

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

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

Кафедра комп'ютерно-інтегрованих технологічних процесів та виробництв

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

ЦИФРОВІ ПРИСТРОЇ

Навчальний посібник

Частина 1

Одеса – 2016

Воробйова О.М. Цифрові пристрої: навч. посіб. – Ч. 1 / О.М. Воробйова, М.П. Савицька, Ю.В. Флейта. – Одеса: ОНАЗ ім. О.С. Попова, 2016. – 112 с.

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

УХВАЛЕНО

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

ЗАТВЕРДЖЕНО

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

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

ЗМІСТ

Передмова	5
Список прийнятих скорочень.....	6
1. ВСТУП.....	8
1.1. Класифікація цифрових пристроїв	8
1.2. Цифрові сигнали.....	9
2. ОСНОВИ ЦИФРОВИХ ПРИСТРОЇВ	12
2.1. Арифметичні основи	12
2.1.1. Системи числення	12
2.1.2. Переведення числа з однієї системи числення в іншу	12
2.1.3. Двійкова арифметика.....	13
2.2. Логічні основи цифрових пристроїв	14
2.2.1. Основні поняття бульової алгебри	14
2.2.2. Форми зображення логічних функцій.....	25
2.2.3. Мінімізація логічних функцій.....	23
2.3. Логічні функції бульового базису	27
2.3.1. Функція логічного заперечення НЕ	27
2.3.2. Функція логічного множення І.....	38
2.3.3. Функція логічного додавання АБО	30
2.3.4. Комбіновані логічні функції	31
Контрольні питання	34
Рекомендована література.....	34
3. СХЕМОТЕХНІКА ІНТЕГРАЛЬНИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ.....	36
3.1. Класифікація та основні параметри логічних елементів	36
3.2. Транзисторно-транзисторна логіка (ТТЛ).....	40
3.3. Транзисторно-транзисторна логіка з діодом Шотткі (ТТЛШ).....	44
3.4. Емітерно-зв'язана логіка (ЕЗЛ)	45
3.5. Інтегральна інжекційна логіка (I^2L)	45
3.6. Логічні елементи на польових структурах	46
3.6.1. МОН-логіка.....	47
3.6.2. КМОН-логіка	48
3.7. Логічний елемент з трьома станами.....	51
3.8. Узгодження логічних мікросхем	53
3.9. Правила схемного включення елементів.....	56
Контрольні питання	59
Рекомендована література.....	60
4. ПОСЛІДОВНІСНІ ПРИСТРОЇ	61
4.1. Тригери.....	61
4.1.1. Параметри тригерів.....	63
4.1.2. Асинхронний RS-тригер.....	64
4.1.3. Синхронний RS-тригер.....	67

4.1.4. D-тригер	69
4.1.5. JK- та MS-тригери	71
4.1.5.1. JK-тригер	71
4.1.5.2. MS-тригер	73
4.1.6. Підвищення завадостійкості тригерів	76
4.2. Формувачі тривалості фронтів	77
4.3. Методи боротьби з дренькотом контактів	80
4.4. Лічильники імпульсів	80
4.4.1. Параметри лічильника	81
4.4.2. Класифікація лічильників	82
4.4.3. Послідовні лічильники	90
4.4.4. Швидкодія лічильників	914
4.4.5. Паралельні лічильники	93
4.4.6. Паралельно-послідовні лічильники	94
4.4.7. Реверсивні лічильники	98
4.4.8. Лічильники з довільним модулем лічби	100
4.4.9. Збільшення розрядності лічильників. Подільники частоти	101
4.5. Регістри	101
4.5.1. Класифікація регістрів	101
4.5.2. Послідовні регістри	102
4.5.3. Паралельні регістри	108
Контрольні питання	109
Рекомендована література	110

ПЕРЕДМОВА

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

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

СПИСОК ПРИЙНЯТИХ СКОРОЧЕНЬ

A – адреса, адресний вхід або вихід
AD – суматор
C – вхід тактових імпульсів
CD – кодер, шифратор
Ст – Counter – лічильник
CD – Data – дані, вхід даних
DB – двонаправлена шина даних
DC – дешифратор
EEPROM – репрограмований ПЗП з електричним стиранням даних
E – дозвіл, вхід дозволу
I – Input – вхід
MPL – програмований логічний пристрій
MUX – мультиплексор
PROM – програмований ПЗП
R – Reset – скид
RAM – пам'ять з довільною вибіркою (ОЗП)
RD – Read – читання
S – встановлення ВС у стан логічної “1”
T – Trigger – тригер
RG – регістр
WR – запис даних
АЦП – аналого-цифровий перетворювач
ВІС – велика інтегральна схема
ЗП – запам'ятовуючий пристрій
ІС – інтегральна схема
КМОП – комплементарна МОП-структура
ЛЕ – логічний елемент
МК – мікроконтролер
МП – мікропроцесор
МОН – метал-окисел-напівпровідник
МПС – мікропроцесорна система
БЕТ – багатомітерний транзистор
ОЗП – оперативний ЗП
ПЗП – постійний ЗП
ПЛМ – програмована логічна матриця
РЗП – регістр загального призначення
НВІС – надвелика ІС
Z – третій стан ЛЕ
ТТЛ – транзисторно-транзисторна логіка
ТТЛШ – ТТЛ з діодами Шоткі
ЦАП – цифро-аналоговий перетворювач
ЦП – цифровий пристрій
ША – шина адреси

ШД – шина даних
ШУ – шина управління
ШФ – шинний формувач
ЕЗЛ – емітерно-зв'язана логіка

1. ВСТУП

1.1. Класифікація цифрових пристроїв

За способом подання інформації системи зв'язку поділяють на аналогові та цифрові. Цифрові системи за точністю обробки сигналу суттєво перевищують аналогові, дозволяють значно збільшити пропускну здатність та завадостійкість, зменшити вплив температури та розкид параметрів на роботу пристроїв.

Ці переваги сприяли тому, що сьогодні цифрова техніка продовжує бурхливо розвиватися.

Пристрої цифрової техніки можна поділити на два класи:

– **імпульсні цифрові пристрої або пристрої першого класу** – це пристрої формування, генерування та перетворення завданих фізичних сигналів у сигнали зі стандартними логічними параметрами. До першого класу належать тактові генератори, формувачі імпульсних сигналів, аналого-цифрові та цифро-аналогові перетворювачі, амплітудні обмежувачі та інші;

– **цифрові пристрої другого класу** – це пристрої перетворення логічних сигналів або цифрових кодів. Такі пристрої носять назву *цифрових автоматів*. До цих пристроїв належать схеми від найпростіших логічних елементів до мікропроцесорів та мікро-ЕОМ. Тут перетворення інформації здійснюється виконанням певної послідовності арифметичних та логічних операцій.

Найпростіші логічні елементи виконують лише одну логічну операцію відповідно до законів бульової алгебри. Розробка ж пристроїв другого класу – це складна схемотехнічна задача, даними якої є функції алгебри логіки.

У загальному випадку на вхід цифрового пристрою надходить множина двійкових змінних x_1, x_2, \dots, x_n , а з виходу знімається множина двійкових функцій y_1, y_2, \dots, y_n . Реальні цифрові автомати мають *скінчену* множину внутрішніх станів і тому їх називають *скінченими*.

Розрізняють *синхронні* та *асинхронні цифрові автомати* в залежності від того, чи вони використовують *синхронізуючі* або *тактові імпульси*, чи ні.

В *асинхронних* цифрових автоматах цих імпульсів немає, через що будь-яка зміна вхідних сигналів відразу викликає певну зміну сигналів вихідних (рис. 1.1,а). Якщо вхідний сигнал $x = 1$ надійшов у момент t_1 , то й вихідний сигнал $y = 1$ з'явився в той же самий момент t_1 . Сигнал $x = 0$ в момент t_2 одразу викликає вихідний сигнал $y = 0$.

У *синхронних* цифрових автоматах обов'язково є *синхронізуючі* або *тактові імпульси*. Через це зміна вихідних сигналів відбувається не одразу після появи вхідних сигналів, а за їхньої наявності лише за командою тактових імпульсів, тобто вхідний сигнал тільки готує автомат для спрацьовування, а саме спрацьовування здійснює тактовий імпульс.

У синхронному цифровому автоматі поява вхідного сигналу $x = 1$ в момент t_3 (рис. 1.1,б) не викликає зміни вихідного сигналу y , а лише готує автомат для спрацьовування, яке відбудеться в момент t_4 за командою тактового імпульсу C . Аналогічно вхідний сигнал $x = 0$, який надійшов у

момент t_5 , викличе появу вихідного сигналу $y = 0$ лише в момент t_6 , коли надійде тактовий імпульс C .

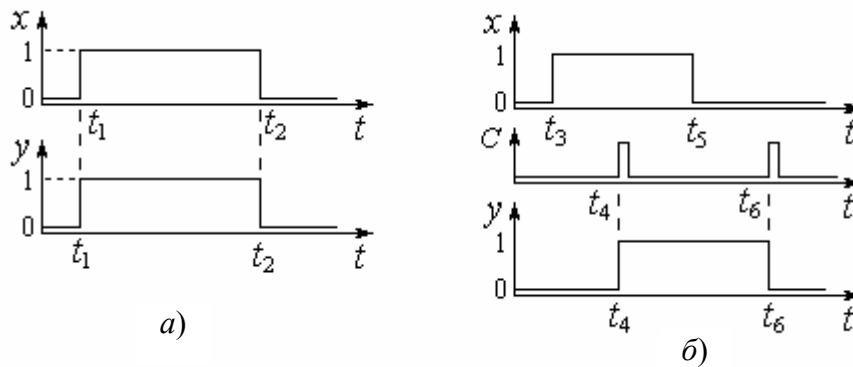


Рисунок 1.1 – Часова діаграма роботи цифрових автоматів: а) асинхронних; б) синхронних

Цифровий автомат, який має лише один внутрішній стан, називається *автоматом без пам'яті*. Вихідний сигнал автомата без пам'яті залежить тільки від комбінації логічних сигналів на входах у *даний момент часу* і не залежить від попередніх станів. Такий цифровий автомат асинхронний і його називають *комбінаційним пристроєм*.

На відміну від комбінаційних пристроїв, значення вихідних сигналів у цифрових автоматах з пам'яттю залежить не тільки від значень вхідних сигналів в даний момент часу, але й від їхніх *попередніх значень*. Такі пристрої реалізують функціональний зв'язок між послідовностями та значеннями вхідних і вихідних сигналів. Тому цифрові автомати з пам'яттю називають *послідовнісними пристроями*.

1.2. Цифрові сигнали

Носієм інформації в електронних пристроях є електричний сигнал. У більшості випадків це напруга або струм. Як функція часу, електричний сигнал може бути неперервним або дискретним.

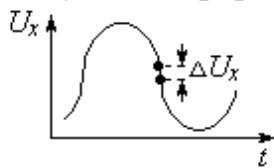


Рисунок 1.2 – Аналоговий сигнал

Аналоговий – це неперервний сигнал (рис 1.2). Інформаційні параметри U_x такого сигналу можуть приймати довільну множину значень в діапазоні зміни часу і тому відстань ΔU_x між двома сусідніми значеннями може бути скільки завгодно малою.

Цифровий сигнал – дискретний або квантований.

Він набуває одне з фіксованих певних значень аргументу $n\tau$ (де τ – інтервал дискретності, а $n = 0, 1, 2, \dots$). Тому дискретний сигнал може мати тільки обмежене число рівнів напруги або струму, які зафіксовані на інтервалах часу $n\tau$.

Дискретний сигнал може бути поданим у найрізноманітніших формах, де окремі елементи значень закодовані системою числення, що характеризується певною сукупністю символів за певною сукупністю правил.

Людина у повсякденному житті користується десятковою системою числення, яка складається з 10 цифр від 0 до 9. Однак цифрові сигнали, створені на основі десяткової системи числення, технічно реалізувати досить важко, тому що в цифрових пристроях в цьому разі треба мати десять рівнів напруги або струму. Через це десяткова система числення не є економічною і в цифровій схемотехніці застосовується лише для зчитування та введення інформації людиною.

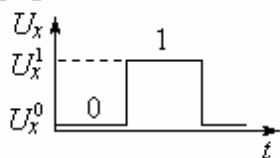


Рисунок 1.3 – Двійковий сигнал

У цифрових пристроях найбільшого розповсюдження знайшла двійкова система – найпростіша з усіх систем числення. Вона складається лише з двох рівнів напруги, які кодуються символами 0 та 1. Двійкова система дозволяє найефективніше відобразити інформацію, якщо, наприклад, низький рівень напруги U_x^0 позначити як логічний нуль “0”, а високий рівень напруги U_x^1 – як логічну одиницю “1” (рис. 1.3).

Цифрові сигнали забезпечують більш високу завадостійкість, ніж аналогові. Це пояснює рис. 1.4. Тут надані аналоговий u_a та цифровий $u_{ц}$ сигнали, які уражені однією й тією самою завадою $u_{зав}$ (рис. 1.4,а та б).

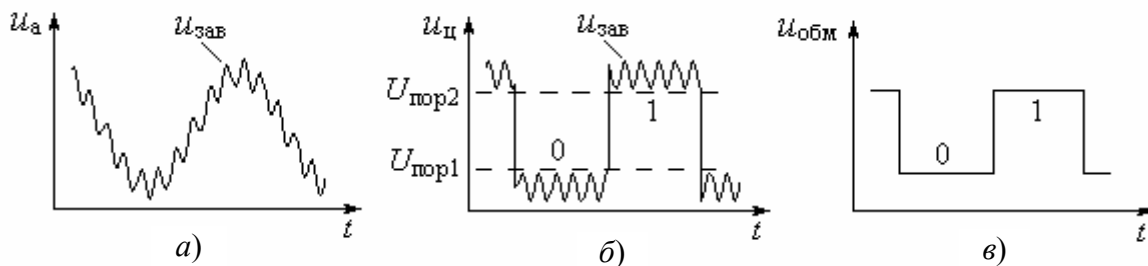


Рисунок 1.4 – Вплив завади на сигнал: а) аналоговий; б) цифровий; в) обмежений цифровий сигнал

В аналоговому сигналі u_a напруга завади $u_{зав}$ спотворює усі інформаційні параметри сигналу, тобто миттєві значення напруги, частоти і фази, і позбавитись її дії ніяк неможливо.

Проте в цифровому сигналі $u_{ц}$ та ж сама напруга завади $u_{зав}$ спотворює тільки амплітуду, яка не є інформаційним параметром. Інформаційні ж параметри (рівні логічного “0” та логічної “1”) вона не зміщує. Якщо цифровий сигнал $u_{ц}$, уражений завадою (рис. 1.4,б), пропустити через обмежувач зверху з порогом обмеження $U_{пор1}$ та $U_{пор2}$, то завада опиниться поза цими порогом. Тому на виході згаданого обмежувача сигнал $u_{обм}$ (рис. 1.4,в) позбавлений завади. Отже, на цифрові пристрої завада певної амплітуди не діє, тобто цифрові сигнали забезпечують завадостійкість, вищу за аналогові. Через це тепер проектують тільки *цифрові системи передачі* (ЦСП).

Двійкова цифра (0 або 1) носить назву “*біт*” інформації. Послідовна комбінація бітів зображує всю інформацію про подію і носить назву “*слова*”, в якому кожний біт є одиницею інформації або даних. Одиницею зображення та обміну даних, якою як єдиним цілим оперують між собою окремі цифрові пристрої чи вузли, є восьмирозрядне двійкове число, яке носить назву *байт*.

Таким чином, 1 байт містить 8 біт. Кожна цифра, що входить у число або слово, залежить від місця її розташування в записі числа, тобто від позиції цифри.

2. ОСНОВИ ЦИФРОВИХ ПРИСТРОЇВ

2.1. Арифметичні основи

2.1.1. Системи числення

Система числення – це сукупність прийомів і правил, зображення чисел цифровими знаками. Системи числення діляться на позиційні та непозиційні.

Непозиційна система числення – це система, у якій значення символу не залежить від його положення у числі. Непозиційні системи виникли раніше позиційних систем. Вони використовувались ще здавна.

Прикладом непозиційної системи числення, що дійшли до наших днів, є римська система числення. У ній число подається у вигляді послідовності ряду, що складається з символів. Наприклад, число $30_{(10)}$ подається трьома символами XXX, де X у будь-якому місці означає $10_{(10)}$.

Позиційна система числення – це система, у якій значення символу залежить від його позиції у ряду цифр, що зображують число. Позиція символу у зображенні числа називається *розрядом*.

Позиційні системи вважаються більш зручними для обрахунку операцій та отримали найбільш широке розповсюдження.

Кількість різних символів, що використовуються у позиційній системі числення для подання всіх чисел у межах розряду, називається її *основою* та позначається літерою S .

У двійковій системі числення ($S = 2$) використовується два символи 0 та 1, у десятичній ($S = 10$) – десять символів від 0 до 9.

Числа у таких системах подаються послідовністю ряду цифр, розділених на 2 групи:

- 1) групу розрядів, що зображує цілу частину числа;
- 2) групу розрядів, що зображує дробову частину числа.

Все число можна подати у вигляді:

$$N = a_2 \cdot S^2 + a_1 \cdot S^1 + a_0 \cdot S^0 + a_{-1} \cdot S^{-1} + \dots \quad (2.1)$$

Одиниці кожного числа присвоюється вага S^k , де S – основа системи числення; k – номер розряду, дорівнюваний індексу при символах, що зображують цифри розрядів.

2.1.2. Переведення числа з однієї системи числення в іншу

В процесі перетворення інформації, яку несуть цифрові сигнали, в цифровій системі виникає необхідність переведення чисел з однієї системи числення в іншу еквівалентну.

Правило переведення цілих чисел N_p , поданого у системі числення з основою p , у систему числення з основою q засновується на методі ділення/множення.

Таке перетворення легко здійснити, якщо володіти правилами, що складаються з таких послідовних кроків:

- ділення в цілих числах заданого числа A_p на основу p тієї системи, в яку переводиться дане число A_p ;
- якщо частка не дорівнює нулю, то її слід взяти за нове число і повторити попередню операцію; якщо частка дорівнює нулю, то перейти до наступного пункту;
- виписати всі остачі в зворотному порядку, починаючи з останньої, і взяти їх за цифри шуканого числа $a_{n-1}, a_{n-2}, \dots, a_1, a_0$.

Розглянемо приклад: нехай треба перетворити десяткове число 13 у двійкове. Кроки при цьому наступні:

$$\begin{array}{r} -13 \mid 2 \\ \underline{-12} \mid 6 \\ 1 \end{array} \qquad \begin{array}{r} -6 \mid 2 \\ \underline{-6} \mid 3 \\ 0 \end{array} \qquad \begin{array}{r} -3 \mid 2 \\ \underline{-1} \mid 1 \\ 1 \end{array} \qquad \begin{array}{r} -1 \mid 2 \\ \underline{-0} \mid 0 \\ 1 \end{array}$$

Над цифровим сигналом у двійковому кодї можна виконувати звичайні арифметичні та логічні операції.

2.1.3. Двійкова арифметика

Арифметичні операції (додавання, віднімання, множення, ділення) над однорозрядними двійковими числами дуже прості, а над багато розрядними числами виконуються за тими самими правилами, що і у десятковій системі числення.

Таблиця 2.1

Додавання	Віднімання	Множення
$0 + 0 = 0$	$0 - 0 = 0$	$0 \times 0 = 0$
$0 + 1 = 1$	$1 - 0 = 1$	$0 \times 1 = 0$
$1 + 0 = 1$	$1 - 1 = 0$	$1 \times 0 = 0$
$1 + 1 = 10$ (перенесення 1)	$0 - 1 = 1$ (займання 1)	$1 \times 1 = 1$

Додавання двійкових чисел виконується у стовпчик, починаючи з молодшого розряду. При додаванні двійкових чисел слід уважно слідкувати за перенесенням у старший розряд (приклад 1).

Віднімання двійкових чисел виконується за звичайним правилом. При відніманні іноді виникає необхідність займати одиницю зі старшого розряду. Ця займана одиниця дорівнює двом одиницям молодшого розряду. Займання виконується кожного разу, коли цифра у розряді від'ємника більша, ніж у розряді зменшуваного (приклад 2).

<p>Приклад 1</p> $\begin{array}{r} 1001 \text{ (} 9_{10}\text{)} \\ + 0101 \text{ (} 5_{10}\text{)} \\ \hline 1110 \text{ (} 14_{10}\text{)} \end{array}$	<p>Приклад 2</p> $\begin{array}{r} 1001 \text{ (} 9_{10}\text{)} \\ - 0101 \text{ (} 5_{10}\text{)} \\ \hline 0100 \text{ (} 4_{10}\text{)} \end{array}$
---	--

Операція додавання та віднімання виконується лише над модулями; знак результату визначається окремо в результаті аналізу знака доданків.

Для виконання арифметичних операцій двійкове число кодується спеціальними машинними кодами: прямими, додатковими та зворотними, які

дозволяють замінити операції віднімання операціями додавання, що спрощує побудову арифметико-логічних пристроїв.

Множення двійкових багаторозрядних чисел містить операції визначення знака множення та визначення його абсолютної величини. Знаковий розряд множення може бути отриманий додаванням цифр знакових розрядів співмножників без формування переносу.

Абсолютну величину значення множення отримуємо шляхом перемноження без врахування їхніх знаків. Множення багаторозрядних двійкових чисел виконується на основі таблиці двійкового множення (табл. 2.1). При множенні двох двійкових чисел множник послідовно помножується на кожен цифру іншого множника, починаючи з молодшої або старшої.

Приклад:
Приклад 1

$$\begin{array}{r} 1101 (13_{10}) \\ \times 101 (5_{10}) \\ \hline 1101 \\ 1101 \\ \hline 1000001 (65_{10}) \end{array}$$

Ділення двійкових багаторозрядних чисел містить у собі дві операції – визначення знака частки і визначення його абсолютної величини.

Множення і ділення двійкових чисел, як правило, здійснюється на спеціалізованих арифметичних процесорах (співпроцесорах) або програмним способом на мікропроцесорах.

Арифметичні операції (додавання, віднімання, множення, ділення) над двійковими числами здійснюються за допомогою спеціальних алгоритмів, які використовуються у двійковій системі числення. Двійкові коди, що несуть числову інформацію, називають *операндами*, де комбінація символів “0” та “1” – це двійкові числа, тобто числа у двійковій системі числення.

2.2. Логічні основи цифрових пристроїв

Цифрові сигнали, що несуть логічну, а не числову інформацію про стан цифрової системи, характеризуються тільки двома значеннями: логічний “0” або логічна “1”. Логічні операції досить легко здійснити за допомогою апарата алгебри логіки або бульової алгебри.

2.2.1. Основні поняття бульової алгебри

Алгебра логіки (бульова алгебра) – це формальний апарат, за допомогою якого описується логічна сторона процесів у цифрових пристроях. Алгебра логіки має справу з логічними змінними, які набувають лише два значення ТАК(1) або НЕ(0). **Алгебра логіки – це алгебра станів**, а не алгебра чисел.

Логічні змінні досить точно описують стан таких механічних об'єктів, як реле, кнопки, тумблери та інші пристрої, що можуть перебувати у двох різноманітних станах: вимкнено – розімкнено. До таких об'єктів можна віднести напівпровідникові схеми, на вході яких може бути лише один із двох станів напруги.

Найчастіше високий рівень напруги приймають за логічну одиницю (1), а низький – за логічний нуль (0).

Основне поняття алгебри логіки – висловлення, тобто речення, в якому міститься суть твердження істинності або його заперечення. Кожне висловлення можна позначити символом x або y , і вважати, що $x = 1$ або $y = 1$, якщо висловлення істинне, а $x = 0$ або $y = 0$, якщо висловлення неістинне. В алгебрі логіки кожній двійковій змінній x ставиться у відповідність обернена до неї інверсна змінна \bar{x} . Наприклад, якщо $x = 0$, тоді $\bar{x} = 1$ (риска “–” означає операцію логічного заперечення).

Логічна функція $y = f(x_1, x_2, \dots, x_n)$ – це складне висловлення з кількох простих логічних операцій. Функція виду $y = f(x_1, x_2, \dots, x_n)$ називається *бульовою*, якщо вона та її аргументи (x_1, x_2, \dots, x_n) , можуть набувати лише два значення: “0” або “1”. Таку назву надано на шану англійського математика кінця ХІХ століття Джорджа Буля, який досліджував ці функції.

Оскільки аргументи бульової функції можуть набувати лише два значення, то область визначення будь-якої бульової функції завжди кінцева.

Сукупність значень бульової функції називається *набором*. Для кожної бульової функції від n змінних існують $Z = 2^n$ різних наборів, а число бульових функцій від n змінних дорівнює $2^Z = 2^{2^n}$.

Якщо логічна функція y набуває тільки одне значення при кожному значенні аргументу x , то вона носить назву *однозначної* і позначається формулою $y = f(x)$. *Двозначна функція* $y = f(x_1; x_2)$ залежить від 4-х комбінацій аргументів $x_1; \bar{x}_1; x_2$ та \bar{x}_2 і набуває значення y або \bar{y} . Кількість аргументів x може бути більша ніж два.

Для функції від однієї змінної ($n = 1$) існують $2^2 = 4$ різних бульових функцій, де $Z = 2^n = 2^1$ – число наборів, на яких функція визначена. Особливо цікава одна з них – інверсія $y = \bar{x}$, решта – тривіальна.

При $n = 2$, тобто для бульових функцій від двох змінних x_1 та x_2 існують $2^4 = 16$ різних функцій, кожна з яких визначена на 4-х наборах. Бульові функції двох змінних, їхнє умовне визначення та назва надані в табл. 2.2.

У бульовій алгебрі діють закони, за якими можна установити аналітичні зв'язки між різними логічними функціями і виконувати відповідні перетворення для спрощення складних виразів.

Серед 16 функцій (табл. 2.2) фактично тільки 10 залежать від двох змінних, решта 6 функцій залежать від двох змінних формально: функції f_0 та f_{15} – тривіальні, функції f_3 та f_5 – це повторення змінної, а функції f_{10} і f_{12} – їхні інверсії. Всі наведені 16 логічних функцій на практиці не застосовуються.

Таблиця 2.2 – Двійкові функції

Функція	Операція	Назва функції	Назва операції	Приклад				
				x_1	0	0	1	1
				x_2	0	1	0	1
f_0	0	Константа 0		0	0	0	0	0
f_1	$x_1 \wedge x_2 = x_1 x_2$	Кон'юнкція (логічне множення)	I		0	0	0	1
f_2	$x_1 \wedge \bar{x}_2 = x_1 \bar{x}_2$	Заборона x_2			0	0	1	0
f_3	x_1	Повторення змінної x_1			0	0	1	1
f_4	$\bar{x}_1 \wedge x_2 = \bar{x}_1 x_2$	Заборона x_1			0	1	0	0
f_5	x_2	Повторення змінної x_2			0	1	0	1
f_6	$x_1 \oplus x_2$	Виняткове АБО (сума за mod 2)			0	1	1	0
f_7	$x_1 \vee x_2 = x_1 + x_2$	Диз'юнкція (логічне додавання)	АБО		0	1	1	1
f_8	$\bar{x}_1 \vee \bar{x}_2 = \bar{x}_1 + \bar{x}_2$	Стрілка Пірса	АБО-НЕ		1	0	0	0
f_9	$x_1 \sim x_2$	Рівнозначність або еквівалентність			1	0	0	1
f_{10}	\bar{x}_2	Інверсія (заперечення) x_2	НЕ		1	0	1	0
f_{11}	$x_1 \vee \bar{x}_2 = x_1 + \bar{x}_2$	Імплікація від x_2 до x_1			1	0	1	1
f_{12}	\bar{x}_1	Інверсія (заперечення) x_1	НЕ		1	1	0	0
f_{13}	$\bar{x}_1 \vee x_2 = \bar{x}_1 + x_2$	Імплікація від x_1 до x_2			1	1	0	1
f_{14}	$\bar{x}_1 \wedge \bar{x}_2 = \bar{x}_1 \bar{x}_2$	Штрих Шеффера	I-НЕ		1	1	1	0
f_{15}	1	Константа 1			1	1	1	1

Можна зауважити, що деякі функції з табл. 2.2 отримують методом декомпозиції або перенумерації аргументів. Така операція носить назву *суперпозиції*. Так, наприклад, функція f_{12} може бути отримана з функції f_{11} , якщо x_1 перенумерувати на x_2 . Функція f_{14} може бути отримана з функції f_{12} шляхом підстановки замість аргументу x_1 функції f_1 .

Отже, застосовуючи метод суперпозиції, можна отримати більш складні логічні функції. При цьому виникає питання, чи можливо мати набір більше простих функцій, за допомогою яких можна було б отримати як завгодно складну логічну функцію. Це питання пов'язане з одним із основних понять бульової алгебри – функціонально повнотою систем логічних функцій.

Система бульових функцій вважається *функціонально повною*, якщо на її основі можна отримати довільну бульову функцію, застосовуючи лише метод суперпозиції. Можна отримати довільну множину функціонально повних наборів – базисів. Основні функції бульової алгебри, їх визначення та назва зведені в табл. 2.3.

Найбільш поширеною серед усіх наборів функцій є система A , що складається з трьох бульових функцій.

$$A \begin{cases} f_1 = x_1 \wedge x_2 = x_1 \cdot x_2 & - \text{кон'юнкція} \\ f_7 = x_1 \vee x_2 = x_1 + x_2 & - \text{диз'юнкція} \\ f_{12} = \bar{x}_1 & - \text{заперечення} \end{cases} \quad (2.2)$$

Властивість набору функцій виражати через себе будь-яку складну функцію носить назву *повноти цього набору*.

Властивість повноти має велику практичну цінність: дозволяє промисловості випускати обмежений набір логічних елементів, із яких можна побудувати будь-які логічні схеми, на базі яких будуються складні функціональні вузли цифрових систем.

Для згаданих операцій кон'юнкції, диз'юнкції та заперечення справедливі наступні аксіоми:

$$- \text{універсальна множина} \begin{cases} 1 \wedge x = 1 \cdot x = x, \\ 1 \vee x = 1 + x = 1; \end{cases} \quad (2.3)$$

$$- \text{нульова множина} \begin{cases} 0 \wedge x = 0 \cdot x = 0, \\ 0 \vee x = 0 + x = x; \end{cases} \quad (2.4)$$

$$- \text{повторення} \begin{cases} x \wedge x \wedge \dots \wedge x = x \cdot x \cdot \dots \cdot x = x, \\ x \vee x \vee \dots \vee x = x + x + \dots + x = x; \end{cases} \quad (2.5)$$

$$- \text{доповнення} \begin{cases} x \wedge \bar{x} = x \cdot \bar{x} = 0, \\ x \vee \bar{x} = x + \bar{x} = x; \end{cases} \quad (2.6)$$

$$- \text{подвійна інверсія} \quad \bar{\bar{x}} = x. \quad (2.7)$$

В алгебрі логіки діють: принцип дуальності, правила Шеннона і де Моргана.

Принцип дуальності є основним принципом бульової алгебри, згідно з яким дві функції дорівнюють одна одній, якщо на всіх можливих наборах змінних вони набувають одного й саме того ж значення, тобто

$$f_1(x_1, x_2, \dots, x_n) = f_2(x_1, x_2, \dots, x_n). \quad (2.8)$$

Правило Шеннона стверджує, що для отримання алгебраїчного виразу інверсної функції необхідно у заданій функції всі змінні замінити на інверсні їм, всі знаки кон'юнкції – на знаки диз'юнкції і, навпаки, всі знаки диз'юнкції – на знаки кон'юнкції.

Правило де Моргана стверджує, що інверсія кон'юнкції дорівнює диз'юнкції інверсій, а інверсія диз'юнкції – кон'юнкції інверсій.

У бульовій алгебрі діють закони, за якими можна установити аналітичні зв'язки між різними логічними функціями і виконувати відповідні перетворення для спрощення складних виразів.

Основні закони бульової алгебри та їхні аналітичні зображення зведені в табл. 2.3.

Таблиця 2.3 – Закони бульової алгебри

Назва закону	Аналітичне зображення
Закон комутативності	$x_1 x_2 = x_2 x_1$
Закон асоціативності	$x_1 (x_2 x_3) = (x_1 x_2) x_3 = x_1 x_2 x_3$
Закон дистрибутивності	$x_1 \cdot (x_2 \vee x_3) = x_1 x_2 \vee x_1 x_3$
Закон склеювання	$x_1 x_2 \vee x_1 \bar{x}_2 = x_1$
Закон поглинання	$x_1 \cdot (x_1 \vee x_2) = x_1; \quad x_1 \vee x_1 x_2 = x_1$
Закон дуальності (правило де Моргана)	$\overline{x_1 x_2} = \bar{x}_1 \vee \bar{x}_2$

Названі закони, аксіоми та правила не вичерпують усіх можливостей булевих рівностей, але вони є основними й їхні знання необхідні для виконання перетворень булевих функцій.

Логічні функції багатьох змінних отримують аналогічно розглянутому випадку для функції двох змінних застосуванням методу суперпозиції та аксіом і законів бульової алгебри. Слід зауважити, що базисні функції обов'язково містять у собі операцію інверсії.

Реалізація функцій можлива у вигляді електронних схем.

2.2.2. *Форми зображення логічних функцій*

Логічну функцію можна зобразити за допомогою таблиць, координатних карт або діаграм, словесно, у формі числового запису, аналітично. Розглянемо деякі форми зображення логічних функцій, які найбільш часто використовуються у цифровій схемотехніці.

Таблична форма зображення передбачає складення *таблиці істинності* логічної функції, яка містить у собі усі можливі поєднання значень аргументів та відповідні значення логічних функцій. Таблиця істинності має 2^n рядків по числу аргументів $x = n$. Кожен рядок має $(n + 1)$ стовпців, якщо логічна функція однозначна, або $(n + m)$ стовпців при багатозначній функції, де m – число значень функції y_1, y_2, \dots, y_m .

У кожному рядку розглядається комбінація станів аргументів x_n і записується відповідне значення функції y .

Таблиця 2.4

x	y
1	0
0	1

Наприклад, для функції інверсії з одним аргументом $y = \bar{x}$ таблиця істинності має 2 стовпці і 2 рядки (табл. 2.4).

Для однозначної функції y з двома аргументами x_1 та x_2 , таблиця істинності має $2^2 = 4$ рядки та $(2 + 1) = 3$ стовпці (табл. 2.5). Наприклад, якщо задана логічна

Таблиця 2.5

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

функція $y = x_1 + x_2$, то таблиця 2.5 істинності цієї функції має такий вигляд: табличне зображення логічної функції – це найпростіша та наочна форма. Однак зі зростанням числа n аргументів x_n табличне зображення є суттєво складним та громіздким, бо зростає число рядків та стовпців таблиці істинності. Так, для логічної функції y , що залежить від 4-х

аргументів x_1, x_2, x_3, x_4 , таблиця істинності буде мати $(4 + 1) = 5$ стовпців та $2^4 = 16$ рядків.

Таблична форма зображення логічних функцій широко використовується при розгляданні логічних функцій з малим числом аргументів. Для довільного числа змінних ця форма досить громіздка.

Словесне зображення логічної функції – це логічне висловлення, що підтверджує істинне значення функції y при істинному або неістинному значенні аргументів x . Наприклад, *словесне зображення інверсної функції* $y = \bar{x}$ має такий вигляд: функція $y = 1$, тобто істинна тоді і тільки тоді, коли x неістинний ($x = 0$), і функція $y = 0$ неістинна тоді, коли істинний аргумент, тобто $x = 1$.

Логічну функцію $y = x_1 + x_2$ *словесно можна зобразити* так: значення функції y дорівнює нулю тоді й тільки тоді, коли обидва аргументи нульові: $x_1 = 0$ та $x_2 = 0$; коли хоча б один із двох аргументів або x_1 , або x_2 , або обидва одиничні, то значення функції y також дорівнює одиниці.

При *аналітичному зображенні* функція задається алгебраїчним виразом, який отримують при застосуванні логічних операцій до змінних бульової алгебри. Така форма зображення зручна при аналізі властивостей функції, при виконуваних різноманітних аналітичних перетворень.

Найбільш раціональним є зображення логічної функції в так званих *канонічних формах*. Початковими, з умов вигоди подальших перетворень, прийняті дві канонічні форми зображення функцій: **удосконалена нормальна диз'юнктивна форма (УДНФ)** й **удосконалена нормальна кон'юнктивна форма (УКНФ)**.

Диз'юнктивною нормальною формою (ДНФ) називається така форма зображення функції, коли останню надано у вигляді диз'юнкції ряду членів, кожний з яких є простою кон'юнкцією аргументів або їх інверсій. Прикладом канонічної форми може бути вираз

$$f(x_1, x_2, x_3) = x_1 \vee x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \vee x_2 x_3. \quad (2.9)$$

Якщо у кожному члені ДНФ показані усі аргументи (або їх інверсії) функції, то така форма носить назву **удосконаленої диз'юнктивної нормальної форми (УДНФ)**. Вираз (2.9) не є прикладом УДНФ, тому що в ньому лише третій член складається з усіх аргументів функції f .

Для переходу від ДНФ до УДНФ треба в кожній з членів функції, де зображені не усі аргументи, ввести вираз виду $x_i \vee \bar{x}_i$. Як відомо за аксіомою доповнення (2.6) цей вираз дорівнює одиниці, тобто

$$x_i \vee \bar{x}_i = 1, \quad (2.10)$$

де x_i – відсутній в члені аргумент.

В такому разі, ввід виразу (2.10), що дорівнює одиниці, значення функції не змінює. Тому така операція задану функцію не пошкоджує.

Розглянемо перехід від ДНФ до УДНФ на наступному прикладі. Нехай задано функцію у формі ДНФ виду:

$$f(x_1, x_2, x_3) x_1 \vee x_2 \bar{x}_3. \quad (2.11)$$

Додавання в перший і другий члени функції (2.11) виразу виду $x_i \vee \bar{x}_i$ перетворить задану функцію до такого вигляду:

$$f(x_1, x_2, x_3) = x_1(x_2 \vee \bar{x}_2) \cdot (x_3 \vee \bar{x}_3) \vee x_2 x_3 \cdot (x_1 \vee \bar{x}_1). \quad (2.12)$$

Розкриття дужок та виконання операції логічного множення, дозволяє отримати задану функцію у вигляді (2.13)

$$f(x_1, x_2, x_3) = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee x_2 \bar{x}_3 \vee \bar{x}_1 x_2 \bar{x}_3. \quad (2.13)$$

З урахуванням аксіоми повторення (2.5) можна записати:

$$x_1 x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 = x_1 x_2 \bar{x}_3. \quad (2.14)$$

Формула (2.13) після зведення подібних членів за правилами (2.14) набуває остаточного вигляду:

$$f(x_1, x_2, x_3) = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee x_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3. \quad (2.15)$$

Остання отримана форма заданої функції – це **УДНФ**.

Досить просто отримати УДНФ, якщо функцію надано в табличній формі. Наприклад, функцію $f(x_1, x_2, x_3)$ надано у формі таблиці істинності табл. 2.6. Для цієї функції УДНФ має вигляд:

$$f(x_1, x_2, x_3) = \bar{x}_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3. \quad (2.16)$$

Таблиця 2.6

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Як бачимо з виразу (2.16), кожний член відповідає набору значень аргументів, за якого функція $f(x_1, x_2, x_3) = 1$. Кожний з наборів аргументів, за яких $f(x_1, x_2, x_3) = 1$ (ряди наборів 3, 4, 6 і 8), перетворює в одиницю відповідний член виразу (2.16), завдяки чому і вся функція дорівнює одиниці.

Можна визначити таке **правило запису УДНФ** функції, яка задана таблицею істинності.

Треба записати стільки членів у вигляді кон'юнкцій всіх аргументів, скільки одиниць належить функції в таблиці. Кожна кон'юнкція повинна відповідати визначеному набору аргументів, які перетворюють функцію в одиницю.

Якщо в цьому наборі значення аргументу дорівнює нулю, то в кон'юнкцію входить інверсія даного аргументу, при одиничному значенні аргументу останній входить в кон'юнкцію безпосередньо.

Слід зазначити, що кожна функція має лише одну УДНФ.

Кон'юнктивною нормальною формою (КНФ) називається форма запису функції у вигляді кон'юнкції деяких членів, кожен з яких є простою диз'юнкцією аргументів (або їх інверсій).

Прикладом КНФ може бути, наприклад, наступна функція:

$$f(x_1, x_2, x_3) = x_1(x_2 \vee \bar{x}_3) \cdot (x_1 \vee x_2 \vee \bar{x}_3). \quad (2.17)$$

В удосконаленій кон'юнктивній нормальній формі (УКНФ) в кожному члені КНФ обов'язково повинні бути подані усі аргументи. Наведена функція не може бути прикладом УКНФ, бо лише третій член має усі три її аргументи.

Для переходу від КНФ до УКНФ необхідно додати до кожного члена, де присутні не всі аргументи, члени виду $x_i \cdot \bar{x}_i$. Як відомо з аксіоми додавання (2.6)

$$x_i \cdot \bar{x}_i = 0. \quad (2.18)$$

Додавання до заданої форми функції виразу (2.18) не впливає на значення функції, тому така операція не може вважатися забороненою.

Розглянемо перехід від КНФ до УКНФ на прикладі функції

$$f(x_1, x_2, x_3) = x_1(x_2 \vee \bar{x}_3). \quad (2.19)$$

При додаванні до першого і другого членів заданої функції (2.19) відсутніх аргументів за правилами (2.18) отримують вираз виду:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \bar{x}_2 \vee x_3 \bar{x}_3) \cdot (x_2 \vee \bar{x}_3 \vee x_1 \bar{x}_1). \quad (2.20)$$

Для здобуття остаточного виразу даної функції у формі **УКНФ** треба розкрити дужки і виконати спрощення, внаслідок чого можна записати:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \cdot (x_1 \vee x_2 \vee \bar{x}_3) \cdot (x_1 \vee \bar{x}_2 \vee x_3) \cdot (x_1 \vee \bar{x}_2 \vee \bar{x}_3) \cdot (\bar{x}_1 \vee x_2 \vee \bar{x}_3). \quad (2.21)$$

Удосконалена КНФ легко будується за таблицею істинності. Для функції, яку задано табл. 2.6, УКНФ має вигляд:

$$f(x_1, x_2, x_3) = (x_1 \vee x_2 \vee x_3) \cdot (x_1 \vee x_2 \vee \bar{x}_3) \cdot (\bar{x}_1 \vee x_2 \vee x_3) \cdot (\bar{x}_1 \vee \bar{x}_2 \vee x_3). \quad (2.22)$$

Вираз має стільки членів, зв'язаних з операцією кон'юнкції, скільки нулів має значення функції $f(x_1, x_2, x_3)$ в таблиці істинності.

Таким чином, кожному набору значень аргументів, на якому функція дорівнює нулю, відповідає відносний член УКНФ, який набуває на цьому наборі значення нуля. Тому що члени УКНФ зв'язані операцією кон'юнкції, то при нульовому значенні одного із членів, уся функція перетворюється в нуль.

Можна записати правила запису УКНФ функції, заданою таблицею істинності.

Слід записати стільки кон'юнктивних членів, які визначають диз'юнкції усіх аргументів, за якого числа наборів значень аргументів функція дорівнює нулю. Якщо в наборі значення аргументів дорівнює одиниці, то в диз'юнкцію входить інверсія цього аргументу, при нульовому значенні аргументу він входить у диз'юнкцію безпосередньо.

Кожна функція має тільки одну УКНФ.

Аналітична форма зображення функції у вигляді УДНФ або УКНФ передбачає надання цих функцій шляхом суперпозиції спеціально введених допоміжних функцій: **мінтермів і макстермів**.

Мінтермом називають бульовий добуток (кон'юнкцію) від n змінних, в якому кожна змінна входить лише один раз у прямому або інверсному виді. **Макстермом** називають бульову суму (диз'юнкцію) від n змінних, в якій кожна змінна входить лише один раз в прямому або інверсному виді.

Кожна переключена функція від n змінних має таке число мінтермів і макстермів, що дорівнює числу наборів, тобто 2^n . Наприклад, для функції від двох змінних існує чотири доінтерми m_i і чотири макстерми M_i , які визначаються за формулами (2.23) і (2.24) відносно:

$$\begin{cases} m_0 = \overline{x_1 x_2} \\ m_1 = x_1 \overline{x_2} \\ m_2 = \overline{x_1} x_2 \\ m_3 = x_1 x_2 \end{cases} \quad (2.23)$$

$$\begin{cases} M_0 = x_1 \wedge x_2 \\ M_1 = \overline{x_1} \wedge x_2 \\ M_2 = x_1 \wedge \overline{x_2} \\ M_3 = \overline{x_1} \wedge \overline{x_2} \end{cases} \quad (2.24)$$

Структурна схема логічного приладу може бути побудована безпосередньо за канонічною формою УДНФ або УКНФ реалізованої функції.

Ці структурні схеми безумовно забезпечують правильне функціонування приладу, але здобуті таким чином схеми, як правило, досить громіздкі, потребують багато елементів для реалізації, мають низькі економічні показники і малу надійність.

В багатьох випадках можна спростити логічний вираз, залишаючи при цьому функціональну роботу без зміни, при цьому відповідна структурна схема виявляється більш простою.

Досить зручним і наглядним зображенням логічної функції є **координатний спосіб**, тобто у вигляді координатної карти логічних станів. Найбільшого поширення в інженерній практиці здобули карти мінтермів або **карти Карно**. Використовується цей метод для невеликого числа змінних (не більше 6). Карта Карно містить $m = 2^n$ клітинок, в кожній з яких записується один з $m = 2^n$ мінтермів. Карта Карно будується так, щоб сусідні клітинки відрізнялися значеннями лише однієї змінної. Сусідніми називаються такі клітинки, які дотикаються своїми сторонами, а також клітинки, що розташовані по краях карти (верхньому і нижньому рядках, правому та лівому стовпцях).

На рис. 2.1, *а* і *б* показано карти Карно для логічних функцій двох змінних.

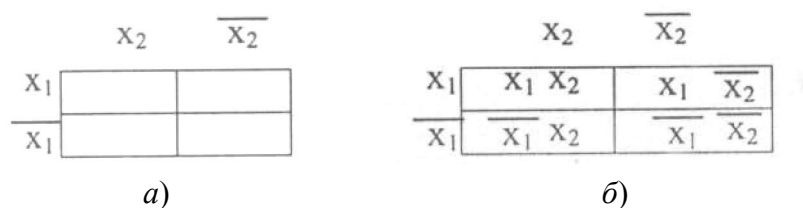


Рисунок 2.1 – Карти Карно для логічних функцій двох змінних: *а*) розподіл клітинок карти; *б*) запис мінтермів функції в кожену клітинку карти

Розподіл клітинок карти відбувається таким чином, щоб сусідні мінтерми відрізнялися тільки одним аргументом, який входить в обидва сусідні мінтерми – в один без інверсії, а в другий – з інверсією.

Аналогічно будуються карти Карно для трьох змінних (рис. 2.2). Така карта має 8 клітинок, що розподілені таким чином: 4 стовпця та 2 рядки.

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

a)

	x_2	x_2	\bar{x}_2	\bar{x}_2
x_1	$x_1x_2\bar{x}_3$	$x_1x_2x_3$	$x_1\bar{x}_2x_3$	$x_1\bar{x}_2\bar{x}_3$
\bar{x}_1	$\bar{x}_1x_2\bar{x}_3$	$\bar{x}_1x_2x_3$	$\bar{x}_1\bar{x}_2x_3$	$\bar{x}_1\bar{x}_2\bar{x}_3$
	\bar{x}_3	x_3	x_3	\bar{x}_3

б)

Рисунок 2.2 – Карти Карно для логічних функцій трьох змінних: а) розподіл клітинок карти; б) запис мінтермів функції в кожену клітинку карти

Для логічної функції, яку надано в УДНФ, в клітинках карти записуються одиниці відповідних мінтерм, для яких функція дорівнює одиниці. Значення функції, що дорівнює нулю, в карті не відображаються.

Якщо логічна функція трьох змінних визначена, наприклад, таблицею істинності (табл. 2.6), то її УДНФ відповідає виразу (2.25)

$$f(x_1, x_2, x_3) = \bar{x}_1x_2\bar{x}_3 \vee \bar{x}_1x_2x_3 \vee x_1\bar{x}_2x_3 \vee x_1x_2x_3. \quad (2.25)$$

Карта Карно для заданої функції складається шляхом постановки одиниць в ті клітинки карти (рис. 2.1), які містять мінтерми виразу УДНФ, останні клітинки – нулі – в карті не відображаються. Отриману карту Карно показано на рис. 2.3.

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

Рисунок 2.3 – Карта Карно для функції, визначеної формулою 2.25

За допомогою карт Карно можна спрощувати складні логічні вирази. На практиці часто доводиться будувати цифрові пристрої, коли форма зображення не завжди економна. Їм властива належність зайвих станів, що підлягає спрощенню або мінімізації.

2.2.3. Мінімізація логічних функцій

Мінімізація – це процес зведення логічної функції до такого виду, який припускає більш просту і дешеву її фізичну реалізацію. Всі названі методи мінімізації базуються на тождественних перетвореннях логічних виразів і поділяються на систематичні і несистематичні.

До найбільш поширеного систематичного методу відноситься **метод Клайна-Маккласкі**.

Пошуки мінімальної форми за цим методом виконуються за трьома етапами.

На першому етапі логічна функція перетворюється у форму УДНФ. Якщо функцію задано у формі УДНФ, необхідність першого етапу відпадає.

На другому етапі виконуються операції склеювання і поглинання (табл. 2.3).

Операція склеювання виконується шляхом пошуків у виразі **пар членів** виду $x_i \cdot x_j$ і $x_i \cdot \bar{x}_j$, які різняться лише тим, що один із аргументів входить без інверсій, а другий – з інверсією. Після цього виконується склеювання таких пар за формулою:

$$x_i \cdot x_j \vee x_i \cdot \bar{x}_j = x_i(x_j \vee \bar{x}_j) = x_i. \quad (2.26)$$

Результати склеювання вводять у вираз функції у вигляді додаткових членів.

Далі виконується операція поглинання. Вона базується на рівності

$$x_i \vee x_i \cdot x_j = x_i(1 \vee x_j) = x_i. \quad (2.27)$$

Таблиця 2.7

x_1	x_2	x_3	$f(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

В цьому разі член x_i поглинає член $x_i \cdot x_j$. При виконанні цієї операції з логічного виразу викреслюються всі члени, які поглинаються додатковими членами, що введені наприкінці проведення операції склеювання.

Таку процедуру послідовно проводять доти, поки виявляється можливим. Розглянемо приклад виконання цих операцій на функції, яку надано табл. 2.7.

Запишемо УДНФ функцію (табл. 2.7).

$$f(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3. \quad (2.28)$$

Порівняння усіх мінтермів попарно виявляє такі пари для склеювання:

$$\bar{x}_1 x_2 \bar{x}_3 \vee x_1 x_2 \bar{x}_3 \text{ (результат } x_2 \bar{x}_3);$$

$$x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \text{ (результат } x_1 \bar{x}_3);$$

$$x_1 x_2 x_3 \vee x_1 \bar{x}_2 x_3 \text{ (результат } x_1 \bar{x}_2);$$

$$x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 \text{ (результат } x_1 x_2);$$

$$x_1 \bar{x}_2 x_3 \vee x_1 x_2 x_3 \text{ (результат } x_1 x_3).$$

Результати склеювання запишемо додатковими членами в початковий вираз функції (2.28). Отримаємо такий вираз (2.29), куди входять і початкові мінтерми, і результати склеювання:

$$f(x_1, x_2, x_3) = \bar{x}_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 x_2 x_3 \vee x_2 \bar{x}_3 \vee x_1 \bar{x}_3 \vee x_1 \bar{x}_2 \vee x_1 x_2 \vee x_1 x_3. \quad (2.29)$$

Пошуки членів, над якими виконаємо операцію поглинання, приводять до наступного висновку: члени $x_2 \bar{x}_3$ поглинають перший і четвертий члени початкового виразу; члени $x_1 \bar{x}_2$ поглинають другий й третій члени; а $x_1 x_3$ – п'ятий член формули. Запишемо перетворений вигляд формули після першої стадії поглинання.

$$f(x_1, x_2, x_3) = x_2 \bar{x}_3 \vee x_1 \bar{x}_3 \vee x_1 \bar{x}_2 \vee x_1 x_2 \vee x_1 x_3. \quad (2.30)$$

Для останнього виду формули (2.30) треба повторити операцію склеювання:

$$x_1 \bar{x}_3 \vee x_1 x_3 \text{ (результат } x_1);$$

$$x_1\bar{x}_2 \vee x_1x_2 \text{ (результат } x_1\text{)}.$$

Знову введемо додатково результат другого склеювання x_1 у формулу (2.30). Отримаємо вираз:

$$f(x_1, x_2, x_3) = x_2\bar{x}_3 \vee x_1\bar{x}_3 \vee x_1\bar{x}_2 \vee x_1x_2 \vee x_1x_3 \vee x_1. \quad (2.31)$$

Результат склеювання x_1 поглинає другий, третій, четвертий і п'ятий члени останнього виразу (2.31). Формула (2.30) після другого циклу склеювання і поглинання набуває вигляду:

$$f(x_1, x_2, x_3) = x_2\bar{x}_3 \vee x_1. \quad (2.32)$$

Подальша робота з операціями склеювання і поглинання виявляється неможливою. Отримана форма запису функції носить назву скороченої СДНФ. У нашому прикладі вона збігається з мінімальною. Члени скороченої форми називаються **простими імплікантами функції**.

Перехід від скороченої форми СДНФ до мінімальної МДНФ виконується за допомогою **імплікантної матриці** (табл. 2.8). Вона складається зі стовпців, число яких дорівнює числу мінтермів заданої функції в УДНФ. Кожен мінтерм записується в один стовпець. Рядки таблиці – це прості імпліканти функції, записаної у скороченій формі СДНФ. Табл. 2.8 складена для нашого прикладу УДНФ (2.28) і СДНФ (2.32).

Таблиця 2.8

Прості імпліканти	Члени УДНФ				
	$\bar{x}_1x_2\bar{x}_3$	$x_1\bar{x}_2\bar{x}_3$	$x_1\bar{x}_2x_3$	$x_1x_2\bar{x}_3$	$x_1x_2x_3$
x_2x_3	X			X	
x_1		X	X	X	X

Відмічаються хрестиками (X) стовпці УДНФ, які поглинаються простими імплікантами. В нашому прикладі перша імпліканта $x_2\bar{x}_3$ поглинає такі мінтерми: $\bar{x}_1x_2\bar{x}_3$ та $x_1x_2\bar{x}_3$. Друга імпліканта x_1 поглинає 2, 3, 4 і 5 мінтерми.

Імпліканти, які входять у ядро, легко визначаються за імплікантною матрицею. Для кожної з них є хоча б один стовпець, який перекривається тільки даною імплікантною.

У нашому прикладі жодна із двох імплікант не є зайвою. Тому вони обидві входять до складу ядра, який визначає вигляд МДНФ.

Для отримання мінімальної форми в загальному випадку достатньо вибрати з імплікант такі, що не входять у ядро, і таке мінімальне їх число з мінімальною кількістю букв у кожній з них, яке забезпечує перекриття усіх стовпців імплікантної матриці, не перекритих членами ядра.

Перехід від СДНФ до МДНФ виконується шляхом зменшення зайвих імплікант.

Слід зауважити, що розглянутий метод мінімізації найбільш ефективний при великому числі змінних (понад 6). Мінімізація таких складних функцій частіше за все відбувається за допомогою комп'ютерів.

Для функцій із числом змінних до 6 (такі функції вважаються більш поширеними) мінімізація відбувається за **табличним методом Вейча-Карно**. Цей метод був спочатку розроблений Вейчем, а потім удосконалений Карно.

Особливістю цього методу є те, що в його основу покладено табличну форму запису мінтермів функції, яку надано у вигляді УДНФ.

При порівнянні з методом Квайна, тут мінімізація відбувається ручним способом і не потребує використовувати ПК.

Хоча метод Вейча-Карно є табличним, спрощення відбувається за рахунок таких самих операцій склеювання і поглинання. В цьому разі цей процес досягається завдяки запису функції у вигляді карт Карно.

Усі клітинки карти Карно, що мають одиниці, об'єднують у замкнені області. При цьому кожна область повинна мати вигляд прямокутника з числом клітинок, що визначаються, як 2^k , де $k = 0, 1, 2, \dots$. Таким чином, число клітинок в області може дорівнюватися 1, 2, 4, 8, Області можуть перехрещуватися (як в нашому випадку). Одні клітинки можуть входити в різні області. Зауважимо, що будувати карту Карно для логічної функції двох змінних недоцільно. У цьому випадку простіше функцію мінімізувати аналітичним методом. Складемо для прикладу карту Карно для функції, яку надано табл. 2.7, а УДНФ – табл. 2.28.

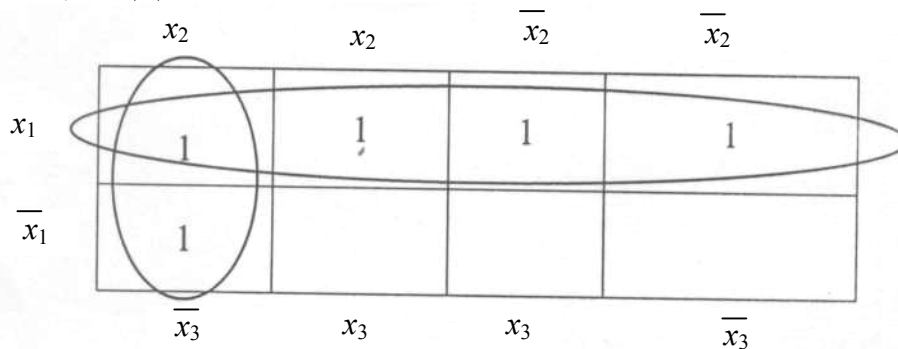


Рисунок 2.4 – Процес мінімізації функції (2.27) за допомогою карт Карно

Процес мінімізації зводиться до знаходження найбільших за площею покриттів сусідніх (заповнених) клітинок. Чим більше клітинок карти Карно в покритті, тим простіше імпліканта. При цьому слід прагнути до того, щоб кожна заповнена одиницями клітинка обов'язково входила в яке-небудь покриття. Для нашого випадку карта Карно повинна мати два покриття (дві перекриті області). Кожна з областей в МДНФ входить у запис членом, число букв в якому на k менше загального числа аргументів функції n (тобто дорівнює $n-k$). Кожний член МДНФ складається лише з тих аргументів, які повторюються в кожній клітинці для даної області і мають значення або з інверсією, або без інверсії.

Для нашого прикладу всі клітинки, що мають одиниці, складають дві області. Для обох областей число аргументів заданої УДНФ $n = 3$. У першу область входять $2^2 = 4$ клітинки. Тобто в МДНФ ця область буде входити членом, що складається з $n - k = 3 - 2 = 1$ літери.

Друга область має тільки $2^1 = 2$ клітинки. Число букв, які будуть визначати другу область, дорівнює $n - k = 3 - 1 = 2$.

Після цього виконується запис виразу МДНФ функції. Першій області відповідає член (імпліканта) x_1 , бо лише він один є спільним для чотирьох мінтермів верхнього ряду; другій області відповідає член (імпліканта) $x_2\bar{x}_3$ тому, що в мінтерми цієї області третій член x_1 входить і без інверсії, і з інверсією. Наприкінці об'єднуються отримані імпліканти диз'юнкцією, при цьому отримується мінімізована диз'юнктивна нормальна форма МДНФ заданої функції.

Таким чином, МДНФ заданої (2.28) функції записується так:

$$f(x_1, x_2, x_3) = x_1 \vee x_2 \bar{x}_3. \quad (2.33)$$

Результат мінімізації збігається з попереднім (2.32), отриманий при виконанні цього процесу методом Квайна.

Вся процедура мінімізації за методом Вейча-Карно полягає у послідовному виконанні таких операцій:

- складання карти Карно і розміщення конституент 1 в ті клітинки, які відповідають мінтермам даної функції, яку зображено в УДНФ;
- об'єднання сусідніх клітинок карти Карно, що мають одиниці контурами по 2, 4, ...;
- зчитування імплікант, що входять у даний контур, вилучаючи з них за законами склеювання та поглинання ті змінні, які утворюють доповнення, тобто входять перший раз з інверсією, а другий – без інверсії;
- об'єднання отриманих імплікант диз'юнкцією, яка відповідає МДНФ заданої функції.

При мінімізації функції малої кількості змінних (не більше трьох) досить ефективним є **метод безпосередніх перетворень**. Це аналітичний метод спрощення логічних функцій за допомогою аксіом та законів бульової алгебри.

У деяких випадках ефективніше мінімізувати інверсію логічної функції. Наприклад, при застосуванні методу Вейча-Карно у тих випадках, коли в заповненій карті Карно порожні місця (для 0) переважають над клітинками з одиницями. Тоді, маючи МДНФ інверсної функції, при технічній реалізації цієї функції достатньо на виході побудованої схеми підключити інвертор для відновлення прямої функції.

2.3. Логічні функції бульового базису

Набір з трьох логічних функцій, що носять назву НЕ, І, АБО називають бульовим базисом. За допомогою законів алгебри можна висловити різноманітні логічні функції через набір трьох найпростіших.

2.3.1. Функція логічного заперечення НЕ

Найпростішою логічною функцією одного аргументу є *функція логічного заперечення* або *інверсії*

$$y = \bar{x}. \quad (2.34)$$

Логічний елемент, що виконує цю функцію носить назву *інвертора* або *схеми НЕ*.

Умовне позначення інвертора показано на рис. 2.5,а. У релейно-контактній схемі інверсну функцію реалізує розмикаючий контакт реле, який замкнутий, поки в обмотці реле нема струмкового сигналу, тобто аргумент $x = 0$ (рис. 2.5,б)

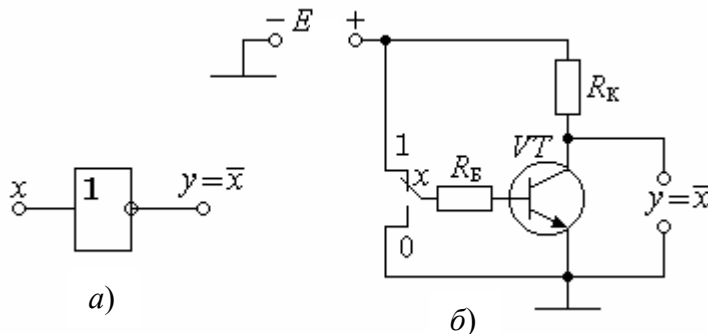


Рисунок 2.5 – Інвертор або схема НЕ: а) умовне позначення; б) реалізація за допомогою розмикаючого контура реле

При $x = 1$ вхід інвертора буде підключений до $+E$, через що транзистор VT відкритий і підключає вихід y до нуля, тобто $y = 0$ і навпаки, при $x = 0$ вхід інвертора підключений до нуля, транзистор VT закритий і підключає вихід y через R_K до $+E$, тобто $y = 1$.

Таблиця істинності (табл. 2.9) логічної функції НЕ (заперечення) має найпростіший вигляд.

Таблиця 2.9

x	y
1	0
0	1

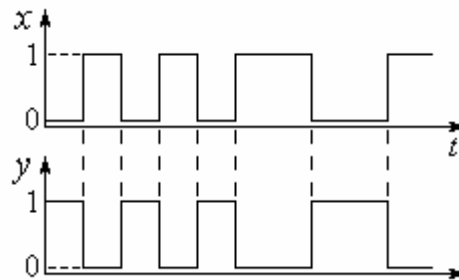


Рисунок 2.6 – Часова діаграма схеми НЕ

Часова діаграма роботи схеми логічного заперечення показана на рис. 2.6, з якої видно наступне.

Вхідний сигнал U_x складається з черги логічних нулів «0» та одиниць «1». Діаграма $U_y = f(t)$ вихідного сигналу теж набуває значення логічних нулів «0» або одиниць «1». Якщо до входу поданий рівень логічного «0» $x = 0$, то на виході з'являється логічна «1» $y = 1$ і навпаки, при логічній «1» на вході схеми $x = 1$ на виході устанавлюється рівень логічного «0» $y = 0$.

2.3.2. Функція логічного множення I

Функція логічного множення I – це функція двох або багатьох аргументів, що виконує операцію логічного множення або кон'юнкцію:

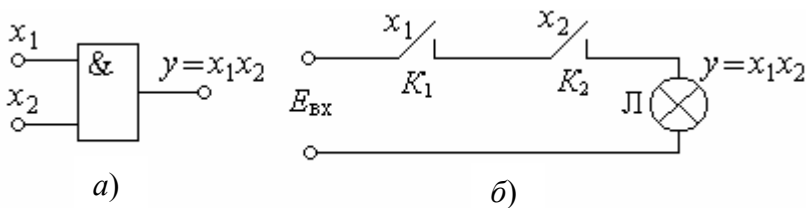
$$y = x_1 \wedge x_2 = x_1 x_2. \quad (2.35)$$

Функція I приймає значення логічної 1 тоді і тільки тоді, коли всі її аргументи x_1, x_2, \dots, x_n дорівнюють 1.

У релейно-контактній схемі функція І реалізується послідовним включенням двох або більшого числа (за кількістю аргументів x) замикаючих контактів, які керуються сигналами-аргументами x_n (рис. 2.7,б). Струм по колу пройде тільки тоді, коли усі контакти замкнуті, тобто знаходяться в одиничному стані. Якщо лише один контакт K_n буде розімкненим (його нульовий стан), то струм відсутній і функція дорівнює 0.

Умовне позначення логічного елемента, що реалізує функцію 2І (кон'юнктора), показано на рис. 2.7,а (цифра 2 показує наявність двох входів).

У релейно-контактній техніці функція І реалізується послідовним включенням двох (K_1, K_2) або більшого числа (за числом аргументів n) замикаючих контактів, які керуються сигналами-аргументами x_n (рис. 2.7, б).



Таблиця 2.10

x_1	x_2	y
1	1	1
0	1	0
1	0	0
0	0	0

Рисунок 2.7 – Двовходовий кон'юнктор або схема 2І:
а) умовне позначення; б) реалізація за допомогою контактів реле

Лампа Л буде світитися тільки тоді, коли і контакт K_1 і контакт K_2 будуть замкнені, тобто перебувати в одиничному стані. Якщо хоча б один із контактів буде розімкненим (в нульовому стані), лампа світитися не буде, тобто функція дорівнює 0.

Двовходова функція 2І – однозначна. Тому для кожної комбінації значення аргументів x_1 та x_2 даної функції існує тільки одне значення функції y . Таблиця істинності для схеми 2І – це табл. 2.10.

Часова діаграма роботи логічної схеми 2І показана на рис. 2.8, де вхідними сигналами є групи П-імпульсів x_1 та x_2 , які діють на відповідних входах. Вихідним сигналом є y .

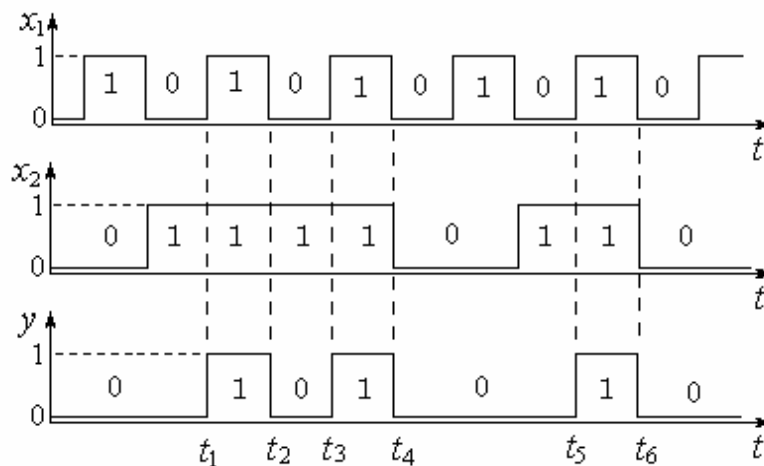


Рисунок 2.8 – Часова діаграма схеми 2І

Часова діаграма (рис. 2.8) наочно ілюструє алгоритм роботи функції, що впливає з таблиці істинності. Дійсно, рівень логічної «1» на виході y устанавлюється і підтримується тоді і тільки тоді, коли і на вході x_1 , і на вході x_2 присутні логічні «1», тобто $x_1 = x_2 = 1$. Збіг логічних «1» на входах x_1 та x_2 спостерігається в часи $t_1 - t_2$; $t_3 - t_4$ та $t_5 - t_6$ на часовій діаграмі. Протягом решти часу, коли на одному вході $x_1 = 0$ або на вході $x_2 = 0$, або обидва входи $x_1 = 0$; $x_2 = 0$ знаходяться під нульовим потенціалом, на виході схеми устанавлюється рівень $y = 0$.

2.2.3. Функція логічного додавання АБО

Функція логічного додавання АБО двох чи багатьох аргументів x_n виконує операцію логічного додавання або диз'юнкцію і записується формулою

$$y = x_1 \vee x_2 = x_1 + x_2. \quad (2.36)$$

Функція АБО набуває значення логічної «1» тоді, коли хоча б один із аргументів x_n дорівнює одиниці. Тут знак « \vee » або «+» означає логічне, а не алгебраїчне додавання.

Так, логічне додавання дає $y = x_1 \vee x_2 \vee \dots \vee x_n = 1 + 1 + \dots + 1 = 1$.

Умовне позначення логічного елемента, який реалізує функцію 2АБО, показано на рис. 2.9,а.

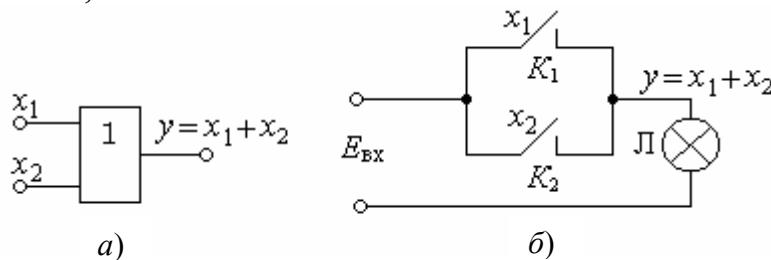


Рисунок 2.9 – Двовходовий диз'юнктор або схема 2АБО: а) умовне позначення; б) реалізація за допомогою контактів реле

У релейно-контактній техніці функція АБО реалізується паралельним з'єднанням двох або більшого числа (за числом аргументів n) замикаючих контактів $K_1, K_2 \dots$ (рис. 2.9,б).

Лампа Л буде світитися тоді, коли хоча б один контакт або K_1 ($x_1 = 1$), або K_2 ($x_2 = 1$), або обидва ($x_1 = 1; x_2 = 1$) будуть замкнені, тобто перебувати в одиничному стані ($y = 1$). Лампа Л буде погашеною ($y = 0$) лише тоді, коли всі контакти будуть розімкненими (у нульовому стані), тобто $x_1 = 0; x_2 = 0$.

Струм по колу (рис. 2.9,б) пройде тоді, коли будуть замкнені контакти K_1 або K_2 , або обидва контакти. При цьому на виході схеми y встановлюється рівень логічної «1». Лише у разі, коли обидва контакти розімкнені, тобто $x_1 = x_2 = 1$, на виході y з'являється логічний «0».

Роботу схеми 2АБО відбиває таблиця істинності (табл. 2.11). Ця функція, як і 2І, теж однозначна.

Роботу схеми 2АБО можна проілюструвати часовою діаграмою (рис. 2.10), з якої видно, що достатньо хоча б на одному з входів x_1

Таблиця 2.11

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	1

або x_2 мати рівень 1, щоб на виході сигнал y дорівнював би 1. Тільки тоді, коли на обох входах $x_1 = 0; x_2 = 0$ (моменти часу $t_2 - t_3, t_4 - t_5, t_6 - t_7, t_8 \dots$) на виході $y = 0$.

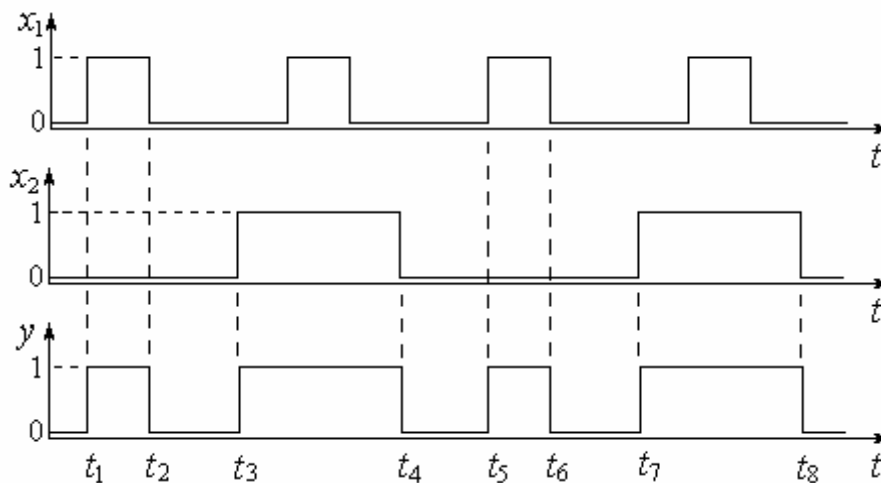


Рисунок 2.10 – Часова діаграма схеми 2АБО

2.3.4. Комбіновані логічні функції

Кожна з розглянутих функцій бульового базису набуває значення логічного “0” або логічної “1” в залежності від аргументів x_n , що входять до складу даної функції.

Якщо мати логічні елементи, які реалізують найпростіші функції НЕ, І, АБО, то можна побудувати більш складні функції шляхом послідовного виконання функціональних залежностей, які зв’язують пари змінних. Таке з’єднання простих логічних елементів дає змогу отримати цифрові пристрої, що реалізують складні логічні функції.

За основний критерій при структурній реалізації береться мінімум апаратних витрат, під яким слід розуміти мінімальну кількість логічних елементів та мінімальне число зв’язків між ними. Значну роль у забезпеченні критерію за мінімумом апаратних затрат відіграє елементний базис, тобто певний набір функціонально повних логічних елементів. Якщо, як у нашому випадку, базис наперед заданий (бульовий), то логічну функцію, яку треба побудувати, необхідно спеціально перетворити. Головна мета цих перетворень – зведення виразу функції до заданого базису.

Наприклад, треба отримати функцію 2АБО-НЕ. Вона утворюється з двох бульових функцій 2АБО та НЕ.

Схемотехнічно така задача розв’язується шляхом послