

**Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В.**

# **Схемотехніка: Пристрої цифрової електроніки**

**ТОМ 1**

**Рекомендовано Методичною радою НТУУ «КПІ»  
як електронний підручник для студентів,  
що навчаються за спеціальністю «Електроніка»**

Київ 2016

Рецензенти: *М.М. Юрченко*, д-р техн. наук, проф. (Інститут електродинаміки НАН України)

*В.В. Каплун*, д-р. техн. наук, проф. (Київський національний університет технологій та дизайну)

Рябенський В.М. Жуйков В.Я. Ямненко Ю.С. Заграничний А.В. „Схемотехніка: Пристрої цифрової електроніки”. Електронний підручник для вищих навчальних закладів. Електронний підручник “СХЕМОТЕХНІКА: ПРИСТРОЇ ЦИФРОВОЇ ЕЛЕКТРОНІКИ” містить в собі фундаментальний матеріал призначений для вивчення методів перетворення та обробки цифрових сигналів на основі апаратних засобів – мікросхем невисокого рівня інтеграції, які є основою для побудови складних цифрових електронних систем різноманітного функціонального призначення- від комунікаційних і комп’ютерних до пристроїв побутової техніки. Він призначений для студентів, що навчаються за напрямком “Електроніка” (06.08.09.00). Підручник може бути повністю використаний студентами, що навчаються за напрямками ”Електротехніка”, “Електромеханіка”, “Прилади”, “Комп’ютерні науки”, “Комп’ютерна інженерія” для вивчення відповідних розділів електронної техніки. Дисципліна “ЦИФРОВА ЕЛЕКТРОНІКА” має не тільки важливе самостійне практичне значення, а і є базою при вивченні мікропроцесорної техніки. Мектодологічно підручник побудований таким чином, щоб значно полегшити студентам вивчення мікропроцесорної техніки.

Автори:

Рябенський Володимир Михайлович, д-р. техн. наук

Жуйков Валерій Якович, д-р техн. наук

Ямненко Юлія Сергіївна, д-р. техн. наук

Заграничний Артур Володимирович

## З М І С Т

### ВСТУП

### Розділ 1. СИГНАЛИ ТА ЇХ ПЕРЕТВОРЕННЯ

<b>1.1. Типи сигналів</b>	
1.1.1. Аналогові сигнали.....	9
1.1.2. Імпульси, імпульсні послідовності.....	11
1.1.3. Оцифрування аналогових сигналів.....	16
<b>1.2. Системи числення</b>	
1.2.1. Основні визначення.....	19
1.2.2. Переведення чисел з однієї позиційної системи числення в іншу.....	21
1.2.3. Переведення цілого числа з десяткової системи числення в $P$ -ічну.....	22
<b>1.3. Коди та їх характеристика</b>	
1.3.1. Коди з паралельною формою представлення інформації.....	25
1.3.2. Послідовні формати передачі даних.....	33
<b>1.4. Форми зображення чисел</b> .....	36
<b>1.5. Виконання арифметичних операцій</b> .....	40
<b>1.6. Основи алгебри логіки</b>	
1.6.1. Основні визначення.....	46
1.6.2. Закони та тотожності алгебри логіки.....	50
1.6.3. Способи задання логічних функцій.....	52
1.6.4. Мінімізація логічних функцій.....	63
<b>1.7. Коди, що знаходять та виправляють помилки</b>	
1.7.1. Особливості кубічної форми представлення логічних функцій.....	71
1.7.2. Коди з виявленням і корекцією помилок.....	73
1.7.3. Коди, що коригують одиночні помилки і виявляють помилки більшої кратності.....	74
1.7.4. Двовимірні коди.....	78
<b>1.8. Завади та їх характеристики</b> .....	78
<b>Контрольні питання</b> .....	87
<b>Вправи і завдання</b> .....	91

### Розділ 2. СТРУКТУРИ БАЗОВИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ

<b>2.1. Характеристика цифрових сигналів</b> .....	103
<b>2.2. Транзисторні ключові схеми</b>	
2.2.1. Ключі на біполярних транзисторах.....	106
2.2.2. Ключі на польових транзисторах.....	116
<b>2.3. Технічні характеристики базових логічних елементів і особливості їх використання</b>	
2.3.1. Транзисторно-транзисторна логіка (ТТЛ).....	123
2.3.2. Особливості використання мікросхем ТТЛ.....	129
2.3.3. Мікросхеми КМОН-структури.....	134
2.3.4. Особливості експлуатації мікросхем КМОН.....	150
2.3.5. Базові елементи БіКМОН-технології.....	159
2.3.6. Інтегральна інжекційна логіка.....	163
2.3.7. Базові елементи емітерно-зв'язаної логіки (ЕЗЛ).....	167

2.3.8. Прилади з переносом заряду.....	170
2.3.9. Арсенід-галієві логічні елементи.....	175
2.3.10. Логічні ключі, що використовують збіднені MESFETи.....	178
2.3.11. Логічні елементи на основі діодів Шоткі (SDFL).....	179
2.3.12. Буферизована FET-логіка.....	180
<b>2.4. Моделі логічних елементів.....</b>	<b>180</b>
<b>Контрольні питання.....</b>	<b>183</b>
<b>Вправи і завдання.....</b>	<b>186</b>

### **Розділ 3. ПРИНЦИПИ ПОБУДОВИ ТА ВИКОРИСТАННЯ КОМБІНАЦІЙНИХ ПРИСТРОЇВ**

<b>3.1. Використання базових логічних елементів.....</b>	<b>189</b>
<b>3.2. Комутатори</b>	
3.2.1. Мультиплексори.....	194
3.2.2. Пристрої зсуву.....	202
3.2.3. Селектори-мультиплексори.....	203
<b>3.3. Кодуючі та декодуючі пристрої</b>	
3.3.1. Дешифратори-демультиплексори.....	211
<b>3.4. Обчислювальні пристрої</b>	
3.4.1. Суматори.....	219
3.4.2. Перемножувачі.....	225
3.4.3. Компаратори.....	226
3.4.4. Арифметично-логічні пристрої.....	228
<b>3.5. Пристрої контролю</b>	
3.5.1. Контроль парності.....	231
3.5.2. Корекція помилок.....	234
3.5.3. Контроль логічних перетворень.....	239
<b>3.6. Буферні елементи.....</b>	<b>241</b>
<b>3.7. Перехідні процеси в логічних схемах.....</b>	<b>242</b>
<b>Контрольні питання.....</b>	<b>250</b>
<b>Вправи і завдання.....</b>	<b>252</b>

### **Розділ 4. ТРИГЕРИ. ТРИГЕРНІ СХЕМИ**

<b>4.1. RS-тригери.....</b>	<b>264</b>
<b>4.2. Загальна характеристика тригерних схем.....</b>	<b>271</b>
<b>4.3. D-тригери.....</b>	<b>277</b>
<b>4.4. JK-тригери.....</b>	<b>287</b>
<b>4.5. T- та TV-тригери.....</b>	<b>294</b>
<b>4.6. Несиметричні тригери.....</b>	<b>297</b>
<b>4.7. Синхронізація в цифрових схемах</b>	
4.7.1. Способи синхронізації та особливості їх використання.....	300
4.7.2. Узгодження вхідних сигналів.....	307
4.7.3. Синхронізатори.....	309
4.7.4. Практичні проблеми проектування синхронних цифрових схем.....	313
<b>Контрольні питання.....</b>	<b>316</b>
<b>Вправи і завдання.....</b>	<b>317</b>

## Розділ 5. СКІНЧЕННІ АВТОМАТИ

<b>5.1. Загальна характеристика скінченних автоматів.....</b>	<b>325</b>
<b>5.2. Автомати Мілі і Мура</b>	
5.2.1. Закони функціонування автоматів.....	329
5.2.2. Способи опису роботи автоматів.....	332
5.2.3. Еквівалентні перетворення автоматів.....	336
<b>5.3. Основи аналізу цифрових автоматів</b>	
5.3.1. Аналіз автоматів з <i>D</i> -тригерами.....	338
5.3.2. Особливості аналізу скінченних автоматів з <i>JK</i> -тригерами.....	346
<b>5.4. Синтез скінченних автоматів</b>	
5.4.1. Основи синтезу скінченних автоматів.....	350
5.4.2. Синтез асинхронних імпульсних автоматів.....	380
5.4.3. Особливості синтезу синхронних автоматів.....	382
5.4.4. Використання теореми Шенона при синтезі скінченних автоматів на основі <i>JK</i> -тригерів.....	386
<b>Вправи і завдання.....</b>	<b>388</b>
<b>Перелік умовних скорочень.....</b>	<b>394</b>
<b>Література.....</b>	<b>397</b>

## ВСТУП

Ера цифрової напівпровідникової електроніки почалась у 60-ті роки минулого століття з появою резистивно-транзисторної логіки (РТЛ), розробленої американською фірмою Fairchild. При використанні РТЛ виявилися переваги цифрової форми обробки сигналів. Але недоліки цієї технології досить швидко привели до витіснення РТЛ більш надійною діодно-транзисторною логікою (ДТЛ). Остання у вітчизняній практиці була реалізована в серії “Логіка Т” та інших. Досить швидко на її зміну прийшла транзисторно-транзисторна логіка (ТТЛ) як більш надійна і маюча значно менші масо-габаритні показники. Авторство ТТЛ належить фірмі Silvania, але масове використання ТТЛ інтегральні схеми знайшли після створення стандартних серій. Перша масова стандартна серія 74xx була розроблена фірмою Texas Instrument. Технічні характеристики різних серій ТТЛ розкрили безмежний простір у розвитку цифрової схемотехніки, що в свою чергу приводило до подальшого зростання вимог до логічних елементів на біполярних транзисторах, до їх схемотехніки.

Наприкінці 60-х років з’явилась також логіка на польових транзисторах. Спочатку це були р-МОН і n-МОН технології, які мали важливі технологічні переваги, завдяки чому почали використовуватись при виготовленні складних і великих інтегральних схем (ВІС). За ними досить швидко з’явилась КМОН-технологія, яка в статичних режимах практично не споживала електроенергії (розробник – фірма RCA). Перші КМОН ІС мали низькі робочі частоти, були досить чутливими до статичної електрики і мали несумісні з ТТЛ рівні логічних сигналів. Але мала споживана потужність на низьких робочих частотах робили її привабливою в апаратурі з батарейним живленням, що і спонукало до її розвитку.

Протягом 70-х років основні напрямки розвитку ІС на біполярних і польових транзисторах інтенсивно розвивались. Були розроблені більш

розвинуті ТТЛ ІС на транзисторах Шоткі (ТТЛШ), з'явилися і інтенсивно розвивались ІС емітерно-зв'язаної логіки (ЕЗЛ), розвивалась інжекційна логіка (ІЛ – інтегральна інжекційна логіка) та інші. Розвиток інтегральної схемотехніки відбувався на якісному рівні – узгоджувались рівні вхідних і вихідних сигналів, перешкодостійкість, підвищувались робочі частоти, але КМОН ІС значно відставали за частотними властивостями від логіки на біполярних транзисторах.

У 80-ті роки вперше з'явилась серія КМОН ІС (74НС), яка за своїми частотними властивостями не відставала від логіки на біполярних транзисторах. За нею виникли ще більш досконалі, з більшими робочими частотами, більш технологічні, що привело до використання КМОН-технологій для виготовлення мікроконтролерів, однокристальних ЕОМ, програмованих логічних матриць, які в останньому десятиріччі витіснили з практики дискретні компоненти невисокого рівня інтеграції.

Сучасний рівень електронної техніки в значній мірі визначається розвитком технологій цифрової схемотехніки. Зменшення розміру дискретного транзистора і збільшення площі використаних кремнієвих пластин дають можливість забезпечувати схемотехнічну реалізацію алгоритмів досить високої складності. Це дало поштовх у розвитку комп'ютерної техніки, телекомунікацій, мікропроцесорної техніки для управління складними технологічними процесами, побутової електроніки, пристроїв енергетичної електроніки. Поряд з розвитком технологій, інтенсивно вдосконалюються програмні засоби, які значно полегшують задачі проектування і моделювання складних електронних пристроїв на базі мікроконтролерів, ОЕОМ, програмованих логічних матриць.

Оволодіння цими досягненнями можливе лише на основі знань фундаментальних основ побудови пристроїв і алгоритмів цифрової електроніки, базових алгоритмів функціонування цифрових систем і умінь грамотно і коректно їх використовувати у взаємозв'язку.

Пропонована книга призначена для вивчення основ цифрової схемотехніки, її базових алгоритмів з різноманітних аспектів їх використання для розв'язання практичних задач.

Автори висловлюють щирю подяку асистенту кафедри теоретичної електротехніки та електронних систем Національного університету кораблебудування ім. адмірала Макарова (м. Миколаїв)

**Буряку Володимиру Святославовичу**

за надану допомогу у створенні підручника.



## Розділ I

### СИГНАЛИ ТА ЇХ ПЕРЕТВОРЕННЯ

#### 1.1. Типи сигналів

##### 1.1.1. Аналогові сигнали

Принципи побудови електронних пристроїв, режими роботи приладів залежать від характеру сигналів, які підлягають обробці.

*Сигналом* називають процес зміни у часі фізичного стану деякого параметру електричного пристрою, який [параметр] використовується для відображення, реєстрації та передачі повідомлень. Повідомлення нерозривно пов'язані з закладеною у них інформацією.

Сигнали, що використовуються в електронній техніці, розділяються на *аналогові, дискретні (імпульсні) та цифрові*.

*Аналогові сигнали* описуються неперервними функціями напруги, струму тощо. Джерелами аналогових сигналів є датчики різноманітних фізичних величин. У залежності від характеру зміни, сигнали розділяються на постійні (такі, що повільно змінюються у часі) та змінні. Носієм інформації в постійних сигналах є його рівень відносно нуля напруги чи струму.

Математична модель детермінованого змінного сигналу в загальному вигляді описується формулою:

$$x(t) = \sum_{n=0}^{N-1} X_{m,n} \sin(\omega_n t + \varphi_n), \quad (1.1)$$

де  $X_{m,n}$ ,  $\omega_n$ ,  $\varphi_n$  – відповідно, амплітуда, частота та початкова фаза  $n$ -ої гармоніки;  $N$  – кількість гармонік.

Інформація, що передається сигналом (1.1), як правило, міститься в амплітудах складових. Сигнали (1.1) одержуються та перетворюються (підсилюються), наприклад, у звуковідтворюючій апаратурі. У системах автоматики та радіотехніки часто використовується моногармонічний сигнал:

$$x(t) = X_{m,1} \sin(\omega_1 t + \varphi_1), \quad (1.2)$$

який може використовуватись як базовий для передачі інформації. При цьому інформація, що передається, може міститись у будь-якому з його параметрів – амплітуді, частоті, фазі – або одночасно в декількох з них.

Процес цілеспрямованої зміни параметрів базового сигналу за законом зміни інформації, що передається, називається *модуляцією*. Базовий сигнал при цьому називається *несучим*, а функція (1.2) – *несучою функцією*.

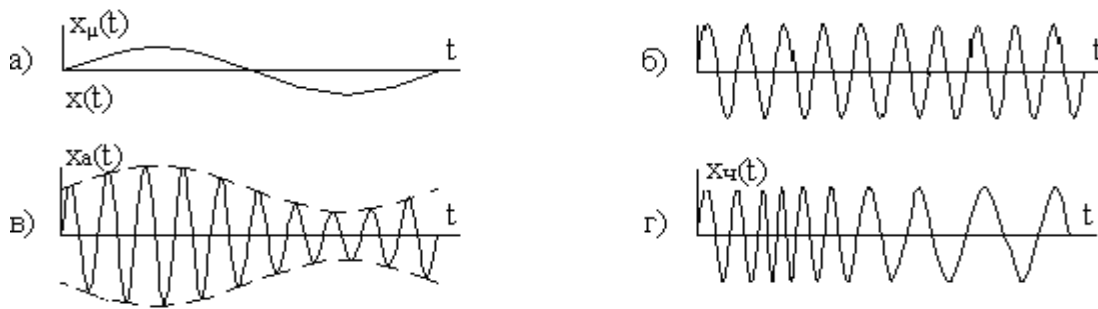


Рис. 1.1

На рис. 1.1, *а, б, в* ілюструється найпростіший випадок, коли несуча моногармонічна функція  $x(t)$  з нульовою початковою фазою та частотою  $\omega_1$  модулюється за амплітудою синусоїдальним за формою сигналом  $x_\mu(t)$  з частотою  $\Omega$ . Промодульований сигнал, представлений на рис. 1.1, *в* записується у вигляді:

$$x_a(t) = X_m (1 + \mu \sin \Omega t) \sin \omega_1 t, \quad (1.3)$$

де  $\mu = \frac{X_\mu}{X_m}$  – глибина модуляції;  $X_\mu$  – амплітуда модулюючого сигналу.

Рис. 1.1, *г* ілюструє зміну базової функції при наявності модуляції за частотою.

Якщо модулюючий сигнал  $x_\mu(t)$  гармонічний, то модуляція називається *гармонічною*. Розрізняють *амплітудну* (АМ), *частотну* (ЧМ) та *фазову* (ФМ) модуляції. Останні дві, завдяки взаємозв'язку між частотою та фазою, часто об'єднуються під назвою *кутової модуляції*.

Пристрої, в яких забезпечується процес модуляції параметрів базової функції, називаються *модуляторами*.

Описані види модуляції використовуються для передачі сигналів з різних датчиків (наприклад, датчиків рівня рідини у цистернах, температури в приміщеннях і т. п.).

Не вдаючись у деталі теорії модуляції, необхідно відмітити, що амплітудна модуляція більш проста в технічній реалізації, порівняно з кутовими. Але, у той же час, цей вид модуляції має низьку стійкість як відносно зовнішніх завад, так і відносно нестабільності параметрів апаратури, в якій вона використовується.

Кутові види модуляції, навпаки, мають достатньо складну технічну реалізацію, але при цьому вони більш стійкі відносно зовнішніх завад. Недолік їх полягає в тому, що такі сигнали займають значно ширший частотний спектр порівняно з АМ.

### ***1.1.2. Імпульси, імпульсні послідовності***

Підтримувати стабільність і точність параметрів аналогових пристроїв досить важко. На їх роботу впливають технологічні допуски, які закладаються при виробництві електронних компонентів, коливання температури, напруги живлення, космічне випромінювання, шуми і наводки, що створюються електронними приладами, іншими колами і пристроями.

До того ж, реалізація математичних і логічних операцій у більшості випадків або дуже складна, або навіть неможлива при роботі з аналоговими сигналами. Для того, щоб впевнитись у цьому, достатньо спробувати реалізувати на аналогових компонентах будь-яку з відомих констант. Рекомендується провести такий дослід: використовуючи операційний підсилювач і решту реальних компонентів, реалізувати схему підсилювача так, щоб на виході підтримувалась напруга у вольтах, рівна за величиною числу  $\pi = 3,14159\dots$  На практиці забезпечити точність відтворення такого аналогового сигналу навіть з похибкою в 1% досить складно.

Окрім синусоїдальних, в якості базових сигналів часто використовують різні за формою імпульсні послідовності.

На рис. 1.2 зображений типовий одиночний імпульс та приведені у взаємозв'язку його амплітудні та часові параметри. До амплітудних відносяться:  $U_M$  – максимальне значення параметра імпульсу (його амплітуда);  $\Delta U$  – спад вершини, що визначається між рівнями  $U_M$  і  $0,9 U_M$ ; до часових відносяться:  $t_\phi$ ,  $t_c$  – тривалість фронту та спаду імпульсу;  $t_i$  – тривалість імпульсу.

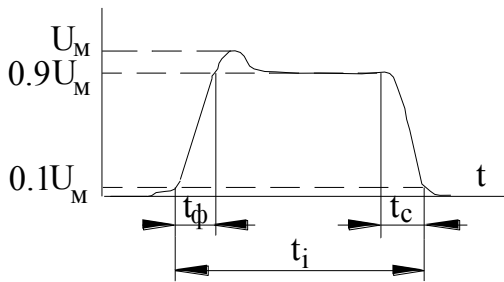


Рис.1. 2

Параметри імпульсу в часі можуть змінюватись у широких межах і, як результат, одиночні імпульси можуть мати різну форму. На рис. 1.3, а – в зображені відповідно експоненціальний, пилоподібний та трикутний імпульси.

Імпульсна послідовність характеризується наявністю пауз між імпульсами. В якості паузи тривалістю  $t_n$  приймається нульовий рівень напруги чи струму або такий рівень, який прирівнюється до нульового (рис. 1.4). Якщо тривалість імпульсів і пауз між ними в послідовності не змінюється, то говорять про *періодичну послідовність*, яка характеризується

періодом  $T = t_i + t_n$ ,

коефіцієнтом заповнення  $s = t_i / T$ ,

та шпаруватістю  $q = s^{-1}$ .

У теоретичних дослідженнях часто використовуються ідеалізації імпульсів. Перша з них – це *функція включення*, або *функція Хевісайда*. У загальному випадку функція включення, зміщена відносно початку осі координат на величину  $t_0$  (рис.1.5 а), записується у вигляді:

$$\sigma(t-t_0) = \begin{cases} 0 & \text{при } t < t_0 ; \\ 0,5 & \text{при } t = t_0 ; \\ 1 & \text{при } t > t_0 . \end{cases} \quad (1.4)$$

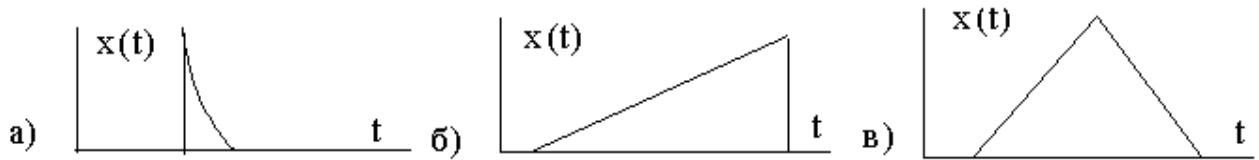


Рис.1.3

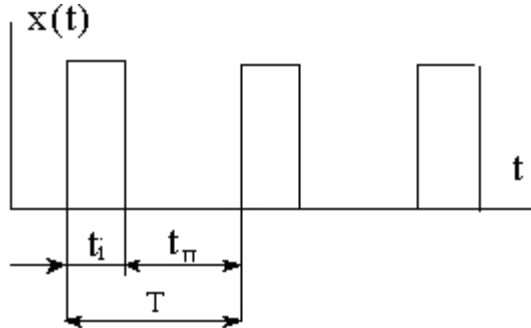


Рис.1.4

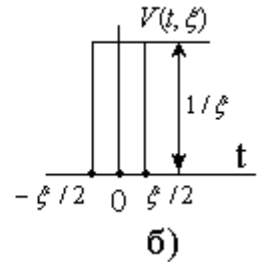
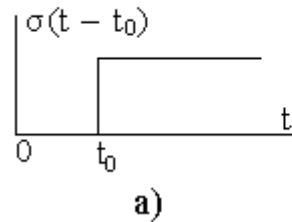


Рис.1.5

У теорії сигналів функції включення широко використовуються для опису розривних та імпульсних сигналів. Розглянемо імпульсний сигнал прямокутної форми, зображений на рис. 1.5 б. Його запис в аналітичній формі:

$$V(t, \xi) = \frac{\sigma\left(t + \frac{\xi}{2}\right) - \sigma\left(t - \frac{\xi}{2}\right)}{\xi}. \quad (1.5)$$

Особливість цього імпульсу полягає в тому, що при будь-якій величині параметру  $\xi$  його площа постійна і дорівнює одиниці

$$S = \int_{-\infty}^{+\infty} V(t) dt = 1 \quad (1.6)$$

Якщо  $\xi \rightarrow 0$ , враховуючи незмінність площі (1.5),(1.6), амплітуда імпульсу  $V(t, \xi)$  повинна збільшуватись до безкінечності.

Межа функції  $V(t, \xi)$  при  $\xi \rightarrow 0$  носить назву *дельта-функції*, або *функції Дірака*:

$$\delta(t) = \lim_{\xi \rightarrow 0} V(t, \xi). \quad (1.7)$$

Таким чином *дельта-функція* (1.7) будучи рівною нулю всюди, окрім точки  $t = 0$ , характерна одиничним значенням інтегралу  $\int_{-\infty}^{+\infty} \delta(t) dt = 1$

і являється математичною моделлю короткого імпульсу одиничної площі. Особливість *дельта-функції* - її фізична розмірність співпадає з розмірністю частоти.

Для передачі інформації імпульсною послідовністю остання модулюється за аналогією з синусоїдою. У залежності від модульованого параметру, розрізняють модуляції *амплітудно-імпульсну* (АІМ), *широтно-імпульсну* (ШІМ), *часово-імпульсну*, яка поділяється на *фазо-імпульсну* (ФІМ) і *частотно-імпульсну* (ЧІМ).

Рис. 1.6 ілюструє особливості модуляції імпульсної послідовності  $x(t)$  (рис. 1.6, б) функцією  $x_{\mu}(t)$  (рис. 1.6, а) відповідно до АІМ ( $x_A(t)$ , рис. 1.6, в), ШІМ ( $x_{ш}(t)$ , рис. 1.6, г), ЧІМ ( $x_{ч}(t)$ , рис. 1.6, д).

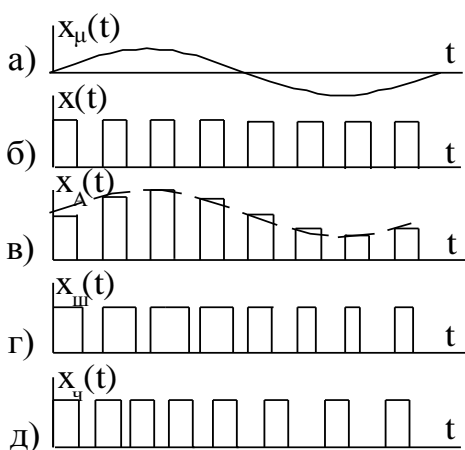


Рис. 1. 6

Перевага імпульсних сигналів порівняно з аналоговими полягає у тому, що при однаковій середній потужності імпульсного і аналогового сигналів енергія перших концентрується в інтервалі тривалості імпульсу. Тому при коротких імпульсах миттєве значення потужності імпульсу набагато перевищує її середнє значення. Завдяки цьому відносний вплив

зовнішніх факторів на сигнал є значно меншим.

Використання процесів модуляції неможливо без наявності зворотних процесів – виділення інформаційного сигналу з модульованого. Такі процеси називаються *демодуляцією* або *детектуванням*. При передачі інформації часто використовуються операції перетворення неперервного сигналу в імпульсний та навпаки. Ця операція досить наочно ілюструється рис. 1.6, в.

Для перетворення аналогового сигналу в імпульсний необхідно *здійснити дискретизацію (квантування) за часом*. Це означає, що неперервний час осі  $t$  замінюється дискретним з кроком (періодом квантування)  $T$  (рис.1.7).

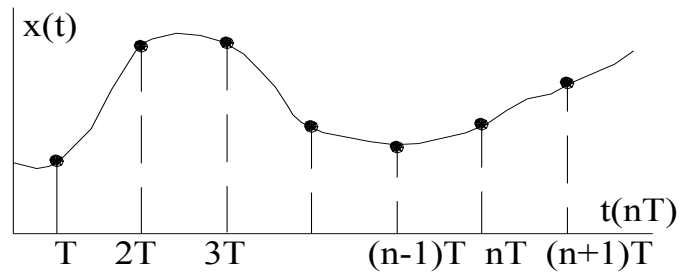


Рис.1. 7

У залежності від способів реалізації дискретизації за часом, сформована імпульсна послідовність може мати різний математичний опис. При перетворенні інформаційних сигналів для цього використовують дельта-функцію (1.7), завдяки особливості її згортки:

$$\int_{-\infty}^{+\infty} x(t) \delta(t - nT) dt = x(nT). \quad (1.8)$$

Ця особливість відображає фільтруючу властивість дельта-функції і полягає в тому, що після виконання операцій множення та інтегрування, функція  $x(t)$  буде визначеною лише в точках дискретизації за часом  $nT$  (тобто функція визначається з точністю до кроку дискретизації  $nT$ ). Функція (1.8) називається *дискретною*. Величина кроку дискретизації за часом вибирається у відповідності до необхідної точності відтворення неперервного сигналу. За теоремою В.А. Котельникова, максимальна частота  $f_g$  спектру сигналу, який може бути відновлений з дискретної послідовності за допомогою ряду Котельникова В.А., не може перевищувати величину  $f_g \leq 1/2T$ .

Спрощена графічна ілюстрація теореми В.А. Котельникова демонструється за допомогою рис. 1.8.

Точність відтворення сигналу з імпульсної послідовності тим вища, чим вища частота дискретизації відносно верхньої частоти спектру інформаційного сигналу. Іншими словами, дискретизація за часом не пов'язана з втратою інформації, якщо частота дискретизації  $f_d=1/T$  як мінімум у два рази перевищує верхню частоту спектру сигналу  $f_b$ . Крок дискретизації  $T$  може бути як

постійним, так і змінним у залежності від вимог, які ставляться при відновленні сигналу.

Дискретні значення функції  $x(nT)$  можуть використовуватись для відновлення сигналу. Знову ж таки, на практиці використовується широка гама методів відновлення сигналу з квантованої послідовності  $\delta$ -імпульсів.

Це може бути лінійна апроксимація (рис. 1.9, а), ступінчаста (рис. 1.9, б) або перетворення в імпульсні послідовності з заданими видами модуляції (рис. 1.9, в).

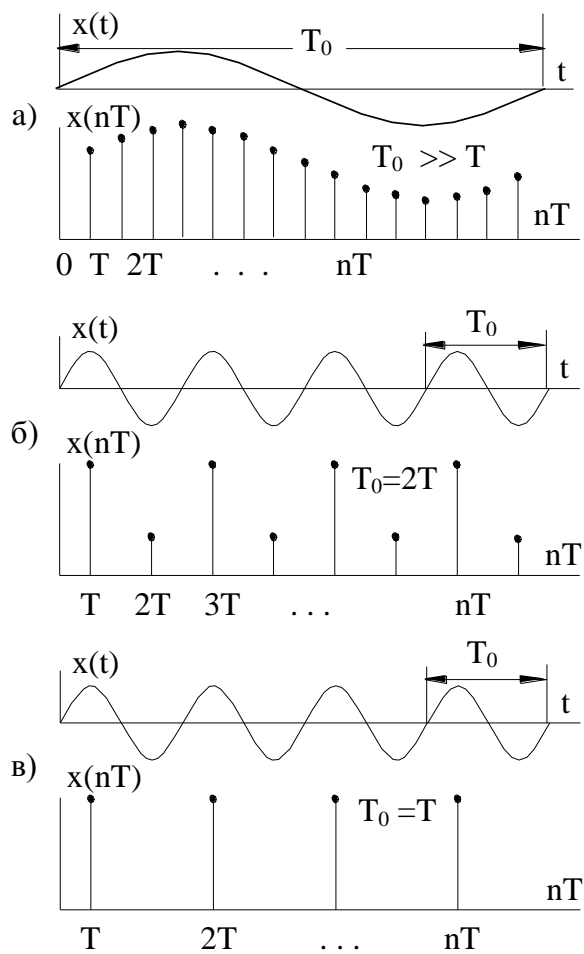


Рис. 1.8

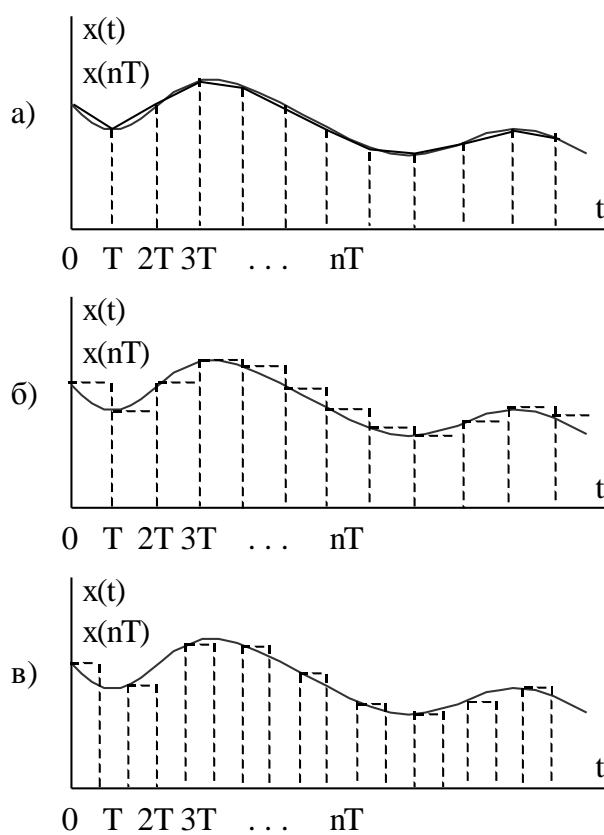


Рис. 1.9

### 1.1.3. Оцифрування аналогових сигналів

Для використання сучасних засобів обробки сигналів, поряд з дискретизацією за часом, використовується *дискретизація за рівнем*. Цей спосіб дискретизації, який часто називають *квантуванням за рівнем*,



полягає у тому, що безперервна множина значень рівнів замінюється дискретною з кроком  $\Delta x$  (рис. 1.10).

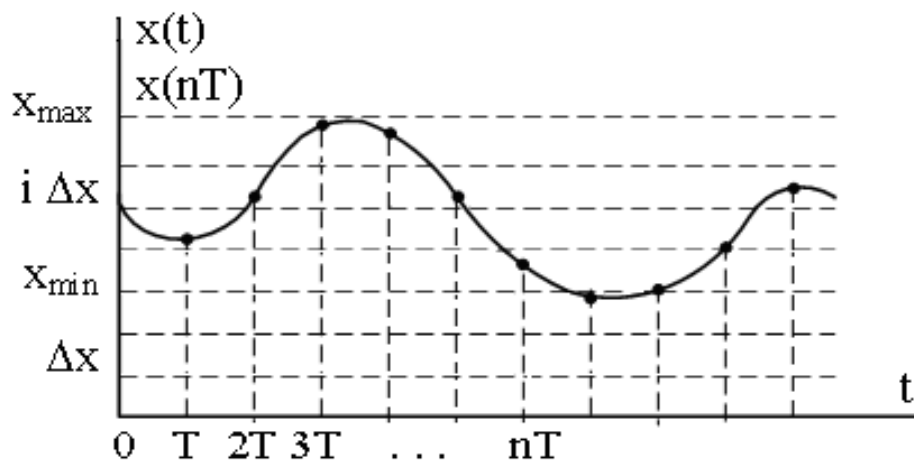


Рис. 1.10

Фактично квантування за рівнем представляє собою округлення значень  $x(nT)$  функції  $x(t)$  з заданою точністю. Квантування за рівнем може бути як *рівномірним*, так і *нерівномірним*. При рівномірному квантуванні кількість можливих рівнів  $m$  дорівнює:

$$m = \frac{x_{\max} - x_{\min}}{\Delta x}, \quad (1.9)$$

де  $x_{\max}$ ,  $x_{\min}$  – відповідно, верхня і нижня межа зміни сигналу  $x(t)$ .

Величина  $\Delta x$  визначає похибку, що має місце при заміні поточного значення  $x(t)$  його дискретним рівнем  $i \cdot \Delta x = x_i$ . Ця похибка, що знаходиться за формулою:  $\xi(x) = x(nT) - x_i$ , називається *шумом квантування*. Якщо при квантуванні за рівнем будь-якому значенню змінної  $x(t)$ , що попадає в інтервал  $\left[ x_i - \frac{\Delta x}{2}, x_i + \frac{\Delta x}{2} \right]$ , присвоюється рівень  $x_i$ , то похибка  $\xi(x)$  не перевищуватиме

половини кроку квантування:  $|\xi(x)_{\max}| = \frac{\Delta x}{2}$ .

У теорії сигналів широко використовується імовірнісна оцінка шумів квантування.

Важливою характеристикою будь-якого сигналу є його *інформативність*, яка визначається кількісною характеристикою інформації. Для її визначення розглянемо спрощену дискретну модель реального сигналу, заданого на інтервалі часу  $T_c$  і квантованого за часом і за рівнем.

Використовуючи умови теореми Котельникова, знайдемо приблизну кількість дискретних за часом значень функції:

$$n = \frac{T_c}{T} = 2f_{\max} T_c,$$

де  $f_{\max}$  – максимальна частота в спектрі сигналу.

Кількість можливих рівнів квантування  $m$  визначається формулою (1.9). Оскільки в кожному дискретний момент часу сигнал може приймати одне з  $m$  можливих значень, то за час  $T_c$  кількість можливих комбінацій сигналу дорівнює:

$$C_c = m^n.$$

Число  $C_c$  дає комбінаторну оцінку кількості інформації, що міститься у дискретному сигналі. Недолік використання  $C_c$  в якості міри інформації полягає в нелінійній його залежності від величини  $n$ , тобто від тривалості інтервалу  $T_c$ . Тому в якості міри кількості інформації використовується логарифмічне перетворення від  $C_c$ :

$$I = \log_b C_c = n \log_b m,$$

в якому маємо лінійну залежність між вказаними параметрами. Вибір параметра  $b$  впливає лише на розмірність, тобто на одиницю виміру кількості інформації. Найчастіше приймають  $b = 2$ , при цьому  $I$  вимірюється у *бітах* (*binary digit*). Один *біт* – це найменша кількість інформації, що відповідає одному з рівноможливих повідомлень (так-ні; ввімкнути-вимкнути тощо). Повідомлення, в якому міститься набір з декількох бітів, називається *словом*. Слово з 8 біт називається *байтом*. Якщо, наприклад, допустити, що кількість рівнів квантування  $m$  описується словом в один байт, то це означає, що весь діапазон рівнів  $x_{\max} - x_{\min}$  розбивається на  $m = 2^8 = 256$  кроків, функція визначатиметься з похибкою  $\xi(x) = (256 \cdot 2)^{-1} (X_{\max} - X_{\min})$  і в кожному дискретний

момент часу може передаватися повідомлення про одне з її 256 можливих значень.

Широко використовуються більші одиниці кількості інформації – *кілобіт* ( $2^{10}$  біт), *мегабіт* ( $2^{20}$  біт), *гігабіт* ( $2^{30}$  біт).

Звернувшись знову до процесу перетворення сигналів і прийнявши за “1” наявність короткочасного імпульсу, а за “0” – його відсутність, кожен відлік сигналу (рис. 1.10) можна передати у вигляді комбінації одиниць та нулів. Процес перетворення повідомлення у дискретний сигнал називається *кодуванням інформації*, а множина різних кодових комбінацій, що отримуються при вибраному правилі кодування – *кодом*.

Оскільки сигнали передаються переважно за допомогою провідників, то слово в один байт може передаватися по двохпроводній лінії як послідовність “1” та “0” у визначені дискретні моменти часу. При цьому як передавач інформації, так і приймач повинні працювати в узгодженому дискретному часі. Така форма передачі інформації називається *послідовною (послідовний формат)* і використовується при довгих лініях зв’язку. При коротких лініях використовується *паралельна форма*, при якій для кожного біта слова, що передається, використовується свій провідник (*паралельний формат*).

Більшість сучасних систем обробки цифрової інформації опрацьовують слова, що вміщують ціле число байтів, – наприклад 1, 2, 4.

У той же час, при вирішенні нескладних задач часто використовуються слова, що є напівбайтовими (чотирьохрозрядними). Іноді використовуються навіть трьохрозрядні слова і числа.

## **1.2. Системи числення**

### ***1.2.1. Основні визначення***

*Система числення* – це спосіб запису (зображення) чисел.

Системи числення, в яких ваговий коефіцієнт кожної цифри залежить від її положення у послідовності цифр, що зображає число, називаються

позиційними. У *непозиційних* системах значення кожної цифри постійне і не залежить від місця її розташування в числі. Всі системи числення, які використовуються в цифровій схемотехніці, є позиційними.

При розгляді позиційних систем важливим виступає поняття *базису*. *Базис системи числення* – це послідовність чисел, яка задає значення (вагу) кожної цифри в залежності від місця її розміщення.

Приклади базисів:

- десяткової системи числення:  $10^0, 10^1, 10^2, \dots, 10^n, \dots$ ;
- двійкової –  $2^0, 2^1, 2^2, \dots, 2^n, \dots$ ;
- вісімкової –  $8^0, 8^1, 8^2, \dots, 8^n, \dots$ ;
- шістнадцяткової –  $16^0, 16^1, 16^2, \dots, 16^n, \dots$

У загальному плані для позиційних систем числення базис можна записати в вигляді послідовних членів геометричної прогресії:

$$\dots P^{-m}, \dots, P^{-2}, P^{-1}, P^0, P^1, P^2, \dots, P^n, \dots$$

Число  $P$  називається *основою системи числення*. У подальшому при розгляді систем числення основа зображатиметься у вигляді нижнього індексу в кінці числа.

Сукупність різних цифр, які використовуються в позиційній системі числення для запису чисел, називається *алфавітом системи*.

Будь-яке натуральне число  $A$  в  $P$ -ічній системі числення записується у розгорнутій і згорнутій формах запису. Наприклад, число  $A$  в  $P$ -ічній системі числення представляється в згорнутій формі так:

$$A = (a_n a_{n-1} \dots a_1 a_0 a_{-1} a_{-2} \dots a_{-k})_P ; \quad (1.10)$$

у розгорнутій:

$$A = a_n \cdot P^n + a_{n-1} \cdot P^{n-1} + \dots + a_1 \cdot P^1 + a_0 \cdot P^0 + a_{-1} \cdot P^{-1} + a_{-2} \cdot P^{-2} + \dots + a_{-k} \cdot P^{-k} \quad (1.11)$$

**Приклад 1.1.** Представимо конкретне число  $A$  в згорнутій і розгорнутій формах десяткової системи числення.

*Розв'язання.* Запис числа  $A$  в згорнутій формі:  $A = 837,25_{10}$ ;

в розгорнутій формі:  $A = 8 \cdot 10^2 + 3 \cdot 10^1 + 7 \cdot 10^0 + 2 \cdot 10^{-1} + 5 \cdot 10^{-2}$ .

**Приклад 1.2.** Запис числа  $61_{10}$  в різних системах числення.

*Розв'язання.* Запис числа  $61_{10}$  у двійковій системі числення –  $111101_2$ ;

у трійковій –  $2021_3$ ;

у четвірковій –  $331_4$ ;

у шістнадцятковій –  $3D_{16}$ ;

з основою 61 –  $10_{61}$ .

Кількість цифр в алфавіті  $P$ -ічної системи числення дорівнює основі системи числення, починаючи з нуля. Тому алфавітом  $P$ -ічної системи числення є натуральний ряд чисел від нуля до  $P-1$ . В якості алфавіту систем числення прийнято використовувати:

- арабські числа, якщо основа менше 10;
- арабські числа і букви латинського алфавіту при основі до 36.

Якщо основа більше 36, то загальних правил не існує.

**Приклад 1.3.** Привести алфавіт шістнадцяткової системи числення.

*Розв'язання.* Алфавіт шістнадцяткової системи числення має вигляд: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

### ***1.2.2. Переведення чисел з однієї позиційної системи числення в іншу***

Якщо число  $A$  представлено в  $P$ -ічній системі числення в формі (1.10), то для переведення в десяткову систему числення його необхідно представити в формі (1.11). Для отримання значення цього багаточлена, записаного в десятковій системі числення, необхідно число  $P$  і коефіцієнти при ступенях  $P$  (цифри алфавіту  $P$ -ічного числа) записати у вигляді десяткових чисел і всі обчислення провести в десятковій системі.

**Приклад 1.4.** Переведемо число  $A = C20B_{16}$  в десяткову форму запису.

*Розв'язання.* Враховуючи, що  $C = 12$ ,  $B = 11$ , можемо записати:

$$A = 12 \cdot 16^3 + 2 \cdot 16^2 + 0 \cdot 16^1 + 11 \cdot 16^0 = 49675_{10}.$$

При обчисленні значення  $P$ -ічного числа за розгорнутою формою зручно користуватися схемою Горнера, яка дозволяє отримати результат з

використанням мінімального числа арифметичних операцій додавання і множення. Загальний вигляд її:

$$A = (\dots(((a_n \cdot P + a_{n-1}) P + a_{n-2}) P + a_{n-3}) + \dots + a_1) P + a_0.$$

**Приклад 1.5.** Перевести двійкове число  $A = 11011_2$  в десяткову систему числення.

*Розв'язання.* Запишемо число в розгорнутій формі:

$$A = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 2(2^2(2+1)+1) + 1 = 27.$$

При програмуванні як на мовах високого рівня, так і на мові Асемблера часто необхідно знати значення ступенів двійки до 16. У цифровій та мікропроцесорній техніці важливо пам'ятати ступені до 10. Приведемо їх, починаючи з 5:

$$2^5 = 32; \quad 2^6 = 64; \quad 2^7 = 128; \quad 2^8 = 256; \quad 2^9 = 512; \quad 2^{10} = 1024.$$

Знання цих чисел дає можливість більш спрощено розв'язувати задачі переведення десяткових чисел у двійковий код.

Приведене вище правило переведення цілих чисел у десяткову систему числення може бути використаним і для переведення дробових чисел.

**Приклад 1.6.** Перевести число  $A = 0,11_2$  в десяткову систему числення.

*Розв'язання.* Запишемо число  $A$  в розгорнутій формі:

$$A = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} = 0,75_{10}.$$

Формула (1.11) здебільшого використовується для переходу від системи числення з меншою основою до системи числення з більшою основою.

Ірраціональні дробові числа представляються у скороченій формі і переводяться аналогічно, або для них використовуються спеціальні алгоритми.

### **1.2.3. Переведення цілого числа з десяткової системи числення в $P$ -ічну**

Для виконання подібних перетворень використовується декілька способів. Один з них полягає у наступному. Запишемо відоме число  $A_{10}$ , представлене в десятковій системі числення в умовній  $P$ -ій системі числення, де коефіцієнти  $a_n$  поки що невідомі:

$$A_{10} = a_n \cdot P^n + \dots + a_1 \cdot P^1 + a_0.$$

Розділивши праву і ліву частини на  $P$ , отримаємо ціле число  $a_n \cdot P^{n-1} + \dots + a_1$  і залишок, величина якого не перевищує значення  $P - 1$ . Таким шляхом отримується остання цифра запису числа в  $P$ -ічній системі числення.

Виконуючи аналогічне ділення десяткового числа  $n$  разів, можемо отримати всі невідомі коефіцієнти  $P$ -ічної системи числення.

**Приклад 1.7.** Перевести число  $123_{10}$  в трійкову систему числення.

*Розв'язання.* Послідовно виконується операція ділення числа 123 на число 3.

$$\begin{array}{r} 123 : 3 = 41 + \text{залишок } 0 \\ 41 : 3 = 13 + \text{залишок } 2 \\ 13 : 3 = 4 + \text{залишок } 1 \\ 4 : 3 = 1 + \text{залишок } 1 \\ 1 : 3 = 0 + \text{залишок } 1 \end{array}$$

Записуючи значення залишків знизу вгору, отримаємо число у трійковій системі числення, тобто  $123_{10} = 11120_3$ .

Переведення десяткового числа в двійковий код виконується шляхом послідовного ділення його на 2, а залишки (0 або 1), що мають місце після ділення на кожному кроці, створюють двійковий код перетворюваного числа, починаючи з молодшого розряду.

**Приклад 1.8.** Перетворити у двійковий код число  $105_{10}$ .

*Розв'язання.* Операція перетворення виконується у послідовності, приведеній нижче.

$$\begin{array}{r} 105 : 2 = 52 + \text{залишок } 1 = a_0 \\ 52 : 2 = 26 + \text{залишок } 0 = a_1 \\ 26 : 2 = 13 + \text{залишок } 0 = a_2 \\ 13 : 2 = 6 + \text{залишок } 1 = a_3 \\ 6 : 2 = 3 + \text{залишок } 0 = a_4 \\ 3 : 2 = 1 + \text{залишок } 1 = a_5 \\ 1 : 2 = 0 + \text{залишок } 1 = a_6 \end{array}$$

Тобто,  $105_{10} = A_2 = a_6 a_5 a_4 a_3 a_2 a_1 a_0 = 1101001_2$ .

Типові помилки при реалізації такого алгоритму наступні: порушення порядку запису цифр, що одержуються; неправильне вписування крайньої ліворуч цифри. Використовуючи ступені числа 2 з прикладу 1.5, скористаємось спрощеним способом переведення числа  $A_{10}$  у двійкову систему числення. Дійсно, можемо записати нерівності  $2^7 > 105 > 2^6$ .

Звідси витікає, що двійкове число представляється 7-ма розрядами, старший з яких  $2^6 = 64$ . Оскільки різниця  $105 - 64 = 41$  знаходиться в інтервалі  $2^6 > 41 > 2^5$ , то стверджуємо, що і наступний по старшинству розряд – шостий ( $2^5 = 32$ ) – дорівнює 1. Наступна різниця  $41 - 32 = 9 = 1001_2$ . П'ятий розряд дорівнює нулю, і в результаті отримуємо ту ж саму відповідь.

Переведення числа  $A$ , що має дробову частину, з десяткової системи числення у двійкову має ту особливість, що ціла і дробова частини переводяться окремо.

Сформулюємо тепер правило переведення дробової частини з десяткової системи числення в  $P$ -ічну. Знову представимо її у розгорнутому вигляді:

$$A_{10} = a_{-1} \cdot P^{-1} + a_{-2} \cdot P^{-2} + \dots + a_{-k} \cdot P^{-k} + \dots \quad (1.12)$$

Перемножуючи ліву і праву частини (1.12) на  $P$  в правій частині виразу отримуємо:

$$a_{-1} + a_{-2} \cdot P^{-1} + \dots + a_{-k} \cdot P^{-k+1} + \dots \quad (1.13)$$

З отриманого результату можемо зробити висновок, що перша цифра  $a_{-1}$  дробової частини числа  $A$  в  $P$ -ічній системі числення дорівнює цілій частині результату перемноження десяткової дробової частини на  $P$ . Після чергового перемноження залишку дробової частини на  $P$  отримаємо значення  $a_{-2}$ :

$$(a_{-2} \cdot P^{-1} + \dots + a_{-k} \cdot P^{-k+1} + \dots) \cdot P.$$

Цей процес продовжується до тих пір, поки дробова частина результату перемноження лівої частини не стане рівною нулю або поки не буде виділений період повторності цифр.

**Приклад 1.9.** Перевести число  $A = 0,375_{10}$  в двійкову систему числення.

*Розв'язання.* Виконуємо операцію множення в приведеній нижче послідовності.

$0,375$	$\times$	$2$	$=$	$0,75$	$0$	– перша цифра результату
$0,75$	$\times$	$2$	$=$	$1,5$	$1$	– друга цифра результату
$0,5$	$\times$	$2$	$=$	$1$	$1$	– остання цифра результату

Внаслідок виконання перетворень отримали результат  $0,375_{10} = 0,011_2$ .

**Приклад 1.10.** Перевести десяткове число  $A = 0,109375_{10}$  у шістнадцяткову систему числення.



*Розв'язання.* Виконуємо операцію множення в приведеній нижче послідовності.

$$\begin{array}{r} 0,109375 \times 16 = 1,75 \quad 1 - \text{перша цифра результату} \\ 0,75 \times 16 = 12 \quad 12 = C - \text{остання цифра результату} \end{array}$$

В результаті отримана відповідь:  $0,109375_{10} = 0,1C_{16}$ .

Переведення числа з однієї недесятькової системи числення в аналогічну іншу виконується шляхом перетворення  $P$ -ічної системи числення в десяткову, після чого виконується друга частина операції. Виключення складає лише переведення з двійкової системи числення в шістнадцяткову і навпаки.

### 1.3. Коди та їх характеристика

#### 1.3.1. Коди з паралельною формою представлення інформації

У попередньому параграфі було розглянуто перетворення аналогового сигналу в послідовність сигналів, які умовно приймають лише значення “1” та “0”. Система числення, в якій використовуються лише два знаки для відображення інформації, називається *двійковою*, основою якої є число 2. За аналогією з десятковою, двійкова система числення є позиційною, і будь-яке ціле десяткове число може бути представлене двійковим рядом, що вміщує лише “1” та “0” у відповідності з алгоритмом:

$$A_{10} = \sum_{i=0}^{k-1} a_i \cdot 10^i \rightarrow A_2 = \sum_{j=0}^{p-1} a_j \cdot 2^j, \quad (1.14)$$

де  $a_i = 0 \dots 9$  – цифри  $i$ -го розряду десяткового числа;  $a_j = 0 \dots 1$  – відповідно, цифри  $j$ -го розряду двійкового числа. Розряди чисел рахуються зліва направо, починаючи зі старшого. Дробові числа представляються доповненням суми (1.14) від’ємними ступенями числа 2.

Електронні системи, що оперують сигналами, які відповідають лише рівням “1” та “0”, називаються *цифровими*. Схеми, на основі яких реалізуються такі системи, також називаються *цифровими*, а розділи електроніки, що вивчають принципи побудови таких схем – *цифровою схемотехнікою*.

Як у теорії інформації, так і на практиці застосування цифрової схемотехніки використовується багато різноманітних кодів. Визначимось з основною термінологією.

*Код* – це універсальний спосіб відображення інформації при її зберіганні, передачі і обробці у вигляді системи однозначних відповідностей між елементами повідомлень і сигналами, за допомогою яких ці елементи можна зафіксувати. Іншими словами, *кодування* – це однозначне перетворення символів одного алфавіту в символи іншого, а код – правило, закон, алгоритм, при якому відбувається це перетворення. Комбінації символів, що належать до даного коду, називаються *кодovими словами*. Символи, за допомогою яких повідомлення трансформується в код, є *вторинним алфавітом*. Процес відновлення вмісту повідомлення за допомогою відповідного коду називається *декодуванням*. Необхідною умовою декодування є взаємно однозначна відповідність кодovих слів у вторинному алфавіті символам первинного алфавіту та їх комбінаціям. При передачі кодovих символів по лініях зв'язку вони повинні бути розділені так, щоб кожен символ міг бути прийнятим самостійно, що виконується з використанням різних принципів їх розділення. Розділення символів може бути *просторовим, часовим і якісним*. *Просторове* розділення по суті є багатоканальним зв'язком, і при його використанні відпадає необхідність у спеціальних методах кодування. При *якісному* розділенні між символами повинно бути як мінімум дві розподільчі ознаки (наприклад, тривалість імпульсу, паузи), які легко відрізняються на приймальній стороні лінії зв'язку. *Якісне* розділення дає можливість одночасної передачі інформації від різних об'єктів по одному каналу зв'язку. Прикладом якісного розділення є частотне розділення (моногармонічний сигнал при імпульсі має одну частоту, а при паузі – іншу). При часовому розділенні використовуються спеціальні комутатори на передаючій та приймальній сторонах, які по чергово з'єднують необхідні лінії зв'язку.

Двійкове кодування десяткових чисел в відповідності з (1.14) не є єдиним. При роботі з двійковими числами широко використовуються й інші коди, які в різних практичних ситуаціях мають свої переваги перед двійковим. Деякі з них для позитивних чисел в інтервалі  $0 \dots 15$  представлені у табл. 1.1.

Таблиця 1.1

$A_{10}$	$A_2$ (двійковий)				$B_2$ (обернений)				$D_2$ (доповняльний)				Код Грея			
	$a_3$	$a_2$	$a_1$	$a_0$	$b_3$	$b_2$	$b_1$	$b_0$	$d_3$	$d_2$	$d_1$	$d_0$	$g_3$	$g_2$	$g_1$	$g_0$
0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	1	0	1	1	1	1	0	0	0	1
2	0	0	1	0	1	1	0	1	1	1	0	0	0	0	1	1
3	0	0	1	1	1	1	0	0	1	1	0	1	0	0	1	0
4	0	1	0	0	1	0	1	1	1	1	0	0	0	1	1	0
5	0	1	0	1	1	0	1	0	1	0	1	1	0	1	1	1
6	0	1	1	0	1	0	0	1	1	0	1	0	0	1	0	1
7	0	1	1	1	1	0	0	0	1	0	0	1	0	1	0	0
8	1	0	0	0	0	1	1	1	1	0	0	0	1	1	0	0
9	1	0	0	1	0	1	1	0	0	1	1	1	1	1	0	1
10	1	0	1	0	0	1	0	1	0	1	1	0	1	1	1	1
11	1	0	1	1	0	1	0	0	0	1	0	1	1	1	1	0
12	1	1	0	0	0	0	1	1	0	1	0	0	1	0	1	0
13	1	1	0	1	0	0	1	0	0	0	1	1	1	0	1	1
14	1	1	1	0	0	0	0	1	0	0	1	0	1	0	0	1
15	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	0

Прямий двійковий код  $A_2$  також називають кодом 8-4-2-1 у відповідності з ваговими коефіцієнтами розрядів.

Обернений двійковий код  $B_2 = b_3 b_2 b_1 b_0$  отримується шляхом інверсії кожного розряду прямого коду  $B_2 = b_3 b_2 b_1 b_0 = \overline{A_2} = \overline{a_3} \overline{a_2} \overline{a_1} \overline{a_0}$ .

При довільній основі  $P$  обернений код  $n$  розрядного числа  $N$  доповнює його до максимального можливого значення  $P^n - 1$ ,  $N_0 = P^n - 1 - N$ .

При цьому, цифра кожного розряду оберненого кода  $N_0$  доповнює відповідну цифру прямого коду  $N$  до найбільшого значення  $P-1$  (для десяткового коду це 9).

Обернений код використовується як самостійно в логічних структурах цифрових систем, так і при виконанні арифметичних операцій для одержання доповнюючого коду  $D_2$ . Останній застосовується при виконанні арифметичних операцій і знаходиться відповідно до формули:

$$D_2 = B_2 + 1 = \overline{A_2} + 1 = \overline{a_3} \overline{a_2} \overline{a_1} \overline{a_0} + 1,$$

де число 1 додається шляхом двійкової арифметики.

*Код Грея*, який часто називається *циклічним*, має ту особливість, що при переході з одного числа до сусіднього проходить зміна “0” на “1” або навпаки тільки в одному розряді. Як видно з таблиці, код, представлений двома, трьома або чотирма розрядами, завжди створює циклічну послідовність, тобто адекватну можливість переходу від самого старшого кодового значення числа до самого молодшого. Ця особливість дозволяє використовувати його при кодуванні кутових переміщень у перетворювачах кута повороту у цифровий код. Код Грея знаходить також широке використання у різних перетворювачах “аналог - код”, де його властивість дає можливість звести похибки неоднозначності при зчитуванні інформації до одиниці молодшого розряду.

Для одержання коду Грея в літературі описані декілька різних прийомів. Один з них дозволяє будувати код Грея безпосередньо з двійкового, використовуючи наступне правило:  $i$ -й біт коду Грея встановлюється в нуль, якщо  $i$ -й та  $(i + 1)$ -й біти відповідного двійкового коду однакові; у протилежному випадку біт  $i = 1$ . У тому випадку, коли  $(i + 1)$ -й біт виходить за рамки розрядності двійкового коду, його значення приймається рівним нулю.

При записі слів двійкового коду може використовуватись шістнадцяткове числення:

$$A_{16} = \sum_{j=0}^s a_j^{16} \cdot 16^j .$$

При його використанні десяткові числа від 10 до 15 замінюються відповідно латинськими літерами A, B, C, D, E, F. Двійковий і шістнадцятковий коди легко взаємно переводяться. Для цього двійкове слово будь-якої довжини

розбивається на тетради, починаючи з молодшого розряду  $i$ , відповідно до табл. 1.1 та вищезазначеними еквівалентами літер та цифр, записується його шістнадцяткове представлення. Наприклад,

$$A_2 = 111011_2 = 0011\ 1011_2 = 3B_{16}.$$

Для позначення цього коду використовують букву  $h$  (*hexadecimal*), яку ставлять після його числового значення:  $3B_{16} = 3Bh$ .

У задачах обчислювальної та мікропроцесорної техніки часто виникає необхідність працювати з десятковими числами. Для цього використовують двійкове представлення кожної десяткової цифри окремо у відповідності з табл. 1.2. Наприклад, числу  $937_{10}$  відповідає код **1001 0011 0111**. Таке представлення називається *двійково-десятковим кодом*. При виконанні арифметичних операцій з використанням двійково-десяткового коду виникають проблеми, пов'язані із переносом з одного розряду до іншого. Для їх вирішення використовуються *самодоповнюючі коди*. До них відносяться *код Айкена*, вагові коефіцієнти розрядів якого **2-4-2-1**, а також код “з надлишком 3”, який отримується з двійково-десяткового шляхом додавання числа 3. Особливості цих кодів можна побачити з табл. 1.2.

Таблиця 1.2

Десяткове число	Двійково- десятковий код				Код Айкена 2-4-2-1				Код «з надлишком 3»			
	$a_3$	$a_2$	$a_1$	$a_0$	$h_3$	$h_2$	$h_1$	$h_0$	$c_3$	$c_2$	$c_1$	$c_0$
0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	0	0	1	0	0	0	1	0	1	0	0
2	0	0	1	0	0	0	1	0	0	1	0	1
3	0	0	1	1	0	0	1	1	0	1	1	0
4	0	1	0	0	0	1	0	0	0	1	1	1
5	0	1	0	1	1	0	1	1	1	0	0	0
6	0	1	1	0	1	1	0	0	1	0	0	1
7	0	1	1	1	1	1	0	1	1	0	1	0
8	1	0	0	0	1	1	1	0	1	0	1	1
9	1	0	0	1	1	1	1	1	1	1	0	0

Розглянемо більш детально взаємозв'язок між кодами на прикладі конкретних чисел. Для числа 3, представленого в кодї “з надлишком 3”, у відповідності до табл. 1.2, маємо:

$$C = c_3 c_2 c_1 c_0 = 0110 = 3_{10}; \quad \bar{C} = 1001 = 6_{10}.$$

В результаті  $C + \bar{C} = 9_{10}$ . Аналогічно в кодї Айкана візьмемо число  $5_{10}$ . З табл.1.2  $H = h_3 h_2 h_1 h_0 = 1011$ . Відповідно:  $\bar{H} = 0100 = 4_{10}$ . В результаті отримаємо:  $H + \bar{H} = 9_{10}$ .

Бачимо, що властивістю самодоповнюючих кодів є те, що їх зворотні комбінації доповнюють прямі до 9. Ця особливість дозволяє мікропроцесорним пристроям виконувати арифметичні операції з десятковими числами.

У мікропроцесорній техніці двійково-десяткові коди широко використовуються для відображення цифрової інформації за допомогою знакосинтезуючих індикаторів. Тому необхідно вміти виконувати перетворення двійкового коду будь-якої розрядності у двійково-десятковий. Виконується ця операція у такій послідовності.

Оскільки, наприклад, однобайтовим двійковим словом може бути закодована цифра, яка у десятковому кодї містить три знаки (до 255), то таке слово повинно бути переміщене з двійкового регістру  $R$  відповідними засобами у три чотирьохбітні регістри – регістр одиниць  $R_0$ , регістр десятків  $R_d$  і регістр сотень  $R_c$ :  $R_c \leftarrow R_d \leftarrow R_0 \leftarrow R$ .

Виходячи з правил двійкової арифметики, при переміщенні кожного розряду з регістра  $R$  у регістри, розміщені зліва, необхідно контролювати значення чисел, що з'являються в регістрах  $R_c, R_d, R_0$ .

Якщо ці числа дорівнюють або перевищують  $5_{10} = 0101_2$ , то до них необхідно додати число  $3_{10} = 0011_2$ .

Контролюючи у такий спосіб вміст кожного регістра, за вісім тактів операції зсуву виконується повний цикл перетворення двійкового числа з регістру  $R$  у відповідні регістри  $R_c, R_d, R_0$ .

Читачам пропонується самостійно поекспериментувати над таким перетворенням.

**Приклад 1.11.** Виконати перетворення числа  $217_{10} = 1101\ 1001_2$  у двійково-десятковий код шляхом використання операцій зсуву та додавання числа  $3_{10}$ .

*Розв'язання.* Керуючись алгоритмом, описаним вище, виконуємо відповідне перетворення.

№ етапу	Регістр сотень $R_C$	Регістр десятків $R_D$	Регістр одиниць $R_O$	Задане число, регістр $R$
	0 0 0 0	0 0 0 0	0 0 0 0	1 1 0 1 1 0 0 1
1	0 0 0 0	0 0 0 0	0 0 0 1	
2	0 0 0 0	0 0 0 0	0 0 1 1	
3	0 0 0 0	0 0 0 0	0 1 1 0	
			1 1	
	0 0 0 0	0 0 0 0	1 0 0 1	
4	0 0 0 0	0 0 0 1	0 0 1 1	
5	0 0 0 0	0 0 1 0	0 1 1 1	
			1 1	
	0 0 0 0	0 0 1 0	1 0 1 0	
6	0 0 0 0	0 1 0 1	0 1 0 0	
		1 1		
	0 0 0 0	1 0 0 0	0 1 0 0	
7	0 0 0 1	0 0 0 0	1 0 0 0	
			1 1	
	0 0 0 1	0 0 0 0	1 0 1 1	
8	0 0 1 0	0 0 0 1	0 1 1 1	
	$2_{10}$	$1_{10}$	$7_{10}$	

Двійкові коди використовуються не тільки для кодування цифрової інформації. Широко відоме двійкове представлення букв, знаків, типових команд. У вітчизняній практиці такий код називається КОІ-7 (код для обміну інформацією, семирозрядний). Аналогічний код використовується в сучасній комп'ютерній техніці, який називається ASCII (*American Standart Code for Information Interchange*) і достатньо детально описаний у ряді літературних джерел.

Передача інформації завжди пов'язана з імовірністю появи похибок у лінії зв'язку. Вони пов'язані з тим, що при передачі інформації в двійковому коді окремі дискретні символи "0" та "1" можуть змінюватись відповідно на "1" та "0" при дії завад. Тому на практиці широко використовуються коди, які називаються *коригуючими* або *завадозахищеними*. Вони дозволяють своєчасно виявляти і виправляти похибки, що з'являються. Побудова таких кодів завжди пов'язана із введенням надмірності у лінію зв'язку та кодові комбінації, що передаються. Найпростішим з завадозахищених кодів є двійковий код з забезпеченням можливості перевірки на парність, у якому до  $n$ -розрядного коду, що передається, додається допоміжний  $(n + 1)$ -й розряд.

№ етапу	Регістр сотень $R_C$				Регістр десятків $R_D$				Регістр одиниць $R_O$				Задане число, регістр $R$							
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	0	0	1
1	0	0	0	0	0	0	0	0	0	0	0	1								
2	0	0	0	0	0	0	0	0	0	0	1	1								
3	0	0	0	0	0	0	0	0	0	1	1	0								
											1	1								
	0	0	0	0	0	0	0	0	1	0	0	1								
4	0	0	0	0	0	0	0	1	0	0	1	1								
5	0	0	0	0	0	0	1	0	0	1	1	1								
											1	1								
	0	0	0	0	0	0	1	0	1	0	1	0								
6	0	0	0	0	0	1	0	1	0	1	0	0								
							1	1												
	0	0	0	0	1	0	0	0	0	1	0	0								
7	0	0	0	1	0	0	0	0	1	0	0	0								
											1	1								
	0	0	0	1	0	0	0	0	1	0	1	1								
8	0	0	1	0	0	0	0	1	0	1	1	1								
	$2_{10}$				$1_{10}$				$7_{10}$											



Детально його використання описується в третьому розділі. Недолік такого коду полягає у відсутності можливості автоматично ліквідувати похибки. Незважаючи на це, він широко використовується в ЕОМ при роботі процесора з запам'ятовуваними пристроями.

До кодів, що виправляють можливі похибки, відносяться *циклічні коди*, які мають можливість виявляти та виправляти не тільки одиночні, а й похибки більшої кратності. До таких кодів відноситься *код Ріда-Соломона*, який широко використовується для захисту інформації в комп'ютерних системах.

У **Розділі 3** пояснюється побудова схемотехніки перетворювача коду на прикладі одного з кодів (*коду Хемінга*), що має можливість виправляти одиночну похибку.

### ***1.3.2. Послідовні формати передачі даних***

Паралельний формат зберігання і передачі даних використовується при малих відстанях між цифровими пристроями і при зберіганні у напівпровідникових запам'ятовуваних пристроях. У той же час, при записі або зчитуванні на магнітні або оптичні носії цифрова інформація повинна передаватись у послідовному форматі. Послідовний формат використовується і при передачі на великі відстані по телефонних і кабельних лініях зв'язку.

У кожному з таких випадків представлена в паралельному форматі інформація повинна перетворюватись у послідовний за допомогою спеціальних апаратно-програмних засобів.

Базовою концепцією передачі інформації у послідовному форматі є жорстка узгодженість побітної передачі з сигналами синхронізації, тобто кожному періоду синхросигналу повинен ставитись у відповідність один біт інформації, що передається. Інтервал часу між двома тактами синхросигналу є *бітовим інтервалом*, протягом якого по інформаційному каналу передається "0" або "1". У такому випадку, незалежно від довжини слова, яке передається, кількість інформаційних провідників не перевищуватиме трьох – загальний,

інформаційний, синхронізації. В інтервалі тактового періоду генератора синхросигналів на інформаційний провідник від джерела сигналу повинен передаватись один біт у вигляді високого або низького рівня напруги. Частота передачі інформаційних сигналів однозначно визначається частотою синхросигналу, розділеною на довжину (у бітах) слів, що передаються. Здебільшого початок інформаційного сигналу (його фронт) співпадає з початком синхроімпульсу, але така особливість передачі не є обов'язковою.

На жаль, описана форма передачі інформації нереальна, оскільки на приймальній стороні неможливо з потоку інформаційних біт виділити окремі слова, принаймні при неузгодженості роботи приймача і передавача. Тому в ряді випадків до раніше визначених провідників додається четвертий, призначений для визначення початку слів. Така форма обміну інформації використовується в комп'ютерній техніці, а також у системах телекомунікацій.

В інших інформаційних системах – наприклад, телефонних і кабельних мережах – використовується лише двохпровідна лінія передачі. У такому випадку всі сигнали спеціально поєднуються, створюючи достатньо складний код послідовного формату, який скоріше можна розглядати як аналоговий сигнал з складними видами модуляції. Використовується і спосіб, при якому в потоці інформаційних біт має місце встановлений порядок, який відомий і жорстко витримується як на стороні передавача, так і на стороні приймача. Такий порядок називається *протоколом обміну*.

При передачі інформації у послідовному форматі використовується декілька способів побудови послідовних кодів. Один з них полягає у потенціальному представленні логічних рівнів "1" та "0" або протягом всього тактового інтервалу, або на половині його. В обох випадках цифровий сигнал представляється у вигляді однополярних імпульсів. Перший з них у літературі називається *NRZ (non-return to zero)* (рис. 1.11).

Інший спосіб формування цифрової послідовності полягає в тому, що логічні сигнали "1" і "0" можуть представлятись у вигляді різнополярних

імпульсів, дія яких триває на всьому періоді тактового сигналу, або на його частині. Прикладом однополярного коду є код *RZ* (*return to zero*). Після встановлення рівня лог. “1” у момент появи синхросигналу інформаційний сигнал діє на половині періоду, після чого встановлюється в нуль. Код з інверсією одиниці представляється на рис. 1.11 біполярним кодом *BPRZ*. Його особливість полягає у відсутності постійної складової в інформаційному сигналі, що підвищує завадостійкість кіл, які отримують цей сигнал.

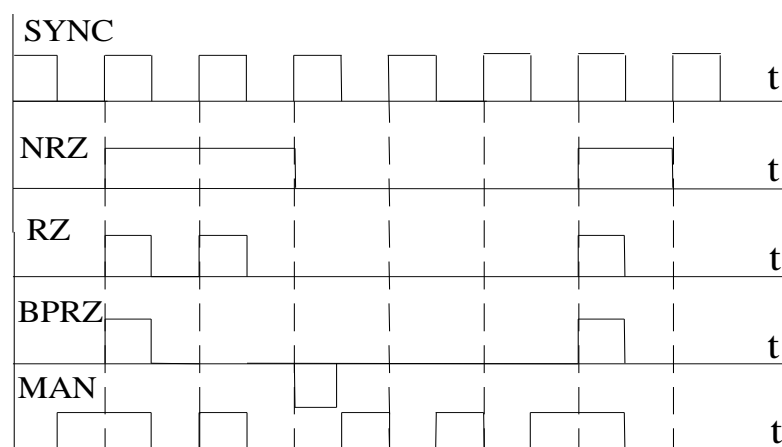


Рис. 1.11

Широко використовуються коди, в яких логічні сигнали кодуються не як потенціальні рівні, а як фронти переходу з “0” в “1” і з “1” в “0”. Прикладом таких кодів є Манчестерський код (*MAN*). Головна перевага такого коду полягає в тому, що незалежно від символів, що передаються, він забезпечує як мінімум одну передачу в бітовому інтервалі. Лог. “0” у такому коді передається як перехід з “0” в “1” посередині бітового інтервалу, а “1” – як перехід з “1” в “0”. Оскільки Манчестерський код має більше переходів 0 – 1 – 0, ніж інші коди, він вимагає більшої смуги пропускання лінії зв’язку. Такі коди широко використовуються в міжкомп’ютерних системах зв’язку.

Кожен із способів має свої недоліки і переваги, які враховуються при прийнятті рішення про їх використання.

### 1.4. Форми зображення чисел

У цифрових пристроях використовуються дві форми зображення чисел: з фіксованою і плаваючою комою.

У попередньому параграфі розглядалися лише цілі позитивні числа. Формула (1.14) дає можливість зображати двійкові числа з цілою та дробовою частиною та з фіксованою комою. Знак двійкового числа з фіксованою комою задається допоміжним розрядом, який встановлюється перед числовими. В додатних чисел значення допоміжного розряду рівне “0”, для від’ємних – “1”. У табл. 1.3 приводяться три варіанти кодування додатних і від’ємних чисел чотирьохрозрядним двійковим кодом.

Таблиця 1.3

$a_3$	$a_2$	$a_1$	$a_0$	Прямий і доповняльний	Знакі величина	Прямий і обернений
0	0	0	0	+0	+0	+0
0	0	0	1	1	1	1
0	0	1	0	2	2	2
0	0	1	1	3	3	3
0	1	0	0	4	4	4
0	1	0	1	5	5	5
0	1	1	0	6	6	6
0	1	1	1	7	7	7
1	0	0	0	-8	-0	-7
1	0	0	1	-7	-1	-6
1	0	1	0	-6	-2	-5
1	0	1	1	-5	-3	-4
1	1	0	0	-4	-4	-3
1	1	0	1	-3	-5	-2
1	1	1	0	-2	-6	-1
1	1	1	1	-1	-7	-0

У першому варіанті, як витікає з таблиці, у кодовій двійковій послідовності мають місце додатній і від’ємний нулі, що призводить до появи проблем при виконанні арифметичних операцій.

Представлення від’ємних чисел у оберненому коді також не вирішує відміченої проблеми. Вона вирішується лише тоді, коли від’ємні числа представляються у доповняльному коді, який обчислюється за формулою:

$$G = B + 1 = \bar{A} + 1.$$

На рис. 1.12 приведена графічна інтерпретація зображення позитивних і негативних чисел відносно нуля з використанням прямого та доповняльного кодів. Як буде показано пізніше, така форма представлення десяткових чисел суттєво спрощує виконання арифметичних операцій.

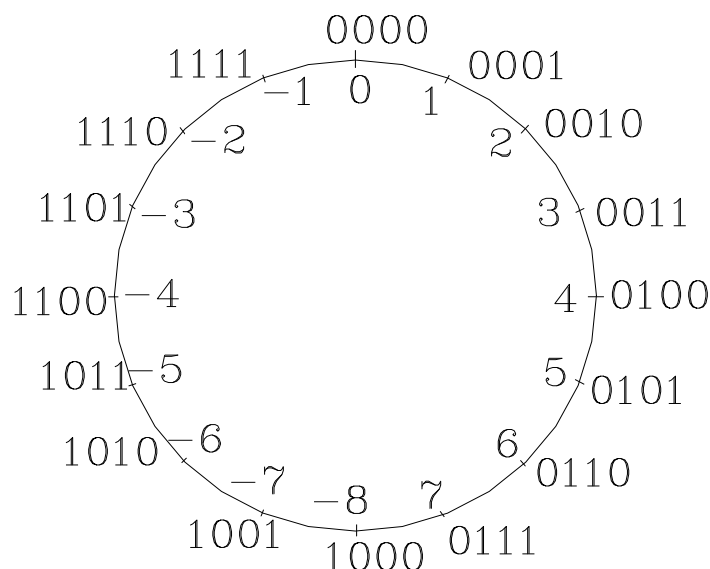


Рис.1. 12

**Приклад 1.12.** Знайти доповняльні коди десятковим числам:  $0_{10}$ ,  $17_{10}$ ,  $-127_{10}$ .

*Розв'язання.* Знаходимо двійкові еквіваленти заданих чисел:

$$0_{10} = 00000000_2; 17_{10} = 00010001_2; -127_{10} = 10000001_2.$$

Знаходимо коди, обернені двійковим – відповідно: 11111111; 11101110; 01111110.

Знаходимо доповняльні коди заданих чисел:  $11111111 + 1 = 100000000_2 = 0_{10}$ ;

$$11101110 + 1 = 11101111_2 = -17_{10}; \quad 01111110 + 1 = 01111111_2 = 127_{10}.$$

Тепер пояснимо суть запису чисел з фіксованою комою. Будь-яке число в цифрових системах зберігається спеціальними пристроями пам'яті, кожен рядок якого складаються з фіксованої кількості елементів. Кома, що відділяє в числі цілу частину від дробової, займає в рядку пам'яті фіксоване положення – перед старшим розрядом або після молодшого.

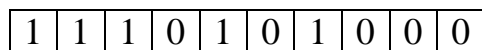


Рис. 1.13

У першому випадку абсолютне значення числа менше одиниці – наприклад,  $0,110101_2$ . Якщо рядок пам'яті призначений для десяти розрядів, то число в ньому запишеться так, як показано на рис. 1.13, де крайній лівий розряд відображає знак числа, а решта – розряди модуля. Вільні молодші розряди заповнюються нулями. Оскільки в розгляданому випадку в рядку пам'яті передбачається запис лише дробової частини числа, то і результати всіх операцій повинні бути з абсолютним значенням, меншим одиниці. Виконання цієї умови забезпечується вибором відповідних масштабних коефіцієнтів, на які помножуються вихідні дані. Якщо масштабний коефіцієнт вибраний невірною, то може з'явитись переповнення розрядів і поява цілої частини, яка буде втрачена, оскільки в розрядній сітці не передбачена її поява. Все це приведе до похибки в результаті, що є недоліком такого способу.

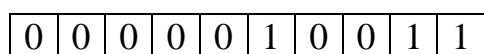


Рис. 1.14

У другому випадку, коли кома фіксується після молодшого розряду, маємо справу з цілими числами. Тоді, наприклад, число  $10011_2$  в рядку пам'яті розміщується в відповідності з рис. 1.14, де лівий розряд знаковий, а слідуючі за ним справа вільні розряди заповнюються нулями. В цьому випадку величина модуля є обмеженою довжиною рядку пам'яті.

Числа з плаваючою комою передбачають зображення числа з використанням мантиси, що помножається на основу системи числення у ступені, який задається порядком. Наприклад, число 200 записується у вигляді  $0,2 \cdot 10^3$ , а число 0,000312 – як  $0,312 \cdot 10^{-3}$ . Відповідно записуються і двійкові числа. Мантиса і порядок зображаються у двійковому коді, а основою є двійка. Наприклад, число  $0,111 \cdot 2^2 = 11 \cdot 10_2$  в десятковій системі зображається як  $0,875 \cdot 2^2 = 3,5_{10}$ . В рядку пам'яті такі числа зберігаються у вигляді двох груп

цифр: перша група – мантиса – визначає саме число, друга – порядок – місце коми в числі (рис.1.15).

0	1	1	1	0	0	0	0	0	0	1	0	0
0	1	2	3	4	5	6	7	8	9	10	11	12

Рис. 1.15

У нульовому елементі рядка пам'яті зображається знак числа (для приведеного вище двійкового числа, що записане у рядок пам'яті – “0”). Далі задаються вісім розрядів самого числа (стовпці 1...8). Якщо воно задається меншою кількістю розрядів, то вільні елементи пам'яті справа від числа заповнюються нулями. У дев'ятому розряді зображається знак порядку, а в решті, за аналогією з мантисою, – число, що визначає порядок. При використанні такої форми запису величина числа порядку задається так, щоб перша значуща цифра мантиси не дорівнювала “0”. Така форма запису називається *нормальною*.

Мінімальне додатне число, що може бути записане при нормальній формі в рядку пам'яті, визначається мінімальною мантисою  $0,1000..0_2$  та максимальним від'ємним порядком  $111..1_2$ . При кількості  $k$  розрядів порядку мінімальне десяткове число, що може бути записаним, визначається формулою:

$$N_{\min} = \frac{1}{2} \cdot 2^{-(2^k - 1)} = 2^{-2^k}. \quad (1.15)$$

Максимальне число матимемо при максимальному значенні мантиси  $(0,111...1)_2$  та максимальному додатному порядку  $(111...1_2) = 2^k - 1$ , тобто

$$N_{\max} = 2^{2^k - 1}. \quad (1.16)$$

Діапазон  $D$  чисел, представлених в нормальній формі, як витікає з формул (1.15) та (1.16), визначається лише числом  $k$ . Наприклад, для  $k = 6$  знаходимо:

$$2^{-2^6} \leq D \leq 2^{2^6 - 1}; \quad 2^{-64} \leq D \leq 2^{63}.$$

Точність запису числа задається кількістю розрядів  $m$  мантиси. Якщо кількість розрядів числа перевершує відведену під мантису кількість розрядів, то число округляється до необхідної довжини. Правило округлення двійкових чисел в цьому випадку таке: якщо старший розряд у частині слова, що відкидається, є одиницею, то до молодшого розряду мантиси додається одиниця. При такому округленні абсолютна похибка  $\varepsilon$  зображення мантиси не перевершує половини вагового коефіцієнта молодшого розряду мантиси, тобто:

$$\varepsilon < \frac{2^{-m}}{2}.$$

Враховуючи, що при нормальній формі запису мантиса не може бути меншою 0.5, відносна похибка  $\eta$ :  $\eta \leq 2^{-m}$ .

Наприклад, при  $m = 24$  маємо:  $\eta \leq 2^{-24} \approx 10^{-7.2}$ .

У сучасних цифрових системах для зображення чисел з плаваючою комою використовується рядок довжиною чотири байти. При цьому 23 розряди задають мантису, а 7 – величину порядку. Діапазон чисел, що зображуються, складає від  $\pm 2^{127}$  до  $\pm 2^{-127}$ .

Використання чисел з плаваючою комою суттєво розширює і спрощує зображення чисел, але виконання операцій над такими числами більш складне, ніж над числами з фіксованою комою.

### 1.5. Виконання арифметичних операцій

Основною операцією, яка використовується в цифрових системах при виконанні різних обчислень, є операція *алгебраїчного додавання*. Вона виконується на основі правил виконання операцій у двійковій системі зображення чисел, які для однорозрядних чисел мають такий вигляд:

$$\begin{array}{cccc}
 + & 0 & + & 1 & + & 0 & + & 1 \\
 & 0 & & 0 & & 1 & & 1 \\
 & 0 & & 1 & & 1 & & 10
 \end{array}$$





Знаходимо суму:

$$\begin{array}{r}
 N = 0' 1 1 0 1 1 \\
 + D = 1' 1 0 0 1 1 \\
 \hline
 (N + D) = 10' 0 1 1 1 0
 \end{array}$$

Відкидаючи 1 переносу в знаковому розряді, отримуємо:

$$N + D = 0' 01110_2 = 14_{10}.$$

**Приклад 1.16.** Змінимо знаки обох чисел на зворотні:

$$N = -27_{10} = 1' 11011_2; \quad K = 13_{10} = 0' 01101_2.$$

*Розв'язання.* Доповняльний код від'ємного числа  $N$ :  $D = 1' 00101_2$ .

Знаходимо суму:

$$\begin{array}{r}
 K = 0' 0 1 1 0 1 \\
 + D = 1' 0 0 1 0 1 \\
 \hline
 (K + D) = 1' 1 0 0 1 0
 \end{array}$$

Знаковий розряд показує, що результат операції від'ємний, а число зображене у доповняльному коді. Для отримання результату в прямому коді необхідно спочатку перейти до оберненого коду, віднімаючи одиницю від результату виконання арифметичної операції, а потім – інвертувати.

Знаходимо обернений код результату:  $1' 10001_2$ ; прямий код:  $1' 01110_2 = -14_{10}$ .

Пряма операція віднімання з використанням операцій зайому зі старших розрядів застосовується лише при порівнянні двох кодів, адже відсутність чи наявність зайому зі старшого розряду дає можливість легко визначити більше з порівнюваних чисел.

Розглянемо тепер особливість виконання операцій у двійково-десятковій системі числення (код **8-4-2-1**). Процедура виконання операції здійснюється на основі правил двійкової арифметики. Якщо одержане число перевищує  $10_{10}$ , то повинна формуватися одиниця переносу, яка передається до наступного десяткового розряду. Але результат відрізнятиметься від правильного, і в нього необхідно внести відповідну корекцію. Наявність у результаті операції числа, що перевершує дев'ятку, легко виявляється за рядом ознак, які читач може сформулювати самостійно.

**Приклад 1.17.** Скласти числа  $A = 5_{10}$  і  $B = 3_{10}$  з використанням коду **8-4-2-1**.

*Розв'язання.*  $A + B = 0101_2 + 0011_2 = 1000_2 = 8_{10}$ .

Оскільки результат менше десяти, то корекція не потрібна.

**Приклад 1.18.** Скласти числа  $A = 8_{10} = 1000_2$  і  $B = 9_{10} = 1001_2$ .

*Розв'язання.*

$$(A + B)_2 = 1000_2 + 1001_2 = 10001_2 = 17_{10}.$$

У даному випадку результат більше десяти, тому необхідно вводити корекцію.

Поява одиниці в п'ятому розряді означає, що число, представлене чотирма молодшими розрядами, збільшилось на 16 одиниць. Але, з іншого боку, одиниця передалась у старший десятковий розряд, що еквівалентно числу 10, тому в молодшому розряді недостає шістьох одиниць.

Це означає, що необхідно до результату додати корекцію – шість одиниць, тобто:

$$\begin{array}{r} + \quad 17_{10} = 1 \ 0 \ 0 \ 0 \ 1 \ 2 \\ \quad 6_{10} = \quad \quad \quad 1 \ 1 \ 0 \ 2 \\ \hline \text{Сума} = 1 \ 0 \ 1 \ 1 \ 1 \ 2 = 0001 \ 0111_{2/10} = 17_{10}. \end{array}$$

Результат представляється одиницею в розряді десятків і сімкою в розряді одиниць.

**Приклад 1.19.** Скласти числа  $6_{10} + 7_{10}$ .

*Розв'язання.*

$$\begin{array}{r} + \quad 6_{10} = 0 \ 1 \ 1 \ 0 \ 2 \\ \quad 7_{10} = 0 \ 1 \ 1 \ 1 \ 2 \\ \hline \text{Сума} = 1 \ 1 \ 0 \ 1 \ 2 = 13_{10} \end{array}$$

У цьому прикладі переносу в п'ятий розряд немає, але результат перевершує дев'ятку, і тому необхідно ввести корекцію. Корекція необхідна не тільки для того, щоб скоригувати результат молодших розрядів, а й для того, щоб перенести одиницю в старший десятковий розряд. Знову додається 10 і відніметься 16, і тому для корекції необхідно додати цифру 6:

$$\begin{array}{r} + \quad 13_{10} = \quad 1 \ 1 \ 0 \ 1 \ 2 \\ \quad 6_{10} = \quad \quad \quad 0 \ 1 \ 1 \ 0 \ 2 \\ \hline \text{Сума} = 1 \ 0 \ 0 \ 1 \ 1 \ 2 = 0001 \ 0011_{2/10} = 13_{10}. \end{array}$$

З двох останніх прикладів бачимо, що в першому випадку при виконанні операції має місце перенесення в старшу тетраду, що і може виступати ознакою необхідності введення корекції результату. В другому випадку такого

перенесення немає, і тому можливість такого результату або необхідно передбачати, або створювати допоміжні заходи для введення корекції. Як вихід з такої ситуації, можна запропонувати використання самокоригуючих кодів – наприклад, коду “з надлишком 3”.

**Приклад 1.20.** Виконати операцію знаходження суми двох чисел з попереднього прикладу при використанні коду “з надлишком 3”.

*Розв’язання.* Відповідні коди чисел:  $6_{10} = 1001$ ;  $7_{10} = 1010$ . Після виконання операції додавання маємо результат:

$$1001 + 1010 = 10011_{2/10} = 0001\ 0011_{2/10} = 13_{10}.$$

При використанні інших кодів для зображення десяткових цифр правила знаходження суми зміняться, але логічний аналіз процедури легко дозволяє знайти правила корекції.

При знаходженні суми багаторозрядних двійково-десяткових чисел від’ємні числа зображуються у зворотному або доповняльному коді. При цьому зворотний код одержується доповненням до 9. Якщо використовувати не код **8-4-2-1**, а коди “з надлишком 3” або **2-4-2-1**, то процедура формування доповнення до 9 значно спрощується.

Виконання операцій множення або ділення двійкових чисел на  $2^j = 2, 4, 8, \dots$  досягається шляхом зсуву цифр числа відповідно вправо або вліво на  $j = 1, 2, 3, \dots$  розрядів.

Для довільних чисел виконання операції множення зводиться до послідовного виконання операцій додавання та зсуву.

**Приклад 1.21.** Виконати множення двох чисел:  $A = 101_2$ ;  $B = 011_2$ .

*Розв’язання.*

$$\begin{array}{r} \times \quad A = 1\ 0\ 1 = 5_{10} \\ \quad B = 0\ 1\ 1 = 3_{10} \\ \hline D_1 = 1\ 0\ 1 \\ + \quad D_2 = 1\ 0\ 1 \\ \hline \text{Сума} = 1\ 1\ 1\ 1 = 15_{10} \end{array}$$

Спочатку виконується знаходження часткового добутку  $D_1$ , потім здійснюється операція зсуву на один розряд вліво і знаходиться частковий добуток  $D_2$ . При наявності більшої кількості часткових добутків знаходяться часткові суми, після чого виконуються операції знаходження часткового добутку та їх зсуву.

Розглянемо ще два приклади. У першому з них операція знаходження суми виконується з переносом, починаючи з другого розряду. У другому демонструється той факт, що множення на нуль не виконується, а приписуються всі нулі справа після виконання операції.

$$\begin{array}{r} \times \quad \quad \quad 1 \ 0 \ 1 \ 1 \\ \quad \quad \quad \underline{1 \ 1 \ 1} \\ \quad \quad \quad 1 \ 0 \ 1 \ 1 \\ \quad \quad \quad 1 \ 0 \ 1 \ 1 \\ \quad \quad \underline{1 \ 0 \ 1 \ 1} \\ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1 \end{array} \quad \quad \quad \begin{array}{r} \times \quad \quad \quad 1 \ 0 \ 1 \ 0 \ 0 \\ \quad \quad \quad \underline{1 \ 0 \ 1 \ 0} \\ \quad \quad \quad 1 \ 0 \ 1 \\ \quad \quad \underline{1 \ 0 \ 1} \\ 1 \ 1 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \end{array}$$

У будь-якому випадку, операція множення складається з операцій зсуву і додавання. Тому в цифровій схемотехніці і мікропроцесорних пристроях вона може виконуватись як на програмному рівні, так і на апаратному.

При виконанні операцій множення і ділення існують спеціальні алгоритми для безпосереднього виконання цих операцій з використанням доповняльних кодів, які описані у відповідній літературі з мікропроцесорної та комп'ютерної техніки.

Найпростіший алгоритм виконання операції двійкового ділення базується на методі зсуву і віднімання. Нижче приводиться приклад використання цього методу для беззнакових десяткових і двійкових чисел.

$$\begin{array}{r} 2 \ 1 \ 7 \ | \ 1 \ 1 \\ \underline{1 \ 1} \quad \quad \quad 1 \ 9 \\ 1 \ 0 \ 7 \\ \quad \underline{9 \ 9} \\ \quad \quad \quad 8 \end{array}$$

$$\begin{array}{r} 1 \ 1 \ 0 \ 0 \ 1 \ | \ 1 \ 0 \ 1 \\ \underline{1 \ 0 \ 1} \quad \quad \quad 1 \ 0 \ 1 \\ \quad \quad \quad 1 \ 0 \ 1 \\ \quad \quad \quad \underline{1 \ 0 \ 1} \\ \quad \quad \quad \quad \quad 0 \end{array}$$

В обох випадках виконання операції множення ми порівнюємо зменшене ділене з множим дільником для визначення, який множник зміщеного дільника відняти. У десятковій системі спочатку вибираємо 11 як найбільший множник, менший 21. Потім вибираємо 99 як найбільший множник, менший 107. При використанні бінарних чисел вибір дещо простіше, оскільки маємо тільки дві можливості для порівняння – нуль і сам дільник.

Методи ділення для двійкових чисел у деякій мірі комплементарні до методу двійкового множення. Типовий алгоритм ділення приймає  $(n + m)$  біт діленого та  $n$  біт дільника і створює  $m$  біт результату та  $n$  біт залишку. Операція ділення є переповнюючою, якщо дільник дорівнює нулю або результат може мати вираз більший, ніж  $m$  біт. У більшості цифрових пристроїв апаратно організовано, що  $n = m$ .

Виконання операцій ділення чисел зі знаком може бути виконано тим же шляхом. Результату приписується додатний знак, якщо знаки чисел, що беруть участь у виконанні операції, однакові, і від'ємний, якщо знаки чисел різні. Залишкові приписується той же знак, що і діленому.

## **1.6. Основи алгебри логіки**

### ***1.6.1. Основні визначення***

У практиці інженерної діяльності часто мають місце ситуації, при яких має значення не рівень сигналів, що поступають з відповідних датчиків, а лише наявність чи відсутність таких сигналів. Наприклад, у системах охоронної сигналізації необхідно знати, замкнені чи не замкнені двері або вікна в приміщенні, що охороняється. У системах автоматики часто необхідно знати, чи не перевершує кількість рідини в цистерні заданий рівень, чи не є тиск у котлі нижчим визначеної межі, чи не перевершує температура в приміщенні задану величину і т. п.

Схеми, що дають можливість розв'язувати поставлені задачі, можуть описуватись виразами типу: “лампочка на пульті охоронної сигналізації горить,

якщо всі вікна замкнені (точніше, замкнено перше і друге і третє і... вікно)”. Або “лампочка не горить, якщо хоча б одне вікно відкрите (тобто може бути відкритим перше **або** друге **або** третє **або** перше і друге **або**...)”. Такі вирази називаються *логічними*.

При проектуванні подібних систем задаються відповідним рівнем напруги живлення, і наявність чи відсутність її дає можливість одержувати відповіді на поставлені питання. Оскільки рівень напруги може бути різним і задаватись прийнятою елементною базою, то з метою формалізації опису подібних схем приймаються деякі умови. Як приклад, високий рівень напруги приймається за “**1**”, низький – відповідно, за “**0**”. У такому разі приведені вище вирази можуть бути формалізовані: якщо контакти, що фіксують положення вікон, позначити як аргументи  $x_1, x_2, \dots, x_n$ , які можуть приймати лише значення “**1**” або “**0**”, то напругу на лампочці можемо розглядати як функцію  $y$ , яка теж приймає одне з двох аналогічних значень.

Математичний апарат, що оперує з аргументами та функціями, які набувають тільки двох значень – “**0**” та “**1**” – називається *двійковою (булевою) алгеброю* або *алгеброю логіки*. Такий математичний апарат для розв’язання задач формальної логіки розробив ірландський математик Дж. Буль.

Логічні змінні (аргументи), як і змінні звичайної алгебри, позначаються літерами латинського алфавіту з різними індексами – наприклад,  $x_0, x_1, x_2, x_3, \dots$ . Індекс при змінній може одночасно означати розряд двійкового числа.

Якщо змінна  $x_i$  набуває значення  $x_i = 1$ , то таке її значення називають *істинним*. Протилежне  $x_i = 0$  називають *хибним* і умовно позначають  $\overline{x_i}$ , що означає заперечення істинного значення аргументу (в зарубіжній практиці операція заперечення позначається апострофом  $x'$ ). Два елементи булевої алгебри – подія істинна і подія хибна – називають її *константами*.

Булева функція позначається літерою  $y$  і є двійковою функцією двійкових аргументів. Умовне її позначення

$$y = f(x_0, x_1, \dots, x_{n-1}).$$

Булева функція, яка залежить від  $n$  аргументів, називається  $n$ -вимірною і є повністю визначеною, якщо вказані значення її для всіх двійкових наборів значень її аргументів. Кількість таких наборів дорівнює  $2^n$ . Тобто, областю визначеності функції  $n$  змінних є сукупність дискретних точок  $n$ -вимірного простору, причому кожна з точок є комбінацією значень цих змінних (кодовою комбінацією). Це, в свою чергу, означає, що кожна функція може бути задана таблицею значень, які вона приймає в точках її області визначеності.

Функція повністю визначена, якщо задані її значення в усіх точках області визначеності. Значення функції вибираються з множини “0” і “1”. Якщо ж значення функції не задано в одній або кількох точках, то вона є *неповністю визначеною*. Кодові комбінації, при яких функція невизначена, називаються *факультативними*. У практиці цифрової схемотехніки існує велика кількість неповністю визначених функцій. Довизначення їх, якщо це необхідно, забезпечується встановленням їх значень – “0” або “1” – довільним шляхом.

Усі можливі логічні функції  $n$  змінних можна створити за допомогою трьох основних операцій:

а) логічне заперечення (інверсія, операція **НІ**); позначається рискою над відповідною функцією або аргументом;

б) логічне додавання (диз’юнкція, операція **АБО**), яке позначається символами ( $\vee$ ), (+);

в) логічне множення (кон’юнкція, операція **І**), яке позначається символами ( $\wedge$ ), ( $\cdot$ ), (&). Для позначення еквівалентності логічних виразів використовується знак (=).

*Запереченням (інверсією)* називається такий зв’язок між аргументом  $x$  та функцією  $y$ , при якому  $y$  істинна тоді і тільки тоді, коли значення  $x$  хибне, і навпаки.

*Логічним множенням (кон’юнкцією)* декількох змінних називається така функція, яка істинна тоді і тільки тоді, коли одночасно істинні всі логічні змінні.



Логічним додаванням (диз'юнкцією) декількох змінних називається така функція, яка хибна тоді і тільки тоді, коли одночасно хибні всі додавані змінні.

Слід пам'ятати, що операція кон'юнкції є старшою операцією і виконується раніше диз'юнкції.

Прикладом найпростіших функцій є наступні:

$$y_1 = \overline{x_1}; \quad y_2 = x_1 \cdot x_2; \quad y_3 = x_1 + \overline{x_2}.$$

**Приклад 1.22.** Записати вираз функції трьох змінних, яка приймає істинні значення при умові, що будь-яка пара змінних одночасно має істинні значення.

*Розв'язання.* Введемо умовні позначення змінних  $x_0, x_1, x_2$ .

У загальному плані функція матиме вигляд:  $y = f(x_2, x_1, x_0)$ .

Оскільки істинні значення функції визначаються будь-якою парою логічних змінних, тобто або  $x_0$  і  $x_1$ , або  $x_0$  і  $x_2$ , або  $x_1$  і  $x_2$ , то аналітична форма запису функції прийме вигляд:

$$y_1 = x_0 \cdot x_1 + x_0 \cdot x_2 + x_1 \cdot x_2.$$

Функція може мати і дещо іншу форму запису, якщо враховувати, що при виконанні будь-якої з умов, що закладені в функцію, обмежень на значення третьої змінної не накладається.

**Приклад 1.23.** Формалізувати та записати у вигляді булевих функцій висловлення: лампочка охоронної сигналізації світиться, коли всі три двері приміщення зачинені.

*Розв'язання.* Позначимо логічні змінні  $x_1, x_2, x_3$  істинними, якщо відповідні двері зачинені. В такому випадку істинне значення функції ("лампочка сигналізації світиться") визначається за формулою:

$$y_2 = x_1 \cdot x_2 \cdot x_3.$$

**Приклад 1.24.** Формалізувати та записати у вигляді булевих функцій висловлення: температурна сигналізація вмикається, коли хоча б один з двох датчиків зафіксує температуру  $70^\circ$ .

*Розв'язання.* Прийmemo за істинне значення логічних змінних  $x_1, x_2$  показ датчика, що відповідає температурі  $70^\circ$ .

У такому випадку логічна функція має вигляд:

$$y_3 = x_1 + x_2 + x_1 \cdot x_2.$$

**Приклад 1.25.** Змагання зі штанги судять три судді: головний, що знаходиться проти помосту, і два бокові. Якщо суддя вважає, що вага взята, він натискає кнопку, яка знаходиться на його столі. Для спортсмена вага вважається взятою, якщо загоряється лампочка біля помосту. Умова загорання лампочки наступна: вона загоряється, якщо головний і, як мінімум, один з бокових суддів натиснули кнопки на своїх столах. Формалізувати умову загорання лампочки.

*Розв'язання.* Прийнемо за логічні змінні кнопки, що знаходяться на столах суддів:  $x_1$  – кнопка головного судді;  $x_2$  та  $x_3$  – бокових. Прийнемо за істинне значення натиснуту кнопку  $x_i = 1$ . Тоді умова загорання лампочки формально може бути записана в вигляді:

$$y = x_1 x_2 x_3 + x_1 x_2 \overline{x_3} + x_1 \overline{x_2} x_3.$$

Технічна реалізація булевих функцій, а, відповідно, і їх фізична інтерпретація добре ілюструється за допомогою контактних схем, в яких логічна змінна  $x_i$  відповідає замкненому контакту. Схеми, що ілюструють реалізацію операцій кон'юнкції та диз'юнкції, наведені відповідно на рис. 1.15, а, б.

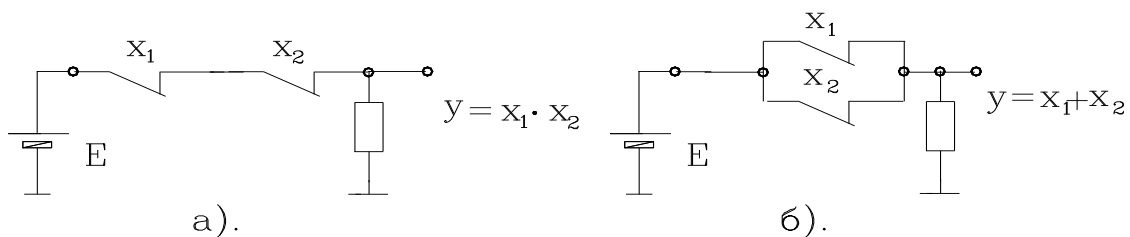


Рис. 1.15

### 1.6.2. Закони і тотожності алгебри логіки

В алгебрі логіки використовується ряд аксіом (тотожностей) та законів. Основними з них є наступні: *переміщувальний (властивість комутативності); сполучний (властивість асоціативності); розподільний (властивість дистрибутивності); інверсії (теорема де Моргана).*

Головні аксіоми та закони булевої алгебри наведені у табл. 1.4.

Використовуючи наведені у табл. 1.4 закони та тотожності, які використовуються при перетворенні логічних функцій, можна створювати нові.

Наприклад:

$$x_1 \cdot (\overline{x_1 + x_2}) = x_1 \cdot x_2;$$

$$x_1 \cdot \overline{x_2} + x_2 = x_1 \cdot \overline{x_2} + x_2(x_1 + \overline{x_1}) = x_1 \cdot \overline{x_2} + x_1 \cdot x_2 + \overline{x_1} \cdot x_2 = x_1 + x_2$$

(у подальшому крапки, що відображають операцію логічного множення у формулах, для спрощення запису приводити не будемо).

Таблиця 1.4

Назва аксіоми чи закону	Вирази
Аксіоми (тотожності)	$0 \cdot x = 0$ $1 + x = 1$ $0 + x = x$ $x \cdot x = x$ $x + x = x$ $x \cdot \overline{x} = 0$ $x + \overline{x} = 1$ $\overline{\overline{x}} = x$
Закони комутативності	$x_1 + x_2 = x_2 + x_1$ $x_1 \cdot x_2 = x_2 \cdot x_1$
Закони асоціативності	$x_1 + x_2 + x_3 = x_1 + (x_2 + x_3) = (x_1 + x_2) + x_3 = (x_1 + x_3) + x_2$ $x_1 \cdot x_2 \cdot x_3 = x_1 \cdot (x_2 \cdot x_3) = x_2 \cdot (x_1 \cdot x_3) = x_3 \cdot (x_1 \cdot x_2)$
Закони дистрибутивності	$x_1 \cdot (x_2 + x_3) = x_1 \cdot x_2 + x_1 \cdot x_3$ $x_1 + x_2 \cdot x_3 = (x_1 + x_2) \cdot (x_1 + x_3)$
Закони інверсії (теорема де Моргана, принцип подвійності)	$\overline{x_1 \cdot x_2} = \overline{x_1} + \overline{x_2}$ $\overline{x_1 + x_2} = \overline{x_1} \cdot \overline{x_2}$
Закони поглинання	$x_1 + x_1 \cdot x_2 = x_1$ $x_1 \cdot (x_1 + x_2) = x_1$

Закони інверсії, які відображають властивість взаємного перетворення операцій логічного множення і додавання в алгебрі логіки, називають *принципом подвійності*.

### 1.6.3. Способи задання логічних функцій

Існують такі способи задання або запису логічних функцій – *аналітичний, табличний, за допомогою карт Карно, графічний та кубічний*.

*Аналітично* логічна функція може бути записана різними комбінаціями кон'юнкцій та диз'юнкцій логічних змінних. Зазвичай логічні функції записуються або у вигляді суми добутків логічних змінних (диз'юнкція кон'юнкцій) або у вигляді логічного добутку їх сум (кон'юнкція диз'юнкцій). Наведення функції у вигляді диз'юнкції кон'юнкцій називають *диз'юнктивною нормальною формою (ДНФ)*:

$$y = x_1 \bar{x}_2 + \bar{x}_1 x_3 + x_1 x_2 \bar{x}_3 ,$$

а запис у вигляді кон'юнкції диз'юнкцій – відповідно, *кон'юнктивною нормальною формою (КНФ)*:

$$y = (x_1 + x_2)(x_2 + \bar{x}_3)(\bar{x}_1 + x_2 + x_3) .$$

Інверсія у відповідності з теоремою де Моргана будь-якої функції, приведений в одній формі, призводить до заміни запису на іншу форму.

Наприклад, інверсія функції  $y = x_1 + x_2 \bar{x}_3 + x_1 \bar{x}_2 x_3$  представляється у вигляді:  $\bar{y} = \bar{x}_1 (\bar{x}_2 + x_3) (\bar{x}_1 + x_2 + \bar{x}_3)$ .

Будь-яка логічна функція, задана в аналітичній формі, може бути перетворена на **ДНФ** або **КНФ** за допомогою тотожностей та законів алгебри логіки. При цьому для однієї і тієї ж функції може існувати декілька рівнозначних диз'юнктивних та кон'юнктивних нормальних форм.

У той же час, існує лише один вид **ДНФ** та **КНФ**, в яких функція може бути записана єдиним чином. Такі форми називаються *досконалими диз'юнктивними (кон'юнктивними) нормальними формами (ДДНФ, ДКНФ)*.

Вони характеризуються тим, що в **ДДНФ** кожна кон'юнкція, а в **ДКНФ** кожна диз'юнкція містять усі логічні змінні даної функції, з інверсіями або без них.

Прикладами **ДДНФ** та **ДКНФ** запису є функції чотирьох змінних

$$y_1 = \overline{x_3} \overline{x_2} \overline{x_1} x_0 + \overline{x_3} x_2 \overline{x_1} x_0 + x_3 \overline{x_2} \overline{x_1} x_0 ;$$

$$1010 \quad 0110 \quad 1011$$

$$y_2 = (x_1 + \overline{x_2} + \overline{x_3} + x_4)(\overline{x_1} + x_2 + \overline{x_3} + x_4).$$

Оскільки кожна кон'юнкція функції, що наведена у **ДДНФ**, визначає її істинне значення, відповідаюче "1", то такі кон'юнкції називаються *конституєнтами одиниці (мінтермами)*. Аналогічно, диз'юнкції функції, що наведені у **ДКНФ**, називаються *конституєнтами нуля (макстермами)*.

Якщо замінити логічні змінні та їх заперечення одиницями та нулями, то кожна кон'юнкція буде представляти собою двійкове число.

Це дозволяє, наприклад, вище наведену логічну функцію  $y_1$  записати у вигляді:

$$y_1 = \bigvee_0^{15} 6, 10, 11.$$

Така форма називається *досконалою скороченою диз'юнктивною формою* або *канонічною сумою*.

**Приклад 1.26.** Функцію з прикладу 1.25 зобразити в скороченій диз'юнктивній канонічній формі.

*Розв'язання.* Перепишемо функцію, дещо змінивши нумерацію і порядок розміщення змінних:  $y = x_2 \cdot x_1 \cdot x_0 + x_2 \cdot x_1 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} \cdot x_0.$

Скорочена диз'юнктивна канонічна форма приведеної функції матиме вигляд (у порядку розміщення кон'юнкцій):

$$y = \bigvee_0^7 5, 6, 7.$$

Аналогічно, функцію можна зобразити і у вигляді добутку макстермів. Така форма запису називається *канонічним добутком*. Наприклад:

$$y = \bigwedge_0^7 2, 4 = (x_2 + \overline{x_1} + x_0)(\overline{x_2} + x_1 + x_0).$$

Легко бачити можливість конвертації в представленні функції у вигляді макстермів та мінтермів, оскільки кожна з них доповнює функцію до повного перебору логічних змінних. Як приклади, можемо записати:

$$y = \bigvee_0^7 2, 6, 7 = \bigwedge_0^7 0, 1, 3, 4, 5;$$

$$y = \bigvee_0^7 0, 1, 3, 5 = \bigwedge_0^7 2, 4, 6, 7;$$

$$y = \bigvee_0^{15} 0, 4, 5, 9, 11, 13, 15 = \bigwedge_0^{15} 1, 2, 3, 6, 7, 8, 10, 12, 14.$$

Індекси біля умовних позначень операцій диз'юнкції та кон'юнкції вказують на діапазон можливих мінтермів та макстермів логічних функцій. Нижній індекс іноді не вказується.

Досконала диз'юнктивна нормальна форма запису дозволяє легко перейти до інших форм запису – *табличної* та *карт Карно*.

Таблиця 1.5

$x_1$	$x_0$	$y_1$	$y_2$	$y_3$	$y_4$	$y_5$
0	0	0	0	0	0	1
0	1	1	1	1	0	0
1	0	1	1	1	0	0
1	1	x	1	0	1	1

У табл. 1.5 наведені функції  $y_1 \dots y_5$  двох змінних  $x_0$  та  $x_1$ . Табличний спосіб полягає у тому, що функція задається у вигляді таблиці відповідності (таблиці істинності станів).

У таблицю вписують усі можливі комбінації аргументів у порядку зростання їх індексів і при кожній комбінації встановлюється значення функції. Кількість всіх можливих сплук аргументів, а, отже, і кількість значень функції дорівнює  $2^n$ , де  $n$  – кількість логічних змінних. З табличної форми запису легко перейти до аналітичної, використовуючи досконалу диз'юнктивну форму запису логічних функцій. Для цього функція записується як диз'юнкція конституент одиниці. Наприклад, функцію  $y_3$  з табл.1.5 можемо записати у вигляді:  $y_3 = \overline{x_1} x_0 + x_1 \overline{x_0}$ .

Ця функція може бути записана і з використанням нульових її значень:

$$\overline{y_3} = \overline{x_1} \overline{x_0} + x_1 x_0.$$

Використовуючи властивість подвійної інверсії, легко встановити тотожність обох форм запису.

**Приклад 1.27.** Зобразити функцію з прикладу 1.25 у табличній формі запису.

*Розв'язання.* Функція трьох змінних у табличній формі (Табл. 1.6) складатиметься з чотирьох стовбців, у трьох з яких буде розміщені повний перебір логічних змінних.

Логічна функція іноді може бути неповністю визначеною. У табл. 1.5 приводиться форма запису функції  $y_1$ , яка є невизначеною при  $x_1 = x_0 = 1$ . При переході до аналітичної форми запису вона повинна бути довизначена.

Таблиця 1.6

$N$	$x_2$	$x_1$	$x_0$	$y$
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Функції двох змінних займають у булевій алгебрі особливе місце. Для двох змінних кількість булевих функцій дорівнює 16. Ці функції називаються *елементарними* і складають максимальний набір функцій двох змінних. У табл. 1.7 наведені всі елементарні функції двох змінних.

Таблиця 1.7

$N$	$x_0$	0	1	0	1	Назва функції	Позначення
	$x_1$	0	0	1	1		
0		0	0	0	0	Константа нуль	0
1		0	0	0	1	Кон'юнкція, <b>I</b>	$x_1 \cdot x_0$
2		0	0	1	0	Заборона по $x_0$	$x_1 \cdot \overline{x_0}$
3		0	0	1	1	Змінна $x_1$	$x_1$
4		0	1	0	0	Заборона по $x_1$	$\overline{x_1} \cdot x_0$
5		0	1	0	1	Змінна $x_0$	$x_0$
6		0	1	1	0	<b>Викл. АБО</b> , сума за mod 2	$x_1 \oplus x_0$
7		0	1	1	1	<b>Диз'юнкція, АБО</b>	$\overline{x_1 + x_0}$
8		1	0	0	0	<b>АБО-НІ</b> , функція Пірса	$\overline{x_1 + x_0}$
9		1	0	0	1	Рівнозначність, еквівалентність	$x_1 \equiv x_0$
10		1	0	1	0	Заперечення $x_0$	$\overline{x_0}$
11		1	0	1	1	Імплікація по $x_0$	$\overline{x_1 + x_0} \cdot x_0 \rightarrow x_1$
12		1	1	0	0	Заперечення $x_1$	$\overline{x_1}$
13		1	1	0	1	Імплікація по $x_1$	$x_1 \rightarrow \overline{x_0} \cdot x_1 + x_0$
14		1	1	1	0	<b>Функція Шеффера I-НІ</b>	$\overline{x_1 x_0}$
15		1	1	1	1	Константа 1	1

За своєю застосованістю вони різні. Найбільш широке використання знаходять функції **I**, **АБО**, **I-НІ**, **АБО-НІ**, **ВИКЛ. АБО**. Вони універсальні тому, що за їх допомогою можна записати будь-яку іншу функцію.

Розроблений математичний апарат аналізу та синтезу булевих функцій найбільш відповідає цим функціям. Набір логічних функцій **I**, **АБО**, **НІ** дозволяє реалізувати будь-яку іншу функцію і називається *функціонально повним*.

У практичній схемотехніці найбільш поширеними є системи, які реалізують логічні функції **I-НІ**, **АБО-НІ**, **ВИКЛ. АБО**. Вони дозволяють найбільш просто реалізовувати різні функції, мати більшу кількість входів, прості в технічній реалізації.

### **Теорема Шеннона**

Широке використання при перетворенні логічних функцій знаходять теорема Шеннона та ряд тотожностей, які витікають з неї.

Теорема Шеннона формулюється так: будь-яку функцію  $n$  змінних можна зобразити в формі:

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \bar{x}_i \cdot f(x_{n-1}, \dots, 0, \dots, x_0) + x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0). \quad (1.17)$$

Теорема Шеннона виявляється дуже корисною при виконанні перетворень логічних виразів, що містять операцію **ВИКЛ. АБО**.

**Приклад 1.27.** Виконати перетворення логічної функції:

$$y = x_2 x_1 \oplus (x_3 + \bar{x}_1) \oplus x_3 x_1.$$

*Розв'язання.* Використовуючи теорему Шеннона, виконаємо наступний ряд перетворень:

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

З теоремою Шеннона (1.17) пов'язані тотожності:

$$\begin{aligned} \bar{x}_i \cdot f(x_{n-1}, \dots, x_i, \dots, x_0) &= \bar{x}_i \cdot f(x_{n-1}, \dots, 0, \dots, x_0); \\ x_i \cdot f(x_{n-1}, \dots, x_i, \dots, x_0) &= x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0). \end{aligned} \quad (1.18)$$



Виходячи з теореми де Моргана, тотожностям (1.18) відповідають наступні тотожності:

$$\begin{aligned}\overline{x_i} + f(x_{n-1}, \dots, x_i, \dots, x_0) &= \overline{x_i} + f(x_{n-1}, \dots, 1, \dots, x_0); \\ x_i + f(x_{n-1}, \dots, x_i, \dots, x_0) &= x_i + f(x_{n-1}, \dots, 0, \dots, x_0).\end{aligned}\tag{1.19}$$

Тотожності (1.18) і (1.19) широко використовуються для спрощення логічних виразів. З них витікають формули, які широко використовуються:

$$\begin{aligned}x + x \cdot y &= x + 0 \cdot y = x; \\ x + \overline{x} \cdot y &= x + \overline{0} \cdot y = x + y.\end{aligned}$$

Приведені тотожності дають можливість суттєво спрощувати складні функції багатьох змінних, особливо при наявності заперечень.

**Приклад 1.28.** Спростити логічну функцію:

$$f(x_1, x_2, x_3) = \overline{\overline{x_2 \cdot x_1 \oplus x_3 \cdot x_2 \oplus x_1 + x_3 \cdot x_2 \cdot x_1}}.$$

Використовуючи тотожність (1.18) відносно  $\overline{x_2}$ , маємо:

$$\overline{0 \cdot x_1 \oplus x_3 \cdot \overline{0} \oplus x_1 + x_3 \cdot \overline{0} \cdot x_1} = \overline{x_3 \oplus x_1 + x_3 \cdot x_2}.$$

З тотожності (1.19) знаходимо:

$$\overline{x_3 \oplus x_1 + x_3} = \overline{x_3 + \overline{0} \oplus x_1} = \overline{x_3 + x_1}.$$

В результаті знаходимо:

$$\overline{\overline{x_3 + x_1 \cdot x_2}} = \overline{x_3 \cdot x_2 \cdot x_1}.$$

Приведені вище тотожності використовуються для того, щоб розкласти складні логічні функції на більш прості.

### *Розкладання Ріда*

Широке використання знаходить також інший тип розкладання функцій – розкладання Ріда. Воно базується на використанні наступної умови: якщо  $x \cdot y = 0$ , то  $x + y = x \oplus y$ .

Дійсно,

$$x \oplus y = x \cdot \overline{y} + \overline{x} \cdot y + x \cdot y + \overline{x} \cdot \overline{y} = x(y + \overline{y}) + y(x + \overline{x}) = x + y.$$

Це дає можливість розкладання Шеннона

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \overline{x_i} \cdot f_0 + x_i \cdot f_1,$$

де

$$f_0 = f(x_{n-1}, \dots, 0, \dots, x_0); \quad f_1 = f(x_{n-1}, \dots, 1, \dots, x_0),$$

зобразити у вигляді:

$$\begin{aligned} f(x_{n-1}, \dots, x_i, \dots, x_0) &= \overline{x_i} \cdot f_0 + x_i \cdot f_1 = \overline{x_i} \cdot f_0 \oplus x_i \cdot f_1 = \\ (1 \oplus x_i) f_0 \oplus x_i \cdot f_1 &= f_0 \oplus f_0 \cdot x_i \oplus x_i \cdot f_1 = f_0 \oplus (f_0 \oplus f_1) x_i. \end{aligned}$$

Отриманий вираз називається *розкладанням Ріда*. Воно дає можливість будь-яку функцію  $n$  змінних зобразити у вигляді полінома  $n$ -го ступеня виду:

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = a_0 \oplus \sum_{i=0}^{n-1} b_1 x_i \oplus \sum_{i=0}^{n-1} b_2 x_i x_k \oplus \dots$$

де  $a_0, b_1, b_2, \dots$  – коефіцієнти ряду Ріда.

Так, наприклад, функція двох змінних може бути записана у вигляді:

$$f(x_1, x_0) = a_0 \oplus b_0 x_0 \oplus b_1 x_1 \oplus b_2 x_1 x_0.$$

Якщо функція описується поліномами першого ступеня, вона називається *лінійною*. Для таких функцій коефіцієнт при логічних добутках змінних дорівнює нулю. У такому випадку для функції двох змінних можемо записати:

$$f(x_1, x_0) = a_0 \oplus b_0 x_0 \oplus b_1 x_1,$$

а для функції з кількістю змінних  $n$

$$f(x_1, x_0) = a_0 \oplus b_0 x_0 \oplus b_1 x_1 \oplus b_2 x_2 \dots \oplus b_{n-1} x_{n-1}.$$

**Приклад 1.29.** Зобразити функцію

$$f(x_1, x_0) = \overline{x_1} x_0 + x_1 \overline{x_0}$$

у вигляді лінійного полінома.

*Розв'язання.* Виконуючи перетворення, знаходимо:

$$f(0, x_1) = x_1;$$

$$f(1, x_1) = \overline{x_1};$$

$$f(x_1, x_0) = x_1 \oplus [\overline{x_1} \oplus x_1] \cdot x_0 = x_1 \oplus x_0.$$

У цифрових пристроях, які реалізують ті чи інші логічні функції, сигнали здебільшого розділяються за функціональним призначенням – наприклад, інформаційні, адресні, керуючі – і, відповідно, мають свої умовні позначення. Це дає можливість за виглядом логічних функцій чітко бачити особливості взаємодії між сигналами, взаємозалежність між ними. Деякі приклади таких функцій:

$$f(a_0, x_1, x_2) = \overline{a_0} x_1 + a_0 x_2 ;$$

$$f(a_1, a_0, x_3, x_2, x_1, x_0) = x_0 a_0 a_1 + x_1 \overline{a_0} a_1 + x_2 \overline{a_0} \overline{a_1} + x_3 a_0 \overline{a_1} .$$

Приведені функції описують роботу простих мультиплексорів. Як буде зрозуміло з подальшого матеріалу (**Розділ 3**), коефіцієнти  $a_0$  і  $a_1$  є також логічними змінними, які мають інше функціональне призначення.

У той же час, остання функція може бути представлена у вигляді:

$$f(x_3, x_2, x_1, x_0) = b_1 x_0 + b_2 x_1 + b_3 x_2 + b_4 x_3 ,$$

де  $b_1 \dots b_4$  – коефіцієнти, функціонально залежні від  $a_0$  і  $a_1$ .

У якості логічних сигналів можуть використовуватись функції. У такому випадку описана логічна функція реалізує функціональний комутатор.

### ***Геометрична інтерпретація логічних функцій***

Однією з форм геометричної інтерпретації логічних функцій є використання карт Карно, представлених у **ДДНФ** (**ДКНФ**).

*Карта Карно* – це компактна форма представлення таблиці істинності логічної функції із застосуванням для позначення (кодування) комбінацій змінних циклічного коду Грея.

Карта для двох логічних змінних наведена на рис. 1.16. Особливість карти Карно полягає в тому, що по горизонталі та по вертикалі задаються координати клітинок, якими виступають аргументи логічної функції. Тому кожна клітина має свою координату – 00, 01, 10, 11 – яка може бути представлена відповідною двійковою або десятковою цифрою. Значення

	$x_0$	0	1
$x_1$	0	0	1
	1	1	0

Рис. 1.16

функції записуються в відповідних клітинах нулями та одиницями. Наприклад, логічна функція  $y = \bigvee_0^3 1, 2$  зображена двома одиницями і двома нулями у відповідних клітинах карти Карно (рис. 1.16).

При використанні карт Карно для функцій з більшою кількістю логічних змінних кількість клітин у карті зростає відповідно до формули  $N = 2^n$ .

**Приклад 1.30.** Функцію, що приведена в скороченій канонічній формі:

$$y = \bigvee_0^{15} 2, 3, 4, 5, 10, 11, 12, 13,$$

зобразити у вигляді карти Карно.

*Розв'язання.* Оскільки числові величини мінтермів перевищують число 7 і знаходяться в межах 15, то приведена логічна функція має чотири змінні, які представимо як  $x_3, x_2, x_1, x_0$ . Карта Карно для чотирьох змінних має вигляд, приведений на рис. 1.17.

	$x_1 x_0$			
	00	01	11	10
$x_3 x_2$	0	1	3	2
00	0	0	1	1
01	4	5	7	6
	1	1	0	0
11	12	13	15	14
	1	1	0	0
10	8	9	11	10
	0	0	1	1

Рис. 1.17

	$x_1 x_0$			
	00	01	11	10
$x_2$	0	1	3	2
0	0	0	0	0
1	4	5	7	6
	0	1	1	1

Рис. 1.18

З карти Карно також легко перейти до табличної форми запису і навпаки.

**Приклад 1.31.** Зобразити карту Карно для функції, що представлена в табл. 1.6.

*Розв'язання.* Відповідна карта Карно приведена на рис. 1.18.

Якщо задана функція неповністю визначена, то, подібно до таблиці, у відповідних клітинах записується знак невизначеності, що дає можливість повністю відобразити всі властивості функції.

Ще однією формою геометричної інтерпретації логічних функцій, яка досить широко використовується останнім часом, є *n-вимірні куби*, або *кубічні комплекси*.

На рис. 1.19 приведені геометричні зображення  $n$ -вимірного куба для різних значень  $n$ , що відповідають словам двійкового коду. Кожна вершина відображає одне кодове слово.

Наприклад, функція  $y = x$  відображається на рис. 1.19, а вершиною з позначенням **1**.

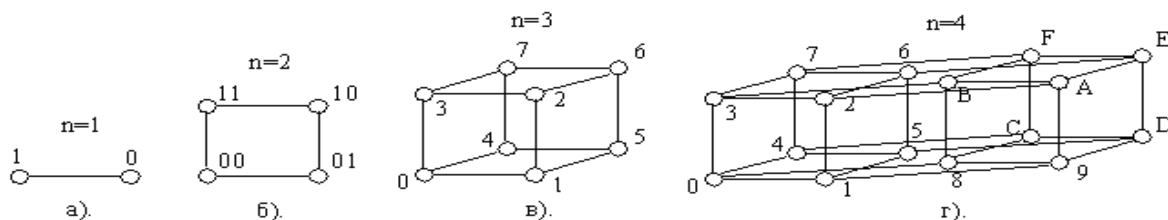


Рис.1. 19

Функція  $y = x_1 x_2 + \overline{x_1} \overline{x_2}$  на рис. 1.19, б відображається двома вершинами, з'єднаними прямою лінією (ребром кубу), тому що приймає одиничне значення при комбінаціях аргументів 11 та 00.

Кожна вершина куба, в якій функція має мінтерм, називається *нульовим кубом* (0-кубом). Функція, яка містить декілька мінтермів, зображається в просторі *нульовим кубічним комплексом*  $K_0$ .

Якщо два нульових куби комплексу  $K_0$  відрізняються тільки по одній змінній, то два сусідніх мінтерми створюють *одиночний куб* (1-куб), який зображається геометрично як ребро  $n$ -вимірного куба. Множина одиночних кубів створюють *одиночний кубічний комплекс*  $K_1$ .

Аналогічно, якщо два одиночних куба комплексу  $K_1$  відрізняються тільки по одній координаті, то вони створюють *двійковий куб* (2-куб). Геометрично таке зображення є гранню  $n$ -вимірного куба. Множина двійкових кубів створюють *кубічний комплекс*  $K_2$ .

Об'єднання кубічних комплексів  $K_0, K_1, K_2, \dots, K_m$  логічної функції  $y(x_0 \dots x_n)$  створюють її *кубічний комплекс*:

$$K(y) = \vee(K_0, K_1, \dots, K_m).$$

**Приклад 1.32.** Для логічної функції

$$y = \bigvee_0^7 3, 4, 5, 6, 7$$

записати кубічні комплекси.

*Розв'язання.* Нульовий кубічний комплекс має п'ять членів за кількістю мінтермів логічної функції:

$$K_0 = (011, 100, 101, 110, 111).$$

Порівнюючи записані 0-куби, знаходимо, що сусідніми є мінтерми 3 і 7, які відрізняються лише старшим розрядом. Тому вони створюють 1-куб виду -11. Аналогічно, мінтерми 4 і 5 створюють 1-куб виду 10-, а мінтерми 4 і 6 створюють 1-куб виду 1-0.

Повний аналіз показує, що одиничний кубічний комплекс матиме вигляд:

$$K_1 = (-11, 10-, 1-0, 11-, 1-1).$$

Двійковий кубічний комплекс містить один 2-куб:

$$K_2 = (1- -).$$

Графічна інтерпретація логічних функцій ґрунтується на використанні умовних позначень логічних елементів, які є стандартними.

На рис. 1.20 приведені умовні позначення, характерні для стандартів, прийнятих у країнах СНД.

Використання приведених умовних позначень дає можливість будувати складні логічні та принципові схеми електронних пристроїв. Логічні елементи, умовні зображення яких приведені на рис. 1.20, реально виготовляються в різних серіях цифрових мікросхем.

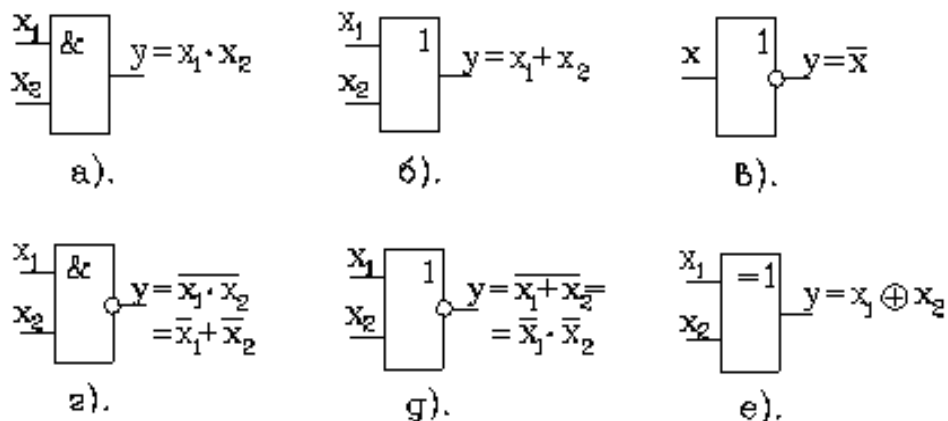


Рис.1. 20

**Приклад 1.33.** Побудувати схему, що відповідає функції

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

*Розв'язання.* Використовуючи лише однофункціональні логічні елементи, будується схема, що приведена на рис. 1.21.

Ще однією формою представлення логічних функцій є *часові діаграми*, які відображають часові співвідношення між вхідними логічними змінними і вихідною функцією (рис. 1.22). Часові діаграми зображаються з урахуванням фронтів імпульсних послідовностей з відображенням моментів початку перехідних процесів і часто мають допоміжні вказівні стрілки, які уточнюють хід перехідних процесів у схемі.

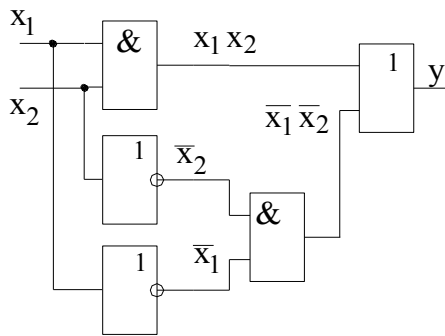


Рис.1.21

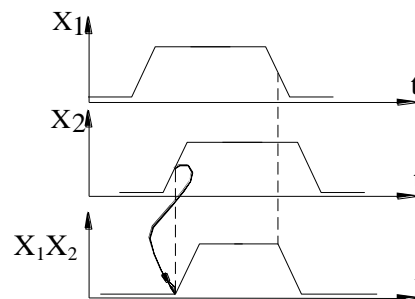


Рис.1.22

На рис. 1.22 приводиться приклад часових діаграм для фрагменту логічної схеми, зображеної на рис.1.21, що реалізує операцію цифрового компаратора. У випадку, що розглядається, приводяться лише вхідні та вихідні сигнали логічного елементу і. При визначенні інтервалів часових затримок, тривалості перехідних процесів приводиться більша деталізація часових діаграм.

#### ***1.6.4. Мінімізація логічних функцій***

Метою мінімізації є зменшення вартості технічної реалізації логічних функцій незалежно від використовуваних апаратних засобів.

Логічні функції апаратно реалізуються за допомогою мікросхем, орієнтованих на виконання тих чи інших операцій. Мікросхеми загального

використання здебільшого можуть реалізовувати декілька простих одиночних операцій. З цієї причини справедливо стверджувати, що чим простішою є аналітична форма запису логічної функції, тим менше використовується логічних елементів і, як результат, тим менше мікросхем необхідно для її реалізації. Складність логічних функцій визначається кількістю логічних змінних, що входять до їх складу в прямому і інверсному виді, та кількістю простих логічних операцій над ними. Будь-яка логічна функція може бути записана різними аналітичними виразами різного рівня складності. Серед них можна знайти такі, які містять мінімальну кількість логічних змінних і операцій над ними. Задача знаходження таких аналітичних виразів називається *мінімізацією логічних функцій*. Звідси витікає, що *мінімізація логічної функції* – це заміна логічної функції, що представлена у вигляді логічної суми мінтермів або логічного добутку макстермів, іншою логічною функцією з мінімальною кількістю логічних змінних та операцій над ними.

Задача мінімізації – це задача неоднозначна, і різними шляхами можна отримати різні вирази мінімізованої функції, які відрізнятимуться між собою кількістю змінних і операцій над ними.

**Аналітичний спосіб мінімізації.** Для зменшення складності логічних функцій здебільшого використовуються операції склеювання:

$$x_1 x_2 + x_1 \overline{x_2} = x_1 (x_2 + \overline{x_2}) = x_1 ;$$

$$(x_1 + x_2) (x_1 + \overline{x_2}) = x_1$$

та поглинання:

$$x_1 + x_1 x_2 = x_1 (1 + x_2) = x_1 ;$$

$$x_1 (x_1 + x_2) = x_1 .$$

Як приклад, розглянемо процедуру спрощення наступної функції:

$$y = x_1 x_2 x_3 + x_1 x_2 \overline{x_3} + x_1 \overline{x_2} x_3 + x_1 \overline{x_2} \overline{x_3} + x_1 x_2 x_3 =$$

$$= x_2 x_3 (x_1 + \overline{x_1}) + (\overline{x_1} x_2 + x_1 \overline{x_2}) (x_3 + \overline{x_3}) = x_2 x_3 + \overline{x_1} x_2 + x_1 \overline{x_2} =$$

$$= x_2 x_3 + x_1 (x_2 + \overline{x_2}) = x_2 x_3 + x_1 .$$

Одержана ДНФ має мінімальну складність.



**Мінімізація за допомогою карт Карно (діаграм Вейча).** Метод, оснований на використанні карт Карно, характеризується своєю простотою і наочністю. Зображення функції в площині координат її параметрів, подібно до алгебраїчних функцій, дозволяє наглядно встановити взаємозв'язки між її змінними і, як результат, виділити ті змінні, які є домінуючими в її визначенні.

Для пояснення методу перш за все сформулюємо основні властивості карт Карно:

- клітини карти, координати яких відрізняються лише параметрами однієї змінної, називаються *сусідніми*;
- сусідні клітини, значення функцій в яких або тільки істинні, або тільки хибні, можуть об'єднуватися в групи по  $2^m$  клітин, де  $m$  – ціле число ( $m = 0, 1, 2, 3, \dots$ );
- при переході до аналітичної форми запису логічної функції з карти Карно вона може записуватись незмінними координатами об'єднаних груп клітин;
- у випадку неповністю визначеної функції невизначені клітини можуть бути довизначеними, виходячи з умови одержання більшої кількості об'єднаних клітин;
- кожна клітина може об'єднуватись у декілька груп.

Використання карт Карно для мінімізації логічних функцій базується на наглядному використанні операції склеювання. Дійсно, дві сусідні клітки відрізняються лише однією змінною. Тому, об'єднуючи їх, ми записуємо лише незмінні координати, тобто виносимо їх за дужки. У дужках залишаються змінні координати, які об'єднуються в відповідальну константу. Для отримання мінімального значення функції,

		$X_1 X_0$			
		00	01	11	10
$X_3 X_2$	00	0 1	1 0	3 0	2 1
	01	4 0	5 1	7 1	6 0
	11	12 0	13 1	15 1	14 0
	10	8 1	9 0	11 0	10 1

Рис.1. 23

представленої картою Карно, окрім правил, викладених вище, необхідно користуватись загальним правилом: одиниці або нулі повинні об'єднуватись мінімальним числом найбільших контурів.

**Приклад 1.34.** Мінімізувати логічну функцію, що представлена у вигляді карти Карно (рис. 1.23).

*Розв'язання.* Об'єднуючи сусідні клітини з істинними значеннями функції (тобто клітини 0, 2, 8, 10 та клітини 5, 7, 13, 15), записуємо їх незмінні координати:

- для центральних клітин:  $y_1 = x_0 x_2$ ;
- для крайніх клітин:  $y_2 = \overline{x_0} \overline{x_2}$ .

Мінімізована логічна функція має вигляд:

$$y = y_1 + y_2 = x_0 x_2 + \overline{x_0} \overline{x_2}.$$

Аналогічний результат буде одержаний і при об'єднанні клітин з нулями. Продемонструємо це.

Для вертикальної групи клітин (клітини 1, 3, 9, 11) знаходимо:  $\overline{y_3} = \overline{x_2} x_0$ .

Для групи горизонтальних клітин (клітини 4, 6, 12, 14):  $\overline{y_4} = x_2 \overline{x_0}$ .

Мінімізована функція  $\overline{y} = \overline{x_2} x_0 + x_2 \overline{x_0}$  операцією інверсії та теоремою де Моргана зводиться до отриманої раніше.

За допомогою карт Карно легко вирішуються задачі мінімізації функцій з кількістю змінних до шести включно. При більшій кількості змінних пошук мінімальних форм запису функцій значно ускладнюється і наочність карт Карно втрачається.

**Мінімізація на основі використання кубічних комплексів.** Кубічна форма зображення логічних функцій дає можливість просто і наглядно, за аналогією з використанням карт Карно, розв'язувати задачі мінімізації. Особливість такого способу мінімізації полягає в тому, що з кубічного комплексу  $K(y)$  завжди можна виділити множину кубів  $\Pi(y)$  таких, що кожний член комплексу  $K_0$ , тобто вершини куба (мінтерми), буде включеним по крайній мірі в один куб множини  $\Pi(y)$ . Множина кубів  $\Pi(y)$  називається *покриттям комплексу  $K(y)$* , або *покриттям логічної функції*. Зрозуміло, що для

будь-якої логічної функції існує декілька її покриттів. У той же час, кожному покриттю  $\Pi(y)$  відповідає своя диз'юнктивна нормальна форма, отримана як сума логічних добутоків відповідно виділеним кубам логічної функції.

**Приклад 1.35.** Для кубічного комплексу з прикладу 1.31 знайти покриття логічної функції.

*Розв'язання.* Кубічний комплекс логічної функції має вигляд:

$$K(y) = (011; 100; 101; 110; 111; -11; 11-; 1-1; 10-; 1-0; 1--).$$

Нульовий кубічний комплекс включає всі вершини куба, тому створює покриття функції:

$$\Pi_1(y) = K_0 = (011; 100; 101; 110; 111).$$

Усі вершини куба включаються також в одиничний кубічний комплекс  $K_1$ , тому він також створює покриття логічної функції:

$$\Pi_2(y) = K_1 = (-11; 11-; 1-1; 10-; 1-0).$$

Перебираючи комбінації кубів різних рангів, можна отримати покриття логічної функції:

$$\Pi_3(y) = K_2 = (011; 11-; 10-);$$

$$\Pi_4(y) = K_3 = (-11; 1-1; 1-0);$$

$$\Pi_5(y) = K_4 = (011; 1--);$$

$$\Pi_6(y) = K_5 = (-11; 1--)$$

і т.п.

Відповідні вказаним покриттям логічні функції мають вигляд:

$$y_1 = \overline{x_2} \cdot \overline{x_1} \cdot \overline{x_0} + \overline{x_2} \cdot \overline{x_1} \cdot x_0 + \overline{x_2} \cdot x_1 \cdot \overline{x_0} + \overline{x_2} \cdot x_1 \cdot x_0 + x_2 \cdot \overline{x_1} \cdot \overline{x_0};$$

$$y_2 = x_1 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + x_2 \cdot x_0 + \overline{x_2} \cdot x_1 + x_2 \cdot \overline{x_0};$$

$$y_3 = \overline{x_2} \cdot x_1 \cdot \overline{x_0} + \overline{x_2} \cdot x_1 + \overline{x_2} \cdot x_1;$$

$$y_4 = x_1 \cdot \overline{x_0} + x_2 \cdot \overline{x_0} + \overline{x_2} \cdot x_0;$$

$$y_5 = \overline{x_2} \cdot x_1 \cdot \overline{x_0} + x_2;$$

$$y_6 = x_1 \cdot \overline{x_0} + x_2.$$

Складність отриманої таким шляхом логічної функції прийнято характеризувати поняттям "ціна покриття" ( $\Pi$ ). Ціна покриття дорівнює сумі цін всіх кубів, які складають дане покриття  $\Pi(y)$ :

$$\Pi = \sum \Pi_k.$$

У свою чергу, ціна одного  $r$ -куба логічної функції  $n$  змінних визначається як різниця повного числа вхідних змінних і рангу відповідного куба, тобто дорівнює числу змінних у відповідній диз'юнкції:

$$Ц_k = n - r .$$

Так, для логічної функції трьох змінних ціна 0-куба дорівнює трьом, а 2-куба – одиниці.

У відповідності до сказаного, задача мінімізації логічних функцій зводиться до пошуку покриття  $\Pi(y)$  кубічного комплексу  $K(y)$ , що має мінімальну ціну.

Покриття  $\Pi(y)$  комплексу  $K(y)$ , що має мінімальну ціну, називається *покриттям Квайна*, а відповідна цьому покриттю логічна функція – *мінімальною диз'юнктивною нормальною формою*.

**Мінімізація неповністю визначених логічних функцій.** Серед логічних функцій зустрічаються такі, досконалі форми яких містять невизначені мінтерми (макстерми). З точки зору інженерної практики це означає, що електронна апаратура, яка розробляється, повинна бути байдужою до відповідних комбінацій логічних змінних. Зустрічаються й інші ситуації, при яких в деяких комбінаціях змінних має місце неоднозначність логічної функції. В обох випадках наявність неоднозначності використовується з метою забезпечення мінімальної форми логічної функції.

Розглянемо приклад логічної функції, що приведена на рис. 1.24.

	$X_1 X_0$			
	00	01	11	10
$X_2$				
0	0	1	3	2
	0	1	1	0
1	4	5	7	6
	1	X	1	X

Рис.1. 24

Невизначеність її значень, що відображена у клітинах 5 та 6, може бути використана для мінімізації. Підстановка в указані клітини значень “1” дає можливість об'єднати групи клітин 1, 3, 5, 7 і 4, 5, 6, 7 та одержати мінімізовану функцію:  $y = x_0 + x_2$ .

Слід зазначити, що для більш складних функцій можуть використовуватись декілька різних способів до визначення, в результаті можна отримувати мінімальні функції різної складності.

Недовизначені функції досить часто зустрічаються при реалізації перетворювачів кодів.

**Сумісна мінімізація декількох логічних функцій.** У практичних задачах цифрової електроніки дуже часто зустрічаються ситуації, коли використовується декілька різних функцій з однаковими змінними. Як приклад, розглянемо дві наступні функції:

$$Y_1 = \bigvee_{3, 6, 7} = \overline{x_2} x_1 x_0 + x_2 x_1 \overline{x_0} + x_2 x_1 x_0;$$
$$Y_2 = \bigvee_{0, 1, 3} = \overline{x_2} \overline{x_1} \overline{x_0} + \overline{x_2} \overline{x_1} x_0 + \overline{x_2} x_1 x_0.$$

Кожну з них можна мінімізувати окремою.

При такій мінімізації маємо:

$$Y_1 = x_1 x_0 + x_2 x_1;$$
$$Y_2 = \overline{x_2} \overline{x_1} + \overline{x_2} x_1 x_0.$$

Обидві функції можуть реалізовуватись окремо. Але, порівнюючи приведені функції, бачимо, що в обох з них є загальний мінтерм. Це дає підставу окремо реалізувати загальний мінтерм, а решту мінімізувати незалежно для кожної з функцій. У такому випадку маємо:

$$Y_1 = \overline{x_2} x_1 x_0 + x_2 x_1;$$
$$Y_2 = \overline{x_2} \overline{x_1} + \overline{x_2} x_1 x_0.$$

При побудові схеми окремо реалізуються всі терми, а потім виконується їх об'єднання через функцію АБО.

**Мінімізація логічних функцій з великою кількістю змінних.** Практичні комбінаційні схеми можуть мати десятки-сотні входів і виходів і описуватись сотнями-тисячами термів. У таких випадках для мінімізації широко

використовується метод декомпозиції. Він полягає у наступному. Логічна функція з великою кількістю змінних – наприклад,

$$y = f(x_0, x_1, x_2, \dots, x_{10}),$$

розбивається на декілька підлеглих функцій виду:

$$y = f(x_0, x_1), f_0(x_2, x_3, x_4), f_1(x_5, x_6), f_2(x_7, \dots, x_{10}).$$

При виборі функцій  $f_0, f_1, f_2$  схемотехніки використовують власний досвід, але при їх вдалому виборі апаратна реалізація мінімізованої функції може бути значно економнішою, ніж будь-які способи прямої реалізації.

Слід зазначити, що інтенсивний розвиток мікросхемотехніки, у тому числі програмованих логічних інтегральних схем (ПЛІС), постійних запам'ятовуючих пристроїв (ПЗП), а також ряду функціональних мікросхем змінив акценти в задачах мінімізації. Широке використання ПЛІС привело до розвитку методів їх програмування, які одночасно використовують і алгоритмічні засоби мінімізації кількості логічних елементів, що використовуються. Одним з таких алгоритмічних методів є *метод Квайна – Мак-Класкі*.

**Мінімізація логічних функцій на ЕОМ за допомогою метода Квайна – Мак-Класкі.** Метод використовується при необхідності мінімізації логічних функцій з великою кількістю змінних. Він, як і інші подібні методи, що мають властивість однозначності алгоритму, широко використовується і розвиваються розробниками програмних засобів для ПЛІС. Здебільшого це фірми-виробники програмованих логічних матриць (наприклад, ALTERA, XILLINX та інші).

Алгоритм пошуку мінімальних диз'юнктивних форм зводиться до наступного:

1. Знаходиться покриття  $\Pi(y)$  заданої функції. Для цього формується кубічний комплекс логічної функції і в кожному  $i$ -му кубічному комплексі відмічають куби (імпліканти), які не утворили  $(i + 1)$ -й кубічний комплекс. Відмічені імпліканти, які називаються *простими*, створюють покриття заданої логічної функції;

2. Будується таблиця покриттів матриці Квайна. Рядки вказаної таблиці відповідають простим імплікантам, а стовбці – 0-кубам (конституентам одиниці) логічної функції. На перетині  $i$ -го рядку та  $j$ -го стовбця ставиться мітка, якщо імпліканта  $i$  покриває конституенту  $j$  (імпліканта  $i$  покриває конституенту  $j$  в тому випадку, якщо вона відрізняється від неї незалежними аргументами);

3. Визначають покриття мінімальної вартості. Для цього:

- виділяють ядро Квайна. Якщо 0-куб заданої логічної функції покривається лише однією простою імплікантою, то остання є суттєвою та входить в ядро Квайна і, відповідно, у покриття мінімальної вартості;

- з таблиці викреслюються стовбці і рядки, які покриті імплікантами ядра Квайна. Якщо в отриманій після викреслювання таблиці містяться прості імпліканти, вони також включаються в ядро Квайна з послідуочим викресленням відповідних рядків і стовбців;

- стискається таблиця по стовбцях, для чого з неї викреслюються стовбці, які повністю включаються в будь-який з решти стовбців;

- стискається таблиця по рядках, для чого з неї викреслюються рядки, які повністю включаються в будь-яку з решти рядків.

Послідовно стискаючи таблицю по рядках і стовбцях, отримують *циклічну таблицю*, імпліканти якої повинні входити в покриття логічної функції мінімальної вартості.

## **1.7. Коди, що знаходять та виправляють помилки**

### ***1.7.1. Особливості кубічної форми представлення логічних функцій***

Зображення логічних функцій у вигляді  $n$ -вимірного кубу відкриває велику можливість не тільки в задачах мінімізації, а також в інших напрямках цифрової схемотехніки. Така геометрична інтерпретація надає можливість по-іншому бачити можливість різних кодів. Наприклад, представлення коду Грея у

вигляді 3-вимірного куба має ту особливість, що код кожної сусідньої вершини відрізняється лише в одному біті (рис. 1.25).

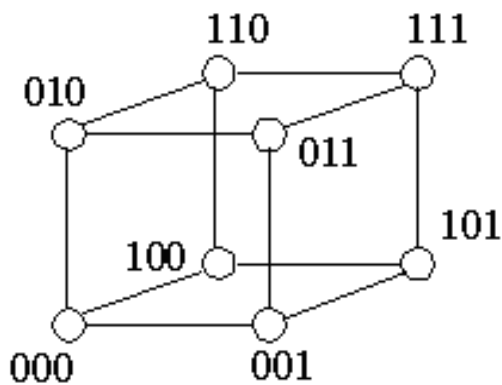


Рис.1. 25

Важливим моментом такої геометричної інтерпретації є *відстань між двома  $n$ -бітними словами*, тобто кількість біт, по яких два слова або дві вершини відрізняються між собою. З точки зору  $n$ -кубу – це мінімальна довжина шляху, який необхідно пройти (по ребрах кубу) від однієї вершини до іншої.

Це, фактично, правило, за яким розраховується кодова відстань. Інше правило розрахунку кодової відстані полягає у знаходженні суми за модулем 2 між двома кодами, які приписані вершинам кубу. Після цього підраховується кількість одиниць в отриманій сумі.

**Приклад 1.36.** Знайти кодову відстань між двома вершинами  $n$ -вимірного кубу, що описуються наступними кодовими комбінаціями:  $A_1 = 10010$ ;  $A_2 = 10111$ .

*Розв'язання.* Знаходимо суму за модулем 2:

$$A_1 \oplus A_2 = 00101.$$

Кодова відстань  $d = 2$ .

Поняття кодової відстані має суттєве значення при розв'язанні задач побудови кодів з виявленням помилок при передачі інформації.

Структури  $n$ -кубу широко використовуються при побудові мультикомп'ютерних і мультипроцесорних обчислювальних систем.

Будь-який  $n$ -куб, подібно до таблиці станів, може розглядатись як поєднання більш простих однотипних структур. Така властивість витікає з таблиць відповідності більшості кодів. Наприклад, двійкові коди десяткових чисел 0 – 3 повторюються в кодах чисел 4 – 7 з різницею в старшому біті коду, що кодує числа в діапазоні 0 – 7.



### 1.7.2. Коди з виявленням і корекцією помилок

Ідеологія побудови таких кодів полягає в тому, що при виникненні одиночної помилки в кодовому слові це слово вже не відноситься до множини кодових слів даного коду. Така ідеологія легко пояснюється на прикладі геометричної інтерпретації коду –  $n$ -вимірного кубу – з використанням поняття *кодової відстані* або *кодового діапазону*.

На рис. 1.26, а приведений приклад коду (кодові слова позначені більш темними вершинами), в якому зміна в одному біті (наприклад, кодове слово **110** замінюється на **111**), залишає це слово в наборі кодових слів, і тому така помилка при передачі інформації не може бути виявлена.

Для коду, що представлений у вигляді  $n$ -кубу, зображеного на рис. 1.26, б, зміна одного біта будь-якого кодового слова призводить до того, що воно випадає зі словника коду, тому поява помилки в одному біті при передачі коду може бути одразу виявлена.

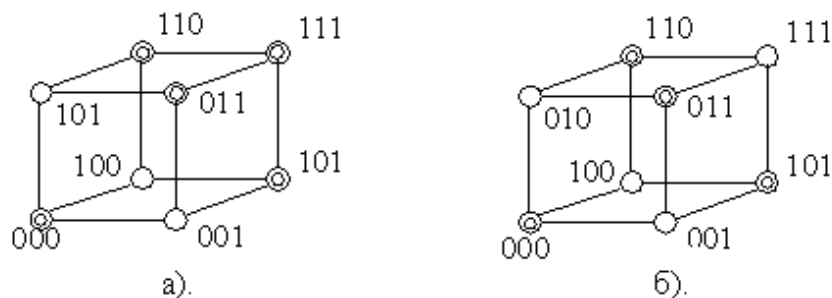


Рис.1. 26

Виходячи з терміну “мінімальна відстань між двома кодовими словами  $n$ -кубу”, можна стверджувати, що в конкретному коді можуть біти виявлені всі одиночні помилки, якщо мінімальна відстань між будь-якими парами кодових слів дорівнює 2.

У загальному плані, якщо ми маємо шину передачі даних в  $n$  біт, то вона може забезпечувати передачу  $2^n$  кодових слів. Для того, щоб виявити одиночну помилку, необхідно мати лише один допоміжний біт, оскільки це надає можливість у  $n$ -вимірному кубі мати мінімальну відстань – 2. На практиці такий біт називається *бітом парності*, що пов’язано з особливостями схемотехніки.

### 1.7.3. Коди, що коригують одиночні помилки і виявляють помилки більшої кратності

Використання більшої кількості бітів у кодових словах дає можливість зробити мінімальну кодову відстань більшою ніж 2, що відкриває можливість не тільки виявляти, а й виконувати корекцію спотворених кодових слів. Прикладом коду, який надає можливість легко виявити помилку в передачі, є *подвійно-п'ятирічний код*. Його значення приведені в табл. 1.8.

Таблиця 1.8

$N$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$	$b_0$
0	0	1	0	0	0	0	1
1	0	1	0	0	0	1	0
2	0	1	0	0	1	0	0
3	0	1	0	1	0	0	0
4	0	1	1	0	0	0	0
5	1	0	0	0	0	0	1
6	1	0	0	0	0	1	0
7	1	0	0	0	1	0	0
8	1	0	0	1	0	0	0
9	1	0	1	0	0	0	0

Старші два біти  $b_6 b_5$  розділяють всі коди на дві групи: молодших десяткових чисел 0 – 4, які кодуються однаково  $b_6 b_5 = 01$ ; старших десяткових чисел 5 – 9, які мають код  $b_6 b_5 = 10$ .

Молодші 5 розрядів формують коди 1 з 5. Як видно з таблиці, поява помилки в будь-якому біті такого коду легко може бути виявлена, а інформація буде визнана помилковою.

Розглянемо тепер, як можна використовувати коди для корекції одиночних помилок або виявлення більшої їх кількості.

Допустимо, що код має мінімальну кодову відстань 3.

Рис. 1.27 показує фрагмент  $n$ -вимірного кубу для такого коду з кодovими словами **0000** і **0111**. При відстані 3 маємо три біти різниці між двома сусідніми вершинами кубу. Допустимо тепер, що передається інформація і при її передачі має місце помилка в одному біті. Якщо передається кодове слово **0000**, то з

помилкою можуть бути отримані слова, які на рис. 1.27 оточують вказане кодове слово.

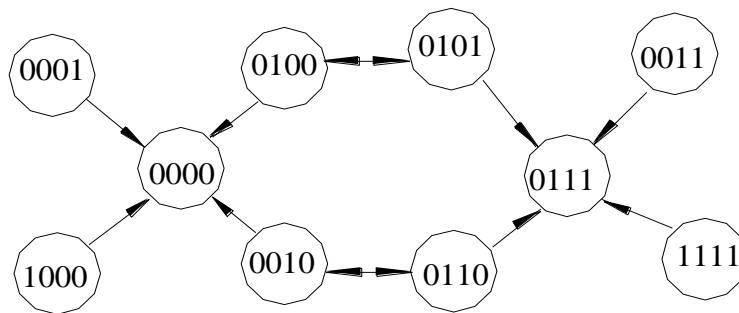


Рис. 1.27

У такому випадку при появі одиночної помилки з'являються позакодові слова, які відрізняються від кодового слова лише на один біт. Вони не тільки легко можуть бути виявлені, а й виправлені, що відображено стрілками на рис. 1.27. Аналогічна картина має місце і стосовно другого кодового слова. Наприклад, створення позакодового слова **0101** з більшою ймовірністю відповідає кодовому слову **0111**, ніж **0000**.

Таким чином, при прийомі інформації необхідно приймати рішення, яке з кодових слів дійсно отримане. Таке рішення є *операцією декодування з виправленням помилок*, а відповідні апаратні засоби називаються *декодерами з виправленням помилок*. Коди, що використовуються для виправлення помилок при передачі інформації, називаються *коригуючими*.

У загальному плані, якщо код має відстань  $d = 2b + 1$ , то він може бути використаним для корекції помилок, що впливають на  $b$  бітів. Якщо ж коди мають мінімальну відстань  $d = 2b + c + 1$ , то вони можуть бути використані для корекції помилок у  $b$  бітах і виявляти помилки в  $c$  бітах.

Прикладом такого коду є *код Хемінга*. Він може бути створений для будь-якого цілого числа  $i$ , яке визначає код довжиною  $(2^i - 1)$  біт з  $i$  розрядів парності та  $(2^i - 1 - i)$  інформаційних розрядів з відстанню між кодовими словами 3.

Положення кожного розряду слова коду Хемінга нумерується від 1 до  $(2^i - 1)$ . Розряди з номерами розміщення в слові  $2^i$ , є *бітами парності*, решта – *інформаційні*. Кожен біт парності об'єднаний у групу з підконтрольними йому інформаційними бітами. Таке об'єднання показано в табл.1.9.

З табл. 1.9 бачимо, що бітами парності є розряди 1, 2, 4.

В першу групу перевірки А входять розряди (символи) коду Хемінга двійкові номери яких мають одиницю в молодшому розряді (всі непарні).

В другу групу перевірки В входять розряди з двійковими номерами, що мають одиницю в другому розряді, в третю групу С входять розряди, номери яких мають одиницю в третьому (старшому) розряді. В кожну контрольну групу входить один біт парності, який стоїть на місці з номером  $2^j$  і ряд інформаційних символів. Значення контрольних бітів вибирається таким чином, щоб загальна кількість одиниць в кожній групі була парною.

Таблиця 1.9

Номер розряду	7	6	5	4	3	2	1	Контрольний розряд
Код розряду	111	110	101	100	011	010	001	
Група А	X		X		X		X	$2^0(1)$
Група В	X	X			X	X		$2^1(2)$
Група С	X	X	X	X				$2^2(4)$
Біти парності				$2^2$		$2^1$	$2^0$	

При більшій кількості інформаційних символів з'являються додаткові біти парності, місця знаходження яких та значення ("0" чи "1") знаходяться по тим же правилам, що і попередні.

У табл. 1.10 приводиться перелік кодових слів для відстані 3 коду Хемінга з чотирма інформаційними бітами.

Таблиця 1.10

Інформаційні біти	Біти парності	Інформаційні біти	Біти парності	Інформаційні біти	Біти парності	Інформаційні біти	Біти парності
0000	000	0100	110	1000	111	1100	001
0001	011	0101	101	1001	100	1101	010
0010	101	0110	011	1010	010	1110	100
0011	110	0111	000	1011	001	1111	111

Якщо при передачі інформації в  $j$ -му розряді кодового слова зміниться значення біта, то, відповідно, зміниться парність у кожній групі, що містить розряд  $j$ . Оскільки кожен інформаційний біт міститься по крайній мірі в одній групі, то по крайній мірі в одній групі буде порушена парність і визначена наявність позакодового слова.

Якщо ж у кодовому слові зміняться два розряди, то групи парності, що містять обидва розряди, не зможуть визначити помилку в слові, котре передається, оскільки парність порушена не буде. Але, оскільки біти з помилками можуть бути лише в різних розрядах, то їх представлення у групах також буде різним. Це означає, що в одній з груп буде представлений лише один з пошкоджених бітів. У такому випадку в цій групі контролем парності буде визначена наявність позакодового слова.

Розглянемо на конкретному прикладі особливості визначення позакодового слова і корекцію помилки.

Допустимо, що на приймальній стороні системи передачі інформації прийнято слово **0100111**. Перевіримо це слово, в якому 7-й, 5-й і 4-й розряди мають нульові значення, за допомогою таблиці парності. Результати перевірки заносяться у контрольний розряд. Для групи *A* 7-й і 5-й розряди мають нульові значення, а 1-й і 3-й – одиниці. Тобто в групі маємо парну кількість одиниць і в контрольний розряд у відповідності з контролем парності заноситься нуль. Аналогічно, в групі *B* маємо одиниці в розрядах 6, 3, 2, тобто непарну їх кількість, і в контрольний розряд заноситься одиниця. Для групи *C* у розрядах 7, 6, 5, 4 маємо лише одну одиницю, і в контрольний розряд теж записуємо “1”.

У відповідності до вагових коефіцієнтів груп, у контрольному розряді записано в двійковому коді число 6, якому відповідає біт кодового слова з прийнятою помилкою. У шостому розряді кодового слова стоїть “1”, а повинен бути “0”. Апаратними засобами така помилка може бути виправлена.

Коди Хемінга з кодовою відстанню 3 і 4 знаходять широке використання для визначення і виправлення помилок у модулях пам'яті комп'ютерних систем

великої ємності – наприклад, в мультипроцесорних системах і мейнфреймах. Привабливість їх використання в таких модулях обумовлена тим, що кількість бітів парності при зростанні розрядності довжини слова зростає в меншій мірі. Наприклад, при кількості інформаційних біт 4 кількість біт парності дорівнює 3, а при кількості інформаційних біт 26 кількість біт парності зростає до 5.

#### **1.7.4. Двовимірні коди**

Інший шлях отримання кодів з великою мінімальною кодовою відстанню – це використання *двовимірних кодів*. Інформаційні біти формуються у двовимірну матрицю і біт парності забезпечує перевірку рядків та стовбців. Код  $C_r$  з мінімальною кодовою відстанню  $d_r$  використовується для перевірки рядків, а, відповідно, код  $C_c$  з мінімальною кодовою відстанню  $d_c$  використовується для перевірки стовбців. Це означає, що біт парності вибирається так, що кожен рядок є кодовим словом у  $C_r$ , а кожен стовбець є кодовим словом у  $C_c$ . Мінімальна відстань двовимірного коду є функцією  $d_r$  і  $d_c$ . Найпростіший двовимірний код використовує один біт парності по рядках і один по стовбцях і має мінімальну відстань  $2 \times 2 = 4$ . Це означає, що будь-яка помилка в 1, 2, 3 бітах призводить до появи непарності у рядку, стовбці або в обох одночасно. Невизначена помилка може мати місце лише при наявності помилок у чотирьох бітах.

Двовимірні коди використовуються для перевірки системи пам'яті персональних комп'ютерів, яким характерна двовимірна матрична структура  $n$  слів по  $m$  біт.

### **1.8. Завади та їх характеристики**

Поряд з корисними інформаційними сигналами, в електронних системах співіснують електричні та електромагнітні сигнали, які заваджають нормальному функціонуванню пристроїв. Такі сигнали називають *перешкодами*, *завадами* та *наводками*. Вони завдають багато проблем

конструкторам і схемотехнікам електронної апаратури, адже знижують її надійність. Тому проблемам боротьби з завадами приділяється завжди не менше уваги, ніж розробці якісних схем.

*Завада* – це небажаний електричний або електромагнітний сигнал, що діє на пристрій або його окремі вузли і може привести до спотворення інформації, яка зберігається, перетворюється чи обробляється. Завади розділяються на *зовнішні* та *внутрішні* по відношенню до пристрою.

*Зовнішні завади* – це сигнали, що діють на електронний пристрій, проникати через джерело живлення, інформаційні входи та заземлення як електромагнітні високочастотні коливання та імпульси, що наводять ЕРС на провідниках схеми.

*Внутрішні завади* – це сигнали, які генеруються окремими вузлами та блоками самого пристрою. Фактично їх причиною виступають реальні, а не ідеалізовані параметри електронних схем та компонентів, які не враховуються при проектуванні. Такі завади можуть з'являтися на будь-яких ділянках пристрою і впливати на роботу його компонентів.

Як і інформаційні сигнали, завади характеризуються своїми параметрами.

*Параметри завад* – це детерміновані або статистичні величини, які характеризують конкретну властивість завад по відношенню до інформаційних сигналів.

Сигнали завад можуть бути як регулярними, так і випадковими, незалежно від місця їх появи по відношенню до схеми.

*Регулярні завади* – це здебільшого гармонічні складові, які попадають на схеми від джерела живлення та силових комутаторів. Їх параметри аналогічні параметрам полігармонічних сигналів.

*Випадкові завади* розділяються на флуктуаційні шуми безперервного характеру та імпульсні завади і характеризуються законами розподілення та кореляційними функціями часу.

*Імпульсні завади* – це послідовність імпульсів довільної форми, амплітуди

та тривалості, що з'являються у випадкові моменти часу і тривалість яких набагато менша інтервалів між ними. У зв'язку з тим, що імпульсні завади проходять через паразитні реактивні елементи зв'язку, на вхід чутливих елементів проходять не самі завади, а їх похідні першого та вищих порядків.

Поняття *наводки* (наведеної ЕРС) включається в поняття завади і пов'язано з наявністю незапланованої електромагнітної взаємодії між елементами схеми, або між елементами схеми та зовнішніми потужними джерелами електромагнітного випромінювання.

Основним джерелом завад у будь-якому електронному пристрої є джерело живлення, яке передає як довготривалі завади від електричної мережі, так і виступає самостійним генератором регулярних та випадкових завад. Напруга електричних мереж характеризується такими параметрами, як провали та сплески, перенапруги, раптові відключення, коливання та відхилення відносно номінальних параметрів. Для потужних електронних систем та автономних мереж необхідно враховувати такі явища, як несиметрія напруг по фазах, спотворення лінійних і фазних напруг внаслідок роботи нелінійних навантажень та статичних перетворювачів, що приводить до появи вищих гармонічних складових, а також потужні електромагнітні імпульси, які пов'язані з комутаційними процесами у довгих лініях.

Усі вказані причини неідеальності напруг електричних мереж здебільшого розглядаються як джерела різноманітних завад, котрі з деякою ідеалізацією можуть розглядатись як сума адитивних завад. Тобто реальна напруга мережі  $u(t)$  представляється як сума ідеальної напруги  $u_i(t)$  та напруг враховуваних

завад  $\sum_k u_{zk}(t)$ :

$$u(t) = u_i(t) + \sum_k u_{zk}(t), \quad (1.13)$$

де індекс  $k$  визначає кількість враховуваних завад.

Оскільки сучасні джерела живлення електронних пристроїв будуються на основі високочастотних перетворювачів електроенергії з забезпеченням



необхідного рівня стабілізації вихідної напруги, то амплітуди вказаних завод будуть знижені компенсаційними стабілізаторами у  $K_k$  разів ( $K_k$  – коефіцієнт зниження амплітуди  $k$ -ї завади у вихідній напрузі джерела живлення).

З деякою ідеалізацією схема заміщення джерела живлення електронної апаратури приводиться на рис. 1.28, де  $r_f$ ,  $C_f$  – параметри вихідного фільтра;  $u_{п1}(t)$  – детермінований ряд гармонік джерела живлення;  $E_g$ ,  $r_g$  – параметри джерела живлення;  $u_{32}(t) = \sum_k u_{3k}(t)$ .

Гармоніки вихідної напруги джерела, як відомо, за допомогою зворотніх зв'язків компенсаційного стабілізатора у багатьох випадках не знижуються, і для їх зменшення до допустимих параметрів використовуються пасивні фільтри.

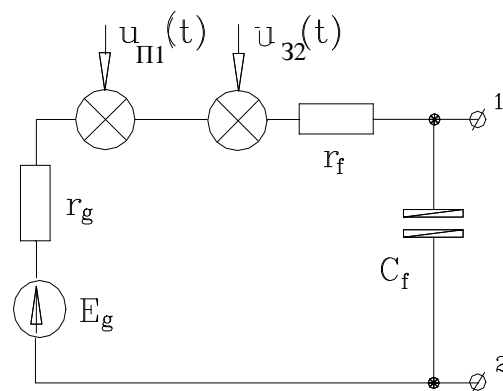


Рис.1.3

Звернемось тепер до причин появи внутрішніх завод. Як відмічалось вище, основна причина їх наявності – це невідповідність елементів та вузлів електронних схем, які використовуються при проектуванні, їх реальним характеристикам. Тому можна сказати, що причинами появи внутрішніх завод виступають:

- неідеальність активних і реактивних компонентів схеми, що проявляється на високих частотах;
- наявність внутрішнього опору джерел живлення, що приводить до коливань вихідної напруги при зміні величини навантаження;
- наявність опору шин живлення що приводить до падіння напруги на них і може сприйматись окремими елементами як інформаційний сигнал;
- наявність ємнісних зв'язків між шинами живлення та інформаційними входами.

Відмічена наявність невраховуваних при проектуванні активних та реактивних складових поєднується з наступною специфікою цифрових систем:

- значні струми живлення, що досягають десятків амперів при низьких

робочих напругах ключових елементів (наприклад, сучасні процесори при напрузі живлення 1,6 В споживають струм до 15 А);

- значною динамікою навантаження цифрових схем та пристроїв, яке обумовлене як швидкою зміною підключення та відключення окремих вузлів та блоків системи, так і значними імпульсними струмами, які споживаються окремими ключами з врахуванням їх синхронної роботи.

Перша специфіка цифрових систем приводить до того, що потенціал загальної шини на різних ділянках відрізняється на величину падіння напруги  $\Delta U_3$  між ними, яка пропорційна величині струму та опору загальної шини. Якщо структура шин побудована без урахування цієї специфіки, то величина  $\Delta U_3$  може досягати сотень мілівольт.

Друга специфіка проявляється через наявні індуктивні опори шин живлення.

На рис. 1.29, а проводиться еквівалентна схема живлення цифрових схем з окремими вузлами, які представлені еквівалентними опорами  $R_{bi}$  та ключами  $K_{bi}$ . Підключення-відключення окремих вузлів еквівалентні появі імпульсних струмів  $I_{bi}$ , які призводять до появи динамічних завад  $e_{pi}$ , котрі визначаються індуктивностями ділянок шини:

$$e_{pi} = L_{ши} \frac{I_{bi}}{t_{фр}}$$

де  $t_{фр}$  – тривалість фронту імпульсів.

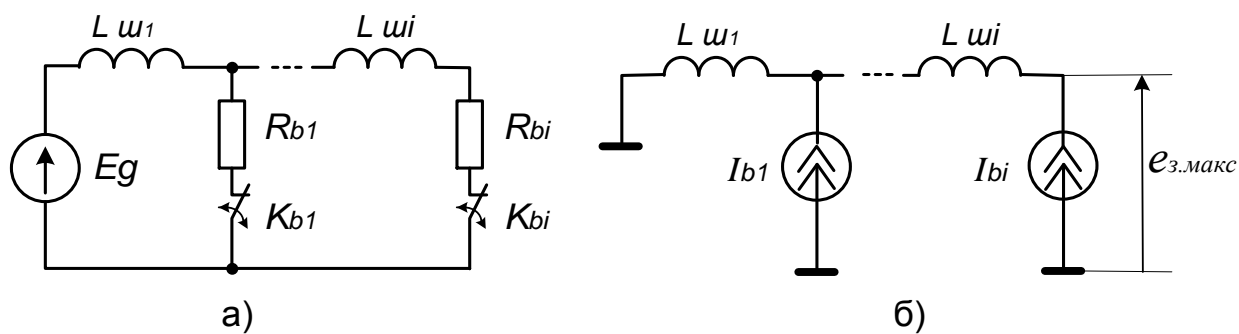


Рис. 1.29

На рис. 1.29, б приводиться схема заміщення для розрахунку величини  $e_{pi}$ . Максимальна величина завади буде при синхронній комутації всіх вузлів:

$$e_{п.мак} = L_{ш i} \frac{I_{bi}}{t_{фр}} i (i - 1).$$

Розглянемо один з прикладів появи завад такого типу – завади, обумовлені затримками у перемиканні логічних елементів.

При розгляді логічних функцій вважалось, що логічні елементи перемикаються миттєво, не маючи часових затримок, і стан виходу логічного елемента оцінювався в статичному режимі, тобто, коли вхідні сигнали не змінюються. Але реальні логічні елементи мають часові затримки при перемиканні, величини яких залежать від різних факторів, тому і значення логічної функції при зміні значень її аргументів також може бути непередбаченим.

Оскільки логічні елементи мають реальні часові затримки при перемиканні, то під час перемикання кола передачі інформаційних сигналів можуть відрізнитись від тих, що мали місце в статичних режимах. Це може приводити до непередбачених ситуацій, тому логічні схеми слід розглядати і в динамічних режимах перемикання. Однією з особливостей поведінки логічних схем у динаміці є поява короткочасних імпульсних завад. Тому проєктантам електронних цифрових систем необхідно передбачати можливість появи таких ситуацій і забезпечувати проєктування за умови неможливості появи імпульсних завад, навіть у найгірших випадках.

Будь-яка комбінаційна схема повинна аналізуватись на можливість появи завад. Але добре спроектовані синхронні цифрові системи здебільшого структуровані так, що забезпечують неможливість появи подібних завад. У синхронних системах всі входи комбінаційних кіл змінюються одночасно, а сигнали на виходах з'являються лише після установаження перехідних процесів. Аналіз на можливість появи завад необхідно проводити лише для

	$x_1 x_0$	00	01	11	10
$a_0$	0	<sup>0</sup> 0	<sup>1</sup> 0	<sup>3</sup> 1	<sup>2</sup> 1
	1	<sup>4</sup> 0	<sup>5</sup> 1	<sup>7</sup> 1	<sup>6</sup> 0

Рис. 1.31

асинхронних цифрових схем, а також для схем зі зворотніми зв'язками, які розглядаються, починаючи з **Розділу 4**.

Статичні завади можуть бути двох типів – поява короткочасного імпульсу нульового рівня

при наявності сигналу одиничного рівня на виході і навпаки – поява короткочасного імпульсу одиничного рівня при наявності сигналу нульового рівня на виході.

Прикладом створення завади першого типу є схема, що приведена на рис. 1.30, а. Це схема мультиплексора, що забезпечує передачу на вихід значення одного з інформаційних входів  $x_0$ ,  $x_1$ , який вибирається адресним сигналом  $a_0$ .

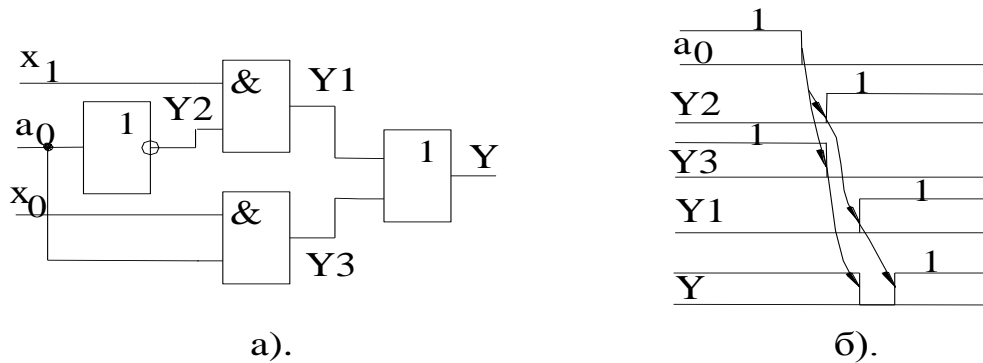


Рис. 1.30

Допустимо, що сигнали  $x_0$  і  $x_1$  мають одиничні значення, а адресний сигнал  $a_0$  змінюється з “1” в “0” (рис. 1.30, б). Оскільки зміна сигналів  $a_0$  і  $Y_2$  відбуваються в різні інтервали часу, то нескладний аналіз часових діаграм і роботи схеми показує, що між моментом зникнення одиниці  $x_1$  на виході  $Y$  і моментом появи там одиниці  $x_0$  має місце короткочасна пауза, яка є імпульсною завадою нульового рівня.

Імпульсні завади одиничного рівня можуть з'явитися на виході схеми у тих випадках, коли має місце перемикання одного сигналу низького рівня на інший, які подаються на вихід схеми через логічні комутатори.

Визначити можливість появи статичних завад у проектованій логічній схемі можливо за допомогою карт Карно. Проаналізуємо карту Карно для розглянутої схеми (рис. 1.31).

Зробивши об'єднання клітинок 2, 3 і 5, 7, отримуємо логічну функцію:

$$Y = \overline{a_0} x_1 + a_0 x_0.$$

Але, у той же час, з такого об'єднання можна зробити висновок, що неоднчасна зміна сигналів, які забезпечують логічні одиниці в клітинах 3 і 7, призведе до появи завади. Звідси також витікає і практичний варіант усунення вказаного недоліку схеми, що розглядається. Він полягає у тому, що необхідно допоміжно об'єднати клітини 3 і 7. При цьому логічна функція прийме вигляд:

$$Y = \overline{a_0} x_1 + a_0 x_0 + x_1 x_0,$$

а на схемі з'явиться допоміжний логічний елемент **2І**, який виключить можливість появи завади.

Паразитні індуктивності мають місце не тільки в шинах живлення, а й у інформаційних провідниках, між інформаційними провідниками, а також між інформаційними провідниками й шинами живлення. Їх наявність призводить до спотворення інформаційних сигналів, котрі, як і завади, можуть привести до порушення нормальної роботи цифрових пристроїв.

У залежності від характеру спотворення сигналів, розділяють завади в коротких та довгих лініях. У першому випадку взаємодія сигналів з реактивностями може бути описана еквівалентними схемами з зосередженими параметрами, у другому основою завад стають хвильові процеси, які мають місце в довгих лініях.

Умовою розділення інформаційних ліній на довгі та короткі виступає параметр

$$\varepsilon = \frac{\tau l}{t_\phi},$$

де  $\varepsilon$  – погонна затримка лінії;  $l$  – довжина лінії;  $t_\phi$  – тривалість фронту імпульсу.

При  $\varepsilon < 1$  виконуються умови коротких ліній; при  $\varepsilon > 1$  проходження інформаційного сигналу розглядається з позицій довгих ліній.

При розгляді лінії зв'язку як короткої лінії вона у загальному випадку представляється еквівалентною індуктивно-ємнісною схемою  $L_{\text{ел}}$ ,  $C_{\text{ел}}$ , на вході якої діє імпульс інформаційного сигналу  $e_{\text{вх}}$  (рис. 1.32) з вихідним опором логічного елемента  $R_{\text{вих}}$  та опором  $R_{\text{ел}}$ , що представляє собою еквівалентне навантаження.

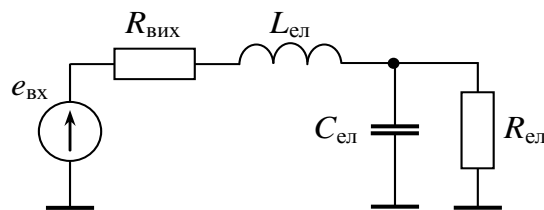


Рис. 1.32

Враховуючи, що детальний аналіз подібних схем детально вивчається в теорії електричних кіл, відмітимо лише дві властивості, які доводиться враховувати проєктантам: можливість резонансних коливань, а також поява непередбачених затримок у лінії. Перша приводить до непередбачених затримок у лінії зв'язку, а друга – до пропусення синхронізації всього пристрою.

Наявність ємності між інформаційними провідниками приводить до появи взаємного впливу між ними, що відображається наявністю *перехресних завод*. Такі завади виникають між двома провідниками інформаційного джгута або кабелю, між двома сусідніми провідниками друкованих плат, а також між силовими провідниками живлення і інформаційним провідником, якщо вони розміщені поряд.

*Перехресні завади* – це такий вид завод, що обумовлюється взаємним впливом двох паралельно розміщених провідників. Характер впливу залежить від співвідношення між довжиною ліній та тривалістю перехідних процесів у них, а його кількісні характеристики можуть бути визначені як за допомогою частотних, так і часових співвідношень.

У зарубіжній літературі завади, які обумовлені наявністю паразитних ємностей та індуктивностей, а також наведені на провідниках електромагнітним полем, розподіляються на завади *нормального (normal voltage)* та *загального виду (common-mode voltage)*: завади нормального виду діють між двома інформаційними провідниками, а завади загального виду діють на входи схем відносно “землі” або інших загальних точок. Незалежно від причини появи того чи іншого виду завад, таке їх розділення формує основні напрямки боротьби з завадами.

Нагадаємо, що завади у довгих лініях обумовлені появою зворотніх хвиль, які виникають з причини неузгодженості хвильових опорів каналу зв'язку та інформаційної лінії. У зв'язку з високими робочими частотами сучасних цифрових схем хвильові процеси в них є характерними і ряд засобів боротьби з ними закладається на етапі проектування.

*Завадозахищеністю (завадостійкістю)* електронного пристрою називають його здібність працювати з заданими показниками безвідмовності, незважаючи на наявність зовнішніх завад з заданими значеннями параметрів. Для цифрових інтегральних схем вона визначає допустиму напругу завад на входах мікросхеми, яка не призводить до зміни її стану.

У залежності від тривалості дії завад, розділяють *статичну* і *динамічну* завадостійкість. *Статична* характеризує стійкість цифрових мікросхем до завад, тривалість яких перевищує тривалість перехідних процесів зміни станів. *Динамічну* пов'язують з завадами, тривалість яких близька або менше тривалості перехідних процесів.

## КОНТРОЛЬНІ ПИТАННЯ

1. Дайте визначення терміну “сигнал”. Чи є напруга, що живить Ваш ПК, сигналом. Чому?
2. Дайте визначення терміну “аналоговий сигнал”. Якими параметрами характеризується моногармонічний аналоговий сигнал?

3. Дайте визначення терміну “модуляція”.

4. Які види модуляції моногармонічного сигналу Вам відомі?

Дайте пояснення кожному з видів модуляції, наведіть приклади використання.

5. У чому полягають недоліки аналогових сигналів?

6. Дайте пояснення параметрам імпульсу та імпульсної послідовності.

7. Дайте визначення функції Хевісайда; функції Дірака. У чому полягають особливості цих функцій?

8. Дайте пояснення імпульсним видам модуляції. У чому полягають переваги імпульсної форми передачі сигналу?

9. Поясніть на конкретних прикладах фізичну суть теореми академіка Котельникова (імпульсна теорема).

10. Яким шляхом здійснюється квантування у часі та за рівнем? Дайте пояснення.

11. Дайте визначення термінам “біт”, “байт”.

12. Дайте визначення системи числення. Наведіть приклади позиційних і непозиційних систем числення. Дайте пояснення.

13. Що Ви розумієте під терміном “алфавіт системи числення”? Наведіть приклади алфавітів двійкової системи числення; шістнадцяткової системи числення; системи числення з основою 20.

14. Скільки і яких цифр потрібно, щоб будь-яке число можна було записати у сімковій системі числення? А в дванадцятковій?

15. Поясніть особливості використання схеми Горнера для переведення чисел з  $P$ -ічної системи числення в десяткову.

16. Дайте пояснення особливостям переведення чисел з десяткової системи числення у  $P$ -ічну.

17. Дайте пояснення особливостям переведення дробових чисел з однієї системи числення в іншу. Наведіть конкретні приклади.

18. Дайте визначення термінам “код”, “кодування”, “декодування”.

19. Що є необхідною умовою для декодування повідомлення?



20. Які принципи розділення кодових символів при їх передачі Вам відомі? Дайте пояснення кожному з приведених принципів.

21. Поясніть взаємозв'язок між прямим двійковим, оберненим і доповняльним кодами.

22. Поясніть особливості коду Грея та області його використання.

23. Наведіть приклади самодоповнюючих кодів. Поясніть їх властивості.

24. Які коди називаються заводозахищеними (коригуючими)? У чому полягає їх властивість?

25. Поясніть переваги та недоліки послідовного і паралельного форматів передачі даних.

26. Дайте пояснення терміну “*протокол обміну даних*”.

27. Які способи послідовної передачі даних Вам відомі? Поясніть їх особливості.

28. Поясніть особливість представлення від'ємних чисел.

29. Поясніть суть запису двійкових чисел у формі з фіксованою комою. У чому полягають недоліки такого способу запису?

30. Поясніть особливість запису чисел у формі з плаваючою комою.

31. Поясніть послідовність виконання арифметичних операцій додавання і віднімання в двійковій системі числення при різних знаках зменшуваного і від'ємника.

32. Поясніть особливість виконання арифметичних операцій у двійково-десятковому коді.

33. Поясніть послідовність виконання операції множення у двійковому коді.

34. Операція віднімання у двійковому коді виконується з використанням доповнюючого коду. Проаналізуйте і дайте пояснення, чи справедлива подібна послідовність виконання арифметичної операції, якщо числа задаються у двійково-десятковому коді; у коді “з надлишком 3”.

35. Дайте визначення терміну “*алгебра логіки*” (“*булева алгебра*”).

36. У чому полягає суть принципу подвійності?
37. Дайте пояснення диз'юнктивній і кон'юнктивній формам запису логічних функцій, а також досконалим формам запису.
38. Перелічіть відомі Вам способи запису логічних функцій. Дайте пояснення взаємозв'язку між ними.
39. Яка функція називається *неповністю визначеною*? Як забезпечити перехід для неповністю визначеної функції від табличного способу запису до аналітичного?
40. Сформулюйте теорему Шеннона. Дайте пояснення її використання при перетворенні булевих виразів.
41. Дайте пояснення суті розкладання Ріда. Поясніть його призначення та специфіку використання.
42. Дайте пояснення суті теореми де Моргана. Приведіть приклади її використання.
43. Поясніть властивості карти Карно.
44. Які логічні операції використовуються при аналітичному способі мінімізації логічних функцій?
45. На яких властивостях карт Карно реалізується задача мінімізації логічних функцій?
46. На якій концепції базується мінімізація з використанням кубічних комплексів?
47. Поясніть особливість мінімізації неповністю визначених функцій.
48. У чому полягає особливість сумісної мінімізації логічних функцій?
49. Дайте інтерпретацію імпліканти і покриття Квайна.
50. Поясніть суть мінімізації логічних функцій методом Квайна і Мак-Класкі.
51. Дайте визначення відстані між двома  $n$ -бітними словами.
52. У чому полягає ідеологія побудови кодів з виявленням помилок?
53. Як будуються коди з виявленням і виправленням помилок?

54. Поясніть особливість побудови коду Хемінга.
55. Яку кодову відстань забезпечує код Хемінга?
56. Поясніть суть двовимірних кодів. Де вони використовуються?
57. Дайте пояснення принципу створення циклічних кодів. Яким шляхом виконується декодування циклічних кодових комбінацій?
58. Дайте визначення термінам “завада”, “зовнішня (внутрішня) завада”, “наводка”.
59. Що виступає основним джерелом завад в електронному пристрої?
60. Перелічіть причини внутрішніх завад електронного пристрою.
61. У чому проявляється специфіка цифрових схем з точки зору їх завадостійкості?
62. Поясніть причину появи імпульсних завад при перемиканні логічних елементів. У чому полягає проблема боротьби з такими завадами?
63. У чому полягає різниця при розгляді завад у коротких і довгих лініях?
64. Дайте інтерпретацію *перехресних завад*.
65. Поясніть різницю між завадами “нормального” та “загального” видів.
66. Дайте визначення терміну “завадостійкість”. У чому полягає різниця між статичною і динамічною завадостійкістю?

## ВПРАВИ І ЗАВДАННЯ

1. Аналоговий сигнал, що змінюється в інтервалі напруг  $0 \dots 15$  В описується чотирьохбітним цифровим словом  $B = b_3 b_2 b_1 b_0$ . Пояснити, скільки рівнів квантування використовується при дискретизації аналогового сигналу. Визначити величину шуму квантування. Визначити кількість інформації, що міститься в одному слові.

2. Аналогові телефонні мережі мають смугу пропускання, що обмежується верхньою частотою 4 кГц. Визначити мінімальну частоту дискретизації аналогового сигналу при переході на цифрові лінії зв'язку.

3. Як виглядатиме у двійковій системі числення десяткове число  $0,125_{10}$  ?

4. Запишіть у системі числення з основою 240 числа 241, 243, 244, 251.

5. Підрахуйте кількість двійкових чисел у діапазоні від  $10_2$  до  $1000_2$  .

6. У паперах дивака-математика була знайдена його автобіографія.

Вона починалася наступними дивними словами: «Я закінчив курс університету 44 років від роду. Через рік, 100-літньою молодою людиною, я одружився на 34-літній дівчині. Незначна різниця у віці всього 11 років сприяла тому, що ми жили спільними інтересами і мріями. Через небагато років у мене вже була маленька сім'я з 10 дітей. Платню я отримував за місяць всього 200 рублів, з яких  $1/10$  доводилося віддавати сестрі, так що ми з дітьми жили на 130 рублів на місяць». Чим пояснити дивні протиріччя у числах приведеної автобіографія?

*Підказка:* “Через рік, 100-літньою молодою людиною...”.

7. Випишіть всі цифри системи числення з основою 25.

8. Випишіть базис п'ятіркової системи числення.

9. Яку мінімальну основу може мати система числення, якщо в ній записані всі наступні числа: 432, 120, 111, 2331?

10. Вкажіть, які числа записані з помилками, відповідь поясніть:  $123_7$ ;  $3005_4$ ;  $12AAC9_{20}$ ;  $13475_7$  .

11. Запишіть у 6-й системі числення число, наступне по порядку за числом 5.

12. Яке число слідує за числом  $111_4$  у 4-ній системі числення?

13. Яке число передує числу  $10_8$  у 8-ній системі числення?

14. Запишіть у розгорненому вигляді числа:  $65_7$ ;  $1998_{10}$ ;  $0,15A_{16}$ ;  $1AF1H, A9_{20}$  .

15. Яке з чисел більше:  $5_{10}$  або  $5_8$ ;  $1111_2$  або  $1111_8$ ?

16. Перетворити у десяткову систему числення числа, записані у двійковій:  $1_2$ ;  $101_2$ ;  $10000_2$ ;  $11011101_2$  .

17. Перетворити числа 118, 57, 196, 238, що представлені в десятковій системі числення, у двійкову.

18. Перетворити числа десяткової системи числення  $255_{10}$ ,  $3959_{10}$  у шістнадцяткову.

19. Перетворити числа десяткової системи числення  $0,27_{10}$  і  $0,625_{10}$  у двійкову з чотирма знаками після коми.

20. Перетворити числа  $0,31_{10}$  і  $0,761_{10}$  у шістнадцяткову систему числення з трьома знаками після коми.

21. Перетворити число  $25,179_{10}$  у двійкову систему числення з трьома знаками після коми.

22. Перетворити десяткове число  $69,42_{10}$  у шістнадцятирічну систему числення.

23. Перетворити двійкове число  $1011,01_2$  у десяткову систему числення.

24. Перетворити число  $3F,9_{16}$  у десяткову систему числення.

25. Перетворити двійкове число  $11110101,1011_2$  у шістнадцяткову систему числення.

*Пояснення:* Переведення чисел двійкової системи числення до шістнадцяткової досягається шляхом представлення кожної двійкової тетради (чотирьох двійкових цифр) однією цифрою шістнадцяткової системи. Тому:

$$\underbrace{1111}_F \underbrace{0101}_5, \underbrace{1011}_B = F5, B_{16}.$$

26. Перетворити число  $C7,9_{16}$  у двійкову систему числення.

*Пояснення:* Переведення у двійкову систему числення забезпечується шляхом заміни кожного числа шістнадцяткової системи його двійковим еквівалентом. Відповідь:  $C7,9_{16} = 1100\ 0111, 1001_2$ .

27. Знайти основи  $P$  і  $Q$  систем числення, для яких була б справедлива нерівність:  $12_P > 21_Q$ .

Відповідь: Умовою існування нерівності є:  $P > 2Q - 1$ . Доведіть.

28. Розробити алгоритм і написати програму для переведення чисел з десяткової системи числення у двійкову (будь-яка мова програмування).

29. Розробити алгоритм і написати програму для переведення чисел з двійкової системи числення у десяткову.

30. Задані числа у шістнадцятковому коді:  $3F_{16}$ ,  $9C_{16}$ ,  $A6_{16}$ ,  $8_{16}$ ,  $2D_{16}$ ,  $7_{16}$ . Знайти їх двійкові еквіваленти, перевести у зворотній код, а останній перевести у шістнадцятковий.

31. Знайти доповняльні коди чисел  $2A_{16}$ ,  $CD_{16}$ ,  $9B_{16}$ ,  $F3_{16}$ .

32. Допустимо, що чотирьохбітне число  $N$  представлене  $n$  цифрами шістнадцяткової системи числення. Довести, що зворотній код числа  $N$  представлений його шістнадцятковим доповненням.

33. Використовуючи дані табл. 1.2, встановити залежність між двійковим кодом і кодом Айкана.

34. Восьмибітним двійковим кодом задаються числа, які відповідають десятковим числам у діапазоні від 0 до 99. Встановити залежність між двійковим і двійково-десятковим представленням цих чисел.

35. Розробити алгоритм, який забезпечував би перетворення двійкового коду у двійково-десятковий, і написати програму на будь-якій мові програмування.

36. Трьохрозрядний декодуючий диск призначений для кодування кута повороту обертання вала у двійковий код. Визначити всі позиції некоректних переходів з одного кодового слова до іншого.

37. Визначити кількість некоректних переходів функції  $n$  механічного декодуючого диску, що використовує  $n$ -бітний двійковий код.

38. Знайти доповняльні коди таких десяткових чисел: +5; -120; +38; -56.

39. Обчислити діапазон чисел, що зображуються у формі з плаваючою комою, якщо кількість розрядів мантиси –12, а порядку –4.

40. Зобразити в нормальній формі такі числа, представлені у десятковій системі числення: 438; -567; 0,00435; 831,00345.

41. Знайти суму наступних пар чисел, заданих у десятковій системі числення: 38 і 43; 24 і 112; 56 і 235, перевірши їх спершу в двійкову систему.

42. Знайти суму наступних пар чисел, заданих у шістнадцятковій системі числення:  $+23h$  і  $-16h$ ;  $-37h$  і  $-45h$ ;  $184h$  і  $-231h$ .

43. Знайти суму чисел, заданих у вісімковій системі числення, перевівши їх у двійково-десятковий код:  $12_8$  і  $34_8$ ;  $11_8$  і  $57_8$ ;  $21_8$  і  $77_8$ ;  $55_8$  і  $41_8$ .

44. Приклади з вправи 43 розв'язати шляхом переведу чисел у код “з надлишком 3”.

45. Виконати операцію множення двох чисел  $a = 1101_2$  та  $b = 1011_2$ .

46. Розробити алгоритм виконання операції множення двох двійкових чисел та реалізувати його на одній з мов програмування.

47. Виконати операцію ділення числа  $a = 1101110_2$  на  $b = 1010_2$ .

48. Розробити алгоритм виконання операції ділення двох двійкових чисел та реалізувати його на одній з мов програмування.

49. Скласти таблиці істинності для двохвходових елементів **АБО**, **І**, **ВИКЛ. АБО**.

50. Побудувати часові імпульсні діаграми, які ілюструють роботу двохвходових логічних елементів **АБО** та **ВИКЛ. АБО**; **І** та **ВИКЛ. І**.

51. Логічні функції трьох змінних задані таблицями істинності (табл.1.11). Представити їх у аналітичній формі.

Таблиця 1.11

$x_2$	$x_1$	$x_0$	$y_0$	$y_1$	$y_2$
0	0	0	0	1	1
0	0	1	0	1	1
0	1	0	1	0	0
0	1	1	1	0	0
1	0	0	0	1	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	0	0	1

52. Реалізувати логічну схему з непарною кількістю входів  $m$ , яка дозволяє вирішити задачу голосування простою більшістю Розв'язати задачу для  $m = 3; 5$ .

53. При розробці електронного календаря виникає необхідність кодування днів тижня. Пропонується довільно закодувати дні тижні та розробити логічну схему їх декодування.

54. Побудувати таблиці істинності для наступних функцій трьох змінних:  
 1) сума за модулем два; 2) нерівність всіх аргументів один одному, тобто вони не співпадають; 3) **ВИКЛ. АБО** (**АБО**-альтернатива, тобто один і тільки один для всіх); 4) диз'юнкція.

55. Довести тотожності:

$$\begin{aligned} (x_1 + x_3) \cdot x_2 &= x_1 x_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + x_1 x_2 x_3 ; \\ x_1 x_3 + x_2 \bar{x}_3 + x_1 x_2 &= x_1 x_3 + x_2 \bar{x}_3 ; \\ x_2 x_3 + x_1 x_4 &= (x_1 + x_2) \cdot (x_2 + x_4) \cdot (x_1 + x_3) \cdot (x_3 + x_4) ; \\ x_1 x_2 + x_1 x_2 x_3 + x_1 \bar{x}_2 &= x_1 ; \\ \bar{x}_1 x_2 + x_1 \bar{x}_2 &= (\bar{x}_1 + \bar{x}_2) \cdot (x_1 + x_2) . \end{aligned}$$

56. Знайти заперечення для кожного з наступних виразів. Результати не спрощувати.

$$\begin{aligned} y_1 &= x_1 + \bar{x}_2 x_3 ; \\ y_2 &= x_1 \cdot (x_2 + x_3) + x_2 \bar{x}_4 \cdot (\bar{x}_1 + x_3) ; \\ y_3 &= [x_1 \bar{x}_2 + x_3 \cdot (\bar{x}_1 + x_4 x_5)] \cdot [\bar{x}_2 + x_1 x_3 \cdot (\bar{x}_4 + \bar{x}_2 \bar{x}_3)] ; \\ y_4 &= \overline{x_1 x_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + (x_1 + x_2 + x_3) \cdot (\bar{x}_1 + \bar{x}_2 + x_3)} ; \\ y_5 &= \overline{(x_1 + x_2 \bar{x}_3) \cdot (\bar{x}_2 x_3 + \bar{x}_1) + (x_1 x_2 \bar{x}_3 + \bar{x}_2)} . \end{aligned}$$

57. Скласти таблицю істинності для наступних виразів:

$$\begin{aligned} y_1 &= x_1 + x_2 \bar{x}_3 ; \\ y_2 &= x_2 x_4 + \bar{x}_1 x_3 + \bar{x}_2 \bar{x}_4 ; \\ y_3 &= x_1 x_3 + x_2 x_3 + x_1 x_2 ; \\ y_4 &= x_2 x_4 + \bar{x}_1 x_3 + \bar{x}_2 \bar{x}_4 . \end{aligned}$$

58. Використовуючи теорему де Моргана, перетворити наступні вирази:

$$\begin{aligned} y_1 &= \overline{x_1 + x_2 + x_1 x_2 x_3 + \bar{x}_1 x_3} ; \\ y_2 &= \overline{(x_1 x_2 + \bar{x}_2 x_3) + (x_2 \bar{x}_3 + \bar{x}_1 x_2)} ; \\ y_3 &= \overline{(x_1 x_2 + \bar{x}_2 x_3) \cdot (x_2 x_3 + \bar{x}_1 \bar{x}_3)} ; \\ y_4 &= \overline{(x_1 + x_2) + (x_1 + x_2) \cdot (\bar{x}_3 + \bar{x}_2) + (x_3 + \bar{x}_2)} ; \\ y_5 &= \overline{(x_1 + \bar{x}_2 x_3 + \bar{x}_1 \bar{x}_2) + (x_1 + x_2 \bar{x}_3 \bar{x}_1)} ; \\ y_6 &= \overline{x_1 + \bar{x}_2 \bar{x}_3 + \bar{x}_1 x_2 x_3 + \bar{x}_2 x_3} . \end{aligned}$$



59. Функції з вправи 56 привести до канонічної суми.  
 60. Функції з вправи 56 представити у табличній формі запису.  
 61. Для функцій, що приведені у вправі 55, привести електричні контактні схеми.

62. Функції з вправи 55 представити у базисі елементів **I, АБО, НІ**.

63. Використовуючи теореми алгебри логіки, спростити вирази:

$$y_1 = x_3 x_2 x_1 x_0 (x_3 \bar{x}_2 x_1 x_0 + x_3 x_2 x_1 \bar{x}_0 + x_3 x_2 x_1 x_0 + x_3 x_2 \bar{x}_1 x_0);$$

$$y_2 = x_1 x_0 + x_3 \bar{x}_2 x_1 x_0 + x_4 x_3 x_1 x_0 + x_4 \bar{x}_2 x_1 x_0 + x_4 x_3 \bar{x}_2.$$

64. Використовуючи розкладання Ріда і Шенона, довести тотожності:

$$x \oplus x = 0; x \oplus \bar{x} = 1; x \oplus 0 = x; x \oplus 1 = \bar{x}.$$

65. Довести тотожності:

$$\overline{x_1 \oplus x_2 \oplus x_3} = x_1 \oplus x_2 \otimes x_3;$$

$$\overline{x_1 \otimes x_2 \otimes x_3} = x_1 \otimes x_2 \oplus x_3;$$

$$(x_1 \oplus x_2 \oplus x_1 x_2) \cdot (x_1 \oplus x_3 \oplus x_1 x_3) = x_1 + x_2 x_3;$$

$$x_1 \oplus x_2 \oplus x_1 \cdot x_2 = x_1 + x_2;$$

$$x_1 \oplus x_2 \oplus 1 = x_1 \otimes x_2.$$

66. Привести до ДДНФ логічні функції:

$$y_1 = \bar{x}_2 x_0 + x_2 x_1 + x_2 \bar{x}_1 x_0;$$

$$y_2 = x_3 \bar{x}_1 x_0 + x_2 x_1 x_0 + x_3 \bar{x}_1 x_0 + x_2 x_1 x_0;$$

$$y_3 = x_2 x_1 x_0 + x_3 \bar{x}_2 x_1 x_0 + x_2 x_1;$$

$$y_4 = (x_2 + x_3) \cdot (x_0 + x_2 + x_3) \cdot (x_0 + x_2 + x_1);$$

$$y_5 = x_3 x_2 x_1 x_0 + x_3 x_2 \bar{x}_1 + x_3 x_2 x_0 + x_3 \bar{x}_1 x_0 + x_2 x_1 \bar{x}_0.$$

67. Представити за допомогою карт Карно наступні функції чотирьох змінних:  $y_1 = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4; y_2 = x_1 x_3 x_4; y_3 = \bar{x}_2 x_4; y_4 = \bar{x}_2.$

Представити у формі карт Карно наступні функції:

$$y_1 = f(x_1, x_2, x_3, x_4) = \vee 0, 1, 2, 3, 4, 6, 7, 8, 12, 13;$$

$$y_2 = f(x_1, x_2, x_3) = \vee 0, 1, 2, 3, 7;$$

$$y_3 = f(x_1, x_2, x_3) = \vee 3, 5, 6, 7.$$

68. Представити за допомогою карт Карно наступні функції трьох

змінних:  $y_1 = x_1 x_2 x_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3$ ;  $y_2 = x_2 x_3$ ;  $y_3 = \bar{x}_3$ .

69. Функції з табл. 1.11 представити у формі карт Карно.

70. Функції, що представлені на рис. 1.33 у формі карт Карно, зобразити в табличній і аналітичній формах запису.

	$X_1 X_0$	00	01	11	10
$X_3 X_2$	00	0	1	3	2
01	4	0	1	7	6
11	12	0	1	15	14
10	8	1	0	11	10

а)

	$X_1 X_0$	00	01	11	10
$X_3 X_2$	00	0	1	3	2
01	4	1	0	7	6
11	12	1	0	15	14
10	8	0	1	11	10

б)

	$X_1 X_0$	00	01	11	10
$X_3 X_2$	00	0	1	3	2
01	4	0	1	7	6
11	12	1	0	15	14
10	8	1	0	11	10

в)

Рис.1.33

71. Записати за допомогою Карно наступні логічні функції, задані у канонічній диз'юнктивній формі:

$$y_1(x_4, x_3, x_2, x_1, x_0) = \vee 5, 7, 13, 15, 16, 20, 25, 27, 29, 31;$$

$$y_2(x_4, x_3, x_2, x_1, x_0) = \vee 0, 7 - 9, 12, 13 - 15, 16, 22, 23, 30, 31;$$

$$y_3(x_4, x_3, x_2, x_1, x_0) = \vee 0, 1, 2, 4, 5, 10, 11, 14, 20, 21, 24 - 30;$$

$$y_4(x_4, x_3, x_2, x_1, x_0) = \vee 0, 2, 4, 6 - 8, 10 - 14, 16, 18, 19, 29, 30;$$

$$y_5(x_4, x_3, x_2, x_1, x_0) = \vee 4, 5, 10, 12, 13, 16, 17, 21, 25 - 27, 29;$$

$$y_6(x_4, x_3, x_2, x_1, x_0) = \vee 4, 6, 7, 9, 11 - 15, 20, 22, 25, 27, 28, 30.$$

72. Функції з вправи 73 мінімізувати з використанням карт Карно.

73. Записати за допомогою Карно наступні логічні функції, задані у канонічній диз'юнктивній формі:

$$y_1(x_5, x_4, x_3, x_2, x_1, x_0) = \vee 1, 5, 9, 13, 21, 23, 29, 31, 37, 45, 53, 61;$$

$$y_2(x_5, x_4, x_3, x_2, x_1, x_0) = \vee 0, 4, 8, 16, 24, 32, 34, 36, 37, 39, 40, 48, 50, 56;$$

$$y_5(x_5, x_4, x_3, x_2, x_1, x_0) = \vee 2, 4, 5, 6, 12 - 21, 28 - 31, 34, 38, 50, 51, 60 - 63.$$

74. Функції з вправи 74 мінімізувати з використанням карт Карно.

75. Використовуючи теорему Шеннона, виконати перетворення

наступних логічних функцій:

$$y_1 = x_0 + (x_0 \oplus x_1 \oplus x_2); y_2 = x_0 x_1 (x_1 x_2 \oplus \overline{x_2} \oplus \overline{x_0});$$

$$y_3 = (x_0 + x_1) \oplus ((x_2 + x_3) \oplus x_1 \oplus (\overline{x_0} \cdot x_2)).$$

76. Використовуючи тотожності (1.18) і (1.19), спростити функцію:

$$f(x_2, x_1, x_0) = \overline{\overline{\overline{(x_0 + x_1)} \oplus (x_2 + x_1)} \oplus x_1 + x_1 x_2 x_3}.$$

77. Використовуючи розкладання Ріда, зобразити функцію

$$f(x_0 x_1 d_0 d_1 d_2 d_3) = \overline{x_1} \overline{x_0} d_0 + \overline{x_1} x_0 d_1 + x_1 \overline{x_0} d_2 + x_1 x_0 d_3$$

у вигляді полінома.

78. Функції з вправи 58 мінімізувати, використовуючи операції поглинання та склеювання.

79. Мінімізувати логічну функцію за допомогою карти Карно:

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

80. Функції з вправи 56 привести до досконалої диз'юнктивної форми запису.

81. Представити у мінімальній диз'юнктивній формі наступні функції:

$$y_1 = x_1 \cdot (x_1 \oplus x_2 \oplus x_3); y_2 = x_1 \cdot (x_1 \otimes x_2 \otimes x_3);$$

$$y_3 = x_1 + (x_1 \oplus x_2 \oplus x_3); y_4 = x_1 + (x_1 \otimes x_2 \otimes x_3).$$

82. Для логічної функції, яка задана картою Карно (рис. 1.34, 1.35) значення ряду клітин, позначених значком "×", є невизначеними. Довизначити функцію з метою отримання мінімальної диз'юнктивної форми і мінімізувати. Проаналізувати різні шляхи довизначення функції та її мінімізації.

		$x_1 x_0$			
		00	01	11	10
$x_3 x_2$	00	0 1	1 1	3 0	2 ×
	01	4 ×	5 ×	7 0	6 ×
	11	12 1	13 1	15 ×	14 1
	10	8 1	9 1	11 ×	10 1

Рис. 1.34

		$X_1X_0$			
		00	01	11	10
$X_3X_2$	00	0 1	1 1	3 0	2 ×
	01	4 ×	5 ×	7 0	6 ×
	11	12 1	13 1	15 ×	14 1
	10	8 1	9 1	11 ×	10 1

Рис. 1.35

83. Виконати сумісну мінімізацію системи логічних функцій, заданих у вправі 68.

84. Для логічних функцій  $y_2, y_3$  із вправи 68 записати кубічні комплекси.

85. Для кубічного комплексу з попередньої задачі знайти покриття логічної функції.

86. Визначити відстань між двома вершинами  $n$ -вимірного кубу  $A_1(1101011)$  та  $A_2(1011101)$ .

87. Знайти шлях намалювання 3-кубу на аркуші паперу так, щоб ніякі лінії не перетинались, або довести, що це неможливо.

88. Повторити вправу 89 для 4-кубу.

89. Написати формулу, що дає кількість  $m$  субкубів  $n$ -вимірного кубу для заданої кількості  $m$ .

90. Маємо шину передачі даних шириною  $n$  біт. Яка розрядність двійкового коду повинна бути, щоб забезпечити відстань  $d = 2$ ? Яку кількість кодових слів може мати відповідний код?

91. Для трьохрозрядного двійкового коду створити відповідний чотирьохрозрядний код парності і код непарності.

92. Знайти відстань між двома кодовими словами двійково-п'ятіркового коду.

93. Із трьохбітного двійкового коду вибрати слова, між якими забезпечується відстань  $d = 2$ . Виконати таку ж операцію для чотирьохбітного двійкового коду.

94. Визначити наявність відстані для двійкового трьохрозрядного коду. Вибрати відповідні кодові слова.

95. Яка відстань  $d$  забезпечується в  $n$ -бітному двійковому коді?

96. Показати, як створити код з  $d = 6$  з чотирма інформаційними бітами. Скласти список його кодових слів.

97. Написати кодові слова коду Хемінга з одним інформаційним бітом.

98. Визначити групи парності для  $d = 3$  коду Хемінга з одинадцятьма інформаційними бітами.

99. *Ранг коду* – це відношення кількості інформаційних біт до загальної кількості біт кодового слова. *Вищий ранг* – той, що наближається до 1 і є бажаним для ефективної передачі інформації. Розробити і побудувати графік, який би порівнював ранг з відстанню 2 коду парності і коду Хемінга з відстанню 3 і 4 до 100 інформаційних біт.

100. Пояснити який з кодів з  $d = 4$  має вищий ранг – двовимірний чи код Хемінга. Задачу розв'язати у стилі вправи 101.

101. При передачі двійкової інформації на приймальній стороні цифрової системи прийняте двійкове слово 1011011. Вияснити з використанням корекції за Хемінгом правильність прийнятої кодової комбінації і виправити помилку у відповідному розряді, якщо вона має місце.

102. Зобразити кодові комбінації у вигляді поліномів: 1001; 101010; 10111.

103. Отримати циклічні коди для кодових комбінацій, приведених у попередньому прикладі.

104. Запис-зчитування інформації на флопі-диск персонального комп'ютера забезпечується в однобайтовому послідовному форматі. Скільки контрольних біт необхідно мати для того, щоб забезпечувати контроль передачі даних з використанням коду Хемінга; з використанням контролю парності?

105. Для кожного з логічних виразів знайти всі причини збудження статичних завад у відповідних дворівневих **I-АБО** або **АБО-I** колах. Спроекувати схеми, які б реалізовували ці ж функції без генерації завад.

$$\begin{aligned}
 y_1 &= x_1 \overline{x_0} + \overline{x_1} x_0; \\
 y_2 &= \overline{x_3} \overline{x_1} x_0 + x_2 x_1 \overline{x_0} + x_1 x_0; \\
 y_3 &= \overline{x_3} x_1 + \overline{x_1} x_0 + x_3 x_2 x_0; \\
 y_4 &= \overline{x_3} x_0 + x_2 \overline{x_1} + x_3 x_2 x_1 x_0 + x_3 \overline{x_2} x_1 \overline{x_0}; \\
 y_5 &= (x_3 + x_0 + x_1) \cdot (\overline{x_0} + x_2); \\
 y_6 &= (\overline{x_3} + \overline{x_1} + \overline{x_2}) \cdot (\overline{x_3} + \overline{x_0} + \overline{x_2}) \cdot (\overline{x_0} + x_1 + x_2); \\
 y_7 &= (x_3 + x_1 + \overline{x_2}) \cdot (\overline{x_3} + \overline{x_0} + x_1 + x_2) \cdot (\overline{x_0} + \overline{x_1}) \cdot (x_0 + x_2).
 \end{aligned}$$

## Розділ 2

### СТРУКТУРИ БАЗОВИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ

#### 2.1. Характеристика цифрових сигналів

Цифрові сигнали лог. “0” і лог. “1”, які використовуються в курсі дискретної математики, виступають ідеалізацією тих сигналів, що мають місце в реальних електронних схемах.

У ключових схемах, які використовуються при двійковому представленні інформації, значенням лог. “0” та лог. “1” присвоюються обмежені діапазони напруг, що розміщуються в інтервалі від нуля до діючої напруги живлення логічних схем.

Не розглядаючи у даному параграфі технічні характеристики апаратних засобів реалізації операцій над цифровими сигналами, визначимось лише з основними параметрами реальних сигналів.

Які б технічні засоби не використовувались, їх параметри загалом можливо оцінити за допомогою характеристики  $U_{\text{ВИХ}} = f(U_{\text{ВХ}})$  (рис. 2.1), де, відповідно,  $U_{\text{ВХ}}$  та  $U_{\text{ВИХ}}$  – напруги, що присутні на вході та виході електронного пристрою з урахуванням їх нестабільності.

Як відомо, логічні елементи у будь-яких цифрових пристроях мають достатньо розгалужені послідовні з’єднання. Тому, проходячи від одного логічного елемента до іншого, інформаційний сигнал може розсіювати свою енергію. Для запобігання цьому кожен логічний елемент повинен забезпечувати підсилення вхідного сигналу як за напругою, так і за потужністю. До того ж, відомо, що функціональна повнота логічних елементів обов’язково включає інверсію сигналу. Тому найпростішим логічним елементом може бути транзистор, включений за схемою із загальним емітером

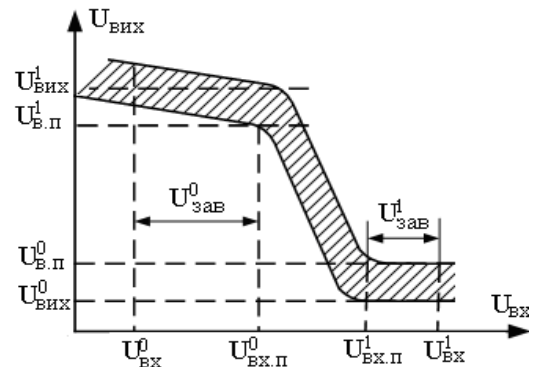


Рис. 2. 1

(загальним витоком для МДН-транзисторів), який, як відомо, інвертує вхідний сигнал. Зображена на рис. 2.1 діаграма відповідає інверсній схемі логічного елемента. Це означає, що цифровий пристрій повинен не тільки формувати вихідні сигнали з параметрами, близькими до ідеальних “0” та “1”, а й за рівнем вхідного сигналу відрізняти значення “0” та “1”.

Як рівень логічного нуля, так і одиниці для будь-якого логічного елемента є не конкретною величиною, а деяким визначеним діапазоном напруг. Для сигналу вихідної одиниці це діапазон напруг в інтервалі від  $U_{\text{вих}}^1$  до порогової напруги  $U_{\text{в.п.}}^1$ , з якої починається перехід цифрового елемента в інший стан. Сигналам вихідної логічної одиниці відповідають низькі рівні вхідного сигналу  $U_{\text{вх}}^0$  та  $U_{\text{вх.п.}}^0$ . Під  $U^1$  та  $U^0$  розуміють номінальні значення напруги цифрового елемента в статичному режимі. *Порогові значення напруги  $U_{\text{п.}}^1$ ,  $U_{\text{п.}}^0$  відповідають найбільшому та найменшому значенням відповідних рівнів, при яких починається перехід логічного елемента в інший стан. Кількісно порогові значення визначаються точкою на характеристиці  $U_{\text{вих}} = f(U_{\text{вх}})$ , в якій модуль диференційного коефіцієнта передачі цифрового елемента дорівнює одиниці. Порогові рівні визначають стійкість до завад логічних елементів. Виділяють статичну та динамічну завадостійкість. Статична завадостійкість оцінюється як мінімальна різниця між значеннями вихідного та вхідного сигналів визначеного рівня:*

$$\begin{cases} U_{\text{зав}}^1 = U_{\text{вх.}}^1 - U_{\text{вх.п.}}^1 ; \\ U_{\text{зав}}^0 = U_{\text{вх.п.}}^0 - U_{\text{вх.}}^0 . \end{cases} \quad (2.1)$$

З (2.1) бачимо, що *статична завадостійкість – це мінімальна величина завади на виході цифрового елемента, що може привести до зміни стану підключеного до нього логічного елемента тієї ж серії. Статична завадостійкість є параметром, який фактично не залежить від часу, тобто тривалість завади набагато перевищує час зміни стану ключового елемента. Якщо ж час дії завади близький до часу переключення елемента, то така завада називається динамічною.*



Цифрові сигнали характеризуються також часовими характеристиками, які здебільшого визначаються як динамічні параметри логічних ключів. До них відносяться:

$t^{1,0}$  – час переходу виходу елемента зі стану “1” в “0” – це інтервал часу, протягом якого напруга на виході елемента змінюється від рівня “1” до рівня “0”. Вказані рівні вимірюються відповідно при значеннях 0,9 та 0,1 амплітуди сигналу  $U_M$  (рис. 2.2);

$t^{0,1}$  – час переходу вихідного сигналу з “0” в “1” – це інтервал часу, протягом якого напруга на виході елемента переходить з “0” в “1”;

$t_\Phi$  – тривалість фронту сигналу – інтервал часу між значеннями сигналу на рівнях  $(0,1 - 0,9) U_M$ , при зміні сигналу з “0” до “1”;

$t_C$  – тривалість спаду сигналу – інтервал часу між значеннями сигналу на рівнях  $(0,1 - 0,9) U_M$ , при зміні сигналу від “1” до “0”;

$t_3^{1,0}$  – час затримки включення елемента – інтервал часу між фронтом вхідного та спадом вихідного імпульсів, заміряні на рівні  $0,5 U_M$ ;

$t_3^{0,1}$  – час затримки виключення елемента – інтервал часу між спадом вхідного та фронтом вихідного імпульсів, виміряні на рівні  $0,5 U_M$  (рис. 2.2);

$t_{3,PC}$  – середній час затримки розповсюдження сигналу, який знаходиться за формулою:

$$t_{3,PC} = \frac{(t_3^{0,1} + t_3^{1,0})}{2}, \quad (2.2)$$

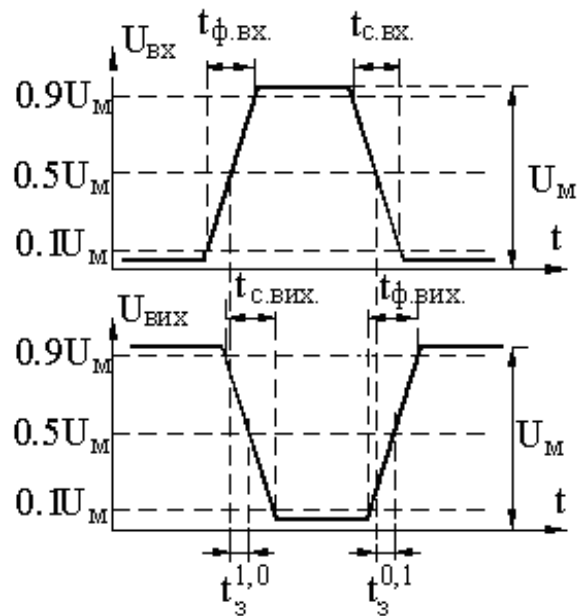


Рис. 2. 2

## 2.2. Транзисторні ключові схеми

### 2.2.1. Ключі на біполярних транзисторах

**Базовий біполярний ключ.** Схема найпростішого транзисторного ключа приведена на рис. 2.3, а. Рис. 2.3, б пояснює режими роботи транзистора за допомогою сім'ї вихідних характеристик.

Лінія навантаження **АВ**, яка описується відомим рівнянням:

$$U_{KE} = E_K - I_K R_K,$$

задає положення робочої точки транзистора на сім'ї його вихідних характеристик.

Для ключового режиму характерними є дві точки – **А** та **В**. Точка **А** на лінії навантаження характеризується тим, що струм бази має зворотній напрямок і визначається зворотнім струмом колекторного переходу:  $I_B = -I_{K0}$

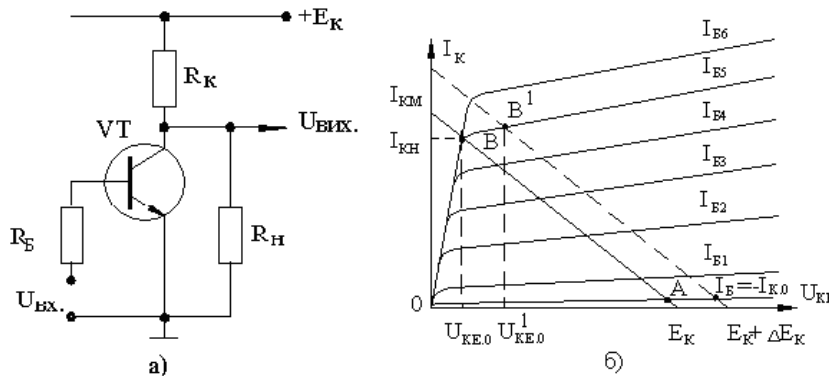


Рис. 2.3

У схемі, що розглядається, не використовуються спеціальні джерела живлення для фіксації або зміщення положення робочої точки. Тому робота транзистора в точці **А** лінії навантаження забезпечується тільки низьким рівнем вхідної напруги і називається *режимом запирання* або *режимом відсічки*. Напруга на виході ключа в цьому режимі  $U_3 = E_K - I_{K0} R_K$  є величиною меншою, ніж напруга живлення  $E_K$ . Звідси витікає, що сигнал логічної одиниці за рівнем не може досягнути значення напруги  $E_K$ , і для зменшення цієї різниці необхідно вибирати транзистори з малим значенням  $I_{K0}$ , а  $R_K$  брати якомога меншим.

Точка **В** характеризує режим, який називається *режимом відкритого стану*. Для цієї точки справедливе співвідношення:

$$I_{\text{КН}} = \beta I_{\text{Б5}} + I_{\text{К0}}(1+\beta) \approx \beta I_{\text{Б5}}, \quad (2.3)$$

де  $\beta$  – статичний коефіцієнт передачі струму транзистора, включеного за схемою з загальним емітером;  $I_{\text{Б5}}$  – базовий струм, при якому транзистор знаходиться на границі активного режиму і режиму насичення.

Залишкова напруга на колекторі відкритого транзистора  $U_{\text{КЕ.0}} = E_{\text{К}} - I_{\text{К}} R_{\text{К}}$  повинна мати мінімальне значення. Реальні величини  $U_{\text{КЕ.0}}$  у залежності від типу транзистора лежать у межах 0,05...1 В.

Особливість цієї робочої точки полягає в тому, що робота транзистора в ній забезпечується як базовим струмом  $I_{\text{Б5}}$ , так і струмами бази, більшими ніж  $I_{\text{Б5}}$ .

Розглянемо більш детально особливості робочої точки **В**, якій відповідає колекторний струм  $I_{\text{КН}}$  і струм бази  $I_{\text{Б5}}$ . Допустимо, що напруга живлення збільшилась на величину  $\Delta E_{\text{К}}$  (рис. 2.3, б). Лінія навантаження зміститься вправо, і робоча точка **В** перейде в **В**<sup>1</sup>. При цьому транзистор перейде в активний режим, а напруга на транзисторі  $U_{\text{КЕ.0}}$ , що відповідала рівню логічного нуля, збільшиться до  $U_{\text{КЕ.0}}^1$ . Подібна ситуація може виникнути також у випадку, коли величина навантаження  $R_{\text{Н}}$  зміниться у бік зменшення опору. Щоб запобігти такому явищу необхідно, як витікає з рис. 2.3, б, збільшити струм бази на таку величину, щоб прогнозовані коливання напруги живлення і величини навантаження не могли впливати на положення точки **В**, тобто повинна мати місце нерівність  $I_{\text{Б}} > I_{\text{Б5}}$ .

Такий режим роботи транзистора називається *режимом насичення* і характеризується параметром  $N$ , який називається *коефіцієнтом насичення*

$$N = \frac{I_{\text{Б}}}{I_{\text{Б5}}}. \quad (2.4)$$

З (2.4) знаходимо

$$I_{\text{Б}} = N \frac{I_{\text{КН}}}{\beta}. \quad (2.5)$$

Параметр  $N$  задається в межах  $1,5 \dots 3$ , а струм бази для виконання умови (2.5) забезпечується вхідними колами відповідно до формули:

$$I_B = \frac{U_{ВХ} - U_{БЕ}}{R_B}, \quad (2.6)$$

де  $U_{БЕ}$  – падіння напруги на відкритому переході база-емітер.

Режим насичення використовується для створення умов надійної роботи транзистора у відкритому стані, адже забезпечує незалежність вихідного нульового рівня від нестабільності навантаження, напруги живлення, а також від коливань параметрів самого транзистора.

Фізична основа цього явища полягає у наступному. В активному режимі роботи транзистора між струмом бази та струмом колектора існує строга

пропорція, яка визначається коефіцієнтом передачі  $\beta$ . У режимі насичення

колекторний струм не залежить від струму бази, якщо його величина знаходиться в інтервалі  $I_{Б5} \leq I_B \leq N I_{Б5}$ . Суттєве

підвищення значення струму бази приводить до накопичення заряду в області бази, що забезпечує стабільність колекторного

струму навіть при наявності відмічених вище дестабілізуючих факторів.

Але наявність режиму насичення має свої недоліки, які проявляються у динамічних режимах роботи ключа. Діаграми, які пояснюють особливість динамічних режимів, приведені на рис. 2.4.

Поява

вхідного відкриваючого імпульсу приводить, відповідно до (2.6), до появи імпульсу базового струму (рис. 2.4, б), який без урахування базових та паразитних ємностей фактично повторює форму імпульсу вхідної напруги.

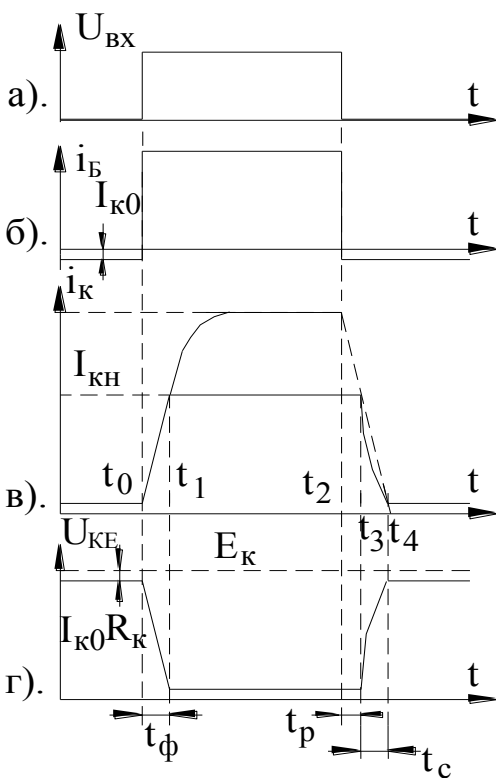


Рис. 2.4

Але характер зміни колекторного струму відрізняється від базового. Пояснюється це інерційністю колекторних кіл транзистора, які можуть бути враховані еквівалентною постійною часу  $\tau_T = \tau_\beta + \tau_k$ , де  $\tau_\beta = (\beta + 1) / f_\beta$  – постійна часу, що визначається обмеженою швидкістю руху зарядів;  $f_\beta$  – гранична частота роботи транзистора при включенні за схемою з загальним емітером;  $\tau_k = r_k C_k$  – постійна часу колекторного переходу ( $r_k$ ,  $C_k$  – його параметри).

Колекторний струм при вказаних умовах змінюється за законом:

$$i_K(t) \approx \beta U_{ВХ} \frac{1 - \exp\left(-\frac{t}{\tau_T}\right)}{R_B}. \quad (2.7)$$

Якщо  $I_B \gg I_{B5}$ , то забезпечується швидке наростання струму колектора до величини  $I_{KH} = \beta I_B$ . Завдяки цьому тривалість фронту вихідного імпульсу

$$t_\phi = t_1 - t_0 = \tau_T \ln\left(\frac{N}{N-1}\right), \quad (2.8)$$

знайдена з (2.7) при умові, що  $I_K(t) = I_{KH}$  при  $t = t_1$ , буде дещо скорочуватись.

Запірання транзистора починається з моменту  $t_2$ , коли імпульс вхідної напруги  $i$ , відповідно, вхідний струм, зменшуються до нуля.

З цього моменту накопичений заряд в області бази транзистора, обумовлений надмірним струмом бази, починає зменшуватись від величини, еквівалентної  $\beta I_B$ , до нуля, з постійною часу  $\tau'_\beta \approx (2 \dots 4) \tau_\beta$ . В інтервалі часу  $t_2 - t_3$  величина заряду зменшується до величини, для якої характерною є пропорційність  $I_K = \beta I_B$ . Для моменту  $t_3$  ця пропорція визначається формулою (2.3). Звідси витікає, що в інтервалі часу  $t_2 - t_3$ , який називається *часом розсмоктування збиткових носіїв в області бази*  $t_p$ , колекторний струм транзистора залишається незмінним.

Зрозуміло, що вказаний інтервал часу, який визначається за формулою:

$$t_p = \tau'_\beta \ln N, \quad (2.9)$$

є вимушеною затримкою вимикання транзистора, причому вказана затримка пропорційна коефіцієнту насичення  $N$ .

Після завершення інтервалу  $t_p$  транзистор, знаходячись в активному режимі переходить від відкритого стану до закритого за інтервал часу спаду  $t_c$ :

$$t_c = \tau_\beta \ln 9 \approx 2,2 \tau_\beta . \quad (2.10)$$

Розглянутий ключ відноситься до типу *біполярних насичених ключів зі статичним навантаженням*.

Як витікає з виразів (2.8) – (2.10), задача створення швидкодіючого біполярного ключа пов'язана із суперечливими вимогами щодо коефіцієнта насиченості.

**Приклад 2.1.** Оцінити часові характеристики перемикання ключа на транзисторі КТ315Б з наступними параметрами:  $\beta = 50 \dots 350$ ;  $\tau_K = 500$  пс;  $f_\beta = 250$  МГц.

*Розв'язання.* Задаємо коефіцієнт насичення  $N = 3$ .

Обчислення ведемо для найменшого значення  $\beta = 50$ .

$$\tau_\beta = \frac{\beta + 1}{f_\beta} = \frac{50 + 1}{250 \cdot 10^6} = 0,204 \cdot 10^{-6} \text{ с; с;}$$

$$\tau_T = \tau_\beta + \tau_K = 0,204 \cdot 10^{-6} + 500 \cdot 10^{-12} \approx 0,204 \cdot 10^{-6} \text{ с;}$$

$$t_\Phi = \tau_T \ln \left( \frac{N}{N-1} \right) = 0,204 \cdot 10^{-6} \cdot \ln \left( \frac{3}{2} \right) \approx 0,083 \cdot 10^{-6}$$

$\tau'_\beta = (2 \dots 4) \tau_\beta$ , вибираємо

$$\tau'_\beta = 3 \tau_\beta = 3 \cdot 0,204 \cdot 10^{-6} = 0,612 \cdot 10^{-6} \text{ с.}$$

$$t_p = \tau'_\beta \ln N = 0,612 \cdot 10^{-6} \cdot \ln 3 \approx 0,672 \cdot 10^{-6} \text{ с;}$$

$$t_c = 2,2 \tau_\beta = 2,2 \cdot 0,204 \cdot 10^{-6} = 0,449 \cdot 10^{-6} \text{ с.}$$

У прикладі маємо досить великі значення часових характеристик, які не характерні для цифрових пристроїв. Але, у той же час, приведені дані говорять про те, що підвищення швидкодії перемикання ключів є дійсно важливою задачею.

У практиці побудови швидкодіючих ключів існує багато шляхів вирішення вказаних протиріч. Розглянемо лише ті, які широко використовуються у цифровій схемотехніці.

**Біполярний ключ з нелінійним зворотнім зв'язком.** Розв'язання протиріч, які відмічались раніше, досягається у схемі, де транзистор не входить у режим насичення, а низький рівень вихідної напруги стабілізується (рис. 2.5).

У схемі використовується діодний обмежувач, робота якого полягає у наступному. Якщо транзистор ненасичений, то напруга на його колекторі перевищує напругу на базі, і діод зміщується в зворотньому напрямку. Весь вхідний струм  $I_{ВХ}$  замикається через перехід

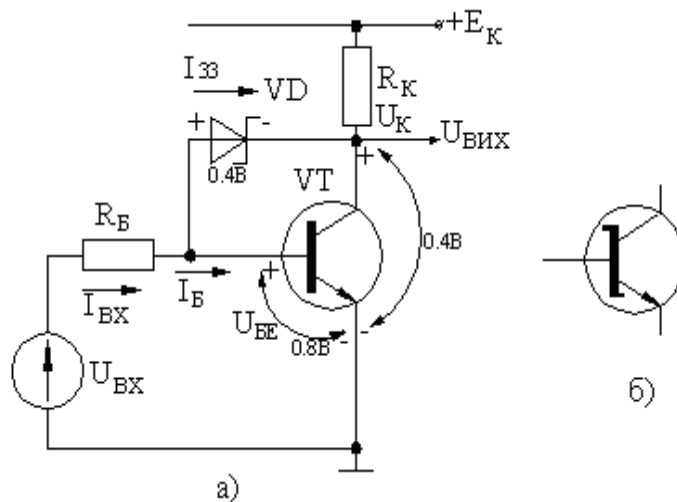


Рис. 2.5

база-емітер транзистора. Коли вхідний струм досягає величини, при якій транзистор входить у режим насичення, величина напруги на колекторі зменшується завдяки наявності збиткового об'ємного заряду в базі, тобто виконується співвідношення  $U_{КЕ} \leq U_{БЕ}$ .

При цьому діод відкривається, і частина вхідного струму (струм зворотнього зв'язку  $I_{ЗЗ}$ ) відгалуджується через нього, замикаючись через зовнішні кола схеми. Але це можливо лише в тому випадку, коли падіння напруги  $U_d$  на відкритому діоді достатньо мале, а точніше:

$$U_d + U_{КЕ} \leq U_{БЕ} . \quad (2.11)$$

Умова (2.11) не може бути виконана при використанні діодів з р-п переходом, оскільки падіння напруги на них приблизно дорівнює падінню напруги  $U_{БЕ}$ . Тому для створення швидкодіючих ключів використовуються діоди, створені не на основі р-п переходу, а на основі переходу метал-напівпровідник. Такі діоди називаються *діодами Шоткі*. Їх особливість полягає у тому, що швидкість переносу зарядів через контакт метал-напівпровідник приблизно на три порядки вища, ніж у р-п переході. Час їх вимикання дуже

малий (до 100 пс, тоді як для р-п переходів – до 100 нс) і не залежить від температури. Інша суттєва особливість діодів Шоткі полягає в тому, що напруга їх відкриття  $U = 0,2 \dots 0,4$  В, що в 2-3 рази менше, ніж для р-п переходу.

Використання діодів Шоткі забезпечує роботу біполярного транзистора на межі насичення, такий транзистор здобув назву *транзистора Шоткі* (рис. 2.5, б), який зазвичай має малий час вимикання і дає можливість форсувати вмикаючі та вимикаючі струми для зниження  $t_{\Phi}$  і  $t_C$ .

### ***Ключ з динамічним навантаженням.***

Як відомо, в аналоговій схемотехніці підсилювач класу А з динамічним навантаженням представляє собою схему, в якій колекторний резистор заміщений транзистором, що працює в режимі джерела струму. Перевага такої схеми полягає в тому, що низькоомний колекторний резистор у режимі підсилення замінюється динамічним високоомним колекторним опором транзистора-навантаження. Це дає можливість суттєво підняти коефіцієнт підсилення каскаду.

У цифровій схемотехніці під терміном “динамічне навантаження” мається на увазі таке колекторне навантаження транзистора, величина якого змінюється зі зміною режиму роботи ключа. Прикладом схеми з динамічним навантаженням може бути підсилювач класу В. Особливість цієї схеми полягає у тому, що, відкриваючи регулюючий транзистор, одночасно збільшують (зменшують) величину колекторного опору. При використанні в якості навантаження транзистора того ж типу провідності, що і регулюючий, у такій схемі повинно забезпечуватись протифазне керування ними. З цією метою використовується спеціальний фазоінверсний каскад. За таким самим принципом будується і ключ, з тією лише різницею, що відкритому стану ключа відповідає закритий стан транзистора-навантаження, і навпаки (рис. 2.6).

Використання такої схеми ключа дозволяє:

- підвищити навантажувальну здатність логічного елемента;



- підвищити пороговий рівень вхідної напруги  $i$ , в результаті, завадостійкість ключа;
- підвищити швидкість перемикання в динамічних режимах.

Розглянемо особливості роботи ключа згідно з рис. 2.6.

При  $U_{ВХ} = 0$  транзистори VT1 і VT3 знаходяться у закритому стані за рахунок зміщуючих потенціалів  $U_{BE.1}$  і  $U_{BE.3}$ . Напряга на колекторі транзистора VT1  $U_{K1} = E_K - I_{K0} R_{K1}$  близька до рівня напруги живлення. Вона забезпечує активний режим транзистора VT2, який у даному випадку працює на навантаження  $R_H$  як

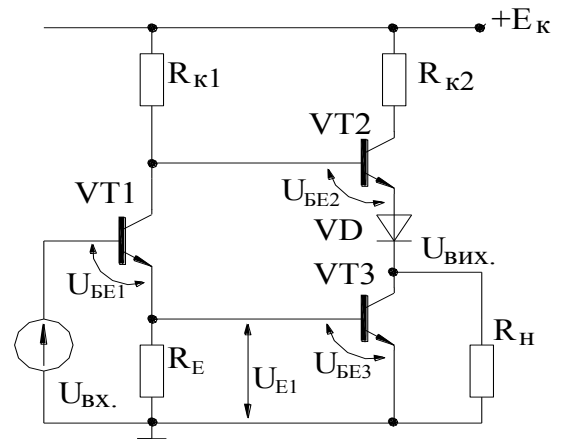


Рис. 2. 6

емітерний повторювач. Тобто  $U_{ВИХ} = E_K - I_{K.0} R_{K1} - U_{BE.2} \approx E_K$ .

При наростанні вхідної напруги транзистор VT1 спочатку знаходиться у закритому стані. З моменту, коли  $U_{ВХ} \geq U_{BE.1}$ , він переходить в активний режим роботи зі струмом бази, який визначається за формулою:

$$I_{B1} = \frac{U_{ВХ} - U_{E1}}{(\beta + 1) R_E}.$$

З ростом базового струму  $I_{B1}$  підвищується струм емітера  $I_{E1}$  і напруга  $U_{E1}$ . Коли створюється нерівність  $U_{E1} > U_{BE.3}$ , відкривається транзистор VT3 і його перехід база-емітер шунтує резистор  $R_E$ .

Струм  $I_{B1}$  зростає, і транзистор VT1 входить у режим насичення. За рахунок струму від джерела  $E_K$  через  $R_{K1}$  і відкритий транзистор VT1 транзистор VT3 також входить у режим насичення.

Відповідно до умов роботи ключа з динамічним навантаженням, транзистор VT2 повинен закритися. Але у розглянутих умовах це неможливо, оскільки потенціал його бази вищий, ніж потенціал емітера. Для вирішення цієї проблеми необхідно підвищити потенціал емітера VT2, що робиться за рахунок

допоміжного діода, який вмикається відповідно до рис. 2.6. Після закривання транзистора VT2 напруга на виході схеми стає близькою до “0”. Максимальний вихідний струм ключа визначається струмом зміщеного у зворотньому напрямку переходу база-емітер транзистора VT2.

При зміні станів транзисторів VT2 і VT3 мають місце ситуації, коли на короткому інтервалі часу один з транзисторів ще не встиг закритися, а інший вже відкрився. В результаті через обидва транзистори проходить значний струм, який називається *динамічним струмом перемикавання ключа*. Величина динамічного струму може досягти величин недопустимих для безпечної роботи транзисторів. Тому для їх захисту в схемі передбачається резистор  $R_{K2}$ , який обмежує величини динамічних струмів на безпечному рівні.

Існують й інші проблеми при побудові ключа, частина з яких буде розглянута при вивченні конкретних схем логічних елементів.

**Схеми з перемиканням струму.** Відома у підсилювальній техніці схема

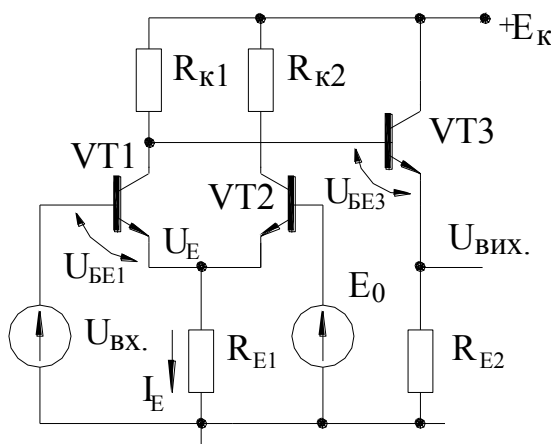


Рис. 2. 7

диференційного каскаду (рис. 2.7, транзистори VT1...VT2) не могла не привернути увагу розробників логічних схем.

Пов'язано це з тим, що розглянуті вище схеми у тій чи іншій мірі не задовольняють основним вимогам, які необхідно закладати у структури логічних елементів:

- ненасиченні транзисторні схеми;
- мінімальний перепад напруг;
- величини струмів повинні забезпечити мінімальний час перемикавання.

Розглянемо особливість роботи приведеної схеми як логічного ключа.

На базу транзистора VT2 подається постійний потенціал  $E_0$ , а вхідний логічний сигнал подається на базу VT1.

При нульовому вхідному сигналі транзистор VT2 відкритий і струм  $I_E$  замикається через  $R_{E1}$ , еміттер-колектор VT2 і  $R_{K2}$ . Падіння напруги на  $R_{E1}$  створює на емітері VT1 потенціал  $U_E$ , який приводить до запирання VT1. В результаті напруга на колекторі VT1 практично дорівнює напрузі живлення  $E_K$ , а вихідний сигнал  $U_{ВИХ}$ , що знімається з резистора  $R_{E2}$  еміттерного повторювача, виконаного на VT3, визначається рівнем (рис. 2.7):

$$U_{ВИХ}^1 = E_K - U_{BE.3}. \quad (2.12)$$

Слід зазначити, що в такому режимі вхідний струм визначається зворотнім струмом колекторного переходу VT1. Транзистор VT2 знаходиться в активному режимі, тому що при закритому VT1 транзистор VT2 працює як підсилювач з глибоким зворотнім зв'язком, і напруга на його колекторі знаходиться за формулою  $U_{K2} = E_K - I_K R_{K2}$ .

Практично в такому ж стані схема залишатиметься в діапазоні вхідних напруг  $0 < U_{ВХ} < E_0$ .

При  $U_{ВХ} = E_0$  починається процес перемикання струму в схемі. При такому співвідношенні обидва транзистори знаходяться в активному режимі з однаковими напругами на колекторах. Активний режим забезпечуватиметься у деякому діапазоні напруг, при якому вхідний струм підвищиться і визначатиметься величиною:

$$I_{ВХ} = \frac{U_{ВХ} - U_{BE.1}}{(\beta + 1) R_E}.$$

Фактично у розглянутій схемі потенціал  $E_0$  є пороговим рівнем, при якому проходить перемикання транзисторів. Звідси витікають два важливі висновки:

- поріг перемикання можна задавати на будь-якому (в межах напруги живлення) рівні;
- порогова напруга виступає не фізичною характеристикою транзистора, а схемним параметром (завжди).

При  $U_{ВХ} > E_0$ , якщо  $E_0 - U_E < U_{BE2}$  ( $U_E = U_{ВХ} - U_{BE1}$ ), транзистор VT2 закривається, і напруга на його колекторі дорівнює  $E_K$ , а струм  $I_E$  замикається

через VT1. При підвищенні вхідної напруги до  $E_K$  потенціал колектора VT1 матиме величину меншу, ніж потенціал бази:  $U_{B1} = E_K > U_{K1}$ , а це означає, що транзистор VT1 знаходиться в режимі насичення. Звідси витікає, що величина вхідного сигналу повинна бути меншою, ніж  $E_K$ , і зменшена на величину  $U_{BE}$ . Таке зниження досягається за допомогою емітерного повторювача на VT3, завдяки якому максимальне значення вхідного сигналу:

$$U_{ВХ}^1 = E_K - U_{BE.3}. \quad (2.13)$$

Але, з іншого боку, це призведе до зміни вихідного сигналу схеми, і нульовий рівень його:

$$U_{ВІХ}^0 = E_K - 2 U_{BE.3}. \quad (2.14)$$

Враховуючи, що з умов завадостійкості величини відкриваючої та запираючої завад повинні бути однаковими, з урахуванням (2.13) та (2.14) знаходимо величину порогової напруги:

$$E_0 = E_K - \frac{3}{2} U_{BE}. \quad (2.15)$$

Приведений у (2.13) – (2.15) незначний діапазон напруг між “0” та “1” при достатньо високій напрузі живлення, як відмічалось вище, є позитивною особливістю схеми. Вона є зрозумілою з фізичної точки зору, якщо враховувати, що вихідні умови і, відповідно, вхідні рівні формує емітерний повторювач.

Оскільки транзистори ключа не входять у режим насичення, то час зміни стану кожного з них визначається постійними часу колекторів транзисторів  $R_K C_K$ , що дає можливість забезпечити швидкодію на межі можливого (порівняйте у прикладі 2.1 постійні часу  $R_K C_K$  і  $\tau_\beta$ ).

### 2.2.2. Ключі на польових транзисторах

**Ключ з пасивним навантаженням.** Схемотехніка цифрових систем з використанням польових транзисторів інтенсивно розвивається, і її використання швидко розширюється з низькочастотної електронної автоматики

в такі галузі, як вимірювальна та обчислювальна техніка завдяки ряду позитивних якостей польових транзисторів:

- низька залишкова напруга на відкритому ключі, яка дозволяє здійснювати комутацію електронних сигналів низьких рівнів;
- високій опір ключа в закритому стані;
- низька потужність споживання ключа, обумовлена особливостями як транзистора, так і схемотехніки, що застосовується;
- висока технологічність створення інтегральних схем;
- площа інтегрального транзистора на кристалі значно менша, ніж біполярного, що дає можливість суттєво підвищувати ступінь інтеграції схем.

В інтегральній схемотехніці використовуються різні типи польових транзисторів. Транзистори на базі керованого p-n переходу (JFET – *Junction Field Effect Transistor*) знаходять використання в аналогових схемах перемикачів, а МДН-структури (MOSFET, MISFET – *Metal-Oxigen(Insulator)-Semiconductor Field Effect Transistor*) використовуються в цифровій схемотехніці. Як ключові елементи використовуються лише транзистори з індукованим каналом.

Принципова схема такого ключа з n- каналом приведена на рис. 2.8, а.

Подібно до біполярних, польові транзистори найбільш повно характеризуються сім'єю вихідних характеристик (рис. 2.8, б), на яких умовно виділяються дві області. Перша з них (позначена цифрою I) – область наростання струму. В цій області, яка традиційно носить назву тріодної, канал транзистора може розглядатись як прилад, що керується напругою затвор-витік  $U_{зв}$ . В області II, яка називається *пентодною*, струм мало залежить від напруги стоку  $U_C$ .

Вигляд вихідних характеристик ілюструє важливу особливість МДН-транзисторів – відсутність залишкової напруги, яка характерна для біполярних структур.

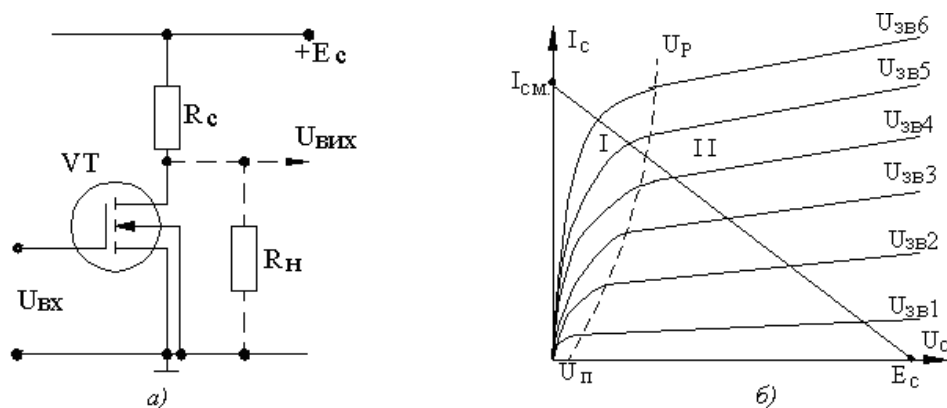


Рис. 2. 8

Зменшення величини опору відкритого каналу досягається або за рахунок підвищення опору  $R_C$ , або шляхом підвищення напруги  $U_{ЗВ}$ , яка має свої обмеження.

Розділююча лінія між двома областями характеристики  $|U_{ЗВ} - U_{П}| = U_P$  визначається розділюючою напругою  $U_P$ .

При аналізі схем з польовими транзисторами широко використовується вхідна стоко-затворна характеристика  $I_C = f(U_{ЗВ})$  (рис. 2.9), на якій порогова напруга  $U_{П}$  визначає той рівень вхідної напруги, при якому з'являється провідність індукованого каналу.

Стоко-затворна характеристика покладена в основу побудови схеми заміщення транзистора з джерелом струму  $I_C$ , що задається крутизною цієї характеристики  $g$  (рис. 2.10), де  $r_{ВС}$  – динамічний опір витік-стік;  $C_{ЗВ}$ ,  $C_{ЗС}$ ,  $C_{ВС}$  паразитні міжелектродні ємності транзистора (позначення В, З, С – відповідно, витік, затвор, стік).

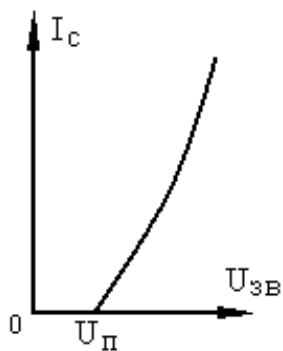


Рис. 2. 9

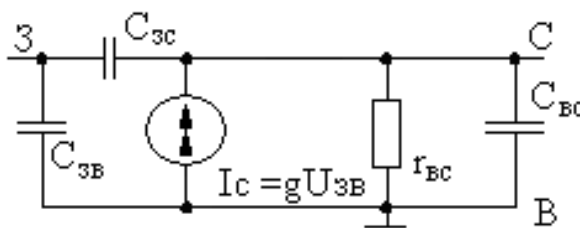


Рис. 2. 10

Робота МДН-транзистора в режимі ключа суттєво відрізняється від роботи біполярного транзистора. До того часу, поки вхідна напруга менше порогового рівня  $U_{П}$ , транзистор знаходиться в закритому стані і напруга на його виході дорівнює напрузі джерела живлення  $E_C$ . При збільшенні  $U_{ВХ}$  транзистор спочатку проходить область пентодних характеристик  $\Pi$ , а потім при досягненні співвідношення  $U_{ВХ} > U_{П} + U_p$  переходить в тріодну область  $I$ , де його внутрішній опір зменшується до величини  $R_B \approx (g U_{П})^{-1}$ .

При цьому вихідна напруга визначатиметься співвідношенням між величинами  $R_C$  і  $R_B$ :

$$U_{\text{вих}} = \frac{R_B E_C}{R_C + R_B}.$$

У зв'язку з тим, що в МДН-транзисторах режим насичення відсутній, величина вихідної напруги ключа (низький рівень) залежить від  $R_C$ . Звідси витікає, що величина опору  $R_C$  повинна розраховуватись, виходячи з умов завадостійкості ключа.

Специфіка процесів при зміні станів ключа проявляється у тому, що його внутрішні процеси мають значно більшу швидкість, ніж зовнішні, пов'язані з зарядом та розрядом паразитних конденсаторів. Найбільший вплив на тривалість фронтів має ємність конденсатора  $C_{BC}$  (рис. 2.10), поєднана з ємністю навантаження, монтажу та ін. Заряд конденсатора  $C_{BC}$  відбувається через резистор  $R_C$  від джерела живлення  $E_C$ , а розряд – через відкритий канал транзистора з опором  $R_B$ . Враховуючи, що  $R_C \gg R_B$  (співвідношення між ними знаходиться в інтервалі 10...20), зарядний струм приблизно на стільки ж менший, ніж розрядний, а це значить, що швидкодія ключа обмежується часом заряду конденсатора  $C_{BC}$  через резистор  $R_C$ . Зменшення часу заряду  $C_{BC}$  за рахунок зниження опору резистора  $R_C$  недоцільно, оскільки це веде до зниження завадостійкості ключа.

**Комплементарні МДН-ключі.** Розглянута схема з резистивним навантаженням має ряд недоліків як схемотехнічного, так і технологічного

характеру, які досить важко усунути. Вихід з цієї ситуації знаходиться у використанні динамічного навантаження, в якості якого використовується транзисторний ключ протилежної провідності (рис. 2.11).

У приведеній схемі обидва транзистори включені однаково за схемою з загальним витоком, і ключ є симетричним по відношенню до вхідної та вихідної напруги. Вхідна напруга для обох транзисторів визначається по-різному: для n-канального VT1 – по відношенню до загальної шини, а для р-канального VT2 – до шини живлення. Тому діаграма напруг керування ключом має вигляд, приведений на рис. 2.12, де  $U_{К.1} = U_{ВХ}$  – напруга керування транзистором VT1, а  $U_{К.2} = E - U_{ВХ}$  – напруга керування транзистором VT2.

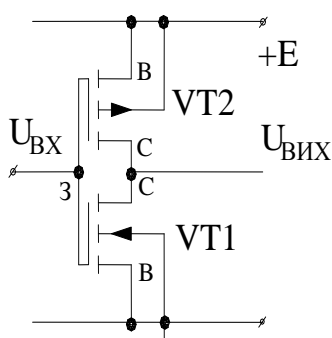


Рис. 2. 11

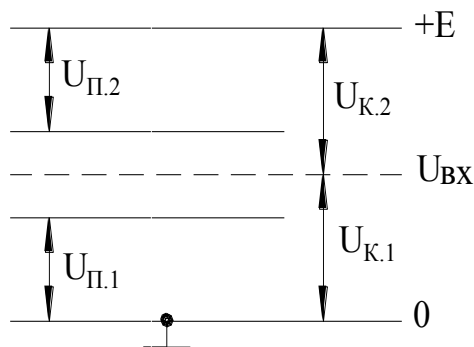


Рис. 2. 12

Враховуючи, що для кожного транзистора порогова напруга  $U_{П}$  є постійною величиною, у залежності від величини напруги живлення  $E$  можливі різні співвідношення між напругою живлення  $E$  та пороговими рівнями:

$$a) E > U_{П.1} + U_{П.2}; \quad б) E < U_{П.1} + U_{П.2}; \quad в) E = U_{П.1} + U_{П.2}. \quad (2.16)$$

Передавальні характеристики  $U_{ВИХ} = f(U_{ВХ})$  для випадків **a** та **б** наведені на рис. 2.13 а,б відповідно. Випадок **в** реалізувати практично неможливо, тому що неможливо точно задати  $U_{П.1}$  та  $U_{П.2}$ .

Конкретні значення високого і низького рівнів вихідної напруги ключа визначаються співвідношенням опорів закритого транзистора  $R_3$  та відкритого  $R_В$ . Реально  $R_3 \gg R_В$ , тому напруга високого рівня практично дорівнює напрузі живлення, а низького – майже не відрізняється від нуля.



Спочатку розглянемо передавальну характеристику ключа при виконанні співвідношення 2.16, а.

При зміні вхідної напруги від нуля до  $U_{п.1}$  транзистор VT1 закритий, а транзистор VT2, на затворі якого підтримується напруга  $U_{3.2} = E - U_{вх}$ , знаходиться у відкритому стані з робочою точкою на триодній частині вихідної характеристики.

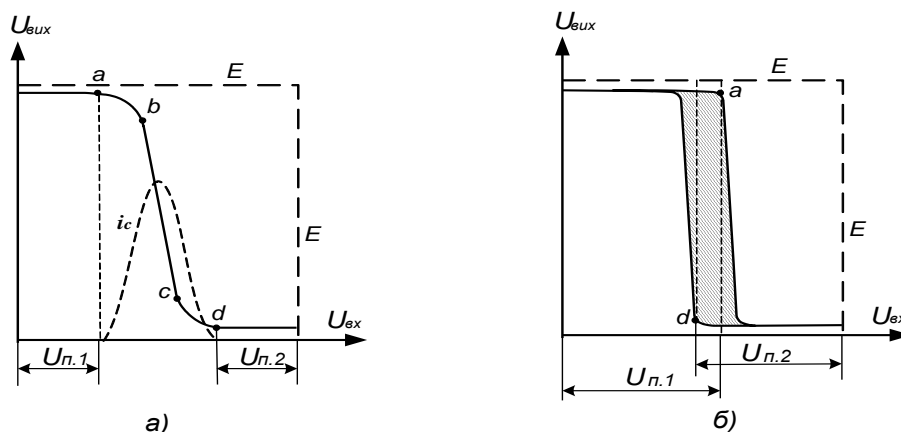


Рис. 2. 13

При перевищенні величиною  $U_{вх}$  значення  $U_{п.1}$ , транзистор VT1 почне відкриватись (точка  $a$  на передавальній характеристиці рис. 2.13,а), знаходячись спочатку в пентодній області вихідних характеристик.

При подальшому збільшенні  $U_{вх}$  транзистор VT1 буде переходити в триодну область характеристик зі зменшенням падіння напруги на ньому (участок  $c - d$  передавальної характеристики).

Транзистор VT2 навпаки зі збільшенням  $U_{вх}$  поступово закривається. Спочатку він працює в триодному режимі (участок  $a - b$ ), потім в пентодному (участок  $b - d$ ) і нарешті, коли  $U_{вх}$  стане за величиною більше ніж  $E - U_{п.2}$  повністю закриється. Транзистор VT1 при цьому буде повністю відкритим.

Характерним є те, що деякий час, коли  $U_{п.1} < E - U_{п.2}$ , обидва транзистори перебувають у відкритому стані, (участок  $b - c$  передавальної характеристики), працюючи у пентодному режимі. Прохідний струм  $i_c$ , показаний на рис. 2.13, а пунктирною лінією досягає при цьому свого максимального значення.

**Приклад 2.2.** При яких вхідних напругах обидва транзистори (рис. 2.11) будуть відкриті, якщо  $E = 10 \text{ В}$ ;  $U_{П.1} = 5 \text{ В}$ ;  $U_{П.2} = 4 \text{ В}$  ?

*Розв'язання.* Обидва транзистори перебуватимуть у відкритому стані в інтервалі вхідних напруг  $5 \text{ В} < U_{ВХ} < (10 - 4) \text{ В} = 6 \text{ В}$ .

Розглянемо тепер ситуацію, яка має місце при виконанні співвідношення (2.16, б), коли сума порогових рівнів перевищує напругу живлення  $E$  (див. рис. 2.13, б).

При підвищенні вхідної напруги до величини  $U_{ВХ} = E - U_{П.2}$  транзистор VT2 закривається. Транзистор VT1 перебуватиме у закритому стані до тих пір, поки  $U_{ВХ} < U_{П.1}$ . Коли  $U_{ВХ} = U_{П.1}$ , транзистор VT1 відкривається і після того, як вхідна напруга стане більшою рівня  $U_{П.1} + U_p$ , переходить у режим круто падаючих характеристик з практично нульовим рівнем вихідної напруги. Прохідний струм у такій схемі не з'являється.

При співвідношенні 2.16,б області *a-b-c-d* передавальної характеристики рис. 2.13, а зникають, з'являється зона перекриття (на рис. 2.13, б – заштрихована) в якій обидва транзистори знаходяться в режимі відсічки.

При цьому рівень вихідного сигналу залежить від співвідношення величин опорів закритих транзисторів і є величиною невизначеною. Вказана невизначеність зникає при підключенні зовнішнього навантаження.

Враховуючи той факт, що порогові рівні для кожного транзистора є величинами постійними, кожен з режимів може мати місце, якщо змінювати напругу живлення. Режим б, як видно з аналізу, корисний з тієї точки зору, що транзисторний ключ ні в статичних режимах, ні при зміні стану не споживає струму (якщо не враховувати наявність паразитних конденсаторів).

При підвищенні напруги живлення схема переходить у режим а, за яким, як буде показано пізніше, можуть наступати аварійні режими.

Часові співвідношення при зміні станів розглянутого ключа набагато кращі, ніж ключа з резистивним навантаженням. Перш за все, це пояснюється тим, що в процесі зарядки паразитних конденсаторів відсутній високоомний опір  $R_C$  і їх перезаряд проходить через відкриті транзистори VT1 і VT2.

Слід зазначити, що час перемикання ключа суттєво залежить від величини порогових рівнів і крутизни характеристики  $I_3 = f(U_{B3})$  транзисторів, які є технологічними параметрами і можуть покращуватись з удосконаленням технологій виробництва інтегральних схем.

## 2.3. Технічні характеристики базових логічних елементів і особливості їх використання

### 2.3.1. Транзисторно-транзисторна логіка (ТТЛ)

*Базовим* є такий логічний елемент, параметри якого найбільш повно характеризують фізичні властивості більшості інтегральних схем даної серії.

Логічним елементом в інтегральних схемах (ІС) ТТЛ-серії є багатоемітерний транзистор (БТ) (рис. 2.14).

Інформаційні вхідні сигнали подаються на емітери транзистора по відношенню до загальної шини. При низькому рівні вхідної напруги  $U_{ВХ}$  на  $i$ -му вході відповідний перехід база-емітер зміщується в прямому напрямку і струм бази

$$I_B = \frac{E_K - U_{BE}}{R_B} = I_E$$

замикається через відповідний емітерний перехід.

Оскільки напруга колекторного живлення дорівнює нулю, транзистор знаходиться в режимі насичення і  $U_{ВІХ} \approx 0$ . Якщо кількість входів транзистора більше одного, то, незалежно від напруги на них, величина струму  $I_B$  не зміниться, що дозволяє невикористані входи залишати розімкнутими, тобто якщо маємо нульовий рівень вхідної напруги на одному вході, то рівень логічних сигналів на решті входів не впливатиме на рівень вихідного сигналу.

Якщо тепер на обидва входи БТ подати сигнали високого рівня  $U_{ВХ1} = U_{ВХ2} \approx E_K$ , переходи емітер-база змістяться в зворотньому напрямку, а перехід база-колектор – у прямому.

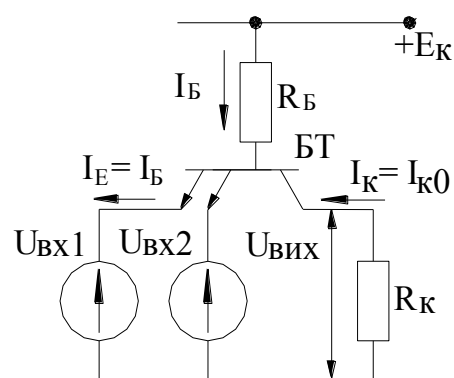


Рис. 2. 14

Завдяки цьому базовий струм буде протікати через колекторний перехід і на виході з'явиться напруга

$$U_{\text{Вих}}^1 = E_{\text{К}} \frac{R_{\text{К}}}{R_{\text{Б}} + R_{\text{К}}}. \quad (2.17)$$

У відповідності до стану емітерного і колекторного переходів, транзистор знаходиться в інверсному активному режимі. Вхідний струм від джерел інформаційних сигналів визначатиметься зворотнім струмом емітерних р-п переходів.

Як результат аналізу, з'ясуємо, що БТ реалізує логічну операцію **I (2I)**.

Але приведений логічний елемент не може використовуватись самостійно, адже рівень логічної одиниці на виході елемента (див. 2.17) нижчий, ніж на

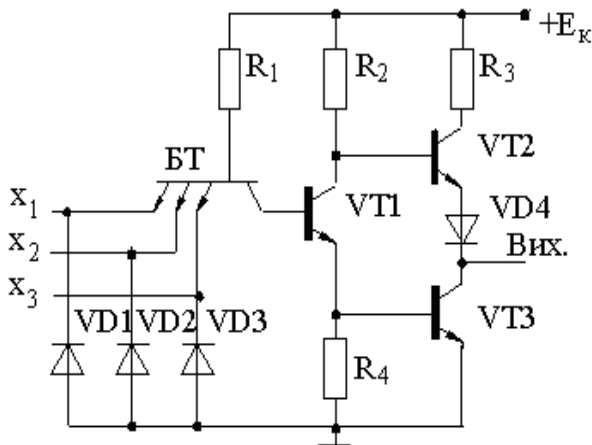


Рис. 2. 15

вході. Тому для підвищення рівня  $U_{\text{Вих}}^1$ , а також з метою забезпечення необхідних динамічних характеристик та завадостійкості, БТ використовують разом з ТТЛ-ключем із динамічним навантаженням (рис. 2.6).

На рис. 2.15 приведена принципова схема базового ТТЛ логічного елемента.

Протягом чверті століття він використовувався в серіях К134, К155 та ряду інших серій з невеликими доповненнями. На входах  $x_1 \dots x_3$  швидкодіючих схем ТТЛ включаються діоди VD1...VD3, які часто називають *антидзвонівими*. Вони обмежують амплітуду негативних завад, які створюються при розповсюдженні логічних сигналів у лініях зв'язку, між мікросхемами із-за відбиття на кінцях неузгоджених ліній. Завдяки цим діодам амплітуда завади, що потрапляє на входи БТ, буде меншою рівня порогу перемикавання. Якщо така завада, відбившись від початку лінії і змінивши свою полярність, поступить на вхід логічного елемента, то вона не зможе змінити його стан.

У мікросхемах ТТЛ, розроблених та використовуваних в останні роки (серії К1513, К1533, К531, К553), принципові схеми базових елементів вдосконалені у напрямках підвищення швидкодії та зменшення потужності споживання.

На рис. 2.16. приведена принципова схема базового елемента підвищеної швидкодії (серія К555).

Особливості цієї схеми, як видно з рисунку, полягає у наступному:

- багатомітерний транзистор замінюється елементом **I**, виконаним на діодах Шоткі, що виключає затримку, яка виникала у попередній схемі за рахунок режиму насичення **БТ**;

- в емітері фазоінверсного каскаду (транзистор VT1) розміщується генератор стабільного струму на транзисторі VT2 та резисторах  $R_4$  та  $R_5$ . Це дозволяє підвищити швидкодію каскаду і всього логічного елемента;

- замість транзистора VT2 (рис. 2.15) для підвищення навантажувальної здатності встановлюється транзисторна пара

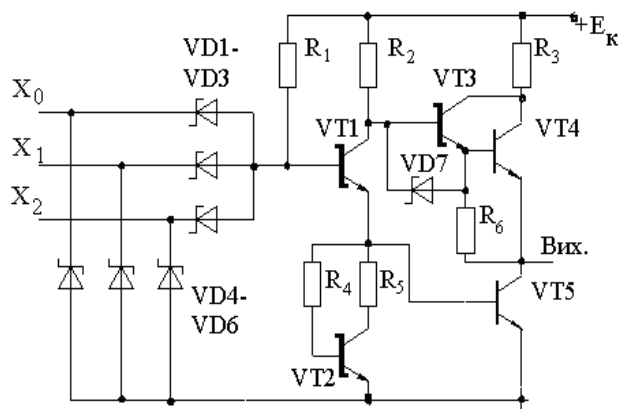


Рис. 2. 16

з включенням за схемою Дарлінгтона, а резистор  $R_6$  створює необхідну напругу на базі транзистора VT3 для його відкриття.

Діод VD7 дає можливість зменшити затримку вмикання ЛЕ за рахунок використання частини енергії, що запасється в ємності навантаження, для підвищення струму колектора VT1 у перехідних режимах.

В останніх розробках (серії К1531, К1533) між логічним елементом та вихідним каскадом з динамічним навантаженням встановлюється допоміжний каскад підсилення, що дозволяє знизити вхідний струм низького рівня, зафіксувати вхідну порогову напругу на рівні 1,5 В. В ІС вказаних серій змінена

технологія виготовлення транзисторів, що дозволило підняти межу їх робочих частот до 5 ГГц.

Для оцінки технічних можливостей різних серій ІС використовується параметр, який називається *енергією перемикання*:

$$E = t_{3,PC} \cdot P_{СП} \quad [ \text{нс} \cdot \text{мВт} = \text{пДж} ], \quad (2.18)$$

де  $P_{СП}$  – потужність споживання логічного елемента.

Параметр (2.18) дає можливість оцінити ефективність вдосконалення технологій. Так, наприклад, за останні чверть століття він зменшився з 120...140 пДж до 5...20 пДж.

У табл. 2.1 приводиться порівняльна характеристика декількох сучасних ТТЛ серій за головними параметрами і параметрами навантаження.

Технічні характеристики витримуються при вказаних параметрах навантаження.

Таблиця 2. 1

Серія ТТЛ	Параметри				Навантаження		
	$t_{3,PC}$ , нс	$F_{МАКС}$ , МГц	$P_{СП}$ , мВт	$E$ , пДж	$C_H$ , пФ	$R_H$ , кОм	Розширення по виходу
K531	3	125	20	60	15	0,28	10
K555	10	45	2	20	15	2	20
KP1531	3	130	4	12	15	0,25	10
KP1533	4	50	2	8	15	2	20
KP1530	1,5	200	22	33	—	—	40

При підвищенні ємності навантаження  $C_H$  підвищується  $t_{3,PC}$ . Вказана величина  $C_H$  може бути забезпечена при виконанні технологічних вимог виготовлення друкованих плат і при вказаній величині коефіцієнта розширення по виходу (коефіцієнт розширення по виходу інформує, скільки входів мікросхем цієї ж серії допустимо підключати до виходу).

У табл. 2.2 приводяться пояснення основних параметрів для ЛЕ ТТЛ, а у табл. 2.3 – їх числові значення для трьох основних серій.

Параметри, приведені у табл. 2.3, відносяться до ЛЕ **I-III** і дозволяють зробити порівняльну оцінку серій на етапі вибору елементної бази.

Таблиця 2.2

Позначення параметра	Назва	Пояснення
$U_{\text{ВХ}}^1$	Вхідна напруга високого рівня	Значення вхідної напруги високого рівня на вході ІС
$U_{\text{ВХ}}^0$	Вхідна напруга низького рівня	Значення вхідної напруги низького рівня на вході ІС
$I_{\text{ВХ}}^1$	Вхідний струм високого рівня	Значення вхідного струму при напрузі високого рівня на вході
$I_{\text{ВХ}}^0$	Вхідний струм низького рівня	Значення вхідного струму при напрузі низького рівня на вході
$U_{\text{ВИХ}}^1$	Вихідна напруга високого рівня	Значення вихідної напруги високого рівня на виході ІС
$U_{\text{ВИХ}}^0$	Вихідна напруга низького рівня	Значення вихідної напруги низького рівня на виході ІС
$I_{\text{ВИХ}}^1$	Вихідний струм високого рівня	Значення вихідного струму при напрузі високого рівня на виході
$I_{\text{ВИХ}(Z)}^0$	Вихідний струм в Z-стані ІС	Значення вихідного струму при Z-стані ІС
$I_{\text{ВИХ}}^0$	Вихідний струм низького рівня	Значення вихідного струму при напрузі низького рівня на виході ІС

Таблиця 2.3

Параметр	K555		K531		KP1531		KP1533	
	Значення параметра							
	Min	Max	Min	Max	Min	Max	Min	Max
$U_{\text{ВХ}}, \text{В}$	2		2				2	
$U_{\text{ВХ}}^0, \text{В}$		0.8		0.8		0.8		0.8
$I_{\text{ВХ}}^1, \text{мкА}$		20		50		20		20
$I_{\text{ВХ}}^0, \text{мА}$		-0.4		-2.0		-0.6		-0.2
$U_{\text{ВИХ}}^0, \text{В}$		0.5		0.5		0.5		0.4
$U_{\text{ВИХ}}^1, \text{В}$	2.7		2.7		2.7		2.5	
$I_{\text{ВИХ}}^0, \text{мА}$		8		20		20		4
$I_{\text{ВИХ}}^1, \text{мА}$		-0.4		-1		-1		-0.4

У зв'язку з тим, що заводи-виробники ІС держав СНД (наприклад, ВО «Інтеграл») переходять на міжнародні умовні зображення, у табл. 2.4 приводиться відповідність між умовними позначеннями ІС одного типу технологій і ідентичних технічних характеристик (аналогами).

Таблиця 2.4

Вітчизняні	Міжнародні	Вітчизняні	Міжнародні	Вітчизняні	Міжнародні
155	SN74	533	SN54LS	1530	SN54AS
158	SN74L	555	SN74LS	KP1530	SN74AS
530	SN54S	1533	SN54ALS	1531	SN54F
531	SN74S	KP1533	SN74ALS	KP1531	SN74F

**Коротка характеристика зарубіжних аналогів.** Першими з ТТЛ-серій з'явилися мікросхеми 74Н (High speed TTL) і 74L (Low power TTL). Ці серії мікросхем мали майже однакову схемотехніку і відрізнялись лише номіналами опорів: у серії 74Н величини опорів менші, і тому меншою є тривалість часових затримок.

У 70-х роках з'явилися транзистори Шоткі і, відповідно, ТТЛ-мікросхеми з транзисторами Шоткі – 74S (Schottky TTL) і 74LS (Low power Schottky TTL). Подальше вдосконалення технологій і схемотехніки привело до появи серій 74AS (Advanced Schottky TTL) і 74ALS. Серія 74F (Fast TTL) є розвитком попередніх серій з більшою швидкістю.

У табл. 2.5 приводяться основні характеристики перелічених серій ТТЛ. Таблиця дає можливість порівняти експлуатаційні параметри ТТЛ мікросхем різних серій і використовувати їх при проектуванні пристроїв цифрової схемотехніки.

Таблиця 2. 5

Параметри мікросхем	Позначення параметра (англомовне)	Серія				
		74S	74LS	74AS	74ALS	74F
Типова затримка (нС)	$t_{3,PC}$	3	9	1.7	4	3
Споживана потужність на елемент (мВт)		19	2	8	1.2	4
Робота перемикачів (пДж)		57	18	13.6	4.8	12
Вхідна напруга низького рівня (В)	$U^0_{ВХ.МАКС.} (U_{ILmax})$	0.8	0.8	0.8	0.8	0.8
Вихідна напруга низького рівня (В)	$U^0_{ВИХ.МАКС.} (U_{OLmax})$	0.5	0.5	0.5	0.5	0.5
Вхідна напруга високого рівня (В)	$U^1_{ВХ.МИН.} (U_{IHmin})$	2.0	2.0	2.0	2.0	2.0
Вихідна напруга високого рівня (В)	$U^1_{ВИХ.МИН.} (U_{OHmin})$	2.7	2.7	2.7	2.7	2.7
Вхідний струм низького рівня (мА)	$I^0_{ВХ.МАКС.} (I_{ILmax})$	-2.0	-0.4	-0.5	-0.2	-0.6
Вихідний струм низького рівня (мА)	$I^0_{ВИХ.МАКС.} (I_{OLmax})$	20	8	20	8	20
Вхідний струм високого рівня (мкА)	$I^1_{ВХ.МАКС.} (I_{IHmax})$	50	20	20	20	20
Вихідний струм високого рівня (мА)	$I^1_{ВИХ.МАКС.} (I_{OHmax})$	-1	-0.4	-2	-0.4	-1



### 2.3.2. Особливості використання мікросхем ТТЛ

Якість роботи будь-якої цифрової схеми повністю залежить від того, наскільки точно дотримуються технічні умови її використання. Тому важливо знати ряд специфічних особливостей, які не завжди безпосередньо витікають із схемотехніки, а більш є надбанням практики їх використання.

Характерною особливістю ТТЛ ІС виступає наявність значних імпульсів струму живлення, які з'являються при зміні станів вихідного каскаду базового логічного елемента, коли один з транзисторів ще не закрився, а інший вже відкрився. Імпульсні струми небезпечні тим, що, протікаючи по шині живлення, вони створюють імпульсні завади. Для захисту від них на шині живлення використовують конденсатори. Рекомендується для кожного корпусу мікросхеми встановлювати високочастотний конденсатор ємністю 1...2 нФ. У цифрових схемах для забезпечення якісного живлення встановлюються також електролітичні конденсатори з розрахунку не менше 1 мкФ на 5 корпусів ТТЛ.

З ростом робочих частот перемикання ІС зростає не тільки частота імпульсів струму споживання, а також збільшується середня потужність споживання, яка досягає максимального значення на частоті  $f = t_{з.р.с.}^{-1}$ . На величину потужності споживання мікросхеми впливає також ємність навантаження, збільшення якої призводить до підвищення вихідних струмів.

*Завадостійкість ТТЛ ІС визначається конкретними значеннями та співвідношеннями напруг на входах і виходах при їх постійному з'єднанні, що ілюструється на рис. 2.17. Приведені цифри характеризують найбільш несприятливий випадок з точки зору температурних умов та живлення. Вихідний рівень напруги логічного нуля не перевищує 0,4 В, а логічної одиниці не знижується нижче 2,4 В.*

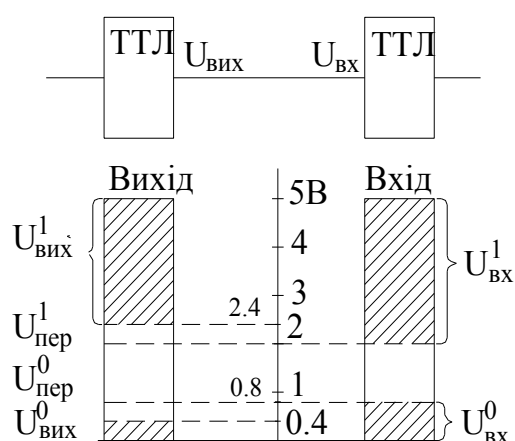


Рис. 2. 17

У той же час, мікросхема буде нормально працювати, якщо на її вході рівень логічного “0” досягне 0,8 В, а рівень логічної “1” знизиться до 2 В. Гарантований запас завадостійкості в обох випадках становить 0,4 В. Статична завадостійкість на низькому рівні, як витікає з рис. 2.17, визначається різницею  $U^0_{ЗАВ} = |U^0_{ВИХ.МАКС} - U^0_{ВХ.МАКС}|$ , на високому рівні :

$$U^1_{ЗАВ} = |U^1_{ВИХ.МИН} - U^1_{ВХ.МИН}|.$$

З приведених формул витікає, що фактично *реальну величину завадостійкості визначають максимальні та мінімальні значення напруг логічних елементів.*

Реальна величина запасу завадостійкості перевищує 1 В. Порогова вхідна напруга, при якій відбувається зміна стану мікросхеми, досягає 1,3...1,4 В при кімнатній температурі, а типові значення вихідних напруг дорівнюють 0,2 і 3 В відповідно для рівнів “0” і “1”. Тому, якщо завада буде діяти на з’єднуючий мікросхеми інформаційний провідник у ситуації, коли вихід попередньої мікросхеми знаходиться в стані “0”, то наступна не реагуватиме на заваду з амплітудою до 1,1...1,2 В. Аналогічно, у стані логічної “1” попередньої мікросхеми наступна буде стійкою до завад на шині живлення з амплітудами до 1,5...1,6 В.

Ще одна особливість ТТЛ полягає в недопустимості з’єднання виходів декількох елементів. Це пов’язано з тим, що можлива ситуація, коли у відповідності з вхідними логічними сигналами на виході одного з них повинна з’явитись логічна “1”, а на виході другого – логічний “0”. При такій комбінації сигналів з’явиться прохідний струм, на який не розраховані елементи. Якщо ж все-таки така необхідність існує, наприклад, для підвищення вихідного струму, то в таких елементах необхідно з’єднувати і їх входи.

Проте у ряді цифрових пристроїв, де кілька вузлів чи блоків працюють на загальне навантаження, таке з’єднання принципово необхідне. У таких випадках використовуються елементи ТТЛ, які поряд з двома звичайними станами мають третій – так званий *Z-стан* (рис. 2.18).

При подачі на вхід  $Z$  сигналу логічної “1” на виході DD1 встановиться низький рівень потенціалу, при якому діод VD1 катодом замикається на загальну шину. При цьому колектор VT1 матиме потенціал, близький до нуля. При наявності сигналу логічного нуля на одному з входів елемента на резисторі  $R_4$  також не буде надіння напруги. Це означає, що базові струми

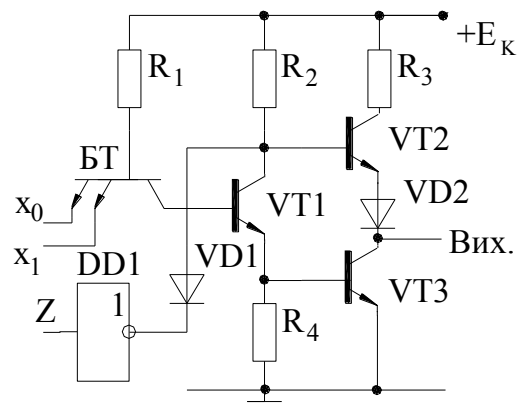


Рис. 2. 18

транзисторів VT2 та VT3 матимуть нульові значення і транзистори знаходяться в закритому стані. Вихід мікросхеми перебуватиме у так званому “висячому”, або  $Z$ -стані. Переведення мікросхеми у  $Z$ -стан означає, що шини, до яких приєднаний її вихід, можуть вільно використовуватись іншими інформаційними пристроями. Але слід пам’ятати, що при  $Z = 0$  залишаються всі проблеми паралельного з’єднання виходів, тому при використанні таких мікросхем необхідно слідкувати, щоб при використанні загальної шини робота мікросхем була рознесена у часі.

Розглянемо тепер особливості мікросхем ТТЛ з точки зору використання їх входів. Нерідко маємо таку ситуацію, коли не всі входи мікросхеми використовуються, і проектуванцю необхідно приймати рішення, що з ними робити. Для прийняття рішення необхідно знати, що одиночний вхід має ємність 1,5...3,5 пФ. Тому підключення великої кількості входів до виходу аналогічної мікросхеми приведе до зростання загальної ємності, що, у свій час, – до підвищення  $t_{з,PC}$  та збільшення потужності енергії, що виділяється на мікросхемі. Тому запаралелення входів мікросхеми не рекомендується як із вказаних причин, так і через те, що кожен вхід додатково навантажує попередню мікросхему. Рекомендується вільні входи підключати до джерела живлення через резистор, величина якого визначається вхідним струмом високого рівня  $I_{ВХ}^1$ . Іноді входи мікросхеми приєднують через резистор  $R_E$  до

загальної шини. В залежності від величини опору резистора, падіння напруги на ньому від вхідного струму  $I_{\text{ВХ}}^0$  може сприйматися мікросхемою як логічний “0”, “1” або може бути створений активний лінійний режим роботи логічного елемента. Величина опору резистора, на якому забезпечується необхідна напруга  $U_{\text{ВХ}}$ , розраховується за формулою:

$$R_E = \frac{U_{\text{ВХ}} R_1}{E_K - U_{\text{БЕ}} - U_{\text{ВХ}}},$$

де  $R_1$  - опір базового резистора вхідного транзистора ЛЕ.

При малих значеннях  $R_E$ , коли падіння напруги на ньому не перевищує рівня логічного нуля, поведінка мікросхеми відповідає сигналу  $U_{\text{ВХ}}^0$ .

Величина  $R_E$  може бути обчислена для конкретних типів мікросхем,  $R_1$  для яких може бути знайдена у довіднику.

Наприклад, для ІС серії К555 величина опору  $R_1 = 20$  кОм. Величина  $U_{\text{ВХ}} < U_{\text{МАКС}}^0 = 0,8$  В (табл. 2.3). При  $U_{\text{БЕ}} = 0,7$  В знаходимо  $R_E \leq 4,5$  кОм.

Для одержання гарантованого високого рівня  $U_{\text{ВХ}} > 2,4$  В потрібно  $R_E \geq 20$  кОм. Проміжні рівні опору можуть забезпечити активний режим логічного елемента.

Резистори на вході мікросхем іноді використовуються для виділення фронтів з вхідних імпульсів за допомогою диференціюючих  $RC$ -компонентів.

Необхідно знати ще про одну особливість використання ТТЛ ІС.

У ряді випадків виникає необхідність керувати за допомогою мікросхем роботою сигнальних ламп, світлодіодів, реле і т.п., які мають або інші робочі напруги, або потужність споживання, яка перевищує потужність сигналів на виходах мікросхем. Для таких випадків використовуються спеціальні мікросхеми, які мають вихідний транзистор з відкритим колектором (рис. 2.19, а). Величина струму колектора  $V_{T2}$ , в залежності від типу використовуваних мікросхем, може мати допустиме значення в діапазоні 20...60 мА, що дає можливість збільшувати потужність навантаження логічних елементів (рис. 2.19, б, в, г).

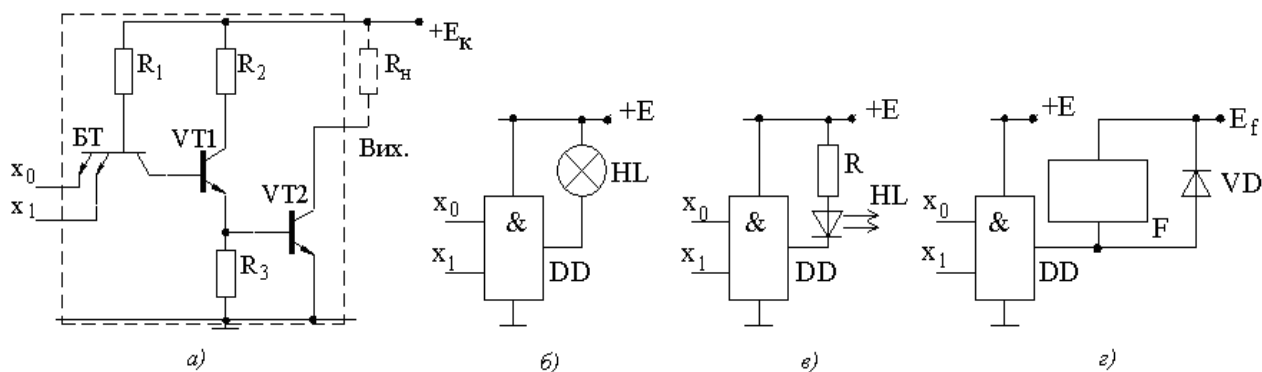


Рис. 2. 19

При реалізації приведених схем необхідно правильно оцінювати значення логічної функції, що реалізується.

Логічний елемент DD реалізує логічну операцію 2І. Але, якщо проектантом закладається умова, що логічна одиниця визначається засвітленням лампи HL, світлодіода HL, або спрацюванням реле F, то у такому випадку приведені схеми реалізують операцію 2І-НІ.

Логічні елементи з відкритим колектором використовуються також для організації “монтажної” логіки. Приклад такої схеми приведений на рис. 2.20, а, а на рис. 2.20, б представлено умовне зображення монтажної логіки в електронних схемах. Ромбик є знаком того, що йдеться не про реальну мікросхему, а про спосіб з’єднання елементів.

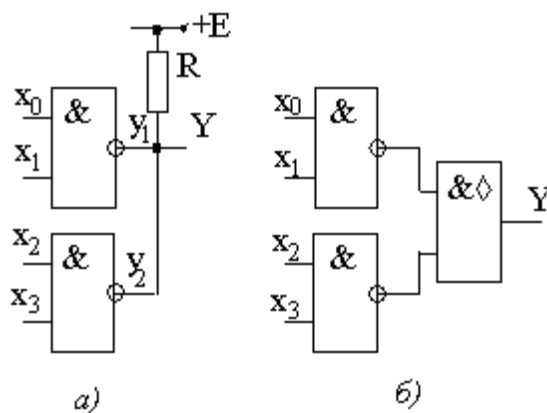


Рис. 2. 20

Для забезпечення  $Y = 1$  необхідно, щоб  $y_1 = y_2 = 1$ , тобто  $Y = y_1 \& y_2$ .

Враховуючи, що  $y_1 = \overline{x_0 \cdot x_1}$ ;  $y_2 = \overline{x_2 \cdot x_3}$ , знаходимо:

$$Y = \overline{x_0 \cdot x_1 \cdot x_2 \cdot x_3} \quad (2.19)$$

Застосовуючи до (2.19) теорему де Моргана, одержуємо остаточну формулу:  $Y = \overline{x_0 \cdot x_1 + x_2 \cdot x_3}$ .

З даної формули витікає, що “монтажне” з’єднання елементів з відкритим колектором функціонує подібно логічному елементу **I-АБО-НІ**.

При використанні “монтажної” логіки необхідно стежити, щоб рівні логічної одиниці  $U^1_{\text{ВИХ}}$  і нуля  $U^0_{\text{ВИХ}}$  не виходили за межі використовуваних стандартних елементів. Оскільки ці рівні залежать від  $E$  та  $R$  (рис.2.20,а), то величина останнього не може бути довільною. Опір резистора  $R$  впливає також на величину потужності, що виділяється на вихідному резисторі; на величину затримки в схемі; на навантажувальну здатність. Тому величину опору резистора  $R$  вибирають в обмеженому діапазоні, мінімальне і максимальне значення якого знаходиться за формулами:

$$R_{\text{МІН}} = \frac{E - U^0_{\text{ВИХ}}}{I^0_{\text{ВИХ}} - K_{\text{ВХ}} I^0_{\text{ВХ}}}; \quad R_{\text{МАКС}} = \frac{E - U^1_{\text{ВИХ}}}{K_{\text{ВИХ}} I^1_{\text{ВИХ}} - K_{\text{ВХ}} I^1_{\text{ВХ}}},$$

де  $K_{\text{ВИХ}}$  – кількість об’єднаних виходів;  $K_{\text{ВХ}}$  – кількість підключених входів.

Розрахункові параметри можна взяти з табл. 2.3. Конкретне значення  $R$  вибирається з умови необхідної швидкодії та потужності споживання.

### 2.3.3. Мікросхеми КМОН-структури

**Інвертор КМОН.** Мікросхеми КМОН використовують в якості базового логічного елемента інвертор, повна еквівалентна схема якого приведена на рис. 2.21.

Безпосередньо функцію інвертора, аналогічного рис. 2.11, виконують транзистори VT1 та VT2. У кожному зі станів інвертора один з транзисторів

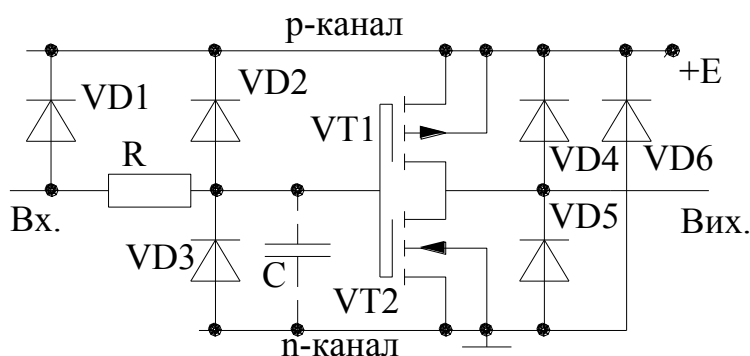


Рис. 2. 21

закритий, і прохідний струм практично дорівнює нулю. Тому  $U^1_{\text{ВИХ}} \approx E$ ;  $U^0_{\text{ВИХ}} \approx 0$ . Потужність споживання також практично дорівнює нулю, адже вхідний опір наступної мікросхеми (як навантаження

інвертора) визначається величиною  $10^{12} \dots 10^{14}$  Ом. Характерною особливістю КМОН-інверторів є виключно висока температурна стабільність. Коливання температури в межах  $-55 \dots +125^\circ\text{C}$  приводять до зміни параметрів ключа не більш ніж на 5%, тоді як вказані зміни температури приводять до зміни параметрів ТТЛ більш ніж на 40%.

Суттєвий недолік КМОН-інверторів полягає в їх надзвичайній “чутливості” до статичної електрики. Фізично це явище пояснюється тим, що затвор розміщується на дуже тонкому шарі діелектрика (товщиною близько 1 мкм), який одночасно виступає ізоляційним матеріалом у паразитній ємності  $C$  (рис. 2.21) між затвором та підкладкою.

Навіть невелика за величиною паразитна ємність ( $C = 2 \dots 15$  пф у залежності від серії ЛЕ) при дуже великих значеннях опору діелектрика, приблизно  $10^{12}$  Ом, створює умови для сприйняття та накопичення статичних зарядів, які можуть виникати на тілі, або одязі людини, зовнішніх приладах, тощо. Діоди на вході та виході ключа призначені для захисту інвертора від статичної електрики. Діоди VD1...VD3 захищають ізоляцію затвору від пробоя. Діод VD1 (лавинного типу) має пробивну напругу 25 В, а VD2 і VD3 – 50 В. Захисний резистор  $R = 200$  Ом – 2 кОм не допускає швидких змін напруги на конденсаторі  $C$ , захищаючи тим самим попередній каскад від імпульсних навантажень. Діоди VD4...VD6 захищають вихід інвертора від можливого пробоя. Діод VD4 має пробивну напругу 50 В, VD5 – 25 В. Діод VD6 захищає ключ від можливої зміни полярності напруги живлення.

Охоронні діоди VD2, VD3 разом з діодами VD4, VD5 створюють діодний міст, у діагоналі якого знаходиться джерело живлення. Всі діоди перебуватимуть у закритому стані до того часу, поки напруга на вході не перестане задовольняти умові:

$$-0,7 \text{ В} \leq U_{\text{вх}} \leq E + 0,7 \text{ В}. \quad (2.20)$$

Якщо вхідна напруга виходить за межі, визначені (2.20), то при низькому внутрішньому опорі джерела живлення струм через охоронні діоди може стати

більшим максимально допустимого  $I_{\text{МАКС.Д}} = 10 \text{ мА}$ , що призведе до їх виходу з ладу. Тому реально використовується обмеження величини струму на рівні  $1 \dots 2 \text{ мА}$ , що забезпечується установкою зовнішнього резистора  $R_3$ . При цьому зростає постійна часу заряду вхідного конденсатора  $\tau = (R_3 + R) C$ , яка впливає на швидкість перемикання ключа.

У ряді спеціальних схем (564ЛН2 – шість буферних інверторів, аналоги провідних західних фірм-виробників 4069ВРС, НСF4069UVC1, ...; 564ПУ2 – шість перетворювачів рівня, відповідні аналоги провідних західних фірм 4050ВDC, НСС4050ВD, ...) передбачена схема захисту, що допускає перевищення вхідної напруги над напругою живлення.

Існує ще одна причина, яка вимагає уважного відношення до напруги живлення. Не вдаючись у деталі побудови структури інвертора, зазначимо, що по відношенню до напруги живлення вона представляє собою структуру, подібну до тиристора, підкладки якої виконують функції керуючих електродів.

При високих напругах живлення ймовірність того, що вказаний тиристор спрацює, значно зростає, що може призвести до виникнення значних струмів через мікросхему, і можливе її пошкодження. Розглянута структура може спрацьовувати при наявності в пристроях, що її використовують, джерел високочастотних імпульсів. Тому при включенні апаратури рекомендується напругу живлення подавати раніше, ніж керуючі сигнали, а виключати – в зворотньому порядку.

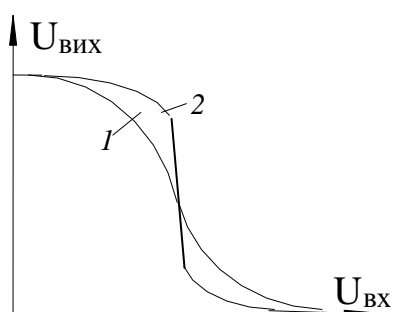


Рис. 2. 22

У динамічних режимах ключ проходить через стан, коли обидва транзистори відкриті. При цьому в мікросхемі виникає імпульс прохідного струму. Послідовність імпульсів прохідного струму створюватиме динамічний струм ключа і мікросхеми. Його величина залежить від тривалості фронтів вхідних імпульсів, швидкодії самого ключа, частоти



вхідних імпульсів. Для зниження величини динамічних струмів, зниження величини  $t_3$ , а також зниження величини опору каналу відкритого транзистора в ряді мікросхем після описаного ключа встановлюють допоміжний буферний двохкаскадний підсилювач, зібраний за такою ж схемою.

На рис. 2.22 приведені передавальні характеристики інвертора без буферного підсилювача (крива 1) та з ним (крива 2), які демонструють значне скорочення часу перемикавання. Використання буферного підсилювача дозволяє формувати вихідні імпульси з крутими фронтами незалежно від характеру вхідного сигналу. Суттєво знижується також величина опору каналу відкритого транзистора (опір р-канального транзистора знижується в сучасних мікросхемах до 200 Ом, а n-канального – до 100 Ом).

Буферні каскади використовуються здебільшого як вихідні каскади мікросхем для підвищення потужності вихідного сигналу. Однак це приводить до того, що вони займають значну площу на кристалі і знижують рівень інтеграції мікросхем. У той же час, слід пам'ятати, що висока крутизна фронтів може призводити до ударного збудження затухаючих коливань у довгих лініях, які з'являються за рахунок розподілених реактивних параметрів друкованих плат монтажу.

Мікросхеми КМОН випускаються як з простими, так і з буферними вихідними каскадами. Рекомендується при однакових умовах функціонування віддавати перевагу мікросхемам з буферними каскадами.

**Інвертор КМОН з трьома станами.** Його схема є різновидом звичайних інверторів з допоміжними ключами (VT1 та VT4), які керуються сигналом V (рис.2.23).

При відкритих транзисторах VT1 та VT4 інвертор працює подібно раніше розглянутому. Якщо ж ключі закриті, живлення інвертора відключається і його

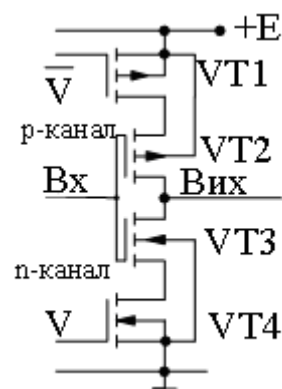


Рис. 2. 23

вихід матиме високий опір по відношенню до шин живлення. Такі інвертори виготовляються як самостійні мікросхеми (наприклад, 546ЛН1 – елемент **БНІ**, аналоги західних фірм-виробників CD4502BD, НСС4502BD, ...) або використовуються у складі мікросхем різного призначення. Враховуючи значну величину вихідного опору, ключ може використовуватись для тривалого зберігання інформації, яка може бути записана в конденсатор, підключений до його виходу.

При використанні мікросхем з трьома станами необхідно враховувати деякі їх особливості роботи в динамічних режимах. Вони проявляються у тому, що в режимі, коли на виході підтримується високий рівень сигналу і одночасно ключ знаходиться у Z-стані, перехід в один з робочих станів (високий або низький рівні виходу) проходить довше, ніж при перемиканні з високого в низький стани безпосередньо. Тому при використанні мікросхем у режимі монтажно-логіки можуть мати місце динамічні струми, що протікатимуть через р- транзистор однієї мікросхеми і n- транзистор іншої. Реально такий струм не призводить до виходу мікросхем з ладу (як у ТТЛ), але він приводить до напружених режимів роботи і зростання рівня завад.

**Двонаправлений ключ.** Цей пристрій, який не має аналогів в мікросхемах інших технологій, складається з двох МОН-транзисторів різного типу провідності, які включені зустрічно-паралельно (рис. 2.24).

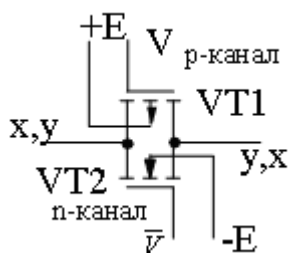


Рис. 2. 24

Керування ними забезпечується взаємоінверсними сигналами  $V$  та  $\bar{V}$ , що подаються на затвори VT1 та VT2. Напряга живлення подається на виводи підкладок:  $+E$  для р- каналного транзистора VT1,  $-E$  – для n- каналного VT2. Двостороння провідність у схемі забезпечується завдяки тому, що транзистори зберігають працездатність, якщо витоки та стоки поміняти місцями.

Паралельне з'єднання транзисторів дозволяє зменшити загальний опір каналів, величина кожного з яких у відкритому стані має величину 100 Ом...1 кОм.

Відкритий стан обох транзисторів забезпечується при умові, коли потенціал затвору транзистора VT2 n- типу наближається до  $+E$ , а затвору транзистора VT1 до  $-E$ . При цьому напруга, що підводиться до інформаційних входів-виходів  $x, y$ , завжди повинна мати позитивний потенціал відносно підкладки n- каналного транзистора, тобто відносно шини  $-E$ , і негативний по відношенню до підкладки p- каналного ( $+E$ ).

Завдяки зустрічно-паралельному з'єднанню транзисторів VT1 та VT2 інформаційні сигнали на виході ключа не залежать від порогів їх відкриття і

можуть змінюватися у межах напруги живлення. Величина опору каналу залежить від величини вхідної напруги, різниці потенціалів між підкладками та величини навантаження і в діапазоні частот роботи транзисторів визначається тільки активною складовою. Ця властивість дає можливість використовувати ключі для комутації як аналогових, так і дискретних сигналів.

Графік залежності опору каналу від величини вхідної напруги для VT1 (крива 1) та VT2 (крива 2) приведені на рис. 2.25.

Крива 3 визначає загальний опір двонаправленого ключа. При негативних вхідних напругах опір ключа визначається відкритим транзистором VT2, при позитивних – відкритим транзистором VT1, а при напругах, близьких до нульового рівня – результуючим значенням частково відкритими двома транзисторами.

У реальних схемах ключів крива 2 зміщується вправо за рахунок керування потенціалом його підкладки. В результаті величина опору каналу

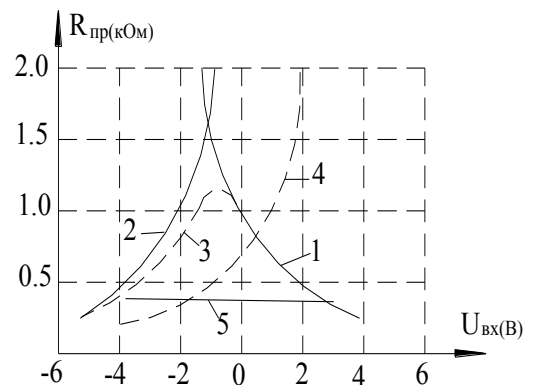


Рис. 2. 25

відповідає кривій 4, а результуючий опір ключа (крива 5) майже не залежить від величини  $U_{ВХ}$ .

Якщо напруги на входах  $V$  змінюють своє значення на протилежне, транзистори закриваються, і величина опору між інформаційними входами/виходами  $x, y$  перевищує  $10^9$  Ом.

На рис. 2.26, а приведена принципова схема реального ключа, виконаного на транзисторах VT3, VT4.

Взаємно інверсні сигнали  $V, \bar{V}$  на його керуючих входах формуються за допомогою інвертора, виготовленого на VT1, VT2. Така структура ключа використовується в багатьох серіях мікросхем, причому вона організована так, що при  $V = U^1$  ключ відкритий, а при  $V = U^0$  – закритий.

На рис. 2.26, б приводиться один з варіантів графічного зображення двонаправлених ключів на принципових схемах.

**Експлуатаційні характеристики КМОН-ключів.** Для ІС КМОН типу характерні великі входні опори ( $\approx 10^{12}$  Ом), завдяки чому входні струми дуже малі. Тому вважається, що КМОН ІС по входах керуються напругами. Друга особливість полягає в тому, що ключі в статичних режимах майже не споживають струм від джерела живлення. Тому електричні параметри ІС мають

свої особливості, які розглянемо більш детально.

**Вхідний струм**, як і для ТТЛ ІС, має два значення –  $I^0_{ВХ}$  та  $I^1_{ВХ}$ . Найбільший вплив на нього має захисна діодна ланка, але, незважаючи на це, його величина при нормальних експлуатаційних режимах не перевищує 0,05 мкА.

Величина **вхідної напруги** визначається рівнем логічної “1” та

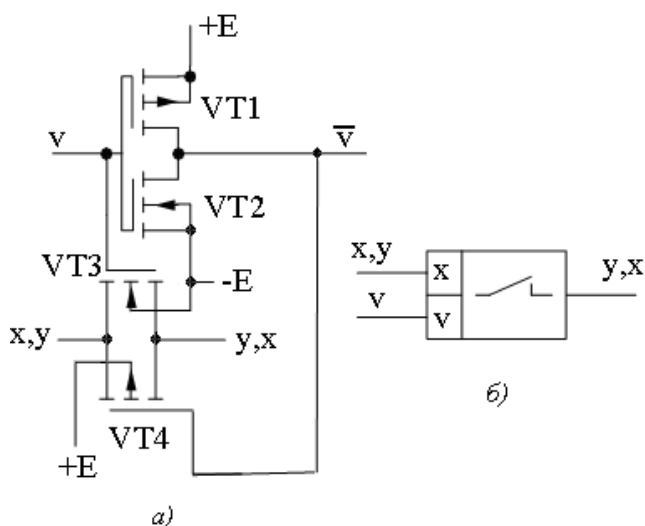


Рис. 2. 26

“0”, а також обмежується необхідними рівнями завадостійкості. Враховуючи, що для КМОН ІС напруга живлення може задаватись у межах 3...15 В, відповідно може змінюватися і рівень вхідної напруги. У той же час, недопустимо, щоб її величина перевищувала напругу джерела живлення  $E$  більш ніж на 0,2 В (за винятком мікросхем-перетворювачів рівнів) і була меншою ніж -0,2 В. Таке обмеження обумовлене можливістю відкриття зворотного діодного мосту, що створений захисними колами. Необхідно також слідкувати, щоб тривалість фронтів вхідних імпульсів була обмеженою, адже протягом тривалості фронту обидва транзистори ключа будуть знаходитися в активному режимі, що призведе до появи значних прохідних струмів. З цієї ж причини недопустимо, щоб входи мікросхем залишалися не приєднаними, оскільки потенціали, що можуть навестись на них, приведуть до аналогічного ефекту.

Вказані вимоги щодо вхідних напруг необхідно враховувати при побудові схем імпульсних генераторів, одновібраторів та ін., в яких вхідні напруги можуть змінюватись у широких межах.

**Вихідна напруга**, за аналогією з вхідною, може приймати значення  $U^1_{\text{вих}}$  та  $U^0_{\text{вих}}$ . Вони відрізняються від нуля і напруги живлення при нормальних експлуатаційних умовах не більше ніж на 10 мВ у режимі холостого ходу мікросхеми. При навантаженні мікросхеми номінальним струмом рівні  $U^1_{\text{вих}}$  та  $U^0_{\text{вих}}$  змінюються і відрізняються від логічних рівнів холостого ходу на 0,5...2,8 В.

КМОН ІС розробки 80-х і 90-х років здебільшого виготовляються з урахуванням необхідності узгодження їх за рівнями сигналів з ТТЛ ІС. Тому їх напруги живлення або мають фіксовані значення, або мають можливість встановлюватись у більш низькому

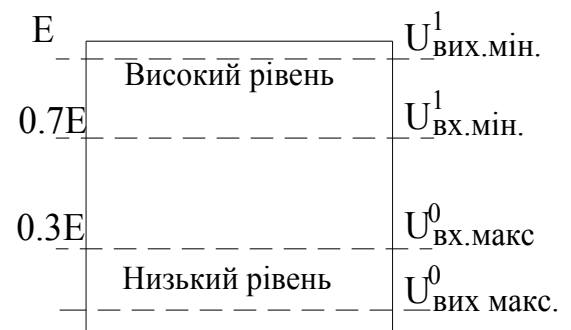


Рис. 2. 27

діапазоні (наприклад, 1,5...6 В). До таких мікросхем відноситься, наприклад, НС-серія (Texas Instrument Co.).

Вихідні та вхідні рівні таких мікросхем пов'язані з напругою живлення досить жорсткими співвідношеннями, які зображені на діаграмі (рис. 2.27).

Параметри КМОН ІС визначаються такими напругами:

$U^1_{\text{ВІХ.МІН}}$  – мінімальна вихідна напруга високого рівня (для мікросхем з  $E = 5 \text{ В}$   $U^1_{\text{ВІХ.МІН}} = 4,9 \text{ В}$ );

$U^1_{\text{ВХ.МІН}}$  – мінімальна вхідна напруга, яка гарантовано сприймається як логічна “1”;

$U^0_{\text{ВХ.МАКС}}$  – максимальна вхідна напруга, яка гарантовано сприймається як логічний “0”;

$U^0_{\text{ВІХ.МАКС}}$  – максимальна вихідна напруга низького рівня (для мікросхем з  $E = 5 \text{ В}$   $U^0_{\text{ВІХ.МАКС}} = 0,1 \text{ В}$ ).

Приведені параметри справедливі лише для КМОН МС, які керуються і керують мікросхемами тієї ж серії.

Виходячи із вказаних граничних параметрів, можна сформулювати наступний висновок. Оскільки  $U^0_{\text{ВХ.МАКС}} = 1,5 \text{ В}$  перевищує величину  $U^0_{\text{ВІХ.МАКС}} = 0,1 \text{ В}$  на 1,4 В, то ця різниця представляє собою запас завадостійкості по постійному струму. Аналогічна картина має місце і при високому рівні вихідного сигналу.

**Вихідний струм** характеризує навантажувальну здатність ІС по постійному струму. Струм  $I^0_{\text{ВІХ}}$  є вхідним при відкритому n- канальному та закритому р- канальному транзисторах. Обидва струми для різних типів мікросхем відрізняються в десятки разів і, у той же час, залежать від напруги живлення ІС. Як приклад, нижче приведені величини вихідних струмів мікросхеми 561ЛА8, яка характеризується підвищеною навантажувальною здатністю:

$$I^0_{\text{ВІХ}} = 0,12 \text{ мА}; \quad I^1_{\text{ВІХ}} = 0,22 \text{ мА} \quad \text{при } E = 5 \text{ В};$$

$$I^0_{\text{ВІХ}} = 0,25 \text{ мА}; \quad I^1_{\text{ВІХ}} = 0,55 \text{ мА} \quad \text{при } E = 10 \text{ В}.$$

Порівняння вхідних та вихідних струмів дозволяє зробити висновок, що такий параметр, як коефіцієнт розгалуження по виходу, для КМОН ІС має чисто умовне значення. У даному випадку він буде обмежуватись не співвідношенням між величинами вихідного та вхідного струмів, а скоріше допустимою ємністю навантаження.

**Особливість роботи КМОН ІС з резистивним навантаженням.** У тих випадках, коли мікросхеми КМОН приєднуються до входів ТТЛ ІС або інших резистивних навантажень, їх робота дещо відрізняється від ідеального режиму.

У будь-якому стані КМОН вихідний транзистор, що знаходиться у включеному стані, має опір, величина якого співрозмірна з відповідним значенням опору навантаження. Протікання струму через опори відкритих транзисторів буде приводити до появи падіння напруги на них, в результаті чого рівні логічних "0" і "1" можуть не відповідати вказаним вище величинам. Визначити необхідні обмеження, які накладаються на роботу КМОН ІС зовнішнім навантаженням, можна на основі резистивної моделі ключа з зовнішнім навантаженням, яка приведена на рис. 2.28, а.

Транзистори КМОН ключа представлені активними опорами  $R_p$  і  $R_n$ .

При кожному з двох можливих станів один транзистор має високий опір (більше 1 МОм), інший – низький. Для сучасних КМОН ІС нижня величина опору резистора  $R_p$  має значення близько 200 Ом, а  $R_n$  – близько 100 Ом. Навантаження моделюється двома резисторами  $R_{H1}$  і  $R_{H2}$ , приєднаними до шин живлення.

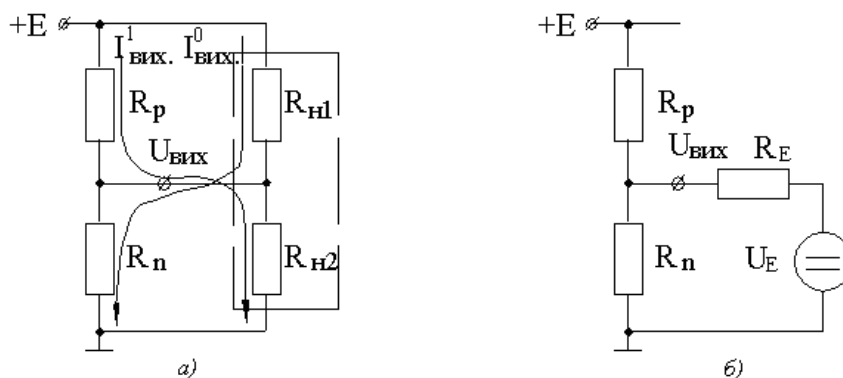


Рис. 2.28

**Приклад 2.3.** Визначити величини напруг  $U^0_{\text{ВИХ}}$  і  $U^1_{\text{ВИХ}}$  на виході мікросхеми при  $R_{Н1} = 1 \text{ кОм}$ ;  $R_{Н2} = 2 \text{ кОм}$ .

*Розв'язання.* Приведена схема заміщення може бути зображена у вигляді еквівалентної схеми, що приведена на рис. 2.28, б, де  $R_E$  – еквівалентний опір, а  $U_E$  – напруга еквівалентного генератора, яка дорівнює для даної схеми:

$$U_E = E \cdot R_{Н2} / (R_{Н1} + R_{Н2}) = 5 \cdot 2 \cdot 10^3 / (2+1) \cdot 10^3 \approx 3,33 \text{ В.}$$

Струм короткого замикання:  $E/R_{Н1} = 5/10^3 = 5 \text{ мА.}$

Еквівалентний опір двополюсника:  $R_E = U_E / I_K = 3,33 / 5 \cdot 10^{-3} \approx 667 \text{ Ом.}$

Якщо КМОН-ключ на своєму виході має низький рівень, вихідна напруга може бути обчислена за формулою:  $U^0_{\text{ВИХ}} = U_E \cdot R_n / (R_E + R_n) \approx 3,33 \cdot 100 / (667+100) \approx 0,43 \text{ В.}$

Аналогічно, при високому вихідному рівні маємо:

$$U^1_{\text{ВИХ}} = U_E + (E - U_E) \cdot R_E / (R_p + R_E) = 3,33 + (5 - 3,33) \cdot 667 / (200 + 667) \approx 4,61 \text{ В.}$$

Реально в практиці проектування електронних схем немає можливості визначити величини  $R_p$  і  $R_n$  і, відповідно, рівні вихідної напруги. Замість цього розробники задають максимальну величину навантаження в кожному із станів інвертора і гарантують у найгіршому випадку необхідну напругу на навантаженні. Навантаження визначається величинами струму:

$I^0_{\text{ВИХ.МАКС}}$  – максимальний струм, який вихід мікросхеми може сприймати в стані лог. „0”, при якому забезпечується вихідна напруга, не більша ніж  $U^0_{\text{ВИХ.МАКС}}$ ;

$I^1_{\text{ВИХ.МАКС}}$  – максимальний струм, який вихід може генерувати у стані лог. „1” при забезпеченні мінімальної вихідної напруги, не меншої ніж  $U^1_{\text{ВИХ.МІН}}$ .

Оцінити величини  $R_p$  і  $R_n$  можна приблизно за паспортними параметрами мікросхеми:

$$R_p = (E - U^1_{\text{ВИХ.МІН}}) / I^1_{\text{ВИХ.МАКС}}; \quad R_n = U^0_{\text{ВИХ.МАКС}} / I^0_{\text{ВИХ.МАКС}}.$$

Розглянуті особливості роботи КМОН ІС справедливі, якщо рівні вхідних сигналів відповідають паспортним значенням. Якщо ж вхідна напруга не відповідає рівням живлення, то транзистори ЛЕ можуть неповністю відкриватись і закриватись. Тому в закритому стані опір транзистора може бути



меншим, ніж 1 МОм, а у відкритому – для  $R_p$  більшим за 200 Ом, для  $R_n$  більшим за 100 Ом. Це приводить до зростання падіння напруги на них і, відповідно, стандартні рівні вихідної напруги можуть вийти за допустимі межі. Таких проблем не виникає у схемах, де використовуються лише КМОН ІС, але при сумісному використанні ТТЛ-КМОН мікросхем такі ситуації необхідно передбачати і не допускати.

**Часові параметри** визначаються тривалістю фронтів  $t^{01}$ ,  $t^{10}$  перемикання ключа, які, в свою чергу, майже повністю залежать від величин паразитних ємностей. Найбільший вплив серед них мають:

- ємності вхідних кіл, які для типових КМОН ІС можуть мати величини 2...10 пФ;
- ємності з'єднуючих провідників, що характеризуються величинами до 0,4 пФ на 1 сантиметр довжини провідника.

Визначити тривалість фронтів можливо за допомогою аналізу перехідних процесів перемикання ключа. Останній проводиться на основі схеми заміщення інвертора, що приведена на рис. 2.29, в якій навантаження моделюється елементами  $R_E$ ,  $U_E$  і  $C$ .

Розглянемо процес перемикання для випадку, коли КМОН-ключ працює в оточенні аналогічних мікросхем. Це дає право вважати, що  $R_E = \infty$  і  $U_E = 0$ . Перехідні процеси для оцінки тривалості фронтів розглядаються при умові миттєвого перемикання ключа з одного стану в інший.

Якщо ключ знаходиться в стані забезпечення на виході лог. „1”, то конденсатор  $C$  буде знаходитись у зарядженому стані з напругою  $E$ . При переході ключа зі стану лог. „1” до стану лог. „0” конденсатор  $C$  почне розряджатись на резистор  $R_n$  згідно закону:

$$U_C = E \exp\left(-\frac{t}{R_n C}\right).$$

Якщо прийняти  $E = 5$  В;  $R_n = 100$  Ом;  $C = 100$  пФ, а напругу низького порогового рівня  $U_{ПН} = 0,3 E = 1,5$  В.

Час перемикання матиме величину:

$$t_C = t^{1,0} = R_n C \ln\left(\frac{U_{\text{пн}}}{E}\right) = 100 \cdot 10^{-10} \ln\left(\frac{1,5}{5}\right) \approx 3,6 \text{ нс}.$$

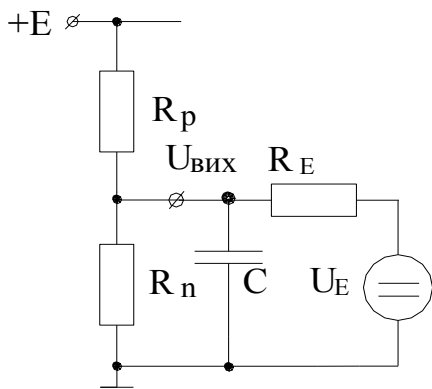


Рис. 2. 29

При переході ключа з низького вихідного рівня до високого конденсатор  $C$  заряджатиметься від джерела живлення, і напруга на ньому змінюватиметься за законом:

$$U_C = E \left[ 1 - \exp\left(-\frac{t}{R_p C}\right) \right],$$

тому тривалість фронту, від нульової напруги до  $0,7 E$  при  $R_p = 200 \text{ Ом}$  і  $C = 100 \text{ пФ}$ , буде:

$$t_{\Phi} = t^{0,1} = R_p C \ln\left(\frac{E - 0,7E}{E}\right) = 200 \cdot 10^{-10} \ln\left(\frac{5 - 3,5}{6}\right) \approx 7,2 \text{ нс}.$$

Якщо оцінювати тривалості  $t_{\Phi}$  і  $t_C$  не за пороговими рівнями, а за максимальними значеннями, то можна знайти, що  $t_C = 12 \text{ нс}$ , а  $t_{\Phi} = 24 \text{ нс}$ , що дає можливість зробити приблизну оцінку тривалості фронту та спаду за постійними часу заряду-розряду конденсатора:  $(R_n C) = 10 \text{ нс}$  і  $(R_p C) = 20 \text{ нс}$ .

Зрозуміло, що реальна швидкодія ключа визначатиметься сумарною ємністю навантаження з урахуванням усіх її складових.

Останні серії мікросхем КМОН мають значно вищі частотні межі роботи. Завадостійкість мікросхем КМОН порівняно з іншими значно більша і забезпечується наступними властивостями:

- поріг зміни стану КМОН ІС – приблизно  $0,5 E$ ;
- напруга на виході мікросхеми стану лог. „0”  $U_{\text{вих}}^0 = 0 \text{ В}$ ;
- напруга на виході мікросхеми стану лог. „1”  $U_{\text{вих}}^1 = E$ .

Тому завадостійкість для КМОН ІС практично не залежить від стану попередніх мікросхем і має величину, близьку до  $0,5 E$ . Завдяки високій завадостійкості, яка до того ж не залежить від робочої температури, мікросхеми

КМОН мають значну перевагу перед іншими при роботі з довгими лініями зв'язку та в умовах значних завад.

КМОН ІС довгий час відносились до мікросхем низької швидкодії. Але з появою ІС серій 1564, КР1554, які за своїми частотними властивостями відповідають ТТЛ ІС 1530, КР1530 з середнім часом затримки біля 10 нс, ситуація з їх використанням суттєво змінилась на краще. Наприкінці 80-х – на початку 90-х років минулого століття з'явилися мікросхеми КР1594 з середнім часом затримки, що не перевищував 3,5 нс, і сумісні з ТТЛ ІС за напругою живлення, вхідними і вихідними параметрами. У цих серіях поєднуються КМОН- і ТТЛ-технології, причому вхідні кола будуються на основі ТТЛ-, а вихідні – з використанням КМОН-технологій. При цьому, звичайно, зменшився діапазон напруг живлення (він став ближчим до рівнів ТТЛ), зменшився поріг перемикання, забезпечуючи в той же час більш високу завадостійкість, ніж у ТТЛ ІС.

Звернемось тепер до мікросхем зарубіжного ринку. Сім'я ІС серії 4000 КМОН технології була першою на ринку цифрових ІС. Сучасні мікросхеми мають загальне умовне позначення **74 FАM xx**, де **74** – номер серії; **FАM** (*Family Alphabetic Mnemonic*) – алфавітна мнемоніка сім'ї; **xx** – число, що визначає функціональне призначення. Мікросхеми різних сімейств з однаковим значенням pp мають однакові функції. Наприклад, 74НС30, 74АС30 і 74АСТ30 мають однакове функціональне призначення – восьмивходовий **NAND (І-НІ)**. Використовуються також мікросхеми серії 54, які мають більш широкий температурний діапазон роботи і застосовуються в спеціальних пристроях.

Першими з мікросхем КМОН-технологій, які ввійшли у серію 74, були сімейства НС (High-speed CMOS) і НСТ (... TTL compatible). Мікросхеми серії НС можуть використовувати напругу живлення в діапазоні 2...6 В. Більш висока напруга використовується для підвищення швидкодії мікросхем, а більш низька напруга забезпечує більш економічне електроспоживання. За рівнями вхідних і вихідних сигналів серія НС несумісна з ТТЛ навіть при напрузі

живлення 5 В. Рівні вхідних і вихідних напруг приведені для декількох серій у табл. 2.6.

Таблиця 2. 6

Параметри (англомовне позначення)	Серія					
	НС	НСТ	АС	АСТ	FCT	FCT-T
$U^1_{ВХ.МІН}$ (В) ( $U_{IH\ min}$ )	3,5	2,0	3,85	2,0	2,0	2,0
$U^0_{ВХ.МАКС}$ (В) ( $U_{IL\ max}$ )	1,35	0,8	1,35	0,8	0,8	0,8
$U^1_{ВИХ.МІН}$ (В) ( $U_{OH\ min}$ )	3,84	3,84	3,76	3,76	2,4	2,4
$U^0_{ВИХ.МАКС}$ (В) ( $U_{OL\ max}$ )	0,33	0,33	0,37	0,37	0,5	0,5

На заміну вказаних серій в 80-х роках прийшли більш досконалі серії АС (Advanced CMOS) і АСТ (... TTL compatible). Між собою вони відрізнялись лише по відношенню до ТТЛ. Особливість цих серій МС полягає в тому, що вони симетричні по виходу, тобто споживаний струм по виходу при низькому рівні останнього дорівнює величині генерованого вихідного струму при високому логічному рівні.

Промислові серії ІС орієнтовані на роботу в температурному діапазоні 0...70°C.

У табл. 2.7 приводяться основні технічні характеристики описаних серій, взяті для двох функціонально однакових мікросхем – елемента **2І-НІ** (74xx00) і дешифратора 3×8 (74xx138).

Таблиця 2. 7

Параметри	Мікро- схеми	Умови	Серії					
			НС	НСТ	АС	АСТ	FCT	FCT-T
Типова затримка (нс)	74xx00		18	18	5.25	5.25	-	-
	74xx138		36	36	9.25	10.5	1.5	1.5
Потужність споживання в статичному режимі (мкВт)	74xx00		24	24	30	30	-	-
	74xx138		40	40	40	40	7.5	7.5
	74xx00		24	24	30	30		
	74xx138		85	85	60	60		
Потужність споживання в динамічному режимі (мкВт)	74xx00	1 МГц	0.6	0.6	0.7	0.7		-
	74xx138	1МГц	2.15	2.15	1.5	1.5	0.75	0.75

Величина часової затримки визначається напівсумою тривалості фронту і спаду імпульсу. Приведена в табл. 2.7 величина затримки для дешифратора значно більша, ніж затримка елемента **2І-НІ**, адже це більш складна мікросхема і величина затримки визначається загальною сумою затримок її складових.

Величина динамічної потужності споживання визначена для частоти 1 МГц і може бути використана при оцінці витрат потужності на інших частотах, оскільки вона є пропорційною частоті перемикання логічного елемента. Наприклад, на частоті 100 кГц вона буде у 10 разів меншою, а при 10 МГц – приблизно у 10 разів більшою.

На початку 90-х років з'явилися нові серії КМОН IC FCT (Fast CMOS TTL compatible), які виготовлені на основі більш сучасних технологій (менші габарити інтегральних транзисторів) і забезпечують більшу швидкість порівняно з АСТ, а також мають меншу потужність споживання. Мікросхеми цих серій повністю сумісні з ТТЛ, а тому можуть без ускладнень використовуватись у схемах на основі ТТЛ. Мікросхеми серії FCT-T мають ідентичні з FCT характеристики. Єдина різниця полягає в дещо зменшеній величині вихідного струму, а високий рівень вихідної напруги повністю відповідає рівню ТТЛ IC.

У табл. 2.8. приводяться допоміжні вхідні і вихідні параметри розглянутих серій КМОН IC.

Таблиця 2. 8

Параметри, умовне позначення (англомовне позначення)	Серія					
	НС	НСТ	АС	АСТ	FCT	FCT-T
Вхідний струм $I_{ВХ.МАКС}$ ( $I_{I_{max}}$ ) (мкА)	±1	±1	±1	±1	±0.5	±0.5
Вихідний струм низького рівня $I_{ВХ.МАКС}^0$ ( $I_{OL_{max}}$ ) (мА)	4	4	24	24	48	48
Вихідний струм високого рівня $I_{ВХ.МАКС}^1$ ( $I_{OH_{max}}$ ) (мА)	-4	-4	-24	-24	-15	-8
Типова вхідна ємність $C_{ВХ.}(C_{Intyp})$ (пФ)	6.5	6.5	4.5	4.5	6	6

Відповідність серій КМОН вітчизняних і зарубіжних виробників приводиться у табл. 2.9.

Таблиця 2. 9

Вітчизняні	Міжнародні	Вітчизняні	Міжнародні	Вітчизняні	Міжнародні
561	CD4000A	КР561	CD4000B	КР1594	74АСТ
564	МС14000А	1564	54НС	КР1554	74АС
-	FCT	-	FCT-T		

Розглянуті серії мікросхем є лише незначною часткою з широкого набору серій відомих і маловідомих фірм, що виготовляють і розробляють електронну елементну базу. Нижче приводиться лише частина назв серій цифрових ІС, які виготовляються відомою фірмою Texas Instrument:

АС, АСТ	Advanced CMOS Logic
АНС, АНСТ	Advanced High-Speed CMOS Logic
ALVC	Advanced Low-Voltage CMOS Technology
НС	High-Speed CMOS Logic
НСТ	High-Speed TTL compatible CMOS Logic
LV	Low-Voltage HCMOS Technology
LVC	Low-Voltage CMOS Technology

#### **2.3.4. Особливості експлуатації мікросхем КМОН**

**Реалізація логічних функцій.** Інвертуючі ключі КМОН ІС виступають базовою структурою для побудови логічних елементів виду **I-НІ**, **АБО-НІ** та більш складних.

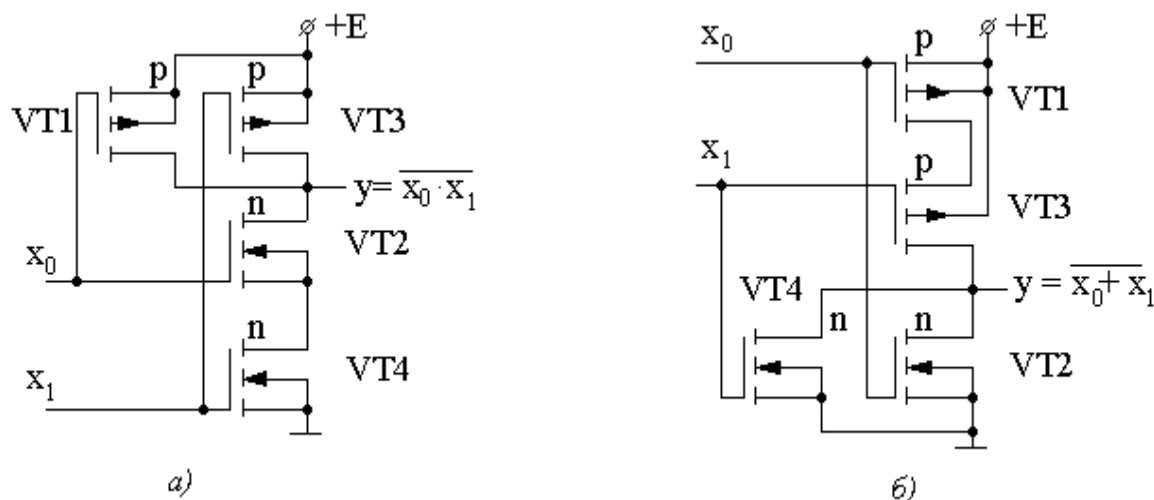


Рис. 2. 30

Організація елементів **I-НІ** забезпечується тим, що нижні транзистори ключа з'єднуються послідовно, а верхні – паралельно (рис. 2.30, *а*). Тобто логічна структура з двома входами будується на двох'ярусній основі. Логічна функція, що реалізується схемою, визначається транзисторами нижнього ярусу. При побудові логічних елементів такого типу послідовно з'єднується декілька транзисторів і, відповідно, зростає їх опір. Це приводить до зростання постійних часу заряду-розряду паразитних ємностей, тобто до погіршення швидкодії.

Щоб запобігти цьому явищу на практиці приходиться збільшувати розміри транзистора (перерізу каналу). Тому для збільшення кількості входів на практиці краще робити каскадне з'єднання декількох більш простих елементів.

Для реалізації операції **АБО** нижні транзистори (VT2, VT4), (рис. 2.30, *б*) з'єднуються паралельно.

Логічні елементи **I-НІ**, **АБО-НІ** з більшою кількістю входів створюються аналогічно.

У сучасних серіях мікросхем елементи **I-НІ**, **АБО-НІ** виготовляються на 2, 3 та 4 входи.

При необхідності збільшення кількості входів використовуються два способи побудови монтажною логіки.

Перший з них подібно до ТТЛ базується на використанні мікросхем з відкритим стоком (наприклад, логічний елемент 564ЛА10 – **2×(2І-НІ)**). Недолік цього шляху полягає в тому, що логічні елементи з відкритим стоком мають лише два входи.

Другий спосіб базується на використанні звичайних елементів **І-НІ**, **АБО-НІ**. На рис. 2.31 приводяться приклади реалізації багатовходових елементів **І-НІ** (рис. 2.31, а) та **АБО-НІ** (рис. 2.31, б) на основі елементів з меншою кількістю входів.

Для схеми (рис. 2.31, а) маємо:

$$y = y_1 + y_2 = \overline{x_0 \cdot x_1 \cdot x_2} + \overline{x_3 \cdot x_4 \cdot x_5} = \overline{x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5}.$$

Відповідно для схеми (рис. 2.31, б) маємо:

$$y = y_1 \cdot y_2 = \overline{x_0 + x_1 + x_2} \cdot \overline{x_3 + x_4 + x_5} = \overline{x_0 + x_1 + x_2 + x_3 + x_4 + x_5}.$$

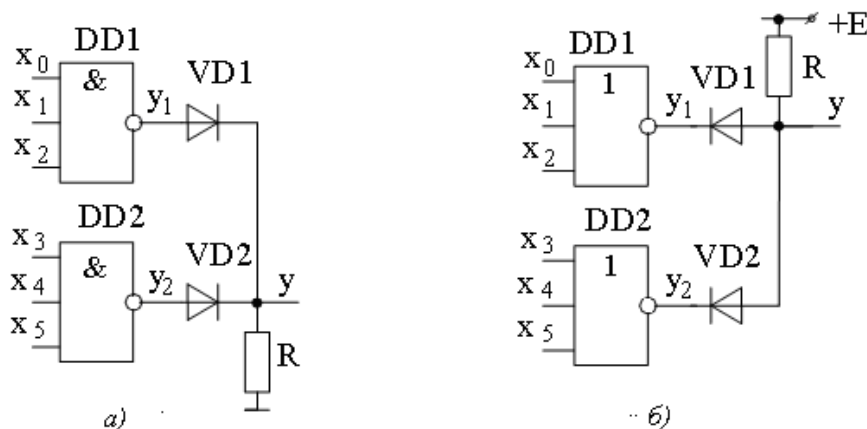


Рис. 2.31

Кількість мікросхем і діодів може бути більшою, ніж приведено на схемі.

**Використання двонаправлених ключів.** Двонаправлені ключі в КМОН-технологіях є складовою частиною багатьох мікросхем – мультиплексорів, перетворювачів кодів та ін. Вони також виготовляються як самостійні вироби, що дозволяють забезпечувати комутацію цифрових та аналогових сигналів зі струмом до 10 мА.

Як приклад, розглянемо можливості використання мікросхеми К564КТ3, що має чотири незалежні ключі, за типом того, що приведений на рис. 2.26.



Опір каналу ключа залежить від напруги живлення і при  $E = 15 \text{ В}$  має величину, що не перевершує  $60 \text{ Ом}$ . Характер опору – активний у діапазоні частот до  $40 \text{ МГц}$ . Канал керування – ізолюваний, з опором ізоляції до  $10^{12} \text{ Ом}$ . Для спеціалістів з аналогової техніки корисними будуть такі дані: коефіцієнт гармонік синусоїдального сигналу  $K_f < 1\%$ ; перехресні завади між будь-якими двома ключами мікросхеми – менше  $50 \text{ дБ}$ .

Більш детальну інформацію можна одержати з довідників.

Керовані ключі знаходять широке застосування. На рис. 2.32 приведені приклади їх використання в якості релейних елементів з відповідними схемами комутації.

На рис. 2.32, д) приведена схема керування магазином опорів, яка може широко використовуватись в аналого-цифрових перетворювачах, у різних аналогових та генераторних схемах з цифровим керуванням. Подібним шляхом можливо організувати керування конденсаторним набором та ін. Широке використання двонаправлені ключі знайшли в серіях К590, К543.

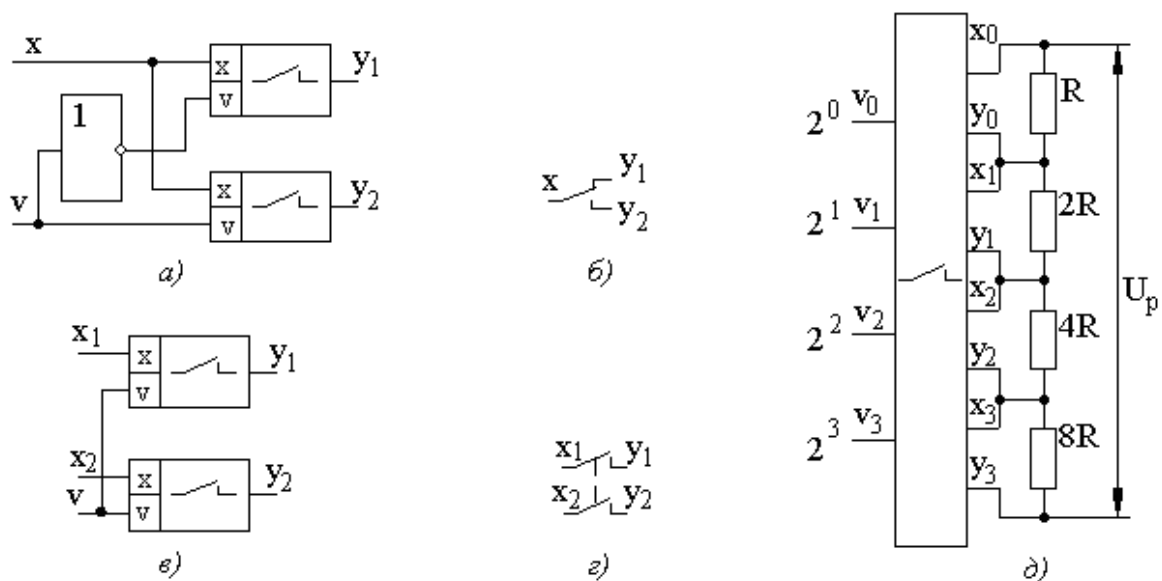


Рис. 2. 32

**Особливості використання монтажної логіки.** Найчастіше монтажна логіка використовується в схемах **2І-НІ**, в яких стік верхнього  $n$ - транзистора виводиться без з'єднання з транзисторами  $p$ - навантаження. Стік може бути

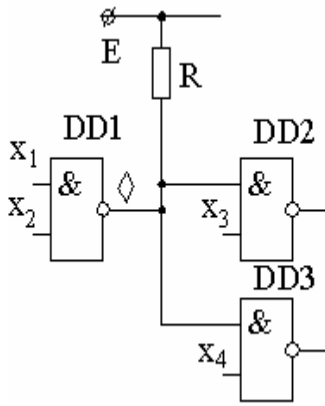


Рис. 2.33

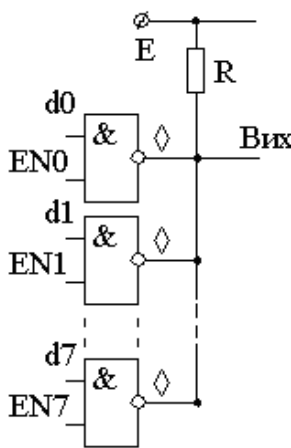


Рис. 2.34

приєднаний до напруги живлення через зовнішні елементи – такі, як резистори, створюючи монтажну логіку, котушки реле, світло діоди і т.д.

На рис. 2.33 приводиться приклад монтажної логіки, де елемент DD1 має відкритий стік. У схемах монтажної логіки резистор зовнішнього навантаження  $R$  повинен мати по можливості менший опір. Але, в той же час, він не може бути дуже малим, оскільки ним задається величина вхідного струму мікросхеми. Наприклад, при величині вхідного струму 5 мА необхідно мати  $R = 1$  кОм. Це значно перевищує  $R_p$ , а тому швидкодія таких ключових схем буде значно меншою.

При використанні схем з відкритим стоком величина опору обмежуючого резистора  $R$  знаходиться за формулою:

$$R = \frac{E - U_{\text{сд}} - U_{\text{вих}}^0}{I_{\text{сд}}}$$

де  $U_{\text{сд}}$ ,  $I_{\text{сд}}$  – відповідно, падіння напруги на світлодіоді (0,3...0,4 В) та його робочий струм (5...20 мА).

Більшість світлодіодів не критичні до величини робочого струму, тому і вибір  $R$  може бути досить вільним.

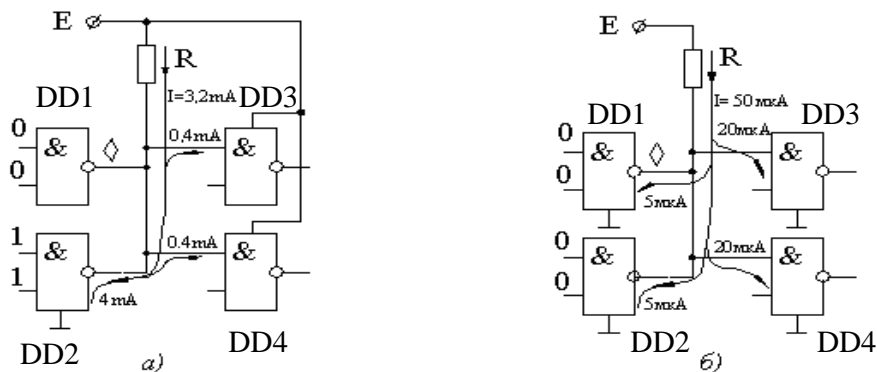


Рис. 2.35

У той же час, при виборі резистора навантаження необхідно враховувати наступні обмеження:

1. Сумарний струм через обмежуючий резистор при стані лог. „0” на виході мікросхеми монтажноі логіки і вхідний струм стану лог. „0” керованих нею мікросхем не повинен перевищувати струм активного виходу в стані лог. „0”. Цю особливість пояснює рис. 2.34, *а*, де струм  $I$  через резистор  $R$  визначається як різниця між вхідним струмом мікросхеми DD2 в стані лог. „0” на виході і вхідним струмом мікросхем DD3, DD4, тобто  $I = 4 - 2 \times 0,4 = 3,2$  мА. Відповідно, мінімальна величина опору резистора  $R$  повинна бути:

$$R = \frac{E}{I} = \frac{5 \text{ В}}{3,2 \text{ мА}} = 1,6 \text{ кОм}.$$

2. Якщо ЛЕ знаходиться в стані лог. „1” на виході, падіння напруги на опорі в високому закритому стані не повинно знижувати вихідну напругу до рівня, нижчого  $U_{\text{мін}}^1$ , гарантуючи мінімальний високий рівень для мікросхем навантаження. Відповідно до рис. 2.34, *б*, загальний споживаний струм складає:

$$I = 2 \cdot 5 \text{ мкА} + 2 \cdot 20 \text{ мкА} = 50 \text{ мкА}.$$

Падіння напруги на резисторі повинно бути меншим ніж 1,5 В, тому

$$R \leq 1,5 / 50 \cdot 10^6 = 30 \text{ кОм}.$$

Таким чином для даної схеми опір резистора  $R$  повинен знаходитись в діапазоні 1,6...30 кОм.

Монтажна логіка на КМОН ІС широко використовується у тих випадках, коли необхідно забезпечувати вибіркочу передачу даних на одну шину. Прикладом схеми, що забезпечує таку передачу є рис. 2.35. Якщо, наприклад, необхідно забезпечити передачу даних з входу  $d_1$  на вихід схеми, то на всі входи EN (enable), за винятком EN1, подаються сигнали низького рівня, які переводять виходи відповідних мікросхем в стан лог. „1”. На вхід EN1 подається сигнал логічної “1”, внаслідок чого дані з входу  $d_1$  будуть передаватись безпосередньо на вихід з інверсією.

**Особливості роботи з мікросхемами КМОН.** Порівняно з TTL, КМОН ІС мають ряд особливостей, які як полегшують роботу з ними, так і ускладнюють. Широкий діапазон напруг живлення, постійне співвідношення напруг перемикачів, висока завадостійкість, мала потужність споживання суттєво спрощують вимоги до джерела живлення. Мікросхеми КМОН малочутливі до пульсацій випрямленої напруги живлення, пульсації на рівні (2...3)% від обраної номінальної величини не впливають на роботу цифрових схем. Широкий діапазон робочих напруг дає можливість використовувати одне джерело живлення при комбінаціях TTL – КМОН (5 В); аналогові схеми – КМОН ІС (15 В) та інших. Особливих умов для узгодження мікросхем у таких випадках немає, необхідно лише врахувати величини навантаження на виходах. У тих випадках, коли рівні навантаження не узгоджуються, використовуються спеціальні узгоджуючі мікросхеми, які випускаються як в серіях КМОН, так і TTL. Вони дають можливість узгодити сигнали не тільки по навантаженню, а і по рівнях напруги, якщо мікросхеми TTL і КМОН живляться від різних джерел.

До особливостей КМОН ІС, які завдають “головного болю” розробникам, слід віднести їх високу чутливість до електростатичних зарядів, малий рівень вихідного струму, наявність охоронної діодної схеми, можливість тиристорного ефекту та ін. Вони вимагають строгого виконання правил користування, які в різній мірі розкривались у попередніх розділах.

Чутливість до електростатичних зарядів приводить до необхідності дбайливого поводження з мікросхемами на всіх стадіях роботи з ними. Детально ці питання розкриті в спеціальній літературі. Нижче приводяться основні з них.

Весь час до монтажу мікросхеми повинні зберігатись так, щоб всі виводи були з'єднані між собою за допомогою фольги; при монтажі виводи необхідно спочатку з'єднувати з загальною шиною, потім підключати шини живлення і, нарешті, інформаційні входи та виходи; вільні виводи повинні бути з'єднані з одною з шин живлення (особливість підключення вільних входів буде

розглянута більш детально згодом); вільні виходи мікросхеми необхідно залишати без підключення; при монтажі весь інструмент радіомонтажника повинен мати надійне заземлення; мікросхеми на плату встановлюються останніми.

Підключення входів до загальної шини чи до шини живлення залежить від логіки роботи мікросхеми. Наприклад, для елемента **I-III** рекомендуються два варіанти підключення вільних входів. Якщо вільні входи підключаються до шини живлення, то відкривання n-канальних транзисторів інверторів, за рахунок постійного зміщення, здійсниться раніше. Загальна порогова напруга буде меншою, ніж у випадку, коли всі входи з'єднуються з інформаційними. Як наслідок цієї особливості, витікає наступне правило. Якщо завади в схемі проходять по загальній шині, то вільні входи мікросхеми доцільно з'єднувати з інформаційними. Якщо ж завади проходять по шині живлення, то вільні входи слід приєднати до шини живлення.

Незначна величина вихідного струму мікросхем приводить до необхідності обмеження в ємності навантаження. Зміна станів ключа супроводжується її зарядом та розрядом. Струми заряду-розряду обмежуються активними опорами каналів транзисторів, і при великій ємності навантаження їх величина може досягати недопустимих величин.

У тих випадках, коли ємність навантаження досить велика, захист транзисторів ключа може бути забезпечений допоміжним обмежувальним резистором  $R_{g1}$ , встановленим безпосередньо на виході ключа (рис. 2.36), який обмежує струм розряду конденсатора  $I_{\text{вих}}^0$ . Але в ситуації, коли при зарядженому конденсаторі вимикається живлення схеми, розрядний струм  $I_{\text{вх}}^1$  конденсатора замикається через вхідний діод VD наступного ключа. Для його обмеження на рівні 1...2 мА, допустимому для діода, встановлюється допоміжний резистор  $R_{g2}$ . Він дає можливість обмежувати також і струм охоронних діодів у тих випадках, коли порушується умова (2.20) по рівню вхідної напруги. Струм охоронних діодів  $I_{gc}$  також необхідно обмежувати

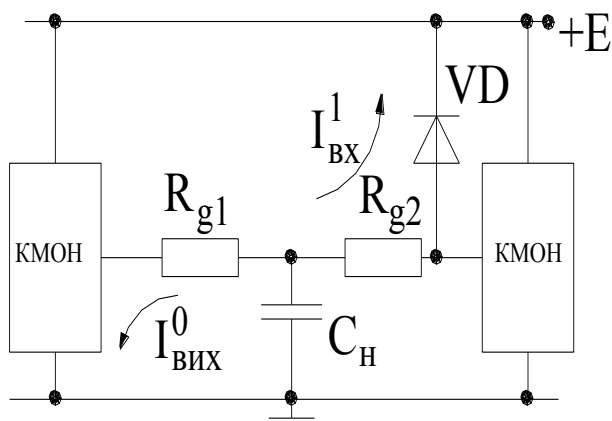


Рис. 2. 36

може досягати напруги живлення.

Наявність резисторів  $R_{g1}$ ,  $R_{g2}$  не знижує статичних характеристик ключів, але може суттєво впливати на їх швидкодію.

### **Особливості сумісного використання КМОН та ТТЛ ІС.**

При проектуванні цифрових систем проєктант повинен підібрати тип використовуваних мікросхем, виходячи з необхідних експлуатаційних параметрів – таких, як швидкодія, потужність, вартість та ряд інших. Але часто мають місце ситуації, коли в обраній серії відсутні ті чи інші типи мікросхем, які існують в інших серіях. У таких випадках з'являється необхідність узгодженого використання різних серій мікросхем. У найбільшій мірі це зустрічається при необхідності сумісної роботи КМОН і ТТЛ ІС.

Існує декілька факторів, які необхідно враховувати при з'єднанні мікросхем різних серій.

Перший з них – це порогові рівні логічних сигналів. Узгодження порогових рівнів необхідно забезпечувати як для сигналів низького рівня, так і для високого. Для сигналів низького рівня це вихідна напруга керуючого попереднього елемента  $U^0_{\text{ВХ.МАКС}}$  і вхідна напруга керованого наступного елемента  $U^0_{\text{ВХ.МАКС}}$ .

Для надійної роботи наступної мікросхеми необхідно виконання умови:

$$U^0_{\text{ВХ.МАКС}} - U^0_{\text{ВХ.МАКС}} > 0.$$

рівнем 1...2 мА. Тому величина  $R_{g2}$  залежатиме від можливого рівня вхідної напруги негативної полярності. Відповідно до (2.22), величина  $R_{g2}$  може бути обчислена за формулою:

$$R_{g2} = \frac{|U_{\text{ВХ}}| - 0,7 \text{ В}}{I_{gc}}.$$

При роботі мікросхем в імпульсних генераторах величина  $U_{\text{ВХ}}$

Аналогічно, для сигналів високого рівня узгодженість забезпечується напругами  $U^1_{\text{вих.мін}}$  керуючого елемента і  $U^1_{\text{вх.мін}}$  керованого елемента. Узгодженість забезпечується при виконанні умови:

$$U^1_{\text{вих.мін}} - U^1_{\text{вх.мін}} > 0.$$

З описаних вище серій мікросхем не всі довільно вибрані пари можуть відповідати вказаним умовам.

Другий фактор, який необхідно обов'язково враховувати при з'єднанні різних типів мікросхем, – це узгодженість вихідних і вхідних струмів. Цей фактор можна не враховувати лише тоді, коли керованими є мікросхеми КМОН. У той же час, якщо мікросхема КМОН працює на декілька входів ТТЛ, то особливу увагу слід приділяти величині струму, який споживається ТТЛ мікросхемами при низькому рівні вхідного сигналу.

Третій фактор узгодження – це ємність навантаження, яка для обох типів мікросхем приводить до зростання величин часових затримок і динамічної потужності розсіювання. Для серій КМОН вона пропорційна величині  $C \times E^2 \times f$ , де  $f$  – частота перемикання логічного елемента;  $C$  – ємність паразитного конденсатора на його виході. Для ТТЛ серій вона значно менша порівняно зі статичною потужністю. Величина ємності навантаження в основному визначається кількістю приєднаних входів до виходу керуючої мікросхеми, тому необхідно керуватись попередньою інформацією відносно невикористаних входів.

### ***2.3.5. Базові елементи БіКМОН-технології***

Успіхи у розвитку цифрової схемотехніки останнього десятиріччя в значній мірі обумовлені БіКМОН-технологією, яка поєднує біполярну і КМОН-технології в одному базовому логічному елементі. Мета такого поєднання – об'єднання переваг КМОН (низької потужності споживання, високого вхідного опору, високих порогових рівнів) з високою навантажувальною здатністю ТТЛ, яка особливо проявляється при зростанні ємності навантаження.

У БіКМОН, як і в КМОН-технології, використовується велика кількість інверторів. Усі вони базуються на використанні n-p-n транзисторів для збільшення вихідного струму КМОН-інверторів.

Найбільш просто це досягається шляхом з'єднання кожного з польових транзисторів інвертора VT1 і VT2 з n-p-n транзисторами VT3 і VT4, як зображено на рис. 2.37, а. Таке з'єднання транзисторів забезпечує високий вхідний опір. Біполярні транзистори фактично з'єднані за такою ж конфігурацією, що і вихідний каскад ТТЛ ІС. Функції інверсії і розщеплення фази вхідного імпульсу забезпечуються парою польових транзисторів.

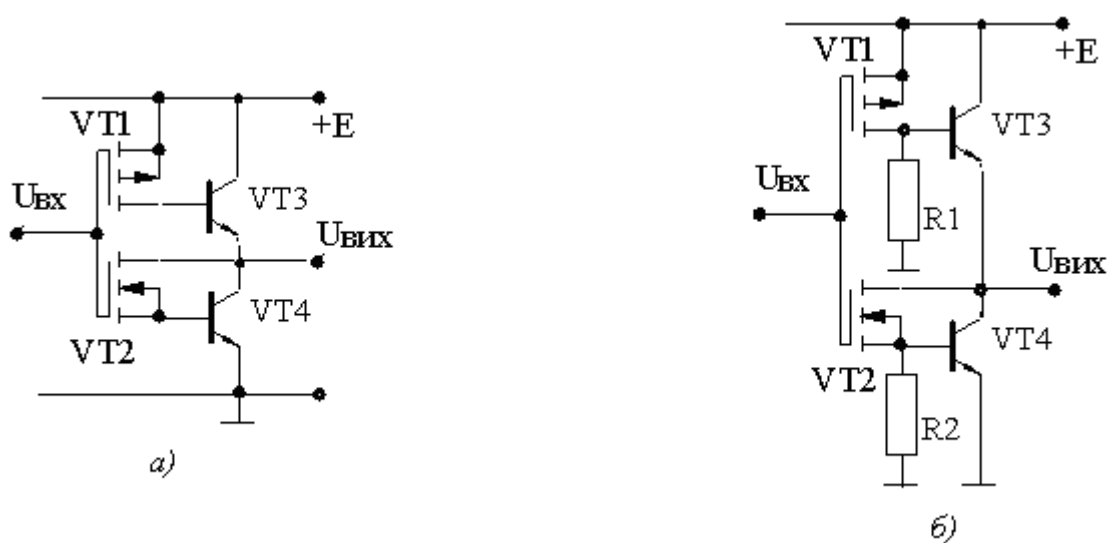


Рис. 2. 37

Робота ключа полягає у наступному. При низькому рівні вхідного сигналу  $U_{вх}$  транзистори VT2 і VT4 знаходяться у вимкнутому стані, а транзистор VT1 включається. Він починає проводити струм від джерела  $E$  в базу транзистора VT3, включаючи його. Останній забезпечує високий вихідний струм для зарядки ємності навантаження інвертора. В результаті час зарядки ємності значно скорочується і зменшується тривалість фронту вихідного імпульсу. Транзистор VT3 виключається, коли вихідна напруга  $U_{вих}$  досягне величини  $(E - U_{BE.3})$ , і тому високий вихідний рівень ключа буде меншим, ніж напруга живлення  $E$ , що є його недоліком. Коли  $U_{вх}$  переймає високий рівень,



транзистори VT1 і VT3 вимикаються, а VT2 вмикається, забезпечуючи своїм струмом базу транзистора VT4. Останній відкривається і забезпечує можливість протікання значного струму розряду конденсатора навантаження, що приводить до суттєвого скорочення тривалості спаду імпульсу і результуючої затримки. Негативна властивість ключа полягає в тому, що VT4 вимикається, якщо  $U_{\text{вих}}$  досягне величини  $U_{\text{BE},4}$  і, відповідно, сигнал низького рівня буде вищим від нуля.

Окрім вказаних недоліків, існує ще один, який полягає в тому, що при протіканні струму через бази транзисторів VT3 і VT4, в них накопичується збитковий заряд і при закриванні транзисторів не існує шляху для його видалення. Як результат, транзистори закриваються повільно, зтягуючи перехідні процеси і створюючи додаткові динамічні струми. Ця проблема може бути розв'язана шляхом підключення до баз транзисторів VT3 і VT4 резисторів  $R_1$  і  $R_2$  (рис. 2.37, б). У такій схемі накопичений у базах транзисторів заряд стікає на „землю” через резистори  $R_1$  або  $R_2$ . Резистор  $R_2$  забезпечує допоміжну позитивну властивість у роботі ключа. Вона полягає в тому, що при  $U_{\text{вх}} = 1$  і після того, як транзистор VT4 закриється, напруга  $U_{\text{вих}}$  продовжує падати нижче  $U_{\text{BE},4}$  і вихідний електрод зменшує свій потенціал через послідовно включені транзистор VT2 і резистор  $R_2$ . Тобто резистор  $R_2$  забезпечує підтягування вихідного потенціалу ключа до нульового рівня. Але це коло має високий опір, тому цей процес повільний. У той же час, резистор  $R_1$  приводить до зростання статичної розсіюваної потужності. При  $U_{\text{вх}} = 0$  має місце коло для протікання постійного струму від джерела  $E$  через транзистор VT1 і резистор  $R_1$  на загальну шину.

На рис. 2.38, а приведена схема, в якій опори  $R_1$  і  $R_2$  замінені n-МОН транзисторами.

Перш за все, така заміна спрощує технологічний процес виготовлення мікросхем, оскільки виготовлення транзистора простіше, ніж резистора.

З іншого боку, транзистори виконують функції резисторів лише тоді, коли

це необхідно. Транзистор VT3 буде проводити тільки при  $U_{ВХ} = 1$  протягом часу, коли його струм буде зворотнім базовим струмом для VT5, прискорюючи його вимикання. Аналогічно, VT4 буде проводити лише тоді, коли  $U_{ВХ}$  зменшується і заряд бази транзистора VT6 необхідно видалити для прискорення його запирання.

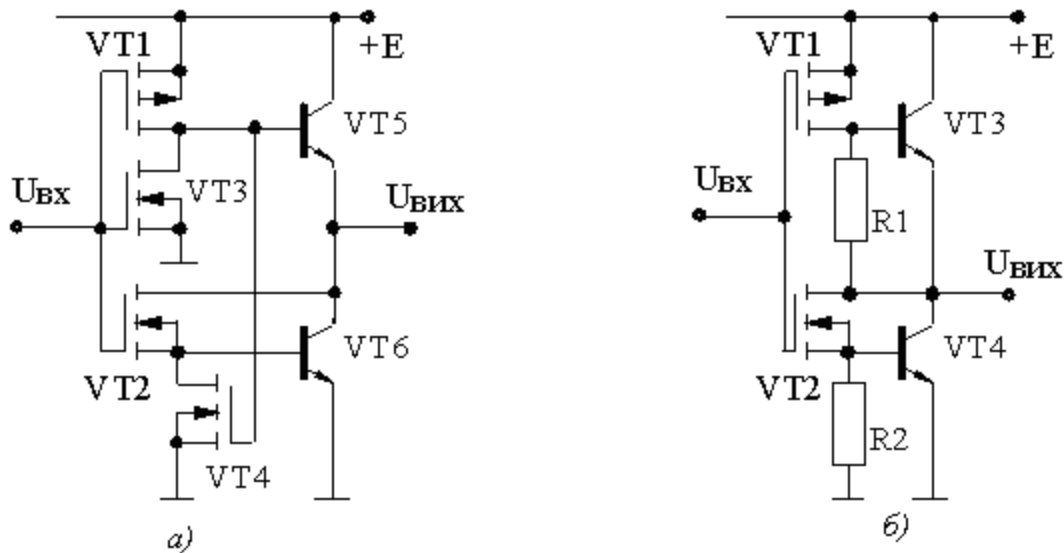


Рис. 2. 38

Останній варіант БіКМОН інвертора, зображений на рис. 2.38, б, називається *R-схемою*. Вона відрізняється від схеми, приведеної на рис. 2.37, б, лише тим, що  $R_1$  приєднаний не до загальної шини, а до виходу інвертора. Така проста заміна має дві позитивні властивості. Перша полягає у зменшенні статичної потужності споживання. Друга – характерна тим, що  $R_1$  використовується як підтягуючий резистор високого рівня вихідної напруги до величини напруги живлення  $E$ , що підвищує верхній пороговий рівень інвертора.

Таким чином, *R-схема* забезпечує вихідні рівні напруги близькими до напруги живлення і нуля.

При проектуванні схеми забезпечуються такі умови перемикання транзисторів VT3 і VT4, при яких вони не перемикаються одночасно і не входять у режим насичення.

Переваги БіКМОН-технологій проявляються в тих випадках, коли ключі працюють з великим струмом навантаження, а також при наявності значної (з точки зору інтегральних технологій) паразитної ємності. Наприклад, при роботі на навантаження з ємністю 1 пФ швидкодія БіКМОН-ключів перевищує швидкодю кращих КМОН-інверторів у 3-4 рази.

Побудова логічних схем з використанням БіКМОН-технологій має загальні риси з КМОН-технологіями. Наприклад, на рис. 2.39 приведена схема двовходового елемента **I-НІ**, принцип роботи якої не потребує допоміжних пояснень.

На сучасному рівні розвитку інтегральних технологій БіКМОН широко використовується при побудові мікропроцесорів, мікроконтролерів,

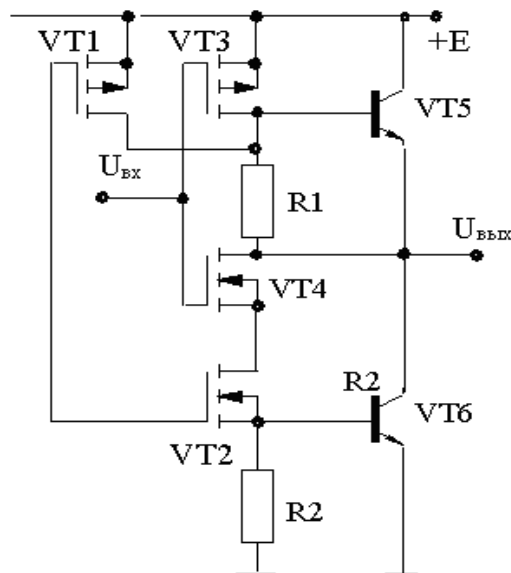


Рис. 2. 39

статичної оперативної пам'яті, програмованих логічних матриць.

Наприклад, фірмою Texas Instrument випускаються такі серії мікросхем за БіКМОН-технологією:

ABT	Advanced BiCMOS Technology
ABTE	Advanced BiCMOS Technology/Enhanced Transceiver Logic
ALB	Advanced Low-Voltage BiCMOS
BCT	BiCMOS Bus-Interface Technology
LVT	Low-Voltage BiCMOS Technology

### 2.3.6. Інтегральна інжекційна логіка

Після вивчення властивостей біполярного транзисторного ключа розглянемо схемотехніку логічних елементів з послідовним та паралельним з'єднанням транзисторів (рис. 2.40, а, б).

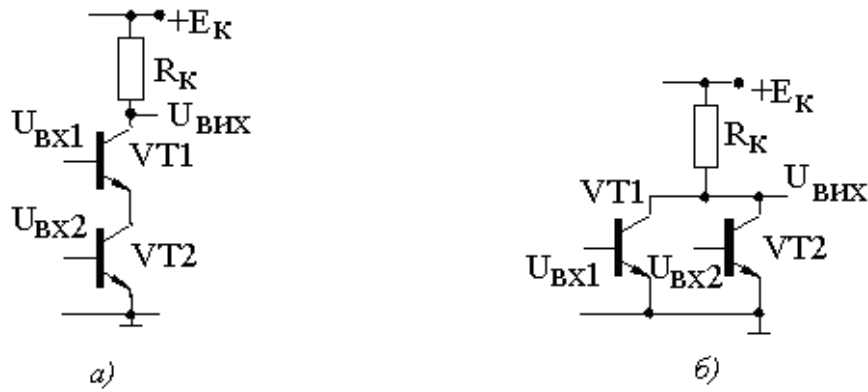


Рис. 2. 40

Основна ідея функціонування приведених схем заключається в порівнянні суми опорів (провідностей) ключових транзисторів з  $R_k$ . Незважаючи на те, що схема **2І-НІ** (рис. 2.40, а) по кожному входу має індивідуальну характеристику, окремо взяті логічні елементи добре виконують свої логічні функції. Але при їх навантаженні на аналогічні схеми неідентичність вхідних характеристик приведе до того, що величини струмів баз транзисторів навантаження будуть відрізнятися у десятки разів. Це означає, що вхід з більшою крутизною характеристики візьме на себе левову частину вихідного струму попереднього ключа. Це явище відоме як *перехоплення струму* і є типовим явищем для паралельного підключених нелінійних елементів. Тому паралельне з'єднання нелінійних елементів у практиці не використовується.

Перехоплення струму приводить до нерівномірного розподілення струмів керування логічних елементів навантаження, через що деякі з елементів можуть навіть не увійти у режим насичення.

Виключити явище перехоплення струму можна лише при використанні у якості навантаження транзисторів з повністю ідентичними вольт-амперними характеристиками базового р-п переходу. Практично це можливо тільки в тому випадку, якщо це буде один і той же транзистор. Тобто функцію розгалуження струмів необхідно організувати в одному транзисторі, який повинен мати декілька колекторів. В результаті приходимо до базової схеми інжекційної логіки  $I^2L$  (рис. 2.41, а).

У приведеній схемі транзистор VT1, включений за схемою з загальною базою, генерує стабільний струм, який подається в базу багатоклекторного транзистора.

Назва “інжекційна логіка” пояснюється тим, що транзистор VT1 (точніше, його емітер) є інжектором р- зарядів, які подаються в базу VT2. Вказані заряди задають величину струму в колекторах VT2. Інвертор VT2 вимикається тоді, коли струм колектора VT1 відбирається з бази VT2 в інше коло – наприклад, у колектор попереднього аналогічного логічного елемента.

Таке перемикання струму забезпечується відповідним зменшенням вхідної напруги  $U_{ВХ}$ . Величина високого логічного рівня задається падінням напруги на зміщеному в прямому напрямку р-п переході база-емітер багатоклекторного транзистора VT2. Низький рівень вихідної напруги визначається потенціалом колектора насиченого ключового транзистора. Малий рівень логічної “1” задає і низький рівень напруги живлення  $E = 2 U_{\text{бе}} = 1,5 \text{ В}$ . Менша величина напруги живлення недопустима через виключення транзистора VT1.

Вихідна характеристика І<sup>2</sup>Л в цілому майже не відрізняється від аналогічної характеристики біполярного насиченого ключа. Величини колекторних струмів можуть змінюватись паралельним з’єднанням колекторів багатоклекторного транзистора, що широко використовується при побудові схем перетворювачів рівнів сигналів. Час включення та виключення ключа задається необхідною величиною струму VT1.

При побудові логічних схем використовується об’єднання колекторів багатоклекторних транзисторів. Іноді інжектор VT1 також будується за багатоклекторною схемою.

На рис. 2.42 приводиться схема, в якій, поряд з операціями інверсії по виходах  $y_1$  та  $y_2$ , виконується операція **АБО-НІ**  $y_{1,2} = \overline{x_1 + x_2}$ .

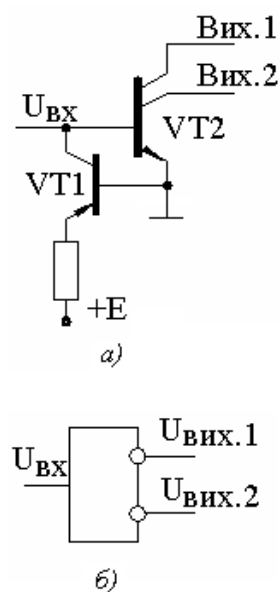


Рис. 2.41

На базі І<sup>2</sup>Л ІС легко будуються тригерні схеми. На рис. 2.43 приведена схема RS-тригера, яка розкриває основні переваги даної технології. Вони заключаються в тому, що інтегральні логічні елементи мають мінімальну кількість напівпровідникових елементів та внутрішніх з'єднань.

Схеми не мають у своєму складі більш складних у технологічному плані елементів – резисторів. Все це дозволяє підвищувати рівень інтеграції в фактично стандартному технологічному процесі біполярної транзисторної логіки. Підвищення швидкодії елементів І<sup>2</sup>Л технології досягається за рахунок переходу на транзистори Шоткі.

На сьогоднішній день використання І<sup>2</sup>Л технології обмежене. У вітчизняній практиці на її основі виготовляються лише декілька видів мікросхем. Це мікросхеми пам'яті серії 541 (541РУ1А, 541РУ21 – мікросхеми статичних оперативних запам'ятовуючих пристроїв; 541РЕ1 – постійний програмований запам'ятовуючий пристрій масочного типу).

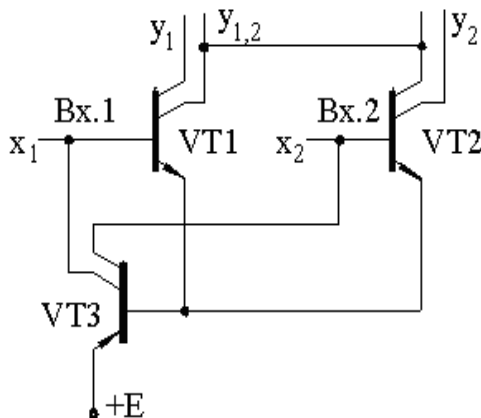


Рис. 2. 42

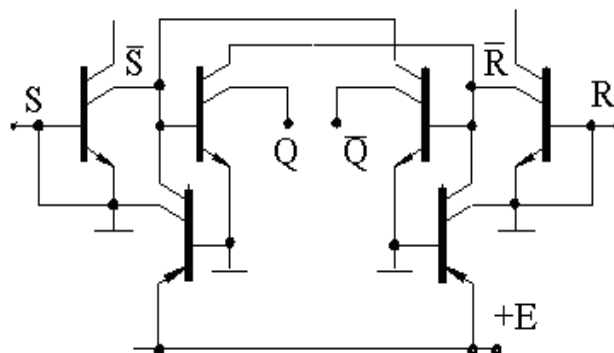


Рис. 2. 43

Реальні технічні параметри (для порівняння з іншими базовими логічними елементами) такі:  $t_3 \approx 2 \div 3$  нс;  $P_{СП} \approx 0,2$  мВт;  $A \approx 0,1$  пДж.

У практиці відомих західноєвропейських та американських компаній технологія І<sup>2</sup>Л також не знаходить поки що широкого використання. Пояснюється це здебільшого значним прогресом у розвитку КМОН- та БіКМОН-технологій.



На рис. 2.45, б приведені характеристики перемикання для прямого та інверсного виходів. З них витікає, що логічні рівні ЕЗЛ змінюються в інтервалі  $-0,8 \dots -1,7$  В. Рівні  $U_{\text{вих}}^1 = -0,8$  В;  $U_{\text{вих}}^0 = -1,7$  В.

Низький рівень логічного перепаду призводить до малої статичної завадостійкості мікросхем.

Враховуючи лінійний режим роботи транзисторів, у забезпеченні завадостійкості важливу роль відіграє опорна напруга  $E_0$ . У схемі, приведений на рис. 2.44, вона задається колом  $R_3$ , VD1, VD2,  $R_7$  і в деякій мірі залежить від напруги живлення. У подальших серіях мікросхем (наприклад, серії K1500) використовуються спеціальні схеми, які забезпечують високу стабільність  $E_0$  при зміні температури та напруги живлення.

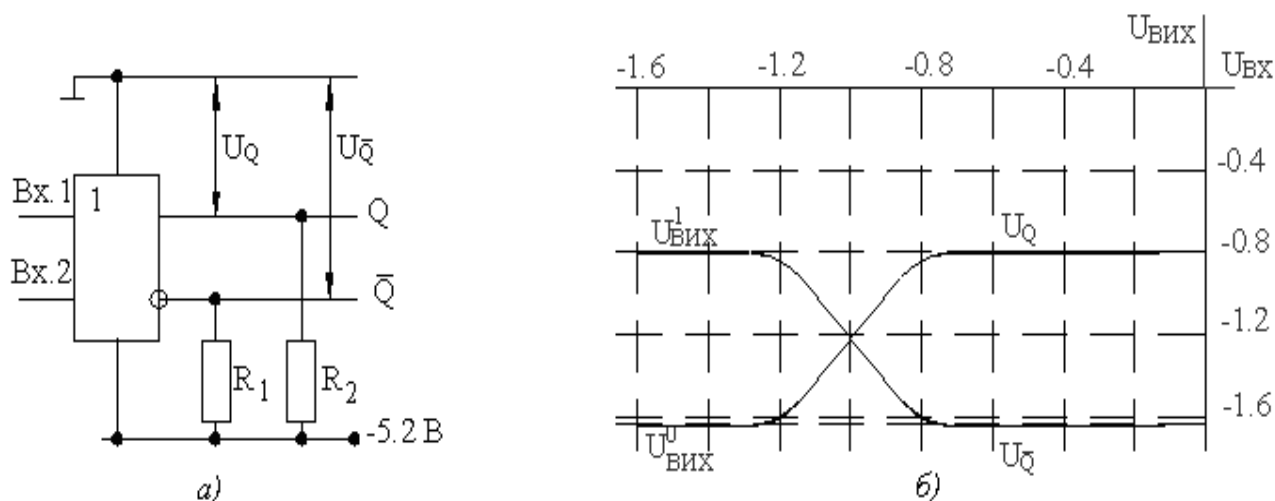


Рис. 2.45

Емітерно-зв'язана логіка в останні роки інтенсивно розвивається, і на її основі виготовляється декілька типів базових логічних елементів. Один з напрямків розвитку ЕЗЛ пов'язаний з переносом емітерних повторювачів з виходу диференційного каскаду на його вхід (рис. 2.46).

Перевага такого базового елемента полягає у тому, що завдяки визначеності коефіцієнта об'єднання по входу, на відміну від коефіцієнта розгалуженості, зменшується загальна кількість резисторів і оптимально розраховуються вхідні елементи схеми. Це приводить не тільки до спрощення



технології, але і до поліпшення технічних характеристик базових елементів. Такі схеми мають назву *емітерно-емітерно-зв'язана логіка* (ЕЕЗЛ або Е<sup>2</sup>ЗЛ).

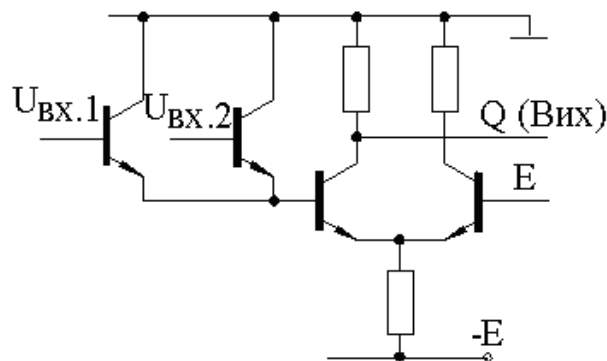


Рис. 2.46

Останнім часом суттєве значення в цифровій схемотехніці має напрямок *деревоподібних* або *багатоступінчастих логічних структур*. Приклад такої структури приводиться на рис. 2.47. На кожному рівні логічного елемента по горизонталі можуть бути реалізовані функції **АБО**, а між рівнями – функція **І**.

Оскільки величини опорних напруг на кожному рівні відрізняються, то схема вимагає обов'язкового узгодження рівнів, але в цілому такі логічні схеми дають значний вигравш за величиною споживаної енергії та рівнем інтеграції.

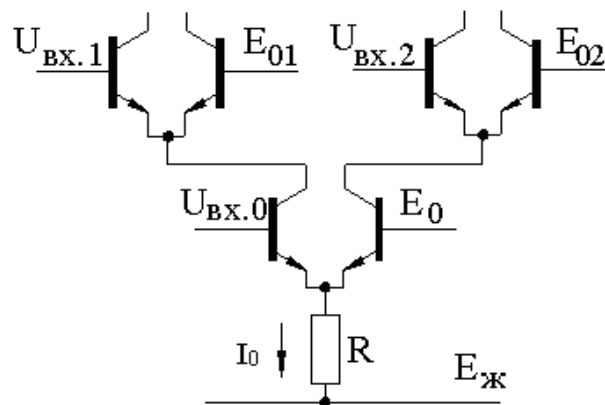


Рис. 2.47

Як приклад багатоступінчастої логічної структури, приводиться схема деревоподібного логічного елемента (рис. 2.48), що реалізує декілька логічних функцій. Нульова ступінь схеми виступає джерелом постійного струму  $I$ , що подається в першу ступінь ЛЕ. Ця ступінь керується логічним входом  $C$ . Логічні входи  $A$  і  $B$  використовуються елементами другого рівня.

Логічні виходи  $Q_0, Q_1, Q_2, Q_3$ , що розгалуджуються завдяки багатоемітерним транзисторам, дають можливість реалізувати чотири види логічних функцій:  $Q_0 = (\bar{A} + \bar{B}) + \bar{C}$ ;  $Q_1 = (A + B) + \bar{C}$ ;  $Q_2 = (\bar{A} + \bar{B}) + C$ ;  $Q_3 = (A + B) + C$ .

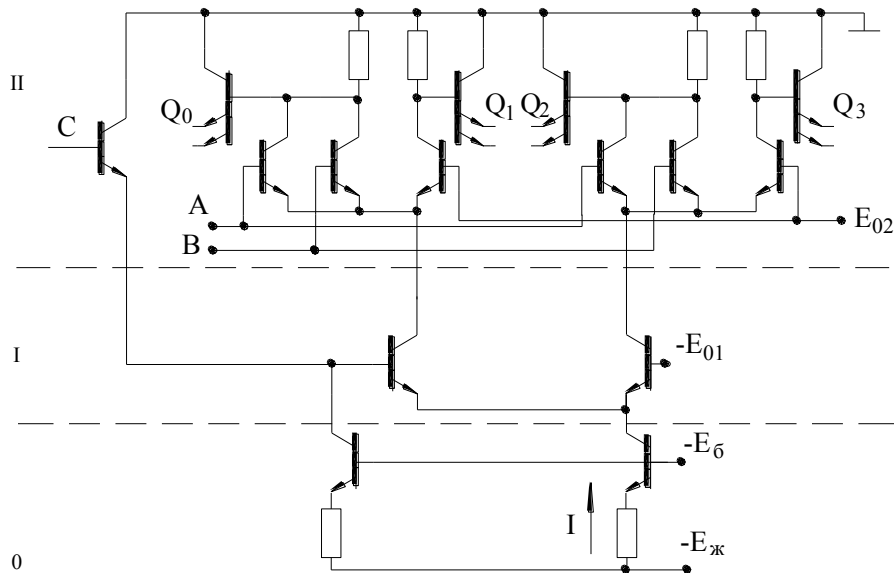


Рис. 2.48

Логічний сигнал  $C$ , який керує першим рівнем структури, перемикає струм  $I$  в одну з схем вищого рівня, забезпечуючи реалізацію логічної операції  $I$ . На верхньому рівні  $\Pi$  можуть бути реалізовані необхідні логічні функції.

Враховуючи той факт, що величина  $t_3$  для елементів ЕЗЛ знаходиться в інтервалі  $0,3 \dots 1$  нс, на базі цієї технології будуються найбільш швидкодіючі мікросхеми. Підвищення швидкодії та зниження потужності досягається шляхом зменшення напруги живлення. Зниження завадостійкості, що має місце при цьому, компенсується тим, що низьковольтні ЕЗЛ-елементи використовують всередині великих інтегральних схем.

Серед мікросхем зарубіжних виробників відомі такі серії, як 10К (10102, 10181, 10209 та ін.), а також 100К (100101, 100117 та ін.). Технічні параметри їх близькі до вказаних вище.

### 2.3.8. Прилади з переносом заряду

Існує два типи приладів, які працюють за принципом переносу заряду від одної ділянки на кремнієвій підложці до іншої – *прилади, що працюють за принципом “пожежних ланцюжків”*, і *прилади з зарядовим зв’язком (ПЗЗ)*.

На рис. 2.49 наведений приклад реалізації пристрою типу “пожежних ланцюжків” на дискретних компонентах.

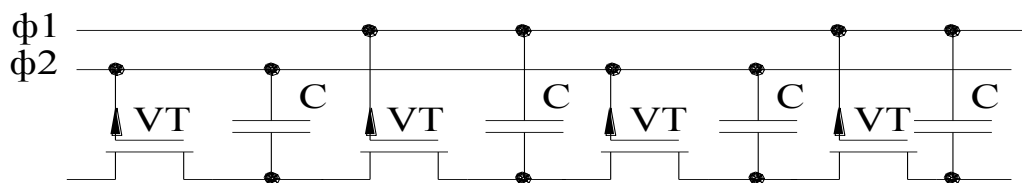


Рис. 2.49

Принцип роботи даного пристрою полягає в тому, що заряд, внесений на один з конденсаторів, шляхом дії двофазної системи синхроімпульсів  $\phi_1$ ,  $\phi_2$  на транзистори VT може переміщуватись від одного конденсатора до іншого. Цим самим може забезпечуватись послідовний запис інформації, її зберігання, затримка на визначений інтервал часу і т.п. Ця схема має дві особливості:

1. Оскільки для запам'ятовування використовується конденсатор, то є можливість оперувати як аналоговою, так і цифровою інформацією;
2. Оскільки через розімкнені транзистори відбувається втрата заряду, ця пам'ять повинна працювати в динамічних режимах.

ПЗЗ (рис. 2.50) простіші “пожежних ланцюжків”. Якщо до одного з електродів прикласти високий позитивний потенціал, то область напівпровідника під ним буде збідненою “дірками” і в ній створиться потенційна яма.

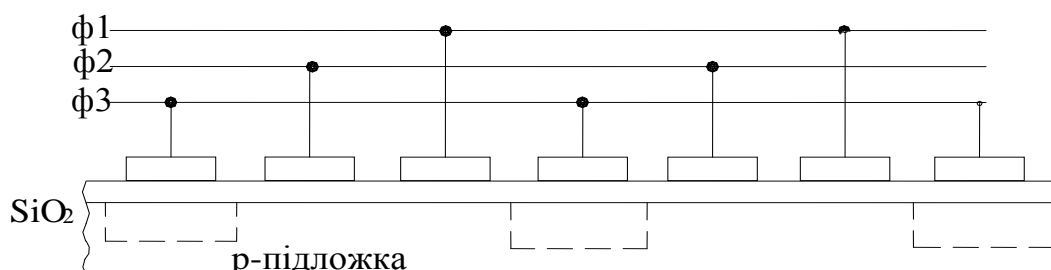


Рис. 2.50

Якщо забезпечити позитивний потенціал  $+φ$  достатньо великим, то напівпровідникова пластинка поблизу електроду змінить тип провідності

(вона стане n- типу ). У будь-якому випадку потенційна яма довго порожньою не буде, адже до неї швидко почнуть притягуватися негативні заряди, що появляються за рахунок термогенерації в тілі напівпровідника.

Пристрій ПЗЗ працює за рахунок накопичення і руху зарядів від однієї потенційної ями до іншої. Трифазна система ПЗЗ найбільш зручна для забезпечення зсуву заряду в одному напрямку вздовж пристрою. Якщо в структуру напівпровідника внесений заряд, то при максимальному потенціалі  $\varphi_1$  він накопичується під першим електродом. Для переміщення під другий електрод спочатку  $\varphi_1$  зменшується і стає тимчасово  $\varphi_1 = \varphi_2$ . Після цього потенціал  $\varphi_2$  зростає, внаслідок чого під  $\varphi_2$  утворюється потенційна яма і заряд стікає під  $\varphi_2$ . Для переводу потенціалу під  $\varphi_3$  потрібно створити  $\varphi_1 = 0$ ,  $\varphi_2 = \varphi_m / 2$ ,  $\varphi_3 = \varphi_m$  і т.д., де  $\varphi_m$  – максимальний потенціал. Пересування потенціалів відбувається послідовно. Оскільки можливо заповнити порожнину ям за рахунок термогенерації, то ПЗЗ можуть працювати тільки у динамічних режимах.

У ПЗЗ є ряд причин втрати заряду, тому його постійно необхідно поновлювати. Швидкість руху заряду хоча і велика, але обмежена. Має місце темновий струм за рахунок термогенерації електронно-діркових пар. Це сприяє заповненню потенційних ям і спотворює сигнал.

Для боротьби з вказаними недоліками розроблені ефективні методи. Наприклад, на рис. 2.51 приводиться варіант ПЗЗ з елементами запису і зчитування інформації.

Вхідна область діє як витік,  $Z_1$  – затвор,  $Z_2$  – стік. Стробуюча напруга  $U_1$  подається на  $Z_1$ , а вхідний сигнал діє на вхідну дифузійну область.

Величина заряду витоку визначається рівнем сигналу, і коли приходить стробуючий імпульс, то перша потенційна яма під  $Z_2$  заповнюється до цього рівня. Заряд також заповнює потенційну яму під  $Z_1$ , але коли знімається стробуюча напруга  $U_1$  – заряд повертається до витоку.

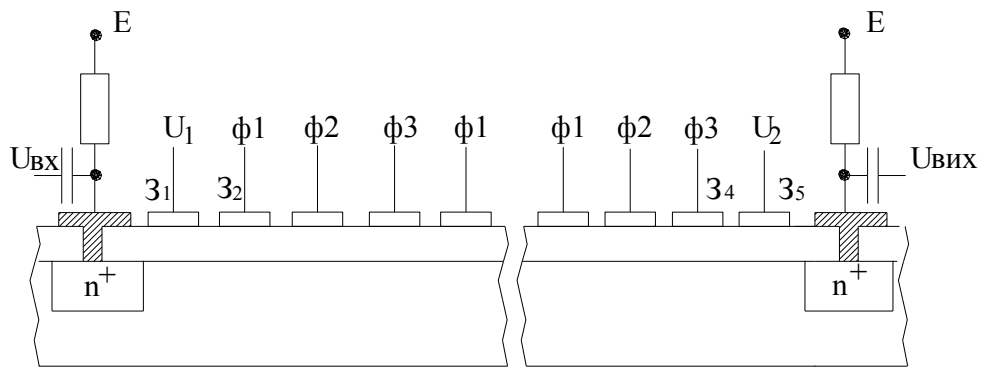


Рис. 2.51

У вихідному каскаді дифузійна n- область діє як стік. Постійне зміщення на  $Z_5$  служить для розв'язки стоку від впливу тактових імпульсів. Заряд від  $Z_4$  проходить через цей потенційний бар'єр і детектується як вихідний сигнал у стоковій дифузійній області. Практично безпосередньо на підложці ПЗЗ розташовані і підсилювачі для підвищення рівня вихідного сигналу. Їх також можна використовувати для отримання незнищеного зчитування вздовж лінії ПЗЗ.

ПЗЗ можуть бути використані для зберігання аналогової та цифрової інформації або для запису і запам'ятовування оптичного сигналу. Якщо на вхід подати аналоговий сигнал, то він автоматично відбирається і зсувається з тактовою частотою так, що на виході з'являється вхідний сигнал, але затриманий і у вигляді імпульсів, модульованих за амплітудою. Для відновлення сигналу його необхідно підсилити та відфільтрувати. Враховуючи дискретний характер передачі інформації, ПЗЗ можуть бути використані у якості регульованих ліній затримки.

Перевагою ПЗЗ у цифровій техніці є простота, яка дозволяє вбудовувати їх у запам'ятовуючі пристрої з дуже великим обсягом пам'яті і отримувати низьку питому вартість на 1 біт. Для підвищення ефективності переносу інформації і запобігання її спотворювання в ПЗЗ вбудовують підсилювачі регенерації, які виконані у вигляді простих порогових пристроїв, що відновлюють вихідні рівні сигналів і встановлюються на певних відстанях уздовж всієї довжини ПЗЗ.

При використанні ПЗЗ для сприйняття зображень на матрицю ПЗЗ спрямовують промені світла. Один з електродів, наприклад,  $\phi_1$  – знаходиться під (+) напругою для утворення потенційної ями. Він збирає (-) заряди з електронно-дірочних пар, які утворюються під дією світла. Величина отриманого заряду пропорційна інтенсивності світла і тривалості його впливу на ПЗЗ. У визначений момент часу сигнал зсувається вздовж лінії і послідовно зчитується з ПЗЗ. Час експонування повинен бути набагато меншим часу, який витрачається на зсув, для того щоб запобігти можливому “розмазуванню” сигналу. Збільшити площу зображення можна за рахунок збільшення кількості рядків, але максимальна кількість елементів у рядку, через які може здійснюватися перенос, визначається ефективністю переносу заряду. Для збільшення цієї кількості можна застосувати мультиплексування.

Реалізація функцій цифрової логіки на основі ПЗЗ потребує взаємодії з інформацією, яка зберігається у зсувних ПЗЗ – регістрах. Таку взаємодію можна реалізувати двома способами. Один з них називається *зчитуванням з руйнуванням інформації*, адже в процесі такого зчитування вихідні логічні сигнали знищуються. Інший спосіб – *неруйнуюче зчитування*, при якому логічні сигнали в ПЗЗ-регістрах зберігають свої логічні рівні.

На ПЗЗ можуть реалізовуватись різні логічні функції як постійного, так і тимчасового призначення, що дає можливість суттєво розширити можливості приладів.

ПЗЗ мають такі технічні характеристики: площа  $(1...2) \times 10^{-3}$  мм<sup>2</sup>;  $t_3 = 50...100$  нс;  $A = 0,2$  нДм.

Технології ПЗС широко використовуються при побудові складних функціональних схем. Наприклад, мікросхема К593БР1 використовується як складна лінія затримки. Мікросхеми серії К528 призначені для фільтрації, запам'ятовування, затримки, накопичення і часового перетворення аналогових сигналів. Мікросхеми серії К1111 використовуються як фільтри, частоти зрізу яких задаються частотою керуючих сигналів.

### 2.3.9. Арсенід-галієві логічні елементи

Усі розглянуті вище базові логічні елементи і інтегральні технології будуються на основі кремнієвих напівпровідників, які майже повністю задовольняють вимогам сучасної електроніки. Але найбільш сучасні напрямки електроніки вимагають забезпечення швидкодії приладів, яка не може бути забезпечена електронною елементною базою, що виготовляється на основі кремнію. Для побудови таких приладів використовується *арсенід галію (GaAs)* – напівпровідник на основі галію, що розміщений у 3-му стовпці періодичної таблиці елементів, і арсеніду з п'ятого стовпця. Тому він відомий як III-V- напівпровідник. Головна перевага GaAs полягає в тому, що n-напівпровідник на його основі має рухомість основних носіїв – електронів – у 5-10 разів більшу, ніж у кремнієвих напівпровідниках. Це означає, що напівпровідник на основі солі GaAs має набагато більшу провідність і, відповідно, більші робочі струми. Це, в свою чергу, дає можливість значно підвищити швидкість заряду-розряду паразитних ємностей.

Арсенід-галієві прилади використовують в останні роки для виготовлення дискретних компонентів підсилювачів для мікрохвильової техніки. Зовсім недавно GaAs почав використовуватись для побудови аналогових (операційні підсилювачі) і цифрових (мікропроцесори) ІС з робочими частотами, які перевищують 1 ГГц.

Базовим елементом, який використовується для побудови різноманітної схемотехніки, є n- канальний польовий транзистор зі скороченою назвою MESFET (*Metal-Semiconductor Field Effect Transistor*). Технологічно такий транзистор, на відміну від кремнієвих, будується не на домішкових структурах, а на підкладці з чистого GaAs, який має дуже низьку провідність і є напівізолюючою структурою. Така особливість суттєво спрощує технологічні задачі ізоляції різних приладів однієї мікросхеми і тим самим дає можливість суттєво зменшити паразитні ємності як між окремими транзисторами, так і між транзисторами і заземленням.

Канал у структурі транзистора будується на основі n- домішкових структур. Тільки на них будуються MESFET. Поряд з транзистором технологічно легко виготовляються діоди Шоткі, які широко використовуються в схемотехніці надвисокочастотних приладів. MESFET р-типу не мають таких переваг, як n- домішкові прилади. Цей недолік приводить до ускладнення схемотехніки базових логічних елементів.

Реальні величини затримок інверторів на базі MESFET-транзисторів складають від 10 до 100 пікосекунд, що є головною перевагою технології. Недоліками є порівняно висока потужність споживання, вузький діапазон робочих напруг, вища, ніж у інших технологій, вартість. Напруга живлення логічних елементів лежить у межах 1,2...1,5 В.

Однією з перших серій, які почали виготовлятися на базі арсенід-галієвих напівпровідників, є DCFL (*Direct Coupled FET Logic*). Схема її базового елемента приведена на рис. 2.52, а. Вона використовує збіднений транзистор VT1 і збагачені транзистори VT2, VT3, призначені для прийому вхідних сигналів. Логічний елемент реалізує операцію **2АБО-НІ**.

Розглянемо особливості роботи цього елемента.

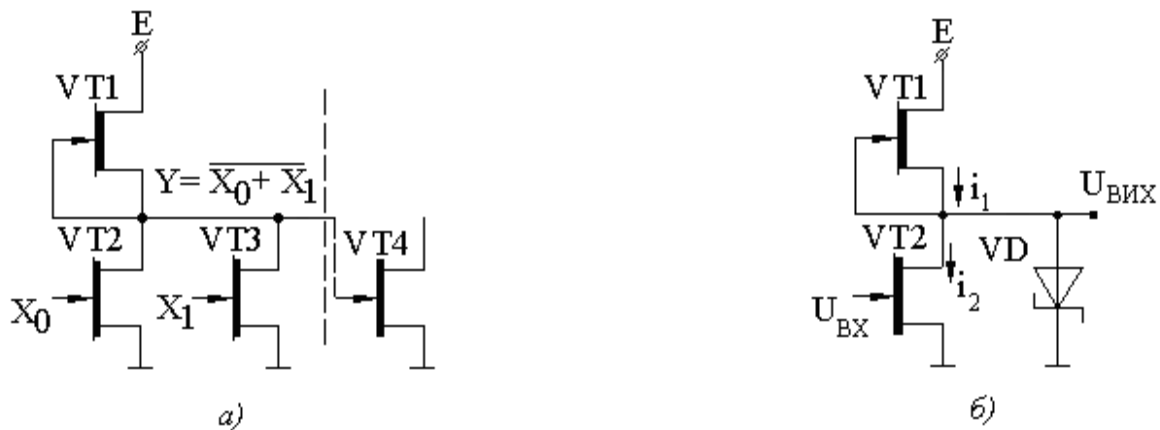


Рис. 2.52

Коли вхідна напруга, представлена логічним сигналом  $X_0$ , прикладається до затвору транзистора VT2 і має величину меншу, ніж її порогове значення  $U_{П}$ , транзистор VT2 знаходиться у виключеному стані. Величина



порогової напруги для MESFET є позитивною з типовим рівнем  $0,1 \dots 0,3$  В. Якщо вихід базового елемента вільний (не приєднаний до входів інших мікросхем), то напруга на виході буде близькою до  $E$ , оскільки кола для протікання струму від транзистора VT1 немає.

Транзистор VT1 працює у режимі джерела струму і при відсутності кола для протікання струму переходить у режим насичення. Реально схема працює на аналогічний логічний елемент (транзистор VT4 наступної мікросхеми). У такому випадку струм, генерований транзистором VT1, проходить через транзистор навантаження. Вище було відмічено, що поряд з арсенід-галієвим транзистором монтується діод Шоткі. У розгляданій структурі базового логічного елемента він приєднується паралельно входу MESFET. При прямому протіканні струму через діод на ньому має місце падіння напруги близько  $0,7$  В. У цьому полягає суттєва різниця між цими схемами і КМОН-структурами.

На рис. 2.52, б приведена схема DCFL-інвертора з навантаженням у вигляді діода Шоткі. При величині вхідної напруги, меншій порогового рівня, струм через ключ VT2  $i_2 = 0$  і весь струм  $i_1$  від транзистора VT1 протікає через прямозміщений діод Шоткі, забезпечуючи падіння напруги  $0,7$  В. Оскільки напруга живлення низька ( $1,2 \dots 1,5$  В), то транзистор VT1 працюватиме в активному режимі. Як тільки вхідна напруга перевищить порогове значення транзистора VT2, останній включиться і почне пропускати струм  $i_2$ . Спочатку величина цього струму буде малою, але вона зменшуватиме струм, що протікає через діод Шоткі. Напруга на VD також зменшуватиметься. Як тільки забезпечиться виконання рівності  $i_1 = i_2$ , діод Шоткі перестане проводити струм і закриється, а подальше зростання вхідної напруги приведе до зниження падіння напруги на транзисторі VT2 до величини  $0,1 \dots 0,2$  В. Звідси маємо, що діапазон вихідних напруг такого ключа знаходиться в інтервалі  $0,1 \dots 0,7$  В, що суттєво сприяє зниженню потужності на перемикання і інтервалу затримки.

### 2.3.10. Логічні ключі, що використовують збіднені MESFETи

Розглянуті вище логічні елементи характеризуються тим, що використовують як збагачені, так і збіднені транзистори, і це суттєво ускладнює технологічні процеси їх виготовлення. Також необхідно при виробництві дуже чітко забезпечувати порогові рівні, щоб коливання напруги і завади не змогли привести до хибних спрацювань. Одним з альтернативних рішень, яке ліквідує вказані проблеми, є побудова логічних елементів з використанням лише збіднених транзисторів.

На рис. 2.53 приведена схема базового логічного елемента, в якому використовуються лише збіднені MESFETи. Сім'я логічних елементів такого типу має назву FL (*FET Logic*). Основу логічного елемента складає

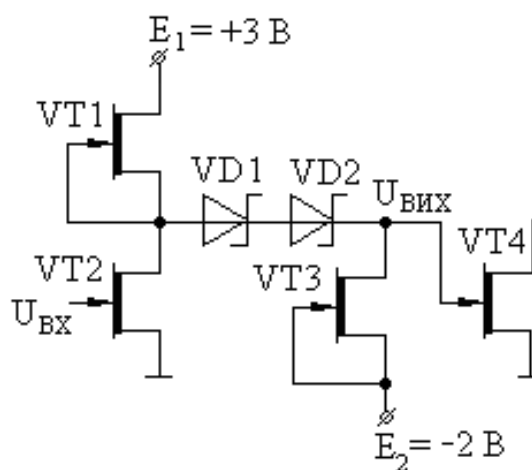


Рис. 2.53

переключаючий транзистор VT2, навантаженням якого є такий же транзистор VT1, включений у режимі перетворювача напруга-струм. Оскільки порогова напруга збіднених транзисторів негативна, то до входу транзистора VT2 необхідно прикладати негативну напругу для його вимикання. З іншого боку, низький вихідний рівень напруги на витоці ключа буде завжди позитивним. Звідси витікає, що

логічні рівні витоку транзистора VT2 несумісні з рівнями, необхідними для входу ключа. Проблема несумісності розв'язується просто шляхом зсуву напруги на витоці транзистора на величину, рівну падінню напруги на двох діодах Шоткі VD1 і VD2, що приблизно дорівнює 1,4 В. Для забезпечення постійного рівня зміщення необхідно, щоб через діоди протікав струм незмінної величини, що забезпечується транзистором VT3. Для забезпечення гарантованих умов роботи транзистора VT3 нижній його електрод (витік) приєднаний до джерела негативної напруги  $E_2$ , величина якої вибрана дещо

більшою, ніж найнижчий рівень вихідної напруги ключа. Транзистор VT3 одночасно генерує струм, необхідний для перезаряду паразитних конденсаторів при низькому рівні вихідної напруги, “підтягуючи” тим самим низький рівень до нуля.

Транзистор VT4, приведений на рис. 2.53, зображений як вхід наступної мікросхеми.

FL-інвертори мають набагато вищі рівні завадостійкості, ніж схеми DCFL. У той же час, вони вимагають двох джерел живлення. FL-інвертори легко можуть бути використані для побудови логічних елементів АБО-НІ шляхом приєднання паралельно VT2 аналогічних транзисторів.

### 2.3.11. Логічні елементи на основі діодів Шоткі (SDFL)

Якщо діоди, що використовуються для забезпечення зсуву рівня вихідної напруги, перенести на вхід логічного елемента, то отримаємо схему, що приведена на рис. 2.54, а. Модифікована схема працює майже повністю так, як її попередній аналог. Отримання логічного елемента, що реалізує функцію АБО-НІ, забезпечується простим приєднанням допоміжного діода Шоткі, паралельно діоду VD1 (рис. 2.54, б).

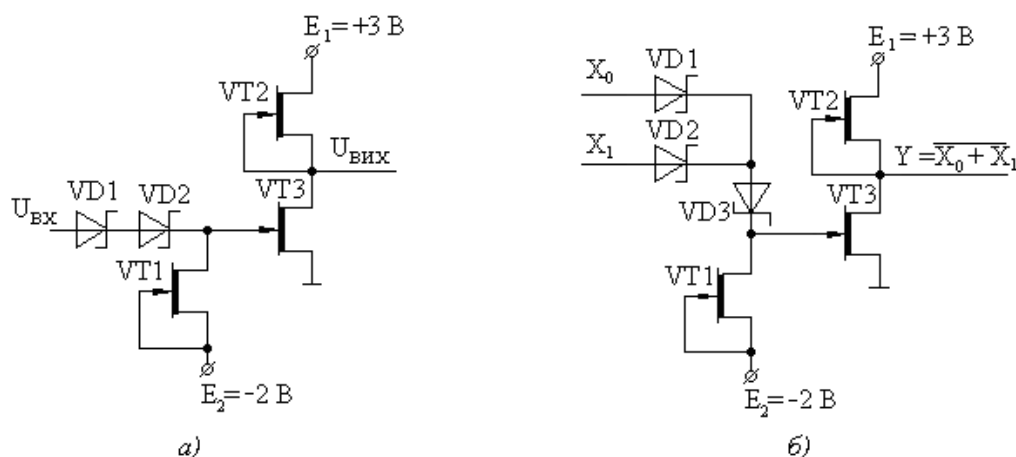


Рис. 2.54

Така схемотехніка відома як *Schottky Diode FET Logic* (SDFL), яка, на відміну від інших MESFET-технологій, займає меншу площу на кристалі,



появою вхідних сигналів і початком їх сприйняття ЛЕ різні, в динамічній моделі повинні бути відображені затримки по кожному з входів і по виходах. Тобто динамічна модель ЛЕ буде мати вигляд, приведений на рис. 2.56, а.

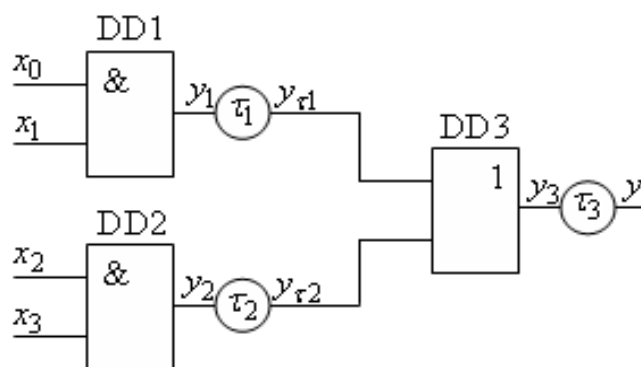


Рис. 2.57

Умовним позначенням логічного елемента зображена його статична модель з входами  $x_0 \dots x_k$  і виходом  $y$ , а динамічні властивості, що обумовлюють затримки сигналів на входах відображаються постійними часу  $\tau_0 \dots \tau_k$ , а відповідна затримка по виходу -  $\tau_y$ .

Недолік такої моделі полягає в тому, що в довідковій літературі на логічні елементи приводяться максимально допустимі затримки розповсюдження сигналу від будь-якого входу до виходу. Тому простіше користуватися спрощеною моделлю, в якій має місце лише одна затримка по виходу (рис. 2.56, б)  $\tau_{ky}$ . Вона передбачає, перш за все, виконання умови, що стани сигналів на входах змінюються не одночасно і не впливають один на інший.

У такій моделі загальна затримка  $\tau_{ky} = \tau_k + \tau_y$  береться з довідкової літератури.

Урахування затримки  $\tau_{ky}$  приводить до того, що логічна функція буде описуватись двома рівняннями:  $y = f(x_0 \dots x_k)$ ;  $y_\tau = y(t + \tau_{ky})$ .

Приведена модель називається *моделлю зі змінною затримкою* і знаходить широке використання у задачах синтезу швидкодіючих скінчених автоматів.

**Приклад 2.4.** Побудувати динамічну модель логічного елемента **2(2І)-АБО**, що описується функцією:  $y = x_0x_1 + x_2x_3$ .

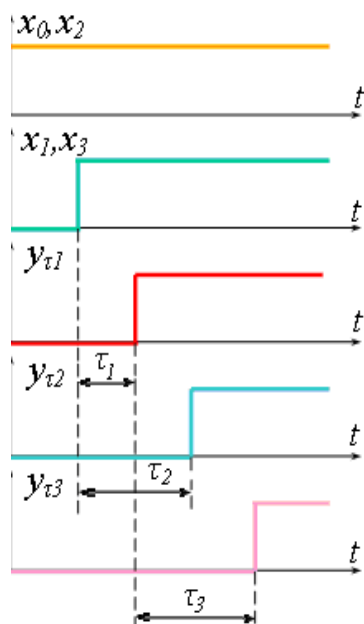


Рис. 2.58

*Розв'язання.* Заданий логічний елемент розглянемо як логічну схему з трьох незалежних елементів, кожен з яких має свою часову затримку (рис. 2.57). Така модель буде справедливою при умові, що зміна вхідних сигналів не співпадає у часі.

**Приклад 2.5.** Побудувати часові діаграми перемикача ЛЕ з попереднього прикладу при умові, що  $x_0 = x_2 = 1$ , сигнали  $x_1$  і  $x_3$  подаються одночасно, а співвідношення між тривалістю часових затримок відповідає нерівності:  $\tau_1 < \tau_2 < \tau_3$ .

*Розв'язання.* Часові діаграми приводяться на рис. 2.58. З них витікає, що величина часової затримки появи вихідного сигналу в даному випадку визначається сумарною затримкою більш швидкодіючого елемента DD1  $\tau_1$  першого ступеня затримкою  $\tau_3$  елемента DD3 другого ступеня. Зрозуміло, що при використанні інших ЛЕ на верхньому рівні, наявність різних часових затримок може привести до появи короткочасних завад. Таке явище розглядалось у першому розділі. Стосовно аналізованого прикладу, при заміні ЛЕ **АБО** елементом **ВИКЛ. АБО** на виході появиться короткочасний імпульс.

## КОНТРОЛЬНІ ПИТАННЯ

1. Дайте пояснення пороговим рівням вхідних напруг логічного елемента.
2. Приведіть фізичну інтерпретацію терміну “порогова напруга”.
3. Дайте визначення термінам “статична завадостійкість” і “динамічна завадостійкість”.
4. Намалюйте часові діаграми перемикання інвертуючого ключа, приведіть усі параметри перемикання, дайте їм пояснення.
5. Чим визначається середній час затримки розповсюдження сигналу?
6. Приведіть схему інвертора на біполярному транзисторі. Дайте пояснення режиму запирання з використанням лінії навантаження транзистора і сім’ї вихідних характеристик.
7. Поясніть особливості положення робочої точки транзистора у відкритому стані.
8. Поясніть позитивні сторони режиму насичення біполярного транзисторного ключа.
9. Поясніть фізичні особливості роботи транзистора у насиченому режимі при наявності дестабілізуючих факторів.
10. Дайте пояснення недолікам у роботі ключа на біполярному транзисторі при наявності режиму насичення збиткових носіїв в області бази.
11. Намалюйте діаграми перемикання інвертора на біполярному транзисторі з поясненням режиму насичення. Поясніть часові параметри перемикання.
12. Намалюйте схему інвертора на біполярному транзисторі з забезпеченням контролю насиченості. Поясніть принцип його роботи. Проведіть аналогію з роботою діодних обмежувачів.
13. Намалюйте схему ключа з динамічним навантаженням. Поясніть принцип роботи і призначення елементів.
14. Поясніть фізичні особливості і переваги використання ключа з динамічним навантаженням.

15. Приведіть принципову схему ключа з перемиканням струму. Поясніть призначення елементів схеми і пояснити принцип її роботи в ключовому режимі.

16. Обґрунтуйте позитивні і негативні сторони роботи диференційного підсилювального каскаду. Які позитивні властивості диференційного підсилювального каскаду використані в схемі з перемиканням струму?

17. Приведіть схему ключового інвертора на польовому транзисторі. Які переваги технології забезпечують інвертору його якісні характеристики?

18. Дайте пояснення особливостям роботи інвертора на польовому транзисторі.

19. Приведіть схему комплементарного МДН-інвертора і поясніть особливості його роботи.

20. Обґрунтуйте позитивні і негативні сторони МДН-інвертора.

21. Дайте визначення терміну *“базовий логічний елемент”*.

22. Поясніть принцип роботи багатоємітерного транзистора. Обґрунтуйте його недоліки при використанні в якості логічного елемента.

23. Намалюйте принципову схему базового логічного елемента ТТЛ серії К155. Дайте пояснення принципу роботи схеми, призначенню елементів схеми.

24. Поясніть, які переваги мають базові логічні елементи ТТЛ ІС останніх розробок.

25. Поясніть вплив ємності навантаження на експлуатаційні параметри ТТЛ ІС. Від чого залежить ємність навантаження мікросхеми?

26. Обґрунтуйте фізичну суть статичної завадостійкості ТТЛ ІС по високому і низькому рівнях сигналів.

27. Дайте пояснення особливості з'єднання виходів ТТЛ ІС, а також використання вільних входів.

28. Намалюйте принципову схему базового елемента ТТЛ з відкритим колектором. Приведіть конкретні схеми підключення до такої схеми



світлодіодів, котушок реле, індикаторних ламп сигналізації. Які умови підключення необхідно виконати в таких схемах.

29. Дайте пояснення терміну “монтажна логіка”. Приведіть приклади схем монтажною логіки для реалізації логічних операцій **АБО, І**.

30. Приведіть принципову схему інвертора КМОН, поясніть призначення її елементів. Назвіть позитивні і негативні властивості інвертора.

31. Приведіть принципову схему інвертора КМОН з трьома станами. Поясніть особливості використання такого інвертора.

32. Приведіть принципову схему двонаправленого ключа КМОН. Пояснити принцип його роботи.

33. Поясніть особливості використання КМОН ІС при роботі на резистивне навантаження.

34. Приведіть схеми і поясніть способи їх побудови, які реалізують логічні операції **АБО, І** з використанням КМОН ІС.

35. Приведіть принципові схеми, які пояснюють області використання двонаправлених ключів. Поясніть їх роботу.

36. Приведіть принципові схеми, які реалізують прості логічні операції на основі монтажною логіки. Поясніть їх роботу та можливі області використання.

37. Поясніть особливості роботи з мікросхемами КМОН.

38. У чому полягають особливості сумісного використання мікросхем ТТЛ і КМОН?

39. Приведіть схему базового логічного елемента і поясніть переваги БіКМОН-технології.

40. Поясніть особливості побудови і приведіть технічні характеристики інтегральної інжекційної логіки. Намалюйте принципову схему базового логічного елемента і поясніть принцип роботи пристрою на основі приведеної схеми.

41. Поясніть принцип роботи базового логічного елемента емітерно-зв’язаною логіки. Обґрунтуйте причини високих динамічних показників ЕЗЛ.

42. Поясніть принцип роботи і області використання приладів з переносом заряду.

43. Поясніть принцип роботи і області використання логічних елементів на основі арсенід-галієвих транзисторів.

## ВПРАВИ І ЗАВДАННЯ

1. При яких рівнях вхідних сигналів ЛЕ **3І-НІ** ТТЛ-типу на його вході буде струм максимальної величини?

2. До виходу МС КР1533 ЛАЗ підключений зовнішній резистор  $R$ . Визначити мінімальне значення опору при включенні резистора  $R$ :

а) між виходом МС та позитивним полюсом джерела живлення;

б) між виходом МС та загальним виводом джерела живлення.

3. Чи коректне висловлювання: «двунаправлених ключів ТТЛ-типу не існує»?

4. Побудувати на основі КМОН інвертора логічні схеми, що реалізують операції **І-НІ**, **АБО-НІ**. Пояснити, яка з них буде більше швидкодіючою?

5. Входи ТТТ ІС, які не використовуються, можуть бути приєднаними до загальної шини, або шини живлення через резистор. В якому випадку буде більшою потужність споживання?

6. Який тип транзисторів використовується при виготовленні КМОН ІС? Привести і пояснити їх струмо-затворні характеристики.

7. Оцінити величини опорів каналів **p** і **n** вихідних транзисторів МС 74СТ257Т, використовуючи довідникові дані на мікросхемі.

8. Світлодіоди при нормальній яркості випромінювання споживають струм 5 мА при падінні напруги 1В. Визначити величини баластних резисторів для КМОН і ТТЛ ІС при різних схемах включення.

9. Використовуючи довідникові дані, визначити порогові рівні завад TTL IC 74LS00 при високому і низькому рівні вхідного сигналу. Дати порівняння TTL IC з точки зору їх завадостійкості при низькому і високому рівнях вхідного сигналу.

10. Обчислити можливий коефіцієнт розгалудження для кожної з наступних пар мікросхем TTL- типу для сигналів високого і низького рівнів:

а) 74LS керує 74LS;

б) 74LS керує 74S.

Дати пояснення.

11. Як ємність навантаження впливає на роботу мікросхем КМОН і TTL типу? Дати пояснення.

12. Мікросхеми КМОН призначені для роботи при напрузі 5В. Пояснити що зміниться в роботі мікросхем, якщо живлення забезпечувати напругою 3В? Дати пояснення.

13. Чому мікросхеми з трьома станами працюють на вимикання швидше, ніж на вмикання?

14. Для кожного з приведених нижче значень резистивних навантажень визначити, чи відповідають вони допустимим навантаженням для МС серії 74LS00?

а) резистор з опором 470 Ом включений між виходом та шиною +E;

б) резистор з опором 390 Ом включений між виходом та шиною +E, одночасно резистор з опором 510 Ом включений між виходом і загальною шиною.

Дати пояснення.

15. Розглянемо динамічний режим роботи КМОН IC при роботі на ємнісне навантаження. Якщо опір верхнього вихідного транзистора з р-каналом вдвічі більший за опір нижнього транзистора з n-каналом, то чи буде час фронту в два рази більше часу спаду? Привести пояснення.

16. Привести схему 4-входового КМОН елемента АБО-НІ і оцінити його швидкодію.

17. Використовуючи довідникові дані визначити величини опорів каналів транзисторів вихідного інвертора мікросхем КМОН серії 74АС.

18. Що буде, якщо до виходу ТТТ-елемента підключити таку кількість входів елементів цього ж типу, яка перевищує допустиму?

19. В якому стані буде знаходитись вихід ТТЛ елемента **I-НІ**, якщо всі його входи не задіяні (нікуди не підключені)?

### Розділ 3

## ПРИНЦИПИ ПОБУДОВИ ТА ВИКОРИСТАННЯ КОМБІНАЦІЙНИХ ПРИСТРОЇВ

### 3.1. Використання базових логічних елементів

Розглядаючи серії мікросхем з різними технологіями виготовлення та різних фірм-виробників, легко встановити високий рівень уніфікації наборів типових логічних елементів та їх технічних характеристик.

Разом з *однофункціональними* елементами (**НІ**, **I**, **АБО**), широко використовуються *двофункціональні* (**I-НІ**, **АБО-НІ**, **ВИКЛ. АБО**). Більш обмежено використовуються *трьохфункціональні* (**I-АБО-НІ**) з різним рівнем розширення. Вказаний набір комбінаційних мікросхем з повного набору серії, що зветься *бібліотекою*, є *базовим набором* або *базовими логічними елементами*. Більш того, проєктовані матричні великі інтегральні схеми (**ВІС**) орієнтуються також на використання логічних функцій, тотожних базовим логічним елементам.

Серед задач, які часто зустрічаються при проєктуванні логічних схем, слід виділити наступні:

- реалізація функцій у базисах **I-НІ**, **АБО-НІ**;
- розширення логічних можливостей елементів.

Задача реалізації логічних функцій розв'язується шляхом використання правила подвійного заперечення та теореми де Моргана. Проілюструємо це на наступних прикладах.

**Приклад 3.1.** Використовуючи елементи **I-НІ**, реалізувати функцію:

$$y = x_1 \cdot \overline{x_2} + (\overline{x_2} + \overline{x_3}) \cdot \overline{x_1}. \quad (3.1)$$

*Розв'язання.* На основі правила подвійного заперечення функція (3.1) може бути зображена у вигляді:

$$y = \overline{\overline{x_1 \cdot x_2 \cdot (x_2 + x_3) \cdot x_1}}.$$

Враховуючи, що  $\overline{x_2 + x_3} = \overline{x_2} \cdot \overline{x_3}$ , логічну схему можемо зобразити у вигляді, приведеному на рис. 3.1.

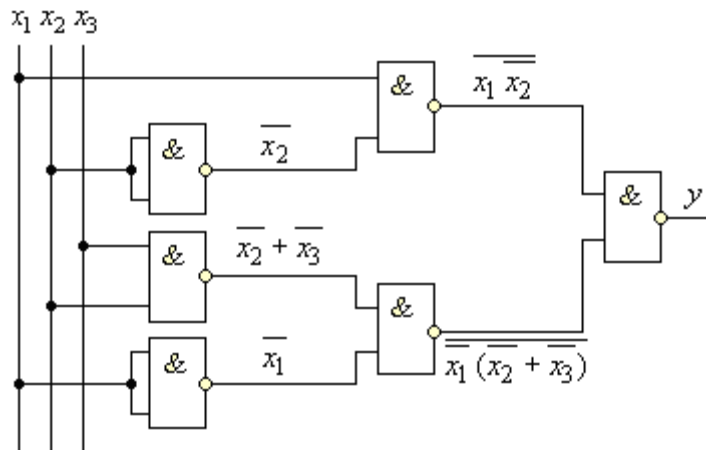


Рис. 3.1

**Приклад 3.2.** Використовуючи елементи АБО-НІ, реалізувати функцію:

$$y = x_1 \cdot (\overline{x_1 + x_2}) \cdot (x_2 + \overline{x_3} \cdot x_4). \quad (3.2)$$

*Розв'язання.* За аналогією з попереднім прикладом, вираз зобразимо у вигляді:

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

Реалізація функції (3.2) приведена на рис. 3.2. Враховано, що

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

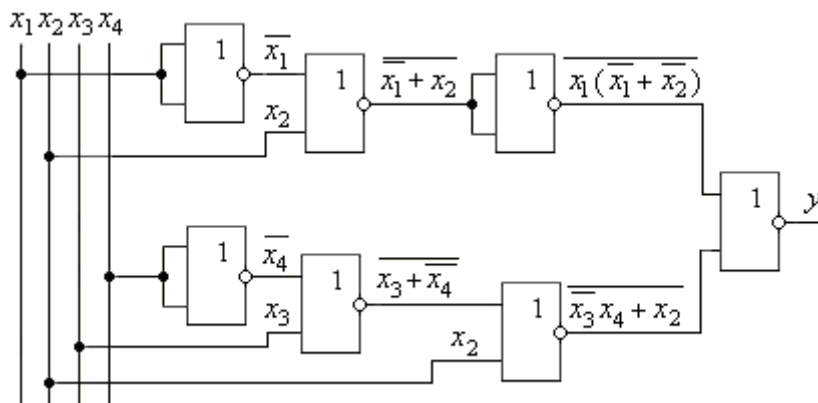


Рис. 3.2

Вирішення задачі розширення логічних можливостей елементів розглянемо на наступних прикладах.

**Приклад 3.3.** Збільшити коефіцієнт розгалуження по виходу  $K_p$  на величину, перевищуючу значення стандартного елемента.

*Розв'язання.* У залежності від величини необхідного коефіцієнта розгалуження, використовуються три варіанти схем, наведені на рис. 3.3, а, б, в.

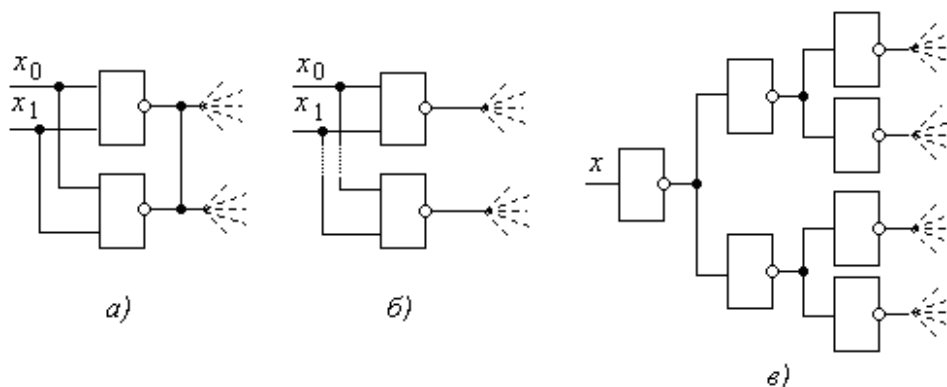


Рис. 3.3

Перший з можливих варіантів – запаралелення двох елементів (рис. 3.3, а). Така схема підвищення потужності виходу ТТЛ-мікросхем використовується лише для двох елементів, що знаходяться в одному корпусі. Збільшення їх кількості призведе до появи вирівнюючих струмів. У КМОН-серіях такого обмеження принципово немає, але слід пам'ятати, що зростання кількості запаралелених елементів приведе до підвищення величини вихідної паразитної ємності.

Схема, наведена на рис. 3.3, б, за кількістю елементів для КМОН ІС також не має обмежень, а для ТТЛ обмеження задається потужністю джерел сигналів  $x_0$  та  $x_1$ .

Схема рис. 3.3, в каскадного розмноження кількості виходів фактично не має ніяких обмежень і використовується в схемах для розводки кіл синхронізації.

**Приклад 3.4.** Використовуючи елементи ТТЛ ІС, розробити схему **I** на 9 входів.

*Розв'язання.* У серіях ТТЛ використовуються багатовхідні елементи **I** 4(**2I**), 3(**3I**), 2(**4I**), а також елементи **I-НІ** – 2(**4I-НІ**), 1(**8I-НІ**), 3(**3I-НІ**) (позначення 2(**4I-НІ**) говорить про те, що в одному корпусі мікросхеми наявні два чотирьохходові логічні елементи **I-НІ**).

При використанні елементів 3(**3I**) (наприклад, мікросхеми К555ЛИЗ (зарубіжні аналоги – 74LS11, 74LS11N, DL011D, UCY74LS11N)) використовується каскадне з'єднання елементів.

При використанні елементів 3(**3I-НІ**) логічну функцію:  $y = x_0 \cdot x_1 \cdot x_2 \cdot x_3 \cdot x_4 \cdot x_5 \cdot x_6 \cdot x_7 \cdot x_8$  перетворимо до виду:

$$y = \overline{\overline{x_0 \cdot x_1 \cdot x_2} + \overline{x_3 \cdot x_4 \cdot x_5} + \overline{x_6 \cdot x_7 \cdot x_8}} \quad (3.3)$$

З (3.3) витікає, що необхідна схема може бути створена на базі мікросхеми 3(**3I-НІ**) (К1533ЛА4), виходи яких об'єднуються елементом 3(**АБО-НІ**) (К1533ЛЕ4).

**Приклад 3.5.** Використовуючи елементи ТТЛ ІС, розробити 9-входову схему АБО.

*Розв'язання.* У серіях ТТЛ ІС наявні мікросхеми 4(**2АБО-НІ**) (7402DC, 7402FC, T7428B1), 3(**3АБО-НІ**) (7427DC, 7427PC, SN7427N), 2(**5АБО-НІ**).

При використанні елементів 3(**3АБО-НІ**) логічну функцію:

$$y = x_0 + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8$$

перетворимо до виду:

$$y = \overline{\overline{(x_0 + x_1 + x_2)} \cdot \overline{(x_3 + x_4 + x_5)} \cdot \overline{(x_6 + x_7 + x_8)}}. \quad (3.4)$$

Функцію (3.4) на практиці можна реалізувати шляхом використання трьох елементів **3АБО-НІ** (мікросхема К1533ЛЕ4 (зарубіжні аналоги – 7427PC, MC7427N, SN7427N)), виходи яких об'єднуються одним елементом **3І-НІ** (К1533ЛА4).

**Приклад 3.6.** Побудувати схему знаходження суми за модулем 2 на вісім входів, використовуючи двовходові елементи ТТЛ, які реалізують таку ж функцію.

*Розв'язання.* У серіях ТТЛ ІС використовуються мікросхеми з двома входами К555ЛПІ5 (зарубіжні аналоги – 74LS86N, SN74LS86J), КР1533ЛПІ5 (аналог – 74ALS86) та К555ЛЛІЗ (вихід останньої – з відкритим колектором). Вони реалізують наступну логічну функцію:

$$y = x_1 \overline{x_0} + \overline{x_1} x_0 = x_1 \oplus x_0.$$

Логічна функція, реалізацію якої поставлено задачею, має вигляд:

$$y = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5 \oplus x_6 \oplus x_7. \quad (3.5)$$

Для побудови пристрою, що реалізує функцію (3.5), скористаємось сполучним законом, завдяки якому виконаємо два варіанти перетворень:

$$y = (\dots(((x_0 \oplus x_1) \oplus x_2) \oplus x_3) \oplus x_4) \oplus x_5) \oplus x_6) \oplus x_7); \quad (3.6)$$

$$y = ((x_0 \oplus x_1) \oplus (x_2 \oplus x_3)) \oplus ((x_4 \oplus x_5) \oplus (x_6 \oplus x_7)). \quad (3.7)$$

Формула (3.6) дає можливість створити послідовну схему (рис. 3.4, а); (3.7) – відповідно, пірамідальну (рис. 3.4, б). Обидві схеми по апаратних затратах еквівалентні і виконуються з використанням двох корпусів, наприклад, мікросхеми К555ЛПІ5. Але величини часових затримок у них суттєво різні.

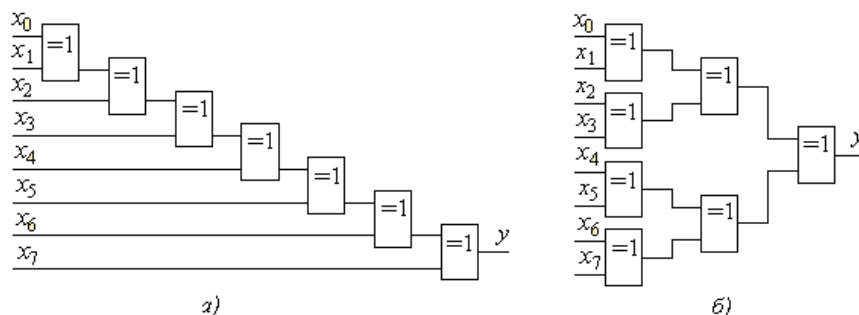


Рис. 3.4



У першому випадку величина затримки виконання операції  $t_3$  пов'язана з затримкою елемента  $t_{30}$  та їх кількістю  $m$  співвідношенням:  $t_3 = t_{30} (m - 1)$ .

При пірамідальному з'єднанні величина  $t_3$  суттєво менша і знаходиться за формулою:

$$t_3 = t_{30} \log_2 m.$$

Перша з розглянутих схем (рис. 3.4) використовується, також, для виділення фронту та спаду імпульсу, для подвоєння частоти імпульсів.

Ці властивості ілюструються схемою та часовими діаграмами, що приведені на рис. 3.5, а, б. Елементи DD1.1...DD1.3 виконують затримку імпульсу  $U_{вх}$  на величину  $t_3 = 3t_{30}$  а DD1.4 виділяє імпульси  $U_{вих}$  відповідно до логіки роботи елемента.

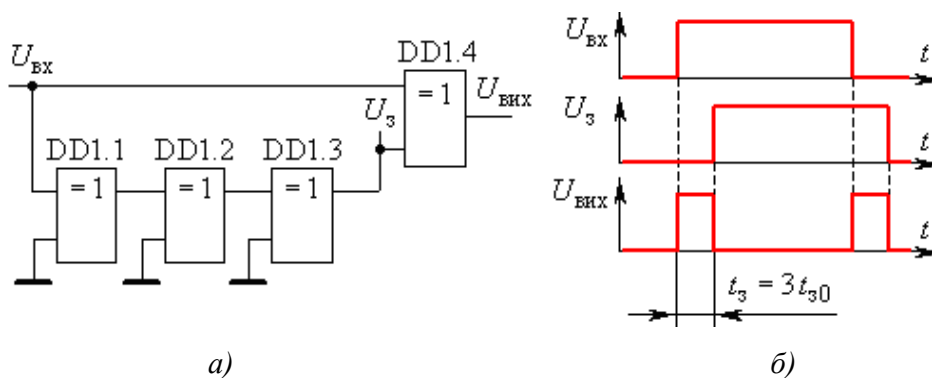


Рис. 3.5

Схема, приведена на рис. 3.4, б, є основою схем, призначених для перевірки на парність багаторозрядних цифрових слів (наприклад, мікросхем K1533ИП2 (зарубіжні аналоги – 74180РС, CDB4180E, SN74180N, UCY74180N) та K561CA1 (зарубіжні аналоги – 4531BDC, GD4531BC, GD4531BDC, GD4531BDM)), які широко використовуються в комп'ютерній техніці.

**Приклад 3.7.** Використовуючи елементи **ВИКЛ. АБО**, розробити схему пристрою для вмикання (вимикання) освітлення в приміщенні з трьома входами, біля кожного з яких повинен розміщуватись вимикач.

*Розв'язання.* Складаємо таблицю станів функції, що реалізується пристроєм, прийнявши за логічні змінні стани вимикачів – відповідно  $x_2, x_1, x_0$  (див. табл. 3.1).

Прийнявши, що при вимкнених вимикачах лампочка освітлення не горить, тобто  $y = 0$  при  $x_0 = 0, x_1 = 0, x_2 = 0$ , істинні значення функції будуть лише при непарній кількості ввімкнених вимикачів, тобто для мінтермів з порядковими номерами 1, 2, 4, 7.

Таблиця 3.1

N	$x_2$	$x_1$	$x_0$	y
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	1

Логічна функція буде описуватись наступним виразом:

$$y = \bigvee 1, 2, 4, 7 = \overline{x_2} \overline{x_1} x_0 + \overline{x_2} x_1 \overline{x_0} + x_2 \overline{x_1} \overline{x_0} + x_2 x_1 x_0 = \\ = \overline{x_2} (\overline{x_1} x_0 + x_1 \overline{x_0}) + x_2 (x_1 x_0 + \overline{x_1} \overline{x_0}) = x_2 \oplus x_1 \oplus x_0 ,$$

оскільки  $\overline{\overline{x_1} x_0 + x_1 \overline{x_0}} = x_1 x_0 + \overline{x_1} \overline{x_0}$ .

Схема пристрою може бути реалізована за одним з двох варіантів, приведених на рис. 3.4, а, б.

## 3.2. Комутатори

*Комутатором* (у загальному випадку на  $n$  входів та  $m$  виходів – скорочено  $n \times m$ ) називають цифровий пристрій, який забезпечує передачу сигналів з  $i$ -го вхідного на  $j$ -ий вихідний виводи.

У цифровій схемотехніці найбільшого поширення набули такі комутуючі пристрої, як *мультиплексори*, *селектори-мультиплексори* (з організацією  $n \times 1$ ) та *пристрої зсуву* ( $n \times n$ ).

### 3.2.1. Мультиплексори

При передачі цифрової інформації виникає велика кількість ситуацій, коли цифровий двійковий код необхідно перетворити з паралельного формату в послідовний з метою його запису або передачі по двопроводній лінії. Як приклад, пристрої запису інформації – накопичувачі на компакт-дисках, гнучкі та жорсткі диски – використовують послідовний формат. З послідовним форматом працюють COM-порти ПЕОМ, модеми, USB-шини.

Розглянемо умови і принцип перетворення двійкового коду з паралельного формату у послідовний.

Припустимо, що по однобайтовій шині даних до пристрою підводиться двійкове слово  $d_7 d_6 d_5 d_4 d_3 d_2 d_1 d_0 = A3_{16}$  (рис. 3.6), яке пристрій повинен зчитати і передати в послідовному форматі на вихід  $Y$ .

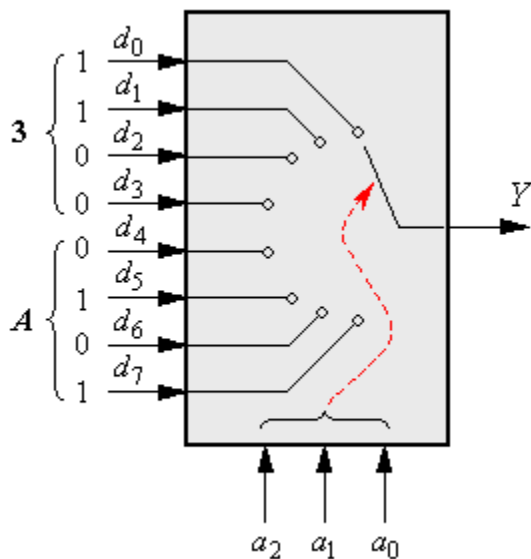


Рис. 3.6

Таблиця 3.2

$N$	$a_2$	$a_1$	$a_0$	$Y$	Код слова
0	0	0	0	$d_0$	1
1	0	0	1	$d_1$	1
2	0	1	0	$d_2$	0
3	0	1	1	$d_3$	0
4	1	0	0	$d_4$	0
5	1	0	1	$d_5$	1
6	1	1	0	$d_6$	0
7	1	1	1	$d_7$	1

Щоб виконати таку операцію, необхідно послідовно звертатись (адресуватись) до кожного біта даних і по черзі приєднувати його до виходу  $Y$ . Зрозуміло, що для забезпечення такої адресації необхідно мати три розряди адресної шини. У механічному еквіваленті адресна шина визначає короткочасне положення бігунка перемикача, який по черзі з'єднує вхід шини даних з виходом  $Y$  пристрою.

Роботі пристрою, що зображений на рис. 3.6, відповідає таблиця станів (табл. 3.2).

Якщо вихідний код  $Y$  розглядати як функцію вхідних сигналів пристрою, то отримаємо наступну залежність:

$$Y = \overline{a_2} \overline{a_1} \overline{a_0} d_0 + \overline{a_2} \overline{a_1} a_0 d_1 + \overline{a_2} a_1 \overline{a_0} d_2 + \overline{a_2} a_1 a_0 d_3 + \\ + a_2 \overline{a_1} \overline{a_0} d_4 + a_2 \overline{a_1} a_0 d_5 + a_2 a_1 \overline{a_0} d_6 + a_2 a_1 a_0 d_7 .$$

Якщо зміну сигналів  $a_2$ ,  $a_1$ ,  $a_0$  забезпечити послідовно в часі з тактовим інтервалом  $T$ , то можемо побудувати часову діаграму (рис. 3.7), яка встановлює послідовність передачі сигналів на входах  $d_7 \dots d_0$  на вихід  $Y$ .

Звідси витікає, що *мультиплексор* – це цифровий комбінаційний пристрій, який забезпечує перетворення інформації, що задана на входах  $d_7 \dots d_0$  у паралельному форматі, в послідовний формат на виході  $Y$ .

Умовне позначення мультиплексора в електронних схемах приводиться на рис. 3.8. Входи  $A$  називають адресними (адресна шина), а входи  $D$  – входами даних (шина даних). Допоміжний вхід  $V$  називають керуючим або *дозволяючим*.

З його урахуванням робота мультиплексора, приведеного на рис. 3.8, описуватиметься наступною логічною функцією:

$$Y = \bar{v} \cdot (\bar{a}_1 \bar{a}_0 d_0 + \bar{a}_1 a_0 d_1 + a_1 \bar{a}_0 d_2 + a_1 a_0 d_3) . \quad (3.8)$$

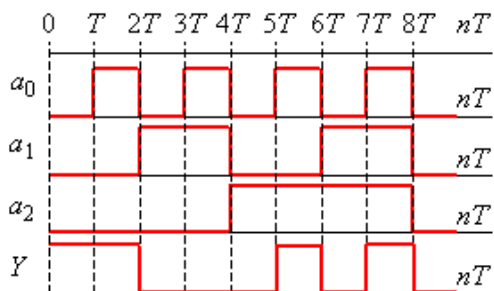


Рис. 3.7

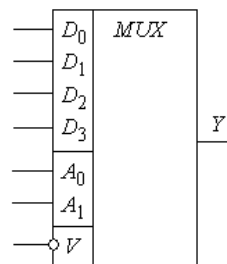


Рис. 3.8

Дозволяючий вхід  $V$  розширює можливості пристрою. Він дає можливість синхронізувати роботу мультиплексора з іншими схемами, а також використовується для нарощування розрядності адресних сигналів. Логічна схема мультиплексора, реалізована у базисі елементів **I-АБО-НІ** у відповідності до формули (3.8), приведена на рис. 3.9.

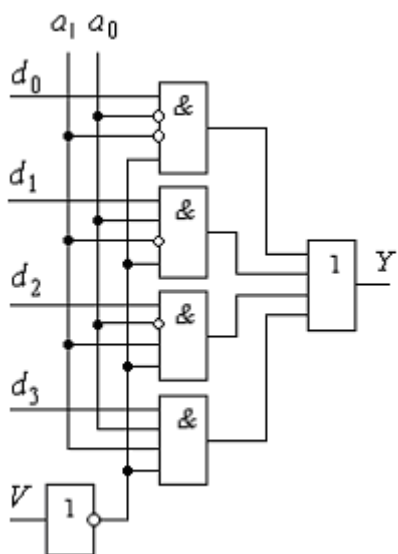


Рис. 3.9

Кількість входів шини даних мультиплексора відображається у його назві. Наприклад, розглянута схема відповідає назві “з чотирьох в один”, або просто “**4:1**”. В залежності від кількості інформаційних входів, вони бувають “**8:1**”, “**16:1**” і т. д. Технічні характеристики мультиплексорів визначаються статичними й динамічними характеристиками базових ЛЕ відповідних серій з урахуванням складності схеми.

Оскільки мультиплексор є пристроєм логіки, то на його базі можуть реалізовуватись різні логічні функції. Як приклад, скористаємось формулою (3.8). Припустимо, що необхідно реалізувати логічну функцію:

$$y = x_0 \cdot x_1 + \overline{x_0} \cdot \overline{x_1}. \quad (3.9)$$

Порівнюючи формули (3.8) і (3.9), бачимо, що, присвоївши адресним сигналам  $a_0, a_1$  значення  $x_0, x_1$  відповідно і забезпечивши виконання умов  $d_1 = d_2 = 0, d_0 = d_3 = 1, v = 0$ , одержимо схему пристрою (рис. 3.10), який реалізує функцію (3.9).

Нескладні перетворення логічних функцій дають можливість реалізовувати на мультиплексорах функції з кількістю змінних, що перевершує кількість адресних входів.

Виконується наступна послідовність перетворень:

- у мінімізованій диз'юнктивній нормальній формі функції виділяються змінні, які мають найвищий ранг, тобто повторюються в найбільшій кількості диз'юнкцій. Наприклад, вибирається  $m$  змінних:

- виконується перетворення функції так, щоб забезпечити присутність виділених змінних в усіх диз'юнкціях. Для цього кожна диз'юнкція домножається на  $1 = x + \overline{x}$ ;

- проводиться перегрупування змінних логічної функції таким чином, щоб виділені змінні були винесені за дужки у відповідних групах диз'юнкцій.

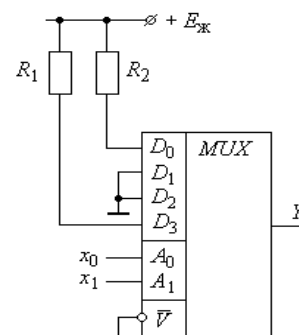


Рис. 3.10

У результаті виконаних перетворень одержується формула вихідної функції, яка реалізується на мультиплексорі “з  $M = 2^m$  в 1”, якщо на його адресні входи подати виділені змінні, а на інформаційних входах забезпечується виконання тих логічних виразів, які залишились в дужках. Якщо вирази в дужках є функціями декількох змінних, то їх реалізація може бути забезпечена або за допомогою іншого мультиплексора, або на основі типових логічних елементів.

**Приклад 3.8.** Функцію п'яти змінних:

$$y = x_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 + \bar{x}_0 \cdot \bar{x}_3 \cdot x_4 + x_0 \cdot \bar{x}_2 \cdot x_3 + x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2$$

реалізувати з використанням мультиплектора “**3 4-х в 1**”.

*Розв'язання.* У використовуваному мультиплексорі наявні два адресні входи – отже, вибираємо дві змінні з найбільшими рангами. Ранги змінних приведені в табл. 3.3.

Таблиця 3.3

$x_0$	$x_1$	$x_2$	$x_3$	$x_4$
3	2	3	4	2

Вибираємо змінні  $x_2$  та  $x_3$ .

Виконуємо перетворення функції:

$$\begin{aligned} y &= x_0 \cdot x_1 \cdot \bar{x}_3 \cdot \bar{x}_4 \cdot (x_2 + \bar{x}_2) + \bar{x}_0 \cdot \bar{x}_3 \cdot x_4 \cdot (x_2 + \bar{x}_2) + x_0 \cdot \bar{x}_2 \cdot x_3 + x_2 \cdot \bar{x}_3 + \bar{x}_1 \cdot x_2 \cdot (x_3 + \bar{x}_3) = \\ &= \bar{x}_2 \cdot \bar{x}_3 \cdot (x_0 \cdot x_1 \cdot \bar{x}_4 + \bar{x}_0 \cdot x_4) + \bar{x}_2 \cdot x_3 \cdot (x_0) + x_2 \cdot x_3 \cdot (\bar{x}_1) + x_2 \cdot \bar{x}_3 \cdot (x_0 \cdot x_1 \cdot \bar{x}_4 + \bar{x}_0 \cdot x_4 + 1 + \bar{x}_1) = \\ &= \bar{x}_2 \cdot \bar{x}_3 \cdot (x_0 \cdot x_1 \cdot \bar{x}_4 + \bar{x}_0 \cdot x_4) + \bar{x}_2 \cdot x_3 \cdot (x_0) + x_2 \cdot x_3 \cdot (\bar{x}_1) + x_2 \cdot \bar{x}_3 \cdot (1) \end{aligned}$$

Логічна сума в перших дужках також реалізується за допомогою мультиплектора **4:1**, якщо на адресні входи його подати змінні  $x_0$  та  $x_4$ .

Принципова схема пристрою, що реалізує потрібну логічну функцію, приведена на рис. 3.11. Але така схема є лише прикладом і на практиці не піддається реалізації, оскільки в серіях мікросхем відсутні мультиплектори **4:1**, а спарений мультиплексор K1533КП2 (зарубіжні аналоги – 74153РС, SN74153N) має об'єднані адресні входи.

Практично задача може бути розв'язана при використанні мультиплектора **8:1**.

Вибравши логічні змінні  $x_0, x_2, x_3$  як змінні з найвищими рангами (Див. табл. 3.3), функцію перетворимо до наступного вигляду:

$$\begin{aligned} y &= \bar{x}_0 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot (x_4) + x_0 \cdot \bar{x}_2 \cdot \bar{x}_3 \cdot (x_1 x_4) + \bar{x}_0 \cdot x_2 \cdot \bar{x}_3 \cdot (1) + \\ &+ x_0 \cdot x_2 \cdot \bar{x}_3 \cdot (1) + \bar{x}_0 \cdot x_2 \cdot x_3 \cdot (0) + x_0 \cdot \bar{x}_2 \cdot x_3 \cdot (1) + \bar{x}_0 \cdot x_2 \cdot x_3 \cdot (\bar{x}_1) + x_0 \cdot x_2 \cdot x_3 \cdot (\bar{x}_1). \end{aligned}$$

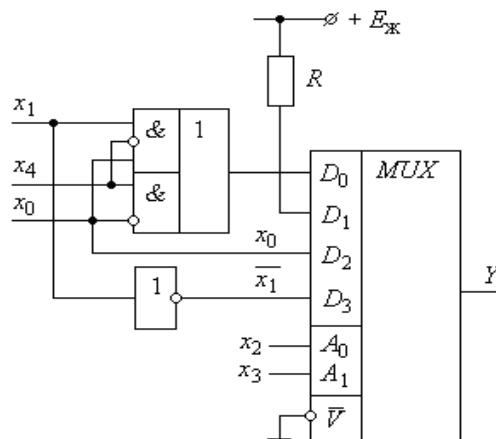


Рис. 3.11

У табл. 3.4 зведені значення необхідних сигналів, які подаються на відповідні інформаційні входи.

Таблиця 3.4

$N$	$x_3$	$x_2$	$x_0$	$D_i$
0	0	0	0	$x_4$
1	0	0	1	$x_1 \cdot \overline{x_4}$
2	0	1	0	1
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	$\overline{x_1}$
7	1	1	1	$\overline{x_1}$

На рис. 3.12 приведена схема, що реалізує задану функцію на мультиплексорі K1533КП7 (його зарубіжні аналоги – мікросхеми MC54151W, SN74151N). Необхідні допоміжні логічні функції реалізуються з використанням одного корпусу мікросхеми K1533ЛА3 (зарубіжні аналоги – 7400РС, МН7400, SN7400N).

**Приклад 3.9.** Використовуючи мультиплексор K564КП2 **8:1**, реалізувати логічну функцію:

$$y = \vee 0, 2, 3, 5, 7.$$

*Розв'язання.* Оскільки ЛФ має три змінні, то вони повинні бути приєднані до адресних входів  $A_0 \div A_2$  (рис. 3.13). Щоб дозволити роботу мультиплексора, вхід  $\overline{V}$  повинен бути заземленим. Входи даних  $d_0, d_2, d_3, d_5, d_7$  необхідно приєднати до шини живлення, а решту входів – заземлити. В результаті схема пристрою може бути реалізована на мультиплексорі K564КП2 (зарубіжні аналоги якого – 4051BDC, 4051BDM).

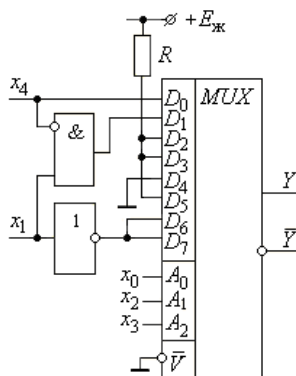


Рис. 3.12.

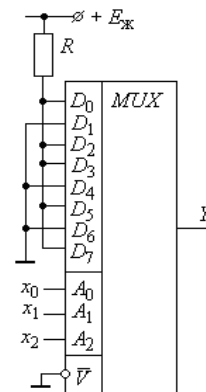


Рис. 3.13

У більшості серій мікросхем мультиплексори, що виготовляються у вигляді самостійних виробів, мають кількість інформаційних входів, не перевищуючу шістнадцяти. Збільшення їх кількості досягається різними прийомами з'єднання окремих мультиплексорів. Наприклад, на рис. 3.14 приведена схема паралельного нарощування.

Для збільшення кількості інформаційних входів необхідно підвищувати кількість адресних провідників (ємність адресного простору, адресної шини).

У схемі, приведеній на рис. 3.14, розширення адресного простору досягається використанням входу  $V$ , який, об'єднаний в обох мультиплексорах через інвертор DD3, виступає старшим розрядом адресної шини. При  $a_2 = 0$  зміна сигналів на  $a_0, a_1$  дозволить вибирати входи  $d_0 \dots d_3$  шини даних і через елемент АБО (DD4) передавати на вихід  $Y$ .

Мультиплексор DD2 при цьому відключений високим рівнем сигналу на виході DD3. При  $a_2 = 1$  зміною  $a_0, a_1$  забезпечується комутація входів  $d_4 \dots d_7$  на вихід  $Y$ .

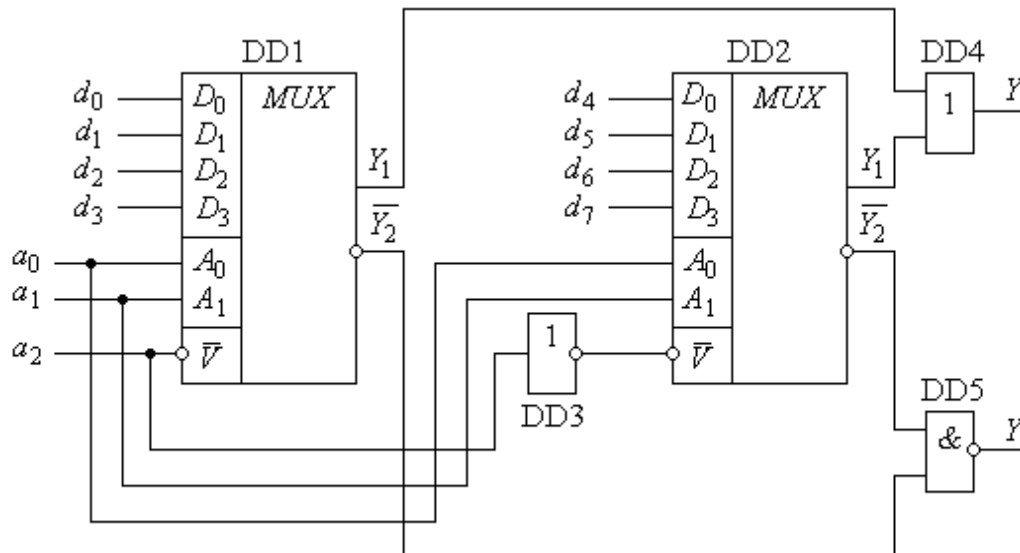


Рис. 3.14

При наявності в мультиплексорах інверсних виходів їх об'єднання забезпечується елементом І-НІ (DD5).



При необхідності суттєвого нарощування кількості комутованих інформаційних входів використовується пірамідальний спосіб нарощування. Приклад його реалізації приведений на рис. 3.15. Ця схема в реалізації є більш гнучкою.

Адресні входи мультиплексорів нижнього рівня DD1...DD4 з'єднуються паралельно і керуються молодшими розрядами адресного простору. Їх кількість визначається тільки кількістю інформаційних входів кожного мультиплексора. Мультиплексор верхнього рівня керується старшими розрядами адресного простору, завдяки яким він забезпечує комутацію виходу кожного з мультиплексорів  $Y_0 \dots Y_3$  на загальний вихід  $Y$ .

При використанні інших комбінаційних пристроїв – наприклад, дешифраторів, про які мова піде нижче – з'являються більш широкі можливості нарощування розрядності інформаційної та адресної шин мультиплексорів.

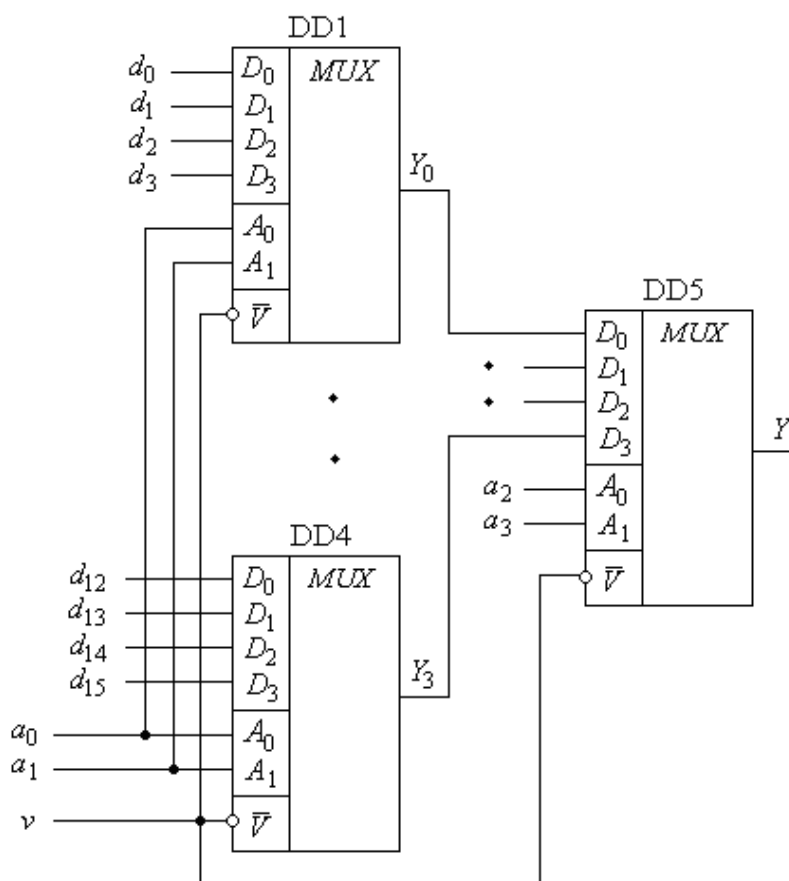


Рис. 3.15

### 3.2.2. Пристрої зсуву

Пристрої зсуву – це комбінаційні пристрої, призначені для забезпечення передачі вхідного коду на вихід зі зсувом на декілька розрядів у залежності від керуючого коду. Забезпечення зсуву двійкових слів необхідно при побудові арифметичних пристроїв, у задачах узгодження шин даних при формуванні заданих часових послідовностей сигналів та ін.

Припустимо, що нам необхідно спроектувати пристрій, на вхід якого подаються дані по лініях  $d_3 d_2 d_1 d_0$ , котрі необхідно передавати на вихідні лінії  $y_3 y_2 y_1 y_0$  зі зсувом від **0** до **3** розрядів у залежності від значення адресного двійкового коду  $A$  (рис. 3.16). Оскільки при одночасному виконанні зсуву можливі лише чотири комбінації, то для їх забезпечення достатньо мати два розряди адресного слова  $A = a_1 a_0$ . Тоді функціональні залежності для кожного з виходів  $y_i$  матимуть наступний вигляд:

$$\begin{aligned}
 y_0 &= d_0 \overline{a_1} \overline{a_0} + d_1 \overline{a_1} a_0 + d_2 a_1 \overline{a_0} + d_3 a_1 a_0; \\
 y_1 &= d_1 \overline{a_1} \overline{a_0} + d_2 \overline{a_1} a_0 + d_3 a_1 \overline{a_0} + d_0 a_1 a_0; \\
 y_2 &= d_2 \overline{a_1} \overline{a_0} + d_3 \overline{a_1} a_0 + d_0 a_1 \overline{a_0} + d_1 a_1 a_0; \\
 y_3 &= d_3 \overline{a_1} \overline{a_0} + d_0 \overline{a_1} a_0 + d_1 a_1 \overline{a_0} + d_2 a_1 a_0.
 \end{aligned}
 \tag{3.10}$$

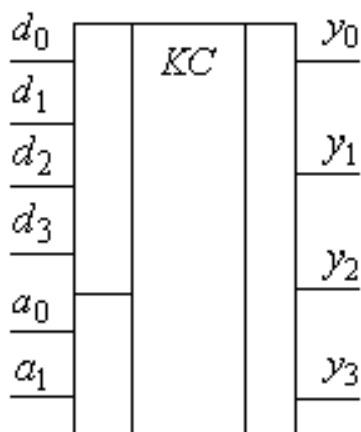


Рис. 3.16

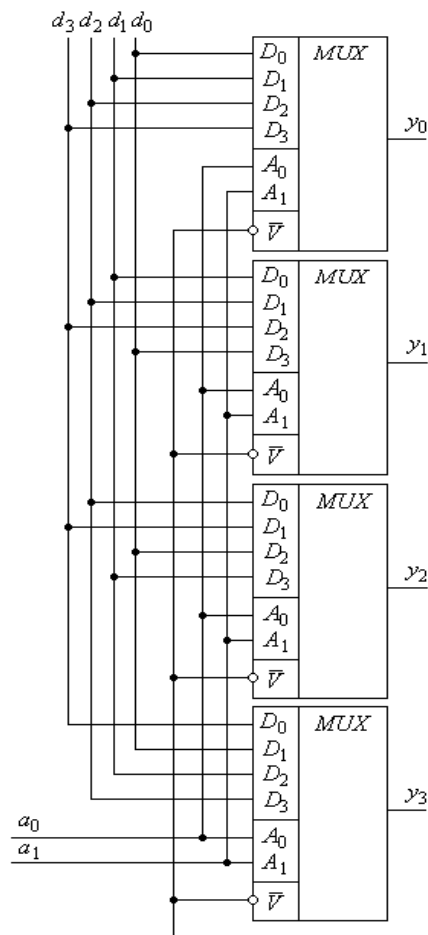


Рис. 3.17

З формул (3.10) витікає, що схема зсуву складається з чотирьох мультиплексорів, адресні входи яких з'єднані паралельно, а входи даних приєднані до загальної шини даних зі зсувом на один розряд (рис. 3.17).

### 3.2.3. Селектори-мультиплексори

Порівняно з мультиплексорами ТТЛ, мультиплексори, виготовлені за КМОН-технологією, мають суттєву різницю, яка полягає в використанні двонаправлених ключів, керованих через спеціальний перетворювач кодів. Тому схема комутації такого приладу має вигляд, приведений на рис. 3.18.

Особливість такої схеми полягає у тому, що ключі DD5...DD8, як відмічалось у **Розділі 2**, мають двонаправлену провідність, низький опір і однаково проводять як імпульсні, так і аналогові сигнали.

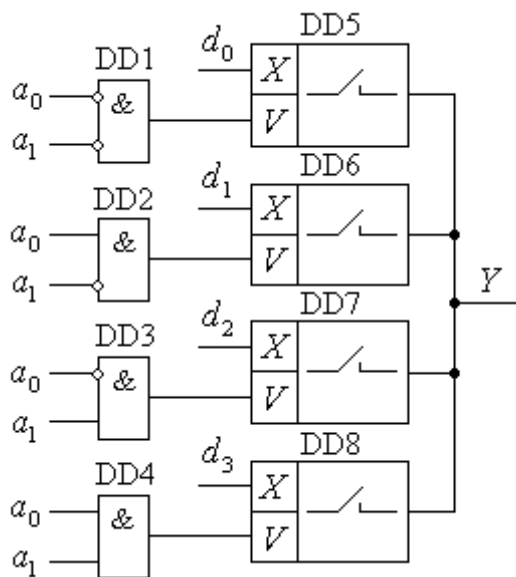


Рис. 3.18

Це відкриває можливість використовувати їх не тільки для комутації входів  $d_0 \div d_3$  на вихід  $Y$ , а й розв'язувати ряд задач іншого характеру.

Одним з таких напрямів використання є *селектори-мультиплексори* – прилади, які дають можливість забезпечувати не тільки побітну цифрову передачу інформації, а й вибирати цифрові потоки інформації з будь-якого входу на вихід.

Інший приклад використання таких приладів – це застосування їх у якості демультимплексорів. Робота у такому режимі полягає в тому, що двійковий цифровий код може подаватись на вхід  $Y$ , а зніматись побітово з виходів  $d_0 \div d_3$ , забезпечуючи тим самим перетворення послідовного формату в паралельний.

Прикладом мультиплексора КМОН, що побудований за таким принципом, можна назвати мікросхему К564КП2 (4051BDC, CD4051BCM) – мультиплексор **8:1** і мультиплексор **2(4:1)**.

### 3.3. Кодуючі та декодуючі пристрої

Велика різноманітність кодів, які використовуються при представленні і обробці інформації (див. **Розділ I**), вимагає наявності спеціальних пристроїв для перетворення інформації з однієї форми на іншу. Найбільш широко використовується кодування десяткових цифр двійковим кодом, переведення двійкового коду в код семисегментних індикаторів та ряд інших. Найпростішим перетворювачем кодів можна вважати кодер, який, незалежно від коду, що подається на його вхід (прямий чи зворотний), на виході завжди видає прямий код (рис. 3.19).

Схема в кожному біті перетворювача реалізує операцію:  
 $y_i = x_i \cdot \overline{x_4} + \overline{x_i} \cdot x_4$ , у зв'язку з чим при наявності у знаковому розряді  $x_4$  одиниці (від'ємне число) код порозрядно інвертується, а при  $x_4 = 0$  передається без змін. У багатьох серіях такий перетворювач може бути реалізований на мікросхемі ЛП5. У розглянутому випадку відповідність між вхідним та вихідним кодами задається формулою. Але частіше необхідні формули витікають із таблиць відповідності кодів. Деякі з них розглядалися у **Розділі 1**.

Розглянемо декілька прикладів розробки перетворювачів кодів.

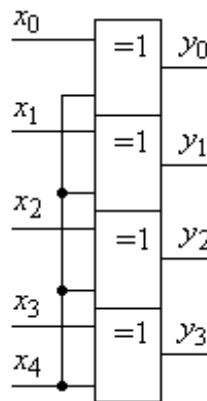


Рис. 3.19

**Приклад 3.10.** Створити перетворювач двійкового коду в прямий код семисегментних індикаторів для відображення цифр десяткової системи числення.

**Розв'язання.** Умовне розміщення світлодіодів матриці семисегментного індикатора наведено на рис. 3.20.

Відповідно до нього, створюємо таблицю істинності перетворювача (табл. 3.5), приписуючи значення логічної “1” тому світлодіодові, що засвічується при відображенні цифри, та логічного “0” – у протилежному випадку. Використовуючи табл. 3.5, можемо записати логічні функції роботи кожного світлодіода у ДДНФ:

$$\begin{aligned}
 a &= \vee 0, 2, 3, 5, 6, 7, 8, 9 = \wedge 1, 4; & b &= \vee 0, 1, 2, 3, 4, 7, 8, 9 = \wedge 5, 6; & c &= \vee 0, 1, 3, 4, 5, 6, 7, 8, 9 = \wedge 2; \\
 d &= \vee 0, 2, 3, 5, 6, 8, 9 = \wedge 1, 4, 7; & e &= \vee 0, 2, 6, 8 = \wedge 1, 3, 4, 5, 7, 9; & f &= \vee 0, 4, 5, 6, 8, 9 = \wedge 1, 2, 3, 7; \\
 g &= \vee 2, 3, 4, 5, 6, 8, 9 = \wedge 0, 1, 7.
 \end{aligned}$$

Даний перетворювач кодів може бути реалізований в базисі ЛЕ **I-АБО-НИ**. Звернемо увагу на те, що проектованим комбінаційним пристроєм не використовуються комбінації вхідних сигналів від **1010** до **1111**, яким відповідають десяткові числа від **10** до **15**, тобто логічні функції, котрі описують роботу світлодіодів семисегментного індикатора, є неповністю визначеними.

Таблиця 3.5

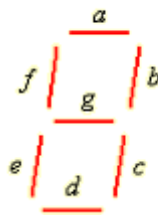


Рис. 3.20

N	$x_3$	$x_2$	$x_1$	$x_0$	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

Скориставшись даними табл. 3.5, представляємо кожен логічний функцію у вигляді карти Карно (рис. 3.21). Довизначивши функції  $a, b, c, d, f, g$  – одиницями, а функцію  $e$  – одиницею та нулями, здійснимо їх мінімізацію.

Після мінімізації знаходимо:

$$\begin{aligned}
 a &= x_1 + x_3 + \overline{x_2} \cdot \overline{x_0} + x_2 \cdot x_0; & b &= \overline{x_2} + x_3 + \overline{x_1} \cdot \overline{x_0} + x_1 \cdot x_0; & c &= x_3 + x_2 + \overline{x_1} + x_0; \\
 d &= x_3 + \overline{x_2} \cdot \overline{x_0} + x_1 \cdot \overline{x_0} + \overline{x_2} \cdot x_1 + x_2 \cdot \overline{x_1} \cdot \overline{x_0}; & e &= \overline{x_2} \cdot \overline{x_0} + \overline{x_3} \cdot x_1 \cdot \overline{x_0}; \\
 f &= x_3 + \overline{x_1} \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + x_2 \cdot \overline{x_0}; & g &= x_3 + x_1 \cdot \overline{x_0} + x_2 \cdot \overline{x_1} + \overline{x_2} \cdot x_1;
 \end{aligned}$$

Технічна реалізація вказаних функцій в базисі **I-АБО-НІ** не викликає складностей.

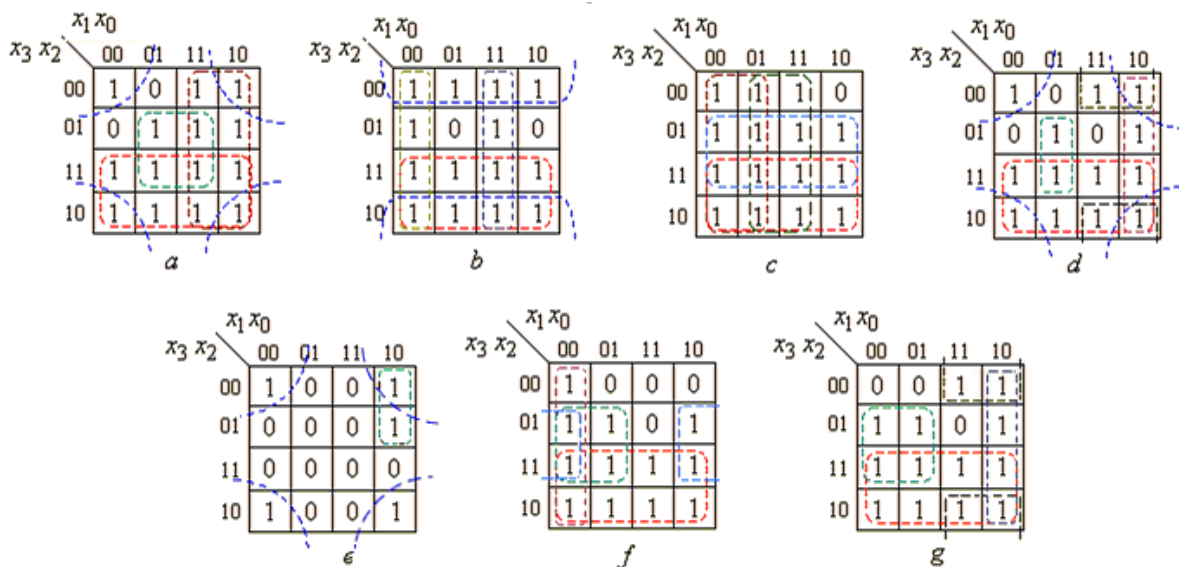


Рис. 3.21

Табл. 3.5 відповідає логіці роботи дешифратора К564ИД5 (CD4056BE, НСС4056BD, ТС4056BF). Умовне графічне позначення дешифратора К564ИД5 приведено на рис. 3.22.

Таким чином, для побудови перетворювачів кодів перш за все необхідно мати таблицю істинності його роботи, яка складається з двох частин. Перша з них – це таблиця перебору

змінних перетворюваного коду, а друга – це таблиця відповідності змінних отриманого коду. З приведенного вище прикладу витікає, що для створення перетворювача кожна з логічних змінних отриманого коду повинна розглядатись як функція, в якій аргументами виступають змінні перетворюваного коду.

У ряді випадків при побудові перетворювачів кодів отримуються такі логічні функції, котрі простіше реалізувати за допомогою арифметичних обчислень.

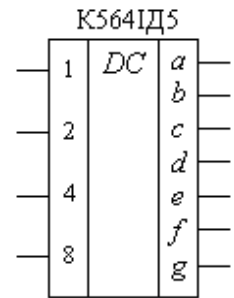


Рис. 3.22

**Приклад 3.11.** Розробити логічну схему перетворювача прямого двійкового коду в доповнюючий.

*Розв'язання.* Доповнюючий код отримується інвертуванням кожного розряду прямого двійкового коду і додаванням до отриманого результату одиниці шляхом двійкової арифметики (таблиця відповідності кодів приводиться у **Розділі 1**).

Карти Карно для вихідних функцій перетворювача прямого двійкового коду в доповнюючий приведені на рис. 3.23.

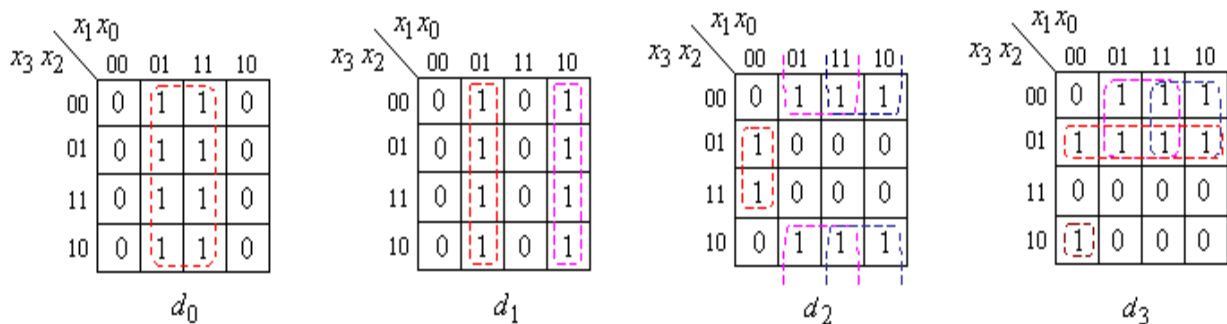


Рис. 3.23

Мінімізовані значення логічних функцій:

$$d_0 = x_0 = 0 \oplus x_0 ;$$

$$d_1 = \overline{x_0} \cdot x_1 + x_0 \cdot \overline{x_1} = x_0 \oplus x_1 ;$$

$$d_2 = x_0 \cdot \overline{x_2} + x_1 \cdot \overline{x_2} + \overline{x_0} \cdot \overline{x_1} \cdot x_2 = (x_0 + x_1) \oplus x_2 ;$$

$$d_3 = x_2 \cdot \overline{x_3} + x_0 \cdot \overline{x_3} + x_1 \cdot \overline{x_3} + \overline{x_0} \cdot \overline{x_1} \cdot \overline{x_2} \cdot x_3 = (x_0 + x_1 + x_2) \oplus x_3 .$$

З аналізу приведених формул можна знайти загальний вигляд функції для  $i$ -го розряду вихідної функції:

$$d_i = (x_0 + x_1 + \dots + x_{i-1}) \oplus x_i .$$

Як витікає з одержаних формул, з підвищенням розрядності формула для знаходження вихідної функції  $d_i$  значно ускладнюється, і її використання фактично стає недоцільним.

Тому для знаходження розрядів доповнюючого коду більш економним для багаторозрядних чисел виявляється перетворення їх у зворотний код з наступним додаванням одиниці до молодшого розряду за допомогою суматора.

Аналогічно вирішуються задачі і при проектуванні двійково-десяткових перетворювачів кодів (Див. **Розділ 1**). Наприклад, для перетворення двійково-десятькового коду в код “з надлишком 3” економніше використовувати суматор, що додає число **11** (3 – в двійковому коді) до молодших розрядів числа.

У залежності від характеру перетворення інформації, ці операції називають *шифруванням* або *дешифруванням* (широко використовується також термінологія *кодування* та *декодування*). Шифрування полягає у перетворенні  $m$ -розрядного коду з  $k_m$  байдужих наборів вхідних змінних в однозначно відповідний йому  $n$ -розрядний код з меншим числом розрядів ( $n < m$ ) і байдужих наборів. Під байдужими наборами в будь-якому коді маються на увазі ті набори, які не використовуються при перетворенні. Пристрої, які виконують вказані операції, називаються відповідно *шифраторами* та *дешифраторами* (*кодерами* та *декодерами*).

**Приклад 3.12.** Розробити схему шифратора для перетворення десятичного коду з клавіатури калькулятора в відповідний йому двійково-десятьковий код.

*Розв’язання.* Прийmemo, що при натисканні відповідної клавіші (0...9) низький рівень відповідного сигналу перетвориться в чотирьохрозрядний двійковий код від **0000** до **1001**. Таблиця шифратора “з 10 в 4” (і, відповідно, дешифратора “з 4 в 10”) приводиться нижче (табл. 3.6).

З табл. 3.6 витікає, що для кожної клавіші вхідного коду характерною є лише одна з можливих комбінацій **1** та **0**. До решти комбінацій проєктований кодер повинен бути байдужим.

Аналогічно, у вихідних комбінаціях значень  $y_3 y_2 y_1 y_0$  не використовуються набори від **1010** до **1111**.

Реалізація такого перетворювача може бути виконана з використанням формул:

$$\begin{aligned} y_0 &= \overline{x_1} + \overline{x_3} + \overline{x_5} + \overline{x_7} + \overline{x_9}; & y_1 &= \overline{x_2} + \overline{x_3} + \overline{x_6} + \overline{x_7}; \\ y_2 &= \overline{x_4} + \overline{x_5} + \overline{x_6} + \overline{x_7}; & y_3 &= \overline{x_8} + \overline{x_9}. \end{aligned}$$



Табл. 3.6 близька до таблиці станів шифратора K555ИВ3 з тією лише різницею, що виходи останнього  $y_0 \dots y_3$  – інвертовані.

Таблиця 3.6

$N$	$x_0$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$	$x_7$	$x_8$	$x_9$	$y_3$	$y_2$	$y_1$	$y_0$
0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	0	0	0	1
2	1	1	0	1	1	1	1	1	1	1	0	0	1	0
3	1	1	1	0	1	1	1	1	1	1	0	0	1	1
4	1	1	1	1	0	1	1	1	1	1	0	1	0	0
5	1	1	1	1	1	0	1	1	1	1	0	1	0	1
6	1	1	1	1	1	1	0	1	1	1	0	1	1	0
7	1	1	1	1	1	1	1	0	1	1	0	1	1	1
8	1	1	1	1	1	1	1	1	0	1	1	0	0	0
9	1	1	1	1	1	1	1	1	1	0	1	0	0	1

До складу кодуєчих пристроїв часто входить пристрій для виділення старшої одиниці. Робота цього пристрою полягає в тому, що одиниця на виході  $i$ -го розряду при наявності одиниці на його вході можлива лише тоді, коли на входах всіх старших розрядів мають місце нулі. Тобто для чотирьохрозрядного перетворювача слова  $x_3 x_2 x_1 x_0$  в  $y_3 y_2 y_1 y_0$  повинні бути справедливими наступні співвідношення:

$$y_0 = x_0 \cdot \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}; \quad y_1 = x_1 \cdot \overline{x_2} \cdot \overline{x_3}; \quad y_2 = x_2 \cdot \overline{x_3}; \quad y_3 = x_3;$$

Для організації такого перетворювача як модуля з можливістю нарощування розрядності його доповнюють вхідним дозволяючим сигналом  $e$  і вихідним дозволяючим сигналом  $q$ .

Технічна реалізація модуля на базі ТТЛ-елементів **I-НІ** приведена на рис. 3.24.

Логіка роботи схеми описується формулами:

$$y_0 = x_0 \cdot e \cdot \overline{y_1} \cdot \overline{y_2} \cdot \overline{y_3} = e \cdot x_0 \cdot \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \quad y_1 = x_1 \cdot e \cdot \overline{y_2} \cdot \overline{y_3} = e \cdot x_1 \cdot \overline{x_2} \cdot \overline{x_3}$$

$$y_2 = x_2 \cdot e \cdot \overline{y_3} = e \cdot x_2 \cdot \overline{x_3} \quad y_3 = x_3 \cdot e \quad q = e \cdot \overline{x_0} \cdot \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$

Якщо використовуються слова, розрядність яких перевищує розрядність модуля, то вони розбиваються на групи, які поєднуються завдяки виходу  $q$

старшої групи і входу  $e$  молодшої. Підключення до виходу приведеної схеми шифратора, що відповідає, наприклад, табл. 3.6, дає можливість створити схему, яка перетворює біт старшої одиниці у відповідний двійковий код. Іншими словами, така схема визначатиме у двійковому коді номер біту найстаршої одиниці.

У мікросхемах ТТЛ подібним пріоритетним шифратором є мікросхема К1533ИВ1 (зарубіжні аналоги – 74148РС, SN74148N). Прикладом використання цієї мікросхеми є схема, приведена на рис. 3.25, яка кодує значення десяткової клавіатури.

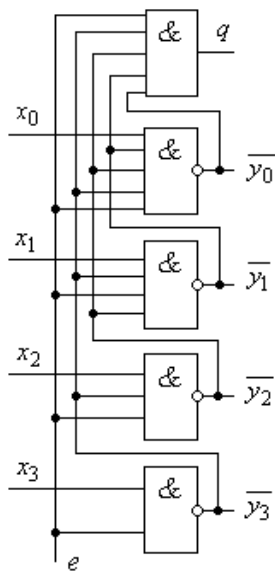


Рис. 3.24

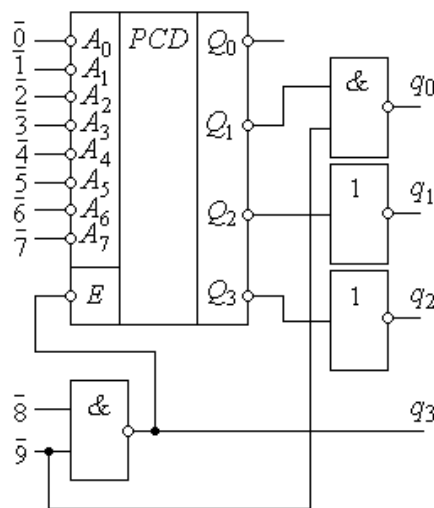


Рис. 3.25

Значення  $\bar{0} \dots \bar{9}$  підключаються до входів клавіатури, а на виходах  $q_0 \dots q_3$  у двійковому коді одержується код натиснутої клавіші. Пропонуємо самостійно розглянути роботу пристрою, схему якого зображено на рис. 3.25.

Пріоритетні шифратори знаходять використання для визначення номеру пристрою, що подає сигнал запиту на обслуговування в мікропроцесорних системах, визначення станів та номерів вимикачів у схемах автоматики та інше.

Дешифратор “з 4 в 10” також описується таблицею, аналогічною табл. 3.6.

Наприклад, мікросхема К561ИД1 (зарубіжні функціональні аналоги – 4028BDC, BU4028B, CD4028AD) є перетворювачем прямого

чотириохрозрядного двійково-десятькового коду в прямий десятковий, який часто називають *унітарним кодом*, оскільки сигнал логічної “1” (“0”) може бути лише на одному з виходів (див. рис. 3.26).

Дешифратори широко використовуються для організації звернення до тих чи інших пристроїв – наприклад, до конкретних пристроїв пам’яті.

### 3.3.1. Дешифратори-демультиплексори

*Демультиплексори* – це цифрові комбінаційні пристрої, функціональне призначення яких протилежне функціональному призначенню мультиплексорів. У них сигнали з одного інформаційного входу  $x$  розподіляються на  $2^m$  виходів  $y_i$ , які комутуються  $m$  адресними входами, тобто фактично вони є генераторами мінтермів. Завдяки цій властивості легко створювати канонічні суми, тобто реалізовувати за допомогою логічних операцій **АБО** логічні функції вхідних змінних.

Якщо прийняти кількість адресних входів  $m = 2$  (рис. 3.26), то роботу пристрою описуватимуть булеві рівняння (3.11). Прикладом табличної форми запису роботи демультиплексора, згідно з рівняннями (3.11), є табл. 3.7.

У відповідності до схеми рис. 3.26, активними станами виходів  $y_0 \div y_3$  є логічні “1”.

У мікросхемах ТТЛ виходи часто мають інверсні значення, і активні стани характеризуються низьким рівнем сигналу.

Таблиця 3.7

$V$	$A_1$	$A_0$	$x$	$y_0$	$y_1$	$y_2$	$y_3$
0	0	0	1/0	1/0	0	0	0
0	0	1	1/0	0	1/0	0	0
0	1	0	1/0	0	0	1/0	0
0	1	1	1/0	0	0	0	1/0
1	0	0	$x$	$\bar{v}$	0	0	0
1	0	1	$x$	0	$\bar{v}$	0	0
1	1	0	$x$	0	0	$\bar{v}$	0
1	1	1	$x$	0	0	0	$\bar{v}$

$$\begin{aligned}
 y_0 &= \bar{v} \cdot x \cdot \bar{a}_0 \cdot \bar{a}_1 \\
 y_1 &= \bar{v} \cdot x \cdot a_0 \cdot \bar{a}_1 \\
 y_2 &= \bar{v} \cdot x \cdot \bar{a}_0 \cdot a_1 \\
 y_3 &= \bar{v} \cdot x \cdot a_0 \cdot a_1
 \end{aligned}
 \tag{3.11}$$

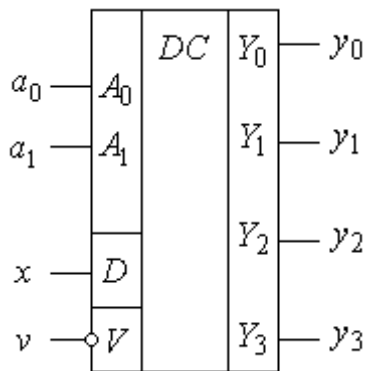


Рис. 3.26

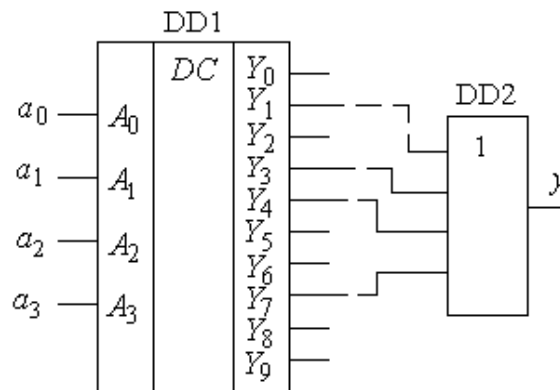


Рис. 3.27

Особливістю більшості серійних мікросхем є їх багатфункціональність. Приведена на рис. 3.27 мікросхема DD1 дешифратора при  $a_3 = 0$  виконуватиме функцію перетворювача трьохрозрядного двійкового коду у вісімковий. Значення  $a_3 = 1$  при цьому буде заборонено. Якщо на вхід  $a_3$  подавати послідовний двійковий код, а входи  $a_0, a_1, a_2$  використовувати як адресні, то одержимо демультимплексор “з 1 на 8”.

**Приклад 3.13.** Розробити перетворювач чотирьохрозрядного двійкового коду в код “2 з 5” та коду “2 з 5” в двійковий код.

*Розв’язання.* Код “2 з 5” використовується для безпомилкової передачі цифрової інформації. В цьому п’ятирозрядному коді кожна допустима комбінація містить дві одиниці та три нулі і, оскільки таких комбінацій десять, то кожна з них відповідає одній десятковій цифрі (табл. 3.8).

Мінімізуючи функції  $a_0 \div a_4$ , отримаємо вирази для вихідних сигналів, які легко реалізуються у базисі комбінаційної логіки або з допомогою дешифратор.

$$a_0 = x_2 \bar{x}_1 \bar{x}_0 + x_1 x_0 + \bar{x}_3 \bar{x}_2 \bar{x}_1 x_0; \quad a_1 = \bar{x}_2 (x_0 \oplus x_3) + \bar{x}_3 \bar{x}_1 x_0; \quad a_2 = x_3 x_0 + x_1 (\bar{x}_2 + \bar{x}_0);$$

$$a_3 = x_2 (\bar{x}_1 + \bar{x}_0); \quad a_4 = x_3 + x_2 x_1 x_0;$$

**Приклад 3.14.** Використовуючи дешифратор K1533ИДЗ (SN74154N), призначений для перетворення чотирьохрозрядного двійкового коду в напругу низького логічного рівня, що з’являється на одному з шістнадцяти виходів  $\bar{0} \div \bar{15}$ , реалізувати логічну функцію:

$$y = x_1 \bar{x}_2 \bar{x}_3 + x_1 x_3 + \bar{x}_1 x_2.$$

Таблиця 3.8

Десяткова цифра	4-розрядний двійковий код				Код "2 з 5"				
	$x_3$	$x_2$	$x_1$	$x_0$	$a_4$	$a_3$	$a_2$	$a_1$	$a_0$
0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1	1
2	0	0	1	0	0	0	1	0	1
3	0	0	1	1	0	0	1	1	0
4	0	1	0	0	0	1	0	0	1
5	0	1	0	1	0	1	0	1	0
6	0	1	1	0	0	1	1	0	0
7	0	1	1	1	1	0	0	0	1
8	1	0	0	0	1	0	0	1	0
9	1	0	0	1	1	0	1	0	0

Для реалізації функції використовувати багатовходові логічні елементи з множини **I**, **I-НІ**, **АБО**, **АБО-НІ**.

*Розв'язання.* Для реалізації функції приведемо її до досконалої диз'юнктивної нормальної форми:  $y = x_1 \bar{x}_2 \bar{x}_3 + x_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + \bar{x}_1 x_2 x_3 + \bar{x}_1 x_2 \bar{x}_3$ .

Понизивши порядок індексів при логічних змінних, отримуємо:  $y = \vee 1, 2, 5, 6, 7$ .

Використовуючи шестивходовий елемент К1533ЛА2 **I-НІ** (зарубіжні аналоги – 7430РС, СDB430Е, МН7430, SN7430N, УСУ7430N), отримаємо схему, яка реалізовуватиме задану функцію (рис. 3.28).

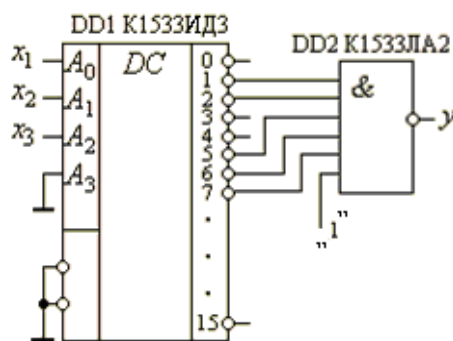


Рис. 3.28

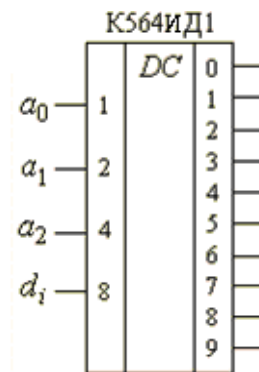


Рис.3.29

**Приклад 3.15.** Використовуючи дешифратор К564ИД1, розробити пристрій для перетворення двійкового однобайтного коду, заданого в послідовному форматі, в код, представлений у паралельному форматі.

*Розв'язання.* Дешифратор К564ИД1 – це перетворювач чотирьохрозрядного двійкового коду в прямий десятковий. Він має чотири входи розрядів двійкового коду (1, 2, 4, 8) і 10 виходів розрядів десяткового коду (0 ÷ 9) (рис. 3.29). табл. 3.9 є таблицею істинності мікросхеми і пояснює її роботу.

Для використання дешифратора К564ИД1 у якості демультимплексора входи 1, 2, 4 необхідно використовувати як адресні  $a_0, a_1, a_2$ , а вхід 8 як вхід даних послідовного формату. На вихідних шинах  $0 \div 7$  отримуватимемо рознесені в часі значення двійкового коду, який подається на вхід 8 в інвертованому виді.

На рис. 3.30 зображені часові діаграми роботи пристрою при подачі на вхід 8 послідовного двійкового коду  $d_i$ .

Таблиця 3.9

Вхідний код Входи				Вихідний код Виходи											
8	4	2	1	0	1	2	3	4	5	6	7	8	9		
0	0	0	0	1	0	0	0	0	0	0	0	0	0		
0	0	0	1	0	1	0	0	0	0	0	0	0	0		
0	0	1	0	0	0	1	0	0	0	0	0	0	0		
0	0	1	1	0	0	0	1	0	0	0	0	0	0		
0	1	0	0	0	0	0	0	1	0	0	0	0	0		
0	1	0	1	0	0	0	0	0	1	0	0	0	0		
0	1	1	0	0	0	0	0	0	0	1	0	0	0		
0	1	1	1	0	0	0	0	0	0	0	1	0	0		
1	0	0	0	0	0	0	0	0	0	0	0	1	0		
1	0	0	1	0	0	0	0	0	0	0	0	0	1		
1	0	1	0	0	0	0	0	0	0	0	0	0	0		
1	0	1	1	0	0	0	0	0	0	0	0	0	0		
1	1	0	0	0	0	0	0	0	0	0	0	0	0		
1	1	0	1	0	0	0	0	0	0	0	0	0	0		
1	1	1	0	0	0	0	0	0	0	0	0	0	0		
1	1	1	1	0	0	0	0	0	0	0	0	0	0		

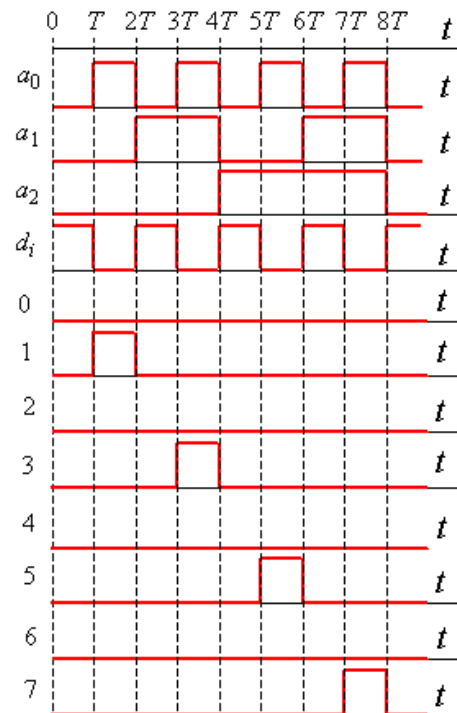


Рис. 3.30

Дешифратори широко використовуються для реалізації логічних функцій. Наприклад, для реалізації функції:  $y = \vee 1, 3, 4, 7$  достатньо об'єднати виходи  $y_1, y_3, y_4, y_7$  за допомогою чотирьохвходового елемента АБО (DD2 на рис. 3.27).

Якщо виходи  $y_0 \div y_{n-1}$  дешифратора інвертовані, то об'єднання виходів та відповідних мінтермів забезпечується елементами, що реалізують операцію І-НІ.

Використання дешифраторів як генераторів мінтермів корисно в тих випадках, коли необхідно реалізовувати систему логічних функцій. При цьому відпадає необхідність їх мінімізації. Більш того, їх необхідно приводити до досконалої диз'юнктивної форми.

**Приклад 3.16.** Комбінаційна схема визначена за допомогою наступних рівнянь:

$$y_1 = x_1 x_2 + \bar{x}_1 \bar{x}_2 \bar{x}_3 ;$$

$$y_2 = x_1 + x_2 + \bar{x}_3 ;$$

$$y_3 = \bar{x}_1 x_2 + x_1 \bar{x}_2 x_3 ,$$

Розробити схему, що реалізує ці три рівняння за допомогою дешифратора та зовнішніх логічних елементів.

*Розв'язання.* Для вирішення цієї задачі перетворимо функції до вигляду:

$$\bar{y}_1 = \overline{x_1 x_2 x_3 + x_1 x_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 \bar{x}_3} = \overline{x_1 x_2 x_3} \cdot \overline{x_1 x_2 \bar{x}_3} \cdot \overline{\bar{x}_1 \bar{x}_2 \bar{x}_3} ;$$

$$\bar{y}_2 = \overline{x_1 + x_2 + \bar{x}_3} = \bar{x}_1 \bar{x}_2 x_3 ;$$

$$\bar{y}_3 = \overline{\bar{x}_1 x_2 x_3} \cdot \overline{\bar{x}_1 x_2 \bar{x}_3} \cdot \overline{x_1 \bar{x}_2 x_3} .$$

Із приведених функцій бачимо:

1) функція  $y_2$  може бути отримана за допомогою звичайного інвертора, підключеного до виходу, відповідного  $x_3 \bar{x}_2 \bar{x}_1 = 100_2 = 4_{10}$  ;

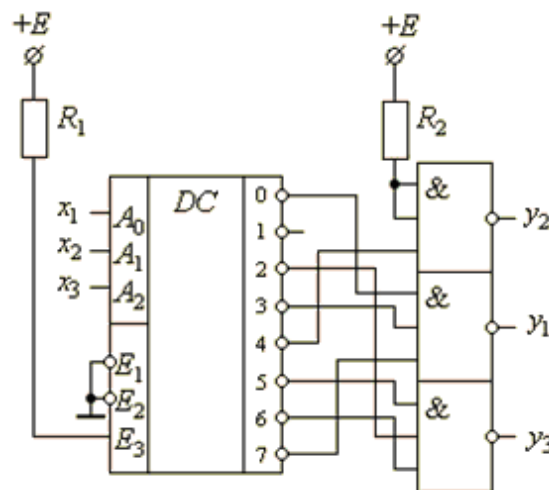


Рис. 3.31

2) функція  $y_1$  отримується об'єднанням через елемент **3І-НІ** виходів, відповідних  $x_3 x_2 x_1 = 111_2 = 7_{10}$ ;  $\bar{x}_3 x_2 x_1 = 011_2 = 3_{10}$ ;  $\bar{x}_3 \bar{x}_2 \bar{x}_1 = 000_2 = 0_{10}$  ;

3) аналогічно, функція  $y_3$  отримується шляхом об'єднання виходів, номери яких відповідають  $x_3 x_2 \bar{x}_1 = 110_2 = 6_{10}$ ;  $\bar{x}_3 x_2 \bar{x}_1 = 010_2 = 2_{10}$ ;  $x_3 \bar{x}_2 x_1 = 101_2 = 5_{10}$ , за допомогою елемента **3І-НІ**.

Використаємо мікросхему K1533JIE4 (або її зарубіжні аналоги – 7427PC, DM7427N, ECG7427, FLN621, MC7427N, SN7427N, SN66N). Принципова схема, яка реалізує вказані функції, зображена на рис. 3.31.

Нарощування дешифраторів забезпечується на тих самих принципах, що і нарощування мультиплексорів.

Наприклад, на рис. 3.32 наведена схема перетворювача “з 4 в 16”. Наведена схема має можливість виконувати різні види кодових перетворень.

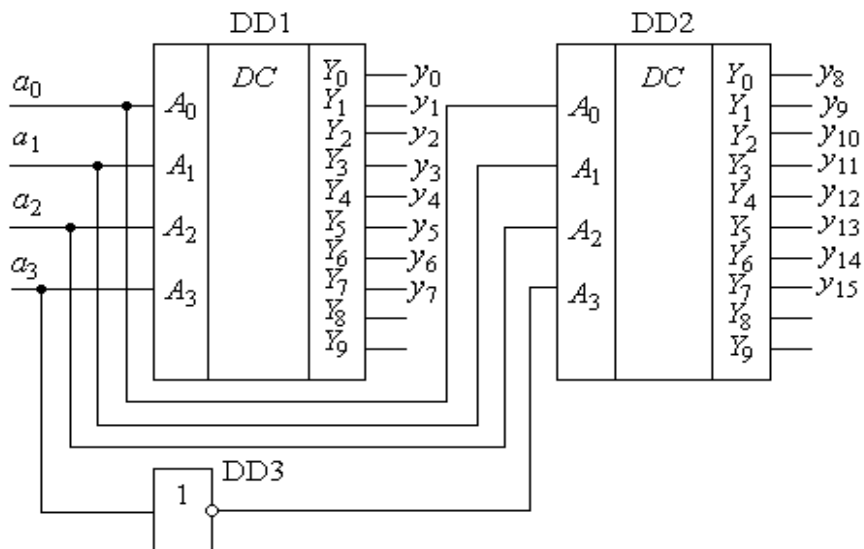


Рис. 3.32

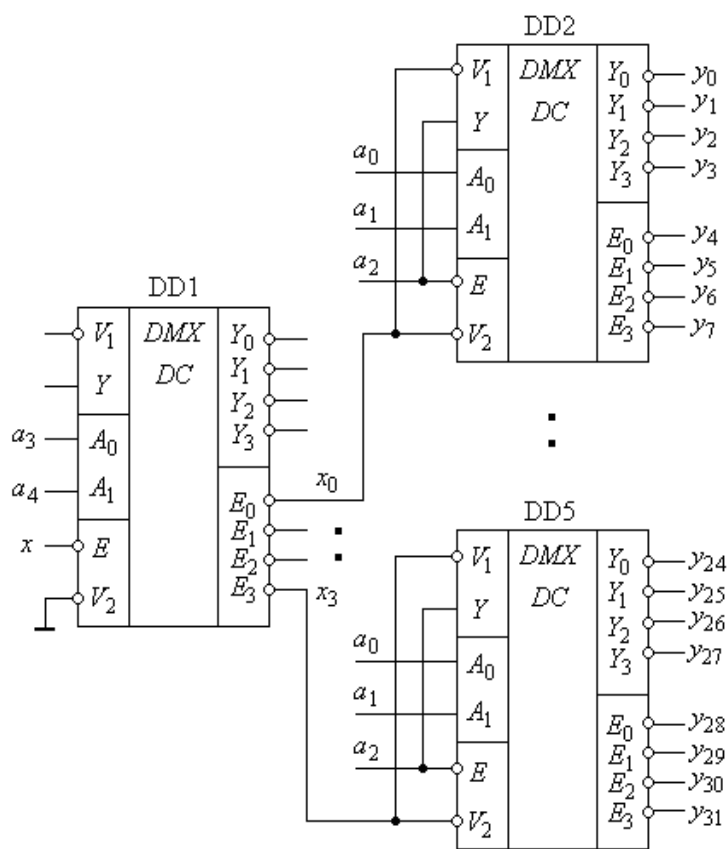


Рис. 3.33

У залежності від необхідного вихідного коду, необхідно використовувати ті виходи  $y_0 \div y_{15}$  схеми, які відповідають вибраному коду.

При пірамідальному нарощуванні перетворювачів кодів організація схеми залежить від характеру їх використання.

На рис. 3.33 приводиться приклад створення демультіплексора на 32 виходи на базі мікросхем К1533ИД4 (зарубіжні аналоги – 74155РС,





**Приклад 3.18.** Використовуючи дешифратори серії 564ИД1, розробити принципову схему дешифрації шестирозрядного двійкового коду на 64 виходи.

*Розв'язання.* Дешифратор 564ИД1 виконує перетворення чотирьохрозрядного двійкового коду в вихідний сигнал високого рівня на одному з 10 виходів.

Із таблиці істинності мікросхеми (табл. 3.9) маємо, що при зміні двійкового коду в інтервалі значень, відповідаючих десятковим числам 0 – 9, має місце однозначна відповідність вхідного двійкового коду і сигналу високого рівня на відповідному виході 0 – 9. При значеннях вхідного двійкового коду, більших **111**, виходи 0 – 7 мають нульовий вихідний сигнал. Таким чином, мікросхема дозволяє однозначно виконувати дешифрацію трьохрозрядного двійкового коду на один з 8 виходів при наявності нуля на вході  $A_3$  (рис. 3.35).

При наявності сигналу високого рівня на вході  $A_3$  виходи 0 – 7 будуть відключені. Тобто вхід  $A_3$  може виконувати функцію входу, який дозволяє роботу мікросхеми з декодування інформації, яка надходить на входи  $A_0, A_1, A_2$ .

Виконаний аналіз вказує на те, що мікросхему можна використовувати для побудови дешифратора шестирозрядного двійкового коду на 64 виходи лише шляхом їх пірамідальної побудови. Для цього 8 дешифраторів запаралелюються по входах  $A_0 - A_2$ , а вхід  $A_3$  підключається до виходів керуючого дешифратора через інвертори. Входи  $A_0 - A_2$  керуючого дешифратора є старшими розрядами адресної шини  $x_0 - x_5$ .

Окрім розглянутих областей використання, кодові перетворювачі виступають складовою частиною керуючих автоматів. Широке використання вони знаходять у цифрових пристроях криптографічного захисту інформації.

На рис. 3.36 приведені два варіанти засекречення інформації.

Спосіб перестановки з допомогою внутрішніх комутаторів (рис. 3.36, а) надає можливість для  $n$ -входового перетворювача створити  $n!$  можливих варіантів з'єднань.

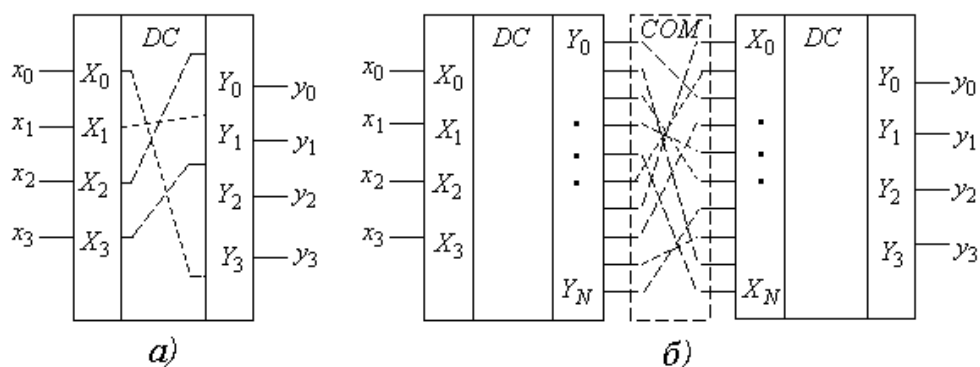


Рис. 3.36

Спосіб підстановки (рис. 3.36, б) розширює кількість варіантів, адже перестановка виконується після перетворення вхідного двійкового коду. Кількість можливих варіантів з'єднань з допомогою комутатора **СOM** підвищується до  $2^n!$ . Зворотнє перетворення інформації забезпечується блоком зі зворотнім законом перестановки. Використання подібних пристроїв дає можливість відкрито передавати та зберігати конфіденційну та секретну інформацію.

### 3.4. Обчислювальні пристрої

#### 3.4.1. Суматори

Найпростішою схемою цього класу є *напівсуматор*, який виконує операцію знаходження суми двох однорозрядних чисел.

Роботу напівсуматора описує табл. 3.10, з якої знаходимо логічні рівняння для суми і для переносу:  $S = a_0 \cdot \bar{b}_0 + \bar{a}_0 \cdot b_0$ ,  $P = a_0 \cdot b_0$ .

При знаходженні суми багаторозрядних слів напівсуматор може використовуватись лише в молодшому розряді. Для знаходження суми старших розрядів схема повинна мати три входи – два інформаційні і третій – вхід переносу з молодших розрядів. Схема, що відповідає цим вимогам, називається *повним суматором*, а логіку її роботи описує табл. 3.11.

Таблиця 3.11

Таблиця 3.10

$a_0$	$b_0$	$S$	$P$
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$N$	$a_i$	$b_i$	$p_{i-1}$	$S_i$	$P_{i+1}$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

З табл. 3.11 знаходимо наступні рівняння:

$$S_i = \forall 1, 2, 4, 7 = p_{i-1} (\bar{a}_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1} (a_i \cdot \bar{b}_i + \bar{a}_i \cdot b_i);$$

$$P_{i+1} = \forall 3, 5, 6, 7 = p_{i-1} (\bar{a}_i \cdot b_i + a_i \cdot \bar{b}_i + a_i \cdot b_i) + \bar{p}_{i-1} a_i \cdot b_i = a_i \cdot b_i + p_{i-1} (a_i \oplus b_i)$$

Знаходження суми багаторозрядних слів може виконуватись як послідовно, так і паралельно. Для послідовного виконання операції використовується один повний суматор, загальний для всіх розрядів. Для виконання операції суми слова подаються на інформаційні розряди послідовно, починаючи з молодших розрядів (рис. 3.37).

Для врахування результату операції з попереднім розрядом вхід та вихід переносу об'єднується через елемент затримки DD2. Цей елемент повинен забезпечити затримку та зберігання сигналу  $P_{i+1}$  на такт виконання операції. Синхронно з інтервалом в один такт повинні подаватись і вхідні дані  $a_i, b_i$ .

Приведена схема виконання операції суми сама по собі нескладна, але вимагає складного апаратного забезпечення. До того ж, послідовне порозрядне виконання операції суттєво знижує швидкодію.

На рис. 3.38 приведена схема організації чотирьохрозрядного паралельного суматора з послідовним переносом.

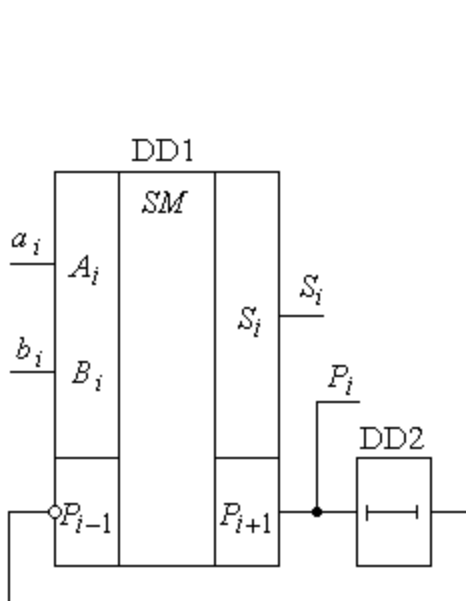


Рис. 3.37

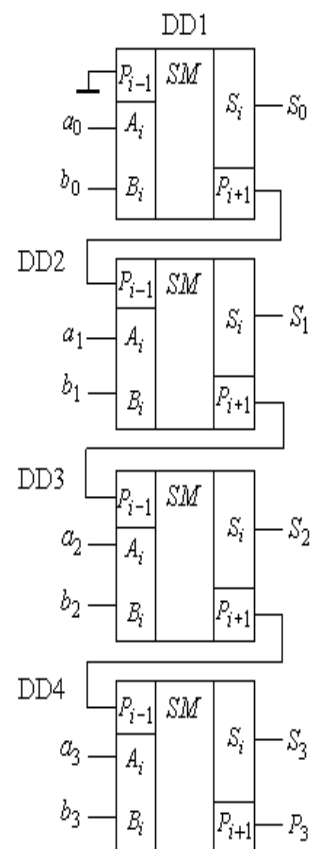


Рис. 3.38

Незважаючи на паралельну одночасну подачу всіх розрядів слів, операція в  $i$ -му розряді починає виконуватись тільки після того, як поступить сигнал переносу з  $(i-1)$ -го розряду. Звідси витікає, що швидкодія суматора визначатиметься сумою затримок передачі сигналу переносу з молодшого розряду на вихід суматора старшого розряду

За схемою, приведеною на рис. 3.38, виготовляються мікросхеми К1533ИМ3 (зарубіжні аналоги – 7483РС, СDB483Е, FLН241, МС7483N, N7483N, SN7482AN, УСУ7483N), умовне зображення якої приведено на рис. 3.39 (мікросхема DD2). Мікросхема може використовуватись для операцій як з додатними, так і з від'ємними числами.

При виконанні операції додавання на вхід  $p_{i-1}$  необхідно подати сигнал низького рівня. Нарощування розрядності суматорів виконується на схемі, приведеній на рис. 3.38. Таблиця станів для суматорів за своєю сутністю є таблицею суми двох чисел (Див. табл. 3.10 – табл. 3.11). Вона має простий вигляд і зручна в користуванні для чисел невеликої розрядності. Суматор К1533ИМ3, як і всі комбінаційні схеми, має широкі функціональні можливості. Перш за все, він може використовуватись для знаходження суми чисел з меншою розрядністю. Схема входів-виходів при цьому може бути різноманітною. Його можливо використовувати і для віднімання двох чисел. Ця операція виконується шляхом додавання зменшуваного і зменшуючого, представленого у доповнюючому коді. Для виконання цієї операції зменшуюче  $b_3 b_2 b_1 b_0$  з допомогою інверторів перетворюється в зворотній код, а потім до нього по входу  $p_0$  додається одиниця.

**Приклад 3.19.** Розглянути мікросхему К1533ИМ3. Пояснити, як використовувати цей чотирихрозрядний суматор для додавання:

- а) двох чотирихрозрядних чисел;
- б) двох дворозрядних та двох одnorозрядних чисел;
- в) двох восьмирозрядних (байтових) чисел.

*Розв'язання.* При додаванні двох чотирихрозрядних слів необхідно вхід переносу  $P_0$  заземлити (рис. 3.40). В дворозрядному суматорі входами є  $A_1, A_2$  та  $B_1, B_2$ , а виходами  $S_1, S_2, S_3$ .

Останній є виходом переносу. В одnorозрядному суматорі використовуються входи  $A_4, B_4$ , а виходами є  $S_4$  та  $P_4$ . При додаванні двох однобайтових слів необхідно наростити два однотипних суматори. При цьому вихід  $P_4$  суматора напівбайтів молодших розрядів з'єднується з входом  $P_0$  суматора напівбайтів старших розрядів слів, що додаються.

Операції знаходження суми та різниці можна поєднати в одному пристрої, як показано на рис. 3.39.

Для виконання операції додавання на вхід  $P_{i-1}$  переносу подається сигнал низького рівня. Відповідно на входах елементів **ВИКЛ. АБО** число  $b_3 b_2 b_1 b_0$  буде представлено в прямому коді. При подачі на вхід  $P_{i-1}$  сигналу високого рівня, як відмічалось раніше, число  $b_3 b_2 b_1 b_0$  буде представлено у доповнюючому коді.

Слід нагадати, що при виконанні операції віднімання результат на виході суматора залежить від співвідношення між числами, що беруть участь в операції.

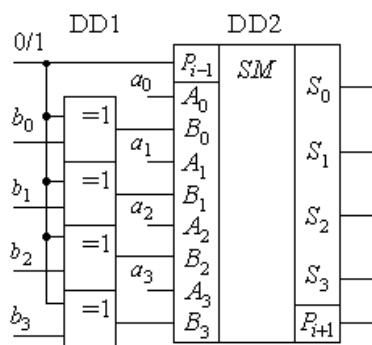


Рис. 3.39

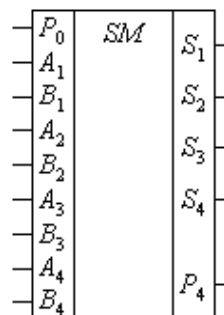


Рис. 3.40

При  $a_3 a_2 a_1 a_0 > b_3 b_2 b_1 b_0$  результат одержується в прямому коді, а в розряді переносу  $P_{i+1}$  з'являється одиниця.

Коли ж  $a_3 a_2 a_1 a_0 < b_3 b_2 b_1 b_0$ , результат отримується в оберненому коді, а в розряді переносу  $P_{i+1}$  одержується 0.

**Приклад 3.20.** Розробити пристрій двійкового множення на суматорах та логічних елементах **I**, що дозволяє множити чотирьохрозрядне число  $A$  на трьохрозрядне число  $B$ .

*Розв'язання.* Після множення трьохрозрядного і чотирьохрозрядного кодів отримується семирозрядний код. Результат операції утворюється за рахунок паралельного множення

множеного на кожен розряд множника і складання проміжних добутоків зі зсувом на один розряд. При підрахунку добутку виконуються наступні дії

			×	A <sub>4</sub>	A <sub>3</sub>	A <sub>2</sub>	A <sub>1</sub>	
					B <sub>3</sub>	B <sub>2</sub>	B <sub>1</sub>	
				A <sub>4</sub> B <sub>1</sub>	A <sub>3</sub> B <sub>1</sub>	A <sub>2</sub> B <sub>1</sub>	A <sub>1</sub> B <sub>1</sub>	
		A <sub>4</sub> B <sub>2</sub>		A <sub>3</sub> B <sub>2</sub>	A <sub>2</sub> B <sub>2</sub>	A <sub>1</sub> B <sub>2</sub>		
	A <sub>4</sub> B <sub>3</sub>	A <sub>3</sub> B <sub>3</sub>		A <sub>2</sub> B <sub>3</sub>	A <sub>1</sub> B <sub>3</sub>			
M <sub>7</sub>	M <sub>6</sub>	M <sub>5</sub>	M <sub>4</sub>	M <sub>3</sub>	M <sub>2</sub>	M <sub>1</sub>		

Тут  $M_1 = A_1B_1$ ;  $M_2 = A_1B_2 + A_2B_1$ ;  $M_3 = A_1B_3 + A_2B_2 + A_3B_1$ ;  $M_4 = A_2B_3 + A_3B_2 + A_4B_1$ ;  $M_5 = A_3B_3 + A_4B_2$ ;  $M_6 = A_4B_3$ ;  $M_7$  – біт переносу із попереднього розряду; знак “+” – знак арифметичного додавання.

Однорозрядне множення забезпечується елементами І. Суми можуть бути реалізовані суматорами К1533ИМ3. Для реалізації цієї операції на рис. 3.41 приведена принципова схема.

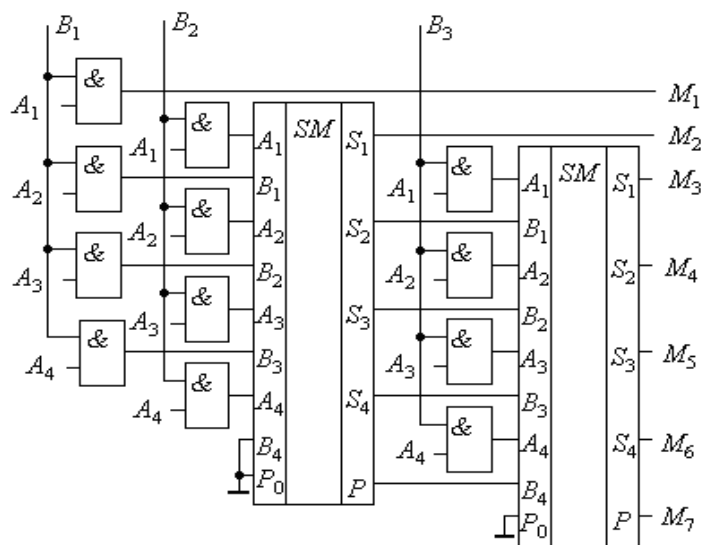


Рис. 3.41

**Приклад 3.21.** Розробити принципову схему суматора, який здійснює додавання однорозрядних десяткових чисел.

*Розв’язання.* Для додавання десяткових чисел необхідно використати їх двійково-десяткову форму представлення. Це дозволяє для виконання операції додавання використати чотирьохрозрядний суматор двох чисел, наприклад, мікросхему К1533ИМ3.

Відомо, що у тому випадку, коли в результаті додавання отримується результат  $S > 9$ , значення молодшого напівбайта є невірним. Для отримання вірного результату необхідно його

відкоригувати. Корегування виконуємо шляхом додавання до результату числа  $6_{10} = 0110_2$ . Цю операцію в системі двійково-десятькового числення можна сформулювати наступним чином. При появі в результаті додавання двійкових кодів  $1010_2 = 10_{10}$ ,  $1011_2 = 11_{10}$ ,  $1100_2 = 12_{10}$ ,  $1101_2 = 13_{10}$ ,  $1110_2 = 14_{10}$ ,  $1111_2 = 15_{10}$  до отриманого результату необхідно додати число  $6_{10}$ . Якщо ці коди не з'являються, то додавання числа  $6_{10}$  не робити.

Зрозуміло, що розроблювана схема повинна мати дві мікросхеми К1533ИМЗ. Перша з них виконуватиме операцію додавання, а друга, за необхідності, виконуватиме корегування результату згідно з логічною функцією:  $y = \vee 10, 11, 12, 13, 14, 15$ .

Представляючи її в формі карти Карно й мінімізуючи, знаходимо:

$$y = x_2 x_3 + x_1 x_3 = x_3 \cdot (x_1 + x_2).$$

При додаванні чисел, результат яких перевищує число  $15_{10}$ , в сумі виникає перенос  $P$  в п'ятий – старший – розряд, який також необхідно враховувати при використанні десяткової корекції. Тому функція  $y$  керування десятковою корекцією набуде вигляду:

$$y = P + x_3 \cdot (x_1 + x_2).$$

Отриманий результат дозволяє побудувати схему суматора (рис. 3.42).

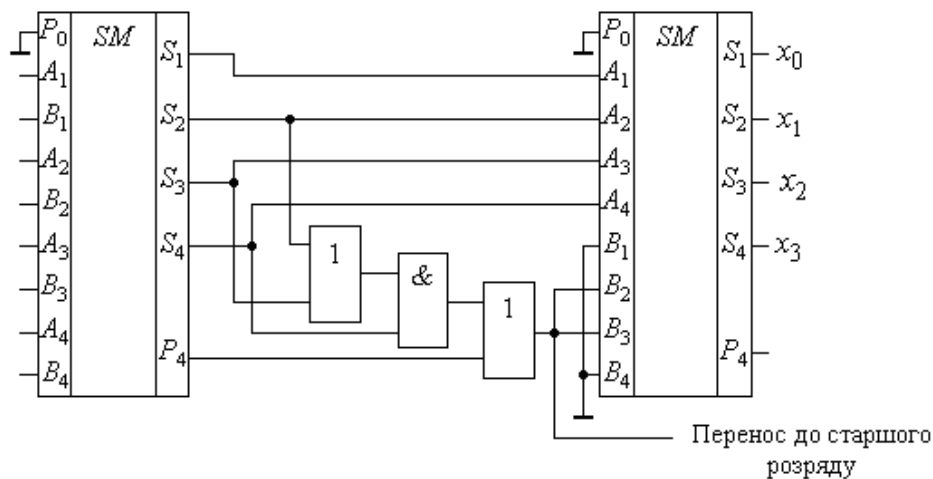


Рис. 3.42

У зв'язку з затримкою переносу, як відмічалось раніше, швидкодія суматора К1533ИМЗ обмежена. Для підвищення швидкодії в ряді серій мікросхем виготовляють спеціальні вузли, які називаються *пристроями прискореного переносу*. Принцип прискореного переносу полягає у тому, що



для кожного двійкового розряду знаходять два допоміжних логічних сигнали:  $g_i = a_i b_i$  (сигнал генерації переносу) та  $h_i = a_i + b_i$  (сигнал розповсюдження).

Сигнал  $g_i = 1$ , коли складові  $i$ -го розряду такі, що перенос в старший розряд  $P_{i+1} = 1$  незалежно від значення вхідного переносу  $P_{i-1}$ .

Сигнал  $h_i$  іноді називають *сигналом прозорості*, тому що відповідно до його визначення при  $h_i = 1$  сигнал переносу з попереднього розряду  $p_i$  вільно пройде наступний розряд  $P_{i+1}$ . За допомогою  $g_i$  та  $h_i$  можна представити роботу вузла переносу одного розряду суматора  $P_{i+1} = g_i + p_i h_i$ .

Будуючи структуру зв'язків між переносами, починаючи зі старшого розряду до нульового, можна створити логічну схему, яка дасть можливість організувати паралельний суматор. Серед мікросхем ТТЛ паралельний перенос має суматор К1533ИМ6, який має затримку розповсюдження сигналу  $t_3$  від входів до виходу  $P_{i+1}$  приблизно у 2 рази меншу, ніж К1533ИМ3. В мікросхемах КМОН суматори також виконуються з вузлом паралельного переносу.

Для прискорення переносу в суматорах з великою кількістю розрядів використовується принцип групового переносу. Суматор розділяється на групи, які мають невелику розрядність і входи-виходи переносу. Сутність групового переносу полягає в тому, що, допоміжно до вузла переносу, в групі будують вузол переносу між групами більш високого рівня, який формує сигнали групових переносів, котрі подаються на входи міні-суматорів. Паралельний перенос між групами суттєво прискорює роботу багаторозрядних суматорів.

### 3.4.2. *Перемножувачі*

Ідеологія перемноження двох бінарних слів була описана в **Розділі 1** і полягала у використанні операцій додавання і зсуву проміжної суми.

Реальні чотирьохрозрядні перемножувачі використовують просту технологію, відповідно до якої для двох слів  $A$  і  $B$ , які необхідно перемножити,

створюється таблиця істинності з вихідним словом  $C$  подвійної довжини. Кожен розряд  $c_i$  слова  $C$  є логічною функцією з логічними змінними слів  $A$  і  $B$ . Тому реалізація чотирьохрозрядного перемножувача є простою реалізацією восьми логічних функцій.

При необхідності реалізувати перемножувач двох восьмирозрядних слів кожне з них розбивається на групи, по 4 біти, і з кожною з груп виконуються операції, як з однією змінною, за принципом знаходження проміжної суми з наступним виконанням операції зсуву.

На базі чотирьохрозрядних перемножувачів і з використанням такої структури будується перемножувач восьмирозрядних слів.

### 3.4.3. Компаратори

*Компаратори* – це пристрої, що виконують порівняння двох чисел.

Мікросхеми компараторів визначають не тільки рівність, але й нерівність двох чисел. Для цього мікросхема має три виходи: « $A > B$ », « $A < B$ » і « $A = B$ », на яких у залежності від співвідношення величин  $A = a_3 a_2 a_1 a_0$ ,  $B = b_3 b_2 b_1 b_0$  з'являється активний рівень сигналу.

Логічна функція, що реалізує алгоритм роботи компаратора для одного розряду ( $a_i = b_i$ ):

$$y_i = a_i \cdot b_i + \overline{a_i} \cdot \overline{b_i} = \overline{a_i \oplus b_i}. \quad (3.12)$$

Для багаторозрядних чисел функція (3.12) повинна виконуватись для кожного розряду. Нерівність  $A > B$  має місце тоді, коли в числі  $A$  одиниці дорівнює більш старший розряд, ніж у числі  $B$ . Для чотирьохрозрядного компаратора така нерівність може бути записана у вигляді:

$$(A > B) = a_3 \overline{b_3} + \overline{(a_3 \oplus b_3)} \cdot a_2 \overline{b_2} + \overline{(a_3 \oplus b_3)} \cdot \overline{(a_2 \oplus b_2)} \cdot a_1 \overline{b_1} + \\ + \overline{(a_3 \oplus b_3)} \cdot \overline{(a_2 \oplus b_2)} \cdot \overline{(a_1 \oplus b_1)} \cdot a_0 \overline{b_0}.$$

Побудова багаторозрядного компаратора можлива на базі суматора. Його реалізація можлива при виконанні операції віднімання.

Дійсно, в відповідності до законів арифметики при  $A = B$  на виходах всіх розрядів суматора  $s_0, s_1, s_2, s_3$  буде  $0$  при  $P_{i+1} = 1$ .

При  $A > B$  значення  $1$  буде як на виході переносу  $P_{i+1} = 1$ , так і хоча б на одному виході  $s_i$ . Тому ознакою  $A > B$  може бути функція  $y_> = p_{i+1} \cdot (\sum s_i)$ .

При  $A < B$  результат наявності переносу  $P_{i+1} = 0$ .

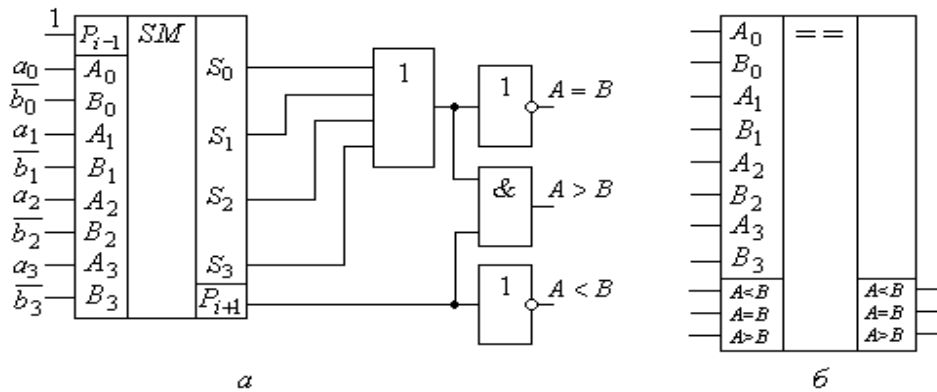


Рис. 3.43

Відповідно до сказаного, логічна схема компаратора на базі суматора К1533ИМ3 (зарубіжні аналоги – 7483РС, CDB483Е, МС7483N, N7483N, SN7483AN, UCY7483N) має вигляд, приведений на рис. 3.43, а.

Прикладом серійних мікросхем є компаратор К564ИП2 (зарубіжні аналоги CD4585BD, CD4585BH, CD4585BK, CD4585BKF3) (КМОП) (рис 3.43, б); К555СП1 (зарубіжні аналоги – 74LS85N, DL083D, SN74LS85N, UCY74LS85N).

Мікросхеми мають розширення по входу « $A > B$ », « $A < B$ » і « $A = B$ », що дає можливість як послідовного нарощування, так і паралельного.

На рис. 3.44 приведена схема послідовного нарощування двох компараторів К561ИП2.

Входи розширення першої мікросхеми необхідно з'єднати відповідно до рис. 3.44, а виходи з'єднуються безпосередньо з входами наступної мікросхеми. При такому способі нарощування розрядності компараторів затримки додаються, а стан виходів другого і послідуєчих компараторів залежить не тільки від побітних значень порівнюваних слів, а й від результату порівняння

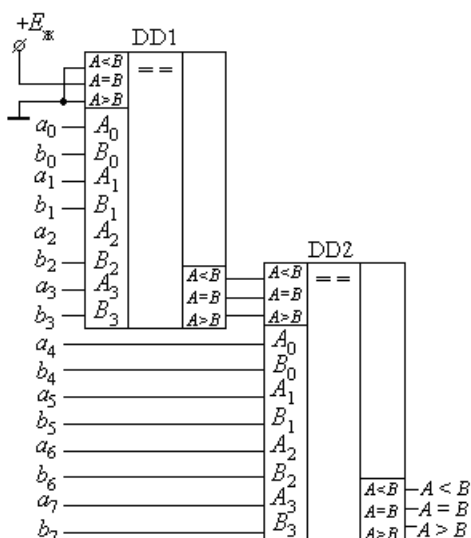


Рис. 3.44

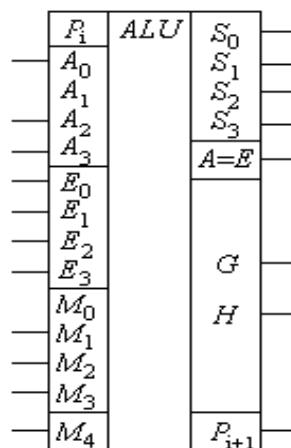


Рис. 3.45

молодших розрядів цих слів. Тому значення виходів кожного з таких компараторів можуть бути описані рівняннями:

$$(A > B)_{\text{вих}} = (A > B)_{n-m} + (A = B)_{n-m} \cdot (A > B)_{\text{вх}} ;$$

$$(A = B)_{\text{вих}} = (A = B)_{n-m} \cdot (A = B)_{\text{вх}} ;$$

$$(A < B)_{\text{вих}} = (A < B)_{n-m} + (A = B)_{n-m} \cdot (A < B)_{\text{вх}} ;$$

де  $n, m$  – діапазон розрядів порівнюваних слів, які приєднані до компаратора.

Компаратори широко використовуються в інформаційних системах для виділення необхідного слова в потоці цифрової інформації, для відмітки часу в часових пристроях, для виконання умовних переходів в обчислювальних пристроях. У пристроях автоматики компаратори використовуються для контролю виходу величин за межі допусків і т.п.

### 3.4.4. Арифметично-логічні пристрої

Арифметично-логічні пристрої (АЛП, ALU – Arithmetic-Logic Unit) – це спеціалізовані мікросхеми, які виконують арифметичні та логічні операції у відповідності до двійкового коду, який подається на керуючі входи мікросхеми. Умовне позначення АЛП в електронних схемах приведене на рис. 3.45.

Мікросхеми АЛП, що належать до різних серій, – наприклад, ТТЛ (К1533ИП3 (зарубіжні аналоги – 74181PC, SN74181N)), КМОН (К564ИП3 (зарубіжні аналоги – 4581BDC, 4581BDM, 4581BFC, GD4581BC, GD4581BDC, HD14581B, MC14581BAL, MC14581BCL, MC14581BCPDS, SCL4581B)), – за своїми функціональними властивостями подібні. Вони призначені для роботи з двома чотирьохрозрядними двійковими словами. Конкретний вид операції задається п'ятирозрядним керуючим кодом, що подається на входи  $M_0$ ,  $M_1$ ,  $M_2$ ,  $M_3$ ,  $M_4$ . Як результат, цей пристрій може виконувати 32 різноманітні команди. Вхід  $M_4$  задає режим АЛП на виконання 16 логічних ( $M_4 = 1$ ) або 16 арифметичних ( $M_4 = 0$ ) операцій. Допоміжні входи  $G$  та  $H$  призначені для використання АЛП в операціях над багаторозрядними словами при одночасному використанні спеціально для цієї мети призначеної мікросхеми К1533ИП4 (зарубіжні аналоги – DM74182N, FLH411, MC74182N, SN74182N, SN84182N, TL84182N, UCY74182N) і К564ИП4 (4582BDC, 4582BDM, 4582BFC, GD4582BC, GD4582BDC, HD14582B, MC14582BAL, MC14582BCL, MC14582BCP, SCL4582B, SCL4582BC). АЛП призначений для роботи з сигналами як позитивної, так і негативної логіки, що відображається у таблиці станів мікросхеми (табл. 3.12).

У табл. 3.12  $L$  та  $H$  позначені відповідно низький і високий рівні сигналів, незалежно від типу логіки. Знаки (+) та (–) – знаки арифметичних операцій. Результати арифметичних дій виражені в доповняльному коді. Так як між оберненим та доповняльним кодами існує однозначний зв'язок, то скрізь, де вказана арифметична дія “–1”, результат на виходах зчитується у оберненому коді.

На відміну від суматорів, сигнал переносу на вході і виході є інверсним по відношенню до сигналів на входах  $A$  та  $E$ . Сигнали переносу діють лише при виконанні арифметичних операцій. При використанні логічних операцій АЛП всі зазначені в табл. 3.12 функції виконує порозрядно.

Таблиця 3.12

Входи вибору функції				Вхід-вихід (негативна логіка)		Вхід-вихід (позитивна логіка)	
$M_3$	$M_2$	$M_1$	$M_0$	Логічні функції $M_4 = H$	Арифметичні дії $M_4 = L;$ $P_1 = L$	Логічні функції $M_4 = H$	Арифметичні дії $M_4 = L$ $P_1 = H$
$L$	$L$	$L$	$L$	$\bar{A}$	$A - I$	$\bar{A}$	$A$
$L$	$L$	$L$	$H$	$\bar{A} \vee \bar{E}$	$AE - I$	$\bar{A}\bar{E}$	$A \vee E$
$L$	$L$	$H$	$L$	$\bar{A} \vee E$	$A\bar{E} - 1$	$\bar{A}E$	$A \vee \bar{E}$
$L$	$L$	$H$	$H$	$I$	$-I$	$0$	$-I$
$L$	$H$	$L$	$L$	$\bar{A}\bar{E}$	$A + (A \vee \bar{E})$	$\bar{A}E$	$A + A\bar{E}$
$L$	$H$	$L$	$H$	$\bar{E}$	$A + (A \vee \bar{E})$	$\bar{E}$	$(A \vee E) + A\bar{E}$
$L$	$H$	$H$	$L$	$\bar{A}(\cdot)\bar{E}$	$A - E - I$	$A(+)\bar{E}$	$A - E - I$
$L$	$H$	$H$	$H$	$A \vee \bar{E}$	$A \vee \bar{E}$	$A\bar{E}$	$A\bar{E} - 1$
$H$	$L$	$L$	$L$	$\bar{A}E$	$A + (A \vee E)$	$\bar{A} \vee E$	$A + AE$
$H$	$L$	$L$	$H$	$A(+)\bar{E}$	$A + E$	$\bar{A}(\cdot)\bar{E}$	$A + E$
$H$	$L$	$H$	$L$	$E$	$A\bar{E} + (A \vee E)$	$E$	$(A \vee \bar{E}) + AE$
$H$	$L$	$H$	$H$	$A \vee \bar{A}E$	$AE$	$AE$	$AE - I$
$H$	$H$	$L$	$L$	$0$	$A + A$	$I$	$A + A$
$H$	$H$	$L$	$H$	$A\bar{E}$	$AE + A$	$A \vee \bar{E}$	$(A \vee E) + A$
$H$	$H$	$H$	$L$	$AE$	$A\bar{E} + A$	$A \vee \bar{E}$	$(A \vee \bar{E}) + A$
$H$	$H$	$H$	$H$	$A$	$A$	$A$	$A - I$

Функція переносу діє тільки при виконанні арифметичних дій, але в результатах цих операцій символи  $AE$  означають операцію I.

Якщо АЛП використовується в якості компаратора (операція  $A \vee \bar{E}$ ), то при  $A = E$  на відповідному виході формується сигнал високого рівня. При  $A \neq E$  відповідний сигнал формується на виході  $P_{i+1}$ .

Не вдаючись у подробиці роботи АЛП, слід зазначити, що в зв'язку з інтенсивним розвитком мікропроцесорної техніки розглянута схема АЛП у реальній схемотехніці майже не використовується. Її вивчення має скоріш методичне значення, адже дає можливість практично вивчати ідеологію побудови мікропрограмних автоматів і мікропроцесорів.

### 3.5. Пристрої контролю

#### 3.5.1. Контроль парності

Незважаючи на високу надійність та завадостійкість цифрових систем передачі інформації, вірогідність виникнення похибок завжди існує. Тому всі високонадійні канали передачі інформації забезпечуються допоміжними схемами, які дають можливість впевнитись в відсутності похибок. У будь-якій структурі каналу передачі інформації з контролем похибок повинна бути надмірність каналу. Наприклад, якщо передається код  $0101_2 = 5_{10}$  і на виході каналу з'являється  $1101_2 = 13_{10}$ , то в загальному плані похибку без спеціальних перевірок визначити неможливо. Але, якщо ми знаємо, що інформація передається у двійково-десятковому коді, то одержаний результат хибний. Тобто, наявність шести надлишкових станів дає можливість виявити деякі похибки.

Простий спосіб визначення похибок в словах, які передаються, базується на припущенні, що найбільша вірогідність збою можлива тільки в одному біті, тобто при появі помилкової одиниці або нуля. Тому для визначення наявності такого збою використовують *контроль парності* або *контроль непарності*

одиниць у переданому слові (*Parity Check*). В основі цього способу лежить операція знаходження суми за модулем **2** всіх двійкових розрядів

Таблиця 3.13

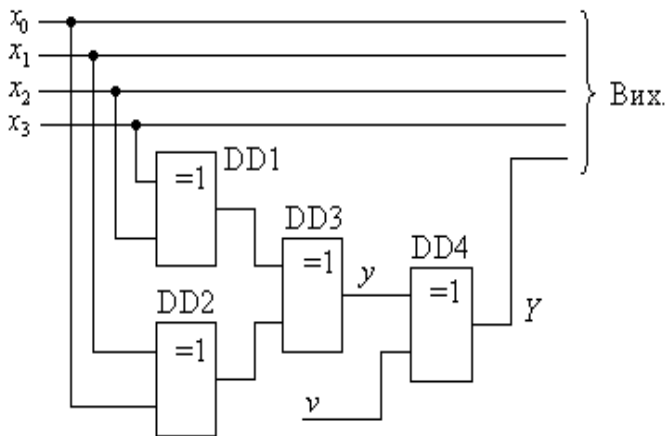


Рис. 3.46

$x_3$	$x_2$	$x_1$	$x_0$	$Y (v = 0)$	$Y (v = 1)$
0	0	0	0	0	1
0	0	0	1	1	0
0	0	1	0	1	0
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	1	1	0
1	1	1	0	1	0
1	1	1	1	0	1

контрольованого слова. При парній кількості одиниць вказана сума дорівнює **0**, а при непарній – **1**. Прикладом побудови такого пристрою контролю є схема, що приведена на рис. 3.46.

Сигнал на виході **Y** визначається наступною логічною функцією:

$$Y = x_0 \oplus x_1 \oplus x_2 \oplus x_3 \oplus v = y \oplus v.$$

З аналізу роботи пристрою, схема якого приведена на рис. 3.46, витікає, що при парній кількості одиниць у слові  $x_3 x_2 x_1 x_0$  значення виходу  $y = 0$ . При непарній –  $y = 1$ . Тому значення **Y** залежить від сигналу **v**, який задає режим контролю – контроль парності або контроль непарності.

При контролі парності задається  $v = 0$ . Якщо відбудеться контроль за непарною кількістю одиниць, то задається  $v = 1$  (див. табл. 3.13).

Вихід **Y** називається *контрольним бітом*. При прийомі інформації одержане слово знов перевіряється на парність або непарність. Якщо при контролі парності в приймальному контрольному пристрої з'явилась **1**, то це значить, що або в інформаційній шині, або в контрольному біті з'явилась похибка. Як бачимо, такий простий контроль не дає можливості виправити похибку, але він дає можливість визначитись з хибною інформацією, щоб потім або використати її, або відкоригувати.



У практиці контролю переважно використовують спосіб контролю по непарності. Пов'язано це з тим, що при появі слова з усіма нулями при контролі по парності цю ситуацію неможливо буде відрізнити від обриву лінії зв'язку.

Приведена схема не може визначити подвійної похибки і будь-якої кількості парних похибок. Пояснюється це лише малою надмірністю лінії зв'язку. Тому для більш глибокого контролю необхідно мати і більшу її надмірність. Оскільки цифрові лінії зв'язку дуже надійні і ймовірність появи двох незалежних похибок  $p^2 = p_1^2$ , тому навіть такий простий контроль суттєво піднімає надійність передачі інформації. Завдяки цьому цей вид контролю використовується практично в усіх каналах передачі цифрових даних, а також при роботі процесора з запам'ятовуючими пристроями, контролюючи тим самим і пристрої пам'яті.

Пристрої контролю парності/непарності двійкових слів випускаються у вигляді самостійних мікросхем у різних серіях (КМОН – К561СА1; ТТЛ – К1533ИП2, К531ИП5).

Мікросхема К1533ИП2 (рис. 3.47) (зарубіжними аналогами мікросхеми 74180РС, СDB4180Е, SN74180N, UCY74180N) має вісім інформаційних входів  $I_0 \div I_7$  та два входи **OE** та **EE**, які дозволяють забезпечити різні режими роботи: керування полярністю вихідних сигналів, нарощування розрядності схем контролю, забезпечення контролю як по парності, так і по непарності одиниць.

Виходи  $S_1, S_2$  взаємно інверсні. Вихід  $S_1$  є виходом непарності.

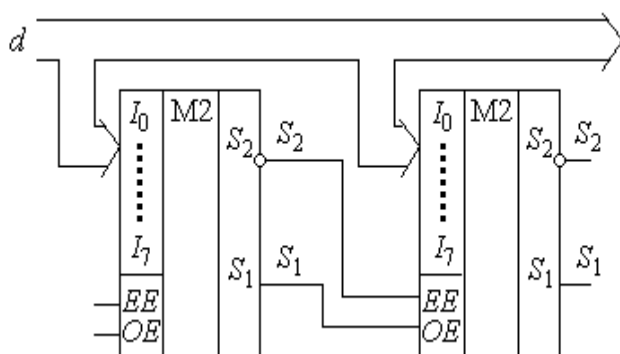


Рис. 3.47

Таблиця 3.14

Число одиниць на $I_0 \div I_7$	EE	OE	$S_1$	$S_2$
Парне	1	0	1	0
Непарне	1	0	0	1
Парне	0	1	0	1
Непарне	0	1	1	0
X	1	1	0	0
X	0	0	1	1

Табл. 3.14 надає можливість визначитись з необхідними вхідними сигналами на входах **EE** і **OE** мікросхеми K15533ИП2 для забезпечення контролю по парності або непарності.

Наприклад, при вхідних сигналах, відповідних  $EE=1$  ;  $OE=0$ , парність коду на входах  $I_0 \div I_7$  відобразатиметься високим рівнем виході  $S_1$  та низьким на  $S_2$ . Якщо ж на входах  $I_0 \div I_7$  код непарний, то на виходах  $S_1, S_2$  буде код, інверсний попередньому.

Слід звернути увагу на те, що при парності коду на входах  $I_0 \div I_7$  маємо наступні логічні співвідношення між вхідними і вихідними сигналами:

$$S_1 = EE ; S_2 = OE ;$$

при непарності коду:

$$S_1 = \overline{EE} ; S_2 = \overline{OE} .$$

Нарощування довжини слова, що підлягає контролю, забезпечується послідовним з'єднанням мікросхем, причому виходи  $S_1$  та  $S_2$  попередньої мікросхеми з'єднані з входами **EE** та **OE** наступної.

### 3.5.2. Корекція помилок

Розглянутий спосіб оцінки достовірності передачі інформації по каналу зв'язку не дає можливості знайти похибку в інформаційному слові. Тому в складних комп'ютеризованих та мікропроцесорних схемах з розгалуженими каналами цифрового зв'язку використовують системи передачі з використанням таких кодів, які дозволяють як знаходити, так і виправляти випадкові похибки. До них відносяться, наприклад, код Хемінга, циклічні коди, які використовуються для виявлення та виправлення похибок не тільки при передачі, але й при зберіганні даних; код Ріда-Соломона, який широко використовується при записі та зчитуванні компакт-дисків, та інші.

Розглянемо ідеологію та принцип побудови цифрових кодерів з виправленням похибок на прикладі використання коду Хемінга.

Припустимо, що необхідно визначити та виправити одиничну похибку двійкового коду з  $n$  інформаційними символами (розрядами). До цього коду додається  $m$  контрольних символів. В результаті загальна кількість символів у коді Хемінга

$$N = n + m. \quad (3.13)$$

При передачі даних може бути спотворений будь який як інформаційний, так і контрольний символ. Окрім того може виникнути ситуація коли жоден з розрядів повідомлення не буде спотворений. Для ідентифікації  $N$  одиночних похибок та однієї безпомилкової комбінації за допомогою  $2^m$  комбінацій контрольних символів їх кількість повинна бути

$$2^m \geq N + 1 \quad (3.14)$$

або відповідно до (3.13)

$$2^m = 2^{N-n} = 2^N \cdot 2^{-n}. \quad (3.15)$$

З (3.14) з врахуванням (3.15) знаходимо:

$$2^N \geq 2^n \cdot (N + 1). \quad (3.16)$$

З (3.16) знаходиться кількість комбінацій інформаційних символів:

$$2^n \leq 2^N / (N + 1). \quad (3.17)$$

Критерій оптимальності коду Хемінга має вигляд:

$$2^{N-n} \geq \sum_{i=1}^r C_{N^r}, \quad (3.18)$$

де  $C_{N^r}$  – кількість комбінацій з  $N$  по  $r$ ;  $r$  – кількість виявлених незалежних похибок.

При  $r = 1$  (тобто для одиночних похибок) формула (3.18) має вигляд:

$$2^{N-n} - 1 = N. \quad (3.19)$$

Формула (3.19) є нижньою межею коду, адже встановлює те мінімальне співвідношення коригуючих та інформаційних розрядів, нижче якого код не може зберігати свої коригуючі здібності.

Як приклад, розглянемо ситуацію, коли інформаційне слово має **11** розрядів ( $n=11$ ). У цьому випадку формула (3.17) перетворюється в рівність при  $N=15$ . Тобто кількість коригуючих розрядів  $m=4$ .

Пояснимо, що можливість виправлення похибки в кодї Хемінга базується на повтореній  $k$  разів процедури контролю парності, але не всього кодового слова одночасно, а  $k$  окремих груп його розрядів. Слово розбивається на групи так, щоб номер кожного розряду визначався за його належністю до цих груп. Вказані групи називаються *групами контролю парності* і компонуються з розрядів кодового слова на основі наступних правил:

а) кожен розряд кодового слова кодується в відповідності до принципів перетворення десяткового коду в двійковий. Оскільки в нашому випадку  $N=15$ , то для кодування кожного розряду кодового слова необхідно мати  $k=4$  розряди двійкового коду. При цьому створюється  $k=4$  групи контролю парності;

б) в кожен  $i$ -у контрольну групу входять ті розряди кодового слова, в двійковому номері яких в  $i$ -й позиції знаходиться одиниця. Ідеологія створення контрольних груп пояснюється рис. 3.48.

На рис. 3.48 показаний приклад організації кодового слова, яке планується передавати по лінії зв'язку. Воно має одинадцять інформаційних розрядів, які позначені малими латинськими літерами  $a \div r$ , та чотири контрольні розряди, позначені буквами грецького алфавіту  $\alpha, \beta, \gamma, \delta$ . Контрольні розряди розміщуються в масиві кодового слова на позиціях тих розрядів, в двійкових номерах яких присутня тільки одна «1» ( позиції 1, 2, 4, 8...).

Відповідно до вказаних правил формування контрольних груп, 4-й розряд входить всього до однієї – третьої – групи; розряд 6 – до другої та третьої; розряд 10 – до другої та четвертої, і т. д. Тому четверта група включає в себе лише розряди від 8-го по 15-й, а третя 4-7 та 12-15. Наймолодший розряд кожної групи є *контрольним*.



сигнали разом з контрольним розрядом на схемі контролю парності **M2**. Виходи блоків **M2<sub>1</sub> – M2<sub>4</sub>** створюють керуючий код **K**. Якщо при передачі кодового слова в одному з його розрядів з'явилась похибка (на рис. 3.48 приведений приклад появи похибки в 5-му розряді, звідки виходять лінії), то буде зафіксовано порушення парності одиниць в розрядах коригуючого коду, відповідних контрольним групам з розрядом похибки. В результаті коригуючий код вказуватиме на номер того розряду кодового слова, в якому з'явилась похибка. Так, на рис. 3.48 контрольний код **0101** вказує на 5-й розряд.

Для відновлення інформації в слові, що передавалося, необхідно в визначеному розряді проінвертувати його значення. Схема, яка вирішує цю задачу, приводиться на рис. 3.50.

Кодове слово, що приймається вхідним регістром, перетворюється на чотири восьмирозрядні кодові групи, які контролюються на парність одиниць восьмирозрядними блоками **M2<sub>1</sub> – M2<sub>4</sub>**. Виходи блоків контролю парності дешифруються, і, якщо в *i*-му розряді має місце похибка, то одиничний сигнал подається на *i*-й керований інвертор і змінює стан розряду інформаційного слова.

При відсутності збоїв в передачі інформації одиничний сигнал з'являється на нульовому вході дешифратора.

Коди Хемінга використовуються там, де вимоги до точності передачі даних досить високі.

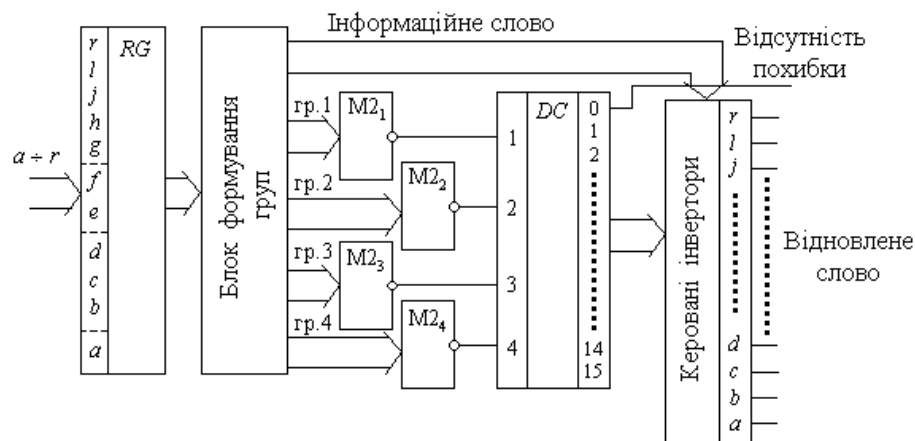


Рис. 3.50

### 3.5.3. Контроль логічних перетворень

Розглянуті в попередніх параграфах засоби контролю правильності передачі кодів виконують контроль лише незмінних даних. Якщо коди піддаються логічним перетворенням, то засоби, контролюючі їх правильність, стають дуже складними. Найпростіший спосіб контролю – це дублювання логічних перетворень та їх контроль (рис. 3.51).

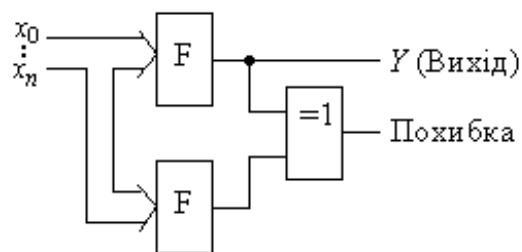


Рис. 3.51

Відповідно до рисунку, логічна схема  $F$  дублюється аналогічною схемою. Це дає можливість забезпечувати безперервний контроль вихідного потоку інформації і у разі їх неспівпадання видавати сигнал похибки. При такій формі контролю слід враховувати той факт, що збої в роботі цифрової схемотехніки можуть з'являтися не тільки за рахунок дефектів схеми, але й при наявності завод по живленню. Тому при однакових цифрових схемах велика ймовірність появи однакових збоїв. Це значить, що дублюючі пристрої необхідно виготовляти на різних схемах. Один з таких варіантів побудови дублюючих схем – це використання взаємно протилежних логічних схем відповідно до теореми де Моргана.

У ряді випадків для передачі контрольованої інформації використовуються парафазні коди, коли дублюється канал передачі інформації, і по другому каналу завжди передається сигнал, інверсний сигналу першого каналу; якщо ці сигнали в кожний тактовий момент скласти за модулем 2, то нульовий результат буде завжди говорити про наявність похибки в одному з каналів.

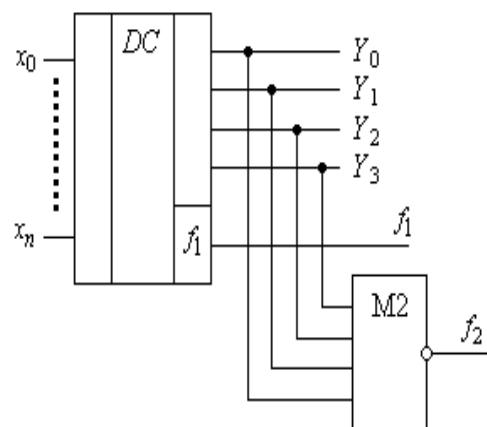


Рис. 3.52

Комбінаційні схеми, що реалізують систему булевих функцій, на відміну від одновихідних схем, можливо контролювати, не використовуючи їх дублювання. Приклад схеми з таким контролем приводиться на рис. 3.52.

Для цього при проектуванні кодового перетворювача необхідно закласти реалізацію ним допоміжної функції  $f_1$ , яка задається як функція аргументів, значення якої співпадають із значеннями суми по модулю 2 решти функцій. Протифазна функція  $f_2$  є схемою контролю парності всіх функціональних виходів перетворювача, окрім  $f_1$ . Одиночна похибка в роботі як схеми перетворювача, так і схеми **M2** змінить значення парності або  $f_1$ , або  $f_2$  і буде зафіксована схемою контролю. Функція  $f_1$  повинна синтезуватись з використанням вхідних змінних  $x_0 \div x_n$ . Якщо ж використати для її побудови вихідні значення  $Y_0 \div Y_3$ , то можливі похибки можуть змінити одночасно як  $f_1$ , так і  $f_2$  і не будуть виявлені. Вказану особливість необхідно враховувати при розробці схем контролю більш складних перетворювачів, в яких декілька різних виходів можуть мати загальні схеми і, відповідно, загальні збої, які потім не можуть бути виявлені схемами типу **M2**.

### 3.6. Буферні елементи

При побудові складних цифрових і мікропроцесорних систем створюється шинна форма обміну інформацією між окремими пристроями. *Шина* – це паралельна група провідників, по якій передаються інформаційні сигнали в обох напрямках і до якої може бути приєднана досить велика група пристроїв, що передають та приймають інформаційні сигнали. В загальному плані в залежності від характеру сигналів, що передаються і приймаються, шини класифікуються на *адресні*, *командні* та *шини даних*. По адресних шинах йдуть сигнали звернення до тих чи інших пристроїв, що приєднані до шин; по шині керування передаються керуючі сигнали, а по шині даних – безпосередньо дані.



Оскільки кожен з пристроїв повинен мати право доступу до шин, то для забезпечення необхідного порядку при роботі з шинами існує велика кількість мікросхем, які спеціально призначені для обслуговування різноманітних пристроїв при їх роботі з шинами. Такі мікросхеми здебільшого не виконують логічних функцій. Вони призначені для:

- формування цифрових сигналів для забезпечення необхідного струму, напруги, потужності;
- забезпечення тимчасової передачі шин від одних пристроїв до інших;
- розділення в часі інформації, що передається в обох напрямках;
- підключення (чи відключення) до шин необхідних пристроїв;
- виконання допоміжних функцій – таких, як тимчасове запам'ятовування інформації, що передається по шині, її інверсія і т. п.

Всі вказані функції забезпечуються спеціальними *буферними елементами*. Основою будь-якого буферного елемента є підсилювач. Оскільки по шині інформація передається в обох напрямках, то підсилювач повинен бути двонаправленим. Через те, що двонаправлений підсилювач може збуджуватись, то він виготовляється у вигляді двох ідентичних, зустрічно-паралельно з'єднаних підсилювачів потужності, кожен з яких вмикається лише при передачі інформації від його входу до виходу. Для забезпечення такої комутації підсилювачів використовується спеціальна логіка. Кількість пар підсилювачів відповідає кількості провідників шини. Якщо, наприклад, шина даних має розрядність 8 біт, то мікросхема шинного підсилювача повинна мати 8 пар зустрічно-паралельно з'єднаних підсилювачів, керованих єдиним логічним вузлом.

Для забезпечення необхідної потужності використовуються різні схемотехнічні прийоми, що детально описані в **Розділі 2**.

У мікросхемах обслуговування широко використовуються логічні елементи з Z-станом. Переведення мікросхеми у Z-стан дає можливість

тимчасово відділити шини від цифрових пристроїв, внаслідок чого нею можуть користуватися інші пристрої.

У технічній літературі буферні елементи часто називають “шинними драйверами”. До них відносяться мікросхеми K1533АП3 (SN74ALS240), K1533АП4 (SN74ALS241), а також мікросхеми груп ЛН, ЛП, ХЛ.

### 3.7. Перехідні процеси в логічних схемах

Відповідно до **Розділу 2**, логічні ключі характеризуються часом затримки  $t_3$ , який пов’язаний з перехідними процесами зміни стану. Тривалість перехідних процесів не є стабільною величиною і залежить від багатьох факторів. При використанні мікросхем, в яких внутрішні затримки малі, при проектуванні цифрових пристроїв слід обов’язково враховувати тривалість перехідних процесів у лініях зв’язку та вплив параметрів ліній зв’язку на тривалість перехідних процесів в мікросхемах. Ці задачі досить складні, і їх розв’язання можливе лише з використанням спеціальних методів моделювання.

При використанні мікросхем з тривалістю зміни станів, не меншою 20 нс (ТТЛ, КМОН), перехідні процеси в лініях зв’язку не враховуються, а характер лінії зв’язку, величина та характер навантаження враховуються максимальною тривалістю затримки. Це значно спрощує процес проектування цифрових пристроїв, адже затримки можуть бути враховані відразу на стані логічного проектування. В таких ситуаціях, коли мають місце значні затримки в лініях зв’язку, вони також можуть враховуватись як дискретні затримки.

На жаль, у паспортних даних на мікросхеми подається тільки максимальна тривалість затримки. Реально ж вона може змінюватись в широких межах, а мінімальна величина затримки паспортно не визначена. Звідси витікає, що рівень виходу елемента в цифровій схемі в інтервалі часу від 0 до  $t_3$  не визначений. Він називається *станом невизначеності* і позначається символом  $x_H$ . Такий стан елемента в схемі впливає на інші елементи, на виходах яких можуть мати місце як визначені стани “1”, “0”, так і

невизначений стан  $x_H$ . Як результат, поведінка логічних елементів в перехідних процесах описується законами трійкової логіки з такими логічними співвідношеннями:

$$\overline{x_H} = x_H ;$$

$$x_H \cdot 1 = x_H ;$$

$$x_H \cdot 0 = 0 ;$$

$$x_H + 0 = x_H ;$$

$$x_H \oplus 0 = x_H ;$$

$$x_H + 1 = 1 ;$$

$$x_H \oplus 1 = \overline{x_H} ;$$

$$x_{H1} \cdot x_{H2} = x_H ;$$

$$x_{H1} + x_{H2} = x_H ;$$

$$x_{H1} \oplus x_{H2} = x_H .$$

Приведені формули широко використовуються при аналізі перехідних процесів, в тому числі при побудові часових діаграм. Останні зображаються одним з двох способів, які приведені на рис. 3.53, *a* для елемента **2I** ( $y = x_0 \cdot x_1$ ).

У практиці аналізу схем використовуються чотири типи зображення перехідних процесів, які використовуються в залежності від необхідного рівня деталізації. Найпростіша з них – діаграма рис. 3.53, *a* – пояснює лише логічні зв'язки між вхідними сигналами та вихідними станами без урахування вихідних затримок.

Часова діаграма, приведена на рис. 3.53, *б*, враховує часові затримки, причому їх максимальне значення. Вона дає можливість оцінити максимальні значення тривалості перехідних процесів. Лінії зі стрілками показують взаємозв'язки між вхідними сигналами і вихідними станами мікросхеми.

Діаграми рис. 3.53, в, з враховують стани невизначеності елементів. Вони дають можливість змоделювати поведінку схеми при будь-яких комбінаціях і часових співвідношеннях внутрішніх затримок. Інтервал  $\tau$  невизначеного стану  $x_H$  дорівнює величині  $t_3$  конкретної мікросхеми.

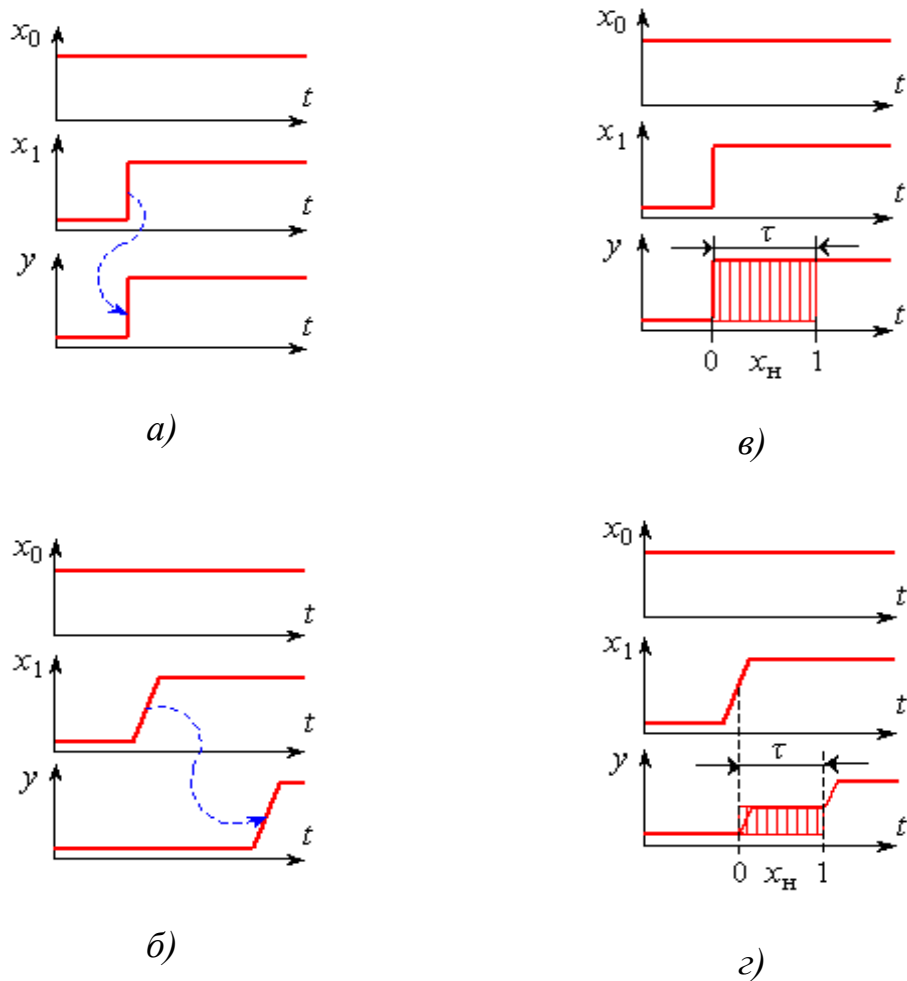


Рис. 3.53

Врахування станів невизначеності дає можливість виявити важливий ефект, який обов'язково необхідно враховувати при аналізі цифрових схем.

Розглянемо схему, що приведена на рис. 3.54, а, на вхід якої подається сигнал  $x_0$  у вигляді одиночного імпульсу тривалістю  $T$ .

Якщо тривалості інтервалів невизначеності  $\tau$  при зміні станів мікросхем з "0" в "1" і з "1" в "0" однакові, то на виході  $y_1$  матимемо сигнал, тривалість і початок якого суттєво залежить від величини затримки  $\tau$  та співвідношення між

затримками при зміні станів з “0” в “1” та навпаки. Якщо, наприклад, тривалість затримок  $\tau = 0$ , то вихідний імпульс  $y_{10}$  повторюватиме вхідний. Якщо затримки на зміну станів з “1” в “0” і навпаки однакові і максимальні, то сигнал  $y_{1m}$  матиме затримку по відношенню до вхідного на величину  $2\tau$ .

При різних співвідношеннях між затримками вихідний сигнал може бути як скороченої довжини  $y_{1c}$ , так і видовженої  $y_{1b}$  (рис. 3.54, б).

Звичайно, що реально сигнал  $y_1$  може мати випадкові співвідношення між тривалістю імпульсу та паузою, випадкову затримку початку і кінця імпульсу. Як результат, імпульс, який буде пропущений через два ідентичні канали, на їх виходах може суттєво відрізнятись. Якщо ж один канал містить  $m$  елементів, а другий  $n$ , то, відповідно, матимемо затримки на  $m\tau$  і  $n\tau$ , і вихідні імпульси можуть навіть не співпадати в часі. Якщо при проектуванні схеми, що містить послідовно з'єднані  $k$  елементи, необхідно одержати імпульс тривалістю  $t_i$ , то в результаті вихідний імпульс з двостороннім допуском буде тривалістю  $t_i \pm k\tau$ .

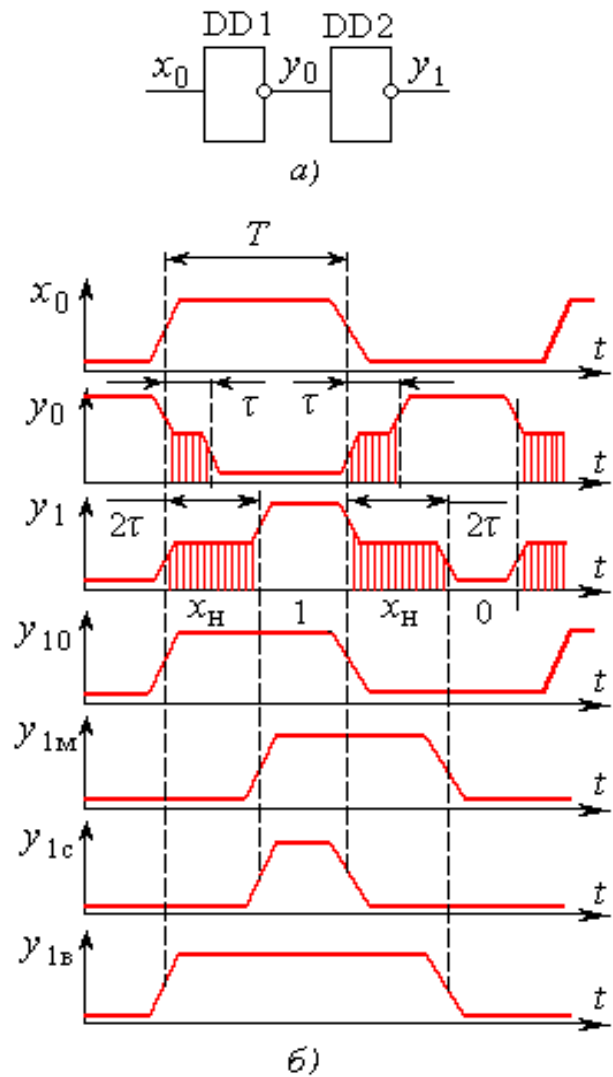


Рис.3.54

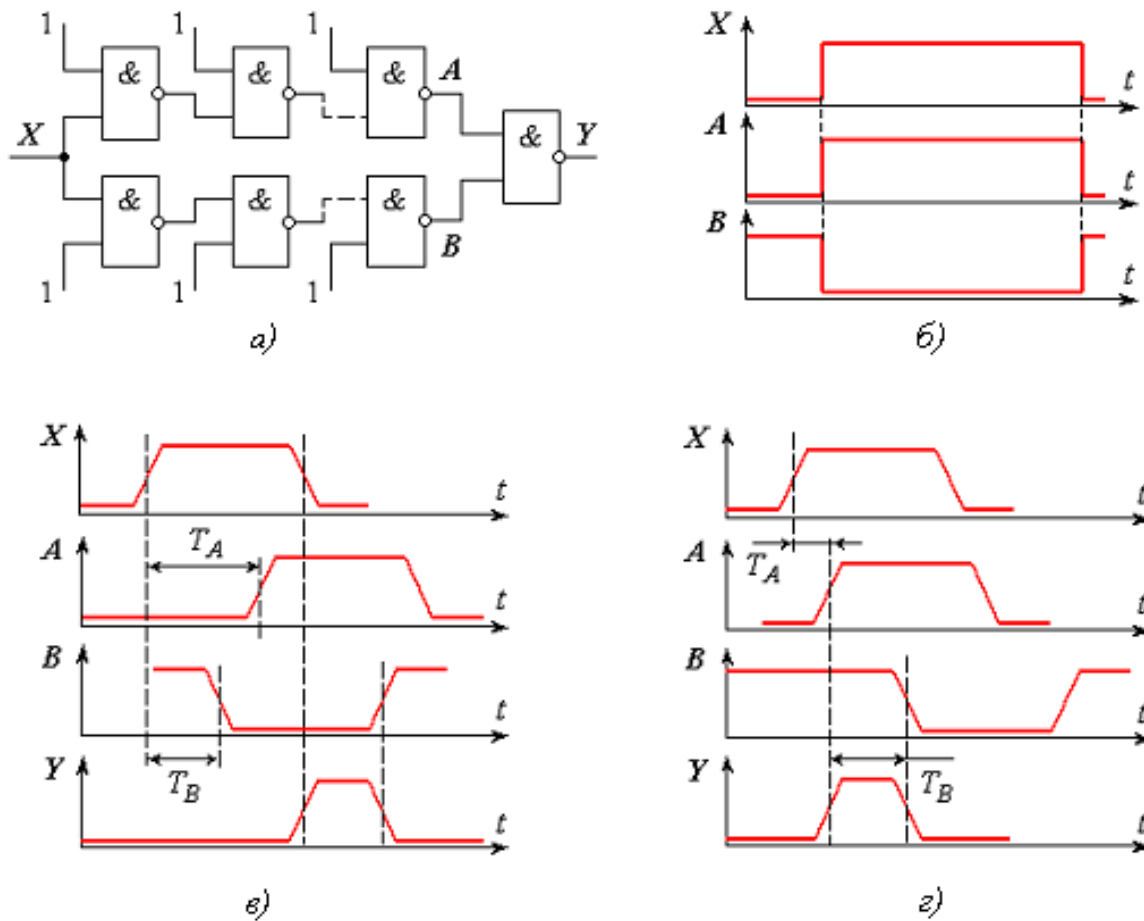


Рис. 3.55

Найбільші проблеми наявність затримок створює при передачі імпульсів через паралельні канали. Розглянемо схему, що приведена на рис. 3.55, *а*. Вхідний імпульс подається на два паралельних канали з елементами **I-НІ**, один з яких назвемо  $A$  – з парною кількістю елементів, а другий  $B$  – з непарною. При наявності логічних одиниць на одному з входів кожного логічного елемента обидва канали є прозорими для сигналу  $x$ . При відсутності часових затримок в каналах, їх вихідні сигнали  $A$  і  $B$  на вході об'єднуючого елемента **I-НІ** будуть строго протифазні і, як результат,  $y = 0$  (рис. 3.55, *б*).

На рис. 3.55, *в*, *г* приведені приклади часових діаграм в тих ситуаціях, коли, відповідно, затримки  $T_A > T_B$  і  $T_B > T_A$ . В обох випадках на виході  $y$  з'явиться сигнал, який неможливо прогнозувати на основі алгебри Буля без врахування часових затримок. Одержаний сигнал є сигналом завади, що може привести до непередбачених наслідків в наступних схемах.

Розглянуте явище називається „гонками” або „змаганнями”. Головна проблема, яка пов’язана з гонками, полягає у тому, що проєктант не має реальних можливостей визначити момент появи можливої завади та її тривалість.

Гонки називаються *критичними* або *недопустимими*, якщо хоча б один вихідний сигнал під час перехідного процесу змінюється більш ніж один раз. Критичні змагання суттєво впливають на роботу цифрових пристроїв та їх проєктування.

Розглянемо схему, що приведена на рис. 3.56. При  $x=1$  на обох виходах  $y_{\tau 1}$  і  $y_{\tau 2}$  матимемо сигнали низького рівня. Зміна  $x$  з “1” в “0” приведе до того, що на виході з меншим значенням  $\tau$  (наприклад,  $\tau_2 < \tau_1$ ) з’явиться одиничний сигнал раніше, ніж на іншому.

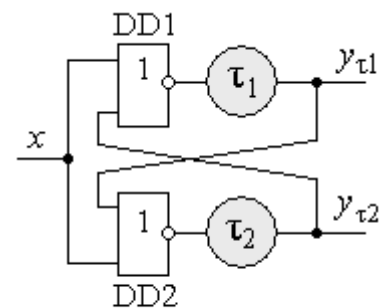


Рис. 3.56

Він забезпечить підтримку одиничного значення на вході DD1, внаслідок чого  $y_{\tau 1} = 0$ . Оскільки значення  $\tau_1$  і  $\tau_2$  невідомі, то такий перехід приведе до невизначеності станів  $y_{\tau 1}$  і  $y_{\tau 2}$ .

На практиці використовуються три наступні способи боротьби з гонками:

- *синхронізація;*
- *побудова протигоночних систем;*
- *врахування мінімального часу затримки.*

*Синхронізація* є найбільш універсальним засобом боротьби з гонками. Її суть полягає у наступному: по всьому цифровому пристрою створюється єдина система синхронізуючих сигналів. У залежності від типу елементної бази, використовуються однофазна або двофазна системи синхронізації. В останньому випадку синхронізація забезпечується двома послідовностями імпульсів *C1* та *C2*, одна з яких має затримку по відношенню до іншої на половину періоду.

Для розглянутої вище схеми з паралельними каналами (див. рис. 3.55, *a*) синхронна схема з двофазною синхронізацією приведена на рис. 3.57, *a*.

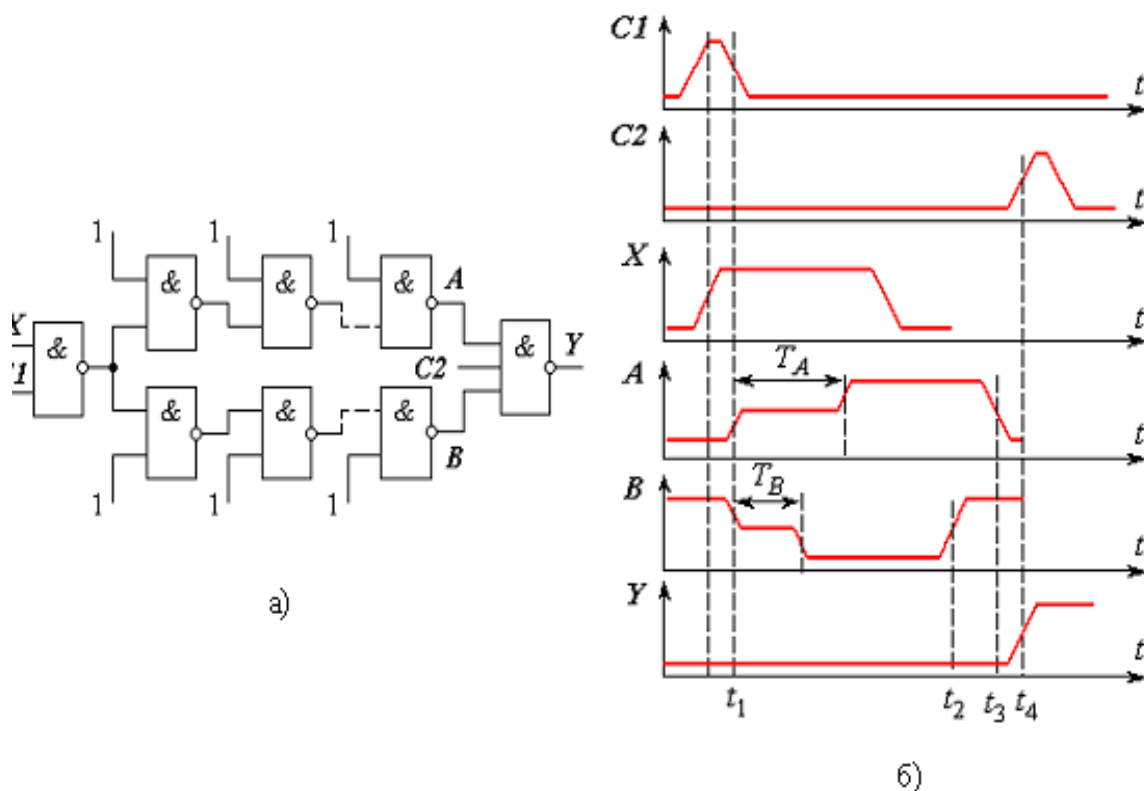


Рис.3.57

За синхросигналом  $C1$  сигнал  $x$  одночасно подається на два канали. Проходячи канали з різними затримками  $T_A$  і  $T_B$ , в асинхронній схемі (рис. 3.55, *a*) на виході  $Y$  в інтервалі часу  $t_2 \div t_3$  з'являється завада. Але у синхронній схемі інформація на виході  $Y$  з'явиться лише після подачі синхроімпульсу  $C2$  в момент  $t_4$ , коли всі перехідні процеси в каналах завершаться. Важливе значення в цьому випадку має інтервал  $t_1 \div t_4$  між двома синхроімпульсами, який вибирається таким, щоб перехідні процеси найбільшої тривалості могли завершитись до подачі  $C2$ .

У залежності від того, що виступає приймачем сигналу  $Y$ , залежить назва сигналу  $C2$ . В розглянутому випадку, коли сигнал  $Y$  не запам'ятовується, а  $C2$  використовується лише для того, щоб в  $Y$  не з'являлась завада, він називається *стробом*, а процес відсікання завад – *стробуванням*. Якщо ж необхідно



запам'ятати в схемі пам'яті, то сигнал  $C2$  подається на цю схему, що робить її синхронною, а  $C2$  – відповідно *синхросигналом*.

Слід зазначити, що синхронізація дає можливість суттєво спростити процес проектування цифрових схем, адже значно спрощує вирішення проблеми гонок.

*Протигоночні системи* – це другий практичний спосіб боротьби з гонками. Вони будуються таким чином, що в них відсутній ризик появи на виході сигналів, не передбачених логікою роботи схеми. Прикладом подвійної схеми може бути два паралельних канали з однаковою кількістю елементів, об'єднаних елементом **АБО**; який би канал не виграв гонку, результат буде однаковим, зміниться лише момент його появи.

Як відмічалось у **Розділі 1**, для усунення гонок необхідно створювати паралельні канали передачі сигналів. Задача створення мінімально необхідної кількості паралельних каналів досить чітко формалізується при використанні карт Карно (діаграм Вейча). Сутність такої формалізації полягає в тому, щоб будь-які сусідні клітини карти Карно були охоплені щонайменше одним багатовимірним кубом. Мінімізована таким чином логічна функція буде вільною від гонок.

Практично асинхронні схеми достатньо складні в проектуванні і поступово витісняються синхронними.

**Приклад 3.22.** Логічна функція задана картою Карно (рис. 3.58). Мінімізувати функцію для отримання диз'юнктивної нормальної форми, вільної від гонок.

*Розв'язання.* Для забезпечення диз'юнктивної форми можна об'єднати наступні групи клітинок: група № 1 – **1, 3, 5, 7**; група № 2 – **8, 9, 12, 13** і група № 3 – **6, 14**. В результаті отримаємо

$$y = \overline{x_3} \overline{x_0} + \overline{x_3} \overline{x_1} + \overline{x_2} \overline{x_1} \overline{x_0}.$$

Функція буде мінімальною, але не вільною від гонок, оскільки такі сусідні клітини, як **6, 7; 1, 9; 12, 14** не охоплюються багатовимірними кубами.

Виконання умови відсутності гонок приводить до необхідності об'єднання вказаних клітин, що може бути забезпечено двома 2-кубами (**7, 6**) і (**12, 14**) і одним 4-кубом (**1, 5, 13, 9**).

		$x_1 x_0$			
		00	01	11	10
$x_3 x_2$	00	0	1	3	2
	01	4	5	7	6
	11	12	13	15	14
	10	8	9	11	10

Рис. 3.58

В результаті отримаємо функцію:

$$y = \overline{x_3} x_0 + x_3 \overline{x_1} + x_2 x_1 \overline{x_0} + \overline{x_3} x_2 x_1 + x_3 x_2 \overline{x_0} + x_1 x_0,$$

яка буде вільною від гонок.

Третій спосіб – *урахування мінімального часу затримок* – знаходить використання при проектуванні таких цифрових схем, які потім виготовлятимуться на одному кристалі. Це пов'язано з тим, що проєктант паралельно зі схемотехнікою закладає й технологічні особливості виготовлення схеми з необхідними параметрами. В результаті у проєкт можна закласти необхідні часові затримки в окремих каналах або схемах. Використання таких прийомів у дискретній цифровій схемотехніці вимагає великого досвіду і уваги.

У будь-якому випадку, після проектування цифрових схем їх аналізують на предмет можливої появи завад в результаті гонок. На практиці з цією метою використовуються спеціальні комп'ютерні програми.

Ще одним прикладом появи завад завдяки гонкам є прийом сигналів з недостатньо крутими фронтами. Така ситуація має місце тоді, коли подвійний сигнал подається на паралельні канали з різними типами мікросхем. У такому випадку в деякому інтервалі рівнів вхідного сигналу для мікросхеми одного типу сигнал відповідатиме рівню логічного нуля, а для іншого – логічної одиниці. Це приведе до появи хибних сигналів. Це явище називається „*гонками по виходу*”. Для боротьби з ним використовуються спеціальні прийоми попередньої обробки вхідних сигналів.

## КОНТРОЛЬНІ ПИТАННЯ

1. Які теореми булевої алгебри використовуються при реалізації логічних функцій в базисі **I-НІ, АБО-НІ**?
2. Які проблеми необхідно розв'язувати при необхідності збільшення коефіцієнта розгалуження? Якими шляхами досягається збільшення коефіцієнта розгалуження по виходу?
3. Якими шляхами досягається нарощування входів? Наведіть приклади.

4. Дайте визначення мультиплектора. Наведіть приклади його використання.

5. Чи можливо за допомогою мультиплектора реалізовувати логічні функції? Чому?

6. Які способи нарощування розрядності мультиплектора Вам відомі? Наведіть приклади.

7. Дайте визначення пристроям здвигу. Навіщо і де вони використовуються?

8. Поясніть призначення селекторів і особливості їх побудови.

9. У чому полягає суть шифрації коду?

10. Поясніть особливості роботи пріоритетних шифраторів.

11. Обґрунтуйте можливість використання дешифраторів-демультиплекторів для реалізації функцій алгебри логіки.

12. У яких випадках корисно використовувати дешифратори для реалізації булевих функцій?

13. Які булеві функції використовуються для виконання арифметичних операцій над однорозрядними словами?

14. Чим визначається швидкодія виконання арифметичних операцій?

15. Як можна підвищити швидкість виконання арифметичних операцій?

16. Які булеві функції використовуються при побудові компараторів?

17. Поясніть необхідність контролю парності при передачі інформації.

18. У чому полягає особливість кодів, в яких передбачене виправлення помилок, що виникають при передачі інформації?

19. Які способи контролю логічних перетворень Вам відомі?

20. У чому проявляється необхідність врахування затримок і перехідних процесів при проектуванні цифрових пристроїв?

21. Що називається “змаганнями” (“гонками”)? Які “гонки” називаються критичними?

22. Які засоби боротьби з гонками використовуються в цифровій схемотехніці?

23. Сформулюйте правило мінімізації логічних функцій для забезпечення мінімальної диз'юнктивної форми, вільної від гонок.

## ВПРАВИ І ЗАВДАННЯ

1. Навести приклади комбінаційних логічних схем, для опису яких необхідно будувати таблиці з сотнями рядків. Яка кількість рядків у таблиці станів повинна бути?

2. Схема пожежної сигналізації складається з 10 гілок, у кожній з яких міститься 10 датчиків. Пояснити, скільки рядків повинна мати таблиця станів логічної функції, що описує роботу пожежної сигналізації.

3. Записати логічну функцію елемента **8І-НІ**. Привести приклад її реалізації на двовходових елементах **2І-НІ**.

4. Реалізувати логічну функцію **8І-НІ** з використанням елементів **2І-НІ** на основі монтажною логіки.

5. Розв'язати вправи **3** і **4** стосовно елемента **8АБО-НІ**.

6. Реалізувати наступні логічні висловлювання: а) “якщо  $A$ , то  $B$ ”; б) “якщо  $B$ , то  $C$  і  $D$ ”; в) “якщо  $A$ , то  $B$  або  $C$ ”; г) “ $C$  тільки при  $A$  і  $B$ ”; д) “ $D$  при будь-яких двох  $A, B, C$ ”; е) “ $A$  при  $B$  або  $C$ ”.

7. Записати логічну функцію, що реалізується схемою, приведеною на рис. 3.59.

8. Логічну функцію  $y = x_2 \overline{x_1} x_0 + \overline{x_2} \overline{x_0} + x_1$  реалізувати у базисі **2І-НІ**.

9. Логічну функцію з вправи **8** реалізувати у базисі **2АБО-НІ**.

10. Визначити максимальну прогнозовану затримку від входу до виходу у схемі, що приведена на рис. 3.60. Усі мікросхеми ТТЛ КР1533ЛА3 (SN74ALS00).

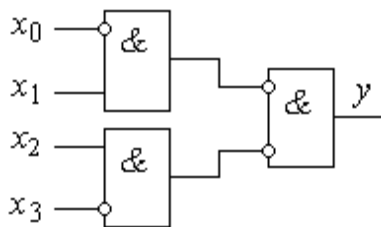


Рис. 3.59

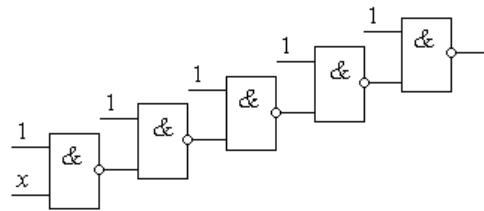


Рис. 3.60

11. Вправу **10** виконати для мікросхеми КМОН КР1554ЛА3 (74АС00).

12. Оцінити максимальну і мінімальну прогнозовані затримки від входу до виходу для пристрою, схема якого приводиться на рис. 3.61 При яких значеннях сигналів  $x_0 \dots x_3$  вони матимуть місце? Усі мікросхеми TTL КР1533ЛП5 (SN74ALS86).

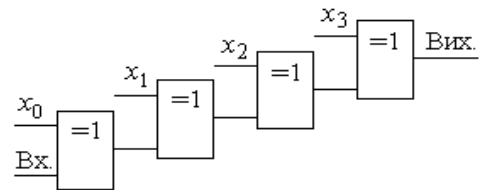


Рис. 3.61

13. Вправу **11** виконати для мікросхем КМОН КР1554ЛП5 (74АС86).

14. Скласти таблицю станів і записати логічну функцію для пристрою, схема якого з використанням американських стандартів приведена на рис. 3.62.

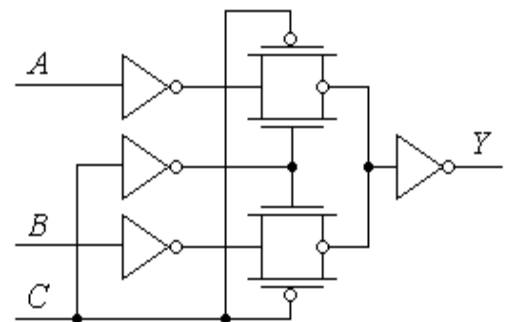


Рис. 3.62

15. На рис. 3.63 приведена логічна схема цифрового пристрою. Побудувати таблицю станів, записати логічні функції по відношенню до її виходів, пояснити призначення цього пристрою, навести приклад його використання.

16. На рис. 3.64 приведена схема цифрового пристрою з використанням американських стандартів. Визначити, які функції виконує цей пристрій. Привести позначення входів і виходів відповідно до умовних позначень цієї групи пристроїв.

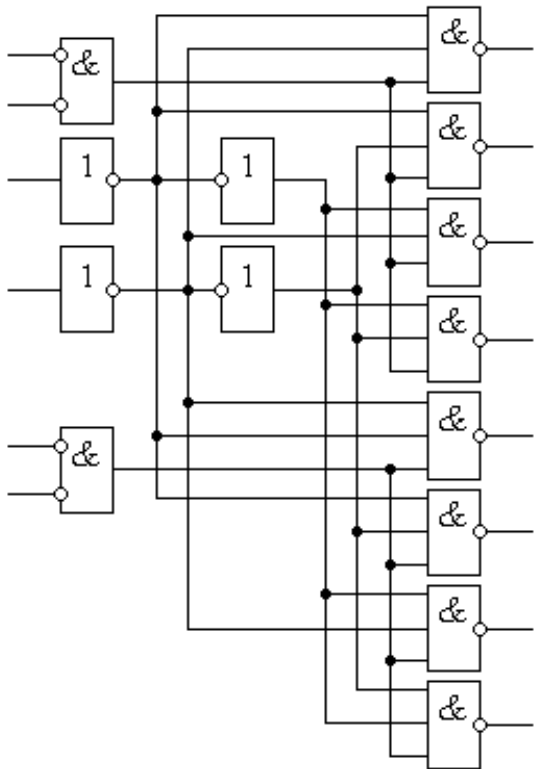


Рис. 3.63

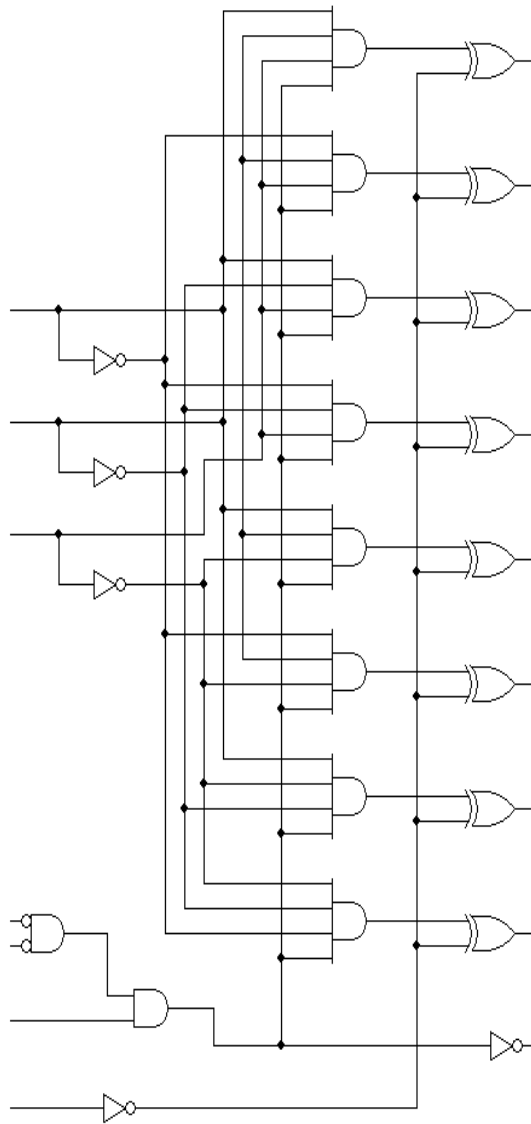


Рис. 3.64

17. Використовуючи мультиплексор “1 з 8” КР1533КП7, розробити схему для реалізації логічної функції:

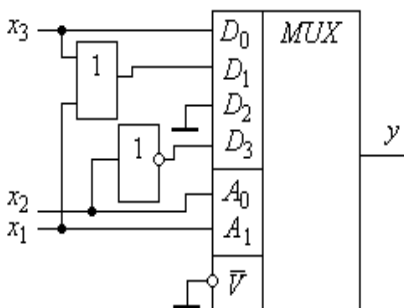


Рис. 3.65

$$y = \bigvee_0^{15} 0, 2, 4, 5, 8, 10, 11, 14.$$

18. На рис. 3.65 приведена схема, яка реалізує невідому логічну функцію. Проаналізувати роботу схеми і записати аналітичний вираз реалізованої логічної функції у скороченій досконалій формі.

19. Використовуючи логічні схеми **ВИКЛ. АБО** та допоміжну логіку, розробити схему компаратора для порівняння двох чотирирозрядних слів, який би видавав на своєму виході сигнал логічної одиниці при рівності двох кодів.

20. З двох каналів даних з частотою 1 кГц зчитуються однобайтні слова, задані в паралельному форматі. Використовуючи мультиплектори **8:1** (KP1533КП7 або аналог SN74ALS151), розробити схему пристрою, за допомогою якого інформація перетворювалася б у послідовний формат з паузою між словами в 2 біти. Побудувати часові діаграми сигналів адресації, які б могли забезпечити необхідне перетворення. Обґрунтувати частоту зміни адресних сигналів для рівномірної передачі інформації у послідовному форматі.

21. Вправу **20** виконати для ситуації, коли інформація приймається з 8 каналів з частотою 10 кГц.

22. Використовуючи мікросхему KP1533КП18 (SN74ALS158), розробити селектор для вибору одного з двох чотирирозрядних слів, заданих у паралельному форматі, і передачі його на паралельну чотирьохрозрядну шину.

23. Розробити пристрій зсуву однобайтного слова. Обґрунтувати організацію адресних сигналів з метою забезпечення у першому розряді будь-якого біта вхідного слова.

24. За аналогією з вправою **22**, розробити селектор для вибору одного з чотирьох восьмирозрядних слів, заданих у паралельному форматі, і передачі його на паралельну восьмирозрядну шину (задача опитування чотирьох датчиків з цифровим виходом). При розробці використати мікросхему KP1533КП2 (SN74ALS153).

25. Спроекувати селектор-мультиплексор для комутації трьох п'ятибітних слів на загальну п'ятибітну шину.

26. Використовуючи формулу (3.10), розробити схему зсуву з використанням мультиплексорів “**з 8 в 1**”.

27. Пристрій зсуву однобайтного слова  $D_0 \div D_7$  на адресних входах має код **101**. Записати кодову послідовність вихідного слова.

28. Дешифратор як комбінаційна схема формує сигнали на виході в залежності від комбінації вхідних сигналів у будь-який момент часу. Розробити варіанти схем пристроїв, які б дозволили формувати вихідні сигнали за тактовим (стробуючим) імпульсом. Привести пояснення їх роботи.

29. Використовуючи **Приклад 3.13**, розробити схему перетворювача чотирьохрозрядного двійкового коду в код “2 з 5” за допомогою дешифратора 564ИД1 і допоміжної логіки.

30. Використовуючи **Приклад 3.18**, розробити схему дешифратора шестирозрядного двійкового коду на 64 виходи.

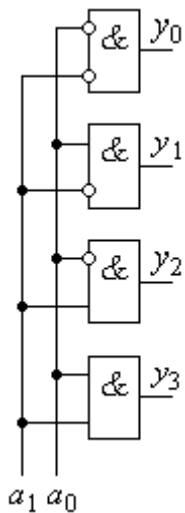


Рис. 3.66

31. На рис. 3.66 представлена схема цифрового пристрою. Пояснити роботу пристрою, побудувати часові діаграми на його виходах при повному переборі вхідних сигналів.

32. Використовуючи двовходові логічні елементи (ЛИ1) і допоміжні інвертори, побудувати схему дешифратора **3:8**.

33. Пояснити роботу пристрою, схема якого приведена на рис. 3.67. Обґрунтувати її позитивні і негативні сторони. Пояснити призначення пристрою, навести можливі області

його використання.

34. Розробити схему дешифратора “1 на 2” на логічних елементах серії КР1533. Підрахувати величину максимальної затримки при:

- а) зміні інформаційного входу;
- б) зміні адресного входу.

35. Вправу **34** виконати для КМОН ІС, порівняти результати.

36. Розробити схему дешифратора “2 на 4” на ІС серії КР1533. Підрахувати можливі максимальні інтервали затримок: а) від інформаційного входу до будь-якого з виходів; б) від адресного входу до будь-якого з виходів.

37. Вправу **36** виконати для ІС КМОН-технології, використовуючи довідкові дані.



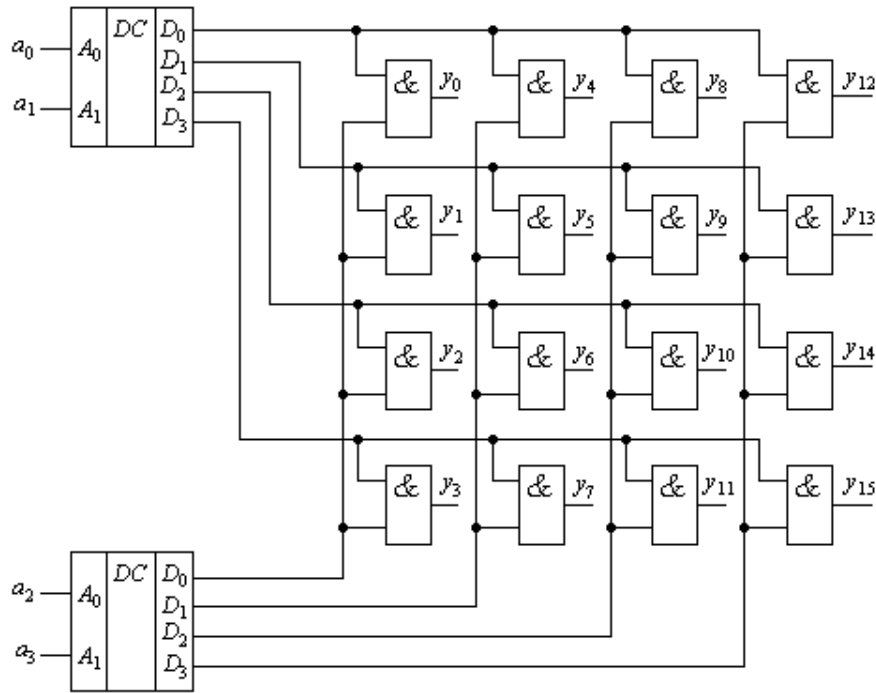


Рис. 3.67

38. Використовуючи дешифратор-демультиплексор “3 на 8” КР1533ИД7 (SN74ALS138), виконаний за ТТЛ-технологією і допоміжну логіку, розробити схему пристрою для реалізації системи логічних функцій:

$$y_1 = \bigvee_0^7 2, 4, 7; \quad y_2 = \bigvee_0^7 1, 3, 5, 6; \quad y_3 = \bigvee_0^7 2, 3, 4, 7; \quad y_4 = \bigvee_0^7 0, 4, 6.$$

39. Підібрати дешифратор-демультиплексор “3 на 8” із серій КМОН IC і розробити схему пристрою для реалізації системи логічних функцій з вправи 38.

40. Обчислити максимальні затримки від адресних входів до виходів  $y_i$  для пристроїв, що реалізовані до вправ 38 і 39. Порівняти отримані результати.

41. Використовуючи дешифратор “з 4 в 16” КР1533ИД3 (SN74LS154) і допоміжну логіку, розробити пристрій для реалізації системи логічних функцій:

$$y_1 = \bigvee_0^{15} 2, 4, 6, 14; \quad y_2 = \bigvee_0^{15} 0, 1, 2, 3, 5, 7, 11, 13; \quad y_3 = \bigvee_0^{15} 1, 3, 4, 5, 12, 14, 15.$$

42. Пояснити, як з використанням дешифраторів “3 на 8” КР1533ИД7 розробити схему пристрою для реалізації функцій з вправи 41.

43. Пояснити, що буде відображено на семисегментному індикаторі з внутрішнім дешифратором, якщо на його входи будуть подані двійкові коди від **1010** до **1111**.

44. Використовуючи схему дешифратора КР1533ИД4 (рис. 3.33) та пояснення до нього, побудувати таблиці станів для кожної з його секцій.

45. Мікросхема ТТЛ ЛП8 містить у собі чотири інвертори, що відкриваються низьким рівнем керуючого сигналу. На рис. 3.68 приведена частина схеми перетворювача однобайтного слова, заданого у паралельному форматі, у послідовний з використанням дешифратора ИД14, що містить два однотипні дешифратори  $2 \times 4$ . Таблиця станів його приведена у табл. 3.15.

Таблиця 3.15

Входи			Виходи			
$\bar{E}$	$A_0$	$A_1$	$\bar{0}$	$\bar{1}$	$\bar{2}$	$\bar{3}$
1	x	x	1	1	1	1
0	0	0	0	1	1	1
0	1	0	1	0	1	1
0	0	1	1	1	0	1
0	1	1	1	1	1	0

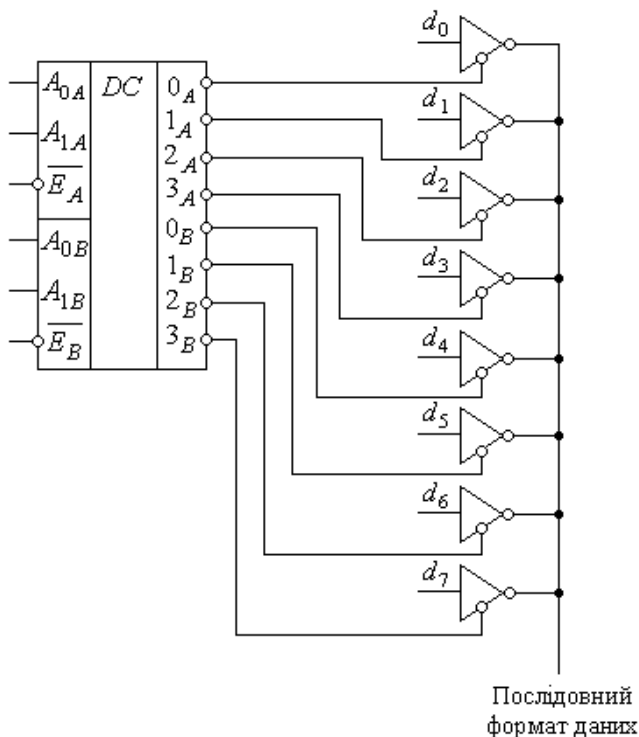


Рис. 3.68

Зробити необхідні з'єднання входів дешифратора, побудувати часові діаграми перетворювача, пояснити його роботу.

46. Як можна використовувати цифрові компаратори для контролю парності? Побудувати схеми і пояснити роботу пристроїв контролю парності на **3** і **4** входи.

47. Спроекувати схему пристрою перевірки на чотири входи  $x_3 \dots x_0$  і один вихід  $Y$ . Вихід  $Y$  має одиничне значення, якщо два і більше входів дорівнюють одиницям, і нульове, якщо або на всіх входах сигнал дорівнює нулю або лише на одному з них наявний одиничний сигнал. Використати ТТЛШ ІС серії КР1533. Спробувати мінімізувати необхідну кількість корпусів.

48. Спроекувати дешифратор  $4 \times 10$  з прямими виходами, використовуючи логічні схеми серії 1533. Підрахувати максимально можливі затримки при перемиканні входів.

49. Використовуючи дешифратори  $3 \times 8$  КР1533ИД7 (SN74ALS138) і  $2 \times 4$  К531ИД14 (SN74LS139), розробити схему дешифратора  $5 \times 32$ . Пояснити роботу пристрою. Обґрунтувати використання дозволяючих входів.

50. Логічний елемент КР1533АП9 (SN74ALS640) – буфер з трьома станами – працює на 10 входів аналогічних мікросхем. Оцінити тривалість переходу виходу мікросхеми з низького рівня у високий і у  $Z$ -стан. Необхідні припущення прийняти самостійно.

51. Розробити схему шифратора “з 10 на 4” для перетворення десяткових цифр від 0 до 9 у двійковий код. Пояснити її роботу.

52. Розробити схему шифратора “з 16 на 4” для перетворення гексадецимального коду у двійковий, використовуючи лише елементи І-НІ на 4 і 8 входів. Пояснити, які активні рівні входів і виходів розробленої схеми.

53. Розробити схему пристрою з використанням мікросхеми ИВ1 для визначення пріоритету серед 8-ми активно високих входів  $I_0 \div I_7$ , де  $I_7$  має найвищий пріоритет. Схема повинна генерувати у прямому двійковому коді  $A_0 \dots A_2$  номер входу з найвищим пріоритетом. При необхідності використати допоміжну логіку ТТЛШ або КМОН у залежності від вибраної мікросхеми.

54. Використовуючи довідкові дані на мікросхему 564КТ3, пояснити призначення і особливості роботи пристрою, схема якого приведена на рис. 3.69.

55. Пояснити роботу пристрою, схема якого приведена на рис. 3.70, при умові, що сигнали  $A$  і  $B$  подаються неодноразово.

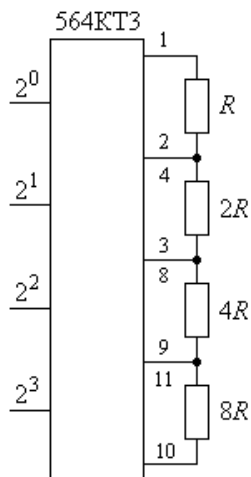


Рис. 3.69

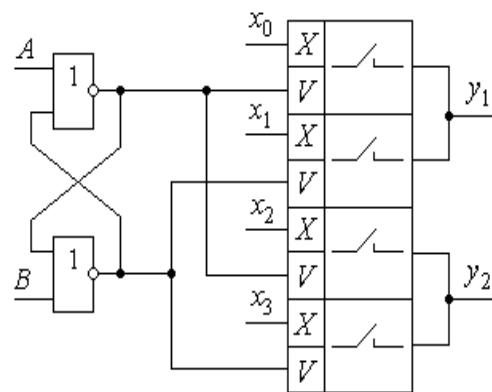


Рис. 3.70

56. На рис. 3.71, *a – б* приведені схеми пристроїв з використанням дешифратора **2:4**. Пояснити призначення кожного з пристроїв. Обґрунтувати принцип їх роботи з використанням часових діаграм.

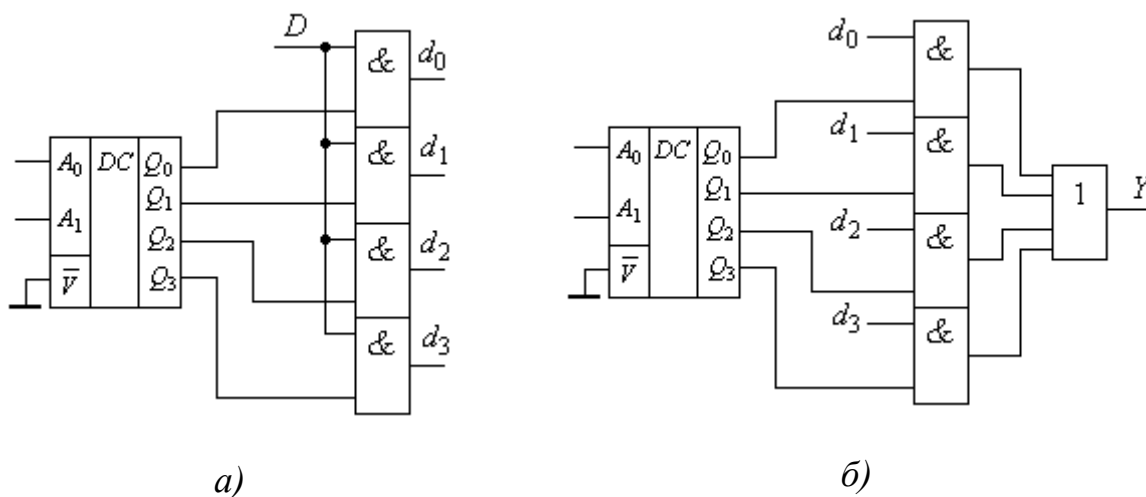


Рис. 3.71

57. Спроекувати дешифратор, алгоритм функціонування якого описується приведеною таблицею відповідності (табл. 3.16).

58. Спроекувати дешифратор, алгоритм функціонування якого описується приведеною таблицею відповідності (табл. 3.17).

59. Спроекувати дешифратор, алгоритм функціонування якого описується приведеною таблицею відповідності (табл. 3.18).

Таблиця 3.16

$\overline{CS}$	$a_2$	$a_1$	$a_0$	Виходи
1	x	x	x	Заборона
0	0	0	x	<i>A</i>
0	0	x	0	<i>B</i>
0	0	1	x	<i>C</i>
0	0	x	1	<i>D</i>
0	1	0	x	<i>E</i>
0	1	x	0	<i>F</i>
0	1	1	x	<i>G</i>
0	1	x	1	<i>H</i>

Таблиця 3.17

$N$	$S_2$	$S_1$	$S_0$	Виходи
0	0	0	0	<i>A</i>
1	0	0	1	<i>B</i>
2	0	1	0	<i>A</i>
3	0	1	1	<i>C</i>
4	1	0	0	<i>A</i>
5	1	0	1	<i>D</i>
6	1	1	0	<i>A</i>
7	1	1	1	<i>E</i>

Таблиця 3.18

$S_3$	$S_2$	$S_1$	$S_0$	Виходи
0	0	0	x	<i>A</i>
0	0	1	0	<i>B</i>
0	0	1	1	<i>A</i>
0	1	x	x	<i>C</i>
1	x	x	0	<i>A</i>
1	x	1	x	<i>B</i>
1	x	0	x	<i>C</i>
1	x	x	1	<i>A</i>

60. Розробити пристрій для виконання операції додавання десяткових однорозрядних чисел у двійково-десятковому коді.

61. Чим відрізняється виконання операції додавання в коді “з надлишком 3” для десяткових чисел з розрядністю, більшою 1?

62. Обґрунтувати особливості виконання операції віднімання в коді “з надлишком 3” для двійкового коду.

(Рекомендується на конкретних прикладах виконати операції, перетворивши і зменшуване, і від’ємник в код “з надлишком 3”. Необхідно дотримуватись правил виконання операції віднімання з використанням доповнюючого коду.)

63. Пояснити, які пристрої комбінаційної схемотехніки працюють у відповідності з алгоритмом:  $y = \overline{x_1} \oplus x_2 = x_1 \oplus \overline{x_2}$ .

64. Пояснити, який пристрій комбінаційної схемотехніки працює у відповідності з алгоритмом ( $y_i \notin Y$ ):  $Y = \bigwedge_{i=0}^3 (x_i \otimes y_i)$ .

65. Використовуючи мікросхему КР1533СП1 (SN74LS85) (чотирихрозрядний компаратор), розробити схему компаратора для порівняння двох дванадцятирозрядних двійкових чисел.

66. Пояснити, чи можливо двійковий компаратор використовувати:

а) для двох чисел, заданих двійково-десятковим кодом;

б) для двох чисел, заданих в коді “з надлишком 3”.

67. Для порівняння двох дворозрядних чисел використовується мікросхема цифрового компаратора КР1533СП1 (SN74LS85). Чи має значення,

які входи компаратора задіяні як інформаційні? Що необхідно зробити з вільними входами? Що необхідно зробити з входами переносу?

68. Розробити схему перетворювача двійкового коду в двійково-десятковий з використанням суматора для чисел від **0** до **15**.

69. Використовуючи результати вправи **68**, визначити максимальну затримку для чотирьохрозрядного суматора ИМЗ з послідовним переносом.

70. Побудувати схему і визначити максимальну затримку для однорозрядного повного суматора, виготовленого з використанням логічних елементів ТТЛ і КМОП.

71. Спроекувати схему пристрою, який приймає з шини **A** потік чотирьохбітних слів, заданих у паралельному форматі, і передає на шину **B** тільки ті, що відповідають контрольному коду. Пояснити його роботу.

72. Розробити схему пристрою, за допомогою якого можна фіксувати будь-яке з 16 заданих положень електричного двигуна.

73. Комбінаційна схема цифрового пристрою описується логічною функцією:

$$y = \bigvee_0^{15} 0, 1, 2, 4, 5, 6, 8, 9, 10, 14 .$$

Використовуючи карту Карно, розробити мінімальну протигоничну диз'юнктивну форму запису функції.

Виконати вправу **73** для функції:

$$y = \bigvee_0^{15} 0, 1, 2, 6, 9, 10, 11, 14 .$$

## Розділ 4

### ТРИГЕРИ. ТРИГЕРНІ СХЕМИ

До цього часу ми розглядали цифрові схеми, у яких мав місце односторонній зв'язок від входу до виходу.

Розглянемо тепер одну з найпростіших схем, в якій є зворотній зв'язок з виходу на вхід (рис. 4.1).

Приведена схема навіть за зовнішнім виглядом має деякі особливості:

- по-перше, ми не можемо говорити про один вихід, бо елементи об'єднані в кільце і виходів буде стільки, скільки елементів;

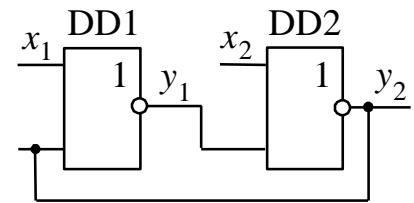


Рис. 4.1

- по-друге, задавши традиційно повний перебір входних значень змінних  $x_1$   $x_2$ , побачимо, що проаналізувати стан схеми неможливо без допоміжних умов, які необхідно задавати заздалегідь;

- по-третє, схема має позитивний зворотній зв'язок, подібно до генераторних схем, наслідком чого є те, що зміна станів окремих елементів відбувається майже одночасно для всіх елементів.

У цифровій схемотехніці такі схеми називаються *тригерними схемами*, або частіше просто *тригерами*.

У загальному вигляді тригерна схема має вигляд, що відповідає рис. 4.2.

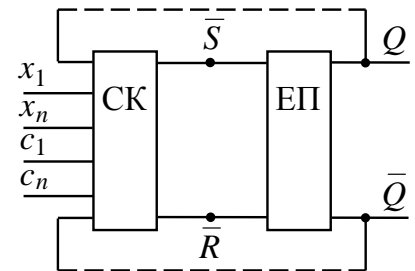


Рис. 4.2

У цифрових пристроях використовується велика кількість різноманітних тригерів, але всі вони мають у своєму складі *схему керування (СК)* та *елемент пам'яті (ЕП)*, який має два виходи – прямий  $Q$  та інверсний  $\bar{Q}$ , котрі можуть приймати значення **1** або **0**. Керування елементом пам'яті відбувається за допомогою входів  $S$  (*Set* – установлення стану  $Q = 1$ ,  $\bar{Q} = 0$ ) та  $R$

(*Reset* – повернення до початкового стану  $Q = 0$ ,  $\bar{Q} = 1$ ). Логічні значення сигналів залежать від значень вхідних логічних змінних  $x_1 \dots x_n$ , сигналів синхронізації  $c_1 \dots c_n$ , станів виходів елемента пам'яті  $Q$  та  $\bar{Q}$  і логіки роботи схеми керування. У залежності від призначення схеми змінюється її логічна функція, кількість входів  $x_n$ , кількість та характер входів синхронізації.

У зарубіжній інженерній практиці всі тригерні схеми розділяються на дві групи.

Перша з них – *flip-flop* – характеризується тим, що вибірка вхідних сигналів і відповідна зміна виходів визначається в моменти дії тактових часових сигналів (синхронні тригери).

Особливість другої групи схем – *latch* – полягає в тому, що вони змінюють свій стан при зміні вхідних сигналів незалежно від наявності чи відсутності часових тактових сигналів.

Розглянемо тригерні схеми на конкретних прикладах.

#### 4.1. RS-тригери

Схема RS-тригера, зібраного на логічних елементах **2АБО-НІ**, приведена на рис. 4.3, а. Вона відрізняється від схеми, приведеної на рис. 4.1, тільки позначеннями входів та виходів.

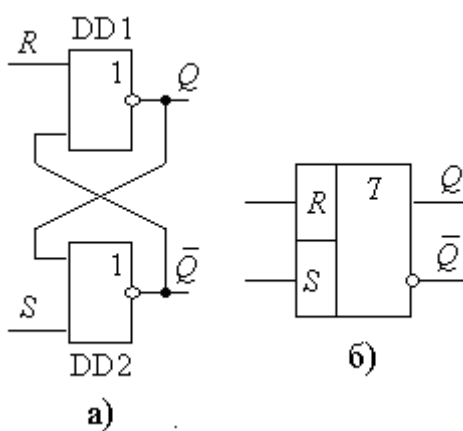


Рис. 4.3

Завдяки симетричному вигляду, вона часто називається *симетричним RS-тригером* (але зовнішня симетрія не означає симетрію електричних режимів роботи логічних елементів DD1, DD2). На рис. 4.3, б приведено його умовне позначення.

Стан тригера часто ототожнюється з сигналом на прямому виході  $Q$ .



Особливість тригерних схем, на відміну від комбінаційних, полягає в тому, що будь-який із станів є стійким при відсутності вхідних сигналів.

Припустимо, що  $R = S = 0$ , а  $\bar{Q} = 1$ . Тоді на виході елемента DD1 зберігатиметься значення  $Q = 0$ , яке забезпечує значення виходу  $\bar{Q} = 1$  елемента DD2, тобто підтверджує нульовий стан тригера.

Припустимо, що у цьому стані на входи тригера подано комбінацію потенційних сигналів  $R = 0$ ,  $S = 1$ . Тоді вихід елемента DD2 прийме значення  $\bar{Q} = 0$ , а вихід елемента DD1 прийме значення  $Q = 1$ . Вказаною комбінацією вхідних сигналів ми запишемо сигнал  $S = 1$  у тригер як один біт інформації. Змінити стан тригера на попередній можливо протилежною комбінацією вхідних сигналів  $R = 1$ ,  $S = 0$ . Якщо ми повторимо подачу цієї комбінації вхідних сигналів при  $Q = 0$ ,  $\bar{Q} = 1$ , то стан тригера не зміниться. Таким чином, стан виходів тригера в момент часу, наступний за моментом подачі вхідних сигналів, залежить не тільки від комбінації вхідних сигналів, але й від попереднього стану його виходів. Тому для аналізу схеми необхідне розподілення станів входів та виходів тригера у часі. Для цього попередні стани позначаються індексом  $n$ :  $S_n$ ,  $R_n$ ,  $Q_n$ . Наступний стан тригера, в який той переходить у результаті дії комбінації вхідних сигналів з урахуванням значень виходів у  $n$ -й момент часу, позначається як  $Q_{n+1}$ ,  $\bar{Q}_{n+1}$ . Завдяки такому розподіленню станів та вхідних сигналів у часі з'являється можливість використовувати логічні функції для запису алгоритму роботи тригерів.

Повна таблиця станів  $RS$ -тригера (табл. 4.1) описує особливості його роботи. Звернемо увагу на те, що при  $S_n = R_n = 1$ , незалежно від  $Q_n$ , стан виходів тригера є невизначеним.

Дійсно, якщо подати на обидва входи таку комбінацію сигналів, то на обох виходах з'являться

Таблиця 4.1

$R_n$	$S_n$	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	x
1	1	1	x

логічні нулі ( $Q_{n+1} = \bar{Q}_{n+1} = 0$ ). Але якщо ці сигнали одночасно зняти, задавши  $Q_{n+1} = R_{n+1} = 0$ , то стан виходів буде невизначеним. Це пов'язано з тим, що після зняття сигналів у кожному з логічних елементів буде протікати перехідний процес зміни станів, і результат його залежатиме від швидкодії елементів. Остання є величиною невизначеною. Тому розглянута комбінація вхідних сигналів називається *невизначеною комбінацією*, а перехід від такої комбінації до нульової – *забороненим переходом*. Фактично це означає, що при проектуванні цифрових пристроїв необхідно приймати міри для виключення подібних ситуацій.

Розглянемо більш детально роботу *RS*-тригера з точки зору забезпечення

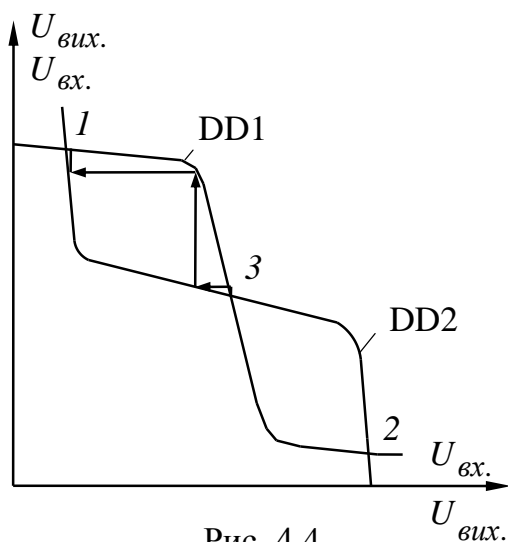


Рис. 4.4

якості та надійності переходу з одного стану в інший, адже він широко використовується в якості елемента пам'яті більш складних тригерних схем. Подібні схеми часто називають *бістабільними*. Така назва витікає з того, що реально вона може знаходитись лише в одному з двох вказаних вище станів. Положення їх визначається точками перетину характеристик “вхід-вихід” логічних ключів, відповідно до рис. 4.4, на

якому зображені характеристики “вхід-вихід” елементів DD1 і DD2.

Точки перетину характеристик 1 і 2 є *точками стабільності*. Але на рис. 4.4 існує ще одна точка перетину – точка 3, яка називається *метастабільною*, оскільки в ній контурний коефіцієнт підсилення буде рівним 1. Положення точки 3 може бути обчислено і аналітичним шляхом, якщо існує аналітична залежність  $U_{\text{ВИХ}} = f(U_{\text{ВХ}})$ , але простіше це зробити графічно, як зображено на рис. 4.4.

Особливість метастабільної точки полягає в тому, що поява незначного шуму миттєво призведе до переходу в одну з стабільних точок, як показано стрілками на рисунку. Дійсно, поява незначного шуму на виході DD1 приведе до аналогічного зростання напруги на вході DD2 з наступним переходом у стабільну точку 1. Наявність метастабільної точки і необхідність її врахування в роботі цифрових схем обумовлено тим, що при незначній тривалості або амплітуді вхідних сигналів  $S$  і  $R$  схема може перейти в метастабільний стан.

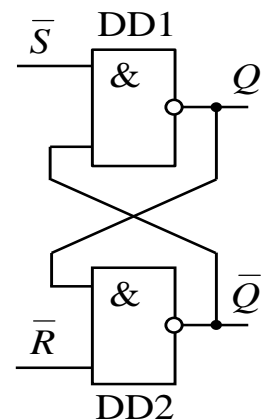


Рис. 4.5

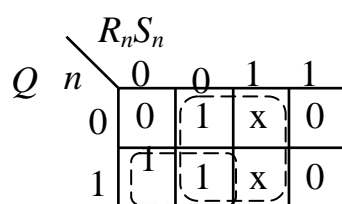


Рис. 4.6

Таблиця 4.2

$R_n$	$S_n$	$Q_{n+1}$
0	0	$Q_n$
0	1	1
1	0	0
1	1	x

$$Q_{n+1} = S_n + Q_n \cdot \overline{R_n}; \quad (4.1)$$

$$\overline{Q}_{n+1} = R_n + \overline{Q}_n \cdot \overline{S}_n.$$

$$Q_{n+1} = \overline{\overline{S}_n \cdot (Q_n \cdot R_n)}; \quad (4.2)$$

$$\overline{Q}_{n+1} = \overline{R_n \cdot (\overline{Q}_n \cdot \overline{S}_n)}.$$

Представивши табл. 4.1 у формі карти Карно (рис. 4.6), помічаємо наступну особливість логічної функції: при значеннях  $S_n = R_n = 0$  маємо  $Q_{n+1} = Q_n$ , а для решти комбінацій входів значення виходу  $Q_{n+1}$  не залежить від значення  $Q_n$ . Це дає можливість мінімізувати табл. 4.1 (див. табл. 4.2) та одержати характеристичне рівняння тригера:

Перетворюючи рівняння (4.1) в базисі логічних функцій **I-НІ**, одержуємо:

Цим рівнянням відповідає схема рис. 4.5. Перше з них виконане на DD1, а друге на DD2. Цей тригер працює в інверсних кодах.

**Приклад 4.1.** Побудувати таблицю станів для асинхронного  $RS$ -тригера, виготовленого на логічних елементах **2І-НІ**, а також часові діаграми сигналів на входах і виходах при різних їх співвідношеннях.

*Розв'язання.* Таблиця станів будується або на основі табл. 4.2, або на основі логічних рівнянь (4.2) і має вигляд табл. 4.3. Високі рівні вхідних сигналів не змінюють стану тригера, а низькі рівні одночасно на двох входах є забороненими для тригера, виготовленого на логічних елементах **2І-НІ**. Установка (запис інформації) забезпечується низьким рівнем по входу  $\bar{S}$ , а обнуління – відповідно, низьким рівнем по входу  $\bar{R}$ . Часові діаграми, що пояснюють роботу тригера, приведені на рис. 4.7.

Таблиця 4.3

$\bar{S}_n$	$\bar{R}_n$	$Q_{n+1}$
0	0	X
0	1	1
1	0	0
1	1	$Q_n$

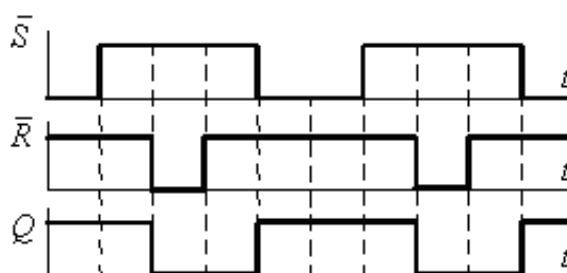


Рис. 4.7

Зміна станів тригерних схем при подачі різних послідовностей вхідних сигналів зображується за допомогою графа переходів (рис. 4.8).

Розглянемо детальніше особливості побудови графа переходів тригера. Виходячи з таблиці станів, можемо розглядати тригер як пристрій з одним виходом  $Q$  ( $Q_{n+1}$ ), оскільки другий вихід  $\bar{Q}$  ( $\bar{Q}_{n+1}$ ) є лише інверсією першого.

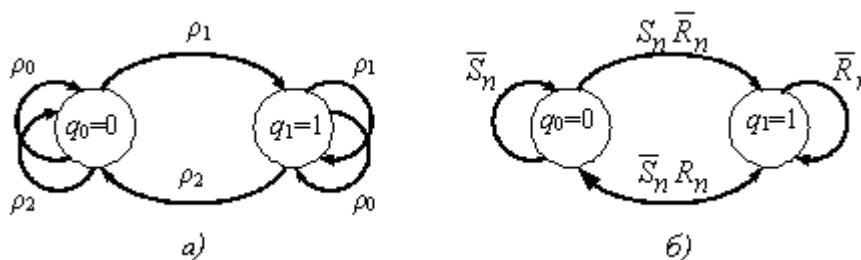


Рис.4.8

Вихід  $Q$  може приймати два значення. Позначимо їх  $q_0 = 0$  і  $q_1 = 1$  як дві вершини графа. Вектор впливаючих сигналів позначимо буквою  $\rho$ . Він може приймати, у відповідності до табл. 4.2, значення  $\rho_0 = \bar{S}_n \bar{R}_n$ ,  $\rho_1 = S_n \bar{R}_n$ ,

$\rho_2 = \overline{S}_n R_n$ , вектор  $\rho_0$  не може змінювати стани тригера. Таку особливість на граф-схемі (рис. 4.8, а) зобразимо у вигляді дуги, що виходить з вершин і замикається на них. Вектор  $\rho_1$  переводить тригер у стан  $q_1 = 1$  і на граф-схемі зображується у вигляді дуги, що виходить з вершини  $q_0$  і закінчується в  $q_1$ . Якщо тригер знаходиться в стані  $q_1$ , то впливаючий сигнал  $\rho_1$  не змінить стану тригера. На граф-схемі це дуга, що замикається на стані  $q_1$ . Аналогічно, впливаючий сигнал  $\rho_2$  переводить тригер зі стану  $q_1$  в  $q_0$ , а в стані  $q_0$  не впливає на нього. Оскільки перехід з  $q_1$  в  $q_1$  забезпечується двома сигналами з однаковим результатом, то поєднаємо їх як:

$$\rho_{01} = \rho_1 + \rho_0 = S_n \overline{R}_n + \overline{S}_n \overline{R}_n = \overline{R}_n. \quad (4.3)$$

Аналогічно маємо:

$$\rho_{02} = \rho_0 + \rho_2 = \overline{S}_n \overline{R}_n + \overline{S}_n R_n = \overline{S}_n. \quad (4.4)$$

В результаті граф-схема переходів RS-тригера прийме вигляд рис. 4.8, б.

Графи переходів відповідають часовим діаграмам, що ілюструють роботу тригера у часі. Крім того, часові діаграми дають більш детальну характеристику перехідним процесам у схемі.

На рис. 4.9 наведені часові діаграми роботи тригера, схему якого зображено на рис. 4.3, а. Елементи тригера перемикаються послідовно. Запуск тригера відбувається за фронтом вхідного сигналу  $S$  у момент часу, коли його значення досягне порогового рівня спрацювання логічного елемента DD2. Вихідний стан  $\overline{Q} = 1$  логічного елемента DD2 змінюється, і в момент часу, коли потенційний рівень спаду сигналу  $\overline{Q}$  зменшиться до рівня порогової напруги елемента DD1, останній починає перемикатись. На рис. 4.9 послідовність перемикання показана стрілками.

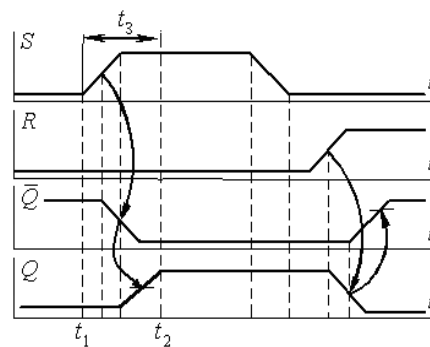


Рис. 4.9

Інтервал часу перемикання визначається з моменту  $t_1$  початку дії сигналу  $S$  до моменту  $t_2$  – завершення перемикання логічного елемента DD1. Цей інтервал часу  $t_3$ , що зветься *часом затримки на перемикання*, дає можливість оцінити мінімальну тривалість вхідних сигналів, при якій гарантовано будуть змінюватися стани тригера.

З опису процесу перемикання витікає, що  $t_3$  тригера визначається сумою часових затримок двох базових логічних елементів, на яких він виготовлений.

У довідковій літературі для кожного тригера приводиться таблиця станів, аналіз якої дозволяє визначити всі режими роботи пристрою.

Таблиця 4.4

$S_n$	$R_n$	$V$	$Q_{n+1}$
0	0	1	$Q_n$
0	1	1	0
1	x	1	1
x	x	0	Z

**Приклад 4.2.** Пояснити особливості роботи *RS*-тригера К564ТР2 (аналоги провідних західних фірм-виробників – 4043BDC, CD4043AD, МС14043BAL, MN4043B, MSM4043B, SCB4043B, ТС4043BP), таблиця станів якого додається (табл. 4.4).

*Розв'язання.* У мікросхемі К564ТР2 дозволяючий вхід  $V$  повинен мати високий рівень потенціалу для забезпечення робочого режиму схеми. При  $V=0$  вихід  $Q$  переходить у високоомний стан. Тригер виготовлений з використанням логічних елементів **2АБО-НІ** і має лише один прямий вихід, оскільки при  $S_n = R_n = V = 1$  значення  $Q_{n+1} = 1$  перебуває у незабороненому стані. При наявності двох виходів сигнали на їх виходах були б однаковими, що оцінювалося б як невизначеність.

*RS*-тригери у багатьох випадках використовуються як самостійні пристрої в тих ситуаціях, коли одним сигналом необхідно встановити якусь умову, а іншим – її зняти. Таке їх використання передбачається у контролерах і мікроконтролерах у складі регістрів ознак. Здебільшого *RS*-тригери використовуються у складі більш складних схем тригерів, модулів пам'яті. Вони знаходять широке використання в пристроях електронної автоматики.

**Приклад 4.3.** На рис. 4.10 приводиться схема пристрою з використанням *RS*-тригера, що призначена для скорочення тривалості вхідного імпульсу. Пояснити роботу пристрою.

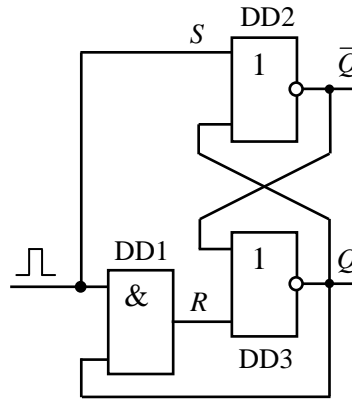


Рис. 4.10

*Розв'язання.* При подачі імпульсу високого рівня на вхід схеми він одночасно подається на вхід  $S$  тригера і встановлює його в стан, при якому забезпечується  $Q = 1$ . Як показано на рис. 4.9, цей процес перемикання триває деякий час, обумовлений часом послідовного перемикання логічних елементів DD2 і DD3. На виході DD1 в цей час має місце низький рівень сигналу. Після перемикання тригера на другому вході DD1 з'являється високий рівень сигналу, який з затримкою на його перемикання встановлюється на його виході і приводить до зворотної зміни стану тригера. Читачам пропонується самостійно побудувати часові діаграми.

## 4.2. Загальна характеристика тригерних схем

Особливість схемотехніки тригерних схем, порівняно з комбінаційною схемотехнікою, полягає в тому, що їх виходи з'єднані зі входами. Ця ситуація часто вимагає застосування допоміжних заходів для протидії електричним завадам  $U_3$ , що виникають на виходах різноманітних електронних пристроїв. Виникнувши на лініях зв'язку і на відповідних входах тригера, завада може викликати зміну його стану. Таким чином, тригер може перетворювати випадкові завади у постійно діючий сигнал. Для виключення подібної ситуації виходи тригера підключаються до

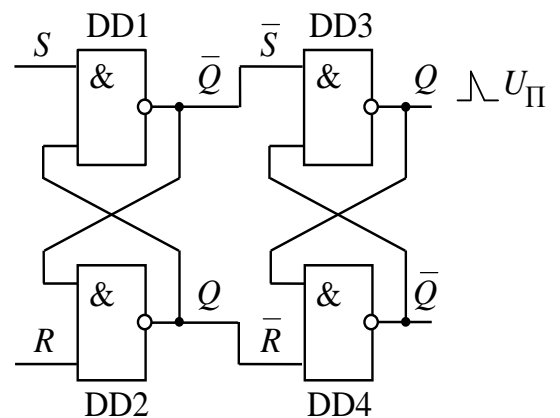


Рис. 4.11

навантаження або лінії зв'язку через буферні елементи, а для підвищення швидкодії цих елементів їх з'єднують також за схемою тригера (рис. 4.11).

У такій схемі на входах буферних елементів DD3, DD4 постійно діє один з потенційних сигналів з виходів DD1, DD2. Тому, якщо навіть імпульсна завада  $U_3$  і призведе до тимчасової зміни станів DD3, DD4, то після її зникнення вхідний сигнал поверне мікросхеми DD3, DD4 у початковий стан.

Незважаючи на простоту, *RS*-тригери в чистому вигляді не використовуються для збереження даних у системах обробки інформації, адже для запису інформації необхідно мати дві послідовності сигналів: спочатку подавати *R*-сигнали для переведення тригера до нульового стану, а потім інформаційний та синхронізуючий – для запису даних.

Вхідні сигнали в залежності від призначення поділяють на: *інформаційні, дозволяючі, командні або синхронізуючі*.

Відповідно до типів сигналів, поділяють і входи складних схем тригерів, які позначають літерами абетки згідно з табл. 4.6.

Через те, що функціональні властивості тригера залежать від вхідної логіки, назви інформаційних входів переносяться на всю схему. Тому, крім *RS*-тригерів, широке використання знаходять *D*-, *DV*-, *JK*-, *T*-, *TV*-тригери, особливості яких розглядаються нижче.

Таблиця 4.6

<i>Позначення входу</i>	<i>Призначення</i>
<i>S</i>	Вхід установки тригерів у стан $Q = 1$
<i>R</i>	Вхід установки тригерів у стан $Q = 0$
<i>J</i>	Вхід установки <i>JK</i> -тригерів у стан $Q = 1$
<i>K</i>	Вхід установки <i>JK</i> -тригерів у стан $Q = 0$
<i>D</i>	Вхід завантаження інформації у <i>D</i> -тригер
<i>T</i>	Тактовий вхід <i>T</i> -тригерів
<i>V</i>	Підготовчий вхід дозволу прийому інформації
<i>C</i>	Вхід синхронізації. Виконавчий вхід прийому інформації



Незалежно від функціональних властивостей, всі тригери за способом введення інформації поділяють на дві групи – *асинхронні* та *синхронні*.

*Асинхронні тригери*, подібно до розглянутого вище *RS*-тригера, змінюють свій стан безпосередньо після зміни інформаційних сигналів.

Для спрацювання *синхронних тригерів* наявності інформаційних сигналів недостатньо. Допоміжно необхідна ще наявність синхроімпульсу (тактового імпульсу) на спеціальному тактовому *C*-вході (рис. 4.12, *a*). Тактові імпульси створюються за допомогою тактових генераторів і використовуються всім цифровим пристроєм. У синхронних схемах інформаційні сигнали (тобто дані, що записуються в тригер) готуються і подаються на відповідні входи раніше, ніж синхросигнали (рис. 4.12, *б*). При цьому для надійного запису інформації необхідними є відповідні часові співвідношення.

У довідковій літературі на будь-який тригер даються два параметри:

- час затримки розповсюдження сигналу на вмикання  $t_{3\ 01}$  ;
- час затримки розповсюдження сигналу на вимикання  $t_{3\ 10}$  .

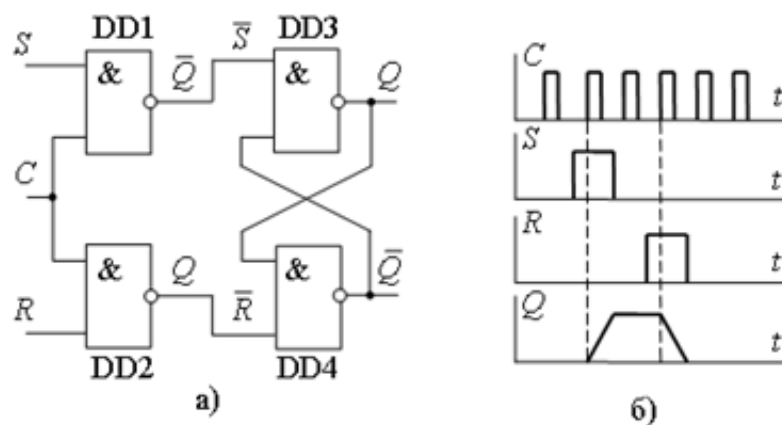


Рис. 4.12

Інтервали  $t_{3\ 01}$  і  $t_{3\ 10}$  – це інтервали часу, що визначають мінімальну тривалість вхідних імпульсів, які повинні подаватися на тригер.

При наявності окремих інформаційних входів і входів синхронізації задаються часові параметри окремо для асинхронних входів установки  $\bar{S}$  і

скидання  $\bar{R}$  і окремо по входу синхронізації  $C$ . Час затримки по входах  $J, K, D$  не задається, оскільки вважається, що інформація на них повинна бути попередньо підготовлена.

Для синхронних тригерів часові інтервали характеризуються ще двома параметрами, які є характерними не тільки для тригерних схем.

Перший з них – *час підготовки*  $t_{\Pi}$  (в англійській літературі – *Setup time*, позначається як  $t_{SU}$ ). Це мінімальний інтервал часу, протягом якого інформаційний сигнал на вході тригера повинен залишатися незмінним до подачі синхросигналу. За цей час всі перехідні процеси устанавлення інформаційного сигналу повинні завершитися.

Другий часовий параметр – *час затримки (підтримки)*  $t_3$  (в англійській літературі – *Hold time*,  $t_H$ ). Це мінімальний інтервал часу з моменту подачі синхросигналу, протягом якого інформаційний сигнал повинен залишатися незмінним. Цей інтервал часу тісно пов'язаний з вищезгаданими інтервалами часу  $t_{3\ 01}$  і  $t_{3\ 10}$ , оскільки необхідно, щоб виконувалися наступні нерівності:

$$t_3 > t_{3\ 01}; \quad t_3 > t_{3\ 10}. \quad (4.5)$$

В окремих випадках в синхронних цифрових схемах використовуються дві зміщені в часі послідовності синхроімпульсів.

На рис. 4.12, *a* наведена одна з найпростіших схем синхронного  $RS$ -тригера, виконаного на логічних елементах **2І-НІ** (наприклад, з використанням однієї мікросхеми 4(**2І-НІ**) К1533ЛА3 (відповідні аналоги провідних західних фірм-виробників – 7400РС, СDB400Е, D100D, МН7400, SN7400N)). Вузол синхронізації зібраний на елементах DD1, DD2. При відсутності синхронізуючого сигналу ( $C = 0$ ) тригер відключається від  $R$ -,  $S$ - входів і перебуває у режимі збереження інформації. При  $C = 1$  схема стає звичайним асинхронним  $RS$ -тригером, і зміна інформації на  $R$ -,  $S$ - входах викликатиме зміну його стану.

У [Пух.] та ряді інших джерел такі тригери називаються  $R$ - $S$ - $L$  тригерами, де  $L = C$ . Цей вхід використовується для забезпечення завантаження тригера ( $L$  –  $Load$ ). Алгоритм його роботи:

$$Q_{n+1} = S_n L_n + Q_n \bar{R}_n + Q_n \bar{L}_n; \quad S_n \cdot R_n \cdot L_n = 0.$$

Друга умова визначає, що  $S = R = L = 1$  подавати заборонено.

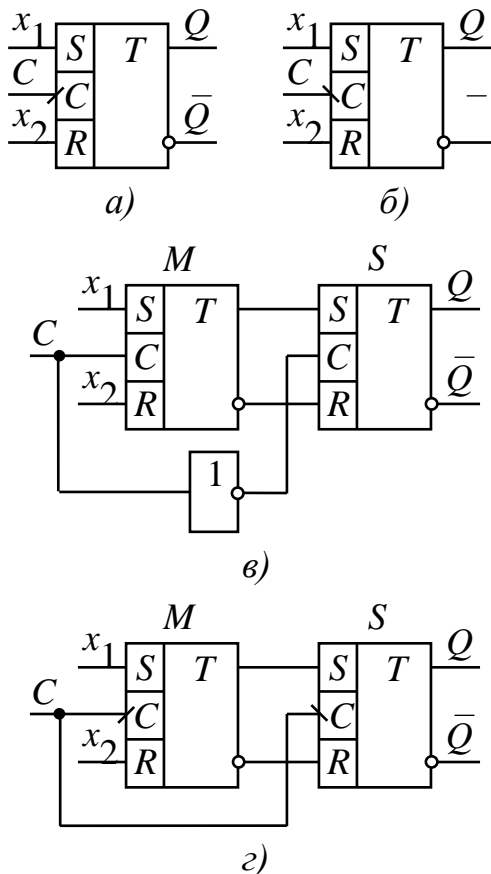
$RS$ -тригер є здебільшого лише елементом пам'яті для різних типів тригерних систем. Головна ж роль у формуванні властивостей системи належить схемі керування, яка перетворює вхідні сигнали  $x_0 \dots x_n$ ,  $C_0 \dots C_n$  у сигнали керування елементом пам'яті.

Недоліком асинхронних тригерів, який обмежує їх використання у швидкодіючих пристроях, є незахищеність перед небезпечними “змаганнями” (“гонками”) сигналів, сутність яких полягає в тому, що сигнали, які подаються на інформаційні входи тригера, проходять по різних колах, через різну кількість логічних елементів. У зв'язку з цим можливі часові зміщення сигналів, величини яких можуть змінюватись у широких межах. Це може призвести до запису в тригери хибної інформації. Синхронізація дозволяє цей недолік ліквідувати. До того ж, синхронізація дає можливість підвищити завадостійкість тригера, адже значно зменшується інтервал часу доступу до його інформаційного входу (входів).

Слід зазначити, що тактовий сигнал є керуючим імпульсом, адже вважається, що до його приходу зміна сигналів на інформаційних входах повинна бути завершена.

Синхронні тригери поділяють на схеми зі *статичним* та з *динамічним керуванням*. Характерним для статичних тригерів є те, що при активному стані тактового входу (наявність одиничного рівня потенціалу) вони поведуть себе як асинхронні. Ця властивість у багатьох випадках є суттєвим недоліком синхронних тригерів зі статичним тактовим входом, адже може призводити до порушень у роботі при наявності завад.

Цей недолік усунений у *тригерах з динамічним тактовим входом*, а також у *тригерах з двоступінчатим керуванням*. Тригери з динамічним керуванням,



відповідно до схеми виконання, реагують на перепад напруги від нуля до одиниці (за фронтом) або від одиниці до нуля (за спадом (зрізом) імпульсу). Останні, які зветься *тригерами з внутрішньою затримкою*, дуже широко застосовуються в системах обробки інформації, адже дозволяють за фронтом тактових імпульсів зчитувати інформацію, а за зрізом – записувати.

В умовних позначеннях динамічних тригерів використовуються різні способи. Далі користуватимемося позначеннями, наведеними на рис. 4.13: *а* – для тригерів, що спрацьовують за фронтом; *б* – за зрізом синхроімпульсу.

Двоступінчаті тригери мають два елементи пам'яті, з'єднані послідовно.

Запис інформації в них виконується послідовно, у неспівпадаючі моменти часу. Така послідовність тригерів (рис. 4.13, *в* – *г*) називається *MS-структурою (Master-Slave)* або просто *MS-тригерами*. На умовних позначеннях *MS-тригери* позначаються двома літерами *TT*.

Функціональні властивості схеми задаються першим тригером, а другий у більшості – звичайний синхронний *RS-тригер*. Двоступінчатий тригер може керуватись не тільки двома, а й одним синхроімпульсом.

При статичному керуванні вхід *C* тригера *S* з'єднується зі входом *C* тригера *M* через інвертор (рис. 4.13, *в*). При  $C_1 = 1$  виконується запис

інформації в перший  $M$ -тригер, а другий –  $S$ -тригер – блокований інверсним синхросигналом. При  $C_1 = 0$  відкриваються входи другого тригера і інформація з виходів першого перезаписується до нього.

При динамічному керуванні запис у  $M$ -тригер здійснюється за фронтом синхроімпульсу, а перезапис у  $S$ -тригер – за зрізом (рис. 4.13, з).

У англійській літературі двоступінчаті тригери з динамічним керуванням називають *flip-flop*, а зі статичним – *latch*. Якщо тип *latch*-тригера не оговорюється, то під цим визначенням розуміють  $D$ -тригер, мова про який піде нижче.

### 4.3. $D$ -тригери

Функціональна особливість тригерів цього типу полягає в тому, що сигнал на виході  $Q$  в  $(n + 1)$ -ому такті повторює значення сигналу на вході  $D$  в  $n$ -му такті.

Роботу  $D$ -тригера пояснює таблиця його станів (табл. 4.7), з якої витікає, що при  $C_n = 0$  значення  $Q_{n+1} = Q_n$ , а при  $C_n = 1$   $Q_{n+1} = D_n$ .

Зобразивши логічну функцію  $Q_{n+1} = f(Q_n, C_n, D_n)$  у вигляді карти Карно (рис. 4.14), після мінімізації знаходимо:

$$Q_{n+1} = C_n D_n + \overline{C_n} Q_n \quad (4.6)$$

Таблиця 4.7

$C_n$	$D_n$	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

$C_n \backslash D_n$	00	01	11	10
0	0	0	1	0
1	1	1	1	0

Рис. 4.14

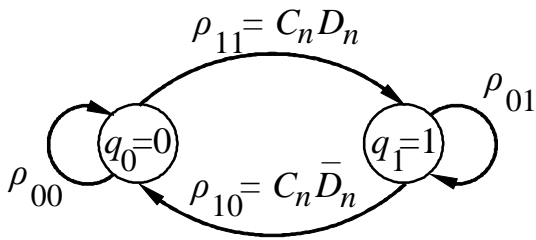


Рис. 4.15

Враховуючи той факт, що друга складова рівняння (4.6) характеризує лише режим зберігання інформації, закон функціонування *D*-тригера виразиться формулою:

$$Q_{n+1} = C_n D_n. \quad (4.7)$$

З (4.7) витікає, що при  $C_n = 1$  сигнал  $Q_{n+1} = D_n$  і будь-яка зміна вхідного сигналу одразу з'явиться на виході. Тому ці тригери часто називають “прозорими”.

Виходячи з умови ліквідації гонок, з карти Карно (рис. 4.14) отримуємо:

$$Q_{n+1} = C_n D_n + \overline{C_n} Q_n + D_n Q_n.$$

Якщо цю формулу перетворити під базис **I-НІ**, матимемо:

$$Q_{n+1} = \overline{\overline{C_n D_n} \cdot \overline{C_n} Q_n \cdot \overline{D_n} Q_n},$$

яка є законом функціонування цифрового пристрою, відомого як *D-тригер Ерла* [Пух.]. Схема, побудована на основі отриманої формули, відрізняється від інших найбільшою реальною швидкодією і використовується в різноманітних пристроях обчислювальної техніки.

Граф переходів у відповідності до табл. 4.7 зображений на рис. 4.15.

Перехід зі стану  $q_0 = 0$  в стан  $q_1 = 1$  відбувається лише при  $\rho_{11} = C_n D_n$ , а зворотній – тільки при  $\rho_{10} = C_n \overline{D_n}$ .

При  $q_0 = 0$  стан тригера не змінюється при:

$$\rho_{00} = \overline{C_n} \overline{D_n} + C_n \overline{D_n} + \overline{C_n} D_n = \overline{D_n} + \overline{C_n}. \quad (4.8)$$

При  $q_1 = 1$  стан тригера залишається незмінним за умови:

$$\rho_{01} = \overline{C_n} \overline{D_n} + C_n D_n + \overline{C_n} D_n = \overline{C_n} + D_n. \quad (4.9)$$

*D*-тригер може бути створений на базі синхронного *RS*-тригера, якщо на його інформаційні входи одночасно подати взаємно інверсні сигнали  $D$  і  $\overline{D}$ .

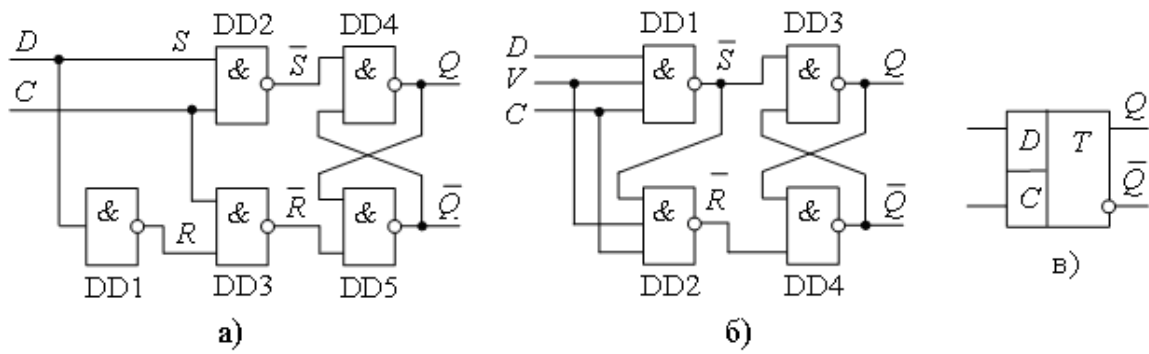


Рис.4.16

На рис. 4.16 приведені два варіанти  $D$ -тригера ( $a$ ,  $б$ ) та його умовне позначення ( $в$ ).

Розглянемо більш детально роботу кожного з тригерів. Спочатку звернемось до тригера, схема якого приведена на рис. 4.16,  $a$ . При  $C = 1$  і  $D = 0$  на входах логічного елемента DD3 будуть сигнали високого рівня, що приведе до появи на вході  $\bar{R}$  внутрішнього  $RS$ -тригера (ЛЕ DD4, DD5) сигналу низького рівня  $\bar{R} = 0$ , а на виході значення  $Q = 0$  і  $\bar{Q} = 1$ .

При появі сигналу високого рівня лише на вході  $D$  буде забезпечуватись значення  $\bar{S} = 1$  і  $\bar{R} = 1$ , при яких попередня інформація внутрішнього  $RS$ -тригера залишається незмінною. Такий режим відображається на часових діаграмах, приведених на рис. 4.17,  $a$ , інтервалом  $t_1 - t_2$ .

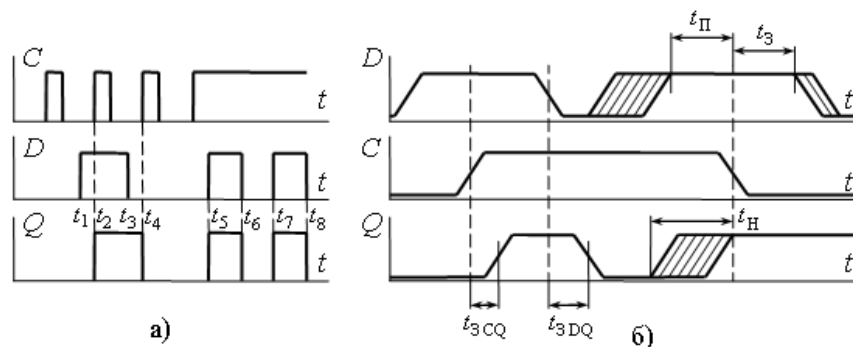


Рис. 4.17

При одночасній дії синхросигналу і сигналу високого рівня на вході  $D$  (момент часу  $t_2$ ) на виході DD2 встановиться рівень  $\bar{S} = 0$ , що приведе до зміни стану внутрішнього тригера, при якому  $Q = 1$  і  $\bar{Q} = 0$ . Такий стан

утримуватиметься до моменту, поки не відбудеться зміна рівня сигналу  $\bar{R}$  з “1” до “0” при наявності сигналу  $\bar{S}=1$ , що призведе до повернення тригера в початковий стан. Оскільки значення  $\bar{R} = 0$  має місце лише при  $C = 1$  та  $D = 0$ , то перехід тригера в початковий стан матиме місце при появі наступного тактового імпульсу (момент  $t_4$ ). У тригері, схема якого зображена на рис. 4.16, б, така ж функція забезпечується тим, що один з входів DD2 приєднаний до виходу DD1. При значенні дозволяючого входу  $V = 1$  тригер, приведений на рис. 4.16, б, працює аналогічно вище описаному.

При практичному використанні  $D$ -тригерів у них фіксується значення сигналу  $D$  при наявності відповідного рівня синхросигналу. Тому в літературі такі тригери часто називають *фіксаторами* (рос. “защёлка”) (*Latch*).

У загальному випадку логічна функція, що описує роботу  $D$ -тригера, має вигляд:

$$Q_{n+1} = V (\bar{C}_n Q_n + C_n D_n) + \bar{V} Q_n . \quad (4.10)$$

В залежності від співвідношення часових параметрів  $D$  та  $C$  сигналів робота  $D$ -тригера характеризується двома режимами. В першому випадку тригер затримує проходження сигналу, що поступає на  $D$ -вхід, на один такт періоду синхросигналу (*Delay* – затримка). Це добре відображають часові діаграми, приведені на рис. 4.17, а (*інтервал  $t_1 \dots t_4$* ).

Друге призначення  $D$ -тригера – зберігати дані (*Datas*), що надходять до  $D$ -входу. Синхросигнали в цьому випадку відіграють роль команди “Запис у тригер”. При одиничному значенні синхросигналу будь-які зміни (*моменти  $t_5, t_6, t_7, t_8$* ) на вході  $D$  приведуть до відповідних змін на виході  $Q$ . Тригер виконуватиме функцію повторювача сигналів присутніх на його вході.

**Приклад 4.4.** Використовуючи асинхронні  $D$ -тригери та допоміжну логіку, розробити схему пристрою для запису, тимчасового зберігання та зчитування напівбайтового слова.

*Розв’язання.* Для побудови такого пристрою використаємо мікросхему ТМ5, що містить в собі 4  $D$ -тригери, кожен з яких має  $D$  – вхід,  $Q$  – вихід та  $E$  – вхід дозволу. Схема кожного з тригерів близька до схеми з зовнішнім інвертором, приведеної на рис. 4.16, а,



в якій вхід  $C$  виконує функцію дозволяючого входу. Відповідно до таблиці станів тригера (табл. 4.8), для запису інформації необхідно, щоб  $E = 1$ .

Схема пристрою матиме вигляд, приведений на рис. 4.18.

При високому рівні сигналу на вході  $WR$  інформація буде записуватись у тригери DD1.1 – DD1.4. Для її зчитування необхідно подати сигнал високого рівня на вхід  $RD$ .

Іноді в  $D$ -тригер вводять вхід  $R$  (рис. 4.16, б). Але використовувати його можна лише при  $C = 0$ . При  $C = 1$  можлива поява забороненої ситуації  $\bar{S} = 0, \bar{R} = 0$ .

У деяких схемах вводять також  $V$ -вхід (від англ. *Valve* – клапан), що дає можливість вимикати тригер. Такі схеми називаються *DV-тригерами*.

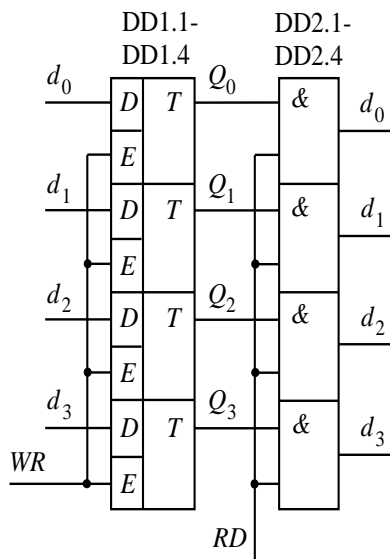


Рис. 4.18

Таблиця 4.8

Режим роботи	Вхід		Вихід	
	$E$	$D$	$Q$	$\bar{Q}$
Дозвіл передачі даних	1	0	0	1
	1	1	1	0
Зберігання	0	x	$Q_{n-1}$	$\bar{Q}_{n-1}$

Якщо в схемі синхронізованого імпульсом  $D$ -тригера замінити вхід  $D$  на інверсний  $\bar{D}$ , то виходи  $Q$  та  $\bar{Q}$  поміняються місцями. Така властивість може бути корисною при використанні мікросхем, в яких виведені лише прямі виходи (див. мікросхеми 555ІР22 та 1533ІР33).

При проектуванні схем з використанням  $D$ -тригерів необхідно, особливо для швидкодіючих пристроїв обробки інформації, строго дотримуватись часових співвідношень між сигналами на  $C$ - та  $D$ - входах. Ці співвідношення для схеми рис. 4.16, б наведені на часових діаграмах рис. 4.17, б, побудованих з урахуванням станів невизначеності. Перехідний процес характеризується

значеннями затримок розповсюдження сигналу по двох каналах:  $t_{3\ CQ}$  – від входу  $C$  до виходу  $Q$  при  $D = const$ , та  $t_{3\ DQ}$  – від входу  $D$  до виходу  $Q$  при  $C = 1$ . Водночас кожен з каналів характеризується величинами затримок при перемиканні з **1** на **0** та навпаки.

Для надійного запису інформаційного  $D$ -сигналу перехідний процес у схемі, обумовлений зрізом  $C$ -сигналу, не повинен накладатись на перехідний процес, обумовлений зміною стану  $D$ -входу. Це означає, що будь-які зміни стану  $D$ -входу повинні закінчитися за деякий час до зрізу  $C$ -сигналу (час підготовки  $t_{\Pi}$ ) та можуть знову початись не раніше ніж через час підтримки  $t_H$ . Вони оцінюються як  $t_{\Pi} = (1 - 2) t_H$ ;  $t_3 = (0 - 1) t_H$ , де  $t_H$  – час невизначеності стану логічного елемента при перемиканні.

Необхідність урахування вказаних інтервалів характерна для всіх функціональних вузлів, що мають вхід синхронізації. Крім цього, існує мінімально допустима тривалість синхроімпульсу, що забезпечує відсутність некоректного розвитку перехідних процесів від фронту та зрізу імпульсів.

У  $D$ -тригері розв'язуються проблеми  $S = R = 1$ , але якщо інтервал часу між спадами  $D$  і  $C$  сигналів досить короткий, то внутрішній  $RS$ -тригер може опинитись у метастабільному стані, з якого, як вказувалось вище, він може перейти у непередбачений стан.

На завершення, розглянемо ще одну важливу ситуацію – ситуацію “гонок” (“критичних змагань”), яка виникає при використанні тригерів.

У схемі (рис. 4.16, б) можлива наступна ситуація. Припустимо, що тривалість зрізу  $C$ -імпульсу набагато перевищує тривалість затримок елементів, на яких виготовлено  $D$ -тригер. Припустимо також, що поріг перемикання DD1 перевищує поріг елемента DD2. Якщо в цій ситуації на  $D$ -вході перед фіксує зрізом буде **1**, то вихід DD1 перемкнеться в **1** відразу після початку зрізу  $C$ -сигналу. Елемент DD2, маючи низький поріг перемикання, сприймає  $C$ -сигнал ще як **1**, тому, прийнявши на свій вхід **1**, він перемкнеться на нуль, записавши “**0**” у вихідний тригер на ЛЕ DD3-DD4, замість очікуваної “**1**”.

Звідси випливає, що протяжність зрізу  $C$ -сигналу повинна бути достатньо малою, порівняно з часом затримки елементів тригера, або близькою до неї.

У схемі рис. 4.16, *a* подібних ситуацій не виникає завдяки допоміжному елементу DD1.

Слід зазначити, що в практиці схемотехніки існує багато варіантів  $D$ -тригерів.

На рис. 4.19, *a – б* наведені ще дві схеми, розглянути особливості роботи яких пропонуються читачам.

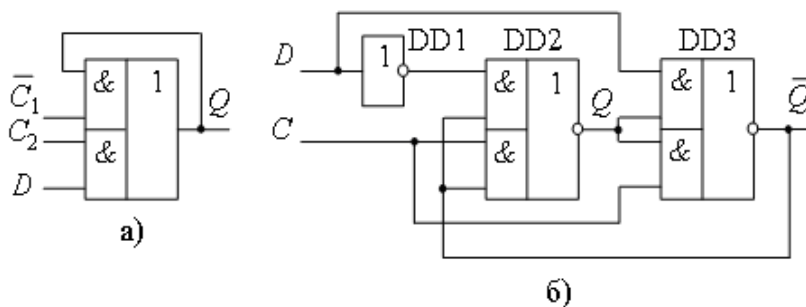


Рис. 4.19

Перша з них називається двотактним  $D$ -тригером, а друга –  $D$ -тригер з активним низьким рівнем  $C$ -сигналу.

Одним з найбільш досконалих варіантів схем тригерів є шестиелементний  $D$ -тригер (рис. 4.20, *a*).

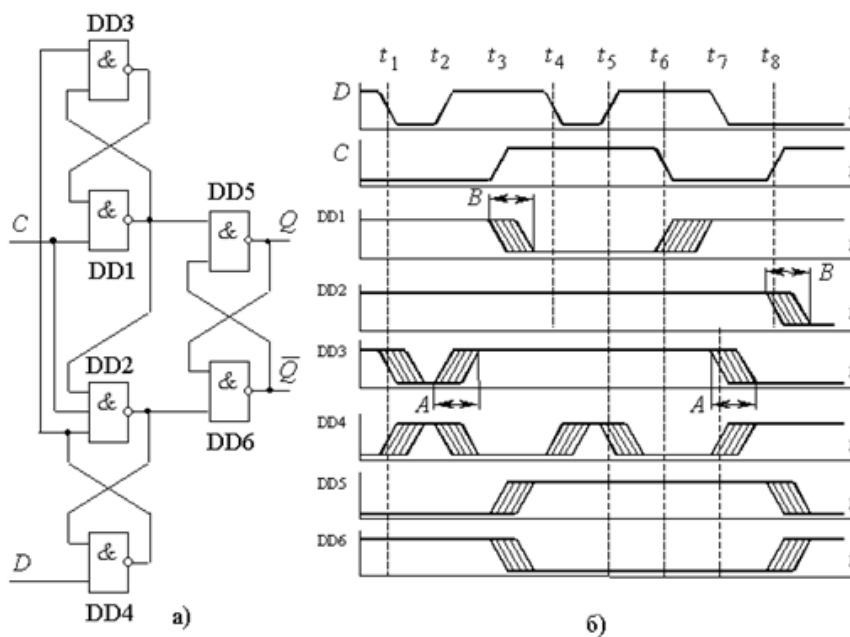


Рис. 4.20

Прикладом шестиелементного  $D$ -тригера є мікросхема ТМ2. Елемент пам'яті в цій схемі представлений  $RS$ -тригером на елементах DD5 – DD6. Елементи DD1, DD2 виконують функцію керуючих  $RS$ -тригером. При  $C = 0$  вони обидва закриті, і тригер перебуває у режимі збереження (моменти часу  $t_1$ ,  $t_2$ ,  $t_7$  на часовій діаграмі рис. 4.20, б).

За фронтом  $C$ -сигналу (моменти  $t_3$  та  $t_8$ ) в залежності від рівня сигналу на  $D$ -вході відкривається один з елементів DD1 або DD2, щоб записати в елемент пам'яті відповідні дані. Мікросхеми DD3 та DD4, які керуються  $D$ -сигналом, своїми сигналами готують DD1 або DD2 для того, щоб він відкрився за фронтом  $C$ -імпульсу.

Наведена схема допускає будь-які співвідношення затримок. Це можна побачити з часової діаграми, яка показує, що всі елементи перемикаються один за одним. У схемі відсутні паралельні шляхи, на яких можливі гонки.

Час підготовки  $t_{п}$ , протягом якого сигнал на вході  $D$  не повинен змінюватись, визначається часом стану невизначеності на всіх входах логічних елементів DD1, DD2. Найдовшим є інтервал перемикавання ЛЕ DD4-DD3 (інтервал часу  $A$ , рис. 4.20, б). Час затримки, протягом якого після фронту  $C$ -сигналу рівень  $D$  не повинен змінюватись, визначається часом, необхідним для того, щоб DD1 заблокував вхід DD2 або щоб DD2 заблокував вхід DD4 (інтервал часу  $B$  після моментів  $t_3$ ,  $t_8$ ). Це найбільш песимістична оцінка при умові, що у елементів нормовані лише максимальні значення затримок. Якщо ж забезпечується стан, при якому затримка логічного елемента DD2 ніколи не

перевершує затримки елемента DD4, то час затримки перетворюється на  $0$ .

Таблиця 4.9

$S_n$	$R_n$	$C$	$D_n$	$Q_{n+1}$
0	0	1	0	0
0	0	1	1	1
0	0	1	x	$Q_n$
0	1	x	x	0
1	0	x	x	1
1	1	x	x	$Q = \bar{Q}$

**Приклад 4.5.** Дати пояснення особливості роботи  $D$ -тригера К561ТМ2 (аналоги західних фірм-виробників – 4013ВDМ, CD4013AD) за допомогою таблиці станів (табл. 4.9).

*Розв'язання.* Виходячи з перших двох рядків таблиці, бачимо, що маємо справу з динамічним тригером, який в синхронному режимі за фронтом синхроімпульсу забезпечує запис інформації з  $D$ -входу.

Третій рядок інформує про те, що за спадом синхроімпульсу тригер на значення  $D$ -входу не реагує. Четвертий і п'ятий рядки характеризують режим роботи асинхронного  $RS$ -тригера, виготовленого на ЛЕ **2АБО-НІ**. Шостий рядок – заборонений асинхронний режим. Для роботи пристрою в режимі  $D$ -тригера асинхронні входи  $S$  і  $R$  необхідно заземлити.

**Приклад 4.6.** На рис. 4.21 приведені умовні зображення  $D$ -тригера К561ТМ2. Зробити необхідні з'єднання в схемі для забезпечення роботи в режимі синхронного  $D$ -тригера і привести відповідні часові діаграми.

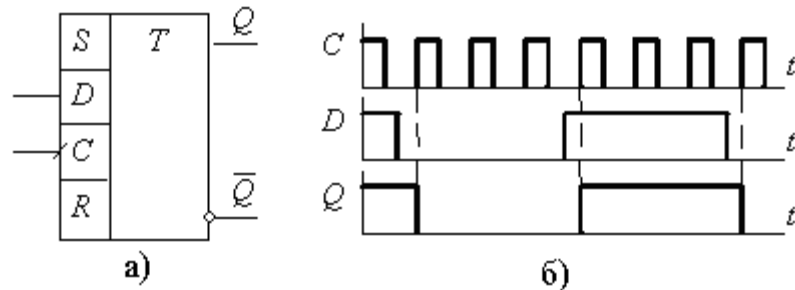


Рис.4.21

*Розв'язання.* Входи  $S$  і  $R$  необхідно приєднати до загальної шини (заземлити). Часові діаграми для виходу  $Q$  не потребують допоміжних пояснень.

Динамічні  $D$ -тригери будуються також на основі використання двох статичних  $D$ -тригерів (рис. 4.22).

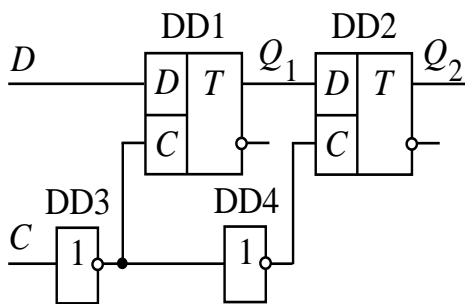


Рис. 4.22

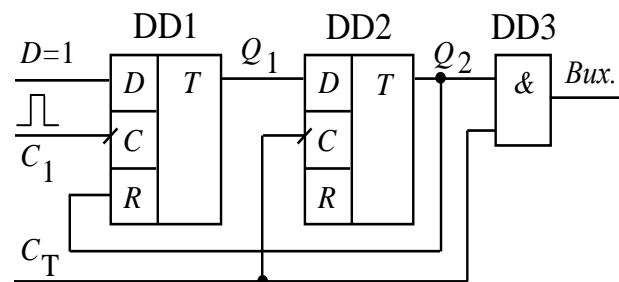


Рис. 4.23

При  $C = 0$  тригер  $DD1$  відкритий по відношенню до  $D$ -входу, тому  $D$ -сигнал перезаписується на вихід  $Q_1$ . При зміні стану синхровходу з **0** в **1** вихід  $DD3$  через інтервал затримки  $t_{3(DD3)}$  інвертується в нульовий стан, і тригер

DD1 стає непрозорим по відношенню до  $D$ -входу. В той же час, через інтервал затримки  $t_{3(DD4)}$  інформація з виходу  $Q_1$  перезаписується на вихід  $Q_2$ .

Для забезпечення роботи тригера за спадом синхросигналу необхідно зі схеми, приведеної на рис. 4.22, вилучити інвертор DD3.

Динамічні  $D$ -тригери знаходять широке використання в пристроях з послідовною передачею інформації з затримкою на такт, а також в пристроях формування імпульсів синхронізації, визначення різниці близьких частот та ін.

На рис. 4.23 приведена схема пристрою, призначеного для формування імпульсу синхронізації. Розглянемо його роботу.

Спочатку обидва тригери DD1 та DD2 знаходяться у стані, при якому  $Q_1 = Q_2 = 0$ , а  $D_1 = 1$  постійно. Як тільки на вхід  $C_1$  буде поданий синхросигнал, за його фронтом на виході  $Q_1$  з'явиться сигнал логічної одиниці, який поступає на вхід  $D$  тригера DD2. За фронтом наступного імпульсу послідовності  $C_T$  вихід тригера DD2 встановиться в одиничний стан, а тригер DD1 за сигналом, що подається на вхід  $R$ , перейде у початковий стан. Наступний тактовий імпульс послідовності  $C_T$  встановить тригер DD2 у нульовий стан. Тривалість імпульсу на виході другого тригера буде рівною тривалості періоду тактової послідовності  $C_T$ , а на виході ЛЕ DD3 з'явиться один тактовий імпульс.

Такий пристрій знаходить використання в цифрових приладах для вимірювання частоти для формування вимірювального інтервалу. На вхід  $C_T$  у такому випадку подається зразкова частота, а до входу  $C_1$  прикладається імпульс запуску чергового циклу вимірювання. Тривалість імпульсу, що отримується на виході DD2, буде тим вимірювальним інтервалом, який заповнюватиметься імпульсами еталонної частоти, а на виході DD3 отримуватиметься підраховувана кількість імпульсів еталонної частоти.

Прикладом пристрою, що може бути використаним для віднімання двох частот, може служити синхронний  $D$ -тригер, на входи  $D$ - і  $C$ - якого подаються послідовності імпульсів, частоти яких відрізняються на 30-40%. Якщо імпульси сформовані у вигляді меандрів ( $t_i = t_n = T/2$ ), то внаслідок поступової зміни

фазового зсуву між імпульсними послідовностями фронти імпульсів, що подаються на вхід  $C$ , співпадатимуть то з позитивними півперіодами послідовності, що подається на вхід  $D$ , то з нульовими. Як наслідок, частота імпульсів на прямому виході  $Q$  тригера визначатиметься як модуль різниці частот двох порівнюваних послідовностей.

#### 4.4. JK-тригери

Цей тип тригерів за логікою роботи подібний до  $RS$ -тригерів, але, на відміну від них, не має невизначених переходів.

Скорочена таблиця станів (табл. 4.10) пояснює наведену різницю, якщо прийняти, що  $S$ -вхід відповідає  $J$ -входу, а  $R$ -вхід – відповідно,  $K$ -.

Таблиця 4.10

$J_n$	$K_n$	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\bar{Q}_n$

З таблиці витікає, що поява комбінації  $J = K = 1$  у кожному такті призводить до зміни стану тригера на протилежний. На відміну від  $RS$ -тригерів, схеми  $JK$ -тригерів виготовляються тільки синхронними.

Однотактні тригери відрізняються наявністю зворотних зв'язків з виходів на входи (рис. 4.24), а також елементами часової затримки (елементи DD3, DD4). Стан виходів  $JK$ -тригера залежить не тільки від сигналів на входах  $J$  та  $K$ , але й від логічно пов'язаних з ними сигналів з виходів  $Q$  та  $\bar{Q}$ .

Робота асинхронного  $JK$ -тригера описується наступним характеристичним рівнянням:

$$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n. \quad (4.11)$$

Для синхронного  $JK$ -тригера є справедливим рівняння:

$$Q_{n+1} = C(J_n \bar{Q}_n + \bar{K}_n Q_n) + \bar{C} Q_n. \quad (4.12)$$

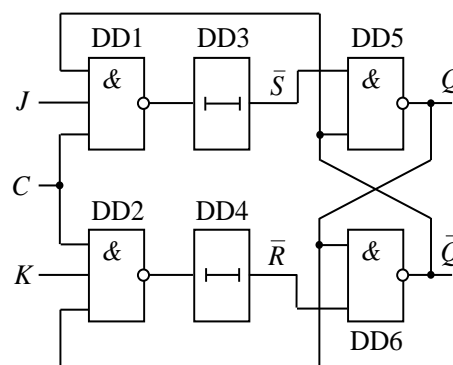


Рис. 4.24

Розглянемо більш детально роботу тригера. Як і в попередніх схемах, в якості запам'ятовуючого елемента в ньому використаний  $RS$ -тригер з входами активного низького рівня  $\bar{S}$  і  $\bar{R}$ . Вхідна логіка задає алгоритм функціонування тригера.

При відсутності вхідних сигналів, тобто при  $J_n = K_n = 0$  на виходах DD1 і DD2 зберігаються високі рівні сигналів, які забезпечують режим зберігання для внутрішнього  $RS$ -тригера (мікросхеми DD5 і DD6). Низький рівень сигналу на виході DD1 або DD2, який може змінити стан  $RS$ -тригера, забезпечується лише при високому рівні сигналів на всіх трьох входах відповідного логічного елемента. Оскільки на один з входів ЛЕ DD1 і DD2 сигнал подається з діагонального виходу тригера, то низький рівень сигналу можливий лише на одному з виходів керованої логіки. Цим і досягається усунення проблеми невизначеності виходу при  $J_n = K_n = 1$ . Тому при появі такої комбінації вхідних сигналів тригер змінить свій стан на протилежний, незалежно від того, в якому стані він перебував раніше. Дійсно, якщо  $Q_n = 0$ ,  $\bar{Q}_n = 1$ , то на виході DD1 з'явиться логічний нуль, який по входу  $\bar{S}$  змінить стан тригера, і на виході  $Q$  з'явиться "1". Аналогічна ситуація матиме місце при  $Q_n = 1$ ,  $\bar{Q}_n = 0$ . У цьому випадку логічний нуль на виході DD2, що по входу  $\bar{R}$  перекине  $RS$ -тригер у нульовий стан.

Елементи DD3, DD4 часової затримки у цій схемі відіграють роль стабілізаторів станів тригера і безпосереднього впливу на його функціональні властивості не здійснюють. Вони створюють часову затримку між моментом подачі вхідної інформації  $J_n \bar{Q}_n$  або  $K_n Q_n$  та початком формування вихідного стану  $Q_{n+1}$  та  $\bar{Q}_{n+1}$ . Без цих елементів виникає можливість генерації коливань в зв'язку з тим, що з кожною зміною вхідних сигналів на входах створювалася б комбінація, яка викликала б нову зміну станів тригера. Для уникнення подібної ситуації необхідно, щоб затримка перевищувала інтервал дії тактових сигналів.



Такі принципи використані в мікросхемах K531TB9, K555TB9 (зарубіжні аналоги яких – SN54112, SN74112).

Уникання можливості появи автогенерації коливань просто забезпечується в тригерах з двоступінчатим керуванням через те, що обидва ступені тактуються послідовно. На такому принципі побудовані тригери КР1533ТВ1 (аналоги західних фірм-виробників – SN5472, SN7472).

Дещо спрощена схема такого тригера приведена на рис. 4.25, а.

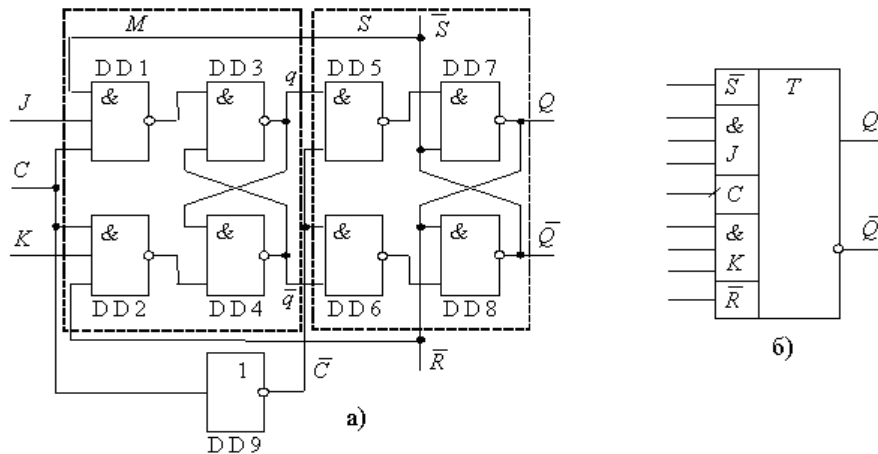


Рис. 4.25

На рис. 4.26 приведені часові діаграми, що пояснюють особливості зміни станів при умові  $J = K = 1$ , при якому тригер змінює стан виходів на протилежний. Цей режим широко використовується в цифровій техніці і називається *T-режимом* (від англ. *Toggle* – перевертатись).

Приведена на рис. 4.25, а схема тригера є статичним тригером, що спрацьовує за зрізом синхроімпульсу, що відображено на рис. 4.26.

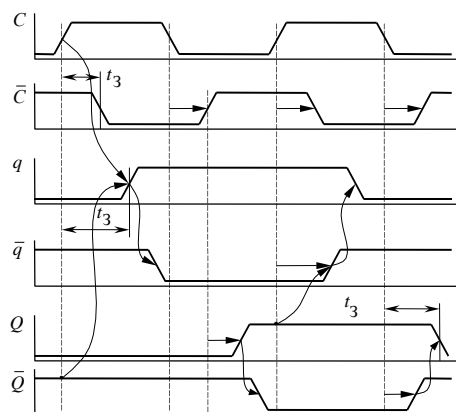


Рис. 4.26

Таблиця 4.11

Режими	$C_n$	$J_n$	$K_n$	$Q_{n+1}$
Збереження	x	0	0	$Q_n$
Обнуління	1	0	1	0
Установка	1	1	0	1
T-режим	1	1	1	$\overline{Q_n}$

Часто замість фронту або зрізу використовується стрілка з направленням вгору або вниз.

У тригері, показаному на рис. 4.25, а, є паралельні шляхи розповсюдження сигналу:  $C$ -імпульс проходить через  $M$ -тригер і інвертор DD9,

а потім обидва сигнали зустрічаються на входах елементів DD5, DD6 (вхід  $q$ , наприклад). Тобто, у схемі існує вірогідність виникнення “гонок”. Якщо затримка інвертора DD9 перевершує затримку  $M$ -тригера, то при появі  $C$ -сигналу новий стан  $M$ -тригера може встигнути перейти в  $S$ -тригер раніше, ніж інвертор DD9 закрий елементи DD5, DD6. Вихід  $Q$  тригера при цьому зміниться за фронтом наступного  $C$ -імпульсу, а не за зрізом, що приведе до збою у наступній схемі.

Приведені на рис. 4.26 часові діаграми показують рекомендовані часові співвідношення між затримками сигналу синхронізації в інверторі DD9 і в  $M$ -тригері. В мікросхемі KP1533TB1 цей недолік усунений.

У схемах динамічних тригерів можливість появи гонок по входу завжди зв'язана з крутизною фронту або спаду  $C$ -імпульсу. Технічно в інтегральних мікросхемах ця проблема вирішується різними шляхами, але при проектуванні тригерних схем не слід їх піддавати необов'язковим випробуванням.

**Приклад 4.7.** Для  $JK$ -тригера  $MS$ -типу, схема якого наведена на рис. 4.27, при початкових умовах  $J = K = Q_1 = Q_2 = 0$  задається наступна послідовність сигналів:

1.  $J, 0 \rightarrow 1$ ;  $K, 0 \rightarrow 0$ ;  $C$  – перший імпульс синхронізації;
2.  $J, 1 \rightarrow 1$ ;  $K, 0 \rightarrow 1$ ;  $C$  – другий імпульс синхронізації;
3.  $J, 1 \rightarrow 0$ ;  $K, 1 \rightarrow 1$ ;  $C$  – третій імпульс синхронізації;
4.  $J, 0 \rightarrow 0$ ;  $K, 1 \rightarrow 0$ ;  $C$  – четвертий імпульс синхронізації.

Зміна сигналів на входах  $J$  і  $K$  відбувається в інтервали часу, коли синхро-сигнали відсутні.

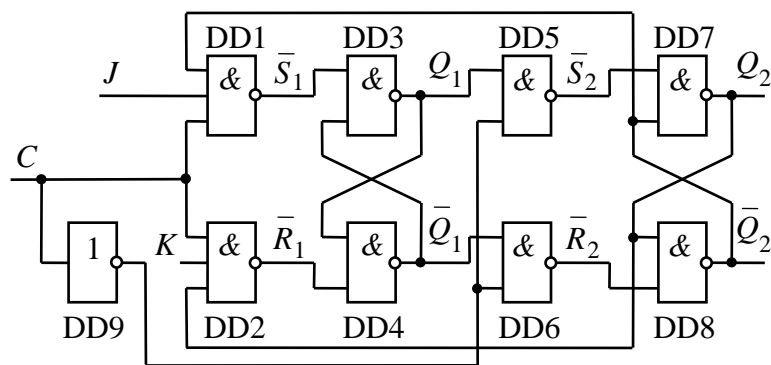


Рис. 4.27

Необхідно побудувати часові діаграми на входах  $J$ ,  $K$  і виходах  $Q_1$ ,  $Q_2$  тригера в інтервали часу, протягом яких були подані чотири синхросигнали  $C$ .

*Розв'язання.* Часові діаграми приводяться на рис. 4.28.

У момент часу  $t_1$  сигнал на вході  $J$  переходить з низького рівня в високий. Через інтервал часу  $t_2$ , який повинен бути достатнім для усталення перехідних процесів на  $J$ -вході, подається сигнал  $C$ . Оскільки на вході DD1 в цей час маємо два сигнали високого рівня  $J$  і  $\bar{Q}_2$ , то на виході DD1 сигнал  $\bar{S}_1$  прийме низький рівень, в той час як вихід  $\bar{R}_1$  ЛЕ DD2 матиме високий рівень логічного сигналу. Таке співвідношення сигналів  $\bar{S}_1$  і  $\bar{R}_1$  приведе до появи на виході  $Q_1$  сигналу високого рівня. Вказане розподілення рівнів напруг

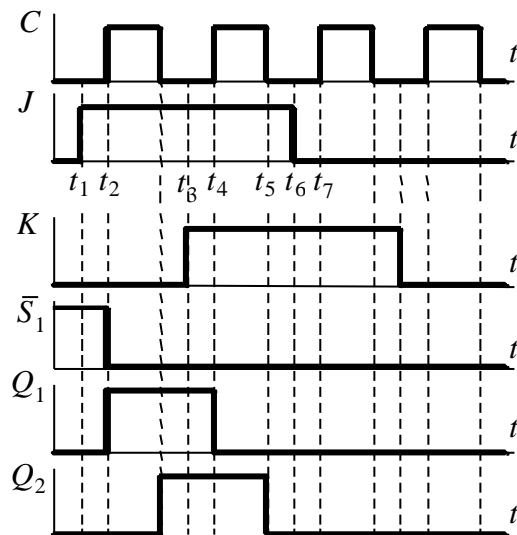


Рис. 4.28

залишатиметься протягом часу тривалості синхроімпульсу. При спаді синхроімпульсу на виході DD9 з'явиться високий рівень сигналу, який призведе до зміни стану DD5 і, відповідно, перезапису сигналу  $Q_1$  на вихід  $Q_2$ . У момент  $t_3$  з'являється сигнал високого рівня на вході  $K$ . На цей час  $Q_2 = 1$  і, відповідно, при  $K = 1$ ,  $Q_2 = 1$  поява в  $t_4$  другого синхроімпульсу приведе до зміни стану DD2,  $\bar{R}_1$  стане рівним нулю і, відповідно,  $\bar{Q}_1 = 1$ ,  $Q_1 = 0$ . За спадом другого синхроімпульсу низький рівень перезапишеться на вихід  $Q_2$ . У подальшому, при дії синхроімпульсів рівні сигналів не змінюються.

Розглянута схема  $JK$ -тригера, в якій сигнали  $J$  і  $K$  подаються на перший ступінь  $M$ , виконану за схемою  $RS$ -тригера, має ще й інші недоліки. Допустимо,

що тригер, зображений на рис. 4.25, *a*, знаходиться у стані  $Q = 0, J = 0$ , а вхід  $K$  знаходиться в одному з можливих станів. У цій ситуації  $C$ -імпульс не повинен змінити стан тригера. Але якщо перед появою  $C$ -зрізу на  $J$  вході з'явиться короткочасна завада, то вона пройде на вихід. Ця властивість називається властивістю „захоплення одиниці”. За аналогією, проявляється і властивість „захоплення нуля”. Якщо врахувати той факт, що завади з'являються в комбінаційних схемах, які створюють сигнали  $J$  та  $K$ , розглянута вище ситуація

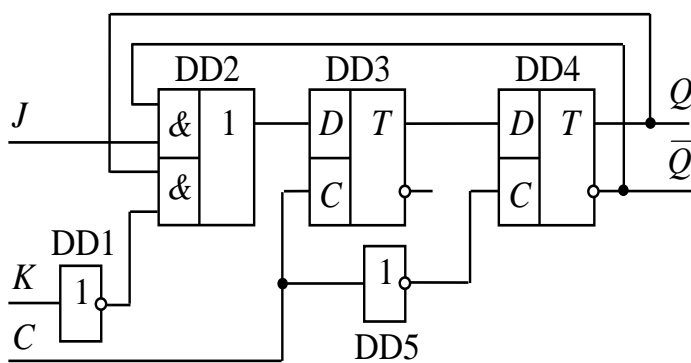


Рис. 4.29

приводить до важливого для проєктантів висновку – необхідно обов’язково забезпечити закінчення всіх перехідних процесів в логічних схемах, які формують сигнали  $J$  та  $K$ , перш ніж подається  $C$ -сигнал. Протягом дії

$C$ -сигналу рівні входів  $J$  та  $K$  не повинні змінюватись. На жаль, розглянута властивість „захоплення” характерна для багатьох тригерів.

Двоступінчаті  $JK$ -тригери можуть бути побудовані не тільки на основі  $RS$ -тригерів, а також і на основі  $D$ -тригерів. Останнім при цьому властива відсутність явища „захоплення”, що дозволяє змінювати стани входів  $J$  та  $K$  при  $C = 1$ .

Прикладом такого тригера є тригер, схема якого зображена на рис. 4.29. Забороняється лише змінювати значення входів  $J$  і  $K$  у короткі інтервали підготовки перед зрізом синхроімпульсу та витримки одразу після зрізу. Такі тригери раціонально використовувати при прийомі інформації з лінії, що забруднена випадковими завадами, адже тригер може відреагувати на них протягом дуже короткого інтервалу часу зрізу, підготовки, витримки.

Решту часу тригер ні на які завади не реагуватиме. Прикладом такого тригера є мікросхема К561ТВ1 (КР1561ТВ1) (аналоги провідних західних фірм-виробників – 4027BDC, CD4027AD, НСС4027ВК).

Як висновок з вищесказаного, слід зазначити, що при проектуванні схем з тригерами слід враховувати наступні часові параметри: затримку розповсюдження від синхровходу до виходу  $t_{3\text{CQ}}$ ; час підготовки  $t_{\text{П}}$ ; затримки  $t_3$  по керуючих входах; максимально допустимий період проходження імпульсів  $T_C$ ; максимально допустиму тривалість синхроімпульсів  $t_{\text{СТ}}$ .

На основі  $JK$ -тригера, як витікає з порівняльного аналізу табл. 4.7 і табл. 4.11, можливо побудувати  $D$ -тригер. Це пояснюється тим, що при наявності синхросигналу  $D$ -режим забезпечується, якщо входи  $J$  та  $K$  матимуть інверсні значення. Тобто, для створення  $D$ -тригера на базі  $JK$ -тригера необхідно з'єднати вхід  $K$  з входом  $J$  ( $D$ ) через інвертор (рис. 4.30).

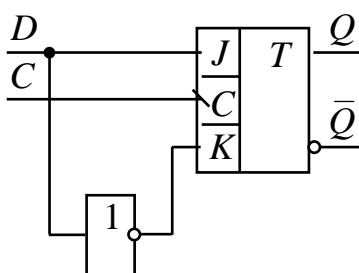


Рис. 4.30

Враховуючи те, що  $JK$ -тригери та  $D$ -тригери відносяться до категорії складних схем, при їх побудові використані всі можливості, щоб надати їм більше функціональних можливостей. Для цього інформаційні входи доповнюються установчими  $R$ - і  $S$ - входами. При цьому схема будується так, щоб  $R$ - і  $S$ - входи мали перевагу в своїй дії перед функціональними. Такі входи приєднуються безпосередньо на входи  $S$ - тригера, минаючи  $M$ -тригер та схеми керування, а тому є асинхронними. Після закінчення дії сигналів на асинхронних входах установлені ними значення виходів  $Q$  та  $\bar{Q}$  використовуються при наступній дії функціональних входів. Типові схеми тригерів з асинхронними установчими входами приведені на рис. 4.31, а, б.

Як правило, ці входи мають активні низькі рівні вхідних сигналів. У двоступінчатому тригері активний асинхронний вхід діє одночасно на тригери (як  $M$ -, так і  $S$ - типів), так і на мікросхеми керування DD1, DD2, виключаючи можливість їх дії при активному рівні  $C$ -сигналу.

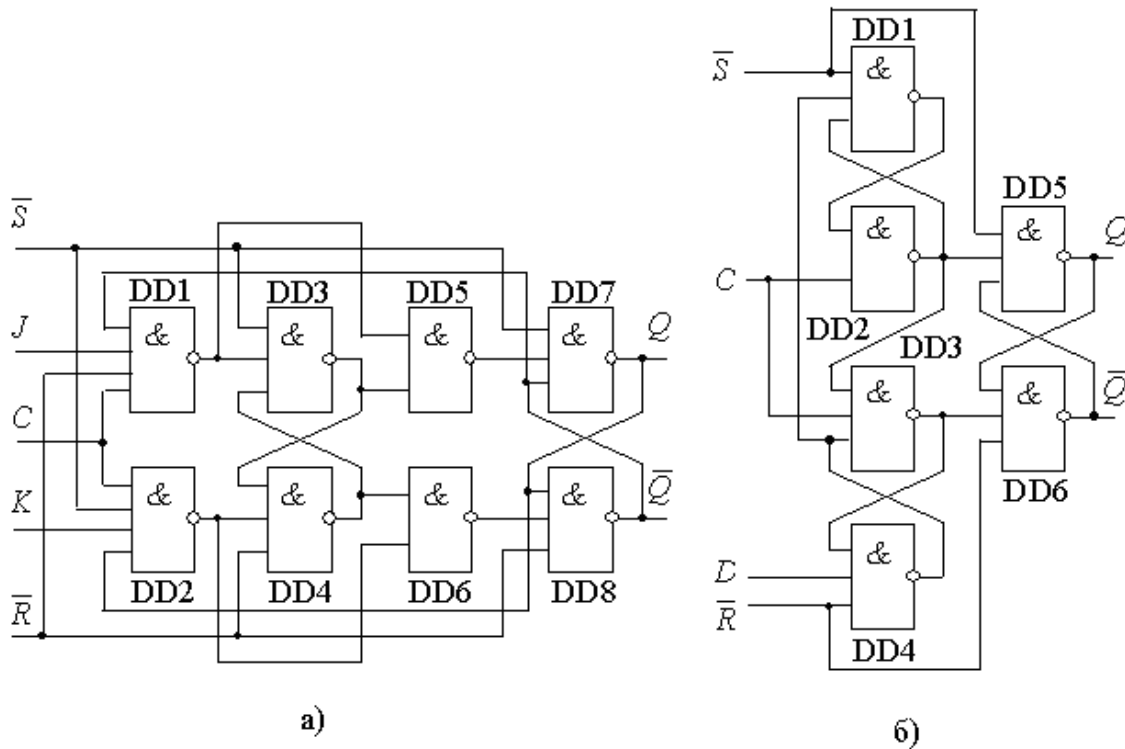


Рис. 4.31

У шестиелементному тригері (рис. 4.31, б)  $\bar{R}$  і  $\bar{S}$  входи перекривають можливі шляхи дії  $C$  і  $D$  входів і встановлюють необхідні значення вихідного  $RS$ -тригера, виготовленого на логічних елементах DD5, DD6. Наявність одночасної дії  $\bar{R}$  та  $\bar{S}$  входів приводить до невизначеного стану і повинна бути виключена.

#### 4.5. $T$ - та $TV$ -тригери

До тригерів  $T$ -типу, як вказувалось раніше, відносяться такі схеми, які за сигналом на  $T$ -вході переключаються у протилежний стан. Це зазвичай тригери з динамічним  $T$ -входом або з динамічним  $C$ -входом і статичним  $T$ -входом. У зв'язку з тим, що вони легко можуть бути організовані на базі динамічних

*D*- або *JK*-тригерів, як самостійні мікросхеми *T*-тригери не виготовляються. У зв'язку з їх широким використанням в лічильниках імпульсів, *T*-тригери з динамічним *T*-входом часто називаються *тригерами з лічильним входом*, або *лічильними тригерами*. В залежності від характеру дії *T*-динамічного входу, вони поділяються на *T*-тригери, які спрацьовують за фронтом *T*-імпульсу, та  $\bar{T}$ -тригери, що спрацьовують за зрізом *T*-імпульсу.

У табл. 4.12 приведений перелік можливих станів обох типів тригерів.

Таблиця 4.12

<i>T</i> -тригер		$\bar{T}$ -тригер	
$T_n$	$Q_n$	$T_n$	$Q_n$
0	$Q_n$	0	$Q_n$
1	$\bar{Q}_n$	1	$Q_n$
1	$Q_n$	1	$\bar{Q}_n$
0	$Q_n$	0	$Q_n$

З таблиці витікає, що у *T*-тригерах, на відміну від раніше розглянутих схем, стан виходу  $Q_{n+1}$  залежить не від значень інформаційних сигналів, а визначається тільки їх станом у попередньому такті.

Рівняння роботи асинхронного тактованого *T*-тригера має вигляд:

$$Q_{n+1} = Q_n \bar{T}_n + \bar{Q}_n T_n. \quad (4.13)$$

Для синхронного *T*-тригера маємо:

$$Q_{n+1} = (Q_n \bar{T} + \bar{Q}_n T) C + \bar{C} Q_n. \quad (4.14)$$

Рівняння (4.13) за виглядом співпадає з виразом для логічної функції „**ВИКЛ. АБО**”, звідки витікає, що *T*-тригер виконує операцію складання за модулем **2** вхідної змінної та функції, що визначає вихідний стан тригера. Рис. 4.32 пояснює особливості роботи тригера з прямим керуванням. З рисунку витікає, що, спрацьовуючи за фронтом вхідних імпульсів, він ділить їх частоту на **2**, при цьому тривалість імпульсу та паузи однакова.

*T*-тригер легко можна одержати з *JK*- або *D*-тригерів. На рис. 4.33 приводяться приклади створення схем *T*-тригерів.

Схеми рис. 4.33, а, б зрозумілі без пояснень. *D*-тригер також можливо перетворити в *T*-тригер. Це пояснюється тим, що вихід  $Q_n$  приймає в такти  $(n + 1)$  значення, що було на вході *D* в *n*-ому такті. Тому з'єднання *D*-входу з виходом  $\bar{Q}_n$  забезпечує зміну стану *Q* кожного разу при появі фронту *T*-входу.

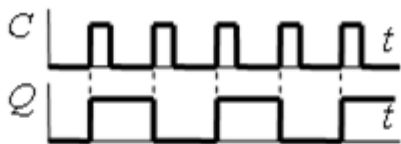


Рис. 4.32

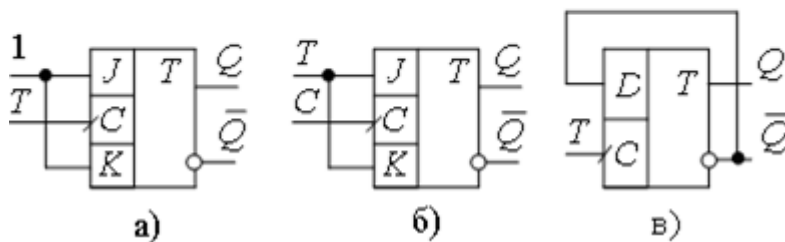


Рис. 4.33

Тригери *TV*, окрім лічильного *T*-входу, мають ще й керуючий – вхід для забезпечення дозволу прийому інформації. Рівняння *TV*-тригера має вигляд:

$$Q_{n+1} = (T_n \bar{Q}_n + \bar{T}_n Q_n) V_n + Q_n \bar{V}_n. \quad (4.15)$$

Цей тригер також легко може бути одержаний з *JK*-тригерів (рис. 4.34).

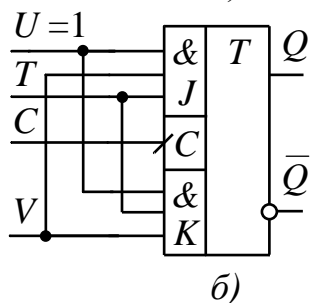
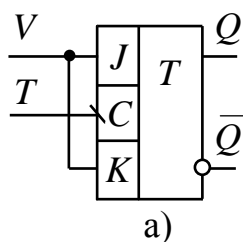


Рис. 4.34

У першому випадку (рис. 4.34, а) маємо асинхронний *T*-тригер, у другому (рис. 4.34, б) – синхронний.

Прикладом реалізації *T*-тригера на основі синхронного *RS*-тригера є схема, що представлена на рис. 4.35, а.

З приведених прикладів видно, що достатньо мати будь-який один тип тригерів, щоб на його основі можна було створювати тригери інших типів. Здебільшого такі перетворення витікають з



порівняння алгоритмів роботи тригерів. Як приклад, використовуючи алгоритм роботи  $RS$ -тригера:

$$Q_{n+1} = S_n + Q_n \overline{R_n} \quad \text{і} \quad S_n \cdot R_n = 0$$

і підставивши  $S_n = J \overline{Q_n}$  і  $R_n = K Q_n$ , отримаємо відому формулу:

$$Q_{n+1} = J \overline{Q_n} + \overline{K} Q_n,$$

тобто  $JK$ -тригер може бути побудований на основі  $RS$ -тригера.

Схема такого тригера приведена на рис. 4.35, б).

Таким же шляхом може бути побудований  $JK$ -тригер на основі  $D$ -тригера. Якщо прийняти, що:  $D_n = Q_{n+1} = J \overline{Q_n} + \overline{K} Q_n$ , то отримаємо схему з  $D$ -тригером, що приведена на рис. 4.35, в).

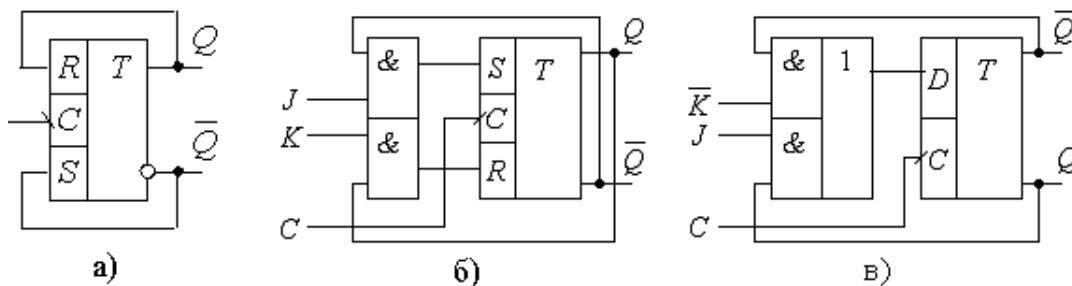


Рис. 4.35

Варіанти взаємних перетворень тригерів, приведені на рис. 4.33 і рис. 4.35, розкривають велику гнучкість і широкі можливості різноманітних тригерних схем.

#### 4.6. Несиметричні тригери

У цифрових пристроях широко використовується ще один тип тригерів, які називаються *несиметричними тригерами*, або *тригерами Шмідта*. Ці схеми теж мають два стійкі стани, зміна яких проходить під дією вхідних сигналів. Виготовляються вони так, що при відсутності вхідного сигналу рівень вихідної напруги конкретно визначений як високий або низький. Характерною властивістю несиметричних тригерів є те, що вхід або входи мають різні

порогові рівні сигналу, при якому тригер змінює свій стан під час наростання сигналу або його зменшення.

На рис. 4.36 зображені: умовне графічне зображення несиметричного тригера (рис. 4.36, а), передаточні характеристики  $U_{\text{Вих}} = f(U_{\text{Вх}})$  без інверсії (рис. 4.36, б) та з інверсією вихідного сигналу (рис. 4.36, в), а також часові діаграми, що пояснюють особливості його роботи.

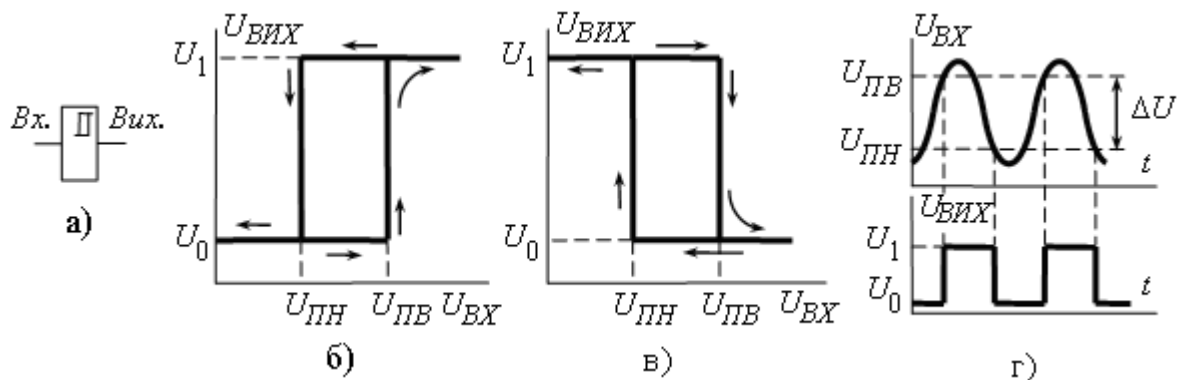


Рис. 4.36

Наявність різних порогових рівнів, при яких проходить зміна станів тригера, називається *гістерезисом*. Різниця між верхнім  $U_{\text{ПВ}}$  та нижнім  $U_{\text{ПН}}$  пороговими рівнями  $\Delta U_{\text{П}} = U_{\text{ПВ}} - U_{\text{ПН}}$  називається *шириною гістерезисної петлі*. Цей параметр визначає чутливість тригера до зміни вхідних сигналів. Чим менша величина  $\Delta U_{\text{П}}$ , тим більш чутливим є тригер.

У практичній схемотехніці виготовляють несиметричні тригери багатовходовими з розширенням по **I**, а також з інверсією вхідного сигналу. Наприклад, мікросхема ТТЛ КР1533ТЛ1 (аналоги провідних західних фірм-виробників – 7413РС, СDB413Е, МС7413N, SN7413N) у своєму корпусі має два тригери Шмідта з чотиривходовим елементом **I** на вході та інверсією по виходу, тобто реалізує функцію  $2(4\mathbf{I}\text{-НІ})$ . Мікросхема КМДН КР1564ТЛ2 (SN74НС14) – шість тригерів Шмідта-інверторів. Несиметричні тригери широко використовуються як порогові елементи для формування прямокутних імпульсів з сигналів довільної форми. Вони мають високу завадостійкість,

стандартизований поріг перемикання і високу точність роботи у випадках, коли швидкість зміни вхідного сигналу невисока.

У тих випадках, коли виникає необхідність відійти від стандартизованих рівнів, несиметричний тригер може бути виготовлений на базі інверторних мікросхем (рис. 4.37).

Робота такого тригера легко пояснюється на прикладі використання КМДН інтегральних схем, для яких можна прийняти достатньо фіксованими наступні параметри:

- вхідний опір  $R_{вх} \approx \infty$ ;
- пороговий рівень спрацьовування логічних елементів:  $U_{\Pi} \approx E_{ж} / 2$ .

При наростанні вхідної напруги від 0 до  $U_{\Pi}$  рівень напруги на вході DD1 еквівалентний логічному 0. Відповідно, і рівень напруги на виході DD2 також дорівнює 0. Тригер спрацює у випадку, якщо виконається співвідношення:

$$U_{вх1} \cdot \frac{R_2}{R_1 + R_2} \geq U_{\Pi}, \quad (4.16)$$

звідки знаходимо:

$$U_{вх1} \geq \frac{U_{\Pi} \cdot (R_1 + R_2)}{R_2} = \frac{E_{ж} \cdot (R_1 + R_2)}{2R_2}. \quad (4.17)$$

При наявності напруги  $+E_{ж}$  на виході співвідношення для зворотнього спрацьовування тригера має вигляд:

$$U_{вх0} = \left[ (R_1 + R_2) / R_2 \right] \cdot U_{\Pi} - \left( \frac{R_1}{R_2} \cdot E_{ж} \right) \approx \frac{(R_2 - R_1) \cdot E_{ж}}{2R_2}. \quad (4.18)$$

Використання цих формул дозволяє вибрати нижній та верхній порогови роботи несиметричного тригера.

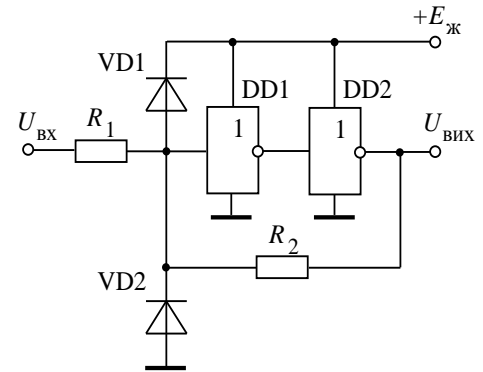


Рис. 4.37

**Приклад 4.8.** Визначити співвідношення між резисторами  $R_1$  та  $R_2$  (див. рис. 4.37) при наступних вихідних даних:  $E_{ж} = 5$  В;  $U_{вх1} = 4$  В;  $U_{вх0} = 1$  В.

*Розв'язання.* Користуючись формулами (4.17) і (4.18), складемо систему рівнянь:

$$\begin{cases} U_{вх1} = \frac{R_1 + R_2}{2R_2} E_{ж} \\ U_{вх0} = \frac{R_2 - R_1}{2R_2} E_{ж} \end{cases}; \quad \begin{cases} 8 \cdot R_2 = 5 \cdot (R_1 + R_2) \\ 2 \cdot R_2 = 5 \cdot (R_2 - R_1) \end{cases}.$$

З кожного з них маємо:  $R_2 = \frac{5}{3} R_1$ .

## 4.7. Синхронізація в цифрових схемах

### 4.7.1. Способи синхронізації та особливості їх використання

З підвищенням складності цифрових схем поява місцевих та загальних зв'язків у них призводить до того, що аналізувати і враховувати гонки в таких схемах стає практично неможливо. Радикальним вирішенням проблеми гонки є *синхронізація*. У практиці побудови систем синхронізації використовуються *однофазна і багаточастотна синхронізація, одночастотна і багаточастотна*.

Розгляд систем синхронізації почнемо з двофазної системи, коли всі схеми синхронізуються двома послідовностями імпульсів  $C_1$  та  $C_2$  однієї частоти  $f_T = T_T^{-1}$  та одного фазового зсуву  $T_\Phi$ . Тривалість імпульсів двох послідовностей однакова і дорівнює  $T_i$ . Для симетричної двофазної синхронізації  $T_T = 2 T_\Phi$ . Для несиметричної  $T_{\Phi1} \neq T_{\Phi2}$ .

При побудові синхронних цифрових схем їх розподіляють на дві групи. В одну групу входять комбінаційні схеми з визначеною кількістю входів та виходів. До іншої групи входять схеми *D-тригерів*, які мають особливість зберігати записану інформацію протягом одного такту.

Узагальнена цифрова схема може бути приведена до структури, що зображена на рис. 4.38 (аналог конвеєрної обробки інформації в мікропроцесорах і мікропроцесорних системах).

Схема включає в себе послідовно об'єднані групи  $D$ -тригерів, позначені на рис. 4.38 як  $DI, DII, \dots, DN$ , і комбінаційних схем, позначені  $KCI, KCII, \dots, KC(N-1), KCN$ . Кожна з груп тригерів об'єднується за принципом синхронізації від одного синхроімпульсу і в загальному плані представляє собою паралельний регістр, виконаний на  $D$ -тригерах.

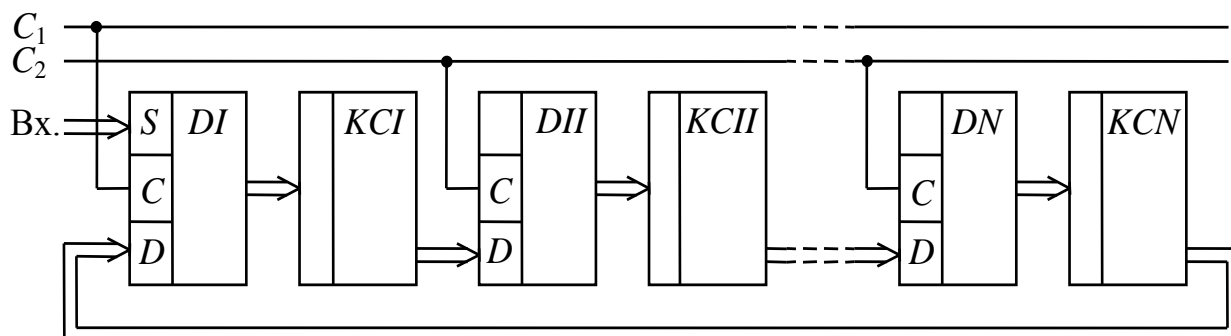


Рис. 4.38

Кожна група  $KC$  включає в себе чисто комбінаційну схемотехніку, яка виконує одночасно ряд логічних функцій, приймаючи інформацію з виходів попереднього регістра пам'яті і передаючи на наступний, який синхронізується другим синхроімпульсом. Внутрішні зворотні зв'язки в групі комбінаційних схем відсутні.

Фізичну суть процесів у схемі та ідеологію проектування цифрових пристроїв з двофазною синхронізацією пояснює рис. 4.39.

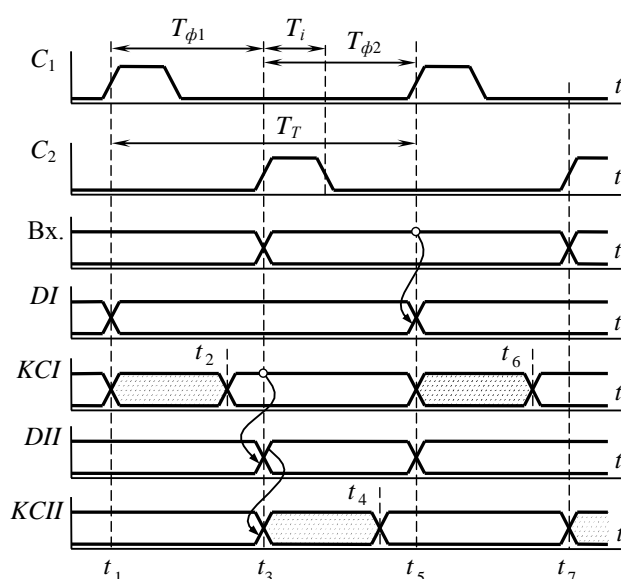


Рис. 4.39

За синхросигналом  $C_2$  чергова група інформаційних сигналів записується в регістрову групу  $D$ -тригерів  $N$  і передається для обробки на вхід комбінаційної схеми  $KCN$ . Через деякий інтервал часу ця інформація з'явиться на вхідних шинах  $D$ -тригерів  $DI$ . У момент часу  $t_1$  з'являється фронт синхросигналу  $C_1$ , за яким як вхідна інформація, що подається на входи  $S$ , так і інформація з виходів  $KCN$  записуються в тригери  $DI$ . По закінченні перехідних процесів в тригерах  $DI$  на їх виходах з'являються сигнали, які починають опрацьовуватись комбінаційною схемою  $KCI$ . Як правило, в схемі мають місце паралельні шляхи розповсюдження сигналів, тому вихідні сигнали  $KCI$  спочатку є невизначеними, адже вони спотворюються перехідними процесами. На рис. 4.39 картина гонок в  $KCI$  відображена на інтервалі часу  $t_1 - t_2$ . Інтервал часу  $t_1 - t_2$  перехідних процесів для  $DII$  не є безпечним через те, що всі тригери в цьому інтервалі часу закриті нульовим рівнем сигналу  $C_2$ . До моменту  $t_2$  всі перехідні процеси закінчуються, сигнали на виході  $KCI$  фіксуються, і в інтервалі часу  $t_2 - t_3$  ніякі стани в схемі не змінюються.

При подачі сигналу  $C_2$  у момент  $t_3$  встановлені значення виходів  $KCI$  записуються в  $DII$  і по завершенні в них перехідних процесів подаються на входи наступної комбінаційної схеми  $KCII$ . Процеси гонок в  $KCII$  проходять в інтервалі часу  $t_3 \dots t_4$  і до моменту  $t_5$  появи фронту синхроімпульсу  $C1$  встановлюються незмінними. При появі  $C1$  результати обробки сигналів в  $KCII$  перезаписуються в наступні регістрові схеми. Як результат, у синхронному пристрої йде циклічна багатоступенева обробка інформації в комбінаційних схемах, при якій комбінаційні схеми працюють по черзі. Завдяки цьому ніякі гоночні процеси в комбінаційних схемах не можуть внести похибку в обробку вхідних сигналів. Для цього необхідно лише, щоб інтервал часу  $T_\Phi$  перевершував максимальну тривалість перехідних процесів. Проектант завжди в змозі забезпечити таке співвідношення на основі паспортних значень максимальних затримок мікросхем.

Величина  $T_{\Phi}$  залежить від величини затримки  $t_3$  комбінаційних схем, яка може змінюватись у широких межах. Якщо  $t_3$  менша вибраної величини  $T_{\Phi}$ , то таке співвідношення не має негативних наслідків, виключаючи лише зниження швидкості обробки інформації. Але якщо затримка деяких комбінаційних схем перевищує величину робочого інтервалу  $T_{\Phi}$ , відповідно до рис. 4.39, схема стає непрацездатною. У подібних ситуаціях можуть використовуватись різні шляхи вирішення проблеми.

Найпростішим з них є збільшення тривалості  $T_{\Phi}$  і, відповідно, періоду синхроімпульсів. Як результат, це може суттєво знизити швидкодію розробленої схеми. Для того, щоб залишити частоту синхронізації незмінною, використовують несиметричну двофазну синхронізацію, при якій  $T_{\Phi 1} \neq T_{\Phi 2}$ . У цьому випадку, якщо можливо, комбінаційні схеми з більшим часом затримки розміщуються в більшому робочому інтервалі. Якщо подібна організація схемотехніки неможлива, то комбінаційну схему з великою тривалістю  $t_3$  розбивають на дві схеми і між ними встановлюють проміжний запам'ятовуючий вузол. Такий спосіб приводить до необхідної наступної перефазовки схеми. Широко використовується спосіб, при якому комбінаційні вузли з низькою швидкодією виділяють окремо і для них знижують частоту синхронізації до необхідної.

Найбільш гнучкий спосіб забезпечення високої швидкодії при наявності комбінаційних схем з великою затримкою – це використання багатofазних схем синхронізації, які використовуються у швидкодіючих пристроях. Переваги таких схем ілюструє рис. 4.40.

У залежності від величини конкретної затримки кожної комбінаційної схеми, на  $C$ -входи пристроїв пам'яті можливо заводити різні фази синхронізації і, відповідно, відкривати тригери-приймачі з затримкою на інтервали часу, кратні  $T_{\Phi}$  ( $T_{\Phi}, 2T_{\Phi}, 3T_{\Phi}, \dots$ ) відносно тієї фази, яка синхронізує передавач інформації. Недопустимо тільки синхронізувати тригери-приймачі синхросигналом тієї фази, якою синхронізувалися тригери-передавачі даної

комбінаційної схеми. Розглянутий спосіб широко використовується на практиці, адже він дає також можливість зменшити неробочі інтервали комбінаційних схем, що мають місце при очікуванні синхросигналу.

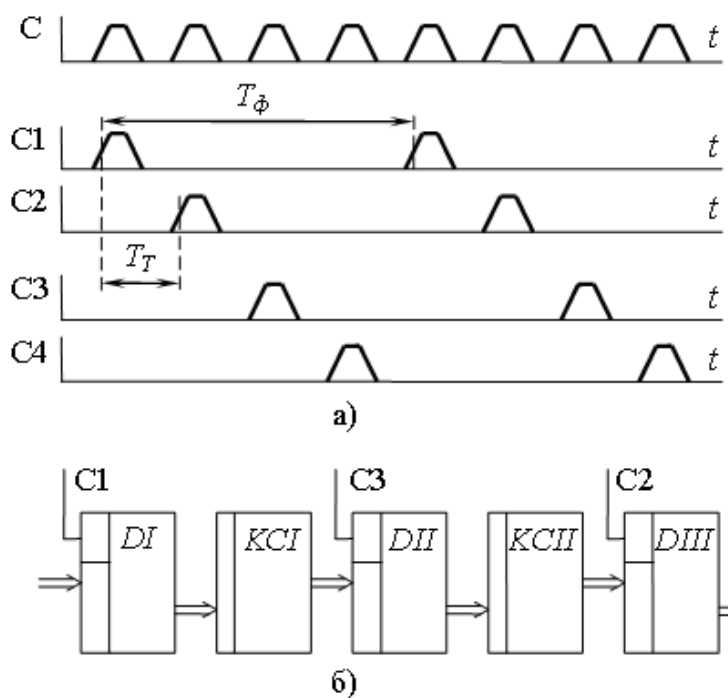


Рис. 4.40

На вибір тактової частоти генератора синхросигналів впливають також типи тригерів, розгалуженість схеми розподілення синхросигналів. Вказані особливості використовуються тільки досвідченими конструкторами при проектуванні складних цифрових схем автоматики. З розвитком потужних мікропроцесорів та машинних методів проектування ці особливості стають неактуальними.

У процесі проектування часто виникає необхідність у створенні зворотних зв'язків у синхронних схемах. У таких випадках необхідно дотримуватись наступних правил:

- в усіх схемах з двофазною синхронізацією петля зворотного зв'язку як з логічними елементами, так і без них повинна починатись з виходів тригерів, що синхронізуються однією фазою і закінчуватись на вході тригерів, що синхронізуються іншою фазою;



- недопустимі зв'язки, які передають сигнали з виходу однієї групи тригерів на вхід іншої, що синхронізується однією і тією ж фазою.

З вказаних правил витікає, що відсутність у схемах з двофазною синхронізацією замкнених кіл забезпечується тим, що у будь-який момент часу хоча б один з запам'ятовуючих пристроїв є відключеним і не передає інформацію з входу на вихід. Ці умови можуть бути забезпечені і при використанні однофазної синхронізації, якщо використовувати тригери, які не є “прозорими” для інформаційного сигналу, – наприклад, динамічні тригери. Особливості використання динамічних тригерів у схемах однофазної синхронізації пояснюються рис. 4.41.

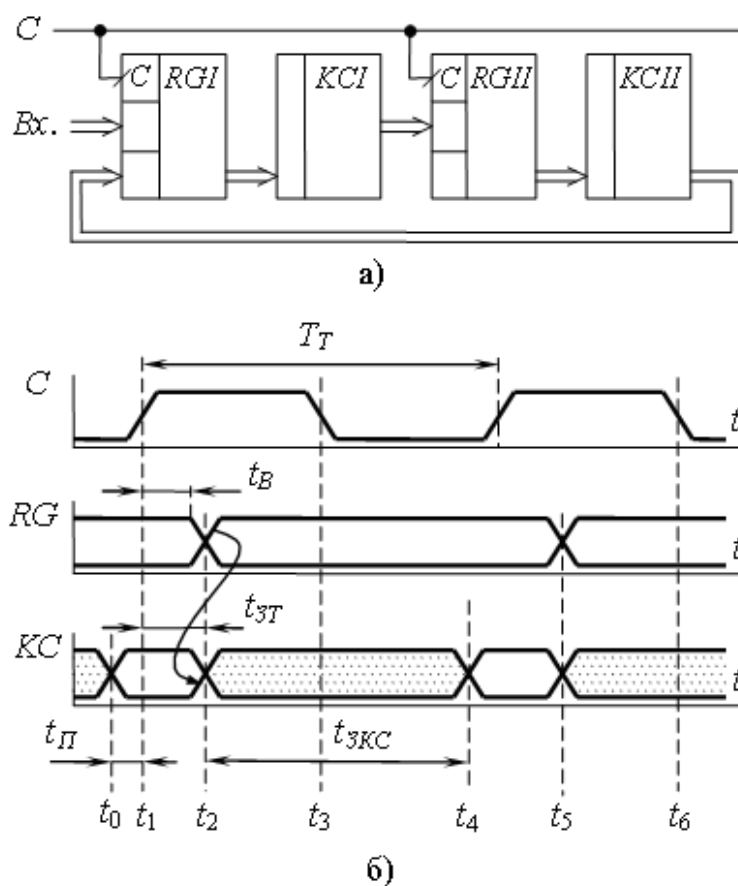


Рис. 4.41

Для розподілення комбінаційних схем  $KCI$ ,  $KCI$  використовуються динамічні тригери, запис інформації в які проходить за фронтом синхроімпульсу. До моменту часу  $t_0$  всі перехідні процеси в комбінаційній

схемі  $KC$  повинні завершитись, залишивши невеликий резерв часу підготовки  $t_{II}$  до появи фронту синхроімпульсу. Поява його в момент  $t_1$ , як відомо з роботи динамічних тригерів, не призводить до миттєвої зміни його станів, а початок цього процесу затримується на інтервал часу витримки  $t_B$ . Через інтервал затримки  $t_{3T}$  у зміні станів виходів тригерів нові дані з'являються на вході комбінаційної схеми (наприклад, схеми  $KCI$ ), і в ній починається можливий процес гонок, який проходить протягом часу затримки комбінаційної схеми  $t_{3KC}$  до моменту часу  $t_4$ . З аналізу розглянутих динамічних процесів витікає, що фактично тактова частота  $f_T = T_T^{-1}$  визначається часом  $t_{3KC}$ .

При однофазній синхронізації допускається використання тригерів, які у різних модулях мають різні моменти зміни станів. Наприклад, тригери пристрою  $RGI$  спрацьовують за фронтом, а  $RGI$  – за спадом. Завдяки такій комбінації виникає можливість використовувати  $KC$  з різними інтервалами затримок.

Особливістю однофазної синхронізації є складність її використання при розгалуженій системі синхронізації. Пояснюється це тим, що на окремих ділянках схеми синхронізації можуть виникати суттєві затримки. Інформаційні сигнали для ланки схеми, що розглядається, можуть не мати затримок. Як результат цього явища, неспівпадіння інформаційних тактів з фронтами синхросигналу – наприклад,  $N$ -го такту синхросигналу з  $(N+1)$  тактом інформаційного сигналу. Проконтролювати подібну ситуацію не завжди можливо, тому однофазна синхронізація знаходить обмежене використання.

Двофазна і багатофазна синхронізація не має вказаного недоліку через те, що вона має можливість попередньо врахувати будь-які затримки як в передачі синхронізуючих, так і інформаційних сигналів.

Важливо звернути увагу і на інші переваги багатофазної синхронізації. Перш за все, модулі пам'яті в багатофазних схемах синхронізації можуть бути побудовані на найпростіших синхронних тригерах, а принципових обмежень на типи тригерів практично немає. Немає обмежень також на часові

співвідношення в імпульсних послідовностях синхросигналу або крутизну фронтів, що є обов'язковими для динамічних тригерів. Вказані переваги багатофазних схем синхронізації, незважаючи на складність побудови розгалуженого дерева синхронізуючих сигналів, приводять до того, що в складних цифрових схемах використовуються переважно вони. Однофазні схеми знаходять використання лише в окремих вузлах або нескладних схемах – регістрах, лічильниках і т. п. Часто однофазна синхронізація використовується в мікроконтролерах, в яких немає необхідності багатоступінчатого розмноження сигналів.

#### **4.7.2. Узгодження вхідних сигналів**

Сигнали, що подаються на цифрові схеми з зовнішніх пристроїв, мають характеристики, які можуть суттєво відрізнитись від стандартів цифрових сигналів. Невідповідність між ними може суттєво вплинути на роботу цифрових схем.

Найпростіше вирішуються задачі узгодження полярності та амплітуд сигналів, адже в різних серіях цифрових мікросхем випускається достатня кількість перетворювачів рівнів сигналів.

Більш суттєвими задачами є такі, як узгодження тривалості фронтів та прив'язування вхідних сигналів до синхросигналів цифрових пристроїв.

**Скорочення фронтів.** Ця задача пов'язана з тим, що ряд джерел інформаційних сигналів має велику тривалість фронтів, яка значно перевищує тривалість фронтів сигналів логічних елементів. Як результат, навіть незначна неідентичність порогових рівнів логічних елементів призводитиме до значного часового розкиду моментів їх спрацьовування. В інтервалі напруг, близьких до порогових рівнів транзистори логічних елементів можуть знаходитись в активному режимі, що приводить до значних потужностей витрат потужності, перегріву, появи автогенерації, наскрізних струмів великих величин.

Для попередження подібних ситуацій необхідно перетворити пологі фронти в круті, які співпадали б з тривалістю фронтів використовуваних логічних елементів. Такі задачі вирішуються за допомогою тригерів Шміда, які мають фіксовані пороги спрацьовування. Ці елементи описані раніше.

**Тремтіння контактів.** Цей недолік будь-яких систем пов'язаний з вібрацією, що має місце при ударі. Це явище спостерігається в будь-яких контактних системах – реле, кнопках, клавіатурі комп'ютерів.

Частота вібраційних коливань контактів при їх замиканні знаходиться в межах від сотень герц до одиниць кілогерц, тому при зміні стану реле або при замиканні кнопкового контакту на вхід цифрової схеми поступає не сигнал, як зображено на рис. 4.42, *a*, а імпульсна послідовність (рис. 4.42, *б*), яка має місце в інтервалі часу  $t_1 - t_2$ .

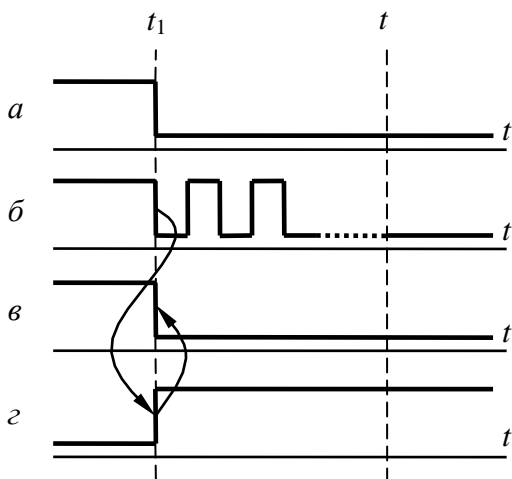


Рис. 4.42

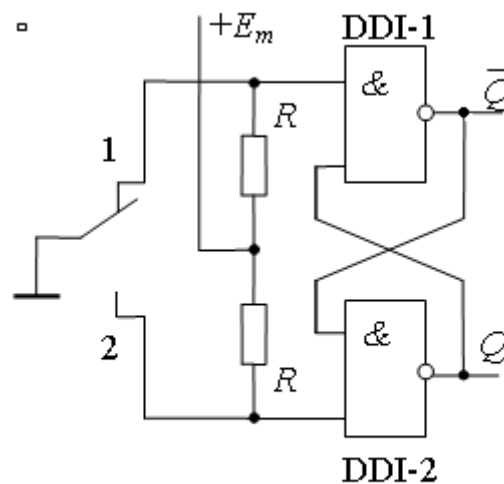


Рис. 4.43

Для ліквідації вказаного недоліку електромеханічних схем використовуються шунтуючі електронні пристрої.

На рис. 4.43 приведена схема пристрою для ліквідації вібраційних коливань реле. Його робота полягає в наступному. При знаходженні контакту реле в позиції **1** нульовий потенціал подається на вхід елемента DDI-1, і на виході *Q* (вихід елемента DDI-2) буде низький потенціал (рис. 4.42, *в*), а на інверсному високий. При зміні положення контакту (з **1** в **2**) на вхід DDI-2 поступить сигнал нульового рівня, і тригер змінить свій стан на одиничний,

тобто прямий вихід  $Q = 1$ , а інверсний вихід  $\bar{Q} = 0$ . Наявність вібраційних коливань, під час яких існують інтервали коли заземлюючий контакт відривається від контакту **2**, але не дотикається до контакту **1**, не зможе привести до зміни стану тригера.

На практиці використовуються різні засоби вирішення проблеми вібраційних коливань. Використовуються, наприклад, схеми одновібраторів з витримкою часу, що перевершує інтервал вібраційних коливань. Знаходить використання спосіб інтегрування вібраційних коливань за допомогою  $RC$ -інтеграторів з наступним використанням порогових елементів. У контролерах клавіатури усунення проблеми вібраційних коливань досягається за рахунок програмного багаторазового опиту стану натиснутої клавіші в заданому інтервалі часу.

### 4.7.3. Синхронізатори

*Прив'язка зовнішніх сигналів до синхроімпульсів* необхідна тому, що синхронні цифрові схеми приймають вхідні сигнали без похибок лише в визначені інтервали часу. Якщо вхідний сигнал подається на комбінаційну схему безпосередньо перед синхроімпульсом, то перехідні процеси в ній можуть не завершитись до появи синхроімпульсу, і в тригери буде записана хибна інформація. Інша причина пов'язана з реакцією на одиночні сигнали. Наприклад, сигнал від натискання кнопки може тривати багато періодів синхросигналу. У той же час, для цифрових синхронних схем його тривалість не повинна перевищувати один період синхрочастоти. Вказані задачі вирішуються за допомогою тригерних схем, які називаються *синхронізаторами*.

На рис. 4.44 приведена схема синхронізатора на базі статичних  $D$ -тригерів. Вхідний сигнал з однаковою ймовірністю може з'явитись у будь-який момент часу на інтервалі періоду синхросигналу.

Якщо його фронт на часовій діаграмі відображається лініями  $a$ ,  $b$ ,  $c$ , то тригер DD1 запише на свій вихід  $Q_1$  значення вхідного сигналу, яке за синхросигналом  $C_2$  переписується на вихід  $Q_2$  тригера DD2 і до появи  $C_1$  передається на вихід синхронізатора через логічний елемент DD4 **3I**. На виході матимемо інформаційний сигнал  $A$ , співпадаючий з синхросигналом  $C_2$ .

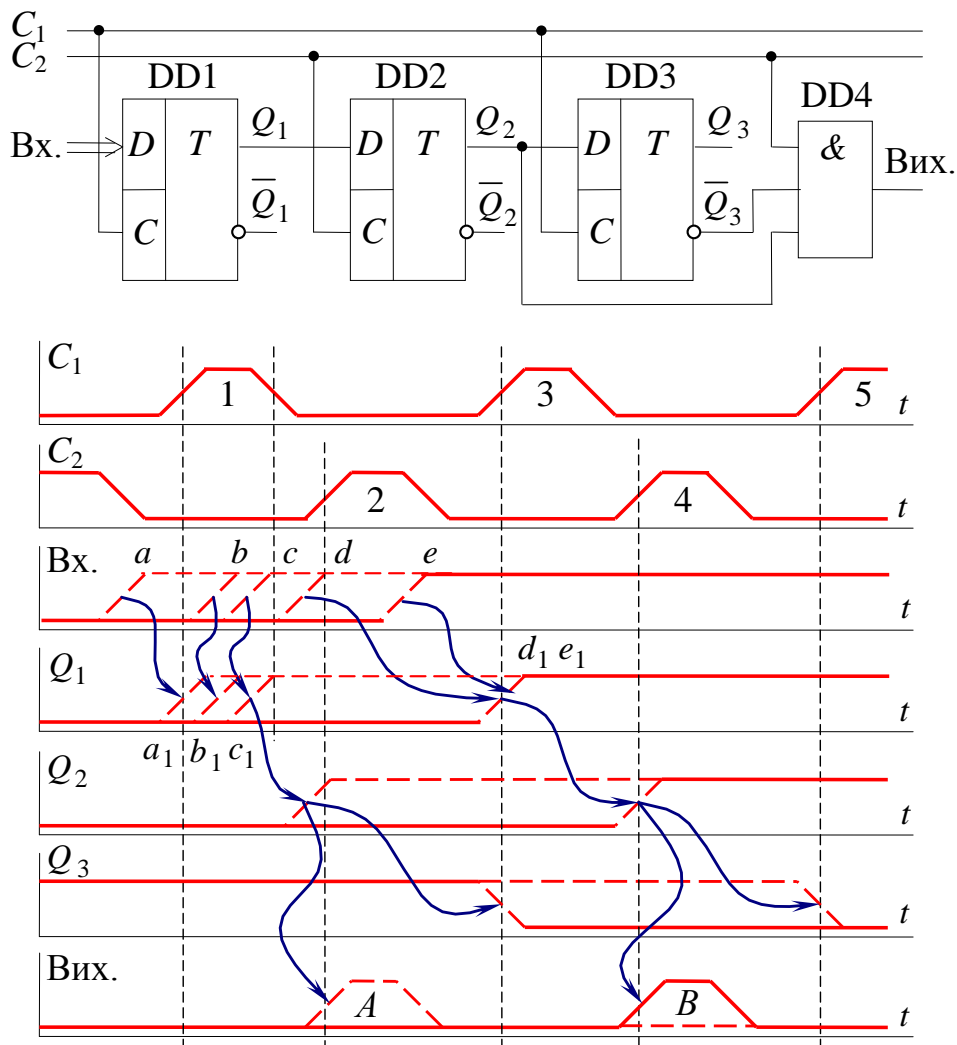


Рис. 4.44

Якщо вхідний сигнал з'являється пізніше синхронізуючого імпульсу  $C_1$  (по лінії  $d_1$ ,  $e_1$ ), то тригер DD1 прийме його за третім синхроімпульсом (лінії  $d_1$ ,  $e_1$ ), який також через DD2 і DD4 за синхросигналом  $C_2$  пройде на вихід у вигляді одиночного імпульсу. Повторення імпульсу неможливе, адже при  $Q_2 = 1$  та  $C_1 = 1$  тригер DD3 змінить свій стан і  $Q_3 = 0$ . Тобто, з'явиться сигнал заборони повтору вхідного сигналу. Зрозуміло, що, приймаючи асинхронний

сигнал за можливу наявність фазового скиду, на один тактовий інтервал не слід звертати увагу.

У розглянутій схемі тригер DD1 визначає, з яким із двох сусідніх синхроімпульсів слід зв'язати вхідний сигнал, але, коли б він не з'явився, приведена схема завжди відреагує на нього одним імпульсом. До того ж, вихідний імпульс буде стандартизованим по відношенню до схеми-приймача.

Якщо з'являється необхідність перетворити вхідний сигнал у постійний високопотенціальний рівень, то це можливо зробити, виключивши DD3.

Розглянута схема може служити основою для перетворення неоднорідної вхідної послідовності сигналів у періодичну.

Подібна схема може бути реалізована і з використанням динамічних D-тригерів та однофазною синхронізацією. Принципової різниці у побудові така схема не має.

У розглянутій схемі в рамках описаних особливостей роботи виникає сумнів щодо необхідності тригера DD2. Ці сумніви справедливі до того моменту, поки не з'явиться ситуація, при якій вхідний сигнал співпадає зі спадом імпульсу 1. Тоді, незважаючи на наявність позитивного зворотного зв'язку в тригерах, тригер DD1 може знаходитись у невизначеному стані відносно тривалий інтервал часу, затягуючи фронт на імпульсі по виходу Q1. Тригер DD2 у такому випадку виконує функцію швидкодіючого порогового елемента.

*Асинхронний обмін інформацією* має місце між цифровими пристроями, кожен з яких має свою власну схему синхронізації. В такому випадку сигнали, що поступають з іншого пристрою, сприймаються приймачем як асинхронні. При інтенсивному обміні інформацією постає питання максимально можливої частоти передачі при асинхронному зв'язку.

Розглянемо, як приклад, однофазну синхронізацію, схема якої приводиться на рис. 4.45, а. Вхідний сигнал з'являється в часовому інтервалі закінчення синхроімпульсу. В зв'язку з тим, що динамічні тригери DD1 і DD2

спрацьовують за спадом імпульсу, на вході DD3 синхроімпульсу з'явиться в один з інтервалів: або *A*, або *B*. Як витікає з часових діаграм (рис. 4.45, б), для появи на вході послідовності синхронних імпульсів необхідно, щоб виконувались умови:

$$T_{i\min} > T_T \text{ і } T_{\Pi\min} > T_T,$$

де  $T_{i\min}$  і  $T_{\Pi\min}$  – мінімальні тривалості імпульсу та паузи вхідної послідовності.

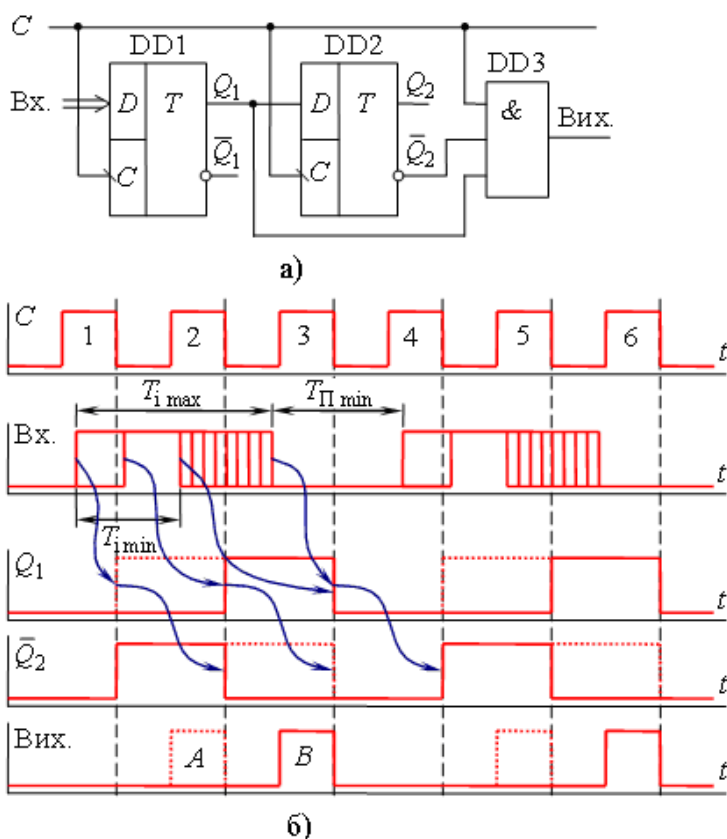


Рис. 4.45

Враховуючи властивість *D*-тригерів відновлювати свій стан при відсутності вхідного сигналу за спадом синхроімпульсу, з аналізу діаграм витікає, що при синхроімпульсному обміні частота передачі інформації зменшується практично в

2-3 рази, порівняно з синхронним. Це пояснюється тим, що при відносній швидкості обміну біт за такт, частота вихідних тактових сигналів падає



в 2,5-3 рази. До того ж, у вихідній послідовності буде порушена періодичність імпульсів, оскільки вони можуть з'являтися в інтервалі  $A$  або  $B$ , а це ускладнює задачу обробки інформації.

#### 4.7.4. Практичні проблеми проектування синхронних цифрових схем

Вище були описані засоби синхронізації як надійний метод побудови цифрових систем. У той же час, з підвищенням робочих частот та, відповідно, зменшенням допустимих інтервалів часу на передачу сигналу від одного тригера до іншого виникає ряд реальних проблем, які можна помітити лише в процесі налагодження. Одна з них – зміщення синхросигналу. Вона обумовлена тим, що синхронні динамічні тригери коректно запам'ятовують і передають інформацію лише тоді, коли всі вони одночасно сприймають вхідний сигнал за фронтом або за зрізом синхросигналу.

На рис. 4.46 приводиться приклад, коли внаслідок затримки проходження синхросигналу по провіднику друкованої плати між тригерами DD1 та DD2 на виході елемента DD2 отримується хибний сигнал.

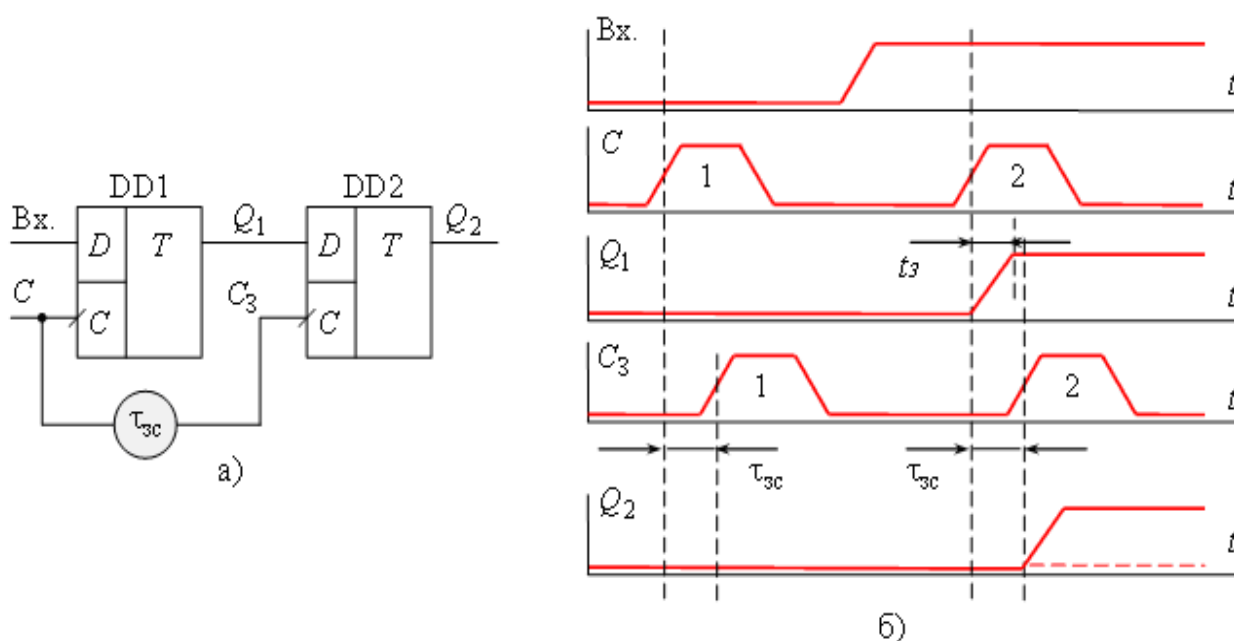


Рис. 4.46

Дійсно, за першим синхросигналом  $C$  низький рівень вхідного сигналу записується в тригер DD1, а в DD2 у цей час також записується низький рівень сигналу з  $Q_1$ . За другим синхросигналом у DD1 записується високий рівень вхідного сигналу, а в DD2 повинен записатися низький рівень з  $Q_1$ . Але внаслідок того, що по лінії передачі синхросигналу мала місце його затримка на величину  $\tau_{3c}$ , що перевищує реальну затримку завантаження тригера, в DD2 записується сигнал високого рівня. Величина  $\tau_{3c}$  називається *часом зміщення синхросигналу*. Фактично наявність  $\tau_{3c}$  еквівалентна появі завади в синхронних схемах. З рис. 4.46 бачимо, що для реально проєктованих схем необхідно, щоб виконувалась умова  $\tau_{3c} < t_3$ .

Звичайно, розглядаючи приведений приклад ізольовано від цифрової схеми, може виникнути питання: “Чому схема спроектована так, що інформаційний сигнал проходить короткий шлях, а синхросигнал – довгий, в той час як вони повинні проходити майже разом?”. Особливість синхросигналів полягає у тому, що, на відміну від інформаційних, вони розповсюджуються від одного джерела по всій схемі електронної системи. Як результат, характер навантаження джерела синхросигналу суттєво відрізняється від характеру навантаження джерела інформаційних сигналів. Для того, щоб зменшити навантаження джерела синхросигналу, створюються копії синхросигналу за схемою, що приведена на рис. 4.47.

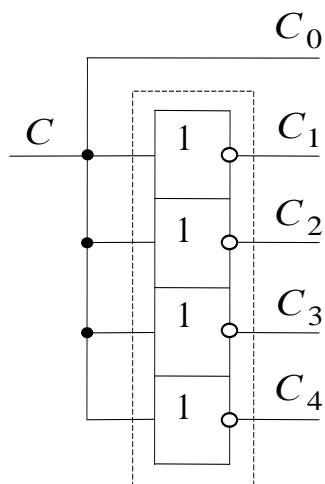


Рис 4.47

Всі копії синхросигналу  $C_1 \div C_4$  повинні створюватись ідентичними буферами, які мають однакові часові затримки і, здебільшого, розміщені в одній мікросхемі. Недопустимо використовувати поряд з копіями синхросигналів  $C_1 \div C_4$  базовий  $C_0$ , оскільки між ними вже існуватиме різниця в часі.

Але копії  $C_1 \div C_4$  необхідно використовувати ідентично, оскільки різне їх навантаження може привести до різних за величиною часових затримок буферних мікросхем.

При проектуванні друкованих плат слід звернути увагу на характер розводки провідників синхросигналу відповідними комп'ютерними програмами. Здебільшого програми орієнтовані на мінімізацію перехідних отворів, внаслідок чого провідники синхросигналу розміщуються по складній конфігурації між корпусами мікросхем. Оскільки для програм розводки друкованих плат всі провідники однакові, то провідники синхросигналу можуть розміщуватись паралельно заземленню, створюючи цим значні ємності і, відповідно, величини затримок. Як наслідок, різні мікросхеми матимуть різні величини затримок, що в складних і швидкодіючих цифрових системах може призвести до появи помилок. Тому в таких ситуаціях розводку провідників синхросигналу не слід довіряти програмам, а виконувати вручну або закладати, якщо це можливо, в програми розробки друкованих плат.

Окрім синхросигналів, у цифрових системах існує ще ряд сигналів, на які слід звертати особливу увагу при проектуванні. Це можуть бути сигнали дозволу, які необхідно подавати до подачі синхросигналу, а також інші керуючі сигнали. Слід також звертати особливу увагу на введення у цифрову систему асинхронних сигналів. Асинхронними є сигнали вводу інформації (наприклад, з клавіатури), сигнали переривань, а також ряд внутрішніх сигналів, що з'являються в результаті виконання обчислень (ознаки). Зрозуміло, що для вводу цих сигналів у синхронну цифрову систему використовуються синхронізатори, які забезпечують вибір асинхронного сигналу в тактовий момент часу. Проблема синхронізації обумовлена розглянутими вище затримками синхросигналу. Якщо, наприклад, асинхронний сигнал одночасно подається на декілька тригерів, то можлива ситуація, коли в один з тригерів інформація буде записана, а в інший, внаслідок затримки синхросигналу, – ні. В результаті в роботі системи виникне помилка.

## КОНТРОЛЬНІ ПИТАННЯ

1. Наведіть приклади метастабільності, які мають місце у повсякденному житті.
2. Поясніть особливості роботи *RS*-тригерів на основі логічних елементів **АБО-НІ**.
3. У чому полягає різниця між *RS*-тригерами, виготовленими на основі логічних елементів **АБО-НІ** та **І-НІ**.
4. Наведіть часові параметри, які характеризують тригерні схеми.
5. Які входи тригерів Вам відомі? Поясніть їх призначення.
6. Дайте пояснення таким параметрам тригера, як “*час підготовки*” та “*час підтримки*”.
7. У чому полягає особливість *MS*-тригерів? Які типи *MS*-тригерів Вам відомі?
8. У чому полягає різниця між тригерами зі статичним та з динамічним керуванням?
9. Дайте пояснення особливостям роботи *D*-тригера.
10. Поясніть переваги шестиелементного *D*-тригера.
11. Поясніть переваги *JK*-тригера перед *RS*-тригером. Які схемотехнічні особливості *JK*-тригера забезпечують ці переваги?
12. Сформулюйте алгоритм функціонування *JK*-тригера.
13. Сформулюйте алгоритм функціонування *T*-тригера.
14. Поясніть, чому можливе взаємне перетворення тригерів.
15. У чому полягають особливості несиметричних тригерів? Яку іншу назву має цей різновид тригерів?
16. Поясніть призначення синхронізації у цифрових системах високого рівня складності. Які види синхронізації використовуються у цифрових системах?
17. Які шляхи і засоби використовуються для вирішення проблеми затримок у складних цифрових системах?

18. Сформулюйте основні правила, яких необхідно дотримуватись при заведенні зворотних зв'язків у цифрових схемах.

19. У чому полягають переваги багатофазних способів синхронізації перед однофазним?

20. Які вимоги до вхідних сигналів пред'являються цифровими системами?

21. У чому проявляється необхідність врахування затримок і перехідних процесів при проектуванні цифрових пристроїв?

### ВПРАВИ І ЗАВДАННЯ

1. Зобразити вихідний сигнал  $RS$ -тригера на елементах **АБО-НІ** для вхідних сигналів, зображених на рис. 4.48. Прийняти наступні припущення: тривалість вихідних фронтів і спадів дорівнює нулю; середня затримка вентиля 10 нс; кожен інтервал часу також дорівнює 10 нс; початковий стан тригера – нульовий.

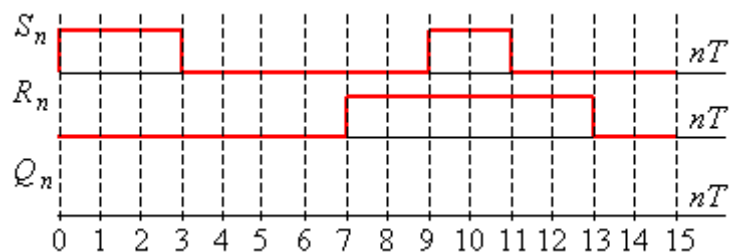


Рис. 4.48

2. Повторити вправу 1, використовуючи вхідні сигнали відповідно до рис. 4.49.

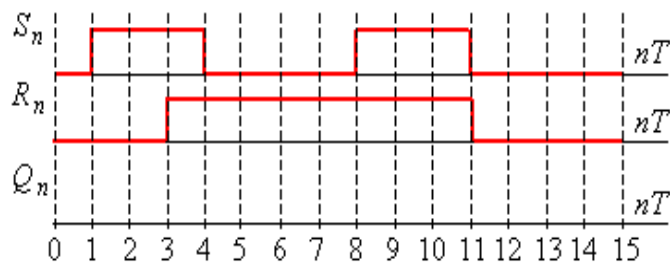


Рис. 4.49

3. Побудувати  $RS$ -тригер, використовуючи один елемент –  $D$ -тригер з синхронізацією фронтом типу КР1533ТМ2 (74ALS74А), без застосування інших компонентів.

4. Використовуючи елемент КР1533ТВ15 (74ALS109А) ( $JK$ -тригер з синхронізацією фронтом), побудувати тригер, еквівалентний КР1533ТМ2 (74ALS74А) ( $D$ -тригер), без застосування інших компонентів.

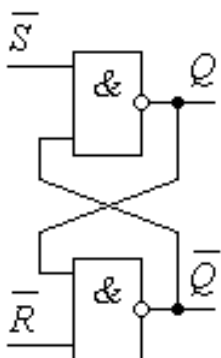


Рис. 4.50

5. Використовуючи елемент КР1533ТМ2 (74ALS74А) ( $D$ -тригер з синхронізацією фронтом), побудувати тригер, еквівалентний КР1533ТВ15 (74ALS109А), без застосування інших компонентів.

6. На рис. 4.50 приведена схема  $RS$ -тригера на логічних елементах **2І-НІ**. Виконати наступні завдання:

- а) використовуючи допоміжні логічні елементи, розробити схему синхронного  $RS$ -тригера;
- б) використовуючи допоміжні ЛЕ, розробити схему синхронного  $RS$ -тригера з прямими входами  $S$ - і  $R$ -;
- в) використовуючи допоміжні ЛЕ, розробити схему розширювача тривалості імпульсу високого рівня;
- г) використовуючи допоміжні ЛЕ, розробити схему розширювача тривалості паузи імпульсної послідовності.

7. На рис. 4.51 приводяться часові діаграми сигналів, що подаються на входи  $S$ - і  $R$ -  $RS$ -тригера, виготовленого на ЛЕ **2АБО-НІ**. Побудувати часову діаграму сигналу на прямому виході тригера.

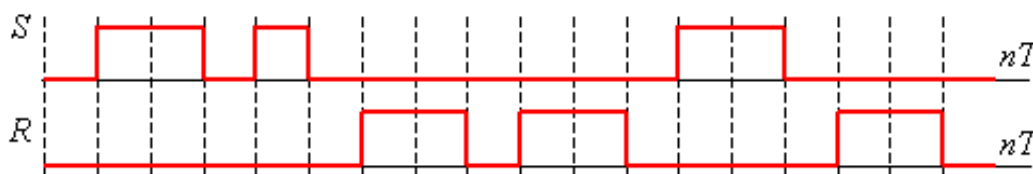


Рис. 4.51

8. Використовуючи  $RS$ -тригери і допоміжну логіку, розробити схему пристрою для зчитування і запам'ятовування (за керуючими сигналами  $Rd$  та  $Wr$  відповідно)  $\frac{1}{2}$  байта інформації, що подається у паралельному форматі.

9. Послідовність синхроімпульсів подається одночасно на групу статичних синхронних тригерів, з'єднаних відповідно до рис. 4.52.

Виконати наступні завдання: а) привести часові діаграми сигналів на кожному з виходів елементів  $Q_1$ ,  $Q_2$ ,  $Q_2$  за умови, що тривалість синхросигналу

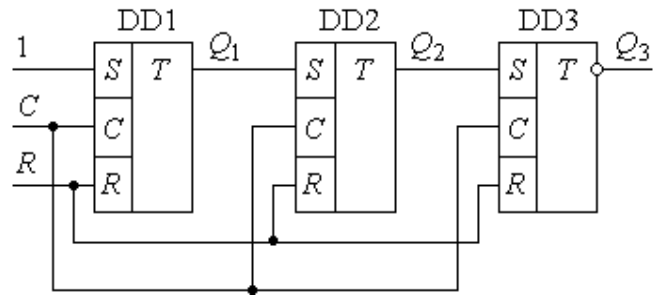


Рис. 4.52

набагато перевищує інтервал часу затримки на розповсюдження при нульових початкових значеннях  $Q_i$ ; б) побудувати часові діаграми сигналів на виходах  $Q_i$  за умови, що всі тригери динамічні і сприймають синхросигнал за його фронтом; в) побудувати часові діаграми сигналів на виходах  $Q_i$  за умови, що всі тригери динамічні і сприймають синхросигнал за його зрізом.

10. Поясніть, чи можливо використати  $RS$ -тригер в якості фазового детектора. (Фазовий детектор – це пристрій, який має залежність вихідної напруги від різниці фаз двох сигналів, що подаються на його входи). Якщо Ви вважаєте, що це неможливо, то дайте пояснення. Якщо можливо, то зобразіть характеристику вихідної напруги від різниці фаз.

11. На рис. 4.53 приводиться принципова схема цифрового пристрою з синхронними статичними  $RS$ -тригерами. Виконати наступне: а) пояснити роботу пристрою; б) побудувати часові діаграми напруг на виходах  $Q_1 \dots Q_4$  при послідовній подачі чотирьох синхросигналів; в) записати логічні функції дешифратора для перетворення вихідного коду, який знімається з  $Q_1 \dots Q_4$ , у двійковий.

12. На рис. 4.54 приводиться схема двійкового лічильника від **0** до **4**. При подачі чергового імпульсу сигнал високого рівня з'являється на відповідному виході. П'ятий імпульс приводить до обнуління схеми. Проаналізувати роботу лічильника. Побудувати часові діаграми сигналів на виходах логічних елементів  $DD1 \dots DD5$ ; на виходах тригерів  $Q_1 \dots Q_4$ .

13. Робота  $RS$ -тригера описується формулою (4.1). Довести, що,

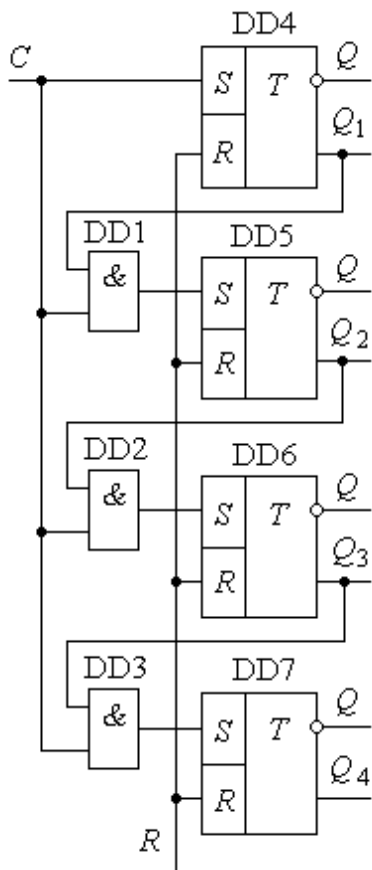


Рис. 4.53

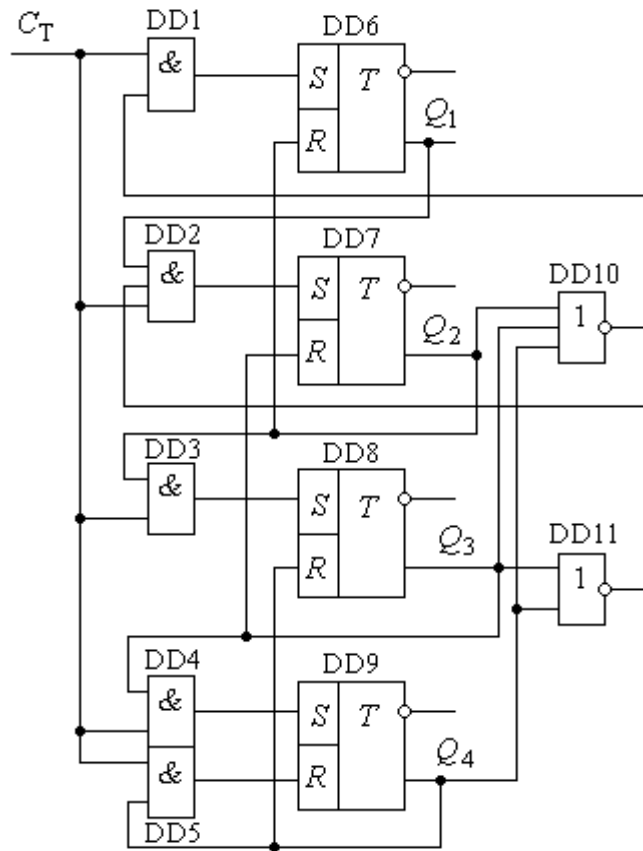


Рис. 4.54

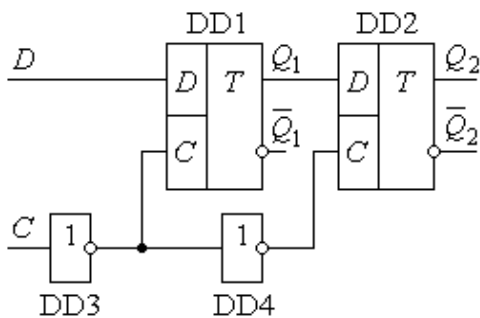


Рис. 4.55

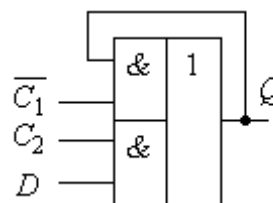


Рис. 4.56

використовуючи допоміжний інвертор DD1 (рис. 4.16, а), матимемо  $D$ -тригер, робота якого описується формулою (4.6).

14. На рис. 4.55 приведена схема двотактового динамічного тригера. Використовуючи часові діаграми, пояснити роботу тригера з урахуванням часових затримок на перемикання.



15. На рис. 4.56 приведена схема *D*-тригера, в якому запис інформації забезпечується синхросигналом  $C_2$ , а його обнуління – сигналом  $\overline{C_1}$ . Привести часові діаграми і пояснити його роботу.

16. Використовуючи алгоритм функціонування *RS*-тригера, розробити алгоритм функціонування пристрою, схема якого приведена на рис. 4.57, відносно входів  $x_1$  та  $x_2$ . Привести часові діаграми роботи пристрою.

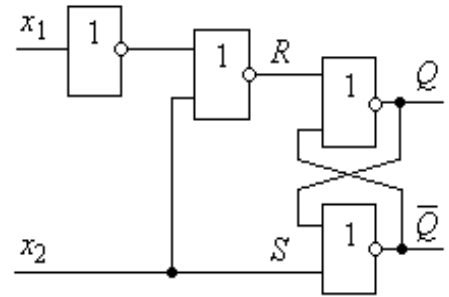


Рис. 4.57

17. Аналогічно до вправи 16, розробити алгоритм функціонування пристрою, схема якого приведена на рис. 4.58. Привести часові діаграми роботи пристрою.

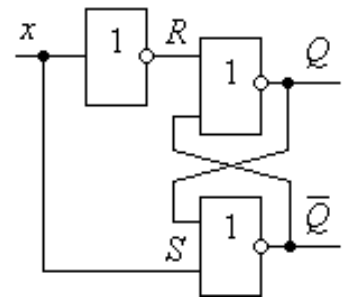


Рис. 4.58

18. Використовуючи синхронний *RS*-тригер і допоміжну логіку, розробити схему синхронного *D*-тригера.

19. Чи можливо в *RS*-тригерах поміняти місцями входи *R*- і *S*-? Проаналізувати мікросхему з *RS*-тригерами (КР1533ТР2 (2S279)); пояснити, що зміниться у роботі мікросхеми.

Таблиця 4.13

$x_1$	$x_0$	$Q_{n+1}$
0	0	0
0	1	$Q_n$
1	0	$\overline{Q_n}$
1	1	1

20. Пояснити, як забезпечується метастабільність у *D*-тригерах, коли відсутні сигнали установки і підтримки. Проаналізувати дію кола зворотного зв'язку всередині схеми.

Таблиця 4.14

$x_1$	$x_0$	$Q_{n+1}$
0	0	0
0	1	$Q_n$
1	0	$\overline{Q_n}$
1	1	0

21. Робота *D*-тригера Ерла описується формулою:

$$Q_{n+1} = \overline{C_n} \overline{Q_n} \cdot \overline{C_n} \overline{D_n} \cdot \overline{D_n} Q_n$$

Побудувати схему тригера у базисі **I-НІ**, пояснити.

22. Розробити тактований фронтом двовходовий тригер, що функціонує відповідно до табл. 4.13, на основі *D*-тригера.

23. Розробити синхронний двовходовий тригер, що функціонує відповідно до табл. 4.14, на базі універсального  $JK$ -тригера.

24. Побудувати граф-схему  $JK$ -тригера. Пояснити, чим вона відрізнятиметься від граф-схеми  $RS$ -тригера.

25. На рис. 4.59 приведена схема тригера. Визначити тип тригера, пояснити принцип його роботи.

26. На рис. 4.60 приведена схема тригера. Побудувати таблицю станів і описати алгоритм його функціонування по відношенню до сигналів  $x_0$ ,  $x_1$ .

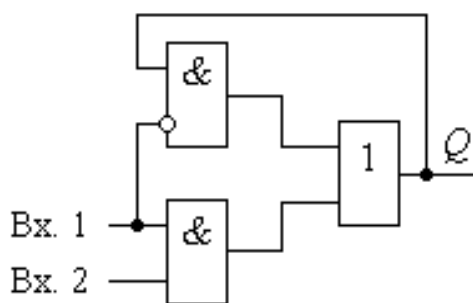


Рис. 4.59

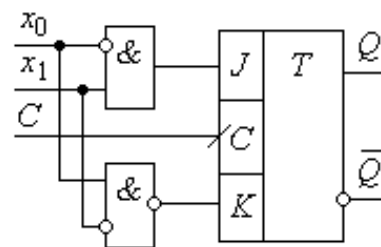


Рис. 4.60

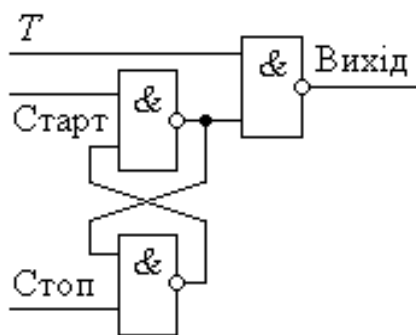


Рис. 4.61

27. На рис. 4.61 приведена схема старт-стопного пристрою. Виходячи з умови, що тривалість імпульсів “Старт” і “Стоп” достатня для зміни стану тригера, а період  $T$  набагато менший інтервалу часу між подачею сигналів “Старт” і “Стоп”, побудувати часові діаграми роботи пристрою.

28. Проаналізувати особливості роботи старт-стопних пристроїв, схеми яких приведені на рис. 4.62,  $a - б$ .

29. На рис. 4.63 приводяться схеми старт-стопних пристроїв. Проаналізувати особливості роботи пристроїв при різних часових співвідношеннях між сигналами “Старт” і “Стоп”.

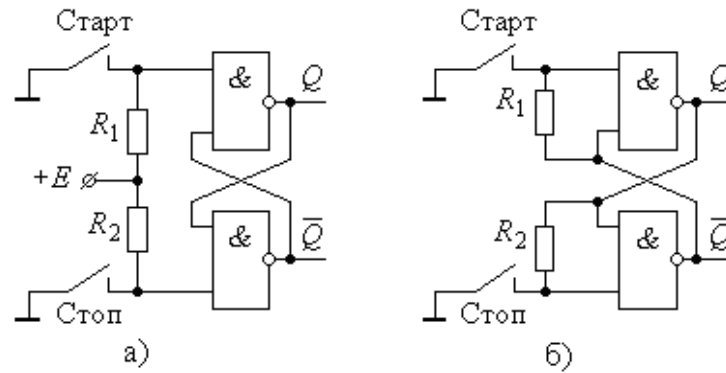


Рис. 4.62

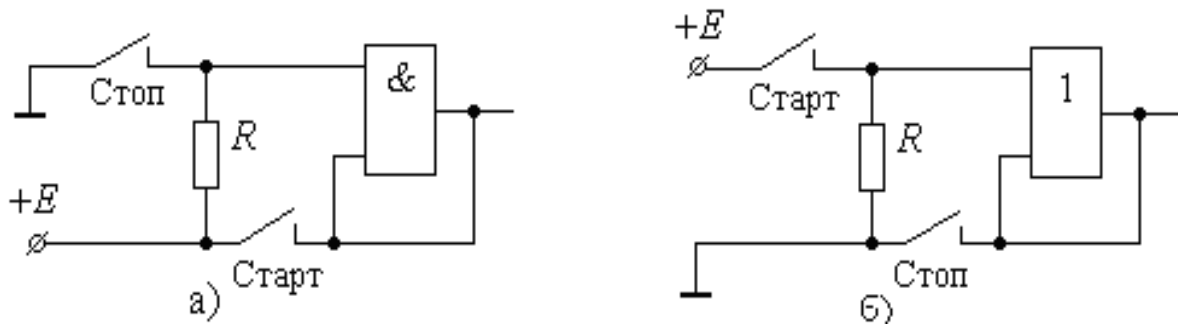


Рис. 4.63

30. Розробити варіанти схем старт-стопних пристроїв з використанням синхронних  $D$ -тригерів;  $JK$ -тригерів.

31. На входи  $D$  і  $C$   $D$ -тригера подаються прямокутні імпульси з близькими частотами (наприклад, 90 Гц і 100 Гц). Пояснити особливості роботи тригера. Навести приблизний вигляд часових діаграм імпульсів на виході. Оцінити кількість імпульсів на виході тригера за інтервал 1 секунда.

32. На рис. 4.64,  $a$  приведена схема на основі двох  $D$ -тригерів, на входи якої поступають сигнали  $T$  і  $C$  відповідно до часової діаграми, приведеної на рис. 4.64,  $b$ . Пояснити роботу пристрою і його призначення. Побудувати часові діаграми сигналів на виходах  $Q_1$  і  $Q_2$  тригерів і на виході пристрою  $Y$  без урахування часових затримок на перемикання тригерів. Який вплив матимуть часові затримки на роботу схеми? Як зміняться часові діаграми, якщо тригер DD2 спрацюватиме не за зрізом, а за фронтом сигналу генератора  $T$ ?

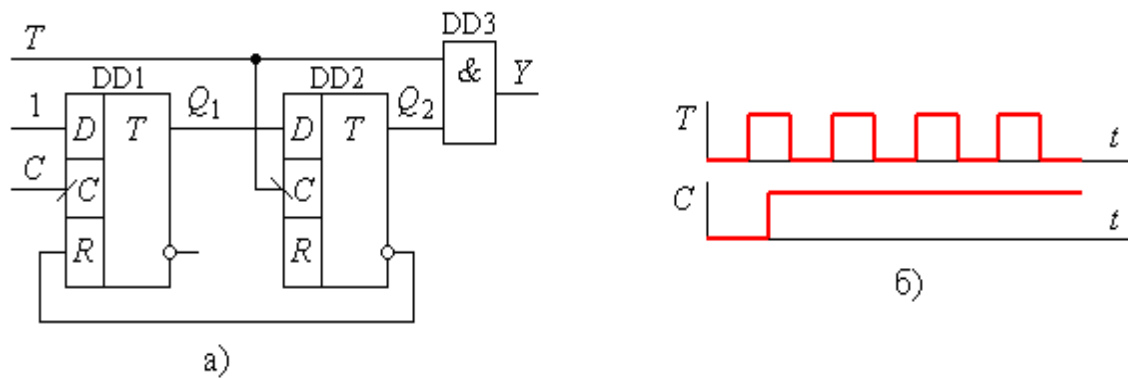


Рис. 4.64

33. Пояснити, як побудувати  $T$ -тригер, використовуючи  $D$ -тригер і, якщо потрібно, допоміжну логіку.

34. Пояснити, як побудувати  $JK$ -тригер, використовуючи  $T$ -тригер і, якщо потрібно, допоміжну логіку.

35. Використовуючи  $D$ -тригер і допоміжну комбінаційну логіку, розробити схему  $T$ -тригера з допоміжним дозволяючим входом.

36. Використовуючи  $T$ -тригер і допоміжну комбінаційну логіку, розробити схему  $D$ -тригера з допоміжним дозволяючим входом.

37. На рис. 4.65 приводяться часові діаграми  $S$ - і  $R$ - сигналів, що діють на

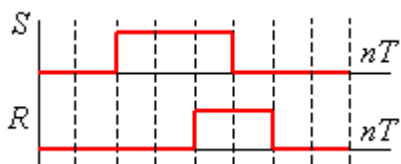


Рис. 4.65

$RS$ -тригер, виготовлений з використанням логічних елементів АБО-НІ. Побудувати часові діаграми сигналу на прямому виході  $Q$  за умови, що у момент часу  $t = 0$  стан виходу  $Q_n = 0$ .

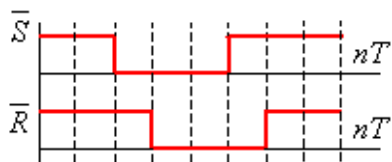


Рис. 4.66

38. На рис. 4.66 приводяться часові діаграми сигналів, що прикладаються до входів  $\bar{S}$  і  $\bar{R}$   $RS$ -тригера, виготовленого з використанням ЛЕ І-НІ. Побудувати часові діаграми сигналу на прямому виході  $Q$  за умови, що у момент часу  $t = 0$  стан виходу  $Q_n = 0$ .

## РОЗДІЛ 5

### СКІНЧЕННІ АВТОМАТИ

#### 5.1. Загальна характеристика скінченних автоматів

Розглянуті вище схеми тригерів та пристроїв на їх основі передбачають наявність у тригері двох стійких станів. Реакція пристрою на вхідні логічні сигнали залежить від того, в якому стані знаходиться тригер. Але існує велика кількість різноманітних схем, в яких використовується декілька тригерів. У такому випадку кількість внутрішніх станів схеми зростає в пропорції  $2^m$ , де  $m$  – кількість тригерів. При цьому, зрозуміло, зростає складність “реакції” такого пристрою на вхідні сигнали, оскільки кожного разу внутрішні стани тригерів можуть бути іншими.

Цифрові пристрої з  $m$  тригерами ( $m > 1$ ), стан виходів яких залежить не тільки від значень вхідних сигналів у даний момент часу, а й від стану використовуваних тригерів у даний та попередні моменти часу, називаються *цифровими автоматами*.

У загальному плані *цифровим автоматом* називається пристрій, який формує ряд вихідних дискретних сигналів у відповідності до вхідної бінарної комбінації сигналів. Найпростішим з таких автоматів є перетворювач кодів, і він називається *комбінаційним*.

У подальшому розглядатимуться автомати з пам'яттю, завдяки якій значення виходів залежать не лише від вхідних сигналів, що надходять у визначений момент часу, а також від внутрішнього стану автомата. Оскільки кількість внутрішніх станів залежить від кількості використовуваних тригерів, то цифрові автомати поділяються на *скінченні*, що мають обмежену кількість станів, і автомати з *нескінченною кількістю станів*, або *послідовнісні машини*. До останніх відносяться ЕОМ.

Будь-який цифровий автомат є сукупністю елементів пам'яті (тригерів) та комбінаційних схем. Тригери містять інформацію про особливості попередньої

роботи автомата. Комбінаційні схеми на основі вхідних сигналів та інформації, яка береться з тригерів, формують вихідні сигнали і сигнали для формування нових станів тригерів. Таким чином, однією з особливостей цифрових автоматів є те, що вони мають свої внутрішні стани, від яких залежить реакція на вхідні сигнали. Одним з найпростіших прикладів цифрових автоматів є кодовий замок, реакція на черговий сигнал якого залежить від попередніх сигналів.

Скінченні автомати можуть бути *синхронними*, зміна станів яких відбувається в тактові моменти часу, що задаються зовнішнім генератором, а також *асинхронними* - зміна станів яких відбувається внаслідок дії вхідних сигналів практично без затримки. Асинхронні автомати вважаються більш швидкодіючими і знаходять використання у швидкодіючих інформаційних пристроях вимірювання і керування різноманітними процесами, де необхідна миттєва реакція на зміну вхідних сигналів.

Синхронні автомати в силу специфіки своєї роботи вносять у процес вимірювання або керування затримку, що визначається величиною періоду синхросигналу.

Здебільшого асинхронні автомати будуються на основі асинхронних елементів пам'яті – асинхронних тригерів. Синхронні автомати будуються з використанням синхронних тригерів.

*Математичною моделлю* цифрового автомата є абстрактний автомат, в якому враховуються вхідні та вихідні сигнали, а також внутрішні стани. Будемо розглядати детерміновані автомати, які завжди мають початковий стан  $Q_0$ , з якого вони починають працювати при дії вхідних сигналів і при повторі перебору вхідних сигналів повторюють послідовність станів і вихідних сигналів.

Абстрактний автомат задається *множиною внутрішніх станів*  $Q = \{ Q_1, Q_2, \dots, Q_M \}$  (*алфавітом станів*), *множиною вхідних сигналів*  $X = \{ X_1, X_2, \dots, X_P \}$  (*вхідним алфавітом*), *множиною вихідних сигналів*

$Y = \{ Y_1, Y_2, \dots, Y_k \}$  (вихідним алфавітом) і початковим станом  $Q_0$ . Перехід з одного стану в інший визначається функцією переходів  $f_p$ , що визначає стан автомата  $Q_s$ , в який він переходить з попереднього стану  $Q_m$  при дії сигналу  $X_p$ :

$$Q_s = f_p(Q_m, X_p).$$

Значення виходів автомата задається функцією виходів  $\lambda$ , що залежить від стану автомата  $Q_m$  і вхідного сигналу  $X_p$ :

$$Y_k = \lambda(Q_m, X_p).$$

Абстрактний автомат працює в дискретному часі, який задається цілими позитивними числами  $t = 0, 1, 2, \dots = n$ .

У кожний момент дискретного часу  $t$ , який зветься *тактом*, автомат перебуває у деякому стані  $Q(t)$  з множини станів автомата  $Q$ .

У початковий момент часу ( $t = 0$ ) він завжди знаходиться в стані  $Q(0) = Q_0$ . Вважається, що реакція автомата на вхідні сигнали не залежить від інтервалів часу між тактовими моментами.

У момент  $t$ , знаходячись у стані  $Q(t)$ , автомат сприймає на своєму вході сигнали, які називаються *буквами (літерами) вхідного алфавіту*  $X_p \in X$ . У відповідності до функції переходів  $f_p$ , він перейде у стан  $Q_s(t)$ , що описується алфавітом станів, тобто  $Q_s(t) \in Q$ .

Аналогічно, у відповідності до функції виходів  $\lambda$ , на виході отримується сигнал  $Y_k$ , де  $Y_k \in Y$ .

**Приклад 5.1.** Розглянути RS-тригер як скінченний автомат.

*Розв'язання.* RS-тригер має два стани із множини станів  $Q$ :  $Q_1 = 0$  і  $Q_2 = 1$ , де  $Q_1, Q_2 \in Q$ . Автомат має чотири значення входів  $X_p$  з алфавіту входів  $X_1, X_2, X_3, X_4$ , де  $X_1 = \bar{S} \cdot \bar{R} = 00$ ,  $X_2 = S \cdot \bar{R} = 10$ ;  $X_3 = \bar{S} \cdot R = 01$ ;  $X_4 = S \cdot R = 11$ . Вихідної комбінаційної логіки автомат не має, тобто залежність  $Y_k = \lambda(Q_m, X_p)$  визначається формулою:  $Y = Q$ .

Скінченна множина букв вхідного алфавіту, вихідного алфавіту і станів називається *словами*. Для скінченного автомату кількість символів цих

алфавітів обмежена. Повний перебір символів вхідного слова повинен привести автомат до початкового стану  $Q_0$ . Автомат, який завжди починає працювати з початкового стану, називається *ініціальним*.

Поняття *станів* в описі автоматів пов'язане з необхідністю враховувати типи і характер попередніх сигналів, тобто таких, які діяли на один або декілька тактів раніше. Стани автомата і є тими відповідними елементами пам'яті, що характеризують попередні сигнали. Введення поняття станів дозволяє усунути час як явну змінну і визначати вихідний сигнал як функцію внутрішнього стану і входу в тактовий момент часу. Такий спосіб опису автоматів має великі переваги перед іншими для асинхронних пристроїв, в яких інтервали часу між сусідніми тактами можуть мати довільні значення, що значно відрізняються між собою. З такої точки зору комбінаційні пристрої відносяться до автоматів, в яких вихід не залежить від попередніх сигналів і повністю визначається комбінацією вхідних сигналів.

У практиці роботи автоматів часто мають місце випадки, коли деякі комбінації значень вхідних символів подавати неприпустимо (згадаємо таблицю станів *RS*-тригера). Такі комбінації є *забороненими* для автомата. Автомати, що мають заборонені вхідні слова, називаються *частковими*. Наприклад, деякий абстрактний автомат має вхідний алфавіт  $X$ , що складається з двох символів  $X_0$  і  $X_1$ , кожен з яких може приймати значення **0** або **1**. Слова  $\overline{X_1} \overline{X_0}$ ,  $\overline{X_1} X_0$ ,  $X_1 \overline{X_0}$  є *дозволеними*, а слово  $X_1 X_0$  – *забороненим*.

Наявність заборонених комбінацій вхідних символів зменшує кількість слів вхідного алфавіту, зменшує кількість станів і дозволяє будувати часткові автомати з меншими затратами, ніж повні, для яких вказаних обмежень не існує.

Функція  $Y_k = \lambda(Q_m, X_p)$  задає значення виходів автомата з алфавіту  $Y$ . Кількість символів алфавіту  $Y$  не співпадає з кількістю символів алфавіту  $X$ . За аналогією з перетворювачами кодів, автомат забезпечує відображення слів вхідного алфавіту в слова вихідного алфавіту. Якщо прийняти можливу



кількість слів вхідного алфавіту за  $P_x$ , вихідного алфавіту за  $K_y$ , то в залежності від співвідношення між  $P_x$  та  $K_y$  автомати можуть за функціональним призначенням розділятися на три групи.

При  $K_y < P_x$  маємо автомати, які призначені для розв'язання задач керування, стиснення інформації, розпізнавання повідомлень. Наприклад, цифровий кодовий замок може мати 3...5 вхідних символів і лише один вихідний. Інший приклад – розпізнавання введеного пароля при отриманні доступу до комп'ютера.

Якщо  $K_y > P_x$ , то маємо ситуацію, коли довжина вихідних слів буде більшою, ніж вхідних, оскільки кількість слів повинна бути однаковою. Вихідні слова будуть нести надмірну інформацію, що використовується, як відмічалось в попередніх розділах, для синтезу завадостійких кодів.

При  $K_y = P_x$  кількість і довжина вхідних і вихідних слів однакові. Такі автомати використовуються для деяких кодових перетворень (наприклад, перетворення двійкового коду в код Грея, і т. д.), для реалізації деяких методів захисту даних від несанкціонованого доступу (метод підстановок) і т. п.

## 5.2. Автомати Мілі і Мура

### 5.2.1. Закони функціонування автоматів

У практиці використання цифрових автоматів можна виділити невелику кількість типових алгоритмів їх функціонування. Найбільшого розповсюдження набули два типи автоматів – *автомати Мілі* і *Мура*.

Закон функціонування автомата Мілі задається рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y(t) = \lambda [Q(t), X(t)]; \end{cases} \quad t = 0, 1, 2, \dots \quad (5.1)$$

Закон функціонування автомата Мура описується рівняннями:

$$\begin{cases} Q(t+1) = f_p [Q(t), X(t)]; \\ Y(t) = \lambda [Q(t)]; \end{cases} \quad t = 0, 1, 2, \dots \quad (5.2)$$

Різниця між автоматами Мілі і Мура полягає лише в тому, що вихідний сигнал останнього залежить тільки від його внутрішнього стану, у той час як у автомата Мілі значення вихідного сигналу залежить також і від вхідного сигналу.

Структура автомата Мілі зображена на рис. 5.1.

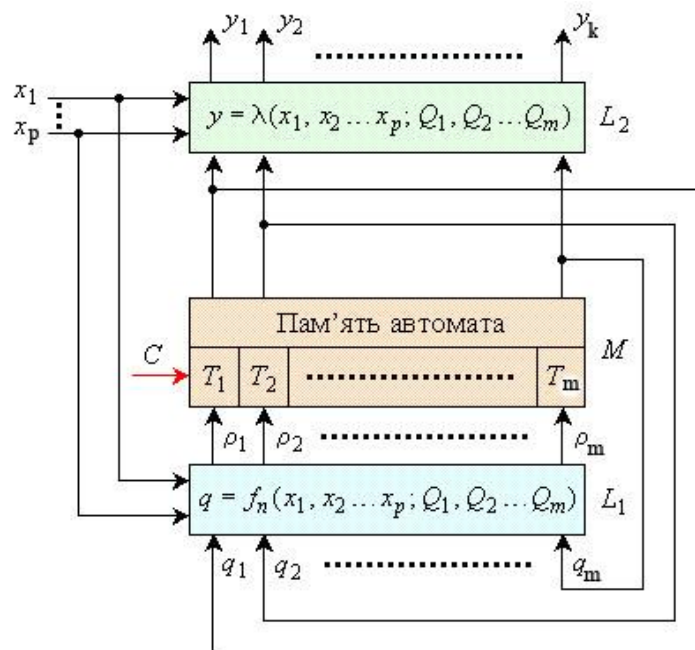


Рис.5.1

Автомат складається з набору елементарних комірок пам'яті (тригерів  $T_1, T_2, \dots, T_m$ ), стани яких  $q_1, q_2, \dots, q_m$  в кожний момент часу визначають внутрішній стан автомата в цілому. Під дією вхідних сигналів  $x_1 \dots x_p$ , які подаються в дискретні моменти часу, відбувається формування сигналів  $\rho_1 \dots \rho_m$ , що забезпечують перемикання тригерів  $T_1 \dots T_m$ . Таким шляхом відбувається послідовна зміна станів автомата. Оскільки комбінаційна схема  $L_1$  формує керуючі тригерами сигнали в залежності від значень виходів тригерів у момент подачі тактового сигналу (для синхронних схем) або в моменти подачі вхідних сигналів (для асинхронних), то значення виходів тригерів у наступний момент часу  $i$ , відповідно, стан автомата однозначно визначається вхідними сигналами і станом автомата в попередній момент часу.

Сигнали керування тригерами  $\rho_1 \dots \rho_m$  формуються комбінаційним пристроєм  $L_1$ , структура якого визначає функцію переходів автомата. Функція виходів реалізується комбінаційним пристроєм  $L_2$ , який формує сигнали  $y_1 \dots y_k$  як функції виходів елементів пам'яті (для автоматів Мура) або як функції виходів елементів пам'яті і вхідних сигналів (для автоматів Мілі).

Переходячи до відносного часу роботи, формули (5.1), (5.2) можна зобразити у наступному вигляді:

- для автомата Мілі:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}, X_{n+1}); \\ Q_{n+1} = L_1(Q_n, X_n); \end{cases} \quad (5.3)$$

- для автомата Мура:

$$\begin{cases} Y_{n+1} = L_2(Q_{n+1}); \\ Q_{n+1} = L_1(Q_n, X_n). \end{cases} \quad (5.4)$$

Рівняння (5.3), (5.4) можуть бути заданими аналітично або у вигляді таблиць станів (таблиць відповідностей).

Взаємозв'язок між поточними  $Q_n$  і наступними  $Q_{n+1}$  станами визначається характеристичними рівняннями використовуваних тригерів.

У табл. 5.1 приведені характеристичні рівняння тригерів, розглянутих у попередніх розділах.

Таблиця 5.1

Тип використовуваного тригера	Характеристичне рівняння
Асинхронний <i>RS</i> - тригер	$Q_{n+1} = S_n + \bar{R}_n Q_n$
Синхронний статичний <i>D</i> -тригер	$Q_{n+1} = C_n D_n$
Динамічний <i>D</i> -тригер	$Q_{n+1} = D_n$
Динамічний <i>JK</i> -тригер ( <i>JK-MS</i> -тригер )	$Q_{n+1} = J_n \bar{Q}_n + \bar{K}_n Q_n$
<i>T</i> - тригер	$Q_{n+1} = C_n \bar{Q}_n$
<i>T</i> - тригер з дозволяючим входом	$Q_{n+1} = C_n (\bar{V} \bar{Q}_n + V Q_n)$

Поєднуючи характеристичні рівняння тригерів і рівняння комбінаційних схем, можна проводити аналіз роботи існуючого автомата або виконувати його синтез.

### 5.2.2. Способи опису роботи автоматів

В практиці аналізу і синтезу цифрових автоматів використовують різні способи опису їх роботи. Найбільш поширеними є *табличний* і *графічний* способи.

Розглянемо спочатку опис роботи автомата з використанням таблиць переходів та виходів. Стовпці (рядки) цих таблиць позначають символами з множини  $Q$ , а рядки (стовпці) – символами з множини  $X$ .

Кількість рядків таблиці переходів визначається кількістю комбінацій вхідних сигналів  $P$ , а кількість стовбців – відповідно, кількість станів  $M$  автомата.

У табл. 5.2 зображена таблиця автомата з  $P = 3$ ,  $M = 2^m = 4$ .

В кожній клітині таблиці переходів записується наступний стан, в який переходить автомат з попереднього стану, що стоїть у заголовку стовпця при дії відповідного вхідного сигналу. Так, наприклад, якщо автомат знаходиться у стані  $Q_0$ , то при дії сигналу  $X_1$  він перейде в стан  $Q_3$ ; при дії сигналу  $X_2$  залишиться в стані  $Q_0$ , а при дії сигналу  $X_3$  перейде в стан  $Q_1$ .

Таблиця 5.2

$Q \backslash X$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$X_1$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
$X_2$	$Q_0$	$Q_3$	$Q_2$	$Q_1$
$X_3$	$Q_1$	$Q_0$	$Q_3$	$Q_2$

Таблиця 5.3

$Q \backslash X$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$X_1$	$Q_3$	$Q_2$	$Q_1$	$Q_0$
$X_2$	$Q_0$	$Q_3$	$Q_2$	$Q_1$
$X_3$	$Q_1$	$Q_0$	$Q_3$	$Q_2$

Таблиця 5.4

	$Y_2$	$Y_1$	$Y_0$
$Q \backslash X$	$Q_0$	$Q_1$	$Q_2$
$X_1$	$Q_2$	$Q_1$	$Q_2$
$X_2$	$Q_0$	$Q_2$	$Q_1$
$X_3$	$Q_1$	$Q_0$	$Q_0$

Таблиця виходів (табл. 5.3) відрізняється від таблиці переходів лише тим, що у кожній клітині записується відповідне значення вихідного сигналу автомата.

Таблиці переходів і виходів автомата Мілі можуть бути представлені у вигляді однієї об'єднаної таблиці, у клітинах якої вказані значення як станів, так і виходів.

Функції переходів і виходів автомата Мура задаються однією таблицею переходів, яка будується так само, як і таблиця переходів автомата Мілі. Різниця полягає лише в тому, що над заголовками кожного стовпця встановлюється окремим рядком значення виходів автомата (табл. 5.4).

Для частково заданих автоматів, у яких функції виходів і функції переходів визначені не для всіх комбінацій  $Q_m$  і  $X_p$ , відповідні клітини залишаються незаповненими.

Як приклад, розглянемо більш детально автомати, що описуються табл. 5.2 – табл. 5.4.

При двійкових вхідних сигналах входи автоматів  $X_1$ ,  $X_2$ ,  $X_3$  можуть бути задані вхідними логічними змінними  $x_1$  та  $x_2$ , тобто з множини  $P=2$ . В алфавіті  $X$  матимемо  $2^P=4$  різних слів, а саме:  $X_1 = \bar{x}_1 \bar{x}_2$ ,  $X_2 = x_1 \bar{x}_2$ ,  $X_3 = \bar{x}_1 x_2$ ,  $X_4 = x_1 x_2$ . Звідси витікає, що  $X_4 = x_1 x_2$  – заборонене вхідне слово для автоматів, що розглядаються. Але така заборона може існувати не для всіх станів автомата, а тільки для деяких. Наочний приклад такого автомата приводиться в, що заданий таблицею переходів табл. 5.5, з якої витікає, що при стані  $Q_0$  на вхід автомата не повинен надходити сигнал  $X_2$ , оскільки перехід у такому випадку не визначений.

Можна прийняти, що  $X_1 = x = 1$ , а  $X_2 = \bar{x} = 0$  і, відповідно, лише значення  $X_1 = 1$  може змінювати стан  $Q_0$  автомата. З табл. 5.5 однозначно визначається і послідовність зміни станів автомата при дії  $X_1$ :  $Q_0 \Rightarrow Q_1 \Rightarrow Q_2 \Rightarrow Q_0$ . Для цього необхідно задати вхідну послідовність  $X_1, X_1, X_1$ , при цьому автомат закінчує

свою роботу переходом до початкового стану. Якщо функція  $\lambda = 1$ , то виходи автомата Мілі одночасно визначатимуться значеннями його внутрішніх станів.

Таблиця 5.5

$X \backslash Q$	$Q_0$	$Q_1$	$Q_2$
$X_1$	$Q_1$	$Q_2$	$Q_0$
$X_2$	—	$Q_2$	$Q_1$

В тому випадку, коли  $X_1$  є тактовим сигналом, що діє лише на тригери, а стани автомата змінюються упорядковано в зростаючому або спадаючому напрямку, автомат називається *лічильним автоматом*, або *лічильником*.

Більш наочним є спосіб опису автоматів за допомогою графів, подібно до того, як описувалася робота тригерів. Різниця полягає в тому, що автомат може мати суттєво більшу кількість станів. На рис. 5.2 показані граф-схеми автоматів Мілі і Мура, які задані табл. 5.2 – 5.3 (рис.5.2.a), табл.5.4 (рис.5.2.б).

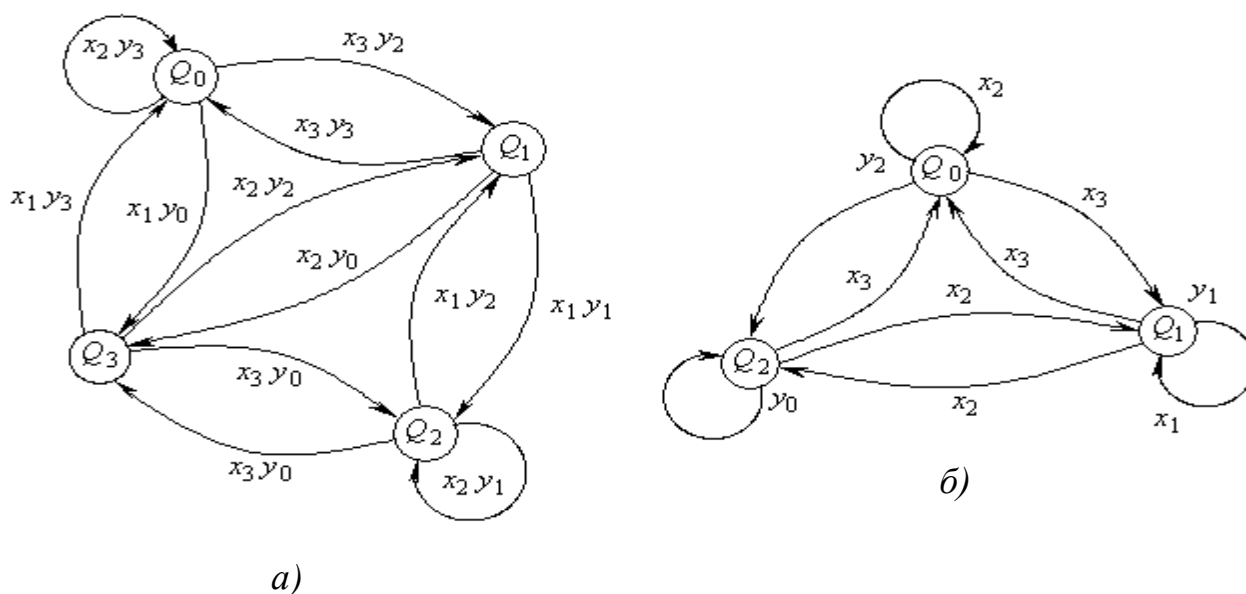


Рис.5.2

Граф-схеми широко використовуються як при аналізі, так і при синтезі автоматів, а також при переході від словесного до формалізованого їх опису.

За допомогою таблиць переходів і виходів, як і за допомогою граф-схеми завжди можна знайти вихідну реакцію автомата на будь-яке вхідне слово, що належить множині  $X$ .

Робота автомата може описуватися у часі за допомогою спеціальної табл. 5.6, яка називається *стрічкою цифрового автомата*.

Таблиця 5.6

Такт	0	1	2	3	4	5	6	7	8	9	10
Вхідний сигнал	$X_2$	$X_1$	$X_1$	$X_2$	$X_3$	$X_1$	$X_1$	$X_3$	$X_2$	$X_1$	$X_1$
Стан	$Q_0$	$Q_2$	$Q_3$	$Q_0$	$Q_2$	$Q_0$	$Q_3$	$Q_1$	$Q_2$	$Q_3$	$Q_3$
Вихідний сигнал	$Y_0$	$Y_1$	$Y_2$	$Y_0$	$Y_0$	$Y_1$	$Y_2$	$Y_2$	$Y_1$	$Y_1$	$Y_2$

Особливість такої стрічки полягає в тому, що для будь-якої пари сусідніх тактів  $i$  та  $(i+1)$  можна виділити четвірку символів (виділена у табл. 5.6 жирною лінією), яка показує, в який стан перейде цифровий автомат в  $(i+1)$ -ому такті і який вихідний сигнал буде сформований під дією вхідного сигналу.

Однією з форм зображення автомата є його *дерево переходів і виходів*. Дерево може мати декілька ярусів, у кожному з яких за допомогою гілок показуються можливі переходи, починаючи з нульового стану.

На рис. 5.3 приводиться приклад дерева переходів, що відповідає табл. 5.2.

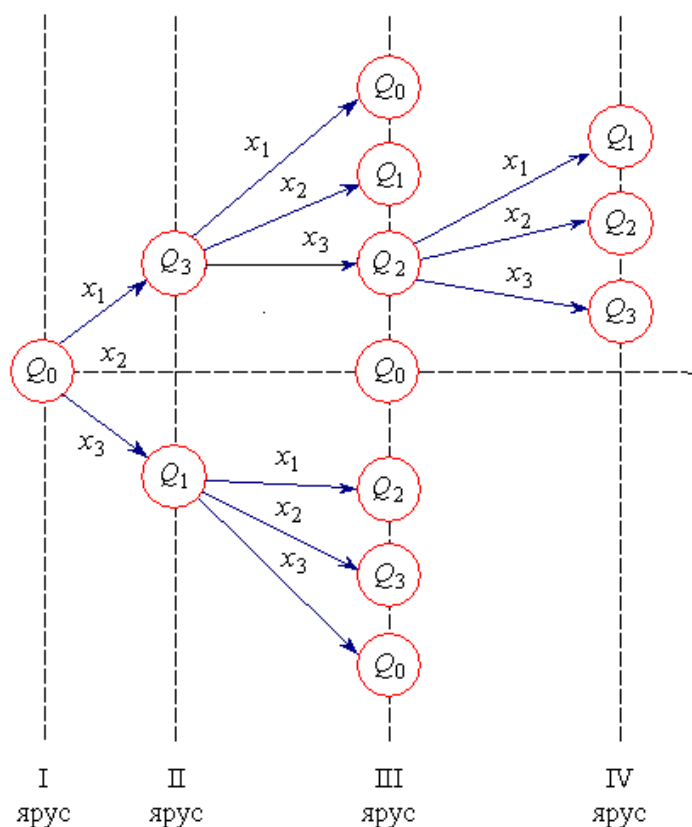


Рис.5.3

### 5.2.3. Еквівалентні перетворення автоматів

Для будь-якого автомата Мілі можна побудувати відповідний автомат Мура, і навпаки.

Розглянемо алгоритми таких переходів на прикладі автомата Мура, що заданий табл. 5.7 і табл. 5.8.

Таблиця 5.7

$X \backslash Q$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$x_1$	$Q_1$	$Q_2$	$Q_0$	$Q_2$
$x_2$	$Q_3$	$Q_0$	$Q_0$	$Q_0$
$x_3$	$Q_2$	$Q_3$	$Q_2$	$Q_3$

Таблиця 5.8

$X \backslash Q$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$x_1$	$Y_3$	$Y_2$	$Y_1$	$Y_1$
$x_2$	$Y_2$	$Y_1$	$Y_3$	$Y_3$
$x_3$	$Y_1$	$Y_2$	$Y_2$	$Y_3$

Поставимо у відповідність кожній парі  $Q_m$  та  $X_p$  автомата Мілі стан  $Q_{mp}$  автомата Мура. До множини станів автомата Мура включимо початковий стан автомата Мілі  $Q_0$ , але позначимо його  $Q_{00}$ . Для прикладу, що розглядається, така відповідність приведена в табл. 5.9.

Якщо автомат Мілі має  $m$  станів і  $p$  вхідних сигналів, то відповідний йому автомат Мура матиме  $mp+1$  станів.

З табл. 5.9 витікає той факт, що стан  $Q_0$  автомата Мілі співпадає зі станами  $Q_{00}, Q_{21}, Q_{12}, Q_{22}, Q_{32}$  автомата Мура.

Таблиця 5.9

$X \backslash Q$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$x_1$	$Q_{00}$	$Q_{11}$	$Q_{21}$	$Q_{31}$
$x_2$	$Q_{02}$	$Q_{12}$	$Q_{22}$	$Q_{32}$
$x_3$	$Q_{03}$	$Q_{13}$	$Q_{23}$	$Q_{33}$

Тобто має місце наступна тотожність:  $Q_0 = Q_{00} = Q_{21} = Q_{12} = Q_{22} = Q_{32}$ .

Аналогічно,  $Q_1 = Q_{01}; Q_2 = Q_{03} = Q_{11} = Q_{23} = Q_{31}; Q_3 = Q_{02} = Q_{13} = Q_{33}$ .



Така відповідність означає, що одному переходу автомата Мілі з  $Q_0$  в  $Q_1$  повинні бути відповідними всі переходи автомата Мура зі станів  $Q_{00}, Q_{21}, Q_{12}, Q_{22}, Q_{32}$  в стан  $Q_{01}$ ; переходу з  $Q_1$  в  $Q_2$  повинні бути відповідні всі переходи автомата Мура зі стану  $Q_{01}$  в стани  $Q_{03}, Q_{11}, Q_{23}, Q_{31}$ , і т.д. Зі сказаного витікає наступний висновок: якщо стан  $Q_{mp}$  входить до множини станів, що співпадає зі станом, наприклад,  $Q_1$ , то стовпець таблиці переходів для стану  $Q_{mp}$  співпадатиме з стовпцем таблиці переходів для стану  $Q_1$ . Значення  $m$  функції виходів для еквівалентного автомата Мура:

$$\lambda(Q_{mp}) = \lambda(Q_m, X_p) \text{ при } Q_{mp} \neq Q_{00}.$$

Наприклад, для стану  $Q_{01}$  функція виходу буде  $Y_3$ , оскільки вона відповідає набору  $Q_0$  і  $X_1$  з табл. 5.8; для стану  $Q_{02}$  функція виходу буде  $Y_2$ , що відповідає набору  $Q_0$  і  $X_2$ , і т. д.

Для початкового стану  $Q_{00}$  значення вихідного сигналу вибирається довільно.

Внаслідок описаних відповідностей і перетворень можна побудувати таблицю переходів і виходів еквівалентного автомата Мура (див. табл. 5.10).

Таблиця 5.10

	$Y_3$	$Y_3$	$Y_2$	$Y_1$	$Y_2$	$Y_1$	$Y_2$	$Y_1$	$Y_3$	$Y_2$	$Y_1$	$Y_3$	$Y_3$
	$Q_{00}$	$Q_{01}$	$Q_{02}$	$Q_{03}$	$Q_{11}$	$Q_{12}$	$Q_{13}$	$Q_{21}$	$Q_{22}$	$Q_{23}$	$Q_{31}$	$Q_{32}$	$Q_{33}$
$X_1$	$Q_{01}$	$Q_{11}$	$Q_{31}$	$Q_{21}$	$Q_{21}$	$Q_{01}$	$Q_{31}$	$Q_{01}$	$Q_{01}$	$Q_{21}$	$Q_{21}$	$Q_{01}$	$Q_{31}$
$X_2$	$Q_{02}$	$Q_{12}$	$Q_{02}$	$Q_{22}$	$Q_{22}$	$Q_{02}$	$Q_{32}$	$Q_{02}$	$Q_{02}$	$Q_{22}$	$Q_{22}$	$Q_{12}$	$Q_{32}$
$X_3$	$Q_{03}$	$Q_{13}$	$Q_{33}$	$Q_{23}$	$Q_{23}$	$Q_{03}$	$Q_{33}$	$Q_{33}$	$Q_{33}$	$Q_{23}$	$Q_{23}$	$Q_{33}$	$Q_{33}$

Розглянемо особливості переходу від автомата Мура до еквівалентного автомата Мілі.

Якщо  $f_{p1}$  і  $\lambda_k$  функції переходів і виходів автомата Мура, то функції переходів  $f_{p2}(Q, X)$  і виходів  $\lambda_2(Q, X)$  еквівалентного автомата Мілі:

$$f_{p2}(Q, X) = f_{p1}(Q, X);$$

$$\lambda_2(Q, X) = \lambda_1(f_{p1}(Q, X)).$$

Звідси витікає, що таблиця переходів еквівалентного автомата Мілі співпадає з таблицею переходів автомата Мура, а в кожен клітинку таблиці виходів записується символ, яким відмічений стан автомата Мура в заданій клітині.

При такому перетворенні граф автомата Мілі відрізняється від графу автомата Мура лише тим, що вихідні сигнали з вузлів графа перенесені на всі гілки, що входять у даний вузол.

### **5.3. Основи аналізу цифрових автоматів**

Аналіз роботи автомата виконується з метою визначення його стану в наступний тактовий момент часу і передбачення подальших станів. Такі задачі з'являються при вивченні роботи невідомих схем, при налагодженні пристроїв цифрової схемотехніки. Задачі аналізу є досить складними і розв'язуються поетапно.

Рекомендуються наступні кроки:

- визначаються стани на наступному тактовому моменті часу і значення вхідної та вихідної функцій комбінаційних пристроїв  $L_1$  і  $L_2$ ;
- використовуються функції пристроїв  $L_1$  і  $L_2$  для побудови таблиці станів, яка повністю визначає наступний стан і значення виходу автомату для кожної комбінації поточного стану і виходів;
- будується діаграма станів, яка містить інформацію з попереднього кроку (граф переходів).

#### **5.3.1. Аналіз автоматів з D-тригерами**

Розглянемо особливості аналізу на конкретних прикладах скінченних автоматів.

Виконаємо аналіз автомату, схема якого приведена на рис. 5.4 і містить вхідну комбінаційну схему  $L_1$ , що реалізована на елементах DD1, DD2, вихідну

комбінаційну схему  $L_2$ , реалізовану на DD5, і модуль пам'яті з використанням динамічних  $D$ -тригерів DD3 і DD4. Автомат, що розглядається, відноситься до групи автоматів Мілі.

Розглянемо роботу пристрою за часом.

За фронтом синхросигналу  $C_n$  кожен з тригерів приймає сигнал, що попередньо був поданий на  $D$ -вхід, і передає його на свій вихід в відповідності до алгоритму (див. табл. 5.1). Сигнали  $q_1$  і  $q_2$  є встановлюючими (збуджуючими) для  $D$ -тригерів у кожному тактовий момент часу.

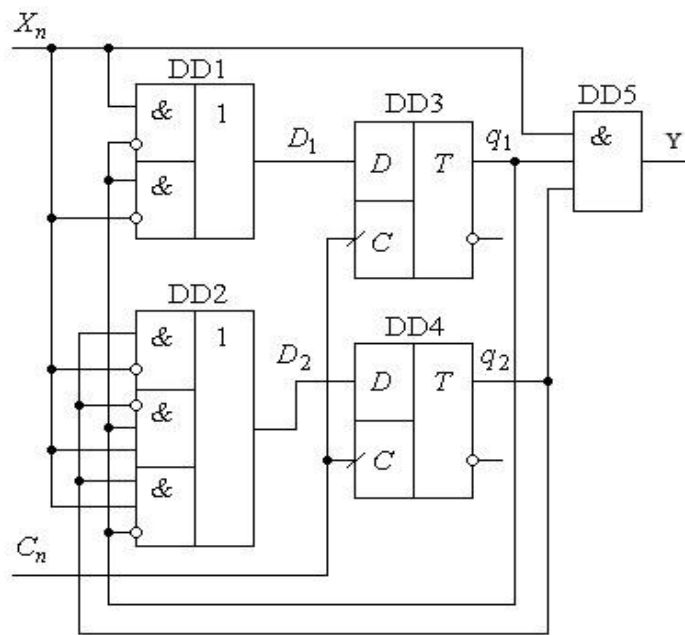


Рис.5.4

Логічні рівняння, що визначають встановлюючі сигнали як функції поточного стану і вхідних сигналів, називаються *збуджуючими рівняннями* (*Excitation equations*) і можуть бути отримані з принципової схеми:

$$D_{1n} = X_n \overline{q_{1n}} + \overline{X_n} q_{1n} = X_n \oplus q_{1n}; \quad (5.5)$$

$$D_{2n} = \overline{X_n} q_{2n} + X_n \overline{q_{1n}} q_{2n} + X_n q_{1n} \overline{q_{2n}} = \overline{X_n} q_{2n} + X_n (q_{1n} \oplus q_{2n}). \quad (5.6)$$

Використовуючи характеристичні рівняння  $D$ -тригерів, можемо записати:

$$q_{1(n+1)} = D_{1n} \cdot C_n; \quad (5.7)$$

$$q_{2(n+1)} = D_{2n} \cdot C_n. \quad (5.8)$$

Підставляючи значення  $D_{1n}$ ,  $D_{2n}$  з (5.5) – (5.6), отримуємо:

$$q_{1(n+1)} = C_n \cdot (X_n \oplus q_{1n}) ; \quad (5.9)$$

$$q_{2(n+1)} = C_n \cdot (\overline{X_n} \cdot q_{2n} + X_n \cdot (q_{1n} \oplus q_{2n})) . \quad (5.10)$$

Отримані рівняння (5.9) – (5.10), що виражають значення стану автомата в  $(n+1)$ -й момент часу в залежності від значень його входів і стану в  $n$ -й момент часу, називаються *перехідними рівняннями*.

Оскільки внутрішній стан автомата визначається значеннями виходів двох тригерів, то можна стверджувати, що існують лише чотири можливі стани автомата:  $(q_1, q_2) = 00, 01, 10, 11$ . Для кожного стану в прикладі, що розглядається, можливі лише два значення вхідного сигналу  $X$ : 0, 1.

В результаті маємо вісім комбінацій стан-вхід (у загальному плані автомат з  $m$  станами і  $p$  вхідними сигналами матиме  $2^{m+p}$  комбінацій стан-вхід).

Перехідні рівняння (5.9) – (5.10) можемо задати в табличній формі (див. табл. 5.11), з якої витікає особливість роботи автомата.

Таблиця 5.11

$C_n$	$X_n$	$q_{2n}$	$q_{1n}$	$q_{2(n+1)}$	$q_{1(n+1)}$
1	0	0	0	0	0
2	0	0	1	0	1
3	0	1	0	1	0
4	0	1	1	1	1
1	1	0	0	0	1
2	1	0	1	1	0
3	1	1	0	1	1
4	1	1	1	0	0

Вона полягає в тому, що при  $X_n = 0$  значення вихідних станів в  $(n+1)$ -й момент часу повторюють відповідні значення в  $n$ -й момент часу. При  $X_n = 1$  автомат виконує функцію двійкового лічильника з максимальним фіксованим значенням  $q_2 q_1 = 11_2 = 3_{10}$ .

Побудуємо тепер на основі табл. 5.11 таблицю переходів автомата. Для цього кожен з можливих станів автомата позначимо відповідним значенням з множини  $Q$ , починаючи з нульового:

$$Q_0 = \overline{q_{2n}} \overline{q_{1n}} = 00; \quad Q_1 = \overline{q_{2n}} q_{1n} = 01; \quad Q_2 = q_{2n} \overline{q_{1n}} = 10; \quad Q_3 = q_{2n} q_{1n} = 11.$$

В результаті отримуємо таблицю переходів у вигляді табл. 5.12 або її скорочену форму – табл. 5.13.

Граф-схема автомата приведена на рис. 5.5.

Послідовність переходів від одного стану до іншого забезпечується при наявності динамічного сигналу  $C_n$  від зовнішнього тактового генератора і потенційного (дозволяючого) сигналу  $x$ . Робота автомата може бути описана і за допомогою часових діаграм (рис. 5.6), які відображають зміну станів тригерів автомата ( $q_{1n}$ ,  $q_{2n}$ ) у відповідні тактові моменти часу, і відповідні стани автомата ( $Q_0 \div Q_3$ ) на кожному тактовому інтервалі.

Таблиця 5.12

$C_n$	$X_n$	$Q_n$	$Q_{n+1}$
1	0	$Q_0$	$Q_0$
2	0	$Q_1$	$Q_1$
3	0	$Q_2$	$Q_2$
4	0	$Q_3$	$Q_3$
1	1	$Q_0$	$Q_1$
2	1	$Q_1$	$Q_2$
3	1	$Q_2$	$Q_3$
4	1	$Q_3$	$Q_0$

Таблиця 5.13

$X \cdot C_n \backslash Q$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
0	$Q_0$	$Q_1$	$Q_2$	$Q_3$
1	$Q_1$	$Q_2$	$Q_2$	$Q_0$

Розглянуту схему автомата можливо спростити за рахунок видалення інверторів, оскільки інверсні значення виходів тригерів можна отримати безпосередньо з тригерів.

Вихідний сигнал автомата визначається внутрішнім його станом і значенням вхідного сигналу, тому функція  $L_2$  описується рівнянням:

$$L_2 = Y = x \cdot q_{1n} \cdot q_{2n}.$$

**Приклад 5.2.** Проаналізувати роботу скінченного автомата, принципова схема якого приведена на рис. 5.7. Побудувати його таблицю станів та часові діаграми роботи.

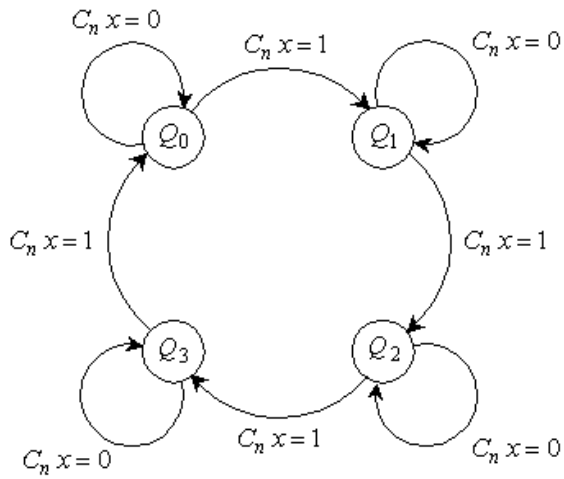


Рис.5.5

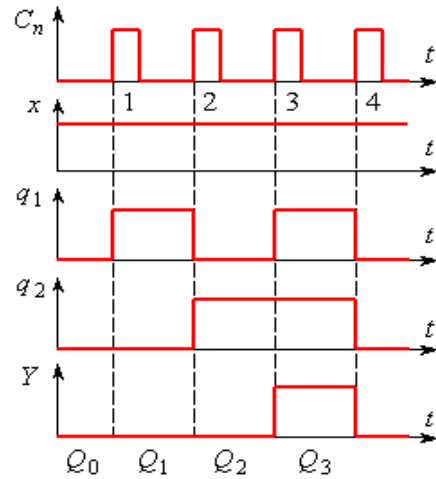


Рис.5.6

*Розв'язання.* Із зовнішнього вигляду принципової схеми бачимо, що аналізований автомат відноситься до класу автоматів Мура, а рівняння для вихідних сигналів  $y_1$  та  $y_2$  мають вигляд:

$$y_1 = \overline{q_2} + q_1 + q_0; \quad (5.11)$$

$$y_2 = q_2 \cdot (q_1 + \overline{q_0}). \quad (5.12)$$

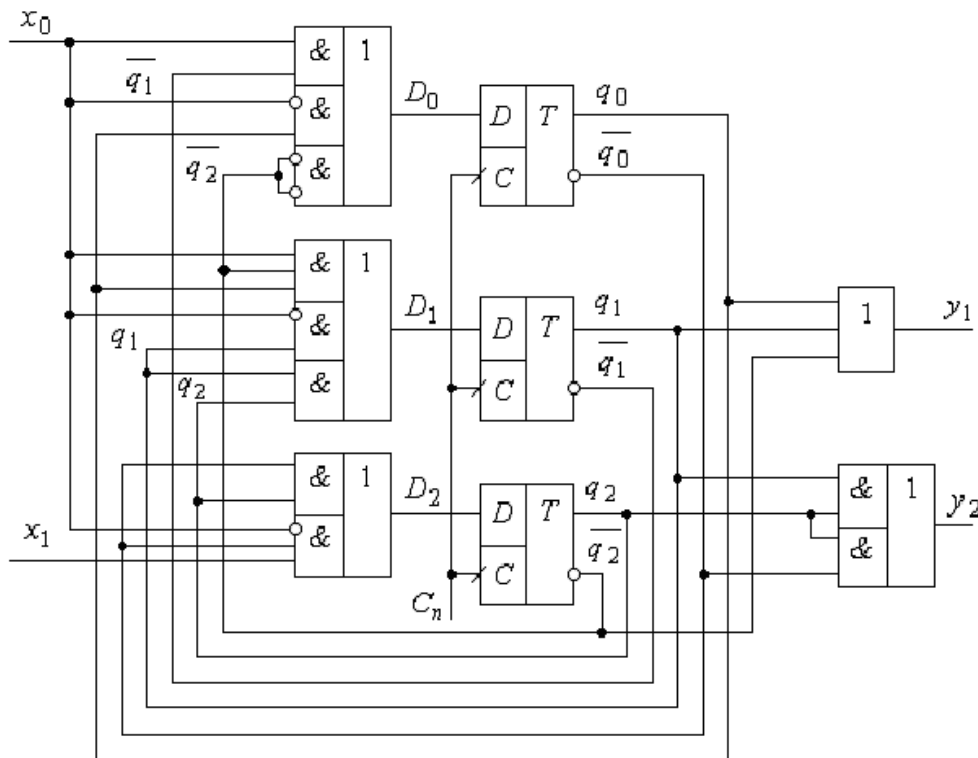


Рис.5.7

Аналізуючи логічну схему автомата, можемо записати рівняння збудження тригерів:

$$D_{0n} = x_{0n} \overline{q_{1n}} + \overline{x_{0n}} q_{0n} + q_{2n};$$

$$D_{1n} = x_{0n} \overline{q_{2n}} q_{0n} + \overline{x_{0n}} q_{1n} + q_{2n} q_{1n};$$

$$D_{2n} = q_{2n} \overline{q_{0n}} + x_{1n} \overline{x_{0n}} q_{0n}.$$

Підставляючи рівняння збудження в характеристичне рівняння  $D$ -тригерів, отримаємо:

$$q_{0(n+1)} = C_n D_{0n} = C_n (x_{0n} \overline{q_{1n}} + \overline{x_{0n}} q_{0n} + q_{2n});$$

$$q_{1(n+1)} = C_n D_{1n} = C_n (x_{0n} \overline{q_{2n}} q_{0n} + \overline{x_{0n}} q_{1n} + q_{2n} q_{1n});$$

$$q_{2(n+1)} = C_n D_{2n} = C_n (q_{2n} \overline{q_{0n}} + x_{1n} \overline{x_{0n}} q_{0n}).$$

Функції  $q_{0(n+1)}$ ,  $q_{1(n+1)}$ ,  $q_{2(n+1)}$  зобразимо в табличній формі (див. табл. 5.14).

Таблиця 5.14

$C_n$	$x_0$	$x_1$	$q_{2n}$	$q_{1n}$	$q_{0n}$	$q_{2(n+1)}$	$q_{1(n+1)}$	$q_{0(n+1)}$
1	0	0	0	0	0	0	0	0
2	0	0	0	0	1	0	0	1
3	0	0	0	1	0	0	1	0
4	0	0	0	1	1	0	1	1
5	0	0	1	0	0	1	0	1
6	0	0	1	0	1	0	0	1
7	0	0	1	1	0	1	1	1
8	0	0	1	1	1	0	1	1
9	0	1	0	0	0	1	0	0
10	0	1	0	0	1	0	0	1
11	0	1	0	1	0	1	1	0
12	0	1	0	1	1	0	1	1
13	0	1	1	0	0	1	0	1
14	0	1	1	0	1	0	0	1
15	0	1	1	1	0	1	1	1
16	0	1	1	1	1	0	1	1
17	1	0	0	0	0	0	0	1
18	1	0	0	0	1	0	1	1
19	1	0	0	1	0	0	0	0
20	1	0	0	1	1	0	1	0
21	1	0	1	0	0	1	0	1
22	1	0	1	0	1	0	0	1
23	1	0	1	1	0	1	1	1
24	1	0	1	1	1	0	1	1
25	1	1	0	0	0	0	0	1
26	1	1	0	0	1	0	1	1
27	1	1	0	1	0	0	0	0
28	1	1	0	1	1	0	1	0
29	1	1	1	0	0	1	0	1
30	1	1	1	0	1	0	0	1
31	1	1	1	1	0	1	1	1
32	1	1	1	1	1	0	1	1

Введемо умовні позначення станів автомата  $Q_m$  у відповідності до табл. 5.15. На основі табл. 5.14, використовуючи позначення станів автомату (табл. 5.15), побудуємо таблицю переходів в компактній наглядній формі (див. табл. 5.16).

Таблиця 5.15

№ п/п	$q_{2n}$	$q_{1n}$	$q_{0n}$	$Q_m$
0	0	0	0	$Q_0$
1	0	0	1	$Q_1$
2	0	1	0	$Q_2$
3	0	1	1	$Q_3$
4	1	0	0	$Q_4$
5	1	0	1	$Q_5$
6	1	1	0	$Q_6$
7	1	1	1	$Q_7$

Таблиця 5.16

$Q_n$	$Q_{n+1}$				$y_1 y_2$
	$C_n x_0 x_1$				
	00	01	10	11	
$Q_0$	$Q_0$	$Q_4$	$Q_1$	$Q_1$	10
$Q_1$	$Q_1$	$Q_1$	$Q_3$	$Q_3$	10
$Q_2$	$Q_2$	$Q_6$	$Q_0$	$Q_0$	10
$Q_3$	$Q_3$	$Q_3$	$Q_2$	$Q_2$	00
$Q_4$	$Q_5$	$Q_5$	$Q_5$	$Q_5$	11
$Q_5$	$Q_1$	$Q_1$	$Q_1$	$Q_1$	10
$Q_6$	$Q_7$	$Q_7$	$Q_7$	$Q_7$	11
$Q_7$	$Q_3$	$Q_3$	$Q_3$	$Q_3$	11

На основі таблиці переходів будується граф-схема автомата (рис. 5.8) і аналізуються умови переходів.

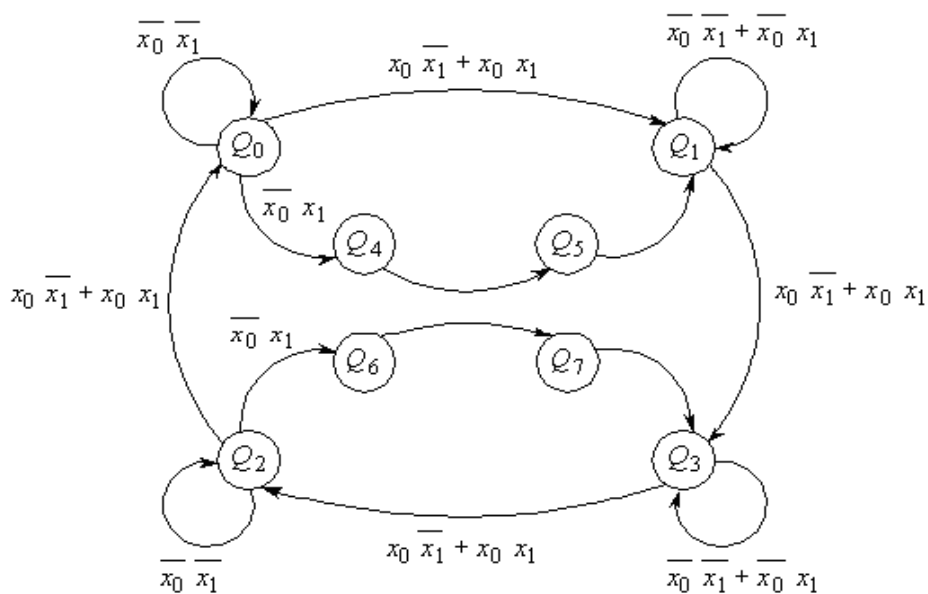


Рис.5.8

Аналіз роботи автомата на основі граф-схеми з урахуванням керуючих сигналів дає можливість виявити деякі аспекти його роботи. Наприклад, перехід зі стану  $Q_4$  в стан  $Q_5$



забезпечується при всіх значеннях  $x_0 x_1$  і є безумовним. Аналогічно, маємо переходи з  $Q_5$  в  $Q_1$ ; з  $Q_6$  в  $Q_7$ ; з  $Q_7$  в  $Q_3$ .

Звідси витікає, що перехід з  $Q_0$  в  $Q_4$  можна замінити на перехід з  $Q_0$  в  $Q_1$ , а перехід з  $Q_2$  в  $Q_6$  – на перехід з  $Q_2$  в  $Q_3$ . Таблиця переходів при цьому може бути спрощена (див. табл. 5.17), а вигляд граф-схеми відповідатиме рис. 5.9.

Таблиця 5.17

$Q_n$	$Q_{n+1}$				$y_1 y_2$
	$C_n x_0 x_1$				
	00	01	10	11	
$Q_0$	$Q_0$	$Q_1$	$Q_1$	$Q_1$	10
$Q_1$	$Q_1$	$Q_1$	$Q_3$	$Q_3$	10
$Q_2$	$Q_2$	$Q_3$	$Q_0$	$Q_0$	10
$Q_3$	$Q_3$	$Q_3$	$Q_2$	$Q_2$	00

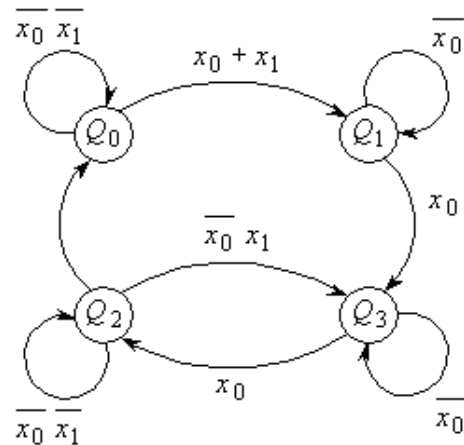


Рис.5.9

Приведений аналіз схеми автомата показує, що при  $x_1 = 0$  матимемо лічильник, який працює у відповідності до коду Грея. При  $x_1 = 1$  виникає можливість забезпечувати своєрідний тригерний режим між станами  $Q_2$  і  $Q_3$ .

Отримані результати дають можливість побудувати часові діаграми роботи автомата. При цьому слід враховувати, що при станах  $Q_4 \div Q_7$  має місце  $y_1 = 1$ , а  $y_2$  переходить в нуль у стані  $Q_5$ , який є нестійким. Тому можна вважати, що  $y_2 = 1$ .

Часові діаграми роботи автомата приведені на рис. 5.10. Часові діаграми виходів  $y_1$  та  $y_2$  не приведені, оскільки вони зрозумілі з формул (5.11), (5.12).

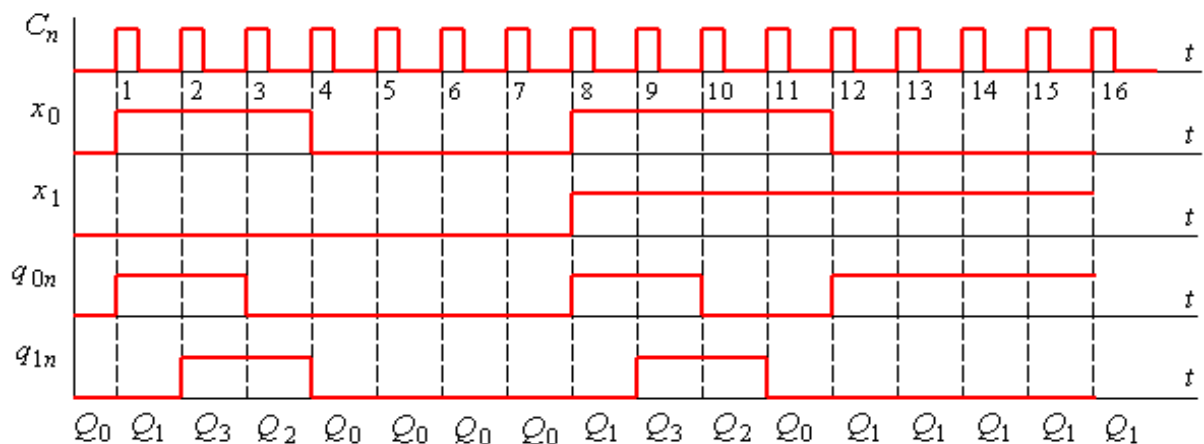


Рис.5.10

### 5.3.2. Особливості аналізу скінченних автоматів з *JK*-тригерами

Синхронізовані скінченні автомати, що будуються на базі *JK*-тригерів, можуть аналізуватись у відповідності до процедури, яка була описана вище. Різниця полягає лише в тому, що необхідно мати два збуджуючих рівняння (*Excitation equations*) для кожного тригера – одне для *J*-входу, друге – для входу тригера *K*.

Для отримання рівнянь переходів (*Transition equations*) кожне з них повинно використовувати характеристичне рівняння *JK*-тригера:

$$Q_{n+1} = ( J_n \overline{Q_n} + \overline{K_n} Q_n ) C_n .$$

На рис. 5.11 приводиться приклад скінченного автомата з використанням *JK*-тригерів.

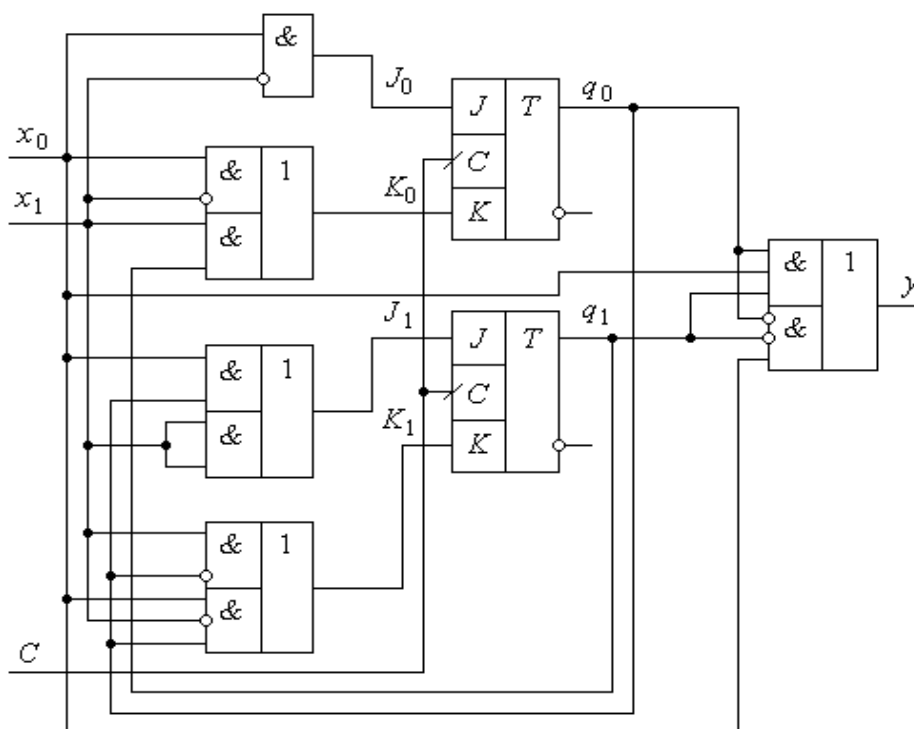


Рис.5.11

З рис. 5.11 можемо записати наступні рівняння збудження для входів *JK*-тригерів:

$$\begin{aligned} J_{0n} &= x_0 \overline{x_1} ; & K_{0n} &= x_0 \overline{x_1} + x_1 q_{1n} ; \\ J_{1n} &= x_0 q_{0n} + x_1 ; & K_{1n} &= x_1 \overline{q_{0n}} + x_0 \overline{x_1} q_{0n} . \end{aligned}$$

Підставляючи їх в характеристичне рівняння тригера, отримуємо рівняння переходів:

$$q_{0(n+1)} = C_n (x_0 \overline{x_1} \overline{q_{0n}} + (x_0 \overline{x_1} + x_1 q_{1n}) q_{0n}) ; \quad (5.13)$$

$$q_{1(n+1)} = C_n ((x_0 q_{0n} + x_1) \overline{q_{1n}} + (x_1 q_{0n} + x_0 x_1 q_{0n}) q_{1n}) \quad (5.14)$$

Таблиця переходів, що базується на цих рівняннях, легко може бути побудована.

Рівняння (5.13) і (5.14) приведемо до досконалої форми:

$$q_{0(n+1)} = C_n (\overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + \overline{x_{0n}} \overline{x_{1n}} \overline{q_{0n}} q_{1n} + \overline{x_{0n}} x_{1n} \overline{q_{0n}} \overline{q_{1n}} + x_{0n} \overline{x_{1n}} \overline{q_{0n}} \overline{q_{1n}} + x_{0n} x_{1n} \overline{q_{0n}} \overline{q_{1n}} + x_{0n} x_{1n} \overline{q_{0n}} q_{1n})$$

$$q_{1(n+1)} = C_n (x_0 x_1 q_{0n} \overline{q_{1n}} + x_0 \overline{x_1} q_{0n} \overline{q_{1n}} + x_0 x_1 \overline{q_{0n}} \overline{q_{1n}} + x_0 \overline{x_1} \overline{q_{0n}} \overline{q_{1n}} + x_0 x_1 q_{0n} q_{1n} + x_0 \overline{x_1} q_{0n} q_{1n} + x_0 x_1 \overline{q_{0n}} q_{1n} + x_0 \overline{x_1} \overline{q_{0n}} q_{1n}) .$$

Таблиця станів для обох функцій  $q_{0(n+1)}$  і  $q_{1(n+1)}$  має вигляд табл. 5.18, а карти Карно приведені на рис. 5.12.

Таблиця 5.18

N	$x_0$	$x_1$	$q_{0n}$	$q_{1n}$	$q_{0(n+1)}$	$q_{1(n+1)}$
0	0	0	0	0	0	0
1	0	0	0	1	0	0
2	0	0	1	0	1	0
3	0	0	1	1	0	0
4	0	1	0	0	0	1
5	0	1	0	1	0	1
6	0	1	1	0	0	1
7	0	1	1	1	1	0
8	1	0	0	0	0	0
9	1	0	0	1	1	0
10	1	0	1	0	0	1
11	1	0	1	1	1	1
12	1	1	0	0	0	1
13	1	1	0	1	1	1
14	1	1	1	0	1	1
15	1	1	1	1	0	0

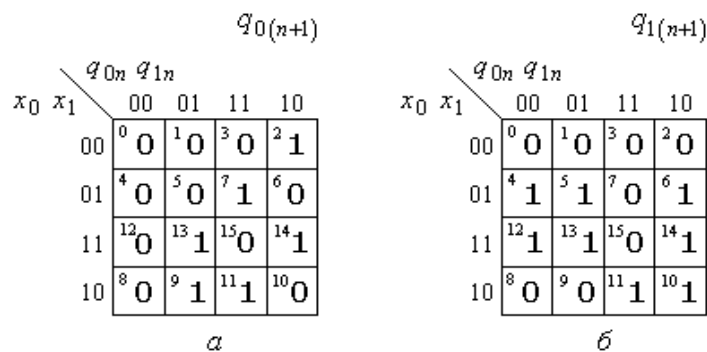


Рис.5.12

Складаємо таблицю кодування станів (табл. 5.19) та таблицю переходів (табл. 5.20).

Таблиця 5.19

$N$	$q_{0n}$	$q_{1n}$	$Q$
0	0	0	$Q_0$
1	0	1	$Q_1$
2	1	0	$Q_2$
3	1	1	$Q_3$

Таблиця 5.20

$Q_n$	$Q_{n+1}$			
	$x_0 x_1$			
	00	01	10	11
$Q_0$	$Q_0$	$Q_1$	$Q_0$	$Q_1$
$Q_1$	$Q_0$	$Q_1$	$Q_2$	$Q_3$
$Q_2$	$Q_2$	$Q_1$	$Q_1$	$Q_3$
$Q_3$	$Q_0$	$Q_2$	$Q_3$	$Q_0$

Будуємо граф-схему переходів (рис. 5.13).

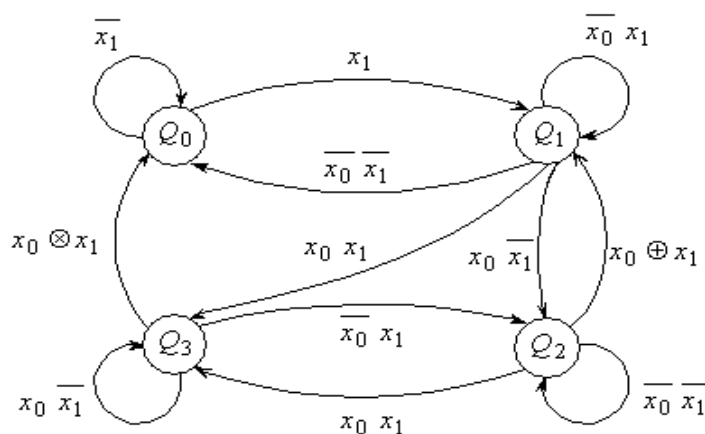


Рис.5.13

Рівняння, що описує залежність вихідного сигналу, має вигляд:

$$y = x_0 q_{0n} q_{1n} + \overline{x_0} \overline{q_{0n}} \overline{q_{1n}} \cdot$$

Слід відмітити, що при використанні  $JK$ -тригерів можна використати і інший шлях побудови таблиці переходів. Він полягає в тому, що створюються рівняння збудження і таблиця збудження відносно входів тригерів  $J_0, K_0, J_1, K_1$ , які потім використовуються в характеристичних рівняннях кожного тригера при побудові таблиці переходів. Використання такого шляху частково демонструвалося в попередніх прикладах.

Звернемося знову до прикладу схеми двійкового лічильника, що приведена на рис. 5.4, і розглянемо інший варіант його побудови з використання

*JK*-тригерів. Для цього побудуємо таблицю станів з урахуванням сигналів на входах *J*- і *K*- кожного тригера і визначимося з логічною функцією  $L_1$ , що задаватиме сигнали *J* і *K*.

Оскільки нам необхідно мати два тригери, то позначимо їх, як і в схемі-прототипі, – через DD3 і DD4. Відповідні входи тригерів позначимо як  $J_3, K_3$  і  $J_4, K_4$ , а виходи –  $q_3$  і  $q_4$  (табл. 5.21).

Таблиця 5.21

$C_n$	$q_{4n}$	$q_{3n}$	$q_{4(n+1)}$	$q_{3(n+1)}$	$J_4$	$K_4$	$J_3$	$K_3$
1	0	0	0	1	0	x	1	x
2	0	1	1	0	1	x	x	1
3	1	0	1	1	x	0	1	x
4	1	1	0	0	x	1	x	1

Таблиця будується в такій послідовності. В стовпці  $C_n$  задаємо послідовність тактових сигналів. У стовпцях  $q_3$  і  $q_4$  задаємо початкові значення виходів тригерів до початку дії синхросигналу і відповідних інформаційних сигналів. Формально в цих стовпцях записується таблиця двійкового коду для двох змінних. У стовпцях  $q_{3(n+1)}$  і  $q_{4(n+1)}$  записуються значення виходів  $q_3$  і  $q_4$  після дії синхросигналу в відповідності до табл. 5.11. В стовпцях  $J_3, K_3, J_4, K_4$  записуються значення входів, які приведуть до відповідної зміни кодових значень в стовбцях  $q_3$  і  $q_4$ . Наприклад: до початку дії синхросигналу в стовпці  $q_{3n}$  маємо “0”, а після його дії – “1”, в стовпці  $q_{4n}$  – “0”, який не змінюється з приходом синхросигналу. До такої зміни станів тригерів призводять сигнали  $J_4=0, J_3=1$ . Значення входів  $K_3, K_4$  – байдужі.

Внаслідок побудови таблиці отримуємо залежності значень виходів  $q_3$  і  $q_4$  тригерів від сигналів на *J* та *K* входах.

Для першого тригера бачимо, що сигнали  $J_3$  і  $K_3$  мають значення, або байдужі, або рівня “1”, тому вважаємо, що на його  $JK$  входах постійно може бути присутня “1”, а, відповідно до алгоритму роботи, це режим  $T$ -тригера.

Для визначення значень входів  $J_4$  і  $K_4$  також бачимо, що вони можуть бути однаковими, тобто  $J_4 = K_4$ , а подальший аналіз зв'язку  $J_4, K_4$  з  $q_3$  приводить до того, що знаходимо  $J_4 = K_4 = q_{3n}$ .

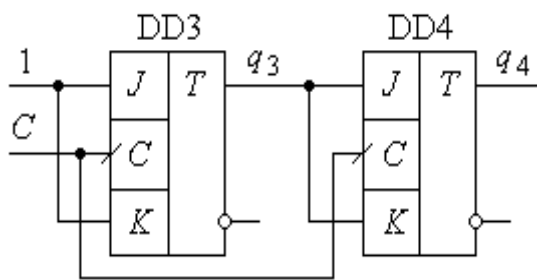


Рис.5.14

Звідси витікає схема лічильника на  $JK$ -тригерах (рис. 5.14).

Проаналізуємо його роботу. Якщо вважати початковим станом  $q_3 = q_4 = 0$ , то поява першого синхроімпульсу призведе до нового стану  $q_3 = 1, q_4 = 0$ .

Другий синхроімпульс призведе до зміни стану DD4, тобто  $q_4 = 1$ , а також до зміни стану DD3:  $q_3$  зміниться в нуль. Третій синхроімпульс приведе до зміни стану  $q_3$  з “0” в “1” і, четвертий імпульс - до обнуління обох тригерів.

Отриманий результат говорить про те, що складність комбінаційної схемотехніки скінченного автомата суттєво залежить від типу використовуваних тригерів. Якщо ж говорити більш точно, то складність комбінаційних схем залежить від уміння розробника використати логічні властивості використовуваних тригерів.

## 5.4. Синтез скінченних автоматів

### 5.4.1. Основи синтезу скінченних автоматів

Задача синтезу (проекування) скінченних автоматів майже діаметрально протилежна попередній задачі аналізу. У ряді літературних джерел приводиться послідовність кроків, які поетапно необхідно робити в процесі розробки автомата. Процедура синтезу характерна наступними основними етапами:

1. Задається закон функціонування автомата;

2. Мінімізується кількість внутрішніх станів автомата;
3. Кодуються стани автомата;
4. Визначаються функції збудження елементів пам'яті і функції виходів, а також забезпечується їх мінімізація;
5. Складається функціональна схема автомата у вибраному елементному базисі.

Перші три етапи відносяться до *абстрактного синтезу* автомата, при якому по значеннях вхідних і вихідних сигналів визначаються і кодуються його стани.

При абстрактному синтезі на основі словесного, табличного, за допомогою часових діаграм і інших засобів опису роботи автомату виявляється закон його функціонування і задаються функції переходів і виходів. На цьому етапі забезпечується мінімізація кількості станів і визначається кількість елементів пам'яті, необхідних для побудови автомата.

Решта етапів забезпечує процедуру одержання структурних схем автоматів, що задаються кодованими таблицями переходів і виходів. Така процедура називається *структурним синтезом*.

Задача структурного синтезу полягає у виборі типів елементарних автоматів (тригерів) і знаходженні такого способу їх з'єднання між собою, який забезпечить функціонування автомата у відповідності до заданих кодованих таблиць переходів і виходів.

Основною задачею цього етапу є синтез комбінаційних схем з урахуванням вибраних тригерів, знаходження мінімальних форм функцій збудження і виходів. Етап закінчується побудовою функціональної схеми автомата.

Розглянемо змістовну частину кожного з етапів на конкретних прикладах.

**Етап 1.** *Задання закону функціонування асинхронного автомата.*

Початковою інформацією для першого етапу може служити словесний опис алгоритму функціонування автомата або аналіз часових діаграм його

роботи. Результатом цього етапу повинно бути отримання формалізованого опису закону функціонування в вигляді таблиць переходів, графів переходів і т. п. Практичний досвід показує, що зручно задавати автомат за допомогою *початкових таблиць переходів* (ПТП), оскільки їхня структура є найбільш простою.

ПТП характеризується тим, що в кожному її рядку є тільки один стійкий стан, а всі переходи між станами прості.

Розглянемо декілька прикладів складання ПТП, оскільки цей етап є одним з найбільш важливих і складних при проектуванні автоматів.

**Приклад 5.3.** Необхідно синтезувати автомат (цифровий пристрій), що пропускає імпульси прямокутної форми  $x_0$ , якщо немає заборони ( $x_1 = 0$ ), і не пропускає, якщо заборона існує ( $x_1 = 1$ ). При цьому імпульси  $x_0$  не повинні спотворюватись незалежно від моменту появи сигналу заборони  $x_1$ .

*Розв'язання.* Такий автомат має назву *генератора з заборonoю*. Для більш формального опису можна побудувати часові діаграми, які пояснюють його функціонування. Відповідні діаграми для вхідних і вихідних сигналів приведені на рис. 5.15.

На основі аналізу зміни вхідних і вихідних сигналів можна вводити стани автомата, які характеризуються станом входу (сигналами  $x_0$  і  $x_1$  з вхідного алфавіту  $X$ ) і станом виходу  $Y$  (сигналом  $y$ ).

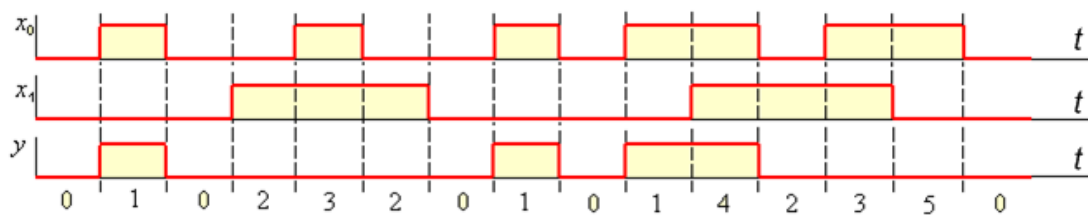


Рис.5.15

Різнманітним комбінаціям вхідних та вихідних сигналів припишемо відповідні номери, які характеризують всі можливі стани автомата. За початковий приймемо стан  $Q_{00} = 0$ , який характеризується значеннями вхідних сигналів  $x_1 x_0 = 00$  і значенням вихідного сигналу  $y = 0$ . Стан  $Q_{00} = 0$  є *стійким*, тобто таким, який не зміниться без дії вхідних сигналів. В таблиці ПТП, що будується, стійкі стани будемо виділяти дужками.



Нестійкі стани автомата – це такі, які можуть змінюватися на стійкі без зміни вхідних сигналів. Такі стани в ПТП будемо позначати їх номерами без дужок.

Якщо на автомат, що знаходиться в стійкому стані  $Q_{00}$ , діятиме вхідний сигнал  $x_1 x_0 = 01$ , то він перейде в новий, спочатку нестійкий стан, який потім в стійкий  $Q_1 = (1)$ . Нестійкий стан відповідає ситуації, коли в результаті зміни вхідного сигналу  $X$  змінились сигнали збудження тригерів, але ще не змінився вихідний сигнал, тобто  $y$  залишився рівним нулю. По закінченні перехідних процесів автомат перейде в стійкий стан  $Q_1 = (1)$ , який характеризується вхідними сигналами  $x_1 x_0 = 01$  і вихідним  $y = 1$ .

Переходу зі стійкого стану (0) в нестійкий 1 в ПТП відповідає переміщення в межах першого рядка з клітини з координатами  $x_1 x_0 = 00$  в клітину з координатами  $x_1 x_0 = 01$ . Вихідний сигнал при цьому залишається незмінним:  $y = 0$ .

Переходу з нестійкого стану 1 в стійкий стан (1) в ПТП відповідає переміщення в межах одного стовпця з координатами  $x_1 x_0 = 01$  з першого рядка до другого. Вихідний сигнал при цьому змінюється і приймає значення  $y = 1$  (табл. 5.22).

Таблиця 5.22

X \ Q	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1			0
1		(1)			1

Звідси маємо, що перехід з одного стійкого стану (0) в стійкий стан (1) відповідає двом фрагментам переходу: спочатку у межах рядка, а потім у межах стовпця.

З часової діаграми, приведеної на рис. 5.15, видно, що стійкий стан (1) при  $x_1 x_0 = 00$  змінюється на стан (0), а потім під дією сигналу  $x_1 x_0 = 10$  переходить в стан (2), і т.д.

Номери станів на часовій діаграмі зображають так, щоб кожна змінювана ситуація в сигналах  $X, Y$  мала свій номер і щоб були враховані всі ситуації, які вимагають спеціального запам'ятовування.

Аналіз всієї часової діаграми дозволяє заповнити ПТП (див. табл. 5.23), але не повністю – частина клітин залишається вільними. Наприклад, у першому рядку ПТП залишилась незаповненою клітиною з  $X = 11$ .

Таблиця 5.23

X \ Q	$x_1 x_0$				y
	00	01	11	10	
0	(0)	1	—	2	0
1	0	(1)	4	—	1
2	0	—	3	(2)	0
3	—	5	(3)	2	0
4	—	—	(4)	2	1
5	0	(5)	—	—	0

Оскільки стан  $Q = 0$  є стійким при  $X = 00$ , то перехід в новий стан, що відповідає  $X = 11$ , вимагає одночасної зміни обох вхідних сигналів  $x_1$  і  $x_0$ , що практично неможливо.

Оскільки перехід  $00 \rightarrow 11$  виконати неможливо, в ПТП в відповідній клітині проставляється риска, яка інформує про наявність невизначеного стану. Аналогічно виключаються інші переходи типу  $00 \leftrightarrow 11$ , а також  $01 \leftrightarrow 10$ .

**Приклад 5.4.** Скласти ПТП для автомата, призначеного для ділення частоти. На виході автомата створюється імпульс, якщо на вхід поступило 3 імпульси (часові діаграми приведені на рис. 5.16).

*Розв'язання.* Приклад демонструє випадок, коли деяким ситуаціям, що повторюються, необхідно поставити в відповідність різні стани. Так, стани 0, 2, 4 (як і 1, 3) на часових діаграмах нічим не відрізняються між собою. Але, в той же час, зрозуміло, що для автомата це різні стани, оскільки в кожному з них повинна запам'ятовуватись різна інформація, що мала місце на попередніх інтервалах часу.

Таблиця ПТП автомата, що розглядається, має лише 3 стовпці, і при її заповненні вільні клітини не з'являються (див. табл. 5.24).

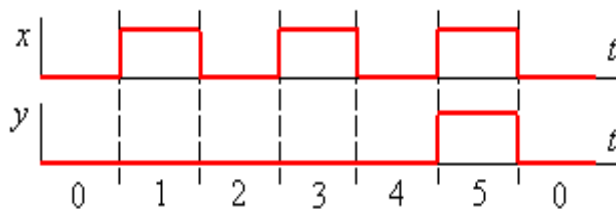


Рис.5.16

Таблиця 5.24

Q \ X	x		y
	0	1	
0	(0)	1	0
1	2	(1)	0
2	(2)	3	0
3	4	(3)	0
4	(4)	5	0
5	0	(5)	1

## Етап 2. Мінімізація кількості внутрішніх станів автомата.

Ідея формальної процедури мінімізації кількості станів полягає в визначенні еквівалентних станів, де два стани є *еквівалентними*, якщо неможливо визначитись з різницею в їх функціонуванні, розглядаючи поточні і майбутні стани виходів автомата. Пара еквівалентних станів може бути замінена одним.

Існують детальні процедури, які дозволяють формально розв'язувати задачу мінімізації кількості станів, але на практиці вони рідко використовуються. Реально це обумовлено тим, що досвідчені інженери з електроніки проектують цифрові автомати з мінімальною кількістю станів без

будь-яких проблем, не використовуючи формальні процедури. Більш того, часто мають місце ситуації, коли збільшення кількості станів не ускладнює проект, а, навпаки, спрощує його, а мінімізуючі процедури не дають необхідної користі.

Як правило, ПТП містить ряд збиткових внутрішніх станів, усунення яких не порушить алгоритму роботи автомата, але, в той же час, дозволить спростити його функціональну схему. В ряді випадків наявність збиткових станів не є очевидним фактом, тому для їх виявлення необхідно проводити детальний аналіз ПТП.

Скорочення частини станів асинхронного автомата базується на виявленні еквівалентних і псевдоеквівалентних станів, а також на виявленні і об'єднанні сумісних внутрішніх станів автомата (рядків ПТП).

*Еквівалентними станами* автомата називаються стійкі стани, які задовольняють наступним умовам:

1. Їм відповідає один і той же стан входу (тобто вони знаходяться в одному і тому ж стовпці ПТП);
2. Їм відповідає один і той же стан виходу (однакові вихідні сигнали автомата);
3. Будь-якій послідовності станів входу автомата відповідає одна і та ж сама послідовність станів його виходу, незалежно від того, який зі стійких станів автомата прийнято за початковий.

**Приклад 5.5.** Автомат заданий ПТП, що приведена в табл. 5.25. Визначити еквівалентні та псевдоеквівалентні стани і скоротити кількість рядків ПТП.

*Розв'язання.* У відповідності до першої умови, еквівалентними можуть бути лише ті стани, які знаходяться в одному і тому ж стовпчику таблиці, тобто стани (3) і (6); (1) і (4); (2) і (5); (2) і (7); (5) і (7).

З отриманих пар необхідно вибрати стани, що відповідають другій умові, тобто такі, яким відповідає один і той же стан виходу. Наприклад, стан (5) має

Таблиця 5.25

Q \ X	$x_1 x_0$				y
	00	01	11	10	
0	(0)	6	1	5	0
1	0	6	(1)	2	0
2	0	3	4	(2)	1
3	0	(3)	1	2	0
4	0	3	(4)	2	0
5	0	3	4	(5)	0
6	0	(6)	4	7	0
7	0	3	4	(7)	1

значення виходу, протилежне значенням виходів станів (2) і (7). Тому можемо стверджувати, що стани (2) і (5), а також (5) і (7) не є еквівалентними. З пар станів, що залишилися – (3) і (6); (1) і (4); (2) і (7) – еквівалентними будуть ті стани, які відповідають ще й третій умові.

Для того, щоб два стійкі стани задовольняли третій умові еквівалентності, в однакових стовпцях таблиці переходів в рядках, відповідних даним станам, повинні знаходитись одні і ті ж цифри, або різні цифри, але які визначають еквівалентні стани. У прикладі, що розглядається, еквівалентними будуть стійкі стани (2) і (7); (1) і (4); (3) і (6).

Після того, як еквівалентні стани виявлені, їх об'єднують. При цьому кожній групі еквівалентних станів приписують один номер, а всі рядки, в які входять еквівалентні стани однієї групи, замінюються одним рядком. У розгляданому прикладі стійкий стан (7) замінюється на (2); (6) – на (3); (4) – на (1). Так само перенумеровуються і нестійкі стани 7, 6 і 4 відповідно на 2, 3, 1.

В результаті отримуємо таблицю зі скороченою кількістю станів (табл. 5.26).

Таблиця 5.26

X Q	X <sub>1</sub> X <sub>0</sub>				y
	00	01	11	10	
0	(0)	3	1	4	0
1	0	3	(1)	2	0
2	0	3	1	(2)	1
3	0	(3)	1	2	0
4	0	3	1	(4)	0

Для співпадиння номерів стійких станів з номерами рядків у табл. 5.26 необхідно стани 5 і (5) перенумерувати на 4 і (4). Для неповністю визначеного автомата, окрім поняття еквівалентності, існує і використовується поняття севдоеквівалентності станів.

*Псевдоеквівалентними* називаються такі два стійкі стани автомата, яким:

- відповідає один і той самий стан входу автомата;
- відповідають стани виходів автомата, між якими немає протиріччя;
- будь-якій послідовності станів входу автомата відповідають

непротирічні послідовності станів його виходу, незалежно від того, який з цих стійких станів взятий за початковий. Іншими словами, серед пар послідовностей станів входу і виходу, що починаються з цих стійких станів, не повинно бути таких, що мають протиріччя.

Після об'єднання еквівалентних і псевдоеквівалентних станів завжди отримується таблиця, в кожному рядку якої, як і в ПТП, знаходиться лише один стійкий стан. Скорочення кількості рядків, порівняно з ПТП, відбувається за рахунок зменшення кількості стійких станів.

Після вказаного об'єднання стійких станів можливе подальше спрощення таблиці переходів за рахунок об'єднання сумісних внутрішніх станів автомата.

Під *сумісними внутрішніми станами* розуміють два або більше станів, яким відповідають рядки з розміщенням цифр в них, яке не має протиріччя, тобто такі рядки, яким відповідають однакові цифри або риси в одних і тих же стовпцях таблиці переходів.

Рядки таблиці переходів, що відповідають сумісним станам, можуть бути об'єднані в один. В об'єднаному рядку стійкі стани можуть бути в будь-якому стовпчику, тобто, при об'єднанні сумісних станів отримуємо таблицю, в рядках якої може бути декілька стійких станів. При об'єднанні сумісних станів буде отриманий автомат, еквівалентний заданому.

Об'єднання рядків з поєднаними стійкими станами виконується на основі наступних правил:

- два або більше рядків можуть бути об'єднані, якщо значення вихідних змінних (станів виходу) для цих рядків співпадають для автомата Мура і можуть бути будь-якими для автомата Мілі, а номери станів, що записані в одних і тих же стовпцях, співпадають між собою, або з рискою;
- при об'єднанні станів з однаковими номерами в дужках і без них результируючий стан повинен бути у дужках (стійким);
- якщо при об'єднанні станів в одному з рядків знаходиться прочерк, а в іншому – номер стану, то в скороченій таблиці пишеться номер стану.

Слід відмітити, що виявляти і об'єднувати еквівалентні і псевдоеквівалентні стани перед виявленням і об'єднанням сумісних станів часто немає потреби. Можна починати з виявлення і об'єднання сумісних внутрішніх станів, при цьому об'єднуються, якщо вони є, еквівалентні і псевдоеквівалентні стани. Більш того, бувають випадки, коли першочергове об'єднання псевдоеквівалентних станів недовизначеного автомата може ускладнити остаточне рішення, тобто може бути отриманий автомат з більшою кількістю внутрішніх станів, порівняно з автоматом, для якого зразу об'єднувались сумісні внутрішні стани.

Суттєву допомогу при аналізі сумісності рядків ПТП можуть надати діаграми (графи) сумісності внутрішніх станів, на яких сумісні стани розміщуються у вузлах, з'єднаних ненаправленими лініями. Множина внутрішніх станів є *сумісною*, якщо всі її стани є попарно сумісними. Таку множину можна замінити одним станом. Вибираючи мінімальну кількість таких множин, які охоплюють всі стани без їх повторення, отримують мінімальну кількість станів, яка достатня для реалізації автомата.

Для забезпечення знаходження максимальних множин сумісних станів іноді використовується трикутна таблиця сумісності, з правилами побудови якої можна ознайомитись.

В якості прикладу розглянемо процедуру отримання скороченої (мінімальної) таблиці переходів з ПТП, що отримана для генератора з заборонаю (див. табл. 5.23). Аналіз цієї ПТП показує, що в ній відсутні еквівалентні і псевдоеквівалентні стани. Сумісними будуть наступні множини станів:  $\{ (0), (2) \}$ ;  $\{ (1), (4) \}$ ;  $\{ (2), (5) \}$ ;  $\{ (2), (3), (5) \}$  за Муром (показано суцільними лініями). Крім того сумісними за Мілі будуть стани  $(1)-(0)$ , та  $(4)-(0)$  (зображено пунктирними лініями)). Загальна діаграма сумісності зображена у вигляді ненаправлених ліній на рис. 5.17.

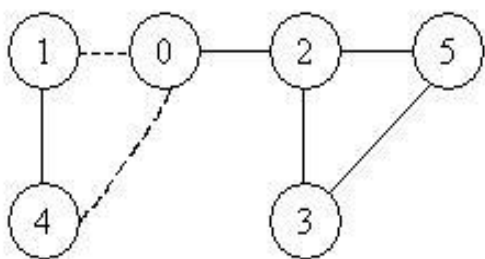


Рис.5.17

Розглянемо спочатку проєктований автомат, як автомат Мура.

Аналіз діаграми сумісності показує, що є можливість виділити дві групи множин сумісності:

1.  $\{ (0), (2) \}$ ;  $\{ (1), (4) \}$ ;  $\{ (3), (5) \}$ ;
2.  $\{ (0) \}$ ;  $\{ (1), (4) \}$ ;  $\{ (2), (3), (5) \}$ .

В обох випадках кількість множин дорівнює трьом. Це означає, що в скороченій таблиці переходів повинно бути три внутрішніх стани.

Скорочена таблиця переходів, яка побудована на основі першого варіанта об'єднання станів, приводиться в табл. 5.27.

Об'єднання рядків слід виконувати у декілька етапів. Спочатку об'єднані рядки зводяться в один, з меншим номером, і всі стани, що мають більші номери, замінюються меншими. Так, при об'єднанні рядків 0 і 2 двійки другого стовпця замінюються на нулі. Якщо в одному з об'єднаних рядків маємо невизначений стан, то при об'єднанні він замінюється визначеним (стовпець при  $x_1x_0=10$  має невизначений стан у другому рядку). Після об'єднання сумісних рядків виконується їх перейменування (в даному прикладі 3-й рядок замінюється на рядок з номером 2), а потім перенумеровуються всі відповідні стани новими цифрами (трійки замінюються на двійки).

Таблиця 5.27

Q \ X	$x_1 x_0$				y
	00	01	11	10	
(0, 2) 0	(0)	1	2	(0)	0
(1, 4) 1	0	(1)	(1)	0	1
(3, 5) 2	0	(2)	(2)	0	0

**Приклад 5.6.** Автомат задається часовими діаграмами, що приведені на рис. 5.18. Побудувати початкову таблицю переходів, діаграму сумісності внутрішніх станів та мінімальну таблицю переходів.

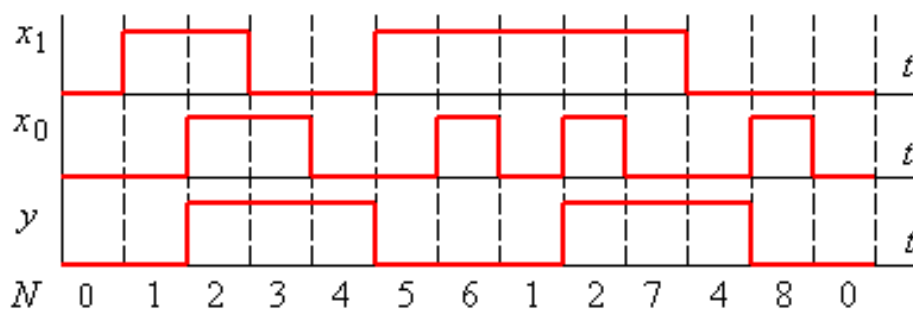


Рис.5.18

*Розв'язання.* Перенумеруємо всі комбінації вхідних сигналів  $x_1 x_0$  і вихідного  $y$ , які мають місце на інтервалі повторюваності. Таких комбінацій маємо 9. Будуємо ПТП (табл. 5.28). Створена ПТП заповнена не повністю, частина клітин залишається вільними, що характеризує відповідні переходи як невизначені або недопустимі. Аналізуючи створену ПТП, бачимо, що в ній відсутні еквівалентні та псевдоеквівалентні внутрішні стани.

Сумісними можуть бути стани (0) і (1); (0) і (6); (0) і (8); (1) і (8); (2) і (3) і т.п. На основі сумісності станів будується діаграма сумісності за Муром (рис. 5.19), в якій пунктиром зображена сумісність між станами для автомата Мілі.

Аналіз діаграми дає можливість виділити дві наступні групи множин сумісності:

1. { (0), (1), (8) }, { (2), (3), (7) }, { (4), (5) }, { (6) };
2. { (0), (6), (8) }, { (1), (3) }, { (2), (7) }, { (4), (5) }.

В обох випадках кількість множин дорівнює чотирьом. Це говорить про те, що у скороченій таблиці переходів матимемо чотири внутрішні стани. При цьому синтезований пристрій проектуватиметься як автомат Мілі, і для його реалізації знадобляться два елементи пам'яті.

Якщо ж розглядати пристрій як автомат Мура, тобто не враховувати сумісність, позначену пунктирними лініями, то матимемо такі дві групи множин:

1. { (0), (1), (8) }, { (2), (3), (7) }, { (4) }, { (5) }, { (6) };
2. { (0), (6), (8) }, { (2), (3), (7) }, { (1) }, { (4) }, { (5) }.

Таблиця 5.28

$Q \backslash X$	$x_1 x_0$				$y$
	00	01	11	10	
0	(0)			1	0
1			2	(1)	0
2		3	(2)	7	1
3	4	(3)			1
4	(4)	8		5	1
5			6	(5)	0
6			(6)	1	0
7	4			(7)	1
8	0	(8)			0

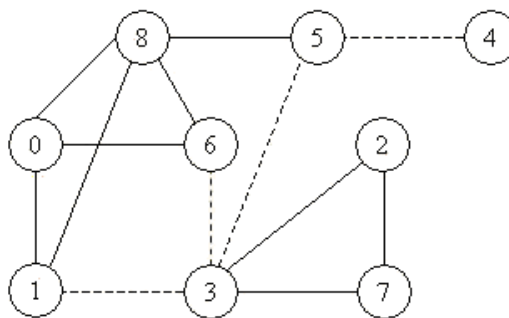


Рис.5.19

В обох випадках мінімальна кількість множин сумісності дорівнює 5, а для реалізації автомата знадобиться вже три елементи пам'яті.

Скорочена таблиця переходів, побудована на основі першого варіанту об'єднання (при проектуванні як автомата Мілі) приведена в табл. 5.29.

Таблиця 5.29

$Q \backslash X$	$x_1 x_0$			
	00	01	11	10
$\frac{(0), (1), (8)}{0}$	(0),0	(0),0	1	(0),0
$\frac{(2), (3), (7)}{1}$	2	(1),1	(1),1	(1),1
$\frac{(4), (5)}{2}$	(2),1	3	3	(2),0
$\frac{(6)}{3}$	—	0	(3),0	0



У приведеній таблиці множина  $\{ (0), (1), (8) \}$  зображена у вигляді одного стану 0; множина  $\{ (2), (3), (7) \}$  – зображена станом 1; множина  $\{ (4), (5) \}$  – відповідно, станом 2; множина  $\{ (6) \}$  – станом 3. Оскільки у автоматів Мілі стійкий внутрішній стан при різних вхідних сигналах може мати різний вихідний сигнал, тому в табл. 5.29 значення вихідного сигналу зображені в клітинах стійких внутрішніх станів через кому.

### Етап 3. Кодування станів автомату.

Найважливішою задачею, що має місце при кодуванні станів автомата є виключення «гонок» його елементів пам'яті при заданих переходах. Таке кодування називається протигоночним. Найпростіший спосіб протигоночного кодування є сусіднє кодування, при якому два стани, що пов'язані між собою простими переходами кодуються наборами двійкових чисел, що відрізняються станом лише одного ЕП (кодом Грея). Основними вимогами до автомата, в якому забезпечується сусіднє кодування станів є:

- в графі переходів автомата не повинно бути замкнутих контурів, що містять непарну кількість вершин;
- два сусідні стани другого порядку не повинні мати більше двох станів, що лежать між ними. При цьому під станами другого порядку маються на увазі два стани, шлях між якими по графу переходів автомата складається з двох ребер (незалежно від орієнтації). На рис. 5.20 зображені два графа, які не задовольняють вказаним вимогам і не можуть бути закодовані сусідніми кодами.

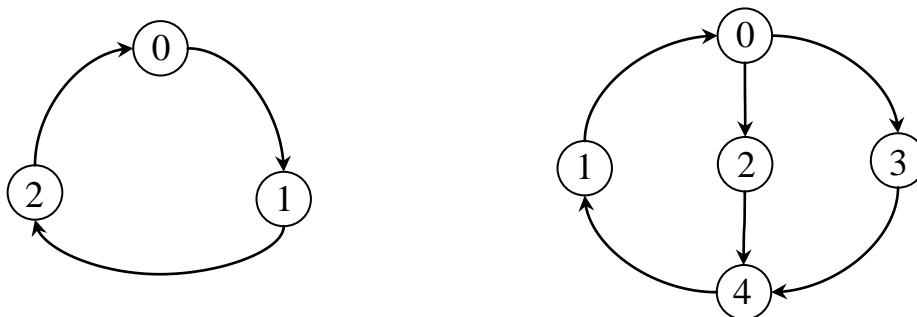


Рис. 5.20

З більш загальними методами протигоночного кодування можна ознайомитися в спеціальній літературі.

Граф переходів для генератора з заборорою (Табл.5.27) зображено на рис.5.21. В розглянутому випадку граф має непарну кількість вершин, але контур незамкнений. Один з варіантів кодування станів приводить до табл.5.30.

Після закінчення кодування станів будується закодована таблиця переходів автомата, що проектується. Для цього в скороченій таблиці переходів замінюються десяткові номери станів їх двійковими кодовими значеннями. табл. 5.27 автомата, що розглядається, прийме вигляд, зображений у табл. 5.31.

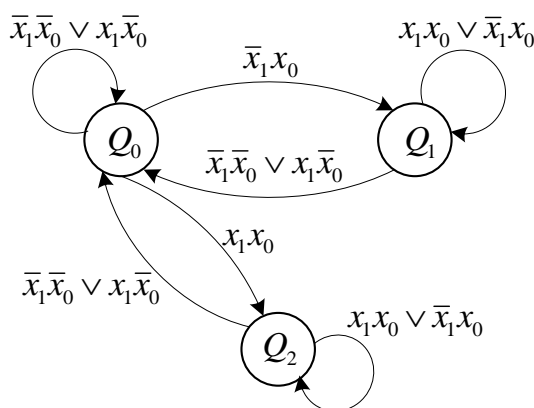


Рис.5.21

Таблиця 5.30

Стани	Код	
	q <sub>1</sub>	q <sub>0</sub>
Q <sub>0</sub>	0	0
Q <sub>1</sub>	0	1
Q <sub>2</sub>	1	0

Таблиця 5.31

Q	X		X <sub>1</sub> X <sub>0</sub>				Y
	q <sub>1</sub>	q <sub>0</sub>	00	01	11	10	
0	0	0	00	01	10	00	0
0	1	0	00	01	01	00	1
1	0	0	00	10	10	00	0

**Приклад 5.7.** Побудувати граф переходів і виконати сусіднє кодування для автомата, скорочена таблиця переходів якого відповідає табл.5.29.

*Розв'язання.* Граф переходів, побудований за табл.5.29 зображений на рис.5.22. Закодуємо стани в відповідності до табл. 5.32.

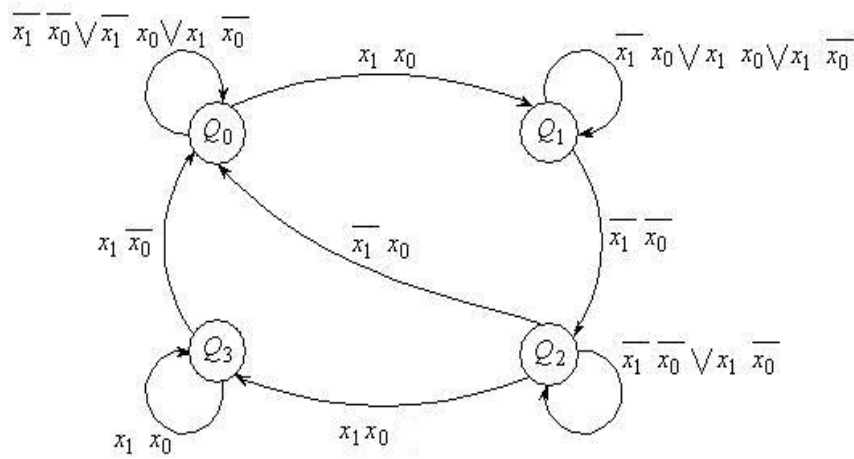


Рис. 5.22

При такому кодуванні маємо перехід з Q2 в Q0 тобто з 11 в 00, який не відповідає умові сусіднього кодування.

Для розв'язання такої проблеми перехід з Q2 в Q0 необхідно замінити більш складним, через Q3 зробивши його нестійким станом для даної комбінації вхідних сигналів. Граф-схема автомата прийме вигляд, зображений на рис. 5.23.

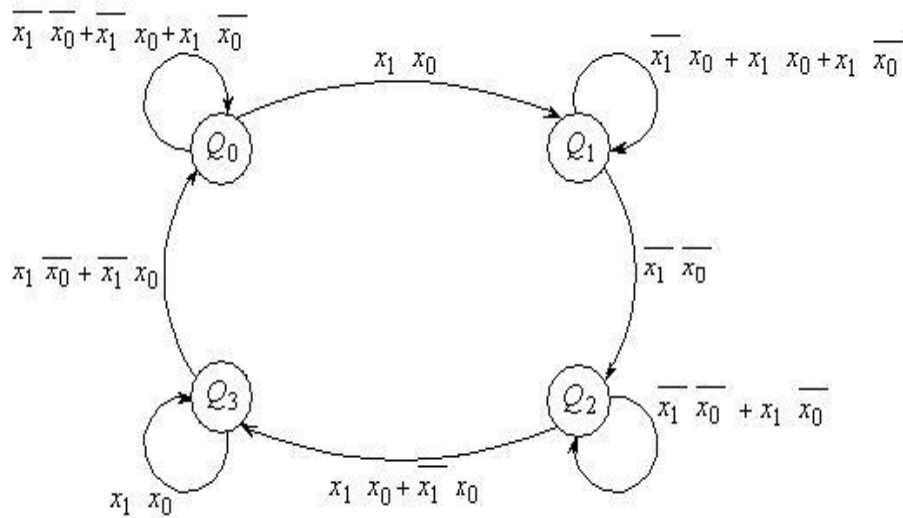


Рис.5.23

Для забезпечення такої зміни графу необхідно зробити відповідні зміни в скороченій таблиці переходів при  $Q=2$  і  $x_1x_0=01$  замінити стан 0 на стан 3, а при  $Q=3$  і  $x_1x_0=01$  показати перехід зі стану 3 в стан 0.

Трансформована, з врахуванням умови сусіднього кодування станів, скорочена таблиця переходів табл. 5.29 в закодованому варіанті прийме вигляд табл. 5.33.

**Етап 4.** Визначення функцій збудження елементів пам'яті і функцій виходів автомата.

Функції збудження ЕП автомата залежать від використовуваних типів тригерів, що вибираються в їх якості.

Таблиця 5.32

Стани	Код	
	q <sub>1</sub>	q <sub>0</sub>
Q <sub>0</sub>	0	0
Q <sub>1</sub>	0	1
Q <sub>2</sub>	1	1
Q <sub>3</sub>	1	0

Таблиця 5.33

Q \ X		X <sub>1</sub> X <sub>0</sub>			
		00	01	11	10
q <sub>1</sub>	q <sub>0</sub>				
0	0	00	00	01	00
0	1	11	01	01	01
1	1	11	10	10	11
1	0	–	00	10	00

Розглянемо послідовно на конкретних прикладах особливості визначення функцій збудження ЕП для кожного з типів тригерів.

Таблиця переходів RS-тригера приводиться в табл.4.2 , а характеристична таблиця має вигляд, зображений в табл. 5.34, де зірочкою (\*) зображені довільні значення відповідної змінної.

Визначимо функції збудження RS-тригерів. Для цього скористаємось закодованою таблицею переходів автомата (табл. 5.31), в якій замість переходів  $Q_n \rightarrow Q_{n+1}$  підставимо значення сигналів збудження відповідних входів тригерів. Тоді вона прийме вигляд, зображений у табл. 5.35.

Таблиця 5.34

q <sub>n</sub>	q <sub>n+1</sub>	R <sub>n</sub>	S <sub>n</sub>
0	0	*	0
0	1	0	1
1	0	1	0
1	1	0	*

Таблиця 5.35

$X$ $Q_n$	$x_1 x_0$															
	00				01				11				10			
$q_1 q_0$	$R_1$	$S_1$	$R_0$	$S_0$	$R_1$	$S_1$	$R_0$	$S_0$	$R_1$	$S_1$	$R_0$	$S_0$	$R_1$	$S_1$	$R_0$	$S_0$
0 0	*	0	*	0	*	0	0	1	0	1	*	0	*	0	*	0
0 1	*	0	1	0	*	0	0	*	*	0	0	*	*	0	1	0
1 1	—				—				—				—			
1 0	1	0	*	0	0	*	*	0	0	*	*	0	1	0	*	0

Заповнення клітин табл. 5.35 забезпечується безпосереднім використанням табл. 5.34. Розглянемо, наприклад, заповнення клітин з координатами  $q_{1n} q_{0n} = 00$  і  $x_1 x_0 = 00$ . В цих клітинах маємо значення  $q_{1(n+1)} q_{0(n+1)} = 00$ . Оскільки  $q_1$  і  $q_0$  – це значення прямих виходів двох тригерів, один з яких має входи  $R_1, S_1$ , другий –  $R_0, S_0$ , то, відповідно до табл. 5.34, значення входів  $R_1 S_1, R_0 S_0$  тригерів повинні відповідати незмінному  $q_1 = q_0 = 0$  стану кожного тригера. Такі значення беруться з першого рядка табл. 5.34 і вписуються в клітини з координатами  $q_{1n} q_{0n} = 00$  і  $x_1 x_0 = 00$  для кожного з входів тригерів. Для клітин з координатами  $q_{1n} q_{0n} = 00$  і  $x_1 x_0 = 11$  маємо  $q_{1(n+1)} = 1$ , тому з табл. 5.34 вибирається значення  $R, S$  з другого рядка, значення  $q_{0(n+1)}$  залишається незмінним на нульовому рівні, тому використовуємо перший рядок. Так послідовно заповнюються всі клітини таблиці функцій збудження.

Для отримання функції збудження для кожного з входів тригерів переносимо їх значення на карти Карно (рис. 5.24, а – г).

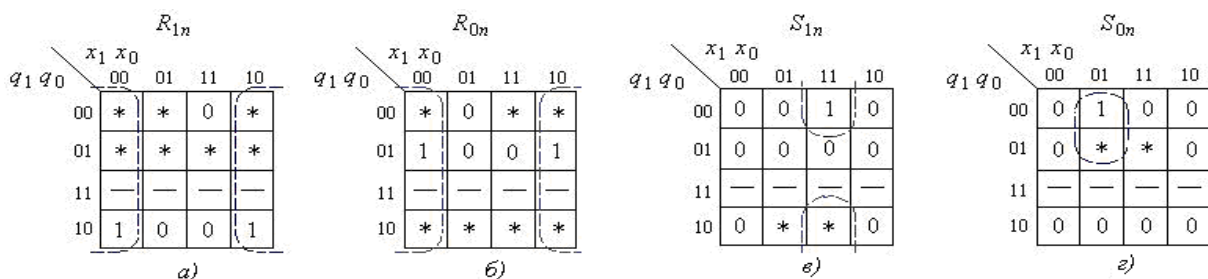


Рис.5.24

Мінімізуючи кожну з функцій збудження входів, отримуємо:

$$R_{1n} = \overline{x_0}; R_{0n} = \overline{x_0}; S_{1n} = \overline{q_0} x_1 x_0; S_{0n} = \overline{q_1} \overline{x_1} x_0.$$

Для визначення функції виходу карта Карно будується на основі скороченої таблиці переходів табл. 5.27, в якій стійкі стани замінюються значеннями вихідного сигналу, а нестійкі з символом \* – довільним значенням функції. В результаті отримуємо карту, зображену на рис. 5.25.

З неї отримуємо мінімізоване значення функції виходу:  $y = q_0$ .

**Етап 5.** Складання функціональної (принципової) схеми автомата в вибраному елементному базисі.

Отримані на попередньому етапі вирази функцій збудження ЕП і функції виходу дають можливість легко розробити функціональну і схему автомата, яка приведена на рис. 5.26.

Розглянемо роботу синтезованого автомату у відповідності до часової діаграми, що приведена на рис. 5.15.

У початковому стані при відсутності вхідних сигналів значення  $q_0 = q_1 = 0$  В першому стані  $x_1 = 0$ , а  $x_0 = 1$ , тому, враховуючи, що  $\overline{q_1} = 1$ , сигнал  $S_{0n}$  з виходу DD1 буде поданий на тригер DD2 і встановить його вихід  $q$  в 1. При повторенні нульового стану тригер DD1 установиться в початковий стан нульовим значенням вхідного сигналу  $x_0$ .

$q_1 q_0$ \ $x_1 x_0$	00	01	11	10
00	0	*	*	0
01	*	1	1	*
11	—	—	—	—
10	*	0	0	*

Рис.5.25

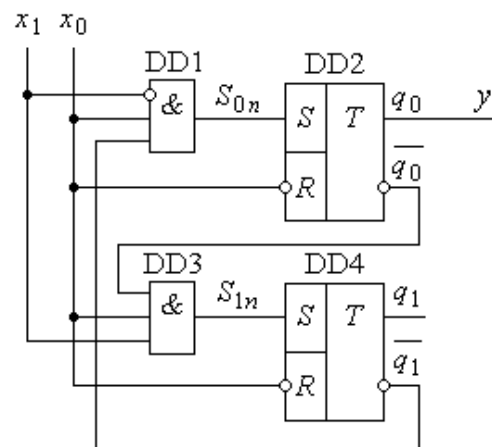


Рис.5.26

В другому – третьому станах при  $x_1 = 1$  на вході DD1 формується сигнал заборони для сигналу  $x_0$  і т.д. З цього короткого опису бачимо, що функціональна схема повністю відтворює часову діаграму, за якою спроектований автомат.

**Приклад 5.8.** Для автомата, заданого закодованою таблицею переходів (табл. 5.33), розробити функціональну схему з використанням *JK*-тригерів асинхронного типу.

*Розв'язання.* Таблиця переходів *JK*-тригера приведена в Розділі IV, п. 4.5. На її основі побудуємо характеристичну таблицю (табл. 5.36).

Таблиця 5.36

$q_n \rightarrow q_{n+1}$	$J_n$	$K_n$
0 0	0	*
0 1	1	*
1 0	*	1
1 1	*	0

Розробимо таблицю для визначення функції збудження тригерів автомата. Вона будується аналогічно табл. 5.35. Після підстановки в табл. 5.33 значень входів тригерів з характеристичної таблиці (табл. 5.36) отримуємо табл. 5.37.

Таблиця 5.37

$X$ $Q_n$	$x_1 x_0$															
	00				01				11				10			
$q_1 q_0$	$J_1$	$K_1$	$J_0$	$K_0$	$J_1$	$K_1$	$J_0$	$K_0$	$J_1$	$K_1$	$J_0$	$K_0$	$J_1$	$K_1$	$J_0$	$K_0$
0 0	0	*	0	*	0	*	0	*	0	*	1	*	0	*	0	*
0 1	1	*	*	0	0	*	*	0	0	*	*	0	0	*	*	0
1 1	*	0	*	0	*	0	*	1	*	0	*	1	*	0	*	0
1 0	—	—	—	—	*	1	0	*	*	0	0	*	*	1	0	*

Карта функції виходу будується на основі табл. 5.29 (рис. 5.27, а).

Заповнюємо карти Карно для функцій збудження *JK*-тригерів (рис. 5.27, б – д).

Мінімізовані функції збудження тригерів мають вигляд:

$$J_{1n} = q_0 \bar{x}_1 \bar{x}_0; \quad J_{0n} = \bar{q}_1 x_1 x_0; \quad K_{0n} = q_1 x_0; \quad K_{1n} = \bar{q}_0 (\bar{x}_1 + \bar{x}_0);$$

З карти Карно функції виходу отримуємо:

$$y = q_0 (\bar{q}_1 + \bar{x}_1).$$

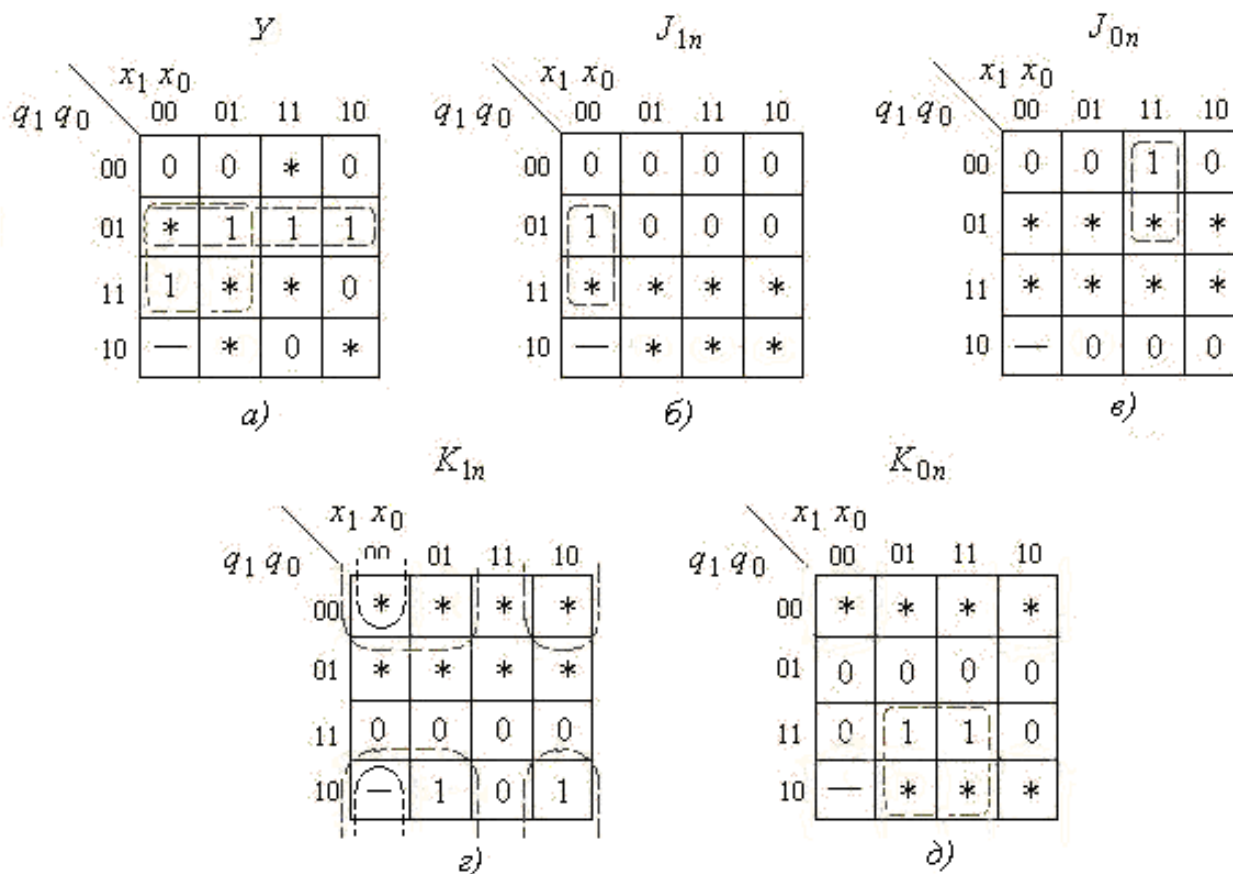


Рис.5.27

За отриманими функціями будуватиметься функціональна схема (рис. 5.28).

Важливим етапом синтезу автоматів є формалізація його опису за допомогою часових діаграм. З цією метою розглянемо приклад синтезу автомата, функціонування якого задається словесним описом.

**Приклад 5.9.** Задача про світлофор.

Розглянемо завантажену автомагістраль, що перетинається сільською дорогою, по якій іноді проїжджають автомобілі. Якщо на перехресті відсутній світлофор, то автомобіль, що рухається по сільській дорозі, не зможе пересікти магістраль через безперервний потік транспорту по ній. Нераціонально встановлювати світлофор періодичної дії, оскільки він буде в більшості безпідставно гальмувати рух по автомагістралі.



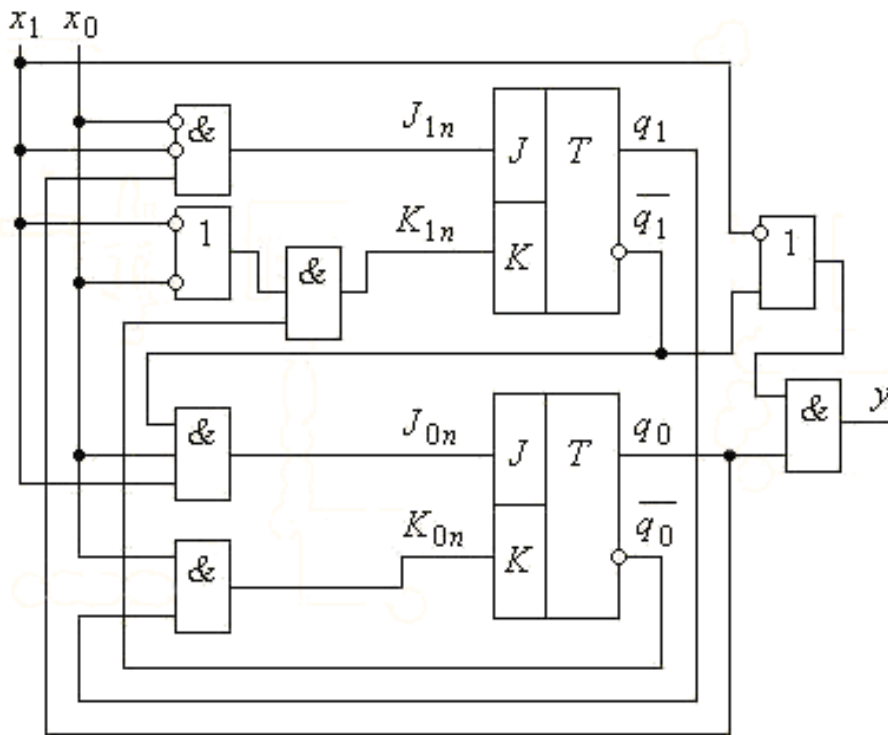


Рис.5.28

Для забезпечення регулювання руху на перехресті необхідно встановити світлофор, який повинен зупиняти рух по автомагістралі тільки в тому випадку, коли біля перехрестя на сільській дорозі зупиняється автомобіль. Рух по автомагістралі забезпечується при зеленому кольорі світлофора, а зупинка – при червоному.

Необхідно розробити схему керування світлофором, яка сприймає сигнали від двох датчиків – реле часу і датчика тиску. Реле часу працює в періодичному режимі, формуючи сигнал  $x_0 = 1$  протягом 30 с і  $x_0 = 0$  протягом 60 с. (Часові співвідношення можуть змінюватись). Червоний сигнал світлофора для автомобілів, що рухаються по автомагістралі, а зелений – для автомобілів сільської дороги може вмикатися лише на інтервалі  $x_0 = 1$  (30 с). Датчик тиску формує сигнал  $x_1 = 1$  при наявності стоячого автомобіля на сільській дорозі перед перехрестям і  $x_1 = 0$  при його відсутності.

*Розв'язання.* Виконаємо послідовно кроки 1 – 5, які приведуть до синтезу необхідного скінченного автомата.

### **Етап 1. Формалізація закону функціонування автомата.**

Побудова ПТП автомата можлива при одночасній або попередній побудові часових діаграм його роботи, на яких необхідно відобразити всі можливі ситуації на перехресті (рис. 5.29).

Допустимо, що часовий сигнал від реле часу відсутній, тобто  $x_0 = 0$ , і автомобілі на сільській дорозі відсутні ( $x_1 = 0$ ). Тоді  $y = 0$  (див. рис. 5.29, а). Позначимо внутрішній стан для такої ситуації 0 (00, 0) (перша цифра вказуватиме на номер внутрішнього стану, а три інші – в дужках – значення входів  $x_1$  та  $x_0$  і значення  $y$ ). Оскільки цей стан є стійким, то його номер виділимо дужками.

Таким шляхом є можливість побудови ПТП. Оскільки автомат світлофора має лише дві вхідні змінні, то його ПТП має чотири стовпці (табл. 5.38).

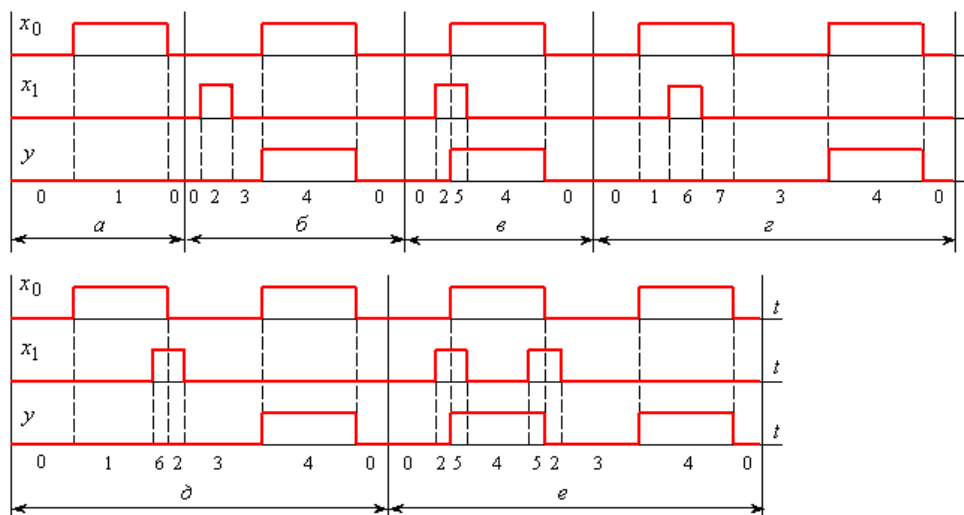


Рис.5.29

Таблиця 5.38

$Q$	$X$				$y$
	$x_1 x_0$	00	01	11	
0	(0)	1	—	2	0
1	0	(1)	6	—	0
2	3	—	5	(2)	0
3	(3)	4	—	—	0
4	0	(4)	5	—	1
5	—	4	(5)	2	1
6	—	7	(6)	2	0
7	3	(7)	—	—	0

Якщо автомат знаходиться в стійкому стані (0), а реле часу змінить вихідний сигнал на  $x_0 = 1$ , то вихідний сигнал  $y$  повинен залишатися рівним 0. Звідси маємо наступний стійкий стан (1) при (01, 0). Для того, щоб автомат перейшов зі стійкого стану (0) в (1), необхідно в нульовому рядку при  $x_1 x_0 = 01$  поставити 1 без дужок. Цим забезпечується простий перехід з нульового стану в перший при зміні комбінації вхідних сигналів  $x_1 x_0 00 \rightarrow 01$  (рис. 5.29, а).

Допустимо тепер, що сигнал  $x_0$  повертається до попереднього значення  $x_0 = 0$ , коли автомат знаходиться у стійкому стані (1). Повернення в стійкий стан (0) відбувається через нестійкий стан 0, що встановлюється при  $Q = 1$  і  $x_1 x_0 = 00$ .

Розглянемо тепер іншу ситуацію (рис. 5.29, б), коли автомат знаходиться в стійкому стані (0) і перед перехрестям на сільській дорозі з'являється автомобіль. Перш за все, необхідно визначитись з характером сигналу  $x_1$ . Найбільш реально, що це повинен бути короткочасний імпульс (прийемо його одиничним), тривалість якого визначається тривалістю положення колеса автомобіля, що рухається, на датчику тиску. Тобто тривалість його набагато менша тривалості імпульсу  $x_0$ . При появі короткочасного імпульсу  $x_1 = 1$  сигнал  $y$  також повинен залишатися рівним 0, оскільки  $x_0 = 0$ . Такій ситуації припишемо стійкий стан (2), який покажемо в ПТП при  $Q = 2$  і  $x_1 x_0 = 10$ .

Нехай автомат знаходиться у стані (2) і  $x_1$  змінюється з 1 в 0, тобто маємо перехід  $10 \rightarrow 00$ . Але це не значить, що за цим слідує зміна стану (2) в (0), оскільки в такому випадку не буде врахований факт наявності стоячого біля перехрестя автомобіля. Звідси витікає, що зі стану (2) автомат повинен перейти в новий стан (3), при якому  $y = 0$ . Особливістю стану (3) є те, що протягом наступного дозволяючого інтервалу від реле часу вихідний сигнал  $y$  повинен прийняти значення 1 ( $y = 1$ ). Такий стан позначимо (4) (рис. 5.29, б). В ПТП заповнюється рядок з  $Q = 4$ , в тому числі і клітина першого стовпця, оскільки при переході  $x_1 x_0 = 01 \rightarrow 00$  автомат перейде в стан (0).

Розглянемо тепер дещо іншу ситуацію, яка полягає в тому, що при появі автомобіля на сільській дорозі ( $x_1 = 1$ ) безпосередньо перед початком дозволяючого часового інтервалу ( $x_0 = 1$ ) він повинен бути пропущений через перехрестя ( $y = 1$ ). На рис. 5.29, в така послідовність відображається відповідно зміною станів:

$$(0) (00, 0) \rightarrow (2) (10, 0) \rightarrow (5) (11, 1) \rightarrow (4) (01, 1) \rightarrow (0) (00, 0).$$

У таблиці ПТП з'явиться рядок з  $Q = 5$  з відповідними переходами.

На рис. 5.29, г маємо ситуацію, яка полягає в тому, що автомобіль, з'явившись перед перехрестям на протязі поточного дозволяючого інтервалу часу, буде пропущений світлофором на наступному дозволяючому інтервалі. Така ситуація характеризується наступною зміною станів:

$$(0) (00, 0) \rightarrow (1) (01, 0) \rightarrow (6) (11, 0) \rightarrow (7) (01, 0) \rightarrow (3) (00, 0) \rightarrow (4) (01, 1) \rightarrow (0).$$

У даному випадку вводяться два нові стани – (6), оскільки він відрізняється від (5) тим, що  $y=0$ , і (7), який відрізняється від (1) тим, що інформація про появу автомобіля повинна зберегтись у пам'яті автомата. Відповідні зміни робляться і в ПТП, тобто зі стану (1) при  $x_1 x_0 = 11$  виникає перехід у нестійкий стан 6, з якого при тій же комбінації вхідних сигналів відбувається перехід у стійкий стан (6). Зі стійкого стану (6) при  $x_1 x_0 = 01$  відбувається перехід у нестійкий стан 7, а потім в стійкий стан (7); і, нарешті, при  $x_1 x_0 = 00$  зі стану (7) автомат переходить в нестійкий, а потім в стійкий стан (3).

Наступна ситуація характеризується тим, що автомобіль, який з'явиться біля перехрестя в кінці поточного дозволяючого інтервалу часу, повинен бути пропущений у наступному дозволяючому інтервалі часу (рис. 5.29, д). Робота автомата характеризується послідовністю зміни станів:

$$(0) (00, 0) \rightarrow (1) (01, 0) \rightarrow (6) (11, 0) \rightarrow (2) (10, 0) \rightarrow (3).$$

У ситуації, що розглядається, нові стани не вводяться, але з'являється новий перехід (6)  $\rightarrow$  (2) при вхідній комбінації  $x_1 x_0 = 10$ , що і відображено в ПТП появою нестійкого стану 2 в рядку з  $Q = 6$  при  $x_1 x_0 = 10$ .

Насамкінець, розглянемо ситуацію, яка характеризується тим, що до перехрестя наближаються два автомобілі, причому перший з них попадає на початок поточного дозволяючого інтервалу, а другий – на його кінець. Відповідні часові діаграми приведені на рис. 5.29, е. Аналогічно попередній ситуації, маємо новий перехід (5)  $\rightarrow$  (2) при  $x_1 x_0 = 10$ , що і відображено в ПТП появою нестійкого стану 2 в рядку  $Q = 5$  при  $x_1 x_0 = 10$ , а також перехід (4)  $\rightarrow$  (5), який заноситься в ПТП.

Якщо не передбачається інших ситуацій, то можна вважати, що побудова ПТП закінчена. Незаповнені клітини заповнюються рисками.

### **Етап 2. Мінімізація кількості станів автомата.**

Аналіз ПТП показує, що деякі стани є сумісними. Такими є стани 0 і 1; 2 і 3; 4 і 5; 6 і 7 (за Муром). Об'єднуючи ці стани і вводячи нові позначення:

$$0, 1 \rightarrow (0); 2, 3 \rightarrow (1);$$

$$4, 5 \rightarrow (2); 6, 7 \rightarrow (3),$$

отримуємо скорочену таблицю переходів (табл. 5.39).

Таблиця 5.39

$Q \backslash X$	$x_1 x_0$				$y$
	00	01	11	10	
$(0), (1)$ 0	(0)	(0)	3	3	0
$(2), (3)$ 1	(1)	2	2	(1)	0
$(4), (5)$ 2	0	(2)	(2)	1	1
$(6), (7)$ 3	1	(3)	(3)	1	0

**Етап 3. Кодування станів автомата.**

Перш ніж кодувати стани автомата, побудуємо його граф переходів, який допоможе визначитись з можливістю сусіднього кодування. Граф-схема автомата приводиться на рис. 5.30.

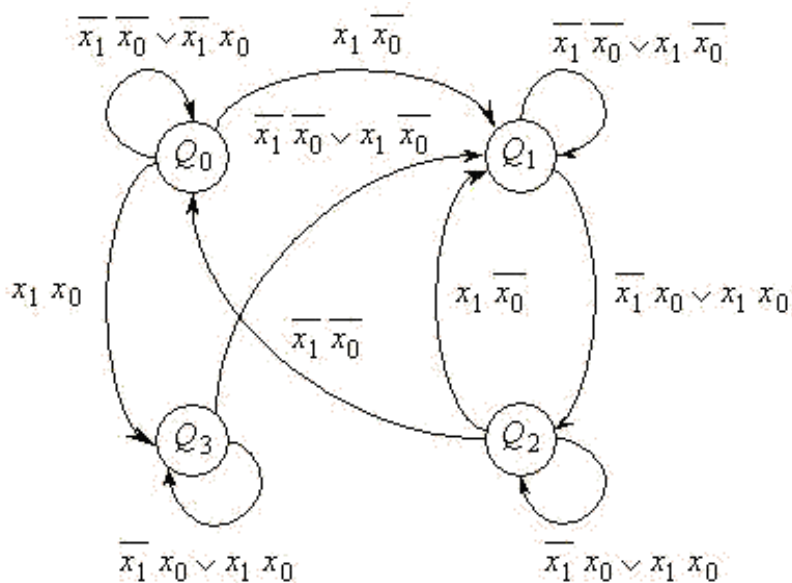


Рис.5.30

З вигляду граф-схеми бачимо, що в ній є замкнуті контури з непарною кількістю вершин  $(Q_0 \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_0)$ ,  $(Q_0 \rightarrow Q_3 \rightarrow Q_1)$ . Але є можливість замінити простий перехід  $Q_0 \rightarrow (x_1 \bar{x}_0) \rightarrow Q_1$  складним  $Q_0 \rightarrow (x_1 \bar{x}_0) \rightarrow Q_3 \rightarrow (x_1 \bar{x}_0) \rightarrow Q_1$  через проміжний стан  $Q_3$ , що дає можливість отримати граф, зображений на рис. 5.31. Вказана заміна не порушує закону функціонування автомата, оскільки стан (3) для  $x_1 \bar{x}_0$  є нестійким.

Змінивши граф-схему автомата, внесемо корекцію в таблицю переходів. У ній необхідно передбачити перехід з  $Q_0$  в  $Q_3$  при  $x_1 \bar{x}_0$ , тобто в відповідній клітині внести заміну стану (1) на (3). Завершивши перекодування таблиці переходів, виконаємо сусіднє кодування автомата (табл. 5.40). При цьому для побудови автомата достатньо всього двох тригерів. З граф-схеми автомата (кодові значення станів під їх позначеннями) бачимо, що будь-який перехід з одного стану в інший супроводжується зміною лише одного розряду коду.

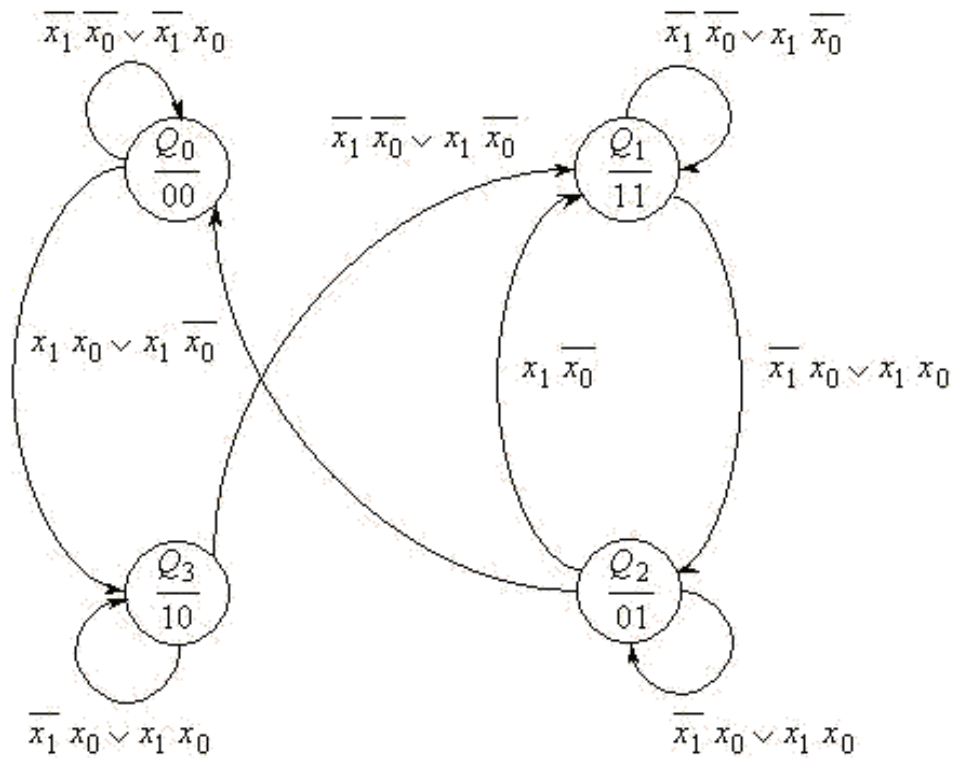


Рис.5.31

Суміщаючи табл. 5.39 і табл. 5.40, отримаємо кодову таблицю переходів, що приведена в табл. 5.41.

Таблиця 5.40

$Q_i$	Коди станів	
	$q_1$	$q_0$
$Q_0$	0	0
$Q_1$	1	1
$Q_2$	0	1
$Q_3$	1	0

Таблиця 5.41

$Q$		$X$			
		$x_1 x_0$			
$q_1$	$q_0$	00	01	11	10
0	0	00	00	10	10
0	1	00	01	01	11
1	1	11	01	01	11
1	0	11	10	10	11

#### Етап 4. Визначення функцій збудження ЕП і функції виходів автомата.

Для реалізації автомата виберемо асинхронні RS-тригери, а комбінаційну частину реалізуватимемо в базисі І-НІ.

Визначимо функції збудження входів тригерів. Для цього табл. 5.41 з врахуванням значень, заданих в характеристичній табл. 5.34, перетворюється в таблицю функцій збудження входів тригерів (табл. 5.42).

Таблиця 5.42

X Q	x <sub>1</sub> x <sub>0</sub>															
	00				01				11				10			
q <sub>1</sub> q <sub>0</sub>	R <sub>1</sub>	S <sub>1</sub>	R <sub>0</sub>	S <sub>0</sub>	R <sub>1</sub>	S <sub>1</sub>	R <sub>0</sub>	S <sub>0</sub>	R <sub>1</sub>	S <sub>1</sub>	R <sub>0</sub>	S <sub>0</sub>	R <sub>1</sub>	S <sub>1</sub>	R <sub>0</sub>	S <sub>0</sub>
0 0	*	0	*	0	*	0	*	0	0	1	*	0	0	1	*	0
0 1	*	0	1	0	*	0	0	*	*	0	0	*	0	1	0	*
1 1	0	*	0	*	1	0	0	*	1	0	0	*	0	*	0	*
1 0	0	*	0	1	0	*	*	0	0	*	*	0	0	*	0	1

Будуємо функції збудження входів тригерів у вигляді карт Карно (рис. 5.32, а – з) і функцію виходу (рис. 5.32, д):

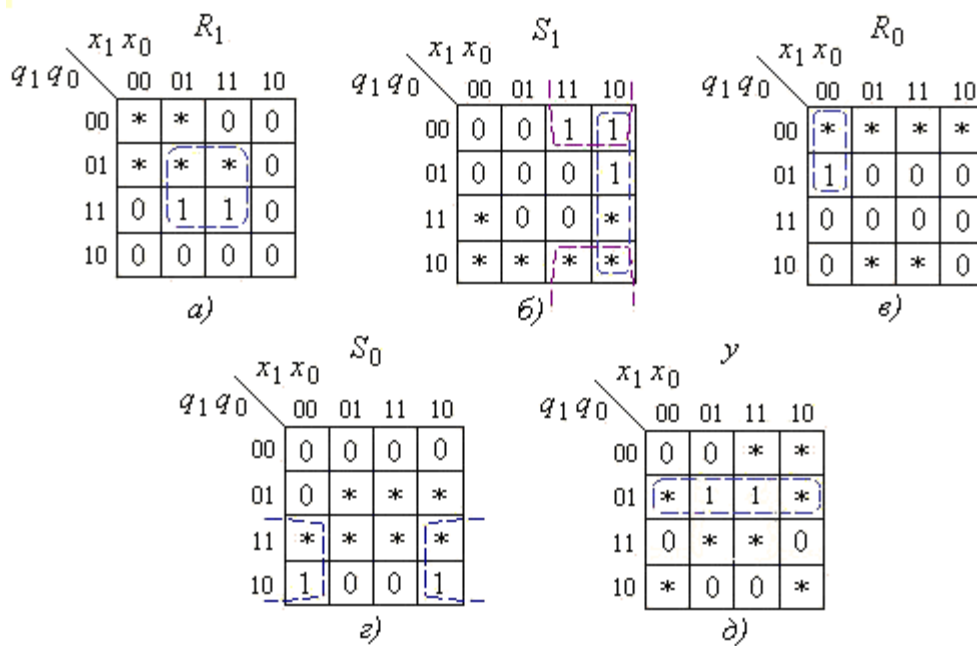


Рис.5.32

Мінімізовані функції збудження входів автомата:

$$R_1 = q_0 x_0; \quad S_1 = x_1 \bar{x}_0 + x_1 \bar{q}_0; \quad R_0 = \bar{q}_1 \bar{x}_1 \bar{x}_0; \quad S_0 = q_1 \bar{x}_0.$$

Функція виходу автомата:

$$y = \bar{q}_1 \bar{q}_0.$$

**Етап 5. Побудова функціональної схеми у вибраному елементному базисі.**

Функціональна схема в базисі **I-НІ** реалізується з урахуванням того, що

$$S_1 = x_1 (\overline{x_0} + \overline{q_0}) = x_1 \overline{x_0 q_0}.$$

Схема приводиться на рис. 5.33.

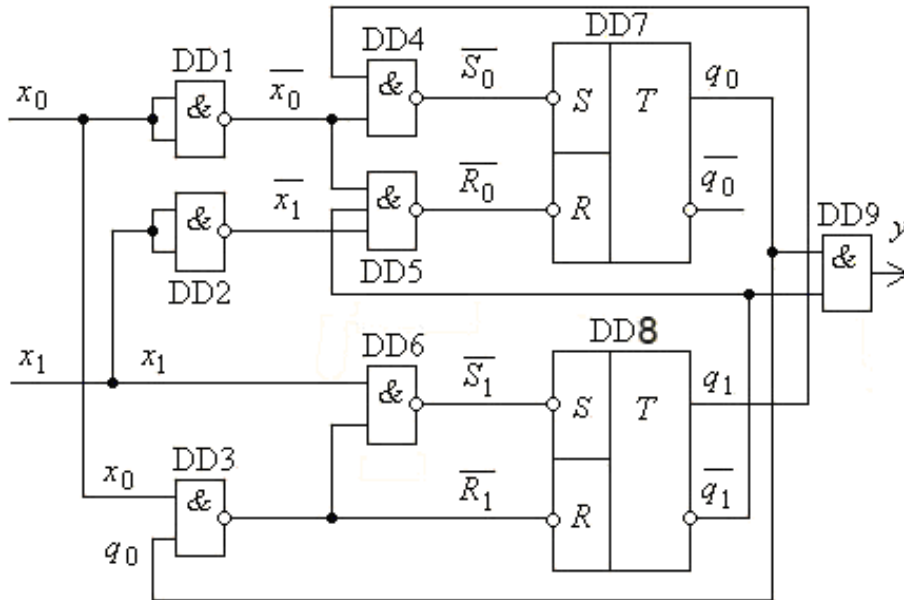


Рис.5.33

Проаналізуємо лише одну з ситуацій, що описувались вище. Допустимо, що подається сигнал  $x_1=1$  при  $x_0=0$ , тобто з'явився автомобіль на сільській дорозі до появи дозволяючого інтервалу часу (рис. 5.29,б). Враховуючи, що початковий стан для даної ситуації  $Q_0(q_1=q_0=0)$ , на виході ЛЕ DD6 виникне сигнал  $\overline{S}_1$  низького рівня, який встановить тригер DD8 в стан  $q_1=1$ . Оскільки  $x_0=0$ , то сигнал  $q_1=1$  через ЛЕ DD4 встановить тригер DD7 в стан  $q_0=1$  і автомат перейде в стійкий стан  $q_1q_0=11$ .

Наступна зміна сигналів  $x_1$  та  $x_0$  до низького рівня ( $x_1=x_0=0$ ) не змінить стану тригерів автомата.

З приходом сигналу  $x_0=1$  на інформаційних входах тригерів встановлюються значення  $\overline{S}_0=\overline{R}_0=1$ ,  $\overline{S}_1=1$ ,  $\overline{R}_1=0$  і автомат перейде в стан  $q_1q_0=01$ , який буде супроводжуватись вихідним сигналом  $y=1$ .

Завершиться ситуація встановленням сигналів  $x_1=x_0=0$  та переходом автомату в початковий стан  $q_1=q_0=0$ .



Розглянутий приклад, як видно, об'єднує в собі всі етапи проектування автомата. Оскільки вони достатньо формалізовані, то найбільшу складність має уміння чітко формалізувати задачу, проаналізувати всі можливі режими роботи автомата і пов'язати їх між собою.

**Приклад 5.10.** Побудувати систему керування ключами перетворювача напруги, що зображений на рис. 5.34.

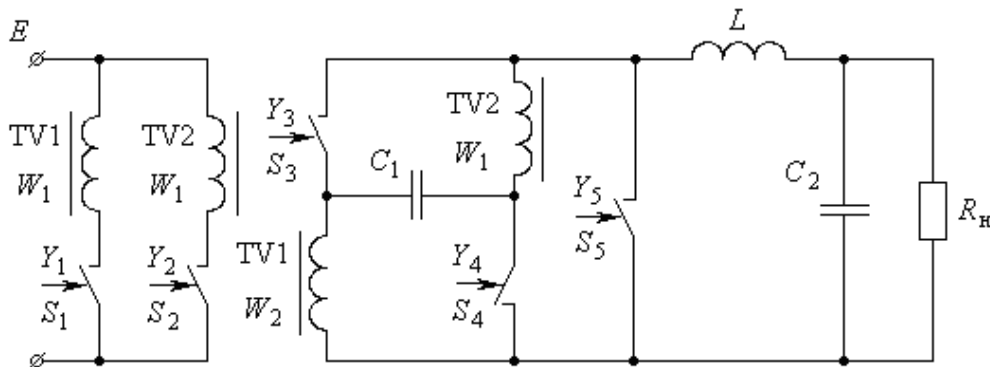


Рис.5.34

Часові діаграми роботи ключів приведені на рис. 5.35.

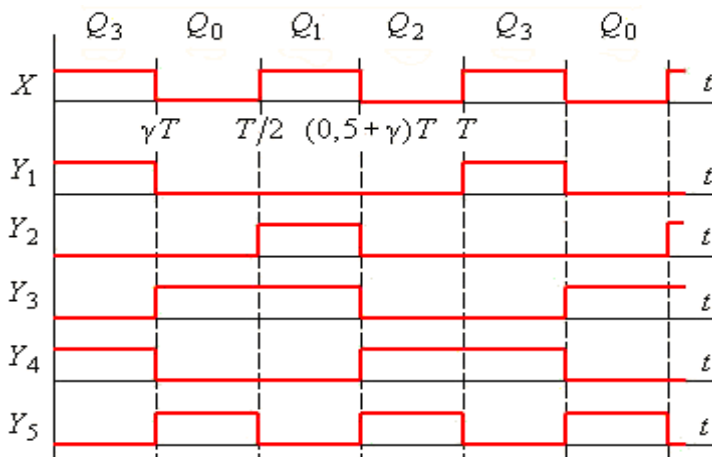


Рис.5.35

*Розв'язання.* Розглянемо систему керування ключами як асинхронний потенційний автомат. На часових діаграмах роботи ключів перетворювача виділимо чотири внутрішні стани автомата  $Q_0 \div Q_3$  (рис. 5.35), зміна яких відбувається під впливом широтно-модульованого вхідного сигналу  $x$ , частота якого в два рази перевищує частоту комутації

$$f_k = \frac{1}{T} \text{ перетворювача.}$$

Таблиця ПТП матиме чотири стани ( $Q_0...Q_3$ ) при двох стовпцях вхідного сигналу  $X$  (табл. 5.43).

З аналізу таблиці витікає, що еквівалентних, псевдоеквівалентних і сумісних станів в ПТП немає, тому її можна вважати скороченою. Для чотирьох станів використовується два ЕП, значення яких дають можливість закодувати стани відповідно до табл. 5.43, використовуючи сусіднє кодування.

Таблиця 5.43

$q_1$	$q_0$	$Q$ \ $X$	0	1	$Y_1$	$Y_2$	$Y_3$	$Y_4$	$Y_5$
0	0	$Q_0$	(0)	1	0	0	1	0	1
0	1	$Q_1$	2	(1)	0	1	1	0	0
1	1	$Q_2$	(2)	3	0	0	0	1	1
1	0	$Q_3$	0	(3)	1	0	0	1	0

Граф автомата (рис. 5.36) підтверджує можливість сусіднього кодування.

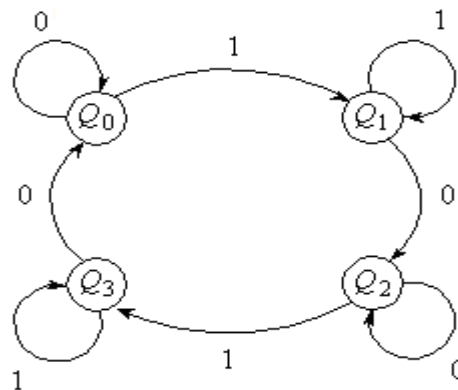


Рис.5.36

Кодована таблиця переходів приводиться в табл. 5.44, а таблиця збудження потенційних  $RS$ -тригерів – в табл. 5.45.

Таблиця 5.44

$q_1$	$q_0$	$Q$ \ $X$	0	1
0	0	$Q_0$	0 0	0 1
0	1	$Q_1$	1 1	0 1
1	1	$Q_2$	1 1	1 0
1	0	$Q_3$	0 0	1 0

Таблиця 5.45

$Q$ \ $X$		0				1				
		$q_1 q_0$	$R_1$	$S_1$	$R_0$	$S_0$	$R_1$	$S_1$	$R_0$	$S_0$
0	0	0 0	*	0	*	0	*	0	0	1
0	1	0 1	0	1	0	*	*	0	0	*
1	1	1 1	0	*	0	*	0	*	1	0
1	0	1 0	1	0	*	0	0	*	*	0

Будуємо карти Карно для функцій збудження входів тригерів (рис. 5.37, *a – з*).

Мінімізовані функції збудження мають вигляд:

$$R_1 = \bar{q}_0 \bar{x}; \quad S_1 = q_0 \bar{x};$$

$$R_0 = q_1 x; \quad S_0 = \bar{q}_1 x.$$

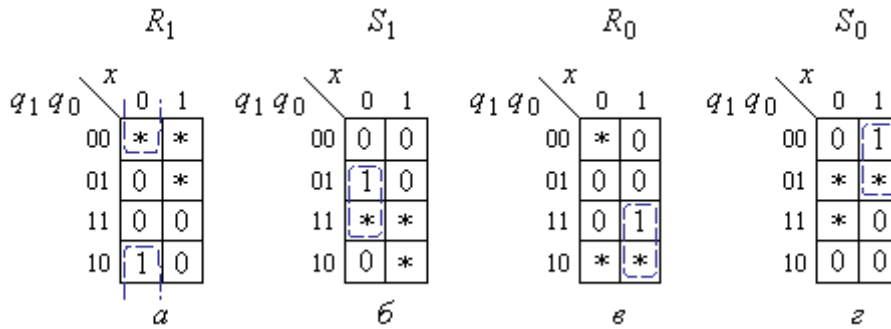


Рис.5.37

Будуємо карти Карно для функцій виходів автомата, скориставшись табл. 5.43. Вони приводяться на рис. 5.38, *a – д*.

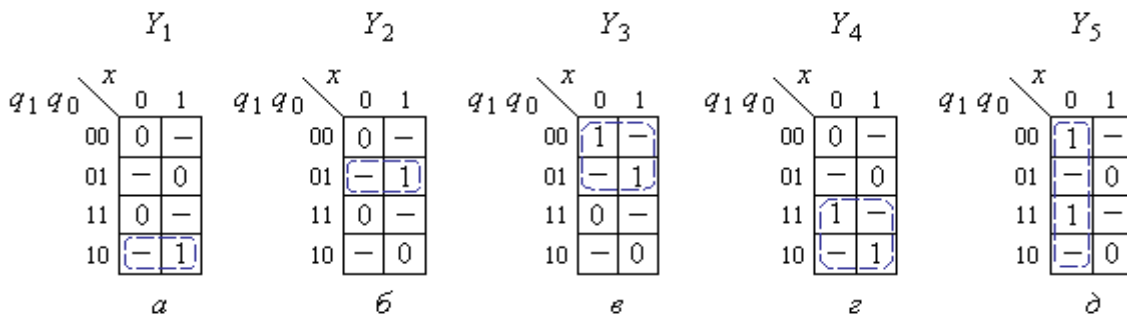


Рис.5.38

Мінімізовані функції виходів матимуть вигляд:

$$Y_1 = q_1 \bar{q}_0; \quad Y_2 = \bar{q}_1 q_0; \quad Y_3 = \bar{q}_1; \quad Y_4 = q_1; \quad Y_5 = \bar{x}.$$

Функціональна схема автомата приводиться на рис. 5.39.

Розглянемо схему і її відповідність часовим діаграмам.

У початковому стані  $Q_0$  і  $x=0$  маємо  $\bar{q}_1 = Y_3 = 1$  і  $Y_5 = \bar{x} = 1$ . На решті виходів – нульові рівні сигналів. При переході в перший стан при  $x=1$ , тригер DD7 встановлюється в значення  $q_0 = 1$ , внаслідок чого маємо  $Y_1 = 0$ ,  $Y_2 = 1$ ,  $Y_3 = 1$ ,  $Y_4 = 0$ ,  $Y_5 = 0$ . При переході в стан  $Q_2$  при  $x=0$  підготовлений на попередньому стані тригер DD6 встановлюється у

значені  $q_1 = 1$ , внаслідок чого проходить зміна вихідних сигналів:  $Y_5 = 1$ ,  $Y_4 = q_1 = 1$ ,  $Y_3 = \overline{q_1} = 0$ ,  $Y_2 = 0$ ,  $Y_1 = 0$ . При зміні сигналу  $x$  в  $x = 1$  настає стан  $Q_3$ , і через DD5 тригер DD7 переходить в стан  $q_0 = 0$ . Внаслідок такої зміни маємо  $Y_1 = 1$ ,  $Y_2 = 0$ ,  $Y_3 = 0$ ,  $Y_4 = 1$ ,  $Y_5 = 0$ . При наступній зміні вхідного сигналу на  $x = 0$  тригер DD6 через елемент DD3 встановлюється в нуль і автомат повертається в початковий стан  $Q_0$ . Приведений аналіз показує, що функціональна схема розробленого автомата повністю відповідає заданому алгоритму роботи.

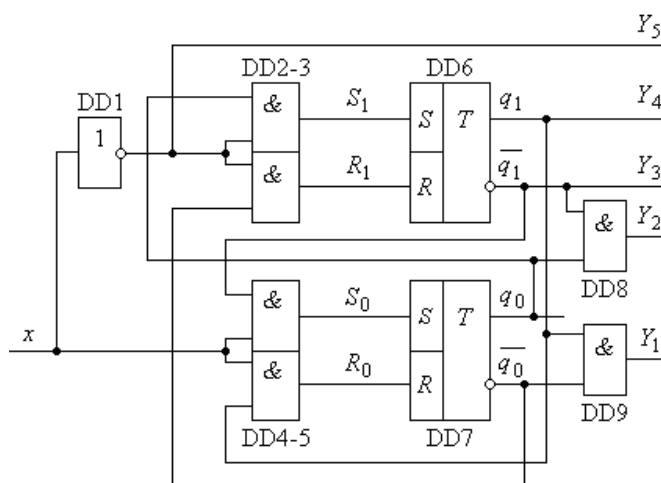


Рис.5.39

#### 5.4.2. Синтез асинхронних імпульсних автоматів

Асинхронними імпульсними автоматами називаються автомати, на які вхідні сигнали впливають лише на коротких інтервалах часу зміни їх рівня з “0” в “1” або навпаки. Точний опис законів функціонування даних автоматів можливий лише на основі їх логічних властивостей, що визначаються функціями переходів. Для аналітичного формування закону функціонування асинхронних імпульсних автоматів вводиться оператор переходу  $d$ . Використання оператора переходу до потенційних сигналів  $x(t)$  дає імпульсний сигнал  $dx(t)$  (рис. 5.40, а, б), який приймає одиничне значення в момент переходу  $x(t)$  з “1” в “0”. Для опису асинхронних імпульсних автоматів, що спрацьовують за фронтом, слід використовувати вираз для імпульсного сигналу у вигляді  $d\overline{x}(t)$  (рис. 5.40, в).

Задача синтезу асинхронних імпульсних автоматів може бути зведена до задачі синтезу асинхронних потенційних автоматів шляхом розробки методики перетворення функцій переходів, в які

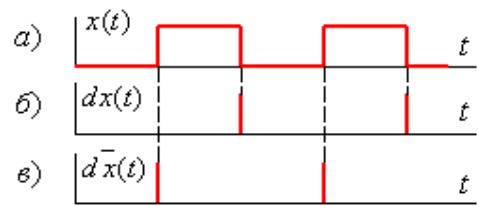


Рис.5.40

входять оператори  $d$ , в функції переходів та виходів, що не містять операторів  $d$ . Таке перетворення називається *інтегруванням функції переходів імпульсного автомата*, або *інтегруванням імпульсного автомата*.

Найбільш просто імпульсні автомати інтегруються табличним методом, оскільки є можливість побудувати таблицю переходів асинхронного потенційного автомата безпосередньо за функцією переходу імпульсного автомата.

Розглянемо це на прикладі синтезу імпульсного  $JK$ -тригера, повна функція переходів якого задається табл. 5.46. Вона практично не відрізняється від відповідної функції потенційного тригера, тому аналітичний вираз для закону функціонування (характеристичне рівняння) має вигляд:

$$Q_{n+1} = dJ\bar{Q}_n + d\bar{K}Q_n.$$

Виконуємо інтегрування характеристики імпульсного  $JK$ -тригера побудовою ПТП асинхронного потенційного автомата, еквівалентного синтезованому імпульсному тригеру. Якщо сигнали  $J$ - і  $K$ - не змінюються, то  $dJ = dK = 0$ , тому в стійких станах, відповідно до характеристичного рівняння тригера, має

Таблиця 5.46

$dJ$	$dK$	$Q_n$	$Q_{n+1}$
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

місце рівність  $Q_{n+1} = Q_n$ . З цього витікає, що ПТП повинна мати в кожному стовпці, що відповідає стану входу, по два стійких стани, які відрізняються значенням функції виходу  $Q=0$  або  $Q=1$ . Оскільки можливих станів входу чотири, то число стійких станів асинхронного потенційного автомата

дорівнює 8. Пронумерувавши їх у ряд, починаючи з нуля, заповнимо діагональ в ПТП (табл. 5.47). Потім, використовуючи характеристичне рівняння тригера, знайдемо переходи між стійкими станами.

Нехай, наприклад, початковим є стан входу  $x=0,1$ , якому відповідають стійкі стани  $Q_2$  і  $Q_3$ . Допустимо, що стан входу змінюється на  $x=1,0$ .

Таблиця 5.47

$Q$	$X(J, K)$	$JK$			
		00	01	10	11
0	(0), 0	2	4	6	
1	(1), 1	3	5	7	
2	0	(2), 0	4	6	
3	0	(3), 1	4	7	
4	1	3	(4), 0	6	
5	1	3	(5), 1	7	
6	1	3	4	(6), 0	
7	0	3	4	(7), 1	

Оскільки при цьому (в відповідності до рис. 5.40, б, в),  $J$  змінюється в напрямку  $0 \rightarrow 1$ , а  $K$   $1 \rightarrow 0$ , тобто  $dJ=0$ , а  $dK=1$ , то автомат повинен перейти в той внутрішній стан, якому відповідає стан виходу, який визначається характеристичним рівнянням тригера  $Q_{n+1} = 0 \cdot \bar{Q}_n + \bar{1} \cdot Q_n = 0$ , тобто, незалежно від вихідного стійкого стану ( $Q_2$  або  $Q_3$ ), він повинен перейти в стійкий стан  $Q_4$  з нульовим рівнем сигналу на виході. Аналогічно заповнюється решта кліток ПТП, які відповідають нестійким станам у табл. 5.47.

Наступним етапом проектування є абстрактний і структурний синтез за методикою синтезу асинхронних потенційних автоматів.

### 5.4.3. Особливості синтезу синхронних автоматів

У синхронних автоматах, на відміну від асинхронних, використовується тактовий сигнал  $C$ , що задає моменти часу, в які можуть мати місце зміни внутрішніх станів автомата. У синхронних автоматах здебільшого використовуються  $D$ - або  $JK$ -тригери. З особливостей роботи синхронних

тригерів витікає, що зміна інформаційних сигналів не може викликати зміни стану ЕП. Особливість синхронних тригерів, яка полягає в тому, що інформаційні сигнали лише керують змінами станів тригерів, переноситься і на синхронні автомати. Оскільки на всі тригери автомата синхросигнал подається одночасно, то вони і спрацьовують одночасно, незалежно від моменту подачі інформаційних сигналів.

У той же час, для синхронних автоматів, як відмічалось раніше, характерним є дискретний час, і стан автомата можна розглядати лише в дискретні моменти. Це дозволяє виключити з розгляду перехідні процеси, що протікають у комбінаційній схемі при зміні її вхідних сигналів.

Оскільки таблиці станів синхронних і асинхронних  $D$ - і  $JK$ -тригерів у цілому співпадають, то методи структурного синтезу синхронних автоматів ідентичні методам, використовуваним для синтезу асинхронних автоматів. У той же час, наявність синхронного режиму роботи всіх ЕП приводить до появи ряду особливостей проектування:

- комбінаційні схеми можуть проектуватись не вільними від гонок;
- допускаються несусідні зміни станів входу;
- допускаються критичні змагання ЕП, тобто допускається несусіднє

кодування внутрішніх станів.

У великій кількості практичних задач синтезу синхронних автоматів перехід від словесного опису до стандартних способів задання функцій переходів і виходів буває дуже простим. Тому для синхронних автоматів часто є можливість, минаючи етап абстрактного синтезу, зразу ж перейти до етапу структурного синтезу. Таке спрощення процедури синтезу є наслідком використання синхронних ЕП, які виконують значно складніші функції, ніж асинхронні потенційні тригери.

**Приклад 5.11.** Синтезувати синхронний лічильник за модулем 5, що задається таблицею переходів (табл. 5.48) з використанням  $D$ - і  $JK$ -тригерів. Характеристичні таблиці переходів для  $D$ - і  $JK$ -тригерів приведені в табл. 5.49.

Розв'язання. Будуємо таблицю збудження для  $D$ -входів в  $D$ -тригерах (табл. 5.50).

З цієї таблиці отримуємо функціональні залежності входів  $D$ -тригерів  $D_0 \dots D_2$  від попереднього стану виходів тих же тригерів (функції збудження):

$$D_0 = \overline{q_2} \overline{q_0} ; D_1 = \overline{q_2} (q_1 \oplus q_0) ; D_2 = \overline{q_2} q_1 q_0 .$$

Таблиця 5.48

$N$	$Q_n$			$Q_{n+1}$		
	$q_2$	$q_1$	$q_0$	$q_2$	$q_1$	$q_0$
$Q_0$	0	0	0	0	0	1
$Q_1$	0	0	1	0	1	0
$Q_2$	0	1	0	0	1	1
$Q_3$	0	1	1	1	0	0
$Q_4$	1	0	0	0	0	0

Таблиця 5.49

$Q_n$	$Q_{n+1}$	$D_n$	$J_n$	$K_n$
0	0	0	0	*
0	1	1	1	*
1	0	0	*	1
1	1	1	*	0

Таблиця 5.50

$Q_n \rightarrow Q_{n+1}$	$Q_n$			$D_n$		
	$q_2$	$q_1$	$q_0$	$D_2$	$D_1$	$D_0$
$Q_0 \rightarrow Q_1$	0	0	0	0	0	1
$Q_1 \rightarrow Q_2$	0	0	1	0	1	0
$Q_2 \rightarrow Q_3$	0	1	0	0	1	1
$Q_3 \rightarrow Q_4$	0	1	1	1	0	0
$Q_4 \rightarrow Q_0$	1	0	0	0	0	0

За отриманими функціями збудження розробляємо функціональну схему лічильника (рис. 5.41).

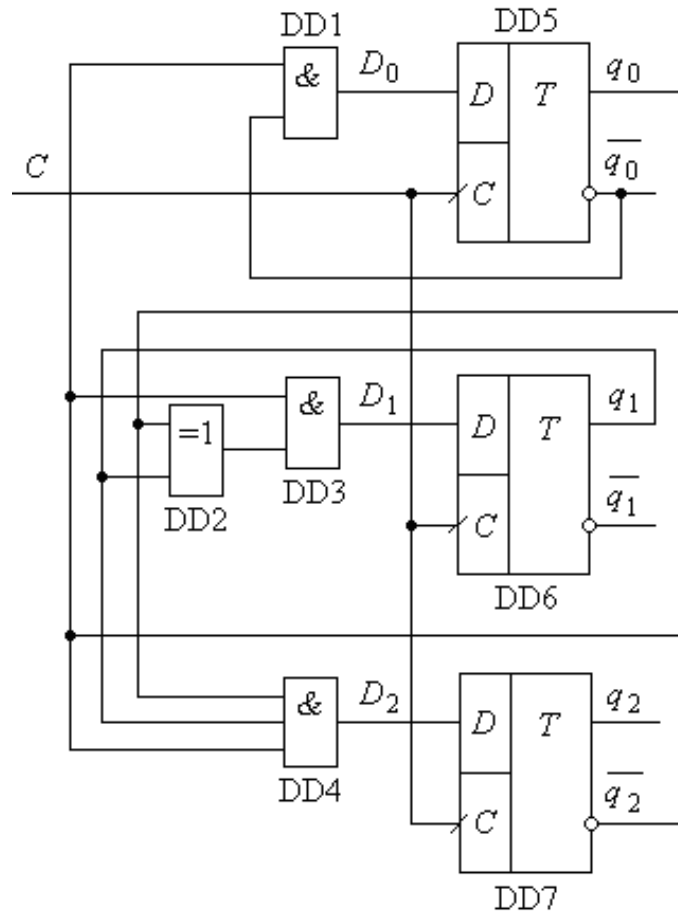


Рис.5.41



У початковому стані  $Q_0$  всі прямі виходи лічильника  $q_2 q_1 q_0 = 000$ , а їх інверсні еквіваленти  $\overline{q_2} \overline{q_1} \overline{q_0} = 111$ . Тому на вході тригера DD5 маємо  $D_0 = 1$ , а на входах тригерів DD6 і DD7, відповідно,  $D_1 = 0$  і  $D_2 = 0$ . Тому при подачі синхросигналу DD5 фіксує на своєму прямому виході  $q_0 = 1$ . Це приводить до зміни значень входів перед другим тактовим імпульсом. Тепер маємо  $D_0 = 0$ ,  $D_1 = 1$ ,  $D_2 = 0$  і по фронту  $C$  обнуляється  $q_0$ , а  $q_1 = 1$ . В подальшому схема циклічно змінюватиме свої стани у відповідності до алгоритму роботи.

Тепер розглянемо особливості синтезу з використанням  $JK$ -тригерів. Будуємо таблицю збудження для входів  $JK$ -тригерів (табл. 5.51).

Таблиця 5.51

$Q_n \rightarrow Q_{n+1}$	$Q_n$			$J_n K_n$					
	$q_2$	$q_1$	$q_0$	$J_2$	$K_2$	$J_1$	$K_1$	$J_0$	$K_0$
$Q_0 \rightarrow Q_1$	0	0	0	0	*	0	*	1	*
$Q_1 \rightarrow Q_2$	0	0	1	0	*	1	*	*	1
$Q_2 \rightarrow Q_3$	0	1	0	0	*	*	0	1	*
$Q_3 \rightarrow Q_4$	0	1	1	1	*	*	1	*	1
$Q_4 \rightarrow Q_0$	1	0	0	*	1	0	*	0	*

Після мінімізації знаходимо:

$$\begin{aligned}
 J_0 &= \overline{q_2} ; & K_0 &= 1 ; \\
 J_1 &= q_0 ; & K_1 &= q_0 ; \\
 J_2 &= q_1 q_0 ; & K_2 &= 1 .
 \end{aligned}$$

Схема лічильника з використанням синхронних  $JK$ -тригерів приведена на рис. 5.42.

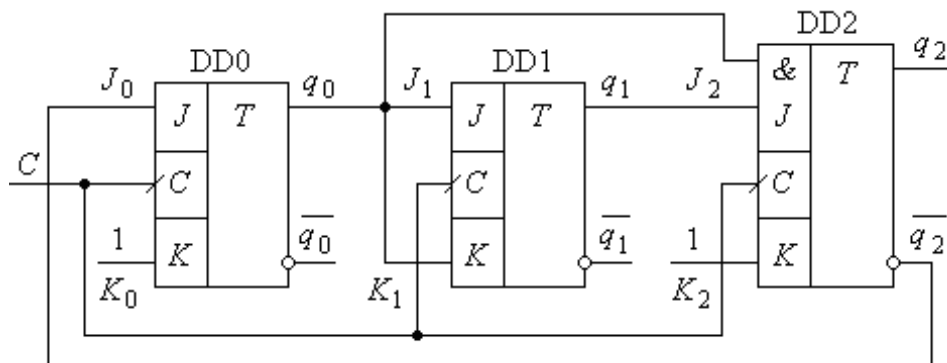


Рис.5.42

Легко проаналізувати і впевнитись, що робота лічильника повністю відповідає заданому алгоритму.

#### 5.4.4. Використання теореми Шеннона при синтезі скінченних автоматів на основі JK-тригерів

Як відомо, характеристичне рівняння JK-тригера має вигляд:

$$Q_{n+1} = J_n \overline{Q_n} + \overline{K_n} Q_n .$$

Звернемо тепер увагу на теорему Шеннона (Розділ 1, формула (1.17)):

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \overline{x_i} \cdot f(x_{n-1}, \dots, 0, \dots, x_0) + x_i \cdot f(x_{n-1}, \dots, 1, \dots, x_0) .$$

Позначимо функції  $f(x_{n-1}, \dots, 0, \dots, x_0)$  і  $f(x_{n-1}, \dots, 1, \dots, x_0)$  відповідно як функції  $f_1$  та  $f_2$ , а формулу (1.17) перепишемо у вигляді:

$$f(x_{n-1}, \dots, x_i, \dots, x_0) = \overline{x_i} \cdot f_1 + x_i \cdot f_2 .$$

Тепер звернемо увагу на той факт, що функцію переходів будь-якого тригера скінченного автомата можемо представити у вигляді:

$$Q_{(n+1)_i} = f(x_0 \dots x_n; Q_0 \dots Q_n) ,$$

тому, розглядаючи її у відповідності до теореми Шеннона, відносно стану заданого  $i$ -го тригера, отримуємо:

$$Q_{(n+1)_i} = f_{1i} \cdot \overline{Q_i} + f_{2i} \cdot Q_i , \quad (5.15)$$

де функції  $f_{1i}$  та  $f_{2i}$  вже не містять відповідно змінних  $Q_i$  та  $\overline{Q_i}$ .

Порівнюючи тепер характеристичне рівняння JK-тригера з формулою (5.15), бачимо, що:

$$f_{1i} = J \quad \text{і} \quad \overline{f_{2i}} = K .$$

Звідси витікає, що, прийнявши в функції переходів тригера  $Q_i = 0$ , отримуємо функцію збудження для входу  $J$ -:

$$J_i = f_{1i} = Q_{(n+1)_i} \Big|_{Q_i = 0} ,$$

а, прийнявши  $Q_i = 1$ , отримуємо функцію збудження для входу  $K$ -:

$$\overline{K_i} = f_{2i} = Q_{(n+1)_i} \Big|_{Q_i = 1} \quad \text{або} \quad K_i = \overline{f_{2i}} .$$

**Приклад 5.12.** Розробити автомат, який в залежності від керуючого сигналу  $G$  міг би працювати як двійковий лічильник ( $G = 0$ ) або як лічильник з відліком у відповідності до коду Грея з  $M = 8$  ( $G = 1$ ). Автомат розробити з використанням JK-тригерів.

Розв'язання. Таблиця переходів автомата має вигляд табл. 5.52.

Таблиця 5.52

G	Q <sub>2n</sub>	Q <sub>1n</sub>	Q <sub>0n</sub>	Q <sub>2(n+1)</sub>	Q <sub>1(n+1)</sub>	Q <sub>0(n+1)</sub>
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	1
0	0	1	1	1	0	0
0	1	0	0	1	0	1
0	1	0	1	1	1	0
0	1	1	0	1	1	1
0	1	1	1	0	0	0
1	0	0	0	0	0	1
1	0	0	1	0	1	1
1	0	1	1	0	1	0
1	0	1	0	1	1	0
1	1	1	0	1	1	1
1	1	1	1	1	0	1
1	1	0	1	1	0	0
1	1	0	0	0	0	0

Рівняння переходів кожного з тригерів:

$$Q_{2(n+1)} = \bar{G} \cdot (\bar{Q}_{2n} Q_{1n} Q_{0n} + Q_{2n} \bar{Q}_{1n} \bar{Q}_{0n}) + G \cdot (\bar{Q}_{2n} Q_{1n} \bar{Q}_{0n} + Q_{2n} Q_{1n} Q_{0n}) ;$$

$$Q_{1(n+1)} = \bar{G} Q_{2n} \bar{Q}_{1n} Q_{0n} + G \bar{Q}_{2n} Q_{1n} Q_{0n} ;$$

$$Q_{0(n+1)} = \bar{G} \cdot (\bar{Q}_{2n} Q_{1n} \bar{Q}_{0n} + Q_{2n} \bar{Q}_{1n} \bar{Q}_{0n}) + G \cdot (\bar{Q}_{2n} \bar{Q}_{1n} Q_{0n} + Q_{2n} Q_{1n} Q_{0n}) .$$

Функції збудження для входів J- і K-:

$$J_2 = \bar{G} Q_{1n} Q_{0n} + G Q_{1n} \bar{Q}_{0n} ;$$

$$\bar{K}_2 = \bar{G} \bar{Q}_{1n} \bar{Q}_{0n} + G Q_{1n} Q_{0n} ;$$

$$J_1 = \bar{G} Q_{2n} Q_{0n} ;$$

$$\bar{K}_1 = G \bar{Q}_{1n} Q_{0n} ;$$

$$J_0 = \bar{G} \bar{Q}_{2n} Q_{1n} + G Q_{2n} \bar{Q}_{1n} ;$$

$$\bar{K}_0 = G (\bar{Q}_{2n} \bar{Q}_{1n} + Q_{2n} Q_{1n}) .$$

Читачам пропонується самостійно побудувати принципову схему синтезованого цифрового автомата у відповідності до цих функцій.

## ВПРАВИ І ЗАВДАННЯ

1. Проаналізувати роботу синхронного автомата, схема якого приведена на рис. 5.43. Скласти таблицю переходів, записати рівняння збудження, зобразити часові діаграми роботи автомата.

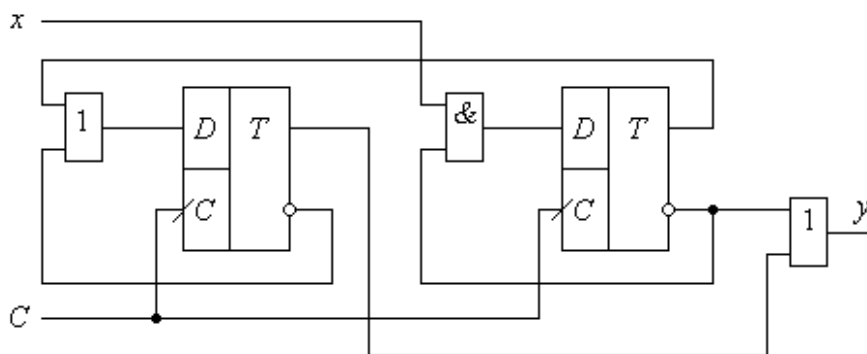


Рис. 5.43

2. Повторити вправу 1, помінявши місцями елементи **2АБО** та **2І**. Порівняти таблиці переходів, дати пояснення.

3. Побудувати граф переходів для скінченного автомата, що описується таблицею переходів (табл. 5.53).

4. Побудувати граф-схему автомата, заданого таблицею переходів (табл. 5.54).

5. Побудувати граф-схему автомата, заданого таблицею переходів (табл. 5.55).

Таблиця 5.53

$X$	$x_1 x_0$				$Y$
	$00$	$01$	$11$	$10$	
$Q_0$	$Q_1$	$Q_1$	$Q_2$	$Q_2$	0
$Q_1$	$Q_3$	$Q_3$	$Q_2$	$Q_2$	0
$Q_2$	$Q_1$	$Q_1$	$Q_4$	$Q_4$	0
$Q_3$	$Q_3$	$Q_3$	$Q_4$	$Q_2$	1
$Q_4$	$Q_1$	$Q_3$	$Q_4$	$Q_4$	1

Таблиця 5.54

$X$	$x_1 x_0$				$Y$
	$00$	$01$	$11$	$10$	
$Q_0$	$Q_0$	$Q_1$	$Q_2$	$Q_1$	1
$Q_1$	$Q_1$	$Q_2$	$Q_3$	$Q_2$	0
$Q_2$	$Q_2$	$Q_3$	$Q_0$	$Q_3$	0
$Q_3$	$Q_3$	$Q_0$	$Q_1$	$Q_0$	0

Таблиця 5.55

$X$	$x$	
	$0$	$1$
$A$	$B$	$A$
$B$	$B$	$C$
$C$	$B$	$D$
$D$	$E$	$A$
$E$	$B$	$F$
$F$	$B$	$G$
$G$	$E$	$H$
$H$	$B$	$A$

6. Розробити таблицю переходів і виходів скінченного автомата, заданого граф-схемою, що приведена на рис. 5.44. Під позначеннями станів  $Q_0 \div Q_7$  вказані значення виходів  $y_1 y_2$  автомата.

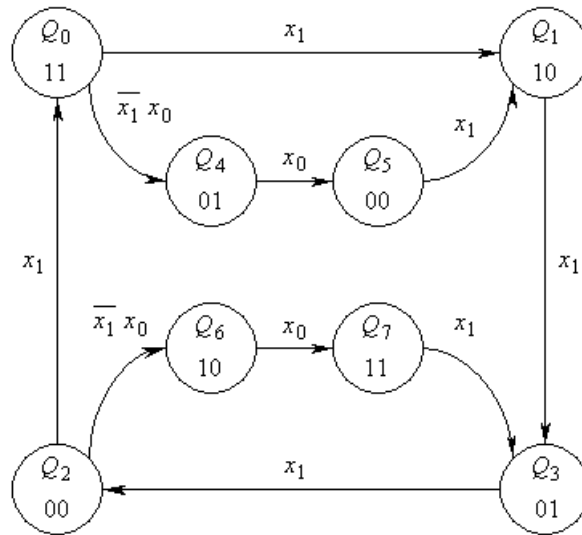


Рис. 5.44

7. Виконати аналіз роботи синхронного автомата, функціональна схема якого приводиться на рис. 5.45. Побудувати часові діаграми його роботи, записати рівняння збудження  $D$ -тригерів, скласти таблицю переходів і граф-схему переходів автомата.

8. За аналогією з вправою 7, виконати аналіз роботи синхронного автомата, функціональна схема якого приведена на рис. 5.46.

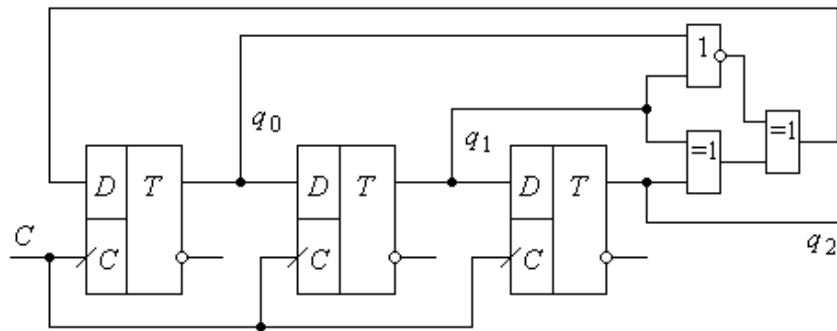


Рис. 5.45

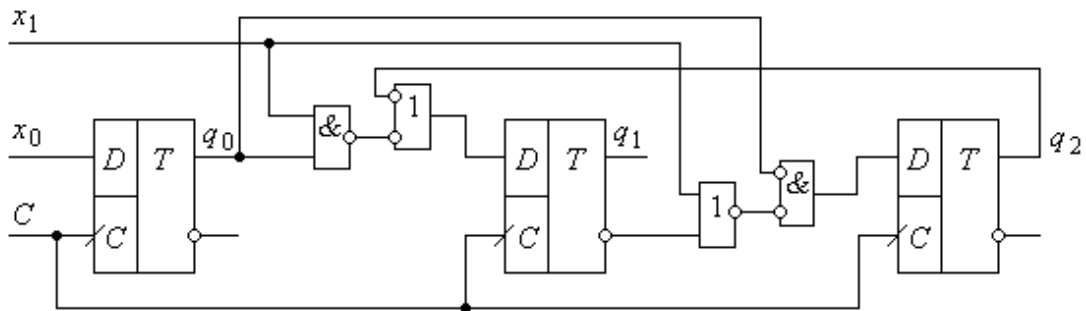


Рис. 5.46

9. За аналогією з вправою 7, виконати аналіз роботи синхронного автомата, функціональна схема якого приведена на рис. 5.47. Аналіз провести для  $x=1$ , починаючи з нульового початкового стану  $Q_0 = q_1 q_0 = 00$ .

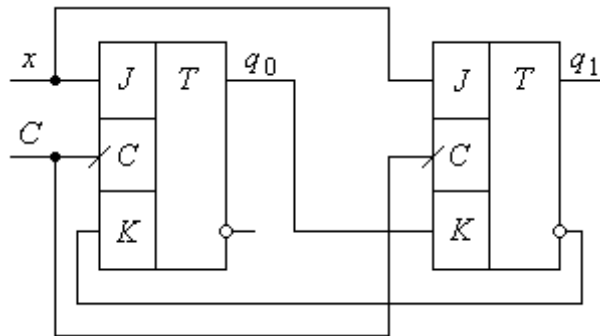


Рис. 5.47

10. За аналогією з вправою 7, виконати аналіз роботи синхронного скінченного автомата, функціональна схема якого приведена на рис. 5.48. Аналіз виконати при умові  $x=1$ .

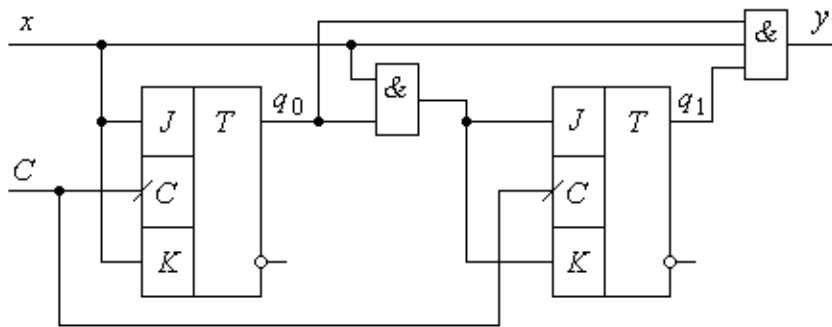


Рис. 5.48

11. За аналогією з попередніми вправами, виконати аналіз синхронного скінченного автомата, функціональна схема якого приведена на рис. 5.49. Автомат виготовлений на  $T$ -тригерах з дозволяючими входами  $L$ .

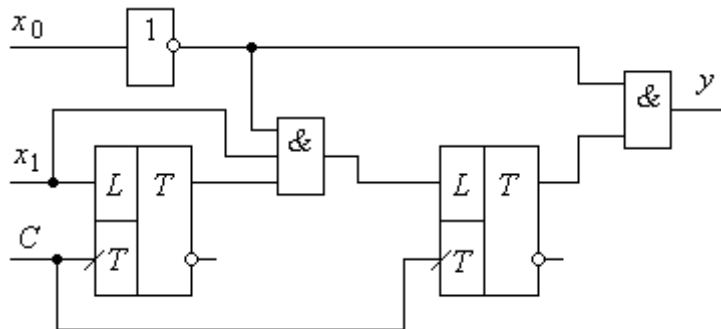


Рис. 5.49

12. Спроекувати синхронний автомат Мура і описати його роботу за допомогою часових діаграм. Автомат має два входи  $x_1$  та  $x_0$  і один вихід  $y$ . Як тільки вхід  $x_0$  встановиться в 1, на виході  $y$  з'явиться низький рівень вихідного сигналу  $y = 0$ . Якщо  $x_0$  змінить своє значення в 0, то  $y$  залишатиметься на низькому рівні при  $x_1 = 0$  протягом двох тактів і перейде в 1 на наступних двох тактах. При  $x_1 = 1$  автомат відразу перейде в стан  $y = 1$  і залишатиметься у такому стані до моменту часу, поки  $x_0$  знову не встановиться в 1.

13. Спроекувати синхронний автомат Мура, який виконував би перевірку послідовних даних, що приймаються, на парність. Автомат повинен мати два входи – вхід синхронізації  $C$ - і вхід даних  $D$ -, а також вхід тактового генератора, і один вихід  $y$ , високий рівень якого вказує на помилку в передачі даних. В автоматі потрібно забезпечити можливість перевірки слова довжиною в 1 байт, можна використовувати  $D$ - або  $JK$ -тригери.

14. Спроекувати синхронний автомат Мура, який виконував би перевірку даних, що передаються, на парність. Автомат повинен мати два входи – вхід синхронізації  $C$ - і вхід даних  $D$ -, і один вихід  $y$ , високий рівень якого встановлює дев'ятий біт в 1, якщо кількість одиниць в одnobайтовому слові непарна. При парній кількості одиниць  $y = 0$ . В схемі можна використовувати  $D$ - або  $JK$ -тригери.

15. Спроекувати тактований синхронний автомат на основі таблиці переходів (табл. 5.56). Для синтезу автомата використовуйте  $D$ -тригери.

16. Повторити вправу 15 з використанням  $JK$ -тригерів.

17. Виконати аналіз автомата, функціональна схема якого приведена на рис. 5.50. Скласти таблицю переходів, побудувати граф-схему та часові діаграми його роботи.

Таблиця 5.56

$Q$ \ $X$	$x$		$y$
	0	1	
$Q_0$	$Q_1$	$Q_3$	0
$Q_1$	$Q_2$	$Q_1$	0
$Q_2$	$Q_1$	$Q_0$	1
$Q_3$	$Q_1$	$Q_2$	0

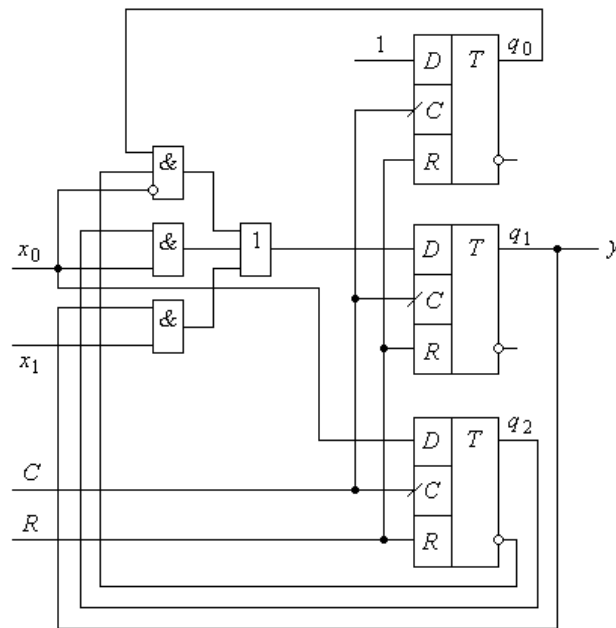


Рис. 5.50

18. Виконати синтез автомата за результатами вправи **17** з використанням *JK*-тригерів.

19. Табл. 5.53 описує автомат, в якому початковий стан  $Q_0$  призначений для його запуску. Якщо цей стан видалити, автомат також працюватиме, якщо керуючі сигнали будуть подані після встановлення перехідних процесів при подачі живлення. Синтезувати новий автомат з використанням *D*-тригерів зі скороченою кількістю станів.

20. Виконати вправу **19** з використанням *JK*-тригерів.

21. розробити схему пристрою для виконання операції додавання двох чотирьохбітних слів у послідовному форматі з запам'ятовуванням сигналу переносу на один такт.

22. Розробити схему кодового замка (асинхронного автомата), який спрацює після правильного послідовного натискання чотирьох цифр.

23. З використанням *RS*-, *D*- і *JK*-тригерів розробити схеми цифрових автоматів для порівняння двох двійкових чисел у послідовному форматі. Порівнювані числа  $X (x_{n-1}, \dots, x_1, x_0)$  та  $Y (y_{n-1}, \dots, y_1, y_0)$ . При порівнянні необхідно встановити їх співвідношення:  $X > Y$ ;  $X < Y$ ;  $X = Y$ , які описуються за допомогою вказаних нижче функцій.



$$F_1(X > Y) = \begin{cases} \mathbf{1} & \text{при } X > Y; \\ \mathbf{0} & \text{при } X \leq Y; \end{cases}$$

$$F_2(X < Y) = \begin{cases} \mathbf{1} & \text{при } X < Y; \\ \mathbf{0} & \text{при } X \geq Y; \end{cases}$$

$$F_3(X = Y) = \begin{cases} \mathbf{1} & \text{при } X = Y; \\ \mathbf{0} & \text{при } X \neq Y. \end{cases}$$

24. Розробити асинхронний автомат для сортування деталей **A, B, C**, що переміщуються на конвеєрі у довільному порядку.

25. Задача про подачу води в водонапірну вежу [Голдсуорт]. Два насоси  $\rho_1$  і  $\rho_2$  використовуються для подачі води до водонапірної вежі. Обидва починають працювати, коли вода опуститься нижче рівня мітки 1, і працюють до того часу, поки рівень води не підніметься до мітки 2. При цьому насос  $\rho_1$  зупиниться, а насос  $\rho_2$  продовжуватиме роботу до того часу, поки рівень води не підніметься до мітки 3, після чого він зупиниться. Запуститися обидва насоси повинні тоді, коли вода опуститься до мітки 1. Розробити автомат керування насосами.

26. Задача визначення кількості відвідувачів музею [Голдсуорт]. Спроекувати автомат, який повинен підрахувати кількість відвідувачів музею за визначений інтервал часу. В якості датчика використовуються дві оптопари  $O_1$  та  $O_2$ , рознесені на відстань 1 м. Відвідувач фіксується, якщо він перетне спочатку промінь першої оптопари, а потім другої. Зворотний процес не враховується.

## ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

АЛМ арифметично-логічний пристрій

АМ – амплітудна модуляція

АІМ – амплітудно-імпульсна модуляція

БіКМОН – поєднання технологій КМОН з біполярними транзисторними ключами на виході

БМК – базовий матричний кристал

БТ – багатомітерний транзистор

ВІС – велика інтегральна схема

ЕЗЛ – емітерно-зв'язана логіка

ЕП – елемент пам'яті

ЗП – запам'ятовуючий пристрій

ДТЛ – діодно-транзисторна логіка

ДНФ – диз'юнктивна нормальна форма запису логічних функцій

ДДНФ – досконала диз'юнктивна нормальна форма запису логічних функцій

ДКНФ- досконала кон'юнктивна нормальна форма запису логічних функцій

ІМС – інтегральна мікросхема

ІС – інтегральна схема

ІКМ – імпульсно-кодова модуляція

ІІЛ – інтегральна інжекційна логіка

КНФ – кон'юнктивна нормальна форма запису логічних функцій

КМОН (CMOS) – комплементарні МОН-структури

ЛЕ – логічний елемент

ЛІЗМОН – МОН структури з лавинною інжекцією заряду

МОН – структура на базі з'єднання метал-окисел-напівпровідник

МС – мікросхема

НВІС – надвелика інтегральна схема

ПЗЗ – прилад з зарядовим зв'язком

ПЗП – постійний запам'ятовуючий пристрій  
ПКЧ – перетворювач код-частота  
ПЛІС – програмована логічна інтегральна схема  
ПЛМ – програмована логічна матриця  
ПМД – послідовність максимальної довжини  
ПМЛ – програмована матрична логіка  
ПТП – початкова таблиця переходів скінченного автомата  
ОЗП – оперативний запам'ятовуючий пристрій  
РПЗП – репрограмований постійний запам'ятовуючий пристрій  
РТЛ – резистивно-транзисторна логіка  
ТТЛ (TTL) – транзисторно-транзисторна логіка  
ТТЛШ – транзисторно-транзисторна логіка з діодами Шоткі  
ФАПЧ – фазова автопідстройка частоти  
ФД – фазовий детектор  
ФМ – фазова модуляція  
ФІМ – фазо-імпульсна модуляція  
ФНЧ – фільтр нижніх частот  
ЦА – цифровий автомат  
ЦІС – цифрова інтегральна схема  
ЧМ – частотна модуляція  
ЧІМ – частотно-імпульсна модуляція  
ШІМ – широтно-імпульсна модуляція  
CPLD – Complex Programmable Logic Device  
EPROM ( РПЗП) – Electrically Programmable ROM  
PROM (РПЗП) – Programmable Read Only Memory  
FPGA – Field Programmable Gate Array  
Flex – Flexible Logic Element Matrix  
GA (БМК) – Gate Array  
ISP – In-System Programming

JFET – Junction Field Effect Transistor

JTAG – Joint Test Action Group

MESFET – Metal Semiconductor Field Effect Transistor

PAL (PIML) – Programmable Array Logic

PLA (PLM) – Programmable Logic Array

SOC – System on Chip

SPI – Serial Peripheral Interface

## ЛІТЕРАТУРА

1. Алексенко А. Г., Шагурин И. И. Микросхемотехника. – М.: Радио и связь, 1990.- 496 с.
2. Барнс Д. Электронное конструирование: методы борьбы с помехами. – М: Мир, 1990.- 238 с.
3. Баскаков С. И. Радиотехнические цепи и сигналы. – М.: Высшая школа, 1983.- 536 с.
4. Бойко В. І., Багрій В. В. Цифрова схемотехніка. – К: ІЗМН, 2001.- 228 с.
5. Большие интегральные схемы запоминающих устройств. Справочник. Под ред. Гордонова А. Ю., Дьякова Ю. Н. – М.: Радио и связь , 1990.-288 с.
6. Борисенко О. А. Цифрові автомати. – Суми: Видавництво СумДУ, 2001.- 168 с.
7. Вениаминов В. Н., Лебедев О. Н., Мирошниченко А. И. Микросхемы и их применение. – М.: Радио и связь , 1989.- 240 с.
8. Власов А. И., Сулимов Ю. И. Электронные промышленные устройства. – М.: Высшая школа, 1988.- 304 с.
9. Голдсуорт Б. Проектирование цифровых логических устройств. – М.: Машиностроение, 1985.- 287 с.
10. Гольденберг Л.М. Импульсные устройства. – М.: Радио и связь, 1981.-224 с.
11. Гулый В. Д., Артеменко М. Б. Методические указания по изучению дисциплины Электронные промышленные устройства . – К.: КПИ, 1986.- 32 с.
12. Гурвич И. С. Защита ЭВМ от внешних помех. – М.: Энергоатомиздат, 1984.- 224 с.
13. Гутников В. С. Интегральная электроника в измерительных устройствах. – Л.: Энергоатомиздат, 1988.- 304 с.
14. Завадский В. А. Компьютерная электроника. – К.: ТОО ВЕК , 1996.- 360 с.

15. Зельдин Е.А. Импульсные устройства на микросхемах. – М.: Радио и связь, 1991.- 160 с.
16. Зельдин Е. А. Цифровые интегральные микросхемы в информационно-измерительной аппаратуре. – Л.: Энергоатомиздат, 1986.- 280 с.
17. Калабеков Б. А. Микропроцессоры и их применение в системах передачи и обработки сигналов. – М.: Радио и связь, 1988.- 368 с.
18. Лебедев О. Н. Микросхемы памяти и их применение. – М.: Радио и связь, 1990.- 160 с.
19. Логические ИС. Справочник в 2-х томах. Бином, 1993.
20. Мейдза Ф. Интегральные схемы. Технология и применение. – М.: Мир, 1981.- 280 с.
21. Новиков Ю. В. Основы цифровой схемотехники. – М.: Мир , 2001.-380 с.
22. Новожилов О.П. Основы цифровой техники / учебное пособие. –М.: ИП РадиоСофт, 2004.- 528 с.
23. Опадчий Ю. Ф., Глудкин О. П., Гуров А. И. Аналоговая и цифровая электроника. – М.: Горячая линия – Телеком, 1999.- 768 с.
24. Петросян О. А., Козырь И. Я., Колядов Л. А., Щетинин Ю. И. Схемотехника БИС постоянных запоминающих устройств. – М.: Радио и связь, 1987.- 304 с.
25. Полупроводниковые БИС запоминающих устройств. Под ред. Гордонова А. Ю., Дьякова Ю. Н. – М.: Радио и связь , 1987.- 360 с.
26. Потемкин И. С. Функциональные узлы цифровой автоматики. – М.: Энергоатомиздат, 1988.- 320 с.
27. Применение интегральных микросхем в электронной вычислительной технике. Справочник. Под. ред. Б.Н. Файзулаева, Б.В. Тарабрина. –М.: Радио и связь; 1987.- 384 с.
28. Пухальский Г. И., Новосельцева Т. Я. Проектирование дискретных устройств на интегральных микросхемах. – М.: Радио и связь, 1990.- 304 с.

29. Пухальский Г. И., Новосельцева Т. Я. Цифровые устройства. – СПб.: Политехника, 1996.- 880 с.
30. Скаржепа В. А., Луценко А. Н. Электроника и микросхемотехника. Кн.1. – К.: Вища школа, 1989.- 430 с.
31. Смирнов В. С. Електронні імпульсні пристрої. – Київ: НТУ КПІ, 1998.- 140 с.
32. Справочник по цифровой схемотехнике / В.И. Зубчук, В.П. Сигорский, А.Н. Шкуро. – К.: Техника, 1990. – 448 с.
33. Точки Р., Уидмер Н. Цифровые системы. Теория и практика, 8-е издание.: Пер. с англ. – М.:издательский дом «Вильямс»,2004.- 1024 с.
34. Трачик В. Дискретные устройства автоматики. – М.: Энергия, 1978.-456 с.
35. Угрюмов Е. Цифровая схемотехника. – СПб.: БХВ, 2000.- 528 с.
36. Цифровые и аналоговые интегральные микросхемы. Справочник. Под ред. Якубовского. – М.: Радио и связь, 1989.-496 с.
37. Цифрова схемотехніка: Підручник у двох томах, том 2. Жуйков В.Я., Бойко В.І., Зорі А.А. та ін. –К.: Аверс, 2002.- 408 с.
38. Цифровые интегральные микросхемы. Справочник. – Минск: Беларусь, 1991.- 494 с.
39. Шило В. Л. Популярные цифровые микросхемы. – М.: Радио и связь, 1987.- 352 с.
40. Энциклопедия кибернетики: в 2-х т. – К.: УСЭ, 1974, т.1.