вхідними сигналами можуть бути сигнали з виходів інших цифрових пристроїв, перемикачів або ключів або датчиків фізичних величин. В останньому випадку зазвичай необхідно перетворювати аналогові сигнали датчиків у потоки цифрового коду.

Вихідні сигнали цифрового пристрою (рис. 7.1) можна подавати на інші цифрові пристрої, використовувати для індикації (екран монітора, цифровий індикатор тощо) або створювати фізичні величини. В останньому випадку необхідно перетворити цифрові коди з цифрового пристрою в безперервні (аналогові) сигнали.

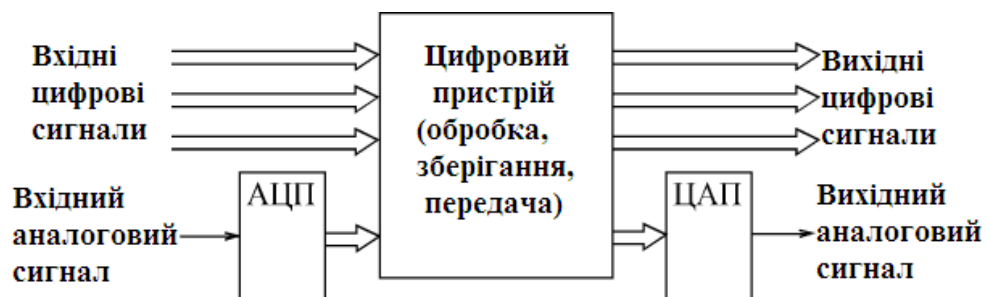


Рис. 7.1. Підключення цифрового приладу

Цифро-аналогові перетворювачі (*DAC* - «цифро-аналоговий перетворювач») і аналого-цифрові перетворювачі (*ADC* - «аналогово-цифровий перетворювач») використовуються для з'єднання цифрових систем із зовнішніми аналоговими сигналами. Отже, АЦП перетворює аналогові сигнали у вихідні цифрові сигнали, які потім надсилаються на цифрові системи для роботи або

зберігання, а ЦАП перетворює ввідні цифрові імпульси цифрових систем в аналогові (див. рис. 7.2).

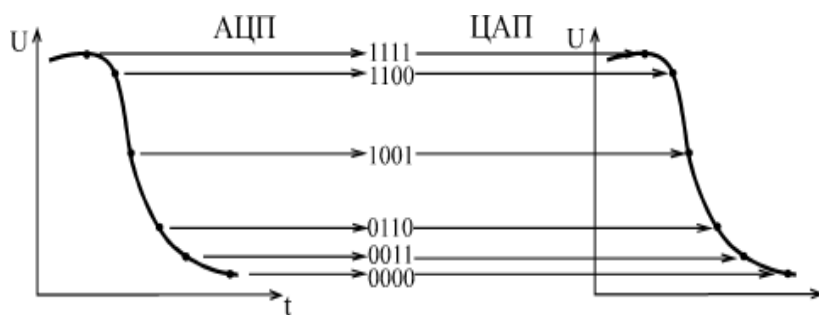


Рис.7.2. Аналогово-цифрове та цифро-аналогове перетворення

ЦАП і АЦП системи використовуються в контрольній техніці (цифрові вольтметри, осцилографи, генератори імпульсів), побутовій техніці (автомобільна електроніка, телевізори, музичні центри,), комп'ютерній техніці (обробка звукових сигналів, відеомонітори, принтери), радіолокаційних пристроях, телефонії та ін. багато інших галузей. Область застосування ЦАП і АЦП перетворювачів стало розширюватися в процесі переведення від аналогових пристроїв до цифрових.

Як ЦАП і АЦП найчастіше використовуються спеціальні мікросхеми виробництва багатьох вітчизняних і зарубіжних компаній. Вміле і професійне застосування мікросхем ЦАП і АЦП вимагає знання аналогових схем, які істотно відрізняються від цифрових. Практичне використання ЦАП і АЦП вимагає розрахунку аналогових схем з урахуванням великої кількості помилок переходу (статичних і динамічних) характеристик і особливостей схем (насамперед операційних підсилювачів).

У розділі ми не буде розрито особливості вибору і підключення деяких мікросхем ЦАП і АЦП, а лише розглянемо основи способів підключення ЦАП і АЦП до цифрових вузлів.

7.2. Основні параметри та класифікація ЦАП перетворювачів

7.2.1. Статичні та динамічні параметри ЦАП

Цифроаналоговий перетворювач — це пристрій декодування, вхідним сигналом якого є цифровий сигнал N , а вихідним — аналоговий сигнал Y , пов'язаний із вхідним коефіцієнтом

$$Y = N * q, \quad (7.1)$$

де q - чутливість (ціна нижчого порядку) перетворювача.

У інтегрованих ЦАП вхідним сигналом зазвичай є двійковий код позиції або заснований на ньому десятковий код. Вихідний сигнал зазвичай є постійною напругою. Такі ЦАП називаються перетворювачами коду в напругу (DCV). Вихідна напруга РСН згідно (7.1) визначається як

$$U_{вихід} = N * U_0 \quad (7.2)$$

де U_0 – ціна молодшого сорту ПКН.

Число N у двійковій системі числення представлено таким виразом:

$$N = a_{n-1} 2^{n-1} + a_{n-2} 2^{n-2} + \dots + a_1 2^1 + a_0 2^0 = \sum_{i=0}^{n-1} a_i 2^i$$

Де i - значення цифр вхідного коду (0 або 1); n – кількість бітів вхідного коду.

Тому вихідна напруга ПКН згідно (7.2) визначається як

$$U_{вих} = \sum_{i=0}^{n-1} a_i 2^i U_0$$

Таким чином, перетворення коду в напругу передбачає підсумовування шаблонів напруги $2^i U_0$ пропорційно вагам бітів вхідного коду. У підсумовуванні беруть участь лише ті стандарти, для яких $i = 1$.

Із збільшенням значення вхідного коду N вихідний сигнал U створюватиме ступінчасту функцію (рис. 7.3, а).

Назвемо таке співвідношення статичною характеристикою перетворення: $U_{out} = f(N)$. Визначає основні статичні параметри ПКН.

В ідеальному РСН всі ступені характеристик перетворення мають однакове значення U_0 , а середини ступенів розташовані на ідеальній прямій 1 (рис. 7.3, а). Основною ознакою ідеального РСН є допустима ємність, яка визначається кількістю біт вхідного коду. Абсолютне значення ємності, що дозволяє такий РСН, дорівнює значенню нижчого класу

$$U_0 = U_{\text{вихід}} / 2^n - 1 \quad (7.4)$$

де $U_{\text{вихід}}$ — максимальна вихідна напруга перетворювача.

Дана проникність потенційно можлива і характеризує лише ідеальні РСН. Наявність інструментальних похибок вузлів реальної ПКС призводить до того, що їх статичні характеристики перетворення суттєво відрізняються від ідеальних розмірів і форми сходів, а також орієнтації в полі координат (рис. 7.3, а). Це спричиняє помилки в перетворювачі та погіршує його роздільну здатність.

Для кількісного опису загальної інструментальної похибки вводиться ряд параметрів.

Абсолютна похибка перетворення в кінцевій точці шкали (повна похибка шкали) $\Delta U_{\text{ю}}$ визначає відхилення значення вихідної напруги U на виході від номінального (розрахункового) значення при максимальному вхідному коді N_{max} (рис. 7.3, а). Повна похибка наведена в довідкових даних в одиницях

молодшого порядку:
$$\delta_{\text{ю}} = \frac{\Delta N_{\text{ю}}}{U_0}.$$

Помилка нульового зсуву на виході $U_{\text{см}}$ — характеризує значення $U_{\text{в}}$ при нульовому значенні вхідного коду (рис. 7.3, а).

Диференціальна нелінійність δ_{ld} характеризує відхилення різниці значень двох вихідних напруг, відповідних двом сусіднім кодам, від значення молодшого розряду.

Похибка нелінійності ΔU_1 характеризує ідентичність приростів (кроків) вихідного сигналу на всьому діапазоні перетворення. Визначається як максимальне відхилення фактичної характеристики перетворення від лінії абсолютної точності 2. Лінія проведена через характерні точки, відповідні мінімальним і

максимальним значенням вхідного коду (рис. 1.1, а). Абсолютне значення похибки нелінійності вимірюється в молодших одиницях:

$$\delta_l = \frac{\Delta U_l}{U_0}$$

Похибка нелінійності (рис. 7.3, б) є найважливішою характеристикою РСН і не повинна перевищувати $\pm \frac{\Delta U_0}{2}$.

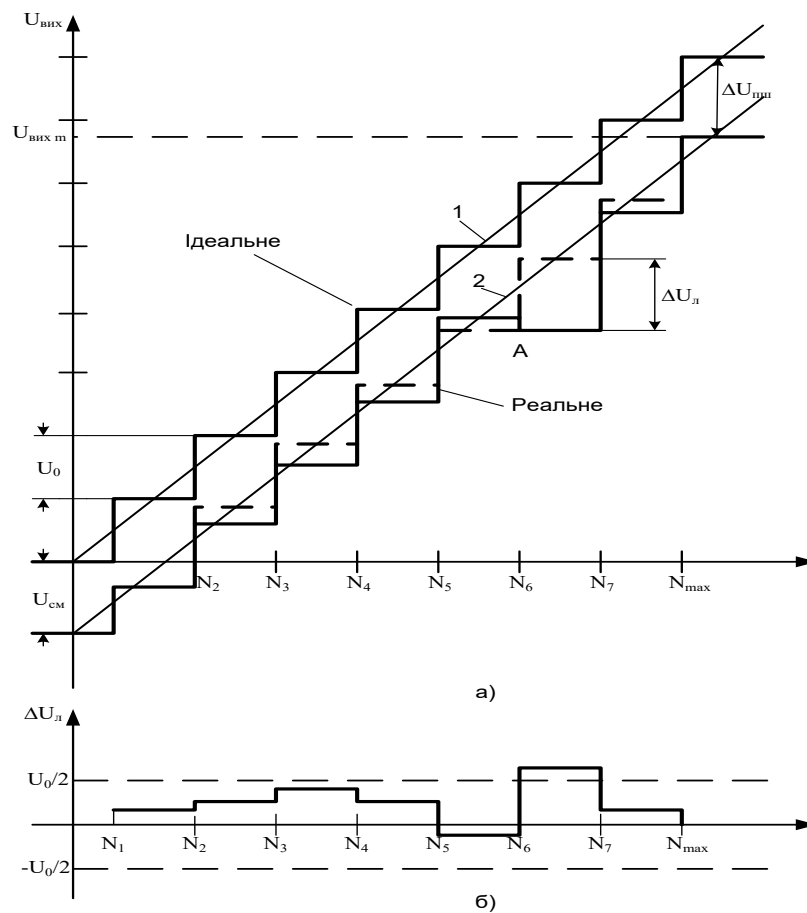


Рис. 7.3. Визначення основних статичних параметрів ПКН:

а) статичні характеристики перетворення ідеальної реальної РСН та основні похибки;

б) похибка нелінійності реальної РСН

В іншому випадку характеристики перетворення можуть бути немотонними. Це означає, що при подачі вхідного коду, що вимагає підвищення вихідної напруги, воно може впасти (ділянка А характеристики реального ПКН на рис. 7.3, а).

Загальна статична похибка PCN залежить в основному від елементів, розглянутих вище. У деяких випадках це можна зменшити шляхом компенсації похибок повної шкали та зміщення нуля. Це досягається зміною підсилення та балансування вихідного підсилювача перетворювача.

Серед динамічних параметрів ПКР найважливішими є час встановлення вихідної напруги та максимальна частота обробки.

Час встановлення вихідної напруги t_{set} — інтервал часу від моменту надходження вхідного коду до моменту встановлення вихідної напруги із заданою точністю.

Максимальна частота перетворення f_{max} — найбільша частота дискретизації (надходження вхідного коду), при якій параметри PCN відповідають заданим значенням.

Крім розглянутих характеристик ПКН, в довідкових даних наводяться параметри вхідних норм і граничні параметри експлуатації.

7.2.2. Класифікація ПКН.

Залежно від набору стандартів існує два методи перетворення коду в напругу:

- спосіб зведення окремих стандартів
- метод підсумовування з урахуванням маси розрядів.

У першому випадку при формуванні вихідної напруги використовується еталон тільки з одним вагою, рівним U_{ref} , а на вхід ПКН подається число - імпульсний код.

У другому випадку використовуються еталони, маси яких пропорційні 2^i , де $i = 0, 1, 2, \dots, n-1$, а на вхід ПКН надходить двійковий код позиції.

Залежно від типу вхідного коду PCN поділяються на *послідовні та паралельні*, які приймають послідовний та паралельний двійковий код відповідно.

За принципом дії розрізняють PCN *пряма трансформація* та PCN з *непрямою трансформацією*. У першому випадку вхідний код буде безпосеред-

ньо перетворений в напругу, у другому - спочатку в непрямий сигнал (кількість імпульсів, струм, інтервал часу), який потім буде перетворений в напругу.

В інтегральній схемотехніці найбільш поширеними є паралельні ПКН з прямим перетворенням і підсумовуванням еталонів з урахуванням маси розряду. Основною перевагою таких РСН є висока швидкість. Залежно від типу використовуваного джерела електричного сигналу вони поділяються на дві групи: ПКН зі стабілізацією напруги та ПКН зі стабілізацією струму.

Обидва можуть бути виконані на матриці резисторів з об'єднувальними опорами і на матриці резисторів $R - 2R$.

Перемикачі напруги та струму можуть використовуватися як ключові елементи в ПКН, але сучасні інтегровані передавачі використовують більш швидкодіючі перемикачі та перемикачі струму.

7.3. ПКН на резисторній матриці з ваговими резисторами

Функціональна схема ПКН наведена на рис. 7.4. Як джерело електричного сигналу він використовує стабілізоване джерело напруги U_e . Перетворювач складається з n перемикачів струму ($n_0 \dots n_{ni}$), матриці резисторів, опір яких "зважений" відповідно до двійкового закону $2^0 R$ у старшому $n - 1$ у старшому $2^{n-1} R$ у молодшому (нульовому) розряді та суматор струму на ОУ, який також перетворює вихідний струм на напругу.

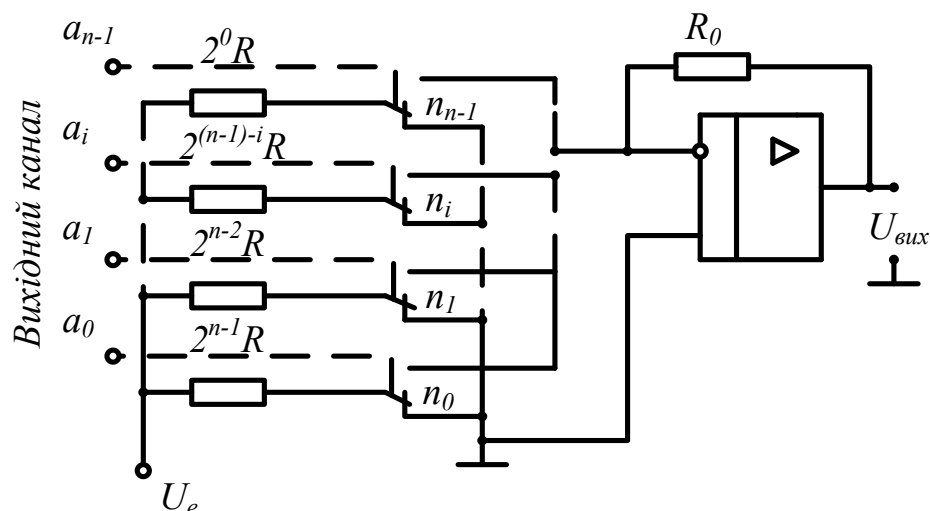


Рис. 7.4. ПКН на матриці резисторів з ваговим опором

На керуючі входи перемикачів подається вхідний код ($n_0 \dots n_{n-1}$), на комутаційні входи подається стабілізована напруга U_e (через відповідні масові резистори матриці).

За принципом дії ПКН схожий на аналоговий n - вхідний суматор на операційному підсилювачі. Коефіцієнт передачі для кожного входу визначається значенням масового резистора матриці, який підключений до зворотного входу ОУ через відповідний перемикач.

При нульовому значенні вхідного коду ($i_{i=0}$) перемикачі замикають усі вагові резистори на загальну шину: $U_{вих} = 0$.

у вхідному коді лише $a_i = 1$, то i -й ключ підключить до входу суматора важкий резистор номіналом $2^{(n-1)-i} R$, через який буде протікати струм $I_i = \frac{a_i U_e}{2^{(n-1)-i} R}$, при

$$\text{цьому } U_{вих_i} = I_i R_0 = \frac{a_i U_e R_0}{2^{(n-1)-i} R}.$$

У загальному випадку з будь-яким вхідним значенням коду

$$U_{вих} = \sum_{i=0}^{n-1} U_{вих_i} = \frac{U_e R_0}{2^{(n-1)-i} R} \sum_{i=0}^{n-1} a_i 2^i = \frac{U_e R_0}{2^{(n-1)-i} R} N.$$

З отриманого виразу видно, що вихідна напруга РКР пропорційна значенню вхідного коду N . Коефіцієнт пропорційності визначає ціну нижнього каскаду цього перетворювача і залежить від співвідношення опору резистора в зворотному зв'язку. шлейф ОУ і резистор максимального навантаження матриці: $U_0 = \frac{U_e R_0}{2^{n-1} R} (2^n - 1)$.

Якщо максимальний вхідний код отримано на вході РКН ($a_i = 1$ у всіх цифрах), тоді перемикачі підключають усі масові резистори до входу суматора. При цьому максимальне значення вихідної напруги

$$U_{\text{вих_макс}} = \frac{U_e R_0}{2^{(n-1)-i} R} (2^n - 1).$$

Основним недоліком РСН з ваговими резисторами є необхідність використання прецизійних резисторів з різними номіналами, максимальна різниця яких становить 2 (від R до $2^{n-1} R$). Коефіцієнт потужності, що розсіюється в опорах граничної ваги, також дорівнює 2^{n-1} . З таким широким діапазоном розсіюваних потужностей важко підтримувати суто двійкові зв'язки між резисторами. Тому РСН з опорами ваги в основному реалізуються на основі гібридних ІС.

В інтегрованих напівпровідникових РСН матриці з масовими опорами використовуються для перетворення відповідних груп кодових бітів і потім об'єднання їхніх вихідних сигналів за допомогою масштабуючих резисторів.

Рис. 7.5 показана схема 8-розрядної РСН, в якій перемикачі та резистори згруповані в квадрати з опорами вагових резисторів $R, 2R, 4R, 8R$ і максимальне співвідношення струму розряду 8:1.

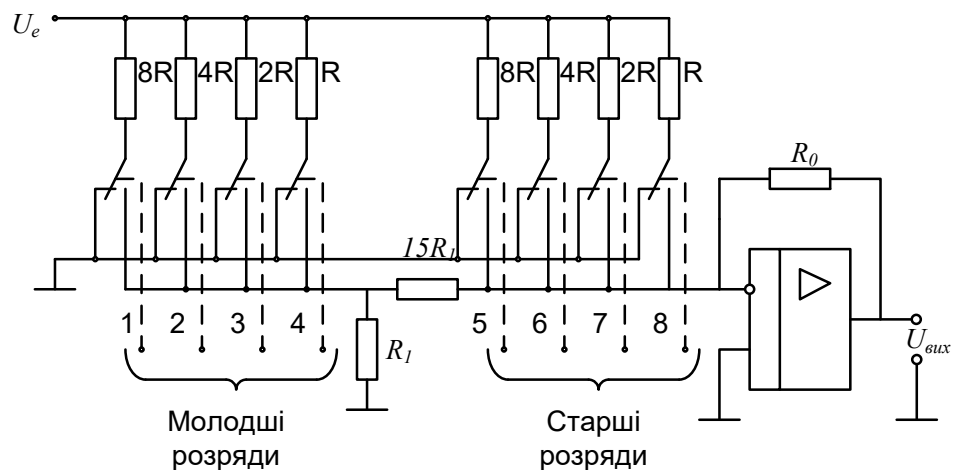


Рис. 7.5. ПКН на комбінованій матриці вагових резисторів і ваг

Струм перетворювача нижчого порядку поділяється на вихідний сигнал за допомогою резистивного дільника 16:1. У перетворювачі всього чотири резистора з різним масовим опором, що значно зменшує проблеми, пов'язані з узгодженням розрядних струмів. Для забезпечення однакової температурної залежності опорів навантажувальних резисторів і резистора зворотного зв'язку R_0 останній, як правило, виготовляють у складі матриці резисторів. Розглянутий принцип побудови матриці вагових резисторів широко використовується в інтегральних ЦАП, наприклад, в 12-розрядному ПКН К594ПА1.

7.4. ПКН на матриці резисторів R-2R

Функціональна схема ПКН наведена на рис. 7.6. Основою 4-розрядного РСН є резистивна матриця R-2R і ключі (в реальному випадку ключі побудовані на транзисторах). Права позиція клавiші відповідає одиниці в заданому розряді введеного коду N (цифри $D_0...D_3$). ОП може бути і вбудованим (ЦАП з виводом напруги), так і невбудованим (у ЦАП з виводом по струму).

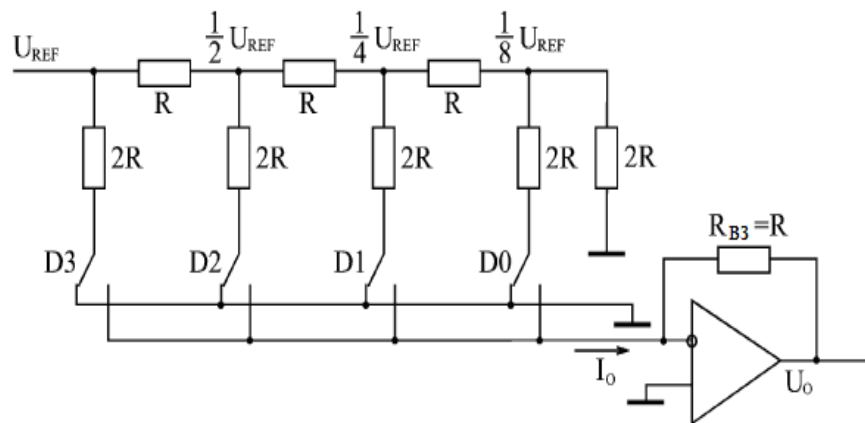


Рис. 7.6. ПКН на матриці резисторів P-2P

Перший ключ (лівий на рис. 7.6) перемикає струм зі значенням $U_{REF}/2R$, другий ключ - струм $U_{REF}/4R$, третій - струм $U_{REF}/8R$, четвертий - струм $U_{REF}/16R$. Це означає, що струми, комутовані сусідніми ключами, відрізняються вдвічі більше, ніж ваги цифр двійкового коду. Токи, поєднуються всіма перемикачами, сумуються та створюються у вивідну напругу операційним підсилювачем з $R_{B3} = R$ у колі негативного оберненого зв'язку.

При правильному стані кожної клавiшi (одиниць у необхідному розрядi ввiдного коду ЦАП) ток, комутується цiєю клавiшею, додається до суми. Коли клавiша знаходиться в iншому положеннi (0 у вибраному розрядi ввiдного коду ЦАП), струм, перемикається клавiшею, i підсумовування не виходить.

Це означає, що доля старшого ключа (старша цифра коду) у вивiдну напругу становить $U_{REF}/2$, другого - $U_{REF}/4$, третього - $U_{REF}/8$, четвертого - $U_{REF}/16$. Отже, при ввiдному кодi $N = 15U_{REF}/16 = 0000$ напруга на виходi схеми буде дорiвнювати нулю, а при ввiдному $N = 1111$ – максимуму.

У загальному випадку вивiдний сигнал ЦАП у $R_{B3} = R$ з ввiдним кодом N i сталою напругою U_{REF} обчислюється за формулою:

$U_{вих} = -NU_{REF}2^{-n}$, де n – декiлька бiт ввiдного коду. Вiд'ємне значення виходить шляхом iнвертування сигналу через операцiйний пiдсилювач. Цей зв'язок також можна проiлюструвати (див. таблицю 7.1)

Таблиця 7.1 – Перетворення ЦАП в уніполярному режимі

Код входу - N	Вихідна напруга U_0
000...000	0
000...001	$-2^{-n} U_{\text{ПОС}}$
...	...
100...000	$-2^{-1} U_{\text{НІ НІ}}$
...	...
111...111	$-(1-2^{-n}) U_{\text{ПОС}}$

Мікро схеми ЦАП можуть забезпечити можливість роботи в двохполярному режимі, при цьому вхідна напруга міняється не від 0 до U_{REF} , а від $-U_{REF}$ до $+U_{REF}$. Отже вивідний сигнал ЦАП U_0 множиться на 2 і зсувається на величину U_{REF} . Залежність між ввідним кодом N і вивідною напругою U_0 , $U_{\text{вих}} = U_{REF}(1 - N2^{1-n})$.

Більш детальний аналіз моделі наведено в таблиці 7.2. Таке біполярне перетворення з можливістю переміни знаку сталої напруги має назву квадратичним перемноженням (і стала напруга, і вивідна напруга може бути як позитивною, так і негативною).

7.5. Перетворювачі коду в часові рамки

Цифро-аналоговий перетворювач, в якому вихідний сигнал є часовим інтервалом, називається перетворювачем коду в час (СТС) або цифровим пристроєм затримки часу, керованим кодом. Завдання перетворення інтервалу часу в код зустрічаються у всіх типах контрольно-вимірювальних приладів і систем. Відомо багато конвертерів, які виконують таку операцію. Найпоширенішими були перетворювачі інтервалів коду в час на основі двійкового лічильника і комбінації двійкового лічильника і цифрового компаратора.

7.5.1 ПКВ на основі двійкового лічильника

У ПКВ можна використовувати додавання або віднімання лічильників із попередньо записаними введеними кодами. Функціональна схема ПКВ на основі сумарного двійкового лічильника наведена на рис. 7.7, а містить двійковий лічильник, тригер RS, логічний елемент і схему формування вихідного імпульсу. Перед початком перетворення в лічильник вводиться додатковий код N_d перетвореного числа $N_d = 2^n - N$, де n — кількість біт вхідного коду. Перетворення починається з надходженням імпульсу початку інтервалу U_n (рис. 7.7, б), який задає RS-тригер до єдиного стану.

Імпульси опорної частоти з перехідним періодом T_e надходять на логічний елемент I на вхід лічильника які зазвичай синхронізуються з імпульсами U_H . Коли надходить N імпульсів опорної частоти, лічильник переповнюється ($N_e + N = 2^n$). U_K створюється на основі сигналу переповнення з виходу пересилання або старшого розряду лічильника (рис. 7.7, б). Він затримується відносно імпульсу U_n на N періодів імпульсу опорної частоти. Отже, інтервал часу $T_n = T_e N$ пропорційний вхідному коду. Коефіцієнт пропорційності (ціна нижчого порядку) дорівнює періоду проходження опорних імпульсів T_e .

Фактичний інтервал часу, отриманий в результаті перетворення, відхиляється від ідеального через наявність інструментальної похибки

$$\Delta t = t_{z.le} + t_{уст.сч} + t_{z.f} + NT_e$$

де $t_{z.le}$ - середня затримка логічного елемента I ; $t_{параграфsch}$ — час встановлення лічильника; T_{zF} — затримка в колишній; NT_e - нестабільність періоду еталонної частоти.

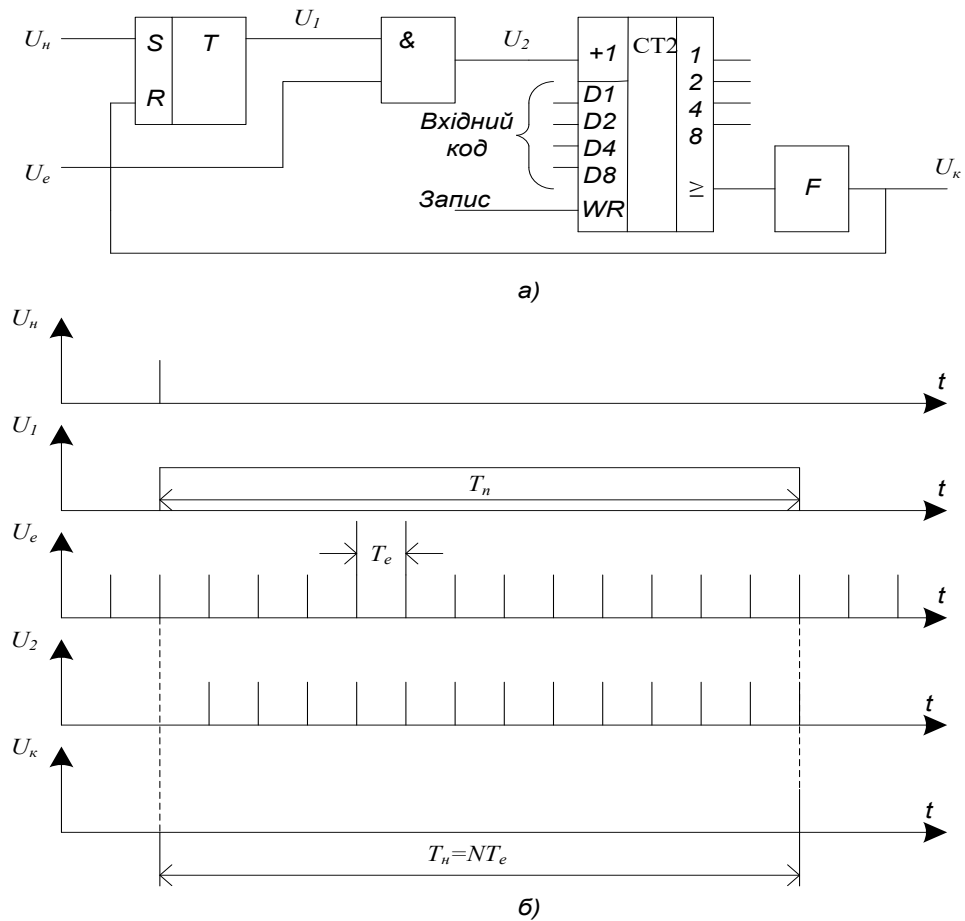


Рис. 7.7. PCV на основі лічильника підсумовування:

а) функціональна схема; б) часові діаграми

Усі компоненти повної похибки є випадковими. Сума перших трьох складових, при відомих середніх значеннях кожної з них, може бути компенсована затримкою імпульсу U_n про Δt час. Для зменшення останньої складової NT_e необхідно підвищити стабільність частоти (періоду) опорного генератора. Для цього використовуються генератори з кварцовою стабілізацією частоти. Операція PCV на основі віднімання відрізняється тим, що прямий код для числа N зберігається в лічильнику. Кінцева точка часового інтервалу визначається під час скидання лічильника.

7.5.2. ПКВ на основі лічильника та цифрового компаратора

Функціональна схема ПКВ наведена на рис. 7.8. Містить: Rigger RS -t , логічний елемент I , двійковий лічильник, цифровий компаратор і регістр.

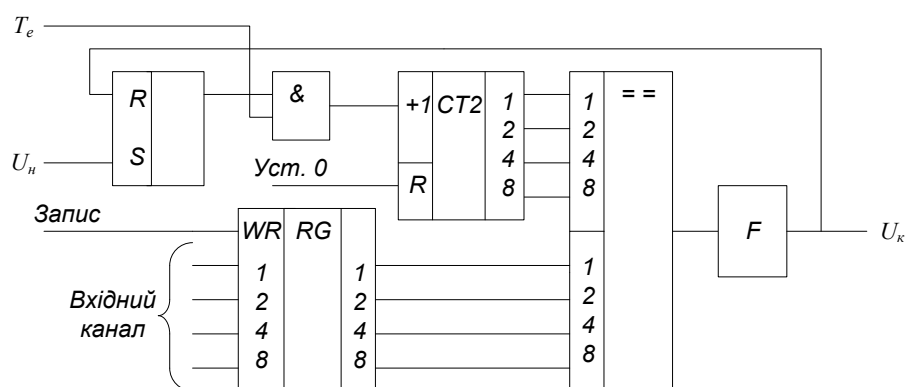


Рис. 7.8. РСВ на основі лічильника та цифрового компаратора

Перед початком перетворення лічильник встановлюється на нуль і вхідний код записується в регістр. За імпульсом початку інтервалу часу U_n RS- тригер встановлюється в одиничний стан, при цьому відкривається логічний елемент I і на лічильник надходять імпульси на опорній частоті. Код з виходу лічильника надходить на цифровий компаратор, де порівнюється з вхідним кодом. Якщо вони рівні, цифровий компаратор видає сигнал еквівалентності, з якого створюється імпульс кінця інтервалу часу. При тому ж імпульсі RS тригер встановлюється на нуль і підрахунок імпульсів опорної частоти припиняється. Інтервал часу між імпульсами U_k і $U_k T_n = NT_e$. Спосіб оцінки похибки перетворювача та способи її зменшення аналогічні розглянутим раніше для ПКВ на основі підсумовувального лічильника.

7.6 Особливості застосування DAC

У загальному вигляді мікросхему ЦАП можливо продемонструвати у вигляді блоку (рис. 7.9), який володіє кількома цифровими ввходів і один аналоговий ввід, і аналоговий вихід.

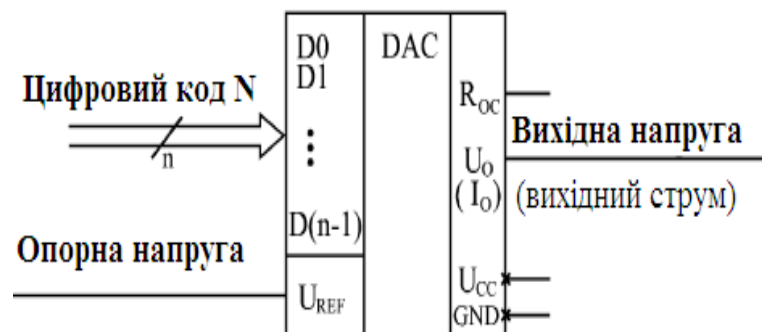


Рис. 7.9. Чіпсет DAC

код N , а на аналоговий вхід подається опорна напруга. U_{on} (інше поширене позначення U_{REF}). Вихідним сигналом є напруга $U_{вих}$ (інший символ U_o) або струм $I_{вих}$ (інший символ I_o). У цьому випадку вихідний струм або вихідна напруга відповідає ввідному коду та сталій напрузі. Для одних мікросхем стала напруга має певний рівень, для інших допускається зміна його значення в широкіх межах, а також зміна поляризації. ЦАП з великим діапазоном змін опорної напруги називається ЦАП з множенням, оскільки його можна використовувати для множення вхідного коду на сталу напругу.

За виключенням інформаційних імпульсів мікросхем ЦАП вимагають підключення одного або двох джерел живлення і загальної шини. Як правило, цифрові вводи ЦАП визначають сумісність зі визначеними виходами схем TTL.

Перш за все, якщо ЦАП має струмоподібний вивід, його вивідний струм перетворюється на вивідну напругу за допомогою зовнішнього операційного підсилювача та резистора, вбудованого в ЦАП R_{OC} , один із висновків якого підключений до зовнішнього висновку ЦАП. мікросхема (рис. 7.10). Тому,

якщо не зазначено інше, у майбутньому ми будемо вважати, що вихідний сигнал ЦАП є напругою U_o .

Принцип перетворення вхідного цифрового коду в аналоговий сигнал досить простий. Він забезпечується в сумуванні визначених струмів (за кількістю цифр ввідного коду), кожен з яких вдва рази більший за попередній. Для отримання визначених струмів застосовуються транзисторні джерела струму, чи резистивні матриці, які комутуються за допомогою транзисторних ключів.

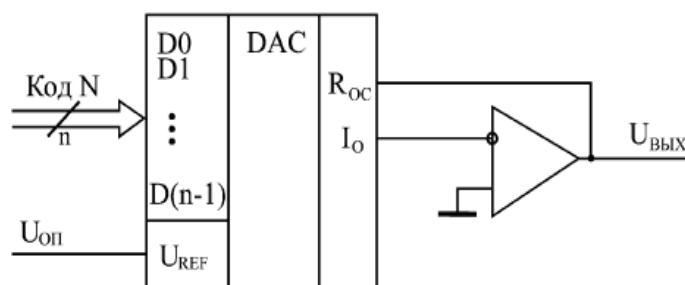


Рис. 7.10. Створення вивідного струму ЦАП у вивідну напругу

Промислові мікросхеми ЦАП поділяються кількістю розрядів (від 8 до 24), значенням затримки переробки (від декількох нс до одиниць мкс), допустимим значенням опорної напруги (зазвичай кілька вольт), значеннями похибок перетворення і деякі інші параметри. Також вони відрізняються технологією виробництва і особливостями внутрішньої будови, що іноді накладає обмеження на їх використання. Тому вибір мікросхеми ЦАП для конкретного застосування повинен здійснюватися на основі детальної довідкової інформації, наданої виробником.

Іноді необхідно зменшити кількість розрядів ЦАП. Для цього на необхідну кількість молодших розрядів ЦАП подаються сигнали логічного нуля. На малюнку 7.11 показано, як 10-розрядний ЦАП можна перетворити на 8-розрядний шляхом додавання нулів до двох нижніх цифр. Збільшення кількості бітів ЦАП є складнішим і вимагає створення складних аналогових схем.

Набагато простіше вибрати мікросхему з необхідною або більшою кількістю цифр.

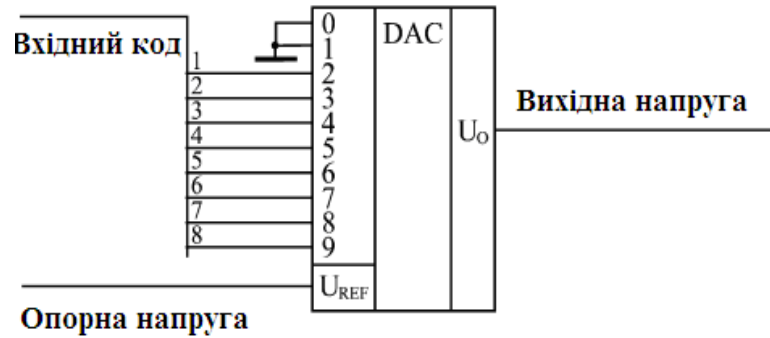


Рис. 7.11. Зниження швидкості передачі ЦАП

Основною областю застосування мікросхем ЦАП є отримання аналогового імпульсу з послідовності кодів (рис. 7.12).

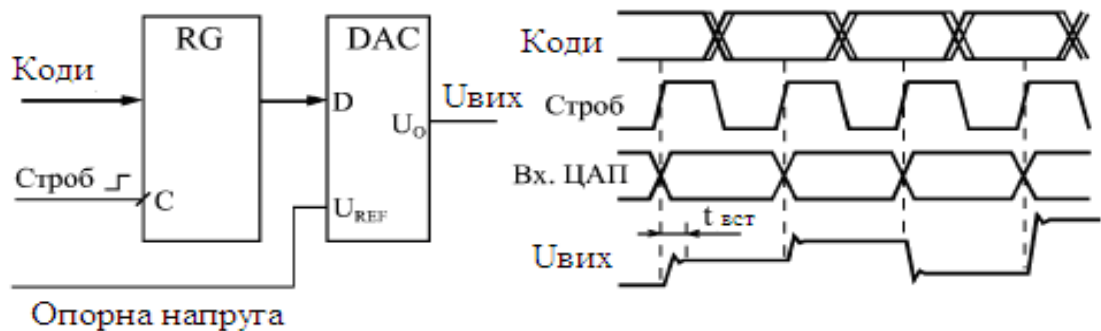


Рис. 7.12. Перетворення рядка кодів у вихідну напругу

Зазвичай, коди підключають на вводи ЦАП через код паралельного буфера, який дозволяє одночасно змінювати всі біти ввідного коду ЦАП. При одночасній зміні цифр ввідного коду на виході ЦАП виникають короткі імпульси напруги великої амплітуди, рівні яких не відповідають жодному з кодів.

Але навіть якщо всі цифри ввідного сигналу ЦАП змінюються одночасно, рівень напруги, що відповідає даному коду, визначається не зразу, а в процесі

установки ЦАП $t_{\text{вст}}$, що пов'язано з недосконалістю внутрішніх компонентів ЦАП. Вивідний струм перетворювача зазвичай виставляється досить швидко, ніж вихідна напруга, тому, що він не залежить від інерції самого підсилювача. Очевидно, що умовою коректної праці ЦАП є те, що час зберігання ввідного коду перевищує час установки ЦАП $t_{\text{вст}}$, інше вивідний код не зможе прийняти величину, що відповідає ввідному коду.

Якщо сигнали надходять на вхід ЦАП з великим інтервалом часу, то, як показано на рис. 7.13, схему можна використовувати, наприклад, в управляємому джерелі живлення, вивідна напруга, яка визначається ввідним кодом. Додатково необхідно використовувати зовнішній підсилювач струму для збільшення вихідного струму.

Змінюючи вхідний код ЦАП на високій частоті, можна отримати генератор (синтезатор) аналогових сигналів будь-якої форми. У цьому випадку коди, що досягають ЦАП, називаються зразковими кодами (миттєвими значеннями) згенерованого аналогового сигналу.

У найпростішому варіанті реалізації джерелом вхідних кодів ЦАП може бути звичайний двійковий лічильник (рис. 7.14). У той же час вихідна напруга ЦАП зростатиме на певну величину $2^{-n}U_{REF}$ з слідуючими тактовими імпульсами, створюючи пилоподібні вивідні імпульси з амплітудою U_{REF} . Тривалість кожного кроку рівний періоду стробуючого генератора T , а період всього вихідного сигналу $2^n T$. Кількість кроків за такт вивідного сигналу дорівнює 2^n . Якщо в цій схемі застосовуються синхронізовані лічильники з синхроперенесенням, то ввідний буфер ЦАП не потрібний, тому, що всі розряди лічильника будуть перемикатися одночасно. При використанні асинхронних лічильників або синхронних лічильників з асинхронним переносом потрібен вхідний регістр DAC.

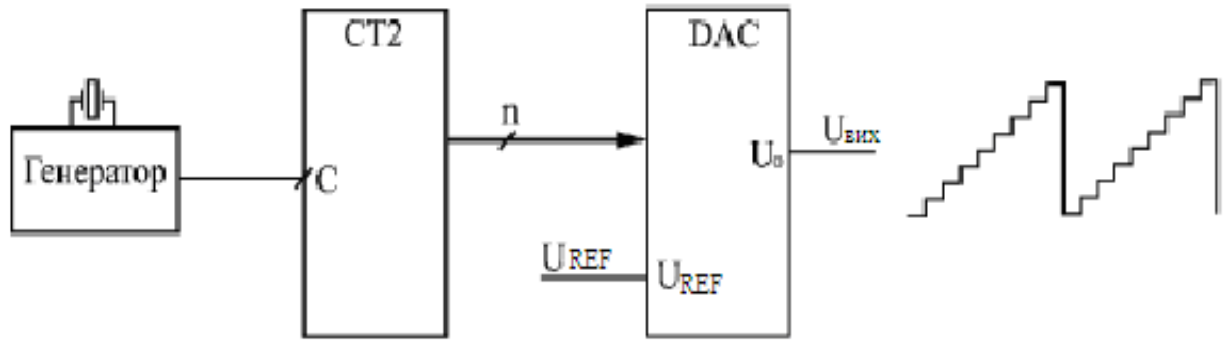


Рис. 7.13. Генератор пилоподібного аналогового сигналу

При необхідності формування аналогових сигналів будь-якої форми (синусоїда, Гауса, шум, трикутник, імпульс тощо) як джерело коду використовується елемент пам'яті, що робить по методу читання (рис. 6.9).

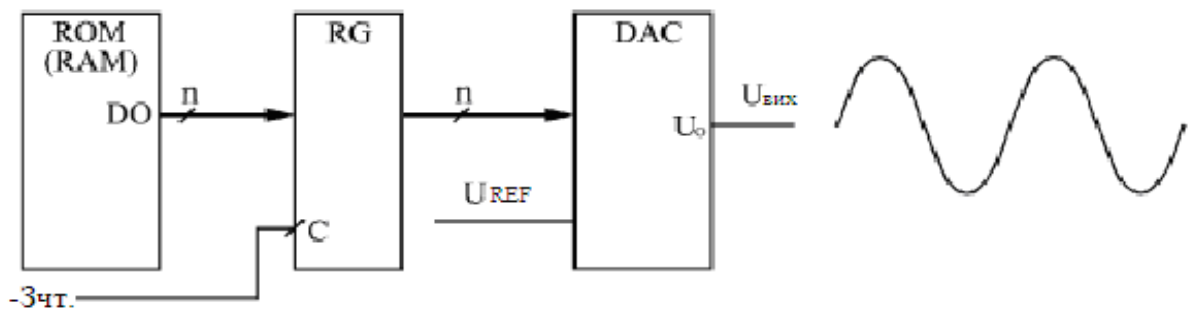


Рис. 7.14. Генерація сигналів довільної форми

Коли запам'ятовування постійне, то набір генерованих форм сигналу визначається один раз. При використанні оперативної пам'яті створюється односпрямований буфер періодичної інформації про режим роботи, що дозволяє записувати коди в пам'ять для генерації довільних сигналів. В обох випадках необхідний вхідний регістр ЦАП, інформація в який записується за допомогою стробоскопа, що зчитує пам'ять (рис. 7.15). Так як і в попередньому прикладі, вихідний сигнал ЦАП складатиметься зі ступенів, висота яких буде кратною $2^{-n}U_{REF}$. Амплітуда вихідного сигналу не перевищує U_{REF} . Якщо адреси пам'яті зчитуються лічильником, період вихідного аналогового сигналу дорівнює $2^m T$, де T - період тактового сигналу зчитування пам'яті "-Count", а m - кількість біт адреси пам'яті.

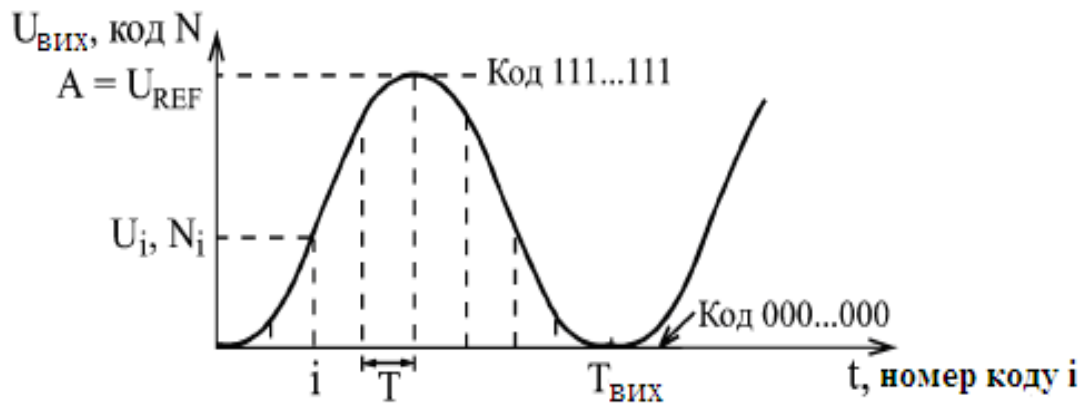


Рис. 7.15. Коды выборок периодического сигнала

Перекодування цифрових сигналів в аналогові сигнали - не одне застосування схем ЦАП. Їх також використовують для управляємої обробки аналогових імпульсів, як наприклад, для підсилення або зменшення аналогового сигналу в певну кількість разів. Найбільш придатними для цього є перемножуючі ЦАП, які дозволяють змінювати рівень сталої напруги в визначених межах.

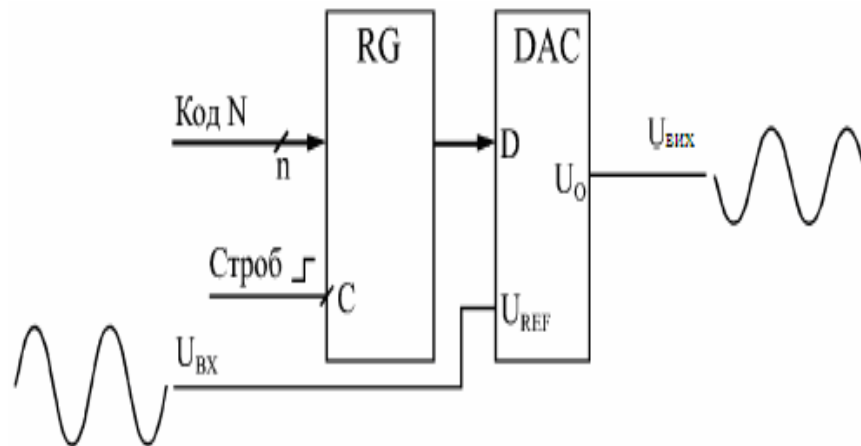


Рис. 7.16. Атенюатор аналогового сигналу в ЦАП

Найпростішою схемою є цифровий аттенюатор (аттенюатор) аналогового сигналу (рис. 7.16), який часто використовують для керування амплітуди вивідного імпульсу генерації на базі ЦАП.

Опорною напругою є змінний аналоговий сигнал (ЦАП - обов'язково помножувач). Вихідний сигнал пов'язаний із вхідним за формулою: $U_{\text{вх}} = -U_{\text{вх}} N 2^{-n}$. Це означає, що вихідний сигнал відповідає ввідному (з інверсією), а коефіцієнт пропорційності визначається вхідним цифровим кодом N . При цьому коефіцієнт пропорційності змінюється від нуля до майже одиниці з кроком 2^{-n} . У цьому випадку також необхідний вхідний регістр ЦАП, оскільки короткі імпульси значної амплітуди можуть накладатися на вихід ЦАП, коли вхідні кодові біти не перемикаються одночасно.

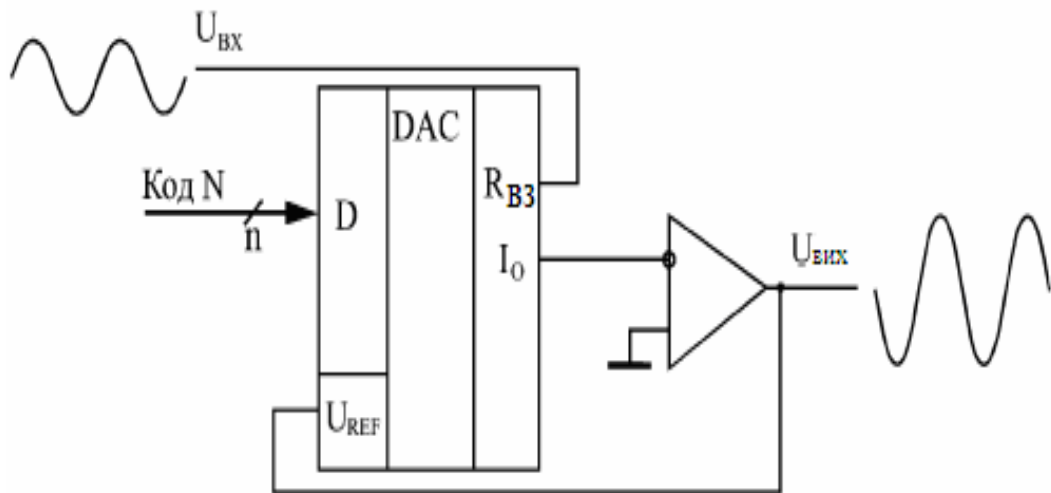


Рис. 7.17. Керований підсилювач вхідного сигналу

Існує також схема ЦАП, яка може служити керованим підсилювачем аналогового сигналу з коефіцієнтом посилення, заданим вхідним кодом N (рис. 7.17).

У цьому випадку вихідний струм ЦАП дорівнює $U_{\text{вх}} / R_{OC}$, а оскільки вихідна напруга служить опорною напругою, коефіцієнт пропорційності дорівнює: $U_{\text{вих}} = -U_{\text{вх}} N 2^{-n} / N$.

Вхідний код може змінюватися від одиниці до $(2^n - 1)$, що дорівнює приросту від 1 до 2^n . Як приклад, при використанні 10-розрядного перетворювача коефіцієнт перетворення пристрою сягає 1024. Як і в раніше, швидкість пере-

ключення ЦАП не важлива, тому, що коефіцієнт підсилення зазвичай не потрібно змінювати. часто. На схемі не показано вхідний регістр ЦАП, який необхідний для того, щоб всі біти вхідного коду перемикалися одночасно.

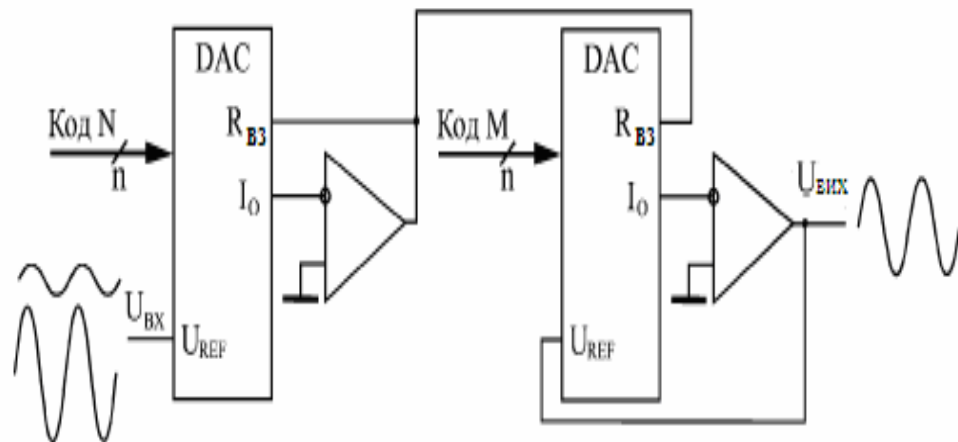


Рис. 7.18. Послідовне включення атенюатора і підсилювача

За допомогою послідовного з'єднання ланцюгів на рис. 7.16 і 7.17 можна довести вхідну напругу до стандартного рівня, який змінюється в досить широкі межі (рис. 7.18). Ця проблема часто виникає в аналого-цифрових системах.

Коефіцієнт всієї системи дорівнює співвідношенню вхідних кодів обох ЦАП *N/M перетворювачів* встановлюється з великою точністю в межах від 0 до 1, так і в діапазоні від 1 до $2n$ (підсилювач).

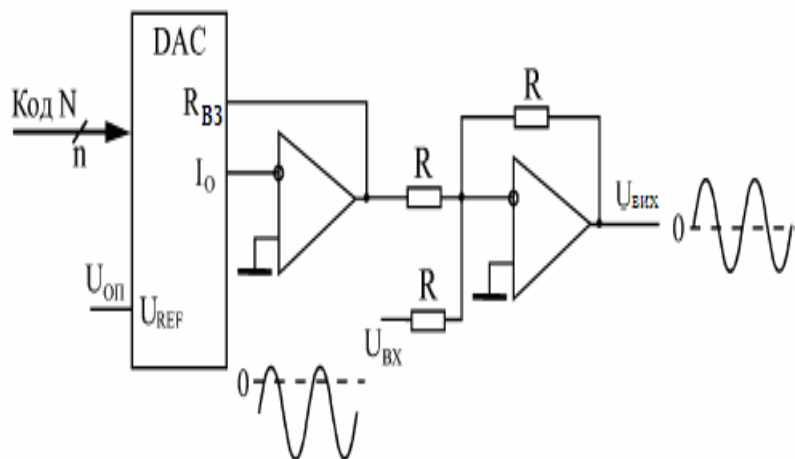


Рис. 7.19 Пристрій управління зсуву сигналу

Дана схема ЦАП, що розглядається, це схема, яка зсуває аналоговий сигнал на величину, визначену вхідним цифровим кодом. Зсув — це сума аналогового сигналу постійної напруги.

Схема комутації (рис. 7.19) містить цифровий код перетворювача вихідної напруги і аналоговий суматор на операційному підсилювачі. Значення напруги зсуву вхідного сигналу дорівнює $N2^{-n}U_{REF}$.

Тому, що використовуються два інвертуючих ОП, перетворення вхідного сигналу на виводі не буде. Якщо необхідно забезпечити можливість біполярного зсуву, необхідно використовувати цифро-аналоговий перетворювач з біполярним вихідним сигналом.

РОЗДІЛ 8. АНАЛОГО-ЦИФРОВІ ПЕРЕТВОРЮВАЧІ

8.1. Основні параметри та класифікація АЦП

Схеми аналого-цифрового перетворення виконують протилежну операцію ЦАП функцію - перетворюють ввідний неперервний сигнал в послідовність двійкових кодів. Схему аналого-цифрового перетворення можна продемонструвати у вигляді блоку з одним аналоговим входом, одним або двома входами для подачі опорної напруги, а також цифровими виходами для передачі коду, відповідного значенню аналогового сигналу (рис. 8.1).

Часто мікросхема АЦП також має вхід, що забезпечує тактовий сигнал, імпульси розрешення CS і сигнал, що вказує на можливість виведення цифрового коду RDY. На схему подається одна чи дві напруги живлення і загальна шина. Отже, схеми АЦП складніші, ніж схеми ЦАП, їх різновидів помітно більше, і для них складно створити одні правила.

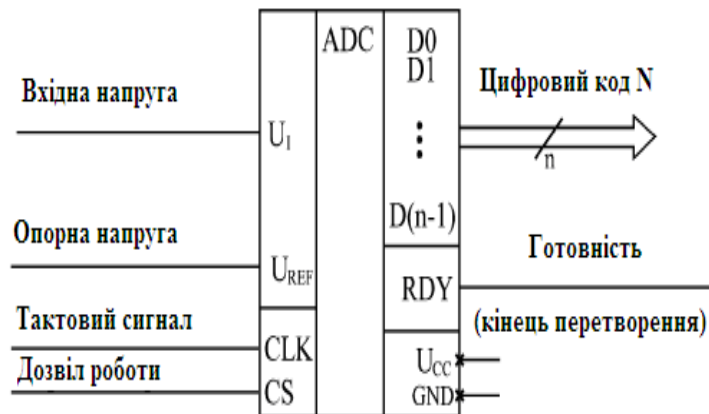


Рис. 8.1. Чіпсет АЦП

Опорна напруга АЦП визначає діапазон вхідної напруги, у якому відбувається перетворення. Він може бути постійним або змінюватися в певних межах. Іноді до АЦП подають дві опорні напруги з різними знаками, тоді АЦП

може працювати з біполярними вхідними напругами. Роботою АЦП керує тактовий сигнал CLK, який задає частоту перетворення. Межа тактової частоти є другим важливим параметром АЦП. Деякі мікросхеми мають вбудований генератор тактових імпульсів, частота яких задається зовнішнім кварцовим резонатором. Сигнал CS дозволяє чіпу працювати.

АЦП являє собою кодований перетворювач, на вхід якого надходить аналоговий сигнал x_{in} , а на виході знімається цифровий код N , який пов'язаний з вхідним сигналом співвідношенням $N = \frac{x_{in}}{\Delta x}$, де Δx - ступінь квантування перетворювача.

Серед АЦП найбільшою популярністю користуються перетворювачі, в яких вхідним аналоговим сигналом є напруга, а вихідним цифровим сигналом є двійковий код. Такі перетворювачі називаються перетворювачами напруги в код (ПНК).

Відповідно до узагальненої схеми АЦП в ПНК вхідна напруга дискретизується і квантується, а результат перетворення кодується. При цьому вхідна напруга $U_{вх}$ підлаштовується під цифровий двійковий код $N = \frac{U_{вх}}{\Delta U_{kv}}$, де величина ΔU_{kv} - ступінь (квант) квантування n -розрядного перетворювача.

Вивідний двійковий код N (n -біт) беззаперечно відповідає значенню ввідної напруги. Цифровий код може сприймати 2^n значення, отже АЦП може виділяти 2^n рівні ввідної напруги. Скільки бітів вивідного коду n є важливим аспектом АЦП. Коли створення вивідного цифрового коду завершено, генерується імпульс завершення кодування RDY, за допомогою якого наступна схема може зчитувати код N .

Статична характеристика квантування ПНК $N = f(U_{вх})$ (рис. 8.2.) подібна до характеристики перетворення ПНК $U_{вих} = f(N)$. Характеристика квантування (рис. 1) також визначає основні статичні параметри ЧПН, але по відношенню до ввідної напруги $U_{вх}$.

Основними характеристиками ПНК є: роздільна здатність, точність і швидкість.

Роздільна здатність ПНК залежить від кількості бітів вихідного коду. Абсолютне значення роздільної здатності ПНС дорівнює значенню одного етапу квантування, яке тісно пов'язане з діапазоном зміни вхідної напруги та кількістю бітів n вихідного коду:

$$\Delta U_{\text{кв}} = \frac{U_{\text{вх_макс}} - U_{\text{вх_мін}}}{2^n}$$

де $U_{\text{вх_макс}}$ і $U_{\text{вх_мін}}$ - максимальне і мінімальне значення вхідної напруги.

Крок квантування характеризує потенційну роздільну здатність перетворювача за рівнем, тобто найменше помітне збільшення вхідного значення.

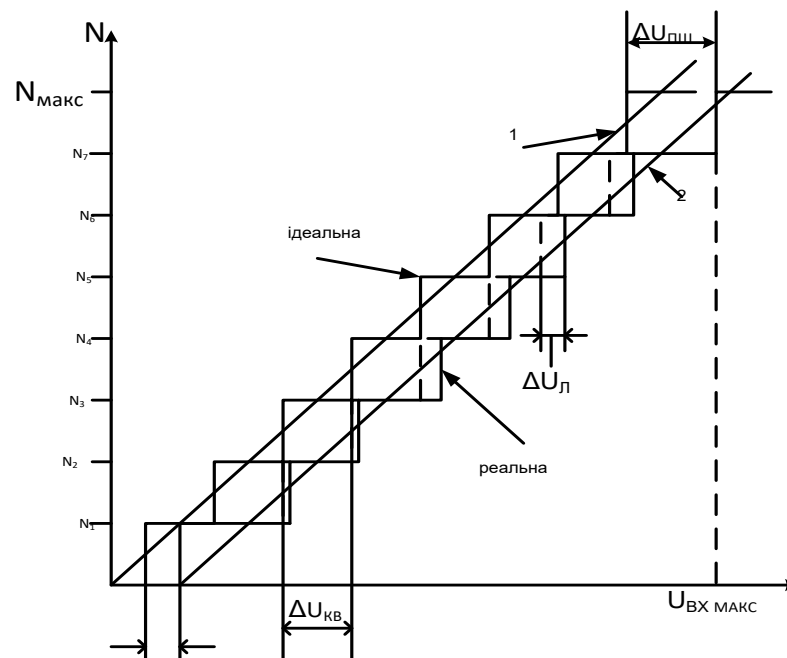


Рис. 8.2. Статичні характеристики ПНК квантування

Етап квантування також визначає методичну похибку аналого-цифрового перетворювача, яка виникає внаслідок заміни безперервного аналогового значення квантованими значеннями (найближчими рівнями квантування).

Крім методологічної похибки квантування, статична точність ПНС характеризується також загальною інструментальною похибкою всіх елементів, що входять до складу перетворювача.

Основними компонентами сумарної інструментальної похибки є:

- абсолютна похибка перетворення в кінцевій точці шкали ΔU_{nc} (повна похибка);
- помилка зміщення нуля на вході U_{var} ;
- диференціальна похибка ΔU_{ld} ;
- похибка нелінійності ΔU .

Зазначені інструментальні складові похибки визначаються статичною характеристикою квантування ПНК $N = f(U_{BX})$ (рис. 8.2).

Абсолютною похибкою перерахунку в кінцевій точці шкали $\Delta U_{пш}$ є відхилення значення вхідної напруги від номінального при максимальному вихідному коді N_{max} (рис. 8.2). Абсолютна повна похибка в еталонних даних надається в одиницях кроку квантування

Похибка зсуву нуля на вході U_{zm} – прикладена до входу напруги відповідає відхиленню початкової ділянки характеристики квантування від номінального значення (рис. 8.2).

Диференціальна нелінійність ΔU_{ld} – це відхилення різниці значень двох вхідних напруг, що відповідають середнім значенням двох сусідніх рівнів квантування, від значення кроку квантування.

Похибка нелінійності ΔU — це максимальне відхилення фактичної характеристики квантування від лінії абсолютної точності 2 (рис. 8.2). Абсолютне значення похибки нелінійності вимірювання в одиницях кроку квантування: $\Delta_{nl} = \frac{\Delta U_{nl}}{\Delta U_{кв}}$. Похибка нелінійності інтегруючих ПНК не повинна перевищувати допустимих меж $\pm \frac{\Delta U_{кв}}{2}$.

Загальну статичну похибку перетворювача неможливо визначити кількісно, оскільки вона залежить від багатьох факторів, на які впливають як час, так і температура.

Швидкість ПНК характеризується динамічними параметрами; час перетворення $t_{пр}$ і максимальна частота перетворення F_m .

Час перетворення $t_{пр}$ інтервал часу від моменту конкретної зміни вхідного сигналу до появи на виході відповідного стабільного коду.

Вибір часу перетворення пов'язаний зі швидкістю зміни вхідного сигналу. Якщо потрібно, щоб протягом часу t_{np} вхідна напруга змінювалася не більше ніж на ступінь квантування ΔU_{kv} , то: $t_{np} \leq \frac{\Delta U_{kv}}{U_{вх}}$, де $U_{вх}$ – максимальна швидкість зміни вхідної напруги.

Якщо ця вимога не виконується, пристрій вибірки та запам'ятовування слід розмістити на вході ПНС, який зберігає зразки вхідної напруги в моменти дискретизації для часу перетворення.

Максимальна частота перетворення F_m – найбільша з них, при якій параметри ПНК відповідають заданим значенням. Максимальна частота перетворення вибирається залежно від ширини спектра вхідної напруги та необхідних показників якості системи, до складу якої входить ЧПУ.

Крім розглянутих характеристик і параметрів, існують параметри, що визначають зовнішні умови роботи перетворювача, такі як:

- параметри вихідного коду;
- динамічний діапазон вхідної напруги;
- параметри джерел опорної напруги та джерела живлення;
- граничні робочі параметри та інші.

Класифікація ПНК. Основною ознакою класифікації ПНК є алгоритм її роботи в процесі перетворення. Алгоритм відображає набір операцій, які встановлюють числову відповідність між вхідним аналоговим значенням і вихідним цифровим кодом.

Залежно від типу алгоритму перетворення ПНС діляться на три основних типи:

- інший рахунок;
- порозрядне кодування (послідовне наближення);
- паралельно (читання).

За принципом перетворення ПНК поділяються на перетворювачі прямої дії і перетворювачі непрямої дії. У ПНС прямої дії вхідна напруга перетворю-

ється безпосередньо в цифровий код. У РНС з непрямим перетворенням напруга спочатку перетворюється в інше аналогове значення (інтервал часу, частоту тощо), яке потім перетворюється в двійковий код.

8.2. Перетворювачі напруги в послідовний код підрахунку

У РНС послідовного лічильника вхідна напруга врівноважується сумою однакових і мінімальних стандартів, рівних кроку квантування. Підраховується кількість еталонів, використаних для балансування, і компаратором визначається момент, коли вхідна напруга дорівнює сумі еталонів. Результатом перетворення є цифровий код лічильника.

Серед РНС серійного рахунку найбільш популярними стали перетворювачі, в яких опорні напруги отримують за допомогою цифро-аналогового перетворювача, розташованого в ланцюзі зворотного зв'язку. До них відносяться: РНС з поступовим розвитком напруги і стеження РНС.

8.2.1. ПНК з поступово розвивається напругою

Функціональна схема ПНК і тимчасові схеми, що пояснюють принцип дії, наведені на рис. 8.3, а Імпульс запуску T_c , який визначає початок циклу перетворення, встановлює тригер в єдиний стан. Рівень логічної одиниці з виходу тригера дозволяє імпульсам на опорній частоті проходити через логічний елемент І на вхід лічильника. Послідовно зростаючий код із виходу лічильника перетворюється на крокову напругу, яка змінюється за лінійним законом, за допомогою РСН. Розмір кроку дорівнює кроку квантування ΔU_{kv} . Коли вхідна напруга РKN перевищує вхідну U_{BX} , компаратор збуджується, встановлює тригер RS на нуль і блокує перехід опорних імпульсів на вхід лічильника. Кількість імпульсів N, зареєстрованих у лічильнику, дорівнюватиме кількості квантів U_{kv} , необхідних для балансування вхідної напруги $N = \frac{U_{ax}}{\Delta U_{kv}}$:

Перед початком наступного циклу перетворення код видається споживачеві, а лічильник скидається.

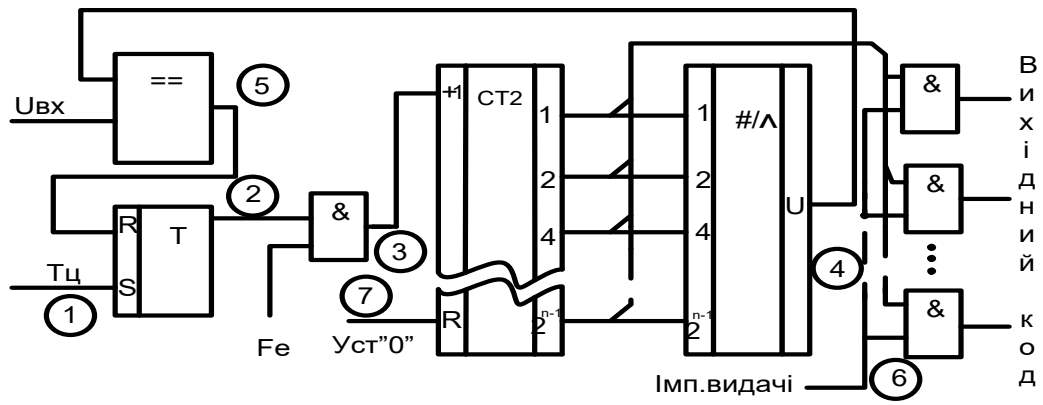
Сумарна похибка перетворювача складається з методичної та інструментальної складових. Методологічний елемент характеризується похибкою квантування і є функцією кількості бітів у вихідному коді. Максимальна похибка квантування дорівнює кроку квантування ΔU_{sq} .

Основними джерелами інструментальних похибок є компаратор і РСН. У гіршому випадку $\Delta U_{i_макс} = \Delta U_{комп} + \Delta U_{кв_пкн}$ загальна інструментальна похибка є сумарною похибкою датчика $\Delta U_{пнк} = \Delta U_{кв} + \Delta U_{комп} + \Delta U_{пкн}$.

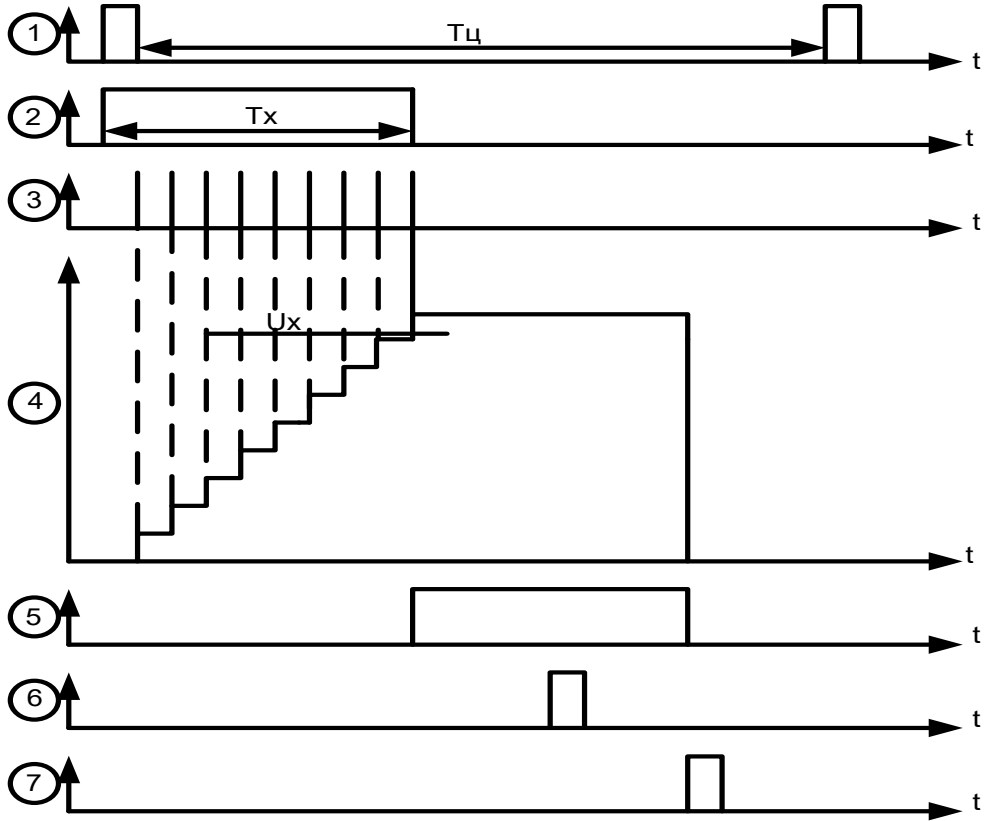
Інструментальні похибки ΔU_{comp} і ΔU_{pcn} містять систематичні та випадкові складові. Систематичні елементи можна компенсувати регулюванням перетворювальних елементів. При оцінці параметрів ПНК слід враховувати випадкові складові (дрейф, шум тощо).

Інструментальні похибки ПНК характеризуються такими основними параметрами, як похибка нелінійності $\Delta U_{л}$, повна похибка $\Delta U_{пш}$ і похибка зсуву нуля $\Delta U_{зм}$.

Час перетворення t_{PNC} послідовного рахунку непостійний і залежить від U_{BX} . Період Tc вибирається з умови $Tc > t_{pr.max} + t_{пнк}$, де $t_{пнк}$ – час, необхідний для видачі коду та встановлення елементів перетворювача у вихідний стан.



a)



б)

Рис. 8.3. ПНК зі ступінчастою розгортанням напруги

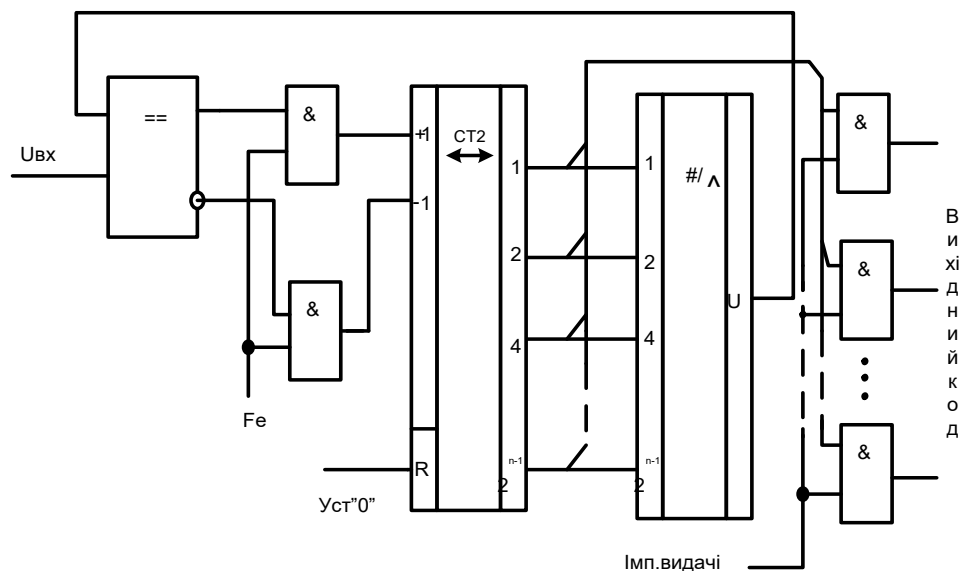
Перевагою ПНК послідовного підрахунку є відносна простота і висока точність, основним недоліком є низька швидкість.

Основні області їх застосування - цифрові вольтметри і радіотехнічні системи, призначені для роботи з постійною і повільно змінною напругою. ПНК послідовного числення, як правило, не виготовляються у вигляді закінчених інтегральних схем. Їх легко реалізувати на основі інтегрованих типових вузлів: ЦАП, компараторів, лічильників і т.д.

8.2.2. Тип відстеження ПНК.

Послідовний лічильник РНК зі ступінчастою напругою розсіювання можна легко перетворити на РНК слідкуючого типу . Для цього замініть підсумковий лічильник на реверсивний і за допомогою прямого та інверсного виходів компаратора керуйте режимом підрахунку. Функціональна схема відстеження ПНК наведена на рисунку 8.4.

На входи компаратора подається вхідна напруга $U_{вх}$ і вихідна напруга ПНК $U_{рkn}$. Залежно від знака різниці між $U_{вх}$ і $U_{рkn}$ на вхід додавання або віднімання лічильника надходять імпульси з опорною частотою. Якщо $U_{вх} > U_{рkn}$, вміст лічильника збільшиться, а якщо $U_{вх} < U_{рkn}$ – зменшиться. Після досягнення рівності вхідної напруги та напруги зворотного зв'язку з виходу РНК в системі стеження встановлюється динамічна рівновага, і вихідний код РНК буде коливатися навколо середнього значення, відповідного вхідній напрузі з точністю одиниці нижчого порядку, (Рис. 8.4, б.) Якщо в цьому стані вхідна напруга починає змінюватися, вихідний код перетворювача буде відслідковувати це з помилкою, що не перевищує одиниці нижчого порядку,



а)

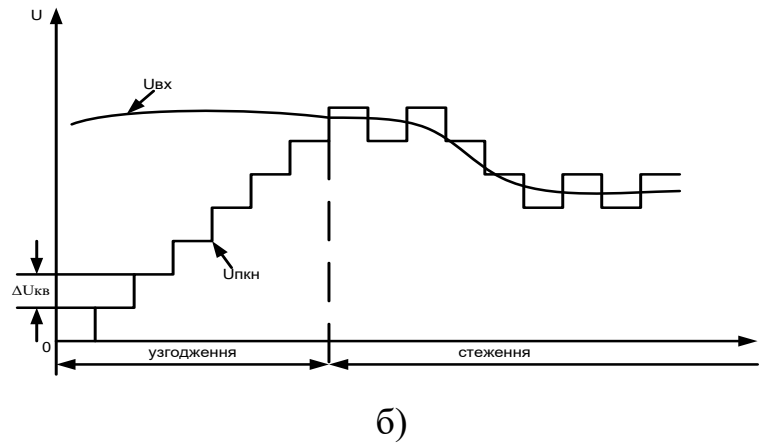


Рис. 8.4. Тип відстеження ПНК

Якщо задане співвідношення виконується, час перетворення повторювача ПНК дорівнює одному тактовому інтервалу TE .

Недоліком найпростішого трасувальника ПНК (рис. 8.4) є тривалий початковий час узгодження значень $U_{рпн}$ і $U_{вкх}$, оскільки за цей час $U_{рпн}$ змінюється від нуля до $U_{вкх}$, а перетворювач працює як номер ПНК серії з градуйованим відмотуванням. напруги (рис. 8.3).

8.3. Побітові перетворювачі напруги в код

Побітове кодування ПНК є одним із найпоширеніших перетворювачів. Його робота заснована на виконанні послідовних порівнянь вхідної напруги з набором n еталонів, значення яких пропорційні степеню 2.

Цей процес аналогічний процесу зважування, у якому невідоме вхідне значення врівноважується еталоном, створеним із набору стандартів.

Балансування починається зі стандарту з максимальним значенням $U_e / 2^1$, де $U_{et} = U_{in.max}$ - максимально можлива вхідна напруга ПНК.

Якщо $U_{вх} < U_e / 2^1$, то в старшому розряді вихідного коду перетворювача встановлюється нуль, а потім вхідне значення $U_{вх}$ балансується наступним шаблоном у значенні $U_e / 2^2$. Якщо $U_{вх} > U_e / 2^1$, то одиниця фіксується в старшому розряді вихідного коду і тоді вхідна напруга врівноважується сумою еталонів $U_e / 2^1 + U_e / 2^2$. Подібні операції виконуються при визначенні значення всіх n бітів вихідного коду.

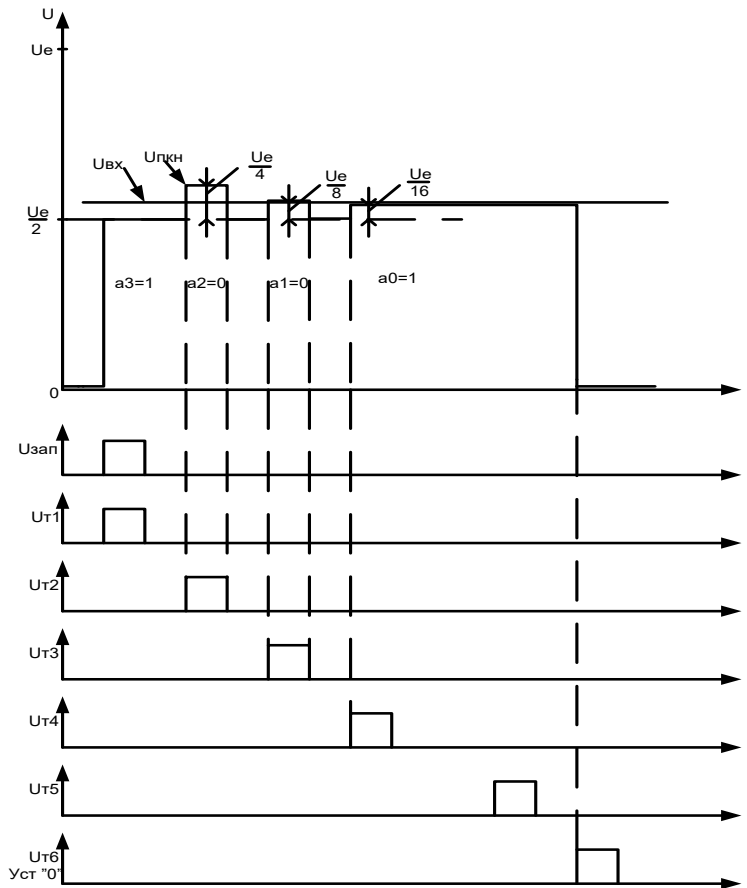


Рис. 8.6. Тимчасові діаграми ПНК з ПНК у циклі зворотного зв'язку

Цикл перетворення починається з надходженням стартового імпульсу $U_{зап}$ і збігається з ним першого тактового імпульсу $U_{Т1}$. Уздовж фронту імпульсу $U_{Тi}$ одиниця записується на тригер вищого порядку регістра пам'яті комп'ютера, який подається на вхід вищого порядку РСН.

На виході ПКН формується напруга $U_{ПКН} = U_e / 2$. У компараторі воно порівнюється з вхідною напругою U_{in} , і якщо $U_{in} > U_e / 2$, то вихідна напруга компаратора $U_K = U^0$.

Логічна одиниця записується в першому розряді регістра зсуву відповідно до фрагмента тактового імпульсу $U_{Т1}$. Він надходить на вхід першого логічного елемента І, на другий вхід якого подається напруга логічного нуля з виходу компаратора ($U_{до} = U^0$). При цьому вихідна напруга логічного елемента І дорівнює U^0 , а логічна одиниця фіксується в тригері старшого розряду (СЗР) регістра пам'яті (значення старшого розряду вихідний код $i_3 = 1$). Якщо

$U_{вх} < U$ дорівнює $U/2$, то $U_{к} = U'$ і після закінчення циклу $U_{т1}$ на виході першого логічного елемента І з'явиться логічна одиниця, яка встановить тригер старшого біта реєстр пам'яті в нуль і встановити старшу значущу цифру коду в нуль джерела ($a_z = 0$).

На фронті наступного тактового імпульсу $U_{т2}$ логічна одиниця записується до другого тригера реєстра пам'яті.

При цьому вихідна напруга $U_{пнк} = a_3 \frac{U_e}{2} + \frac{U_e}{4} \text{ПКН}$

Компаратор порівнює цю напругу з $U_{вх}$. Якщо $U_{ін} > U_{пкн}$, то логічна одиниця встановлюється у другому тригері реєстра пам'яті, тобто $a_2 = 1$. Коли $U_{ін} < U_{пкн}$, тригер встановлюється на нуль ($a_2 = 0$). На наступних двох етапах перетворення (через $UT3$ і $UT4$) аналогічним чином визначаються значення вихідного коду a_1 і a_0 . При цьому вхідна напруга послідовно врівноважується напругою з виходу ПКН системи з похибкою, що не перевищує крок квантування $U_e/16$. Цикл перетворення закінчується, і споживач отримує вихідний код $a_3 a_2 a_1 a_0$, відповідний вхідній напрузі U_x на вихідному імпульсі $U_{т5}$. Перед початком нового циклу перетворення реєстр зсуву та реєстр пам'яті скидаються за допомогою імпульсу $U_{т6}$ (встановлюється на «0»). Найпоширенішим типом ПНС є перетворювачі, побудовані за допомогою побітового кодування з ПНС у ланцюзі зворотного зв'язку.

Серійно випускаються спеціалізовані інтегральні схеми - перетворювачі напруги для порозрядного кодування К572ПВ1, КИ08ПВ1, КШЗПВ1 та інші. Наприклад, ІМС К572ПВ1 — це цифро-аналоговий перетворювач зі схемою керування та логічним пристроєм, що містить реєстр послідовних наближень. Після підключення компаратора ІС К572ПВ1 він може реалізувати функції 12-розрядного кодування ПНК bit-by-bit.

8.4. АЦП паралельного типу

Усі біти вихідного коду обчислюються одночасно (паралельно), що робить їх швидшими, ніж послідовні АЦП. Але при такому робочому алгоритмі потрібна більша кількість компараторів ($2^n - 1$), що визиває суто технологічні проблеми при більшій кількості розрядів (приклад, при 11-розрядному перетворювачі потрібно 2048 компараторів).

У схему АЦП (мал. 8.7) входить резистивний дільник з 2^n таких же резисторів, який розділяє опорну напругу на $(2^n - 1)$ рівні.

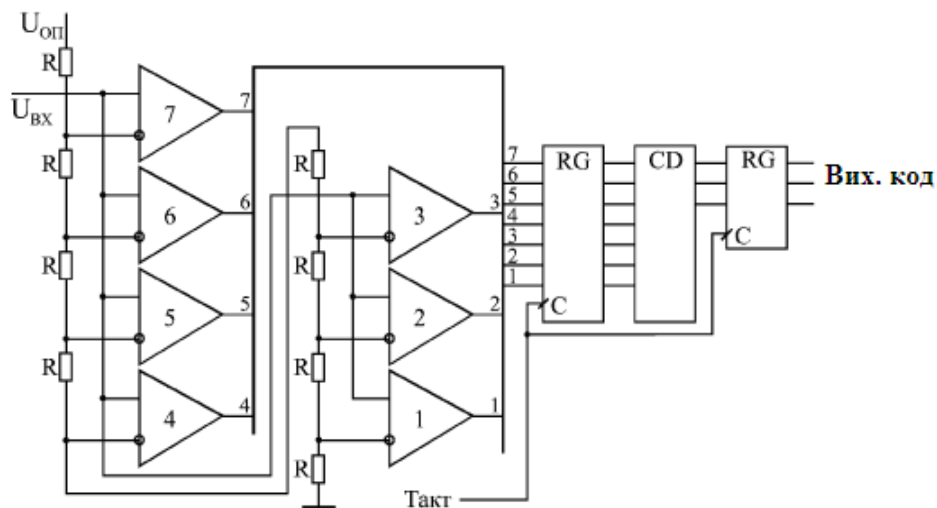


Рис. 8.7. АЦП паралельного типу

Ввідний сигнал зрівнюється за допомогою компараторів з рівнями, створеними резисторними дільниками напруги. Вивідні імпульси компаратора перетворюються в n -розрядний двійковий код за допомогою кодера. Кодер видає номер останнього спрацьованого компаратора (який подав сигнал логічній одиниці). Наприклад, для 3-розрядного АЦП (рис. 8.7), коли вхідна напруга становить від 0 до одної восьмої опорної напруги, вивідний код буде 000, коли вхідна напруга становить від $1/8$ до $2/8$ від еталонної напруги, буде працювати 1 компаратор і схема генеруватиме вивідний код 001, при вхідній напрузі від двох восьмих до трьох восьмих еталонної напруги будуть працювати порівнювачі 1 і 2, а вихідний код буде 010. Перетворення в паралельному АЦП дуже

швидке, тому частота перетворення досягає сотні мегагерц. Для підвищення швидкості паралельного АЦП іноді застосовується контейнерний принцип: вивідний сигнал компараторів записується в $(2^n - 1)$ -розрядний паралельний регістр. Вихідний код шифру також зберігається в n -розрядному паралельному регістрі. Одночасно два регістри керуються одним сигналом керування. Ця дія знижує вимоги до швидкості компараторів і кодера. Однак вивідний сигнал АЦП затримується на 2 такти частотисинхронізації. Величезність конструкції паралельного АЦП призводить до того, що в деяких АЦП використовується комбінований паралельно-послідовний принцип, який трохи знижує швидкість АЦП порівняно зі любым паралельно працюючим АЦП, але в той же час дозволяє використовувати більшу кількість бітів без збільшення числа компараторів до $(2^n - 1)$.

Щоб будь-який тип АЦП працював на повну потужність, слід подбати про те, щоб діапазон зміни аналогового вхідного сигналу відповідав допустимому діапазону (динамічному діапазону) вхідної напруги АЦП.

На малюнку 8.8 показано чотири можливі випадки залежності між динамічним діапазоном АЦП (від 0 до U_{REF} або від U_{REF1} до U_{REF2}) та вхідним сигналом. У випадках а) і б) вхідний сигнал менший за діапазон, тому АЦП функціонуватиме правильно, але не використовуватиме всі свої можливості. Випадок с) виникає, коли амплітуда вхідного сигналу занадто велика, тому частина його значення не буде оброблена. Тільки у випадку d) АЦП буде працювати як n -розрядний і перетворювати всі значення вхідного сигналу. Для узгодження вхідного сигналу з динамічним діапазоном АЦП можна використовувати підсилювачі, зменшувачі та пристрої зсуву. У визначених випадках погодження буде досягнуто простим перебором значення опорної напруги.

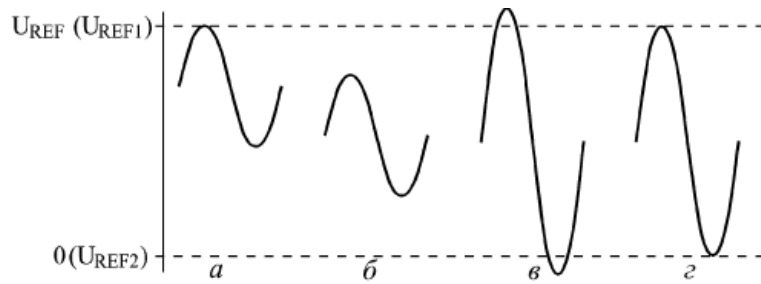


Рис. 8.8. Співвідношення вхідного сигналу та динамічного діапазону АЦП

При необхідності можна зменшити кількість вихідних бітів АЦП, відключивши необхідну кількість молодших вихідних бітів. На рис. 8.9 показано використання АЦП.

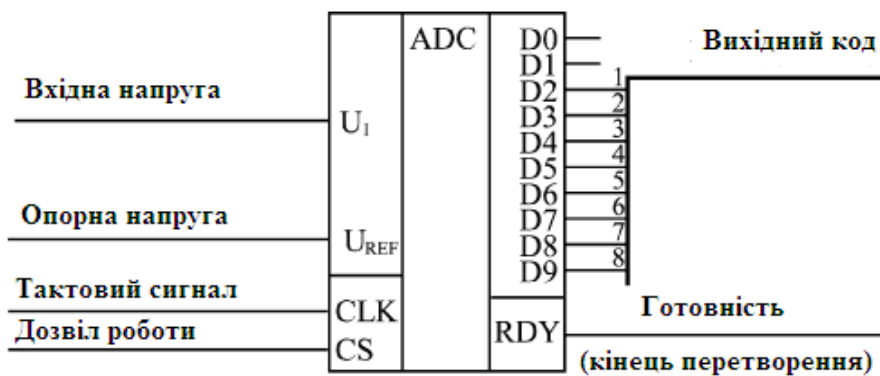


Рис. 8.9. Зниженн кількості бітів у вихідному коді АЦП

Протилежна проблема - збільшення швидкості передачі АЦП - виникає частіше. Існує багато схематичних рішень для об'єднання декількох мікросхем АЦП для зростання кількості бітів у вихідному коді, але більшість із них вимагає складних кінцевих обчислень похибок перетворення та використання аналогових вузлів. Тому розглядати їх не будемо. Зверніть увагу, що при необхідності збільшення розрядності спочатку потрібно спробувати знайти чіпсет з відповідною розрядністю і тільки потім розглядати можливість підключення декількох АЦП.

Подивимося деякі типові схеми підключення АЦП, що використовуються в перетворювальних схемах.

Перша схема (рис. 8.10) спрямована на визначення моменту, коли вхідний аналоговий сигнал перевищує певний пороговий рівень напруги. Схема генерує вихідний сигнал (позитивний фронт), коли вхідний аналоговий сигнал перевищує встановлений рівень, який визначається цифровим пороговим кодом. Пороговий код зрівнюється з вивідними двійковими сигналами АЦП за допомогою компаратора кодів. Вихідний сигнал компаратора кодів реєструється в тригері з сигналом RDY від АЦП перетворювача, що дозволить виключити дію дуже коротких імпульсів, що створюються на виводі порівнювача при зміні вхідних кодів. Використання цього перемикача задержує вивідний сигнал на один період.

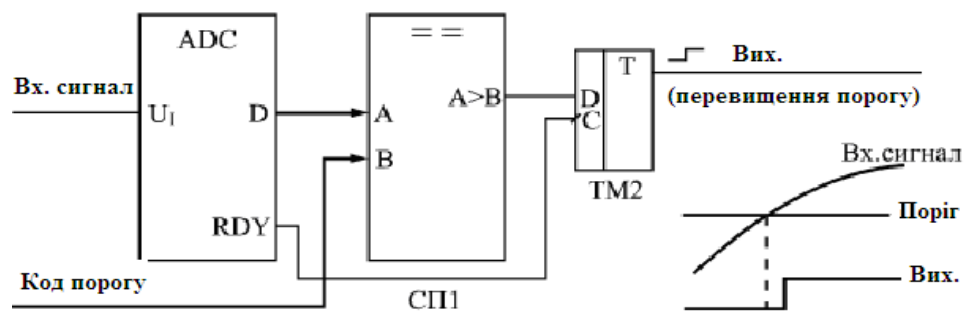


Рис. 8.10. Фіксатор перевищення вхідним сигналом встановленого рівня

Може здатися, що використання АЦП в цьому випадку недоцільно, але слід мати на увазі, що в схемах АЦП, який перетворює вхідні імпульси в рядок двійкових кодів, зазвичай вже є отже, додатковий АЦП не потрібен, просто додайте компаратор до схеми та тригерних кодів.

АЦП також використовуються для обчислення амплітуди вхідного аналогового сигналу. Для таких розрахунків може бути використана схема вказівки екстремального значення вхідного коду. Як джерело вхідної кодової послідовності використано АЦП (рис. 8.11).

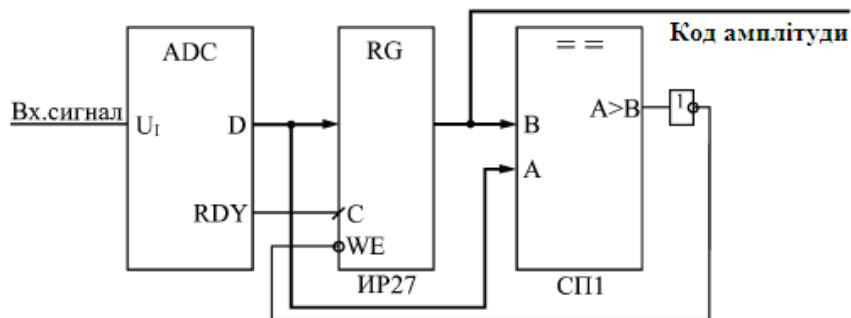


Рис. 8.11. Обчислювач амплітуди аналогового сигналу

У регістрі з записуючим вводом сигналу з виводу АЦП записується в сигнал RDY, якщо текуче значення сигналу більше значення, яке раніше зберігалося в регістрі. Очевидно, що після першого такту ввідного сигналу регістр буде містити двійковий код величини ввідного сигналу. Протягом періоду перетворення АЦП компаратор коду і регістр повинні встигнути спрацювати.

Таке визначення величини ввідного сигналу використовується в складній аналого-цифровій системі, яка вже має АЦП-перетворювач, який безперервно перетворює вхідний сигнал у коди, додатково знадобляться лише кодовий компаратор і регістр.

Найпоширенішим використанням перетворювача є переведення ввідного сигналу в стім кодів, які потім зберігаються в буферній пам'яті. У цьому випадку найбільше підходить односторонній регістр з визначеним режимом роботи. Це означає, що на початку в реєстрову пам'ять записується масив зразкових кодів вхідних сигналів, а потім даний масив читається для обробки (рис. 8.12). Прикладом може бути цифровий осцилограф, використовуваний для візуального спостереження сигналів, побудований таким чином. На діаграмі на рис. 8.12, сигнал RDY від АЦП використовується як строб запису. Справжні аналого-цифрові пристрої набагато складніші, реалізують синхронізацію запису з ввідним сигналом, засобами попередньої обробки аналогового сигналу, але принцип залишається незмінним.

Нарешті, розглянемо діаграму на рис. 8.13, що дозволяє подвоїти швидкість АЦП, а точніше подвоїти частоту записування кодів відліків ввідного сигналу в реєстрову пам'ять.

Принцип дії схеми досить простий: використовуються два АЦП і два буфера, що роблять по черзі, парні відліки вхідного сигналу обробляються одним АЦП зі своїм регістром, а непарні відліками іншим АЦП зі своїм регістром. Завдяки цьому досягається швидкість запам'ятовування кодів вхідних сигналів на частоті, що вдвічі перевищує частоту обробки кожного АЦП.

Сигнали синхронізації АЦП та імпульси RDY на виводах АЦП повинно бути зміщено один відносно одного на половину тактового періоду. Зчитування зареєстрованих кодів з обох буферів також повинно здійснюватися по черзі: перший код читається з першого буфера, другий з другого, третій знову з першого і т. д. У цьому випадку обсяг обох буферів додається. Наприклад, якщо кожен буфер організовано у форматі 64x8.

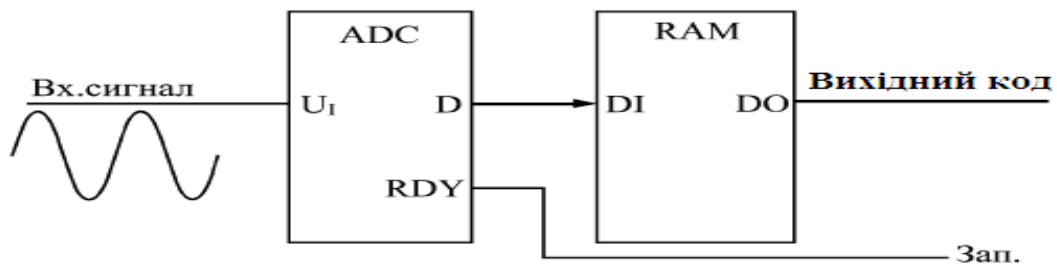


Рис. 8.12. Використання буферної пам'яті для запам'ятовування кодів з виходів АЦП

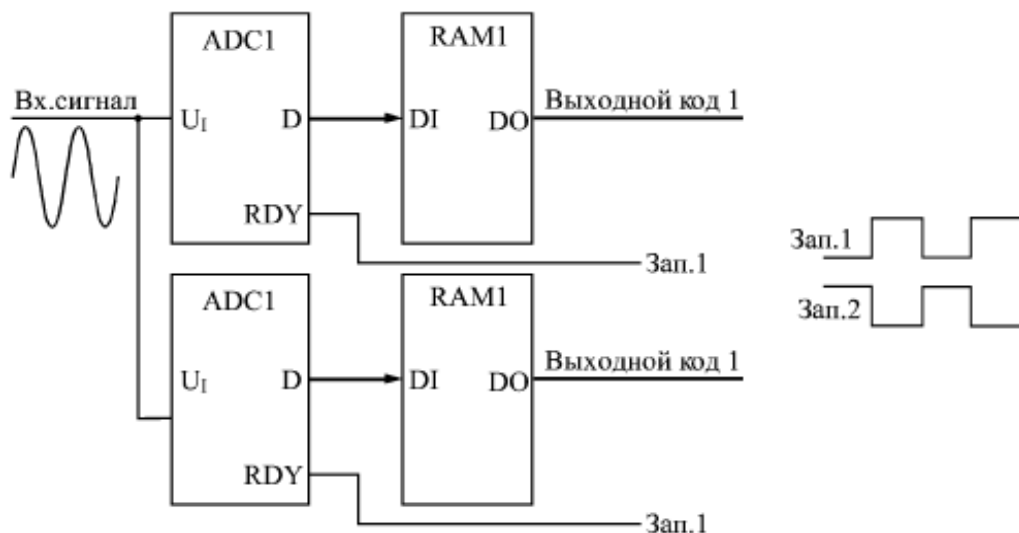


Рис. 8.13. Збільшення вдвічі частоти перетворення вхідного сигналу за допомогою двох АЦП з буферами

Використовуючи цей принцип, частоту обробки вхідного сигналу можна збільшити в три, чотири і більше разів. При роботі необхідно лише координувати три, чотири і т. д. АЦП, кожен з яких повинен мати свій буфер пам'яті.

Крім розглянутих АЦП послідовного і паралельного типу, існує також АЦП непрямого перетворення. У ньому вхідний аналоговий сигнал перетворюється за допомогою інтегратора в інтервал часу між оцифрованими сигналами чи в частоту наступних цифрових сигналів. Вивідний оцифрований код, відповідний ввідному змінному сигналу, створюється шляхом вимірювання тривалості відрізка часу або частоти наступних цифрових імпульсів (рис. 8.14).

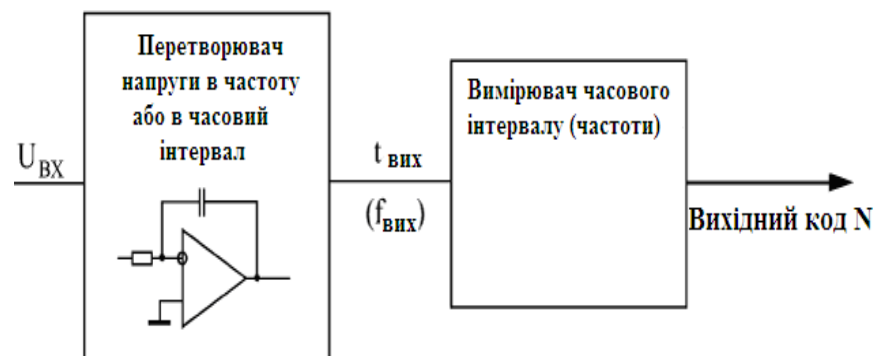


Рис. 8.14. АЦП з проміжним перетворенням

При використанні вивідної частоти, такий АЦП називається «перетворювачем частоти напруги» (VFC). Даний метод може за допомогою простого обладнання досягти високої точності переводу, який не залежить від великої кількості параметрів компонентів і умов навколишнього середовища.

Заміри затримок інтервалів і частот відстеження імпульсів виконується за допомогою простих цифрових схем. Такі вимірювання можна проводити з високою точністю завдяки використанню дуже стабільного еталона часу - кварцового генератора. Ще однією перевагою PNP є можливість легко передавати вихідний сигнал на великі відстані.

РОЗДІЛ 9. ОСНОВИ МІКРОПРОЦЕСОРНОЇ ТЕХНІКИ

9.1 Класифікація процесорних пристроїв

Прийнято такий розподіл процесорних засобів, що реалізовані в одному кристалі:

- мікропроцесори для чисельної обробки інформації;
- мікроконтролери для керування/контролю;
- сигнальні процесори цифрової обробки імпульсів;
- матриці програмованої логіки.

Дана класифікація суттєво змінилася з використанням більш нових схем архітектури ядра процесора та впровадженням новітніх технологій. Враховуючи взаємодію між різними типами архітектур мікропроцесорів, у майбутньому можуть знадобитися нові правила вибору мікропроцесорів.

Універсальні процесори використовуються для використання в комп'ютерних засобах: персональних комп'ютерах, а останнім часом і в масових паралельних суперкомп'ютерах. Їх головною особливістю є наявність передових пристроїв для ефективною реалізації операцій з плаваючою комою над 64-бітними аргументами.

Цифрові сигнальні процесори призначені для обробки в реальному часі цифрових потоків, отриманих в результаті оцифровки аналогових сигналів. Це призводить до їх відносно низької пропускної здатності та обробки переважно цілих чисел. Однак сучасні сигнальні процесори здатні виконувати обчислення з плаваючою комою на 32-40-бітових операндах.

Цифрові сигнальні процесори (DSP) є відносно новою категорією процесорів. Метою DSP є отримання поточних даних від аналогової системи та створення відповідної відповіді. DSP і їх ALU (Arithmetic Logic Unit - арифметико-логічний блок, який є апаратним інструментом для виконання обчислень) працюють на дуже високій швидкості, що дозволяє обробляти дані в масштабі

реального часу. DSR часто використовуються в мікрофонах з активним шумозаглушенням, встановлених у літаках (другий мікрофон забезпечує сигнал навколишнього шуму, який віднімається від сигналу першого мікрофона, таким чином дозволяючи придушити шум і залишити лише голос), або для придушення зображення розщеплення телевізійних сигналів.

Розробка алгоритмів DSR є особливим розділом теорії управління. Викладання цієї теорії вимагає глибоких математичних знань і виходить за рамки цієї книги (хоча пізніше ви дізнаєтеся про «нечітку логіку», нетрадиційний метод керування системами за допомогою комп'ютера).

Пристрої DSR не призначені для автономного використання. Зазвичай вони є частиною систем і служать пристроями для управління зовнішніми пристроями, а також для обробки вхідних сигналів і формування відповідної реакції.

Програмовані логічні інтегральні схеми — це матричні інтегральні схеми, які дозволяють програмно створити електронний засіб в одному корпусі, еквівалентному схемі, що складається з кількох десятків до кількох сотень стандартизованих інтегральних схем. Порівняно з іншими мікроелектронними технологіями технологія матриць програмованої логіки схем забезпечує рекордно сжаті проектно-робочий цикл (від кількох днів до кількох тижнів), мінімальні витрати на створення, максимальну оперативну змінність у разі модифікації апаратного забезпечення.

Найбільшою спеціалізацією і різноманітністю функцій характеризуються мікроконтролери, які використовуються у вбудованих системах вимірювання та керування, в тому числі побутової техніки. Загальна кількість кристалів з різними системами команд перевищує 500, і всі вони, завдяки наявності продуктів, що їх використовують, мають власну стабільну частку ринку.

Основним призначенням вбудованих мікроконтролерів є забезпечення гнучкого (програмованого) керування об'єктами та зв'язку із зовнішніми пристроями за допомогою недорогих засобів. Ці мікроконтролери не призначені для виконання набору складних функцій, але вони здатні забезпечити ефекти-

вний контроль у багатьох сферах застосування. Ми розглянемо недорогі мікроконтролери, вартість яких коливається від 1,0 \$ до 20,0 \$ за штуку (ціна залежить від технічних характеристик, кількості клем корпусу, обсягу закупівлі).

Вбудовані мікроконтролери містять значну кількість допоміжних пристроїв, що забезпечує їх включення в реалізовану систему за допомогою мінімальної кількості додаткових елементів. Ці мікроконтролери зазвичай включають:

- Схема першого запуску процесора (скидання)
- Генератор тактових імпульсів
- Центральний блок
- Програмна пам'ять (E(E)R)ROM та програмний інтерфейс
- Пам'ять даних RAM
- Інструменти введення/виведення даних
- Таймери, що встановлюють кількість циклів команд

Структура мікроконтролерів показана на рис. 9.1. Дана структура дає представлення про те, як мікросхема спілкується із навколишнім світом.



Рис. 9.1. Структура мікроконтролера

Більш складні мікроконтролерні системи додатково можуть реалізувати наступні можливості:

- Вбудований програмний монітор/дебагер

- Внутрішні засоби програмування програмної пам'яті (ROM)
- Робота з перериваннями з різних джерел
- Аналоговий вхід-вихід
- Інтерфейс послідовний ввід/вивід (синхронний і асинхронний)
- Інтерфейс паралельний ввід/вивід (включаючи комп'ютерний інтерфейс)
- Використання зовнішньої пам'яті (режим мікропроцесора)

Всі ці можливості значно підвищують зручність використання мікроконтролерів і спрощують процес створення систем на їх основі. Слід зазначити, що реалізація цих можливостей у більшості випадків вимагає розширення функцій зовнішнього виведення.

Наразі кілька десятків компаній, серед яких: *Analog Devices*, *Oki*, *Infineon Technologies*, *кремній Зберігання Technologies*, *Temic* та інші досі випускають аналоги мікроконтролера 8051 (Intel) - прабатька всіх мікроконтролерів. Мікроконтролер 8051 реалізує архітектуру ядра процесора CISC (Complex Order Set Computer), яка працює з повним набором інструкцій. У класичному мікроконтролері 8051 більшість команд вимагає 1-2 машинних циклів. *Компанія Dallas Semiconductor* виробляє мікроконтролери 8051, в якому базові команди виконуються за чотири такти. Аналоги мікроконтролера 8051, виготовлені компаніями *Infineon* і *Philips Semiconductors*, вимагають шести машинних циклів для виконання основних інструкцій. Всього в мікроконтролері 8051 реалізовано 255 команд. Intel також випускає мікроконтролери MCS151/251, які повністю сумісні на рівні коду команд з мікроконтролером 8051, *Philips Компанія Semiconductors*, яка випускає понад 60 модифікацій мікроконтролера 8051, створила оригінальний 16-бітний мікроконтролер 8051XA на основі популярної 8-бітної архітектури, сумісний на рівні коду команди з мікроконтролером 8051XA в двох режимах: розширеному та режимі сумісності. Розширений режим використовує нові можливості мікроконтролера, включаючи потужні інструкції для багатозадачності. У мікроконтролері 8051XA більшість інструкцій з регістру в регістр виконуються за три машинних цикли (100 нс) на тактовій частоті 30 МГц. Всього в мікроконтролері 8051XA реалізовано 479

інструкцій. У таблиці 9.1 наведено основні параметри найпопулярніших 8-розрядних мікроконтролерів.

Однією з проблем використання повного набору інструкцій (архітектури CISC) є складність реалізації компілятора. Використання повного набору інструкцій, створеного не компілятором, не має сенсу. Рекомендується використовувати скорочену інструкцію. В умовах пошуку найкращої архітектури для чисельної обробки з'явилася архітектура ядра процесора із обрізаним набором команд - RISC архітектура. Ідея RISC-архітектури народилася серед творців великих комп'ютерів. Вони виявили, що деякі типи інформації мають не визначену частоту появи. Найбільша частота (понад 50%) – це локальні дані, а виклики процедур займають не менше 30% загального часу обробки. Однак однією з типових структур даних у чисельній обробці є векторні дані. Поняття вектора, яке часто використовується в інформатиці, відрізняється від поняття вектора в математиці та фізиці.

Поняття вектора — впорядкований список інформації. Кількість елементів списку — це розмір вектора. В ідеалі весь перелік можна записати у файл регістрів ядра процесора. У цьому випадку, отримавши доступ до даних, що містяться у файлі реєстру, можна досягти максимальної продуктивності. Реєстровий файл або векторні регістри можна розглядати як буферну пам'ять, розміщену між операційним блоком ядра процесора та основною пам'яттю. Крім того, можливість використовувати одну інструкцію для виконання операції над великою кількістю даних, які можуть бути в пам'яті або файлі реєстру, є надзвичайно вигідною. Такі інструкції можуть реалізувати ортогональну структуру, яка дозволяє виконувати будь-яку операцію над любим регістром, використовуючи любий набір команд одночасно. Зменшення кількості інструкцій може зменшити площу, яку займає ядро процесора на кристалі, і, отже, дозволяє збільшити площу для внутрішніх регістрів (реєстровий файл). У мікропроцесорах зі скороченим набором інструкцій підвищення продуктивності також досягається завдяки використанню компілятора для безпосереднього перекладання програмного коду, що написані на мовах високого рівня. Сьогодні багато компаній випускають 8- і 16-розрядні мікросхеми контролерів з

RISC архітектурою. Це *Atmel* (серія AVR) з *Texas Instruments* (серія MSP430), *Infineon* (серія 166) тощо.

Таблиця 9.1 – Головні параметри мікроконтролерів

Фірма, тип	Тактова частота, МГц	Вбудована пам'ять		Напруга живлення, В	Таймери	АЦП, ЦАП	Послідовний інтерфейс
		OTP/ROM/Flash/EEPROM	RAM				
Analog Devices AduC824	12...16	Flash (8 KB) EEPROM (640 B)	256 B	3/5	3 станд. 8051 (16-розр.)	24- і 16-розр., ЦАП	UART, I2C, SPI
AduC812						12-розр., ЦАП	
Atmel AT89	0...33	Flash (1.32 KB) EEPROM (128..512 B)	128...512 B	2.7...6	1...3 (16-розр.)		UART, SPI
AT90	0...12	Flash (1.8 KB) EEPROM (0..512 B)	0...512 B	2.7...6	1...4 (8- і 16-розр.)	10-розр.	UART, SPI
ATtiny	0...8	Flash (1.2 KB) EEPROM (0..128 B)	0...128 B	1.8...5.5	1...2 (8- і 16-розр.)	10-розр.	UART, SPI
ATmega	0...6	Flash (16..128 KB) EEPROM (512...4 KB)	1...4 KB	2.7...5.5	3...4 (8- і 16-розр.)	10-розр.	UART, SPI
Cybernetic Micro Systems P-51	1...60	-	8 KB програм і 4 KB даних	3.3/5	3 (16-розр.)	-	UART
<u>Infineon</u> SABC500	0...40	ROM (8...64 KB) OTP (8...64 KB)	256...3328 B	4.5...5.5	3...5 (16-розр.)	10-розр.	1... 2 (UART, USART)
<u>Philips</u> 87C51Fx	33	OTP (8...32 KB)	256 B	2.7...5.5	3 (16-розр.)	-	UART
Rx2	20	Flash (16..64 KB)	512...1024 B	5	3 (16-розр.)	-	UART
<u>Hitachi</u> H8/3664	10	Flash (32 KB)	2 KB	2.5...5.5	3 (8- і 16-розр.)	10-розр.	UART, I2C, SPI
H8/3802	10	OTP (16 KB)	1 KB	2.5...5.5	5 (8- і 16-розр.)	8-розр.	UART, SPI

9.2. Мікропроцесори у вимірювальних пристроях

Загальні відомості. Як уже зазначалося, сучасний етап розвитку техніки характеризується все більш інтенсивним і глибоким проникненням мікропроцесорів в різні галузі, докорінно змінюючи властивості багатьох пристроїв і відкриваючи нові можливості їх застосування. За широтою й ефективністю використання мікропроцесорів контрольно-вимірювальні прилади посідають одне з перших місць.

Природно виникає питання: «Які переваги використання мікропроцесорів у вимірювальних пристроях як ефективних рішень для схем на основі мікропроцесорної системи?»

Загалом, відповідь може критися в назвах ряду журнальних статей та інформаційних матеріалів: «Мікропроцесор революціонує електронне приладобудування». І справді це так. Але навіть якщо утриматися від пафосних інтонацій і перейти на звичайну технічну мову, варто зауважити, що використання мікропроцесорів у техніці вимірювання, що дозволяє радикально підвищити точність приладів, істотно розширити їх можливості, підвищити надійність, швидкість роботи і вирішити проблеми, які раніше взагалі були нерозв'язними.

Детальний розгляд застосувань мікропроцесорних схем у приладах вимірювання показує, що з їх допомогою можна досягти багатофункціональності приладів, спрощення керування процесом вимірювань, автоматизації регулювань та автоматичної перевірки, покращення метрологічних властивостей приладу, що виконує обчислення, статистична обробка показників спостережень, знаходження та переведення в лінійний вид функціоналу фізичної величини, що вимірюється утворення програмованих, повністю автоматизованих пристроїв. З'явився новий клас «розумних» пристроїв, які ще називають «мислячими» або «інтелектуальними».

Докорінно перетворився процес побудови пристрою. Мікропроцесор став основною частиною пристрою, що призводить до зміни конструктивних рішень схем, компонування, управління та входження опрацювання даних у

процес вимірювання (виконується без участі персоналу). Впровадження МП відкрило можливість створення різнофункціональних пристроїв з багатьма програмами функціонування, виконало пристрої економічнішими спростило вирішення проблеми доступу до стандартної інтерфейсної шини (каналу загального користування) та управління інтерфейсом. Все це спростило роботу пристроїв і радикально збільшило працездатність користувачів.

Якщо докладніше, то можливості, або особливості пристроїв, що включають мікросхемні системи, і з'ясуємо в результаті цих можливостей.

Ідея створення багатофункціональних вимірювальних пристроїв, що використовується для визначення декількох параметрів сигналу або показників об'єкта випробувань, не є новою. Його впроваджують понад три десятиліття. Проте до використання мікросхем широкофункціональні пристрої являли сукупність кількох різних вузлів, об'єднаних в одну структуру. При використанні таких пристроїв переведення від одного вимірювання до іншого проводиться за допомогою комутаційних засобів. При такій комутації з'єднувальних ланцюгів персонал збирає з деяких вузлів певний засіб для визначення необхідного параметра сигналу або характеристики досліджуваного об'єкта. Алгоритм роботи вимірювального приладу, встановлений при його розробці, залишається незмінним під час експлуатації. Іншими словами, *традиційні* МФУ розроблені за схемою з *чіткою логікою*. Для нього характерне протиріччя між багатофункціональністю, кількістю можливих функцій пристрою з одного боку та економічністю та технічною ефективністю з іншого. Проблема перемикання та керування ніколи не втрачала уваги при проектуванні пристроїв, призначених для виконання багатьох функцій, і не завжди вирішувалася ефективно.

Мікропроцесорна система, впроваджена в багатофункціональний вимірювальний прилад, докорінно змінила його, перетворивши пристрій з надійною логікою роботи в *пристрій з програмним керуванням*. Функціональні можливості такого пристрою залежать від програми, що виконується, і можуть бути легко змінені перемиканням на іншу програму, що зберігається в постійній пам'яті. Отже, логіку програмування таких пристроїв інколи називають

збереженою. Вона створює гнучкість при переплануванні, робить можливим нарощувати можливості модернізації пристрою без суттєвих змін його схеми. Використання логіки програмування зазвичай знижує вартість пристрою.

Збільшення точності пристроїв, під точністю пристрою вимірювання розуміється якість засобу вимірювання, яка відображає наближеність до мінімуму його помилок. Відповідно наближення до нуля постійних похибок показує правильність вимірювального приладу, а наближення до нуля випадкових похибок визначає точність показань вимірювального приладу.

Помилки вимірювального приладу належать до його метрологічних властивостей. Можливо назвати наступні способи збільшення точності вимірювального засобу, які можливі за рахунок впровадження у прилад мікросхемних систем:

- компенсація (усунення) систематичної похибки, зокрема безпосереднє встановлення початку відліку перед вимірюванням;
- робота автоматичного калібрування;
- здійснення самоперевірки;
- обмеження впливу випадкових помилок, методом проведення багаторазових вимірювань (одноразових вимірювань) з наступним вирахуванням середнього результату;
- знаходження та усунення грубих помилок;
- відображення відомостей про значення помилок під час вимірювань.

Розширення вимірювальних можливостей приладів. Застосування МП дозволяє значно розширити можливості вимірювання широкого діапазону параметрів сигналів і характеристик приладів. В першу чергу це пов'язано з використанням, здавалося застарілих методів визначення: непрямих і кумулятивних.

У зв'язку з необхідністю використання кількох пристроїв, проведення деяких відліків і подальших вичислень косвенні вимірювання визначаються як примітивні та застарілі. Навіть з застосуванням калькуляторів в ряді випадків можуть займати тривалий проміжок часу, і головне вимагають сталої уваги та роботи і не можуть досягти великої продуктивності. А також, оцінка похибок

косвенних вимірювань не зовсім проста і без неї жодне вимірювання не може вважатися надійним.

Ситуація докорінно змінюється, коли в пристрій впроваджується мікропроцесорна система. За командою, отриманою з клавіатури, автоматично вибирає режими вимірювання відповідно до визначеної програми, записує результати отриманих вимірювань, виконує визначені обчислення та виводить обчислене значення фізичної величини, що вимірюється на систему відображення. Хоч і вимірювання по принципам залишаються непрямими, персонал використовує їх як прямі, тому що, приєднавши пристрій до джерела дослідження, він безпосередньо отримує результат вимірювання.

Наприклад можливе вимірювання потужності P , яка розсіюється на резисторі навантаження, цифровим вольтметром. Вимірювання проводять за формулою $P = U^2 / R$, де U - падіння напруги на опорі ; R – сам опір. Цифровий вольтметр отримує програму, за якою на початку визначається опір резистора і записується значення, далі визначається напруга на опорі, а потім визначається потужність.

В якості інших прикладів можна навести вимірювання опору резисторів за формулою $R = U / I$ закону Ома вимірювання посилення підсилювача, як визначено $K = U_{a\delta} / U_{a\delta 0}$.

Наведені приклади стосуються відносно простих математичних співвідношень. Однак на практиці часто доводиться опосередковано знаходити значення і таких фізичних величин, які залежать від великої кількості безпосередньо виміряних інших фізичних величин. Використання мікропроцесорних систем робить дані визначення простими для користувача, який має прямі показання з пристроєм і не має відчуття, що насправді виконуються непрості косвенні вимірювання.

Мікросхемні пристрої ще більш якісні при загальних вимірюваннях, тобто одночасному вимірюванні декількох фізичних величин, у яких знайдені значення визначаються розв'язуванням системи рівнянь, отриманих у результаті прямого вимірювання різних комбінацій цих величин.

Спростіть і полегшіть керування пристроєм. На перший погляд, розширення функцій, які виконуються програмними пристроями, має призвести до зростання кількості органів керування. Але насправді це не так. Критерієм великого показника використання програмного коду у вимірювальному приладі є складність його панелі керування.

Загальновизнано, що розумний пристрій має найпростіший набір елементів керування. Сучасні пристрої, що містять мікросхеми, характеризуються кнопковою системою керування, виконаною у вигляді клавіатур, яка нагадує звичайку клавіатуру.

В цифрових мультиметрах, які мають багато функціональних можливостей, зміна функцій, границь вимірювань і методів роботи (всього 44 комбінації) управляється з використанням клавіатури, що має декілька клавіш. Це досягається тим, що вибрана клавіша управляє аналоговими схемами опосередковано - через мікроконтролер, який вибирає різноманітні комбінації сигналів, що створюються при нажиманні клавіш.

Другим прикладом спрощення управління та зменшення кількості ручок і кнопок може бути невеликий семи-розрядний частотомір, що працює в межах 10 Гц...1 ГГц. На лицьовій панелі даного пристрою є всього два ввідних контакту (для вимірювання з частотою 10 ... 75 МГц, інший - для сигналів з частотою 70 МГц ... 1 ГГц), ручка регулювання точності вимірювань та двохваріантний кнопковий вимикач тривалість тимчасового стробування (час вимірювання): 1 с та 1 мс.

За рахунок автоматизації вибору меж вимірювань, інтервалу дискретизації напруги досліджуваного сигналу та інших режимів роботи пристрою радикально зменшено кількість елементів керування. У деяких пристроях, якщо експериментатор робить неправильні кроки, з'являється сигнал тривоги, а на дисплеї з'являються інструкції, що вказують, що експериментатор повинен зробити, і правильну послідовність дій.

Здатність отримувати математичні функції вимірюваних величин. Залежно від розв'язуваної задачі експериментатора може цікавити не безпосеред-

ньо значення фізичної величини, отримане під час вимірювання, а різні її математичні функції. Багато пристроїв, що містять мікропроцесорні системи, дозволяють автоматично здійснювати запрограмовані функціональні перетворення. Прикладами таких перетворень можуть бути:

1. Множення знайденого значення A на константу c . Одночасно зчитує прилад $A_n = cA$. Стала вводиться кодом після натискання кнопки.

2. Отримання похибки результатів вимірювання від визначеного значення A_H : абсолютних $A - A_H$ і відносних, виражених у відсотках від номінального значення, $100(A - A_H) / A_H$ тобто.

3. Зміщення, яке передбачає обчислення сталої за результатом вимірювання.

4. Розрахунок коефіцієнтів: шляхом поділу на сталі значення (приклад, при знаходженні значення сталого струму через опір на основі значення падіння вольтажу на даному опорі, виміряного вольтметром), визначення частки від поділу одного значення виміру на інше вимірювання результат (при знаходженні коефіцієнта посилення на основі результатів вимірювання напруги на виході та вході підсилювача).

5. Вивід результату визначення на логарифмічній шкалі. Пприклад, чотиріполосне затухання, визначене в децибелах: $a = 20 \lg(U_{ex} / U_{вх})$.

6. Залежності повинні бути лінеаризовані. Ця потреба поширена у випадку електричних вимірювань різних величин (температури), якщо напруга сигналу на виводі датчика є нелінійною функцією величини на визначеному вході. У таких ситуаціях значення вихідної напруги датчика буде розраховано відповідно до: за допомогою АЦП перетворювача в числа, які обробляються мікропроцесорною системою за заданою програмою, в результаті чого виходить лінійна залежність між показаннями пристрою і значеннями вимірюваної величини на вході щупа.

В інших пристроях можливий розрахунок будь-яких (в певних межах, звичайно) математичних коефіцієнтів за бажанням користувача.

Отримання статистичних характеристик. Серія вольтметрів, що включає мікропроцесорну систему, дозволяє оцінювати такі ймовірнісні характеристики випадкової величини, що перевіряється як середнє значення, середньоквадратичне значення, дисперсія, відхилення, і коефіцієнт взаємоподібності двох випадкових величин. . Більш широкими можливостями володіють мікропроцесорні пристрої, які спеціально створені для визначення характеристик сигналів.

Мініатюризація та економія обладнання. Швидке зменшення кількості елементів схеми приладу завдяки реалізації багатьох функцій мікропроцесорною системою, їх відносно низька вартість і значне зниження енергоспоживання дозволяють будувати компактні та економічні пристрої.

Підвищення надійності пристрою. Це обумовлено зменшенням кількості елементів схеми, реалізацією самодіагностики, використанням вузлів з некаліброваними значеннями (підсилювача в каналі вертикального відхилення осцилоскопа), з можливістю проведення корекції помилок, що підвищує вимірювальну надійність.

Зменшення часу створення. Часто для отримання нових властивостей пристрою на базі мікросхемних систем не потрібно істотних змін ні в схемі, ні тим конструкції засобу. Основним змістом дослідження є створення необхідного програмного забезпечення. Враховуючи те, що для загальноживаних МП вже накопичено бібліотеку створених типових додатків процедур вимірювання, розробка програмного забезпечення пристрою зводиться здебільшого до оптимального підбору доступних програм.

Структура систем вимірювальних. Пристрій, що містить МР, зазвичай містить інтерфейси, що можуть підключити його до стандартної шини інтерфейсу. Що дає можливість об'єднати певний набір приладів в одну контрольно-вимірювальну систему (вимірювально-розрахункову одиницю).

9.3. Поліпшення метрологічних властивостей приладів

Розглянемо основні можливості та методи зменшення похибок у пристроях, що містять мікросхеми мікропроцесорів.

Виключення систематичної помилки. Систематичні помилки викликані зміщенням нуля, невідповідністю фактичного коефіцієнта передачі тракту імпульсів визначеному значенню, флюктуаціями амплітудно-частотної характеристики системи передачі імпульсів та впливом характеристик аналого-цифровий перетворювач (АЦП).

У мікро пристрої є можливість виправляти та усувати систематичні помилки. Коротко пояснимо, як вирішити проблему. Щоб виключити зсув нуля, в цифрових вольтметрах, його ввідні клеми замикають і підключають до точки нульового потенціалу (маси). При цьому число, отримане на виході АЦП, характеризується зсувом нуля. Він запам'ятовується і вимінусовується з показників систем. Далі при вимірюванні напруги, що подається на вхідні висновки пристрою, буде автоматично внесена корекція для усунення систематичної похибки, викликаній зсувом 0.

Методи корекції систематизованої похибки, пов'язаної з тим, що міра коефіцієнта пропускання тракту імпульсів (характеризує внесене ним посилення або ослаблення) від номінального значення, полягає в наступному:

У пам'яті пристрою зберігається число B , яке не знищується при відключенні енергопостачання засобу і відповідає точно знайденому значенню ввідної напруги A_0 , тобто числу, яке буде отримано на виході АЦП, бо на вхід сигналу вольтметра підключена напруга A_0 і коефіцієнт передачі - перетворення АЦП відповідає їх номінальним значенням. У середині пристрою є цифро-аналоговий перетворювач (DAC), який містить приклад джерела живлення. При додаванні числа B на входи ЦАП на його виході буде створено напругу, значення якої дорівнюватиме A_0 . Ця напруга подається на вхід пристрою. У результаті аналого-цифрового перетворення виходить B' , яке відрізняється від

числа B наявністю систематичної похибки. Характеризується співвідношенням чисел $\alpha = B/B'$. Значення коефіцієнта α обчислюється мікропроцесором і зберігається в пам'яті. Тому пам'ять містить поправочний коефіцієнт.

Коли виміряна постійна напруга надходить на вхід пристрою, на виводі АЦП-перетворювача буде отримано число C' , що є значенням цієї напруги. Використання поправочного коефіцієнта, це перемноження числа C' на коефіцієнт α , виконане мікропроцесором, дає істинний результат визначення - числа C .

Завдання усунення систематичної помилки, що виникає внаслідок нерівномірності частотних характеристик кола передачі імпульсів, особливо складна при застосуванні великодіапазонних вольтметрів. Наявність у пристрої мікросхемної системи значно спрощує вирішення цієї проблеми.

До виносного вимірювального щупа (вимірювальної головки) приєднана табличка, яка показує масштабні коефіцієнти (поправочні коефіцієнти) для набору частот в діапазоні приладу 0...2 ГГц. Для усунення систематичної похибки, викликані відмінністю значення коефіцієнта пропускання K_i від номінального значення, K_0 в пристрій можна ввести будь-який коефіцієнт масштабування, відповідний певній частоті f_i . Значення каліброваних коефіцієнтів зберігається в пам'яті $K_0 = 1000$. Це значення може відобразитися на дисплеї пристрою та змінюватися командою з клавіатури (або інтерфейсної шини). Якщо експериментатор введе значення, K_i відмінне від K_0 , фактичне (введене) значення коефіцієнта буде відображено на дисплеї, щоб попередити експериментатора про те, що калібровка змінилося.

При знаходженні напруги мікросхема виконує операцію перемноження на уточнюючий коефіцієнт, тобто використовує використовуємий масштабний коефіцієнт.

Обмеження випадкової помилки. Як відомо, визначана частина похибки вимірювань не може бути виключена. Її вплив можна обмежити раціональною обробкою результатів спостережень.

Для врахування випадкових помилок використовуються ймовірнісні характеристики. З теорії ймовірностей ми знаємо, що більшість абсолютно випадкових величин визначаються законами ймовірностей. Але при розв'язуванні декількох задач вимірювань, достатніми властивостями випадкових похибок є їх простіші числові признаки: середня величина (математичне сподівання) і середнє квадратичне відхилення. При кількості N експериментів завжди обмежена, фактично використовуються числові статистичні характеристики, які називаються характеристичними оцінками.

Оцінка середнього значення результатів спостереження розраховується за формулою

$$A_{\text{сер}} = \frac{1}{N} \sum_{i=1}^N A_i, \quad (9.1)$$

де A_i – результат i -го спостереження, не спотворений систематичною похибкою; N – кількість спостережень.

Вираз використовується для оцінки середнього квадратичного відхилення випадкової похибки результату спостереження

$$\hat{\sigma}_v = \sqrt{\frac{1}{N-1} \sum_{i=1}^N v_i^2}, \quad (9.2)$$

Де $v_i = A_i - A_{\text{сер}}$ — відхилення i -го результату від його середнього виміру.

В теорії помилок виявляється, що середнє квадратична похибка результату визначення, визначене як $\hat{\sigma}_A$, розраховане для N груп ряду незалежних спостережень, для великого числа N , набагато нижча, ніж середньовизначене квадратичне відхилення $\hat{\sigma}_v$. Розраховується $\hat{\sigma}_A$ за формулою

$$\sigma_A = \hat{\sigma}_v / \sqrt{N} = \sqrt{\frac{1}{N(N-1)} \sum_{i=1}^N v_i^2}. \quad (9.3)$$

Формула (9.2) визначає абсолютну похибку. Щоб знайти відносну середню квадратичну випадкову похибку δ_v , значення, $\hat{\sigma}_v$ обчислене за (9.2), присвоюється $A_{сер}$.

Письмові вирази показують, що виконання кількох вимірювань і подальше усереднення є ефективним способом зменшення впливу випадкової похибки на результат вимірювання.

Компенсація внутрішнього шуму. Ця операція дозволяє підвищити чутливість вимірювального приладу і збільшити діапазон кількості значень, що вимірюється в бік менших значень. Компенсація, яка використовується в ви- значниках рівня високочастотного сигналу, зводиться до наступного.

Прилад містить вимірювальний перетворювач, який перетворює напругу перемінного струму високої частоти в напругу незмінного струму, величина якого відповідає середньому значенню напруги перемінного струму. До подачі тестового імпульсу $s(t)$ під час автоматичного обнулення $n\{t\}$ вимірюється середній квадрат шумового сигналу на вході перетворювача. Результат вимірювання $\overline{n^2(t)}$ зберігається. Після подачі корисного сигналу на вхід приладу на виводі засобу перетворення виводиться сума сигналу і шуму. У перетворювачі загальний сигнал $s(t) + n(t)$ зводиться в квадрат, щоб отримати сигнал $s^2(t) + 2s(t)n(t) + n^2(t)$. Середнє значення цього сигналу дає

$$\overline{s^2(t) + 2s(t)n(t) + n^2(t)} = \overline{s^2(t)} + \overline{n^2(t)}$$

(оскільки сигнали $s(t), n(t)$ незалежні, середнє значення їх добутку дорівнює нулю). Попередньо виміряний середній квадрат шумового сигналу віднімається з результату усереднення $\overline{n^2(t)}$, і визначена різниця рівняється $\overline{s^2(t)}$. Визначення корня дає середньовизначене квадратичне значення чистого корисного імпульсу $s(t)$, тому, що шум компенсується.

9. 4. Похибки вимірювання

Розгляд процесу вимірювання процесора як послідовності аналого-цифрових, цифрових і аналогових вимірюваннях, а також перетворень дозволяє представити похибку у виді суми складових. Не відповідність результатів цифрових перетвореньщо вимірюються від необхідних визначається похибками вимірювання процесора. Щоб дослідити ці типи помилок, існує три виклики їх появи:

- алгоритмічна помилка, викликана не відповідністю визначеного алгоритму визначення від адекватного;
- помилка округлення, визвана наявністю округлення проміжних цифрових перетворень у результаті вимірювання;
- динамічна помилка процесора, джерелом якої є обмежена швидкодія мікропроцесора (процесора).

Розвиваючи ідею класифікації на основі відбору факторів, що призводять до помилок, ми введемо поняття гіпотетичного алгоритму вимірювання , що дозволяє отримати справжнє значення вимірюваної величини. Якщо їх можна сформулювати, то він визначений значенню величини, що вимірюється. Наприклад, використовуючи в закон Ома, можливо інтерпретувати визначені позначки току I , напруги U і опору R як гіпотетичні алгоритми вимірювання, виконання яких дозволяє визначити справжнє значення I , U або R . Тоді виникнення алгоритмічного похибка буде визначатися різницею між прийнятим алгоритмом вимірювання та гіпотетичним.

Використання мікропроцесорів у вимірювальних пристроях передбачає необхідність оцінки інструментальних похибок, які можуть виникнути під час обчислювальних процедур через обмеження бітової мережі процесора. Це так звані помилки округлення.

зазвичай діє умова абсолютної похибки розрахунку не перевищує $\Delta_0 = 2^{-n}$, де n – швидкість передачі МП.

Операції округлення в МП зазвичай виконують шляхом простого відсікання неврахованих цифр або симетричного округлення з урахуванням значення старшого розряду, який відкидається. Якщо l є кількістю цифр, які не включені в округлення при рівномірному розподілі ймовірностей округлення (що вірно в більшості випадків з $n \geq 8$), то дисперсії помилок простого усічення та симетричного округлення (з відносно малим l)

$$D_y \cong 2^{-2n} / 12 = \Delta_0^2 / 12. \quad (9,4)$$

Сподівання математичної похибки округлення

$$M_y = 2^{-(n+l+1)}. \quad (9,5)$$

Помилки мають негативний знак у всіх арифметичних операціях над числами, представленими в прямому та непрямому кодах. Тому при значній кількості використовуваних арифметичних операціях помилки можуть збільшуватися і перевищувати допустиме значення. Дуже важливо оцінити цю помилку у випадку розрядної мережі МП системи, наближеної до коефіцієнта аналого-цифрового переведення.

У вимірювальних приладах, що працюють під керуванням МП, істотну роль грають часові затримки запуску АЦП по відношенню до необхідної швидкості вимірювання. Ці затримки визначаються часом, витраченим процесором на опрацювання переривань, керування, запам'ятовування та виконання функцій інтерфейсу. Тому, коли відбуваються метрологічні випробування мікропроцесорних пристроїв вимірювальної техніки, де еталонна помилка залежить від вимірюючих пристроїв і програмних схем керування, необхідно передбачити визначення характеристик і похибок.

Вплив еталонної помилки датування на сумарну похибку мікропроцесорного вимірювального пристрою Δ залежить від часу зміни ввідного сигналу dx/dt :

$$\Delta = \Delta_{МПЗВ} + \frac{dx}{dt} \delta_t, \quad (9,6)$$

де $\Delta_{МПЗВ}$ – помилка мікропроцесорного вимірювального приладу; δ_t - помилка в датуванні граф.

Похибка δ_t може бути визначена як різниця між астрономічним часом і фактичним часом вимірювання або як значення відхилення інтервалу між наступними запитами АЦП від визначеного інтервалу, який повинен підтримуватися постійним.

При організації огляду аналого-цифрових перетворювачів їх швидкодію підбирають так, щоб можна було використовувати лінійну інтерполяцію між двома послідовними вимірюваннями в межах допустимих похибок $dx/dt = const$. Однак при аналізі швидких процесів ця умова зазвичай не виконується, що веде до отримання помилки.

9.5. Архітектура процесора

Мікропроцесорна система (MPS) — це набір взаємодіючих цифрових інтегральних схем і аналогових схем, організованих в вичислювальну або керуючу схему з мікросхемним процесором як схемою обробки інформації.

Узагальнену структуру мікропроцесорної системи наведено на рис. 9.2.

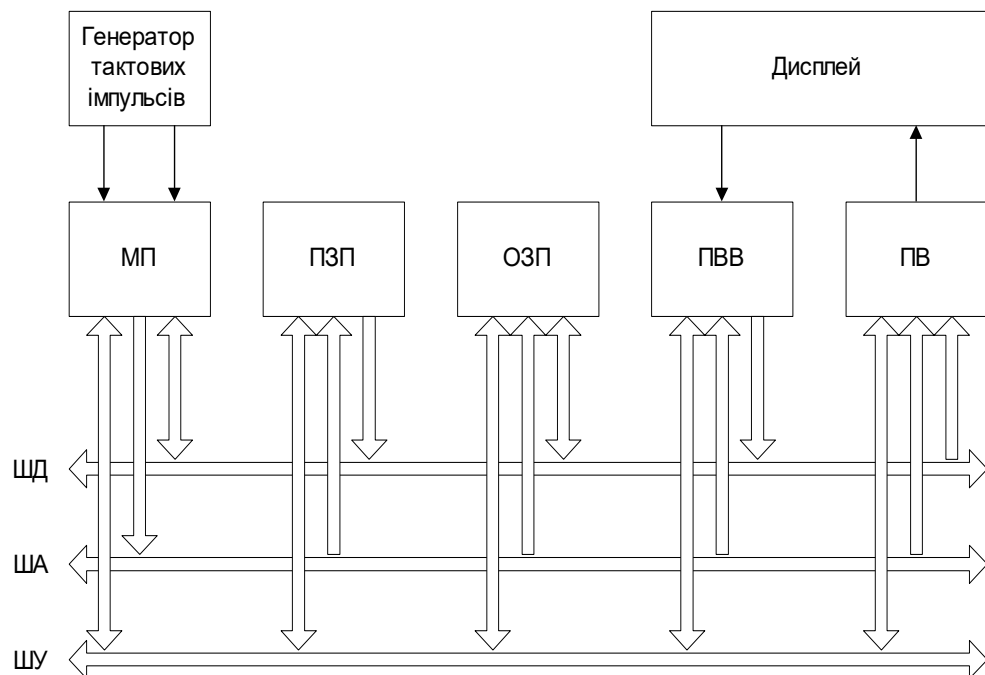


Рис. 9.2. Узагальнена структурна схема мікропроцесорної системи

Генератор тактуючих імпульсів є джерелом слідування прямокутних імпульсів, які використовуються для керування працею МП у часі. Для сучасних мікропроцесорів зовнішнім генератором стробуючих імпульсів не потрібен: він включається в його схему.

Основна пам'ять системи (зовнішня по відношенню до МП) складається з пристроїв постійної пам'яті (ПЗП) і оперативної пам'яті (ОЗУ).

Флеш-накопичувач — це пристрій, який зберігає програму та набір констант. Вміст ПЗУ не видаляється після вимкнення живлення. ПЗУ служить програмною пам'яттю.

Оперативна пам'ять - це пам'ять програм і даних, на якій базуються результати обробки та обчислень .

Для обробки даних у систему входить пристрій введення/виведення (I/O). Пристрій виведення (П) перетворює вихід у вид, зручний для отримання або зберігання користувачем. Магнітні диски, системи управління, монітор, аналого-цифрові та цифро-аналогові перетворювачі, пристрої для створення діаграм, пристрої друку.

Шина — це декілька ліній передачі, які використовуються для роботи певної функції (одна лінія для кожного біта передачі). Визначтком структури МПС є організація зв'язківна основі магістралі між системами, що входять до її складу. Це робиться з використанням 3 шин. Дані шини об'єднують МП із запам'ятовуваними пристроями (ПЗП, ОЗУ) та інтерфейсами введення-виведення , в результаті чого можливий обмін даними між розглянутими модулями системи.

Шина даних (SD) є двонаправленою шиною: по можуть передаватися до або з МП. При цьому слід зауважити, що потрібна передача інформації в двох напрямках не може відбуватися. Дані процедури розділені в часі в результаті мультиплексування часу.

З використанням шини адреси (ША) дані передається в одному напрямку - від МП до системи пам'яті.

Шина керування застосовується для передачі різноманітних сигналів, що забезпечують синхронізацію та спільне використання всіх системних модулів і вузлів МП.

Достоїнством конструкції таких шин є можливість використання нових модулів до MPS, наприклад кількох блоків RAM і PZP, для отримання необхідного обсягу пам'яті.

Порівняння архітектур ISS і RISS. Сьогодні існує багато процесорів RISC, оскільки вважається, що RISC є швидшим за процесори CISC. Ця думка не зовсім вірна. Існує багато процесорів, які називаються RISS, але насправді належать до CISC. Крім того, у деяких завданнях процесори CISC виконують програмний код швидше, ніж процесори RISS, або вирішують завдання, які процесори RISS не можуть виконати.

У чому реальна різниця між RISC і CISC. Мікропроцесори CISC виконують великий набір системних команд із розширеною адресацією (пряма, тощо) і дають програмісту можливість використати найбільш відповідну команду для створення потрібної операції. У RISS мікросхемах набір виконуваних команд зведеться до мінімуму. У той же час програміст повинен комбінувати команди для реалізації більш складних операцій. Потім буде показано, як виконуються операції завантаження в стек і виведення зі стека в мікропроцесорах RISS.

Здатність однаково використовувати всі регістри в процесорі називається ортогональністю або симетрією процесора. Це визначає додаткову гнучкість при виконанні певних операцій. Розглянувши виконання умовних переводів у програмах. У мікропроцесорах CISC умовний перевід зазвичай виконується відповідно до певного біта (прапора) у регістрі. У процесорах RISS перехід можливо буде відбуватися на певному значенні біта, розташованому в визначеному місці запам'ятовуючого пристрою. Що спрощує операції з прапорцями та використання програм, які їх використовують.

Успіх використання RISS визначається тим фактом, що їх простіші командам потрібно значно менша кількості машинних циклів для виконання.

Отже досягається значний приріст продуктивності, що дозволяє процесорам RISS вирішувати складні завдання.

Порівняння архітектур. Дана архітектура комп'ютера більш відома як архітектура фон-Неймана за ім'ям наукового лідера цієї розробки (рис. 9.3).

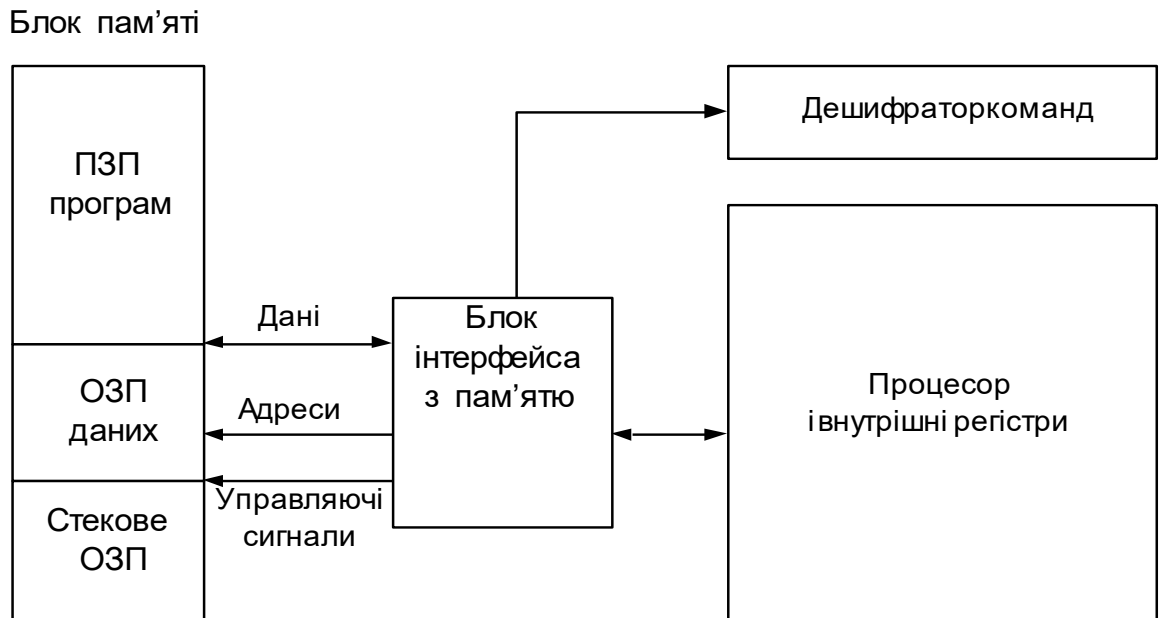


Рис. 9.3. Структура комп'ютера із Принстонською архітектурою

В даній архітектурі інтерфейс пам'яті вирішує запити до пам'яті, забезпечуючи отримання команд, а також читання та запис даних, що зберігаються в пам'яті або регістрах. Можна показати, бо система інтерфейсу є найбільшим впливим місцем між мікропроцесором і ОЗУ, тому що одночасно з даними необхідно вибрати іншу команду. Але у багатьох мікропроцесорах з архітектурою Princeton дана проблема рішення шляхом отримання наступної інструкції. Ця операція називається попередньою дискретизацією і реалізована в більшості мікропроцесорів з визначеною архітектурою.

Гарвардський університет описав розробку комп'ютера, який використовував окремі банки пам'яті щоб зберігати програмне забезпечення та дані (рис.9.4).

Структура Принстона виграла конкурс, оскільки більше відповідала технологіям сучасності. Застосування загальної пам'яті стало кращим за рахунок ненадійності електроніки 60-х років (до використання транзисторів) - при цьому надійність була вища.

Гарвардська побудова практично не застосовувалася до кінця 1970-х років, коли створювачі мікроконтролерів поняли, що архітектура надає переваги пристроям, які вони проектують.

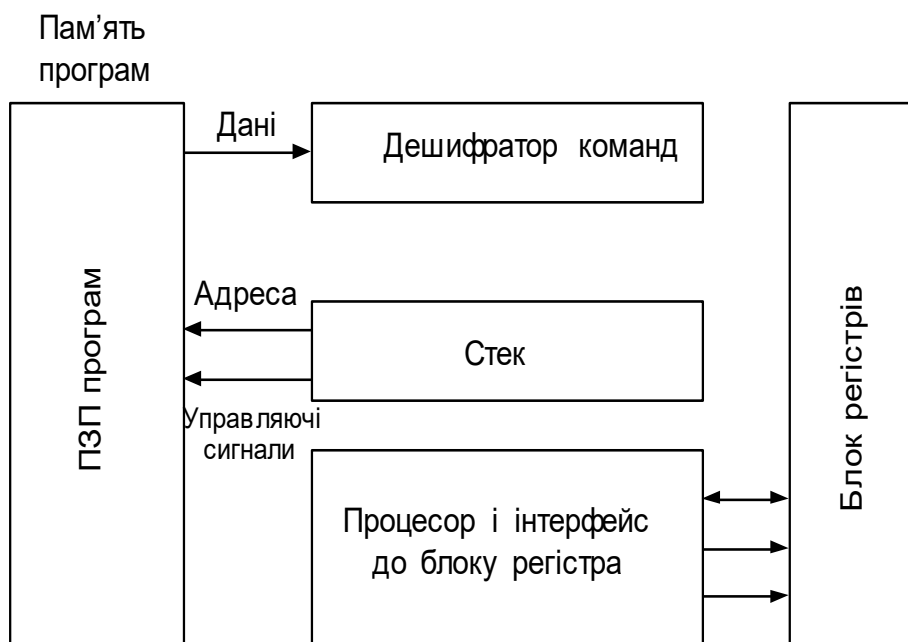


Рис. 9.4. Структура комп'ютера Гарвардської архітектури

Основна перевага архітектури фон-Неймана полягає в тому, що вона спрощує пристрій мікропроцесора, оскільки забезпечує доступ лише до однієї спільної пам'яті.

У випадку мікропроцесорів головним є те, що склад оперативної пам'яті (оперативна пам'ять) можна використовувати як для запам'ятовування даних, і для програм. У визначених програмах необхідний стек. Все це забезпечує велику гнучкість для розробника програмного забезпечення, насамперед з точки зору операційних систем реального часу, як буде обговорено. Гарвардська структура виконує інструкції за меншу величину тактів, ніж структура фон

Неймана. Це пов'язано з тим, що він має більші можливості для проведення паралельних операцій. Наступна інструкція може бути завантажена разом з виконанням першої, і немає потреби зупиняти процесор під час чекання інструкції.

Наприклад, якщо процесору архітектури Princeton потрібно прочитати байт і зберегти його в накопичувач, він виконує декілька дій, зазначену на рис. 9.5. В циклі з пам'яті визначається команда, у наступному циклі з пам'яті зчитуються дані, які потрібно помістити в накопичувач.

Цикл 1: - читання (вибірка) команди

Цикл 2: - читання даних та їх розміщення в акумуляторі

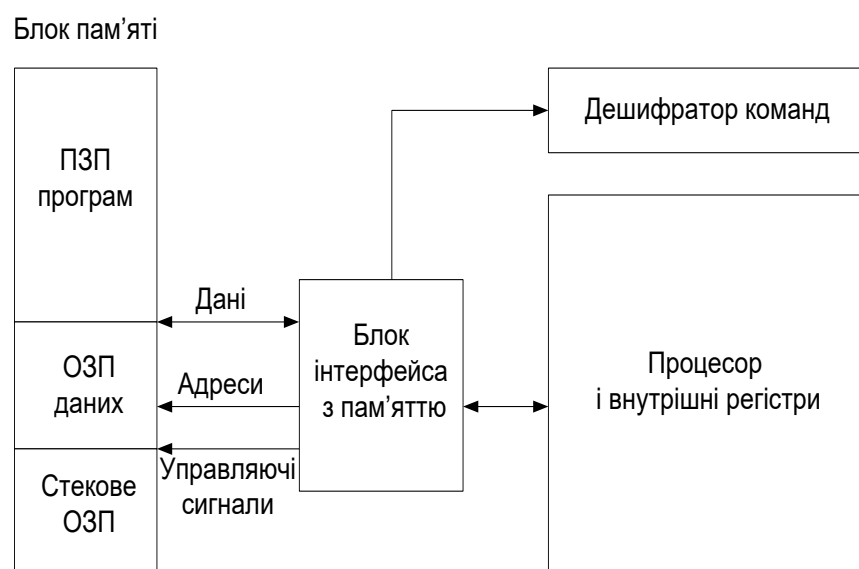


Рис.9.5. Виконання команди *mov Acc, Reg* у Принстонській архітектурі

В системі Гарвардській, яка показує вищий ступінь операційного одночасного, виконання текучої операції можна поєднати з вибором наступного коду (рис. 9.6). Операції також робляться в два цикли, при цьому вибір наступної команди відбувається одночасно з виконанням попередньої. Отже, команда виконується лише за один цикл (під час читання наступної команди).

Такий спосіб реалізації операцій паралелізм дозволяє виконувати команди за однакову кількість тактів, що полегшує визначення часу виконання

тактів і складних ділянок програми. Ця обставина особливо важливо при виборі мікроконтролера для завдань, де необхідно забезпечити строго певний час виконання.

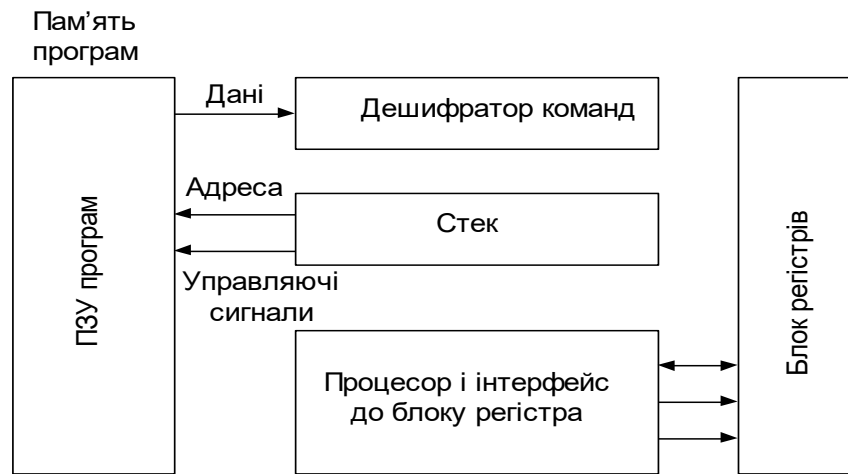


Рис . 9.6 . Приклади команди *mov Acc, R* Гарвардська структура.

Цикл перший: - закінчення першоїї команди - вибір команди *mov Acc, R*

Цикл другий: - закінчення команди *mov Acc, R* – та вибір наступної команди

Мікроконтроллер PIC фірми *Misroshir* робить будь-яку команду, за винятком, що змінюють вміст лічильника програми, за 4 такти. Це значно полегшує реалізацію критичних за часом процедур порівняно з мікросхемою *Intel 8051*, де інструкції можуть займати від 16 до 64 тактових циклів. З цієї причини часто неможливо вручну розрахувати чіткий час виділений на програму, і необхідно користуватися симуляторами або емуляторами.

Отже, ці загальні методи порівняння працездатності не слід застосовувати до всіх процесорів і мікроконтролерів, які реалізують ці дві архітектури. Порівняння найкраще робити на основі конкретного застосування. Різні архітектури та пристрої мають свої особливості, які дозволяють найкраще реалізувати певні програми. У деяких випадках конкретну програму можна реалізувати лише за допомогою певної архітектури та специфічних функцій мікроконтролера.

9.6. Типи пам'яті мікроконтролерів

У мікроконтролерах використовуються три основних типи пам'яті. Пам'ять коду програм - це постійна пам'ять, призначена для вмісту програмного коду та сталих. Дана пам'ять не міняє вміст під час виконання програми. Пам'ять даних призначена для зберігання змінних під час виконання програми. Регістри мікроконтролера – цей тип пам'яті включає внутрішні регістри процесора та регістри, які використовуються для керування периферійними пристроями.

Програмна пам'ять. Для збереження програм зазвичай використовується один з типів енергонезалежної пам'яті: PROM (одноразово програмований ROM), EPROM (електрично програмований ROM з ультрафіолетовим стиранням), EEPROM (електричний запис і стирання ROM, до цього типу також відноситься сучасна Flash пам'ять мікросхеми) або ПЗП (програмоване ПЗП). Усі ці типи пам'яті є енергонезалежними, що означає, що вміст пам'яті зберігається, коли мікроконтролер вимикається. Така пам'ять необхідна тому, що мікроконтролер не містить запам'ятовуючих пристроїв (магнітних дисків), з яких програма завантажується в комп'ютери. Програма постійно зберігається в мікроконтролері.

Під час роботи програма читається з даної пам'яті, а блок керування (вибору команд) забезпечує її використання та виконання потрібних операцій. Склад пам'яті програми не можна змінити (перепрограмувати) за час роботи програми. Отже функціональне використання мікроконтролера не зміниться, доки вміст його пам'яті програм не стерти (якщо це можливо) і перепрошито (записано нові командами).

Варто відзначити, що кількість розрядів мікроконтролера (8, 16 або 32 біта) вказується в залежності від швидкості передачі через шину даних. У гарвардській схемі команди можуть бути більшими за дані, щоб усю команду можна було прочитати за один такт. Наприклад, ІС, в залежності від моделі, використовують команди довжиною 12, 14 або 16 біт. У мікроконтролерах AVR

команда повинна мати розрядність 16 біт. Однак всі дані мікроконтролери мають 8-розрядну шину даних.

У схемах з прінстонською побудовою швидкість передачі даних зазвичай визначає пропускну здатність застосовуваної шини. У мікросхемах Motorola 68HC05 24-розрядний код розміщений в трьох 8-розрядних комірках програмної пам'яті. Щоб повністю перевірити таку команду, необхідно виконати три цикли читання цієї пам'яті.

Коли пристрій називають 8-розрядним, це означає кількість даних, яку здатний обробити мікроконтролер.

ПЗУ (ROM) використовується при введенні програмного коду в мікроконтролер при виробництві. Програма завчасно налаштовується і перевіряється, і передається на виробництво і програмування, де програма трансформується в малюнок маски на скляному фототрафареті. Отриманий шаблон з маскою застосовується в процесі створення зв'язків між елементами, що складають пам'ять програми. Тому таку пам'ять часто називають масово програмованим ПЗП.

ПЗУ є найдешевшим типом енергонезалежної пам'яті для масового виробництва. Однак вона має ряд суттєвих недоліків, через які в останні роки цей вид пам'яті майже не застосовується. Головними недоліками є суттєві витрати ресурсів і часу на створення нового набору фотошаблонів і запуск їх у виробництво. Зазвичай цей процес займає близько десяти тижнів і є прибутковим для виробництва кількох цих пристроїв. Отже за таких обсягів виробництва створюється перевага ROM над E(E)PROM. Присутнє і обмеження, зв'язане з можливістю використання таких мікроконтролерів тільки в певній області застосування, оскільки їх програма забезпечує виконання жорстко встановленої послідовності дій і не буде використана для вирішення інших задач.

Електронна пам'ять програмування ERROM створюється з комірок, запрограмованих за допомогою електричних сигналів і стертих за допомогою ультрафіолетового світла. ПЗП можна запрограмувати лише раз. Цей запам'ятовуючий пристрій містить одноразові з'єднання, які перегорають під час запису програми. Зараз така пам'ять використовується вкрай рідко.

Пам'яті ERROM - це МОП-транзистор із затвором, оточений двоокисом кремнію (SiO_2). Стек транзисторів підключений до "землі", а джерело підключено до напруги електроживлення. У нульовому стані (перед записом) плаваючий затвої не включає заряд, а МОП-транзистор не пропускає струм. У даному случаї високий потенціал утримується в джерелі, і коли здійснюється доступ до комірки, «1» зчитується. Заповнення пам'яті проводиться записом «0» у відповідні комірки.

Програмування проводиться подачею високої напруги на контрольний затвор. Ця напруга повинна бути достатньою для забезпечення пробою між провідним затвором і затвором, що плаває після чого заряд від провідного затвора буде передано плаваючому затвору. МОП-транзистор переходить у відкритий стан, замикаючи джерело на землю. В даному випадку при запиті до комірки зчитується «0».

Щоб прибрати вміст комірки, вона піддається впливу ультрафіолету, який дає заряду плаваючого затвора достатньо енергії, щоб опустити затвор. Даний цикл триває від декількох секунд до декількох хвилин.

Зрозуміло, мікросхеми ERROM виготовляються в корпусі з кварцовим вікном, що забезпечує доступ ультрафіолету. Коштує даний корпус досить дорого, отже значно зростає вартість схеми. Для здешевлення мікросхеми ERROM поміщені в безвіконний корпус (версія ERROM для одноразового програмування). Зниження затрат при застосуванні даних корпусів буде значним, бо ці версії ERROM тепер частіше використовуються замість масово програмованого ROM.

До цього мікросхеми програмувалися тільки при допомозі паралельних протоколів, що було досить складно реалізувати. В наш час протоколи запису сучасних пам'яті ERROM і EERROM суттєво змінилися, що дало можливість програмувати мікроконтролер безпосередньо в складі системи, в якій він працює. Цей метод програмування називається внутрішньосистемним програмуванням або ISP. Мікроконтролери ISP можна програмувати після припаювання на плату. При цьому зменшуються витрати на програмування, оскільки

відпадає необхідність використання спеціального обладнання – програматорів.

Пам'ять EERROM (Electrically Erasable Programmable Memory) можна вважати новим поколінням пам'яті EERROM. У такій пам'яті осередок стирається не ультрафіолетом, а електричним з'єднанням плаваючої шторки з "землею". Використання EEPROM може стирати та прошивати мікроконтролер, не виймаючи його з плати. Таким чином ви можете періодично поновлювати його код програми.

EERROM дорожчий за EROM (удвічі дорожчий за одноразовий програваний EROM), а EERROM трохи повільніший за EROM. Основною перевагою використання EEPROM є можливість перепрограмувати його кілька разів без необхідності видаляти його. Дане дає величезну перевагу на початкових етапах створення систем на основі мікроконтролерів або під час їх тестування, коли багато часу витрачається на багатофакторний пошук причин збою засобу та застосцвання слідуючих циклів стирання та запису програм для програмної пам'яті.

Функціонально флеш-накопичувачі мало чим не схожі на EERROM. Основні відмінності стосуються способу видалення збереженої інформації. У EERROM-пам'яті стирання виконується окремо для кожної комірки, тоді як у Flash-пам'яті стирання виконується цілими блоками. Якщо ви хочете змінити вміст однієї комірки флеш-пам'яті, вам доведеться перепрограмувати весь блок. У мікросхемах з елементами пам'яті EEPROM окремі частини програми можливо замінювати без перепрограмування всього пристрою.

Часто говорять, що мікросхема має флеш-пам'ять, хоча зазвичай, вона містить пам'ять EEPROM. Наразі між цими типами пам'яті мало відмінностей, тому деякі виробники використовують ці терміни як еквіваленти.

Пам'ять даних Коли ви вперше дізнаєтеся про мікроконтролер, багато хто буде здивований малим об'ємом його оперативної пам'яті, яка зазвичай становить десять слотів або сотні байтів. Якщо мікроконтролер використовує для зберігання даних EEPROM, то його обсяг не більший десятків байт.

Якщо пишуться програми для ПК, ви, ймовірно, задаєтеся питанням, що ви можете робити з такою малою пам'яттю. Програми на вашому ПК, ймовірно, включають змінні, об'єм яких кілобайти, на додаток до застосовуваних масивів інформації. При застосуванні масивів потрібний обсяг пам'яті може досягати сотень кілобайт. Отже, що ви можете зробити з 25 байтами оперативної пам'яті?

Отже, програмування мікроконтролера дотримується інших принципів, ніж програмування ПК. Дотримуючись кількох простих правил, можна вирішити багато проблем, використовуючи невеликий обсяг оперативної пам'яті. При програмуванні мікроконтролерів константи, по можливості, не записуються у вигляді змінних. Можливості мікроконтролерів використовуються максимально, щоб максимально зменшити розташування даних в оперативній пам'яті. Значить, що при створенні програм потрібно насамперед подбати про економію пам'яті. Програми повинні зосереджуватися на роботі без застосування великих наборів даних.

У мікроконтролерах оперативна пам'ять використовується для організації викликів підпрограм і роботизованих перериваннями. За даних операцій вміст лічильника програми та головних регістрів (накопичувача, регістра стану, індексних регістрів тощо) зберігається, а потім відтворюється коли відбувається повернення до основної програми. Стек — структура пам'яті даних, яка працює подібно до свого фізичного аналога — стосу паперів. Щойно щось поміщено в купу, воно залишається там, доки його не покладуть назад. Уявіть собі різнокольорові аркуші паперу, складені один на одного. Після зняття листів їх переміщують в оберненому порядку. Тому стек називають чергою LIFO (Last In, First Out) — останнім увійшов, першим вийшов.

В архітектурі Принстона оперативна пам'ять використовується для реалізації багатьох апаратних функцій, включаючи функції стека. У той же час продуктивність пристрою знижується, оскільки для доступу до різних типів пам'яті потрібні численні доступи, які неможливо виконати разом. Тому Принстонська структура зазвичай вимагає великої кількості годинників, ніж Гарвардська архітектура.

Процесори в гарвардській системі мають три області пам'яті, до яких можна звертатися паралельно (одночасно): пам'ять програм, пам'ять даних, яка включає простір введення/виведення, і стек. У гарвардській архітектурі операції зі стеком можуть виконуватися в пам'яті, призначеній спеціально. Це значить, що при роботі команди визову підпрограми "call" процесор гарвардської архітектури виконує декілька операцій одночасно. У Принстонській архітектурі, коли виконується інструкція «всі», наступна інструкція вибирається після того, як вміст програмного лічильника надходить у стек.

Пам'ятайте, що контролери двох архітектур мають обмежений обсяг пам'яті для збереження даних. Перебір даної межі може спричинити проблеми під час виконання програми.

Привиділенні в процесорі окремого стеку і кількість записаних даних більша його ємності, то вміст покажчика стека циклічно змінюється і стек починає видавати сигнали, посилятися на заповнений регістр стеку. Це значить, що після великої кількості команд виклику в стеку з'явиться неправильна адресація повернення, що була записаною замість правильної адресації. Якщо мікросхема користується в загальну область пам'яті для пошуку даних і стека, існує небезпека, що в разі переповнення стека відбудеться запис в область даних або буде зроблена спроба записати дані, завантажені в стек, в область ПЗП.

Тепер розглянемо можливості запису вмісту регістрів у стек. У деяких архітектурах немає інструкцій щодо завантаження вмісту регістрів у стек "push" і вивантаження його зі стеку "pop". З чотирьох сімейств мікроконтролерів, описаних у цій книзі, лише дві мають такі команди. Однак команди "push" і "pop" можна легко реалізувати за допомогою індексного регістра, що вказує на область стека. При цьому замість команд «push» і «pop» використовуються наступні дві команди:

Запис; Завантаження даних у стек.

переміщення[індекс], зростання; Збережіть вміст накопичувача в стек.

коефіцієнт збитковості; Перейдіть до наступної клітинки в стеку.

читання; Вивантажити дані зі стеку.

індекс приросту ; Перехід до попередньої клітинки.

рухатися за зростанням, [індекс]; Вставте вміст пачки в батарею.

Звичайно, це рішення є неефективним, бо використання спеціальних команд "push" і "pop", і застосовуємії реєстр може знадобитися для других цілей. Однак це рішення забезпечує моделювання стека при застосуванні процесорів, які не мають таких команд.

Існує ще одна проблема з наведеними вище прикладами. Що відбувається, якщо виникає переривання між першою та другою командами для імітації операцій push і push. Якщо обробник переривань використовує стек, дані, що зберігаються в ньому, будуть втрачені. Щоб запобігти цьому, ви можете вимкнути переривання перед виконанням цих команд або переставити їх у такому порядку:

преса; З даними, які надсилаються в стек.

коефіцієнт збитковості; Перейдіть до наступної клітинки в стеку.

переміщення [індекс], asi ; Збережіть вміст накопичувача в стек.

pop; Для завантаження даних зі стеку.

перемістити asi, [індекс] ; Помістіть значення стека в акумулятор.

швидкість росту ; Перейти до попередньої комірки в стеку.

Якщо програма переривається після першої команди, вміст стека не буде втрачено після обробки переривання.

9.7 . Регістри мікроконтролера. Простір введення-виведення

Як і всі комп'ютерні системи, мікросхеми мають багато реєстрів, які використовуються для управління різними засобами, підключеними до процесора. Отже, можуть бути реєстри, реєстри керування (реєстри керування перериваннями, реєстри керування таймером) (реєстри даних і реєстри керування паралельним, послідовним або аналоговим введенням-виведенням). Доступ до цих записів можна отримати різними способами.

Методи, реалізовані мікроконтролером для доступу до реєстрів, впливають на їх продуктивність. Тому дуже важливо розуміти, як отримати доступ

до регістрів, щоб писати ефективні прикладні програми для мікроконтролерів. У RISC-процесорах усі регістри (і часто акумулятор) розташовані за чітко визначеними адресами. Це забезпечує більшу гнучкість у роботі процесора.

Приклад:

<i>Акумулятор = IOPort;</i>	Завантажити вміст регістра IOPort в накопичувач
<i>Акумулятор = Акумулятор & (1 <math>^{\text{Bit}}</math>);</i>	Маскувати всі біти накопичувача, крім Bit ;
<i>якщо ZeroFlag $\neq 0$ перейти до адреси;</i>	Якщо нуль, то біт=1

Ця процедура буде скопійована в послідовність операцій:

якщо IOPort.Bit == 1 перейти до адреси

Монтажний вхід для мікроконтролерів Microchip PIC:

btfsc IOPort, bit; Пропустити наступну команду, якщо bit=0 йти за адресою

Даний процес більш ефективний реалізована в мікроконтролері Intel 8051:

jb IOPort.Bit, адреса ; Пропустити, якщо біт=1

Використовуючи процесор, який має прямий доступ до будь-якого регістру, ви можете отримати переваги при написанні простих прикладних програм. Наприклад, у мікроконтролері PIC вміст накопичувача та регістра стану не змінюється після забезпечення керування в залежності від значення біта в регістрі IOPort.

Важливим питанням є організація реєстрів в адресному просторі. У деяких процесорах всі комірки реєстрів і оперативна пам'ять розташовані в одному адресному просторі. Це означає, що пам'ять пов'язана з реєстрами. Цей підхід називається «відображення пам'яті пристроїв введення-виведення».

В других процесорах простір пристроїв введення-виведення відокремлено від загального простору пам'яті. Основною перевагою розміщення реєстрів введення-виведення в окремому адресному просторі є спрощення схеми використання пам'яті програм і інформації до загальної шини. Пристрій зазвичай займає невеликий масив адрес, що робить важким декодування його адреси разом із багатьох блоків основної пам'яті. Визначений простір вводу-виводу надає деякі переваги процесорам гарвардської архітектури, надаючи можливість читати команду під час доступу до реєстру вводу-виводу.

9.8. Зовнішня пам'ять

Не звертаючи уваги на великі переваги застосування вбудованої пам'яті, у визначених випадках до мікросхеми необхідно підключати додаткову зовнішню пам'ять (пам'ять програм і даних).

Є декілька основних способів підключення зовнішнього накопичувача.

1 спосіб - це використання зовнішньої пам'яті до мікросхеми, як і до мікропроцесора. Багато мікроконтролерів містять спеціальне обладнання для цього підключення.

2 метод полягає в підключенні пам'яті до пристроїв введення-виведення та реалізації доступу за допомогою програмного забезпечення. Цей метод дозволяє використовувати прості пристрої введення-виведення без необхідності реалізації складних шинних інтерфейсів.

РОЗДІЛ 10. АВТОМАТИЧНЕ ПРОЕКТУВАННЯ ЦИФРОВИХ ПРИСТРОЇВ У САД РСAD 2004

10.1 Загальні відомості про технологію виготовлення друкованих плат

Друкована плата є основою конструкції сучасних електронних пристроїв. Напівфабрикат заготовки являє собою діелектрик, вкритий фольгою, на яку за допомогою певних засобів нанесено малюнок печатного монтажу провідників. Далі іде наступна операція технологічного травлення, що розчиляє незахищену поверхню фольги, створюючи необхідну мережу з'єднань на поверхні діелектрика.

За способом монтажу відрізняють плити односторонні, двосторонні та багатошарові.

Односторонні плати є найпростішими за технікою виготовлення. Дані плати мають напрямні, надруковані лише на одній стороні дошки. Електронні складові вставляються в плату з тильного боку, не покриті фольгою.

Двосторонні плати Вони мають надруковані напрямні з обох сторін. У порівнянні з односторонніми, їх створення вимагає ьільки однієї лишньої технологічної операції – прометалювання отворів. Перехідні отвори забезпечують підключення друкованих проводів на протилежних сторонах плати. В результаті проблема уникнення перерізання друкованих проводів вирішується дуже легко, оскільки дроти прокладені паралельно один одному з обох сторін друкованої плати.

Їх структура нагадує сендвіч з окремо створених тонких двосторонніх дошок. Отож, крім двох наружних, вони також можуть мати ряд внутрішніх шарів провідників.

Основними елементами друкованої плати (рис. 10.1) є контактні площадки (1, 2) і друковані проводи (3).

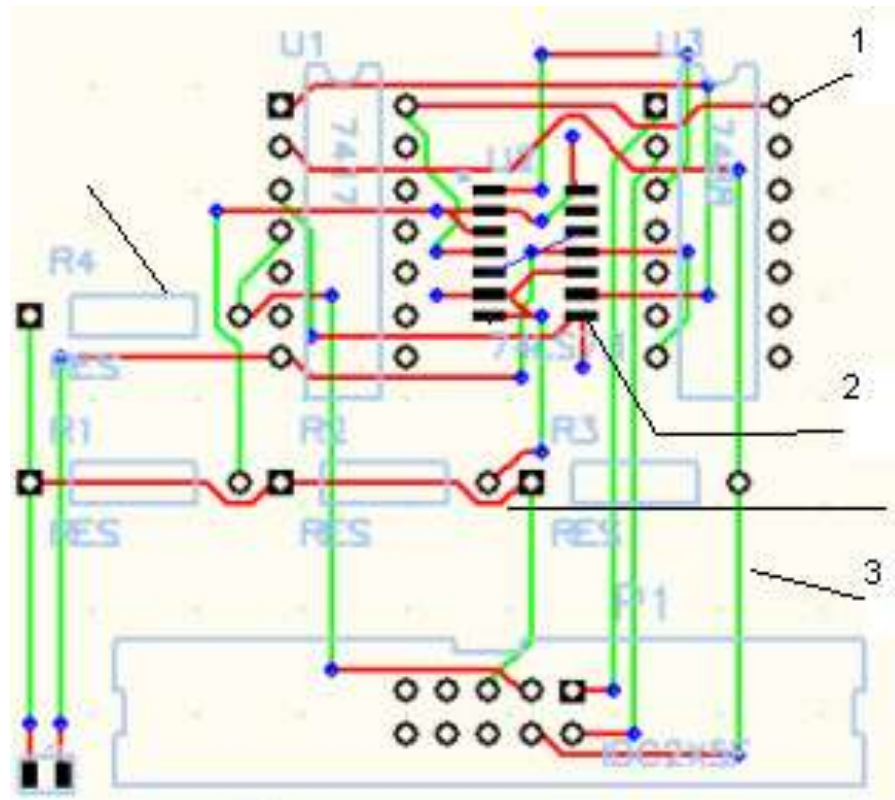


Рис. 10.1. Елементи друкованої плати

Макети розміщуються навколо монтажних отворів, в які вставляються компонентні дроти (1), через отвори для підключення друкованих проводів, а також у місцях припайки до плати плоских компонентних проводів (2) Для побудови зображень проводів і контактних майданчиків найчастіше використовується фотокоординатор.

Елементи з штирьовими клемами вставляють в отвори, розташовані на платі з одного боку так, щоб кінці клем виступали над поверхнею плати на 1 - з іншого боку 2 мм.

Планарні елементи більш мініатюрні в порівнянні з піновими, оскільки володіють меншою відстанню між выводами, а також їх можна встановлювати з обох сторін плати. Найсучасніші та відмінні компоненти виготовляються в корпусах без клем.

Такі комплектуючі набагато менше і дешевше звичайних. Вони виготовляються на довгій паперовій стрічці, намотаній на котушку. Це полегшує роботу-маніпулятору підбирати компонент.

Система автоматичного проектування CAD 2004

Система P-CAD2004 є одним із найпоширеніших інструментів проектування друкованих плат у світі. Призначений для роботи в середовищі Windows, має зручний інтерфейс. P-CAD 2004 має багато інструментів, наприклад, для переводу форм даних, взятих з інших програм, але всю роботу, пов'язану зі створенням схем і проектуванням плат, можливо виконати лише за допомогою двох або трьох програм, а саме:

- *Schematic* – редактор, що дозволяє створювати та редагувати діаграми;
- *PCB* – редактор для утворення та редагування друкованих плат;
- *Library Executive* – програма для праці з бібліотеками;
- *Редактор символів* і - редагувати зображення позначень елементів;
- *Редактор шаблонів* для створення та редагування зображень суцільних елементів.

10.2.1. Схематичний графічний редактор

Графічні модулі системи P-CAD 2004 володіють традиційною для Windows-додатків структуру та інтерфейс: у зовнішній частині вікна програми розташоване головне меню, внизу розташовані кнопки для найбільш часто виконуваних операцій, а внизу — рядок стану (рис. 10.2), в якому містяться повідомлення та дані про стан редактора

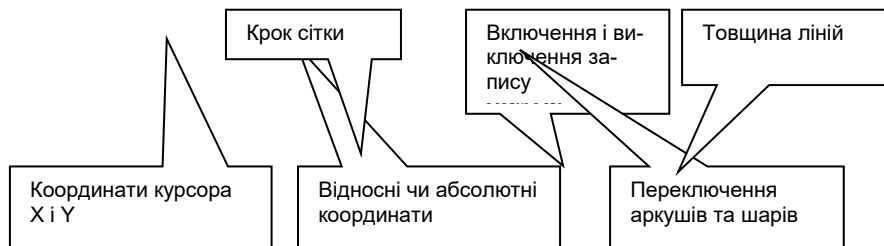
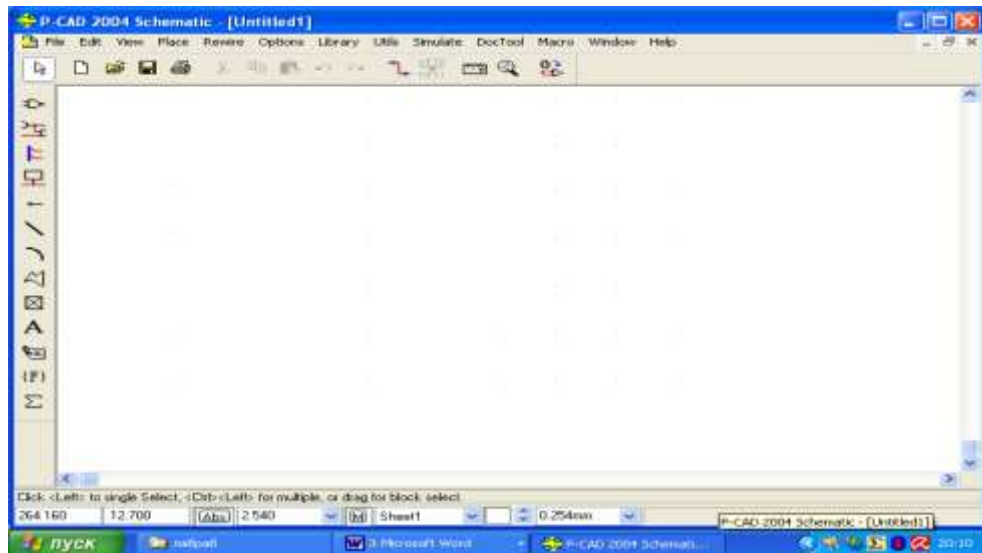


Рис. 10.2 Інтерфейс графічного редактора P-CAD 2004

Перші два поля рядка стану відображають інформацію про поточні координати X та Y курсора.

Кнопка ABS | Rel , залежно від стану, в якому попередні поля відображають визначені координати. У першому випадку відлік ведеться від нижнього лівого кута аркуша, у другому – від будь-якої точки, абсолютні координати якої можна задати у вікні, відкритому командою *Параметри > Обхвати*

Крок координатної сітки відображається праворуч від зазначеної кнопки, як наслідок, щільність розташування елементів у створюваному малюнку. Щоб налаштувати крок сітки, необхідно додати значення у вікні «Параметри сітки» (рис. 10.3), яке можна викликати з Параметри\Сітки.

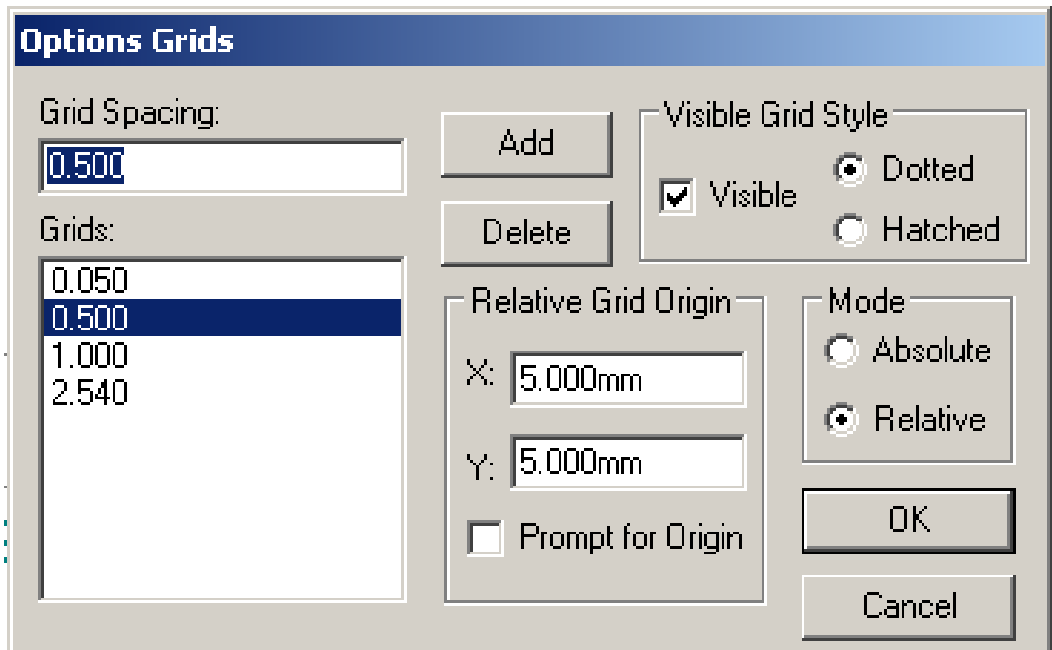


Рис. 10.3. Параметри вікна сітки

Значення кроку можна ввести в це поле за допомогою клавіатури або вибрати зі списку попередньо введених значень. Ви також можете ввести числові значення в поля координат X і Y, тоді курсор автоматично зміщується в потрібне місце на аркушу.

Ви можете змінити одиницю вимірювання на будь-якому етапі роботи з проектом, відкривши вікно конфігурації за допомогою команди *Параметри > Налаштувати*.

Зправа від поля шагу координатної сітки розташована кнопка **M**, натискання починає або зупиняє запис команд у макросі. Це робиться для швидкого повторення всіх команд, записаних в макросі. Для цього просто натисніть клавішу *E*.

Потім у рядку стану є два поля, які дозволяють вибрати схему схеми в редакторі *схем* або шар зображення в *редакторі друкованої плати*. Друге поле відображає колір, який використовується для відображення елементів у цьому шарі.

Останнє поле рядка стану визначає товщину лінії. З його правого боку відображається коротка інформація про обраний на екрані об'єкт. Вибір об'єктів і робота з ними відбувається за загальними принципами, прийнятими в Windows. Щоб отримати більш детальну інформацію про вибраний об'єкт і,

якщо необхідно, відредагувати його частину, клацніть правою кнопкою миші на екрані та виберіть у контекстному меню пункт «Властивість» .
Для перегляду зображень у графічних редакторах використовуйте клавіші переміщення курсору, смуги прокручування та команди меню «Вигляд».

10.2.2 Графічний редактор Schematic

Налаштування конфігурації. Перед створенням діаграми виконайте команду *Options > Configure* (рис. 10.4). Аркуш у наведених вище стандартних форматах завжди орієнтований таким чином, щоб його довша сторона була горизонтальною. Ви також можете встановити для аркуша будь-які розміри та орієнтацію, вибравши кнопку *спеціальних параметрів* і заповнивши поля *Ширина* та *Висота* .

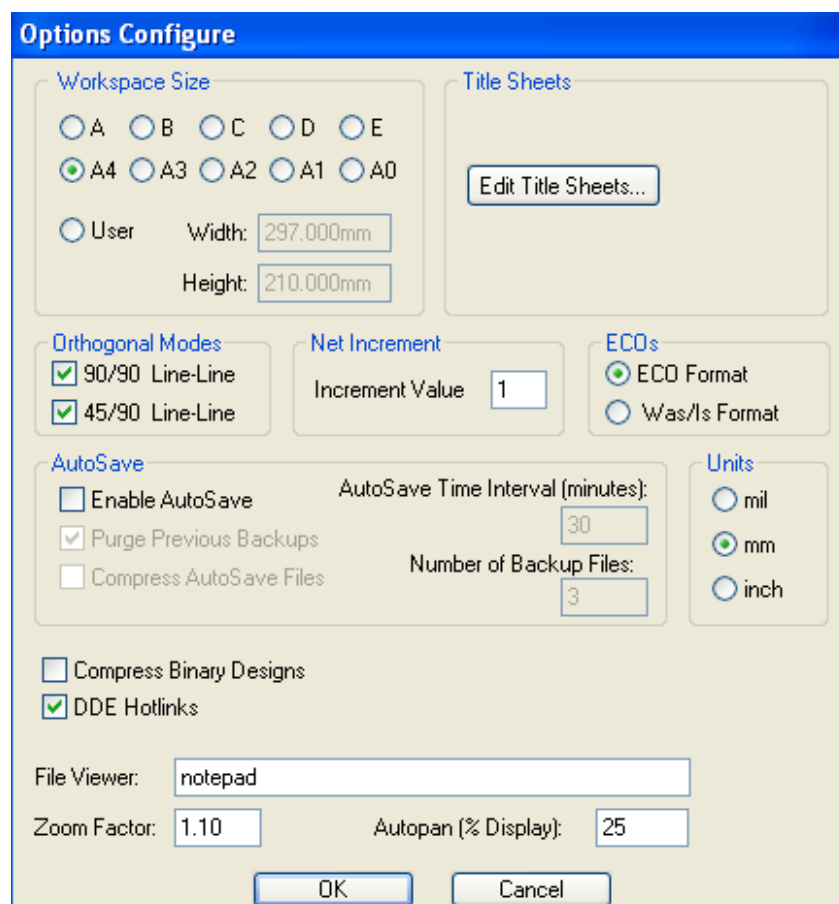


Рис. 10.4 Параметри вікна налаштування

Аркуші створюються, видаляються та перейменовуються за допомогою кнопок «Додати», «Видалити» та «Змінити», розташованих на вкладці «Аркуші» у вікні, відкритому за допомогою команди «Параметри» > «Аркуші». Аркуш, що відображається у вікні редактора, вибирається за допомогою кнопок рядка стану (рис. 10.4). Напрямні переміщуються від аркуша до аркуша за допомогою ліній групування.

Інші важливі параметри встановлюються у вікні «Параметри» > «Налаштувати», зокрема:

- *Ортогональні режими* визначають кути, під якими можна малювати лінії на діаграмі:
90/90 дозволяє малювати лінії лише під кутами, кратними 90° , а 45/90 також дозволяє малювати лінії під кутами, кратними 45° . Обмеження, встановлені за допомогою цього параметра, можна вмикати та вимикати клавішею *O*.
- *Чистий приріст* визначає, наскільки змінюється тег позиції компонента, коли його наступне входження розміщується на діаграмі. Наприклад, якщо *Increment Value = 1*, то після встановлення резистора *R1* його наступним входженням автоматично призначатимуться позиційні позначки *R2*, *R3* тощо.
- *Функція автозбереження* дозволяє встановити параметри автоматичного збереження файлів на диск, щоб запобігти втраті інформації у разі раптового відключення електроенергії або зависання комп'ютера.
- *Одиниці* визначають одиниці вимірювання: милі (mil), міліметри (mm) або дюйми (inches). Один мільйон 0,001 дюймадорівнює одному дюйму 2,54 мм.
- *DDE Hotlinks* встановлює «гарячий зв'язок» між редактором *схем* і *друкованою платою*, так що підсвічування *певних* компонентів або проводів в одному редакторі негайно викликає те саме в іншому.
- *Коефіцієнт масштабування* — це співвідношення збільшення або зменшення зображення під час натискання клавіш «+» і «-» на цифровій клавіатурі (сірі) або відповідних кнопок на панелі інструментів. Щоб забезпечити плавне налаштування масштабу зображення, рекомендуємо встановити *коефіцієнт масштабування = 1,1*.

- *Автопанорама* (% відображення) показує, на який відсоток від свого розміру прокрутиться зображення аркуша, якщо перемістити курсор за допомогою клавіатури за межі видимої частини вікна.

Внесення змін до конфігурації завершується натисканням кнопок *OK* або *Змінити* та *Закрити* . Натискання лише кнопки «*Закрити*» закриває вікно без внесення змін.

Розміщення символів. Звичайні зображення компонентів (чіпів, резисторів, роз'ємів тощо) у P-CAD називаються *символами* . Щоб розмістити на схемі необхідні символи, з'єднайте бібліотеки, виконайте команду *Бібліотека > Налаштування* та за допомогою кнопок *Додати ...* та *Видалити* створіть потрібний список бібліотек (рис. 10.5).

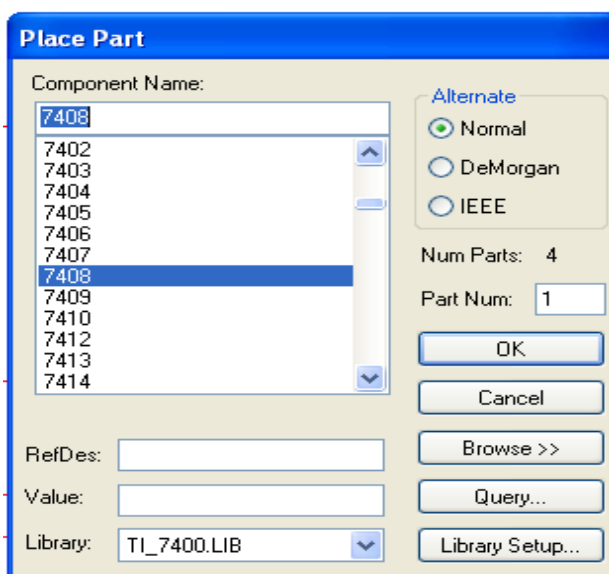


Рис. 10.5 Вікно вибору бібліотеки компонентів

У результаті відкриється діалогове вікно (рис. 10.6), у якому в полі *Бібліотека* можна вибрати бібліотеку та в полі *Назва компонента* компонент цієї бібліотеки .

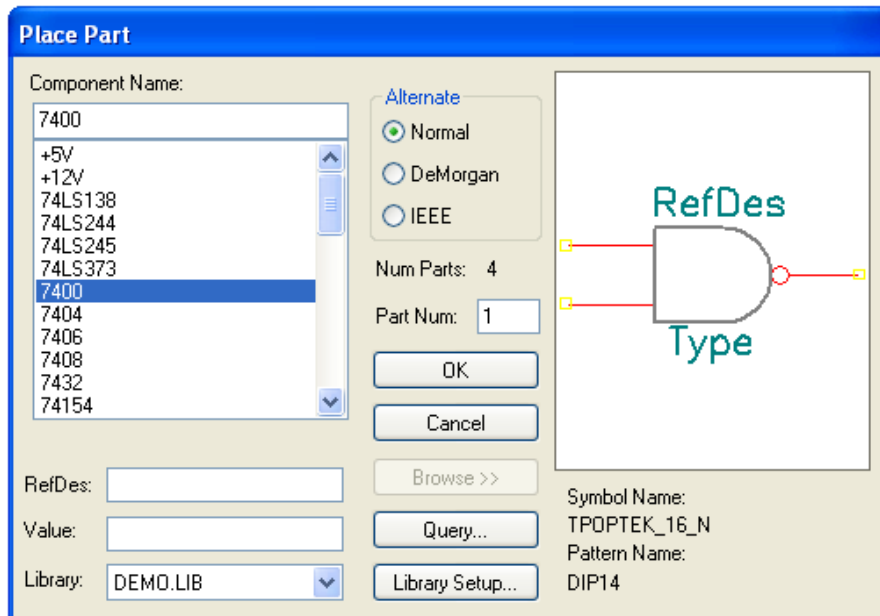


Рис. 10.5 Вікно вибору компонентів

Натискання кнопки *Огляд* відображає вибраний компонент у правій частині діалогового вікна. Перед розміщенням елемента на схемі можна заповнити поля *Посилання* та *Значення*, де вказується позиційне позначення елемента та його номінал.

Поле *RefDes* можна використовувати для заміни позначення США національними позначеннями.


У поле *Значення* введіть номінали резисторів, конденсаторів, кварцових резонаторів, марок діодів, транзисторів тощо.


Коли ви натискаєте *ОК*, діалогове вікно закривається, а курсор перетворюється на *X*. Екземпляри вибраного компонента буде вставлено скрізь, куди ви клацнете курсор. При цьому їх позиційні позначки будуть автоматично пронумеровані. Позначення позиції D можна вказати за допомогою поля *RefDes*. Що стосується цифрової нумерації перерізів, то це можна забезпечити відповідними налаштуваннями програми.

Вставлення екземплярів вибраного компонента завершується натисканням *правої* кнопки миші або клавіші *Esc*.

Потім ви можете вибрати та розмістити наступні компоненти на діаграмі за допомогою методу, описаного вище.

Зображення компонента можна *повернути або віддзеркалити* . Для цього виберіть компонент і натисніть *R* або відповідну клавішу Φ

Розташування ліній зв'язку. Для розміщення ліній електрозв'язку (проводів) натисніть кнопку  або виберіть у меню: *Місце > Кабель* .


Жодні інші лінії, наприклад ті, які розташовані за допомогою кнопки  не створюють електричних зв'язків між точками, які вони з'єднують.

Щоб розташувати провід, клацніть лівою кнопкою миші в тому місці, де він починається, де він зігнутий або з'єднаний з іншими проводами, і де дріт закінчується.

Коли ви закінчите впорядковувати цей посібник, натисніть праву кнопку миші або клавішу *Esc* , *перш ніж почати впорядковувати наступний* . Це означає, що, як ви могли помітити, натискання правої кнопки миші або клавіші *Esc* використовується в P-CAD для позначення виконання команди «багатопрхідної».

У P-CAD кінці проводів, які ні до чого не підключені, відображаються квадратами

 , а місця з'єднання проводів – крапками  .

При редагуванні траєкторії направляючої ключову роль відіграють її вузлові точки. Ви можете перемістити їх разом із сегментами, що їх з'єднують, або розірвати зв'язки між вибраними вузлами за допомогою клавіші *Delete* . У більш складних випадках може знадобитися встановлення додаткових вузлових точок на існуючих ділянках. Ось для чого призначена кнопка .

За замовчуванням напрямні мають імена *NET00001* , *NET00002* тощо. За потреби ви можете змінити назву напрямної або відобразити її на схемі. Для цього виберіть провідник, у контекстному меню виберіть пункт *Властивості* та заповніть відповідні поля в діалоговому вікні.

Створення файлу підключення. Схеми, створені в редакторі *схем* , зберігаються у файлах із розширенням *sch* .

Вони відкриваються та зберігаються на диску звичайними командами *Файл > Відкрити*, *Файл > Зберегти*, *Файл Зберегти як ...*. Перенесення інформації з графічного редактора *Схеми*, в якому створюється схема пристрою, до графічного редактора *друкованої плати*, де вона друкується. Плата буде розроблена, виконується за допомогою файлів підключення, які зазвичай мають розширення *net* або *alt* і містять перелік елементів на схемі та опис з'єднань між їхніми виходами.

Щоб створити файл підключення в *графічному редакторі схеми*, виберіть у меню «*Інструменти*» > «*Створити список мереж*». У результаті відкриється діалогове вікно, в якому вибирається ім'я файлу підключення та його формат. Ім'я файлу за замовчуванням і папку розташування можна змінити за допомогою кнопки «*Ім'я файлу списку мережі*» (Малюнок 10.7).

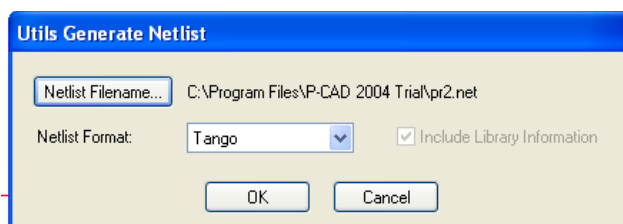


Рис. 10.7 Вікно створення файлу підключення

Вибір формату файлу підключення залежить від вибору роутера - програми, яка буде розташовувати друковані кабелі. Для таблиць низької складності використовуйте формат *Tango*.

10.2.3 Робота з графічним редактором РСВ

Налаштування конфігурації. Основні параметри конфігурації редактора *RSV* налаштовуються за допомогою команди «*Параметри*» > «*Налаштувати*». Це одиниця виміру довжини, розміру аркуша (його слід обирати трохи більше розміру дошки) тощо. Під час роботи з редактором *RSV* важливі параметри екрана, встановлені за допомогою команди «*Параметри*» > «*Відобра*

зити». Відкриває однойменне діалогове вікно, яке містить таблицю, у якій стовпці відповідають різним елементам зображення (*Elements*), а його шари (*Layers*) — рядкам. Наприклад, за замовчуванням елементи зображення, такі як лінії, багатокутники та тексти, розташовані на рівні Top, *тобто* у верхній частині дошки (з боку встановлення компонента), і такі ж елементи з *проти-лежного боку дошки* (*нижній шар*) відображаються червоним кольором – колір зелений



Список імен і призначень для елементів зображення та шарів зображення, який встановлюється за замовчуванням під час створення нової дошки:






- Через- Наскрізні отвори
- Pad – контактні площадки
- Лінія – провідники та лінії
- Багатокутники
- Текст - Текст
- Верх – верхня сторона ПП
- Нижня - нижня сторона ПП
- Широкий контур
- Паяльна маска
- Шовкотрафаретний друк
- Паста – паяльна паста на верхній стороні

Цей набір шарів описує двосторонню дошку. Інші шари, включаючи додаткові шари провідників, які використовуються для створення багатошарових плат, *користувач може додати* (*Add*) або видалити (*Remove*) у діалоговому вікні *Layers Options* , де відображається інформація про призначення, стани та кількість шарів. наступним чином: *Стовпець шарів* .

Буква *S* (*Signal*) позначає сигнальні шари, в яких розміщені друковані провідники, буква *P* - (*Flat*) - суцільні шари металізації для екранування та живлення, а літера *N* (*Nonsignal*) - інші несигнальні, допоміжні шари.

D і E відповідно. За допомогою кнопок *Вимкнути* і *Увімкнути* можна вимкнути або увімкнути відображення будь-якого вибраного шару, крім поточного (*Поточний*), а кнопками *Вимкнути все* і *Увімкнути все* відображати всі шари дошки. Бажаний напрямок розташування кабелю в сигнальних шарах позначається літерами: *H* - горизонтальне, *V* - вертикальне, *A* - вибирається автоматично (за замовчуванням).

Методи розробки друкованих плат. Редактор *RSV* дозволяє проектувати друковані плати без схем. Щоб зробити це, ви можете скористатися кнопкою  «*Розмістити*» > «*Компонент*» або командою , щоб розмістити потрібні вам компоненти на дошці, і скористатися кнопкою  «*Розмістити*» > «*З'єднання*» або командою для встановлення з'єднань між їхніми виходами. Потім можна запустити програму маршрутизатора, яка правильно розташує шляхи надрукованих кабелів.

Ви навіть можете розробити свою друковану плату вручну від початку до кінця. Для цього потрібно вручну розмістити контактні площадки компонента на платі  (*Місце* > *Панель*), а потім вручну знову з'єднати їх із надрукованими напрямними у вигляді ліній  (*Місце* > *Лінія*), дуг  (*Місце* > *Дуга*) і багатокутників.  (*Місце* > *Багатокутник*). Звичайно, дроти доведеться прокладати різними шарами, з'єднуючи їх за допомогою перехідних отворів, якщо необхідно  (*Місце* > *Наскрізне*). Ця технологія створення друкованих плат підходить тільки для найпростіших схем,

Автоматизоване створення плати передбачає передачу інформації з графічного редактора *Schematic* , де була створена принципова принципова схема, до графічного редактора *друкованої плати* за допомогою файлу підключення.

Перед завантаженням цього файлу в редакторі *PCB* необхідно об'єднати бібліотеки всіх елементів на діаграмі

Зв'язування бібліотек виконується так само, як і в *редакторі схем*. Виконайте команду *Бібліотека* > *Налаштування та* за допомогою кнопок *Додати*

та Видалити *створить потрібний* список бібліотек (рис. 10.8). P-CAD 2004 має спільні бібліотеки для редакторів *схем* і *RSV*.

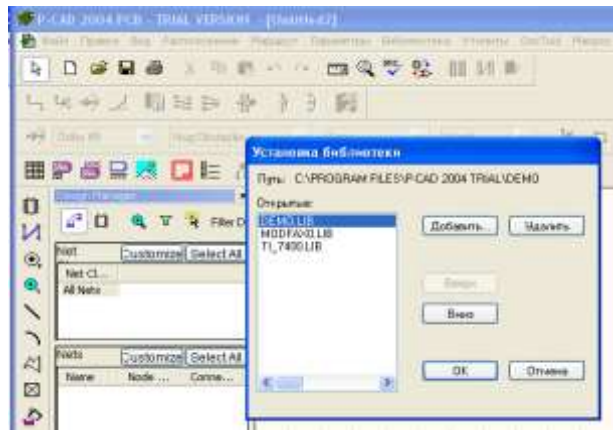


Рис. 10.8 5 Вікно вибору бібліотеки корпусу

Файл підключення завантажується за допомогою команди *Utils > Generate Netlist* (Малюнок 10.9).

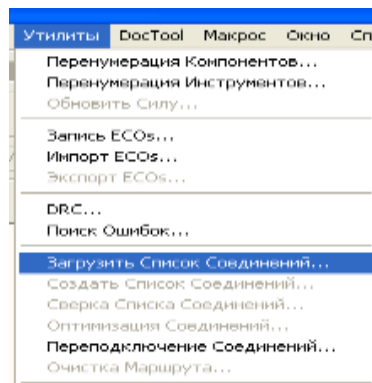


Рис. 10.9 Вікно вибору файлу підключення

У результаті відкриється діалогове вікно, у якому можна вибрати назву та тип файлу (рис. 10.10).

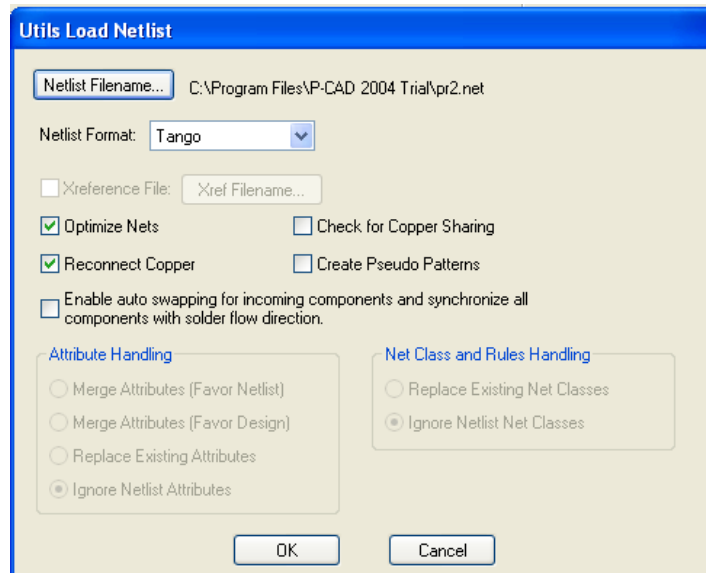


Рис. 10.10 Вікно завантаження файлу підключення


Коли файл з'єднання успішно завантажено, на екрані відображається вигляд елементів схеми, а з'єднання між їхніми виходами відображаються у вигляді відрізків.

Розташування елементів на дошці. Перш ніж розміщувати компоненти на платі, у рядку стану або в меню «*Параметри*» > «*Сітка*», необхідно встановити крок сітки, який дорівнює відстані між контактами компонента.

При проектуванні друкованих плат дуже важливо оптимальне розташування компонентів на платі. Це залежить від довжини друкованих кабелів, кількості перехідних отворів, інших особливостей, що визначають якість і надійність плати, і можливості її виготовлення в заданих розмірах.

До складу P-CAD 2004 входить програма SPECCTRA, яка вирішує завдання оптимального розташування елементів на платі та подальшого розміщення кабелю (трасування).

Після завантаження файлу підключення всі елементи діаграми будуть відображені на екрані. З'єднання клем компонентів показано прямими лініями. Якщо вказано розміри дошки, намалюйте контури дошки на шарі *дошки* поруч із елементами або навколо них. Якщо розміри плати можуть бути довільними, її контури можна побудувати пізніше, після завершення розміщення компонентів, а деякі маршрутизатори, такі як *QuickRoute*, дозволяють це зробити навіть після трасування.

Щоб вручну змінити розташування компонента на дошці, виберіть компонент у режимі вибору , клацнувши на ньому мишею та перетягнувши мишу в потрібне місце. Коли компонент переміщується, його зв'язки з іншими компонентами підсвічуються разом з ним. Розташування елементів слід підбирати таким чином, щоб довжина з'єднань і кількість точок їх перетину були якомога меншими.

Вибраний компонент обертається навколо своєї осі за допомогою *клавіші F*, що фактично означає переміщення його на протилежну сторону дошки. Як відомо, розташування компонентів по обидва боки плати стосується лише компонентів із плоскими контактами.

Щоб допомогти контролювати маршрутизацію певних ліній електричного зв'язку під час розміщення компонентів, команда «Правка» > «Мережі» надає можливість увімкнути або вимкнути відображення кожного з них (Малюнок 10.11).

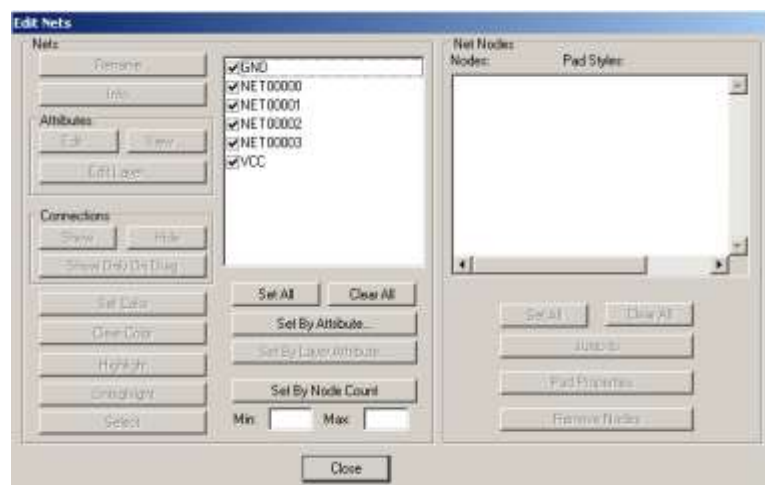


Рис. 10.11 Вікно редагування файлу підключення

Для цього можна встановити або зняти прапорець біля назви лінії зв'язку в графі *Мережі* або вибрати потрібну групу ліній і скористатися кнопками *Приховати підключення* та *Показати підключення*.


Кнопки «Виділити» та «Оновити» використовуються для виділення виділених рядків кольором, а також для інших корисних функцій. Наприклад,

вибравши лінію зв'язку в *стовпці «Мережі»*, у стовпці *«Вузли»* ви побачите список виходів компонентів, до яких вона підключена, а вибравши вихід у стовпці *«Вузли»* та натиснувши кнопку *«Перейти до вузла»*, ви може встановити курсор на цей вивід.

Після того, як ви розмістили компоненти на платі, вам потрібно оптимізувати мережу взаємозв'язку, переставивши логічно еквівалентні розділи та виходи, що виконується за допомогою команди *«Інструменти» > «Оптимізувати мережі»*. У меню команд виберіть *Авто* (автоматична оптимізація).

Потім збережіть створений файл на диск. Отриманий файл має стандартне розширення *pcb*, але може бути як у двійковому (*двійковий файл *.pcb*), так і в текстовому (*файл ASCII *.pcb*) форматах.

Автоматичне відстеження платежів. Маршрутизація - це процес прокладання шляхів друкованих кабелів, який виконується спеціальною програмою - маршрутизатором. Щоб виконати трасування, вам потрібно відкрити *файл pcb*, в якому компоненти розміщені на платі. Якщо *планується використання маршрутизаторів PRO Route або SPECCTRA*, друковані плати мають бути намальовані на *шарі плати*.

Якщо на дошці є певні зони, через які друковані напрямні не повинні проходити, контури цих областей, які називаються маршрутними бар'єрами, слід намалювати на дошці за допомогою команди *«Розмістити» > «Захист»* або піктограми . Перед застосуванням бар'єру команда *«Параметри» > «Поточне навантаження»* визначає його параметри: чи є бар'єр лінією чи багатокутником і чи застосовується бар'єр лише до поточного шару проводів чи до всіх шарів, як у випадку з отвором у дошка, наприклад.

Вікно маршрутизатора (рис. 10.12) відкривається *командою Route Autorouter*.

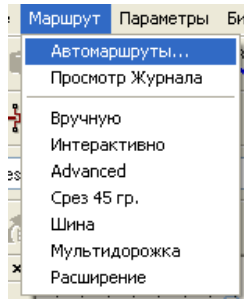


Рис. 10.12 Вибір методу маршрутизації

Перш за все, в полі *Autorouter* виберіть тип маршрутизатора. Радимо вибрати *Quick Route* - просту програму, яка дозволяє відстежувати плати з не більше чотирьох шарів металізації. Нижче у вікні *Route Autorouters* вказані стандартні назви файлу стратегії маршрутизації та вхідних і вихідних файлів (Мал. 10.12). Файл політики описує правила, за якими відбувається маршрутизація.

Маршрутизатор *Quick Route* може являти собою файл друкованої плати, в якому вже прокладено певну кількість проводів. Потім роутер, не вносячи жодних змін у вже прокладені провідники, відстежує решту. Ця функція цього маршрутизатора дозволяє виконувати поетапне перетасування, коли це необхідно. Це полегшується тим, що вихідний файл має той самий формат і те саме розширення, що й вхідний файл — *pcb* (рис. 10.13).

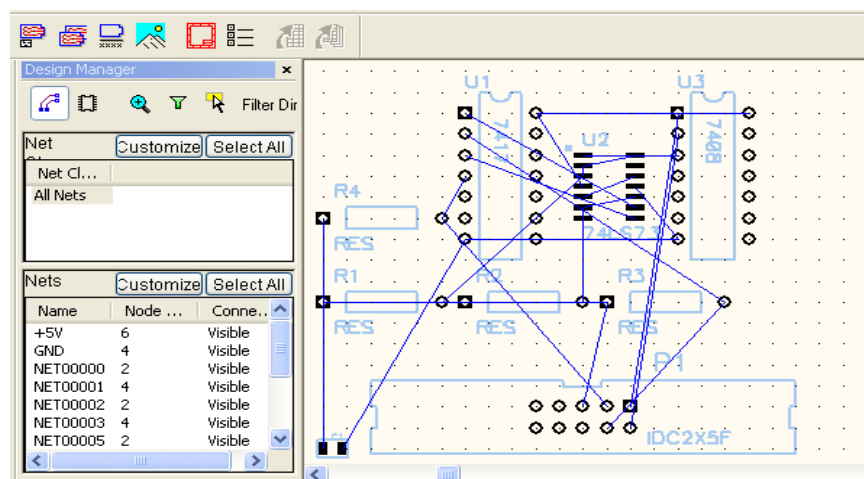


Рис. 10.13 Вікно результатів завантаження файлу підключення

За замовчуванням ім'я вихідного файлу створюється з назви вхідного файлу шляхом додавання літери *R* перед ним. Звичайно, маршрутизатор не може відтворити вихідний файл, якщо його попередня версія має таке саме ім'я знаходиться в тій же папці, відкривається редактором *RSV* або іншою програмою. У цьому випадку роутер відобразить повідомлення про те, щоб закрити цей файл в інших вікнах.

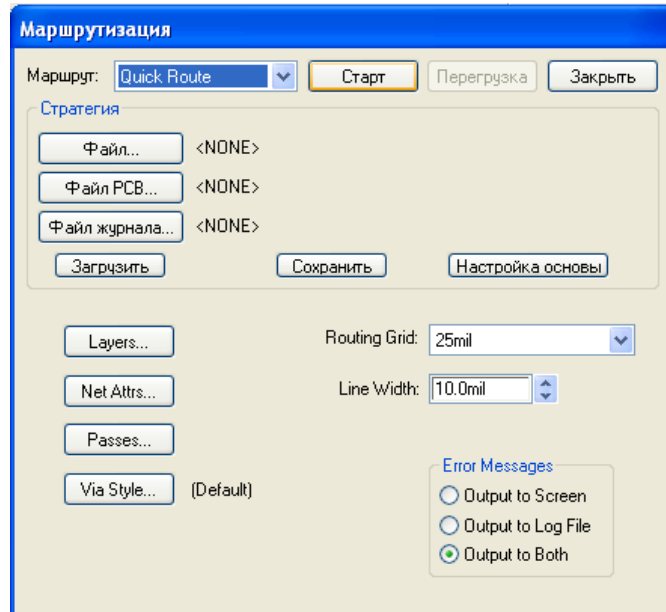


Рис. 10.14. Вибір способу маршрутизації

У нижній частині вікна *Route Autorouters* розташовані кнопки для зміни параметрів шарів (*Layers*), напрямних (*Net Attrs*), маркування переходів (*Passes*) і наскрізних отворів (*Via Style*), а також поля, що дозволяють вибрати ступінь сітки маршрутизації (*Routing Grid*), ширини проводу (*Line Width*) і місця відображення інформації про помилку (*Error Messages*) Рис. 10.14.

Якщо живлення подається на мікросхеми через друковані провідники, а не через суцільні металізовані шари, то ці провідники товщі за інші, тому що через них тече більше струму. Крім того, зменшення опору силових ліній сприяє зниженню рівня перешкод, які поширюються по ним в результаті зміни струму, споживаного мікросхемами.

Щоб збільшити товщину цих провідників, ви повинні надати їм відповідні атрибути перед прокладкою. Для цього у вікні *Route Autorouters* натисніть кнопку *Net Attrs* або виконайте команду *Edit > Networks*. У стовпці *Мережі* виберіть назви ліній живлення: *GND*, *+5V* тощо. Потім у вікні *Правка > Мережі* натисніть кнопку *Редагувати атрибути*, у вікні *Атрибути* натисніть кнопку *Додати*, а у стовпці *Ім'я* у вікні «*Атрибути місця*» виберіть «*Ширина*» та в полі «*Значення*» введіть бажану ширину цих проводів, принаймні 20 миль.

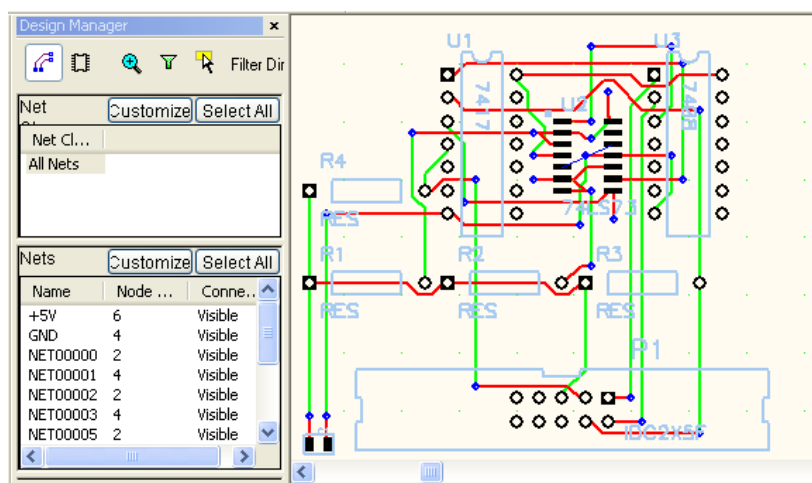


Рис. 10.15. Результат створення друкованої плати

Автоматична маршрутизація плати починається натисканням кнопки «*Пуск*» у вікні «*Маршрутизація автомаршрутизаторів*». Перебіг і результати відстеження можна спостерігати у вікні редактора *RSV*, де вхідний файл автоматично закривається, а вихідний файл відкривається (Малюнок 10.15).

Список літератури

1. Абрамов В.О. Базові технології комп'ютерних мереж: навч. посіб. / В.О. Абрамов, С.Ю. Клименко. - К.: Київ, ун-т ім. Б. Грінченка, 2011. - 291 с.
2. Буров Є.В. Комп'ютерні мережі: підруч. - Львів: Магнолія плюс, 2006. - 264 с.
3. Буров Є. Комп'ютерні мережі. 2 - ге оновлене і доповн. вид. – Львів: БаК, 2003. – 584 с. 4. Кулаков Ю.О., Луцький Г.М. Комп'ютерні мережі. Підручник / За ред. Ю.С. Ковтанюка. –К.: Юніор, 2003. –400 с.
4. Теорія і проектування комп'ютерних систем і мереж (Частина 1. Проектування систем обробки та захисту інформації) Навчальний посібник / Кулініч О.М., Касаткін Д.Ю., Лахно В.А., Сагун А.В. – К.: Видавництво, 2023 – 418 с.
5. Сучасна елементна база проектування цифрових систем. Навчально-методичний посібник / Касаткін Д.Ю., Кулініч О.М., Лахно В.А. – К.: Видавництво, 2022 – 168 с.
6. Завадський В. А. Комп'ютерна електроніка. – К.: ВЕК, 1996. – 236 с.
7. Лебедев О. М., Сидорів О. М., Ладик О. І. Цифрові пристрої телекомунікаційних послуг / за ред. М.Ю. – К.: КВІУЗ, 1999. – 182 с.
8. 10. Лебедев О. М., Ладик О. І. Цифрова техніка. - К.: Політехніка, 2004. —316 с.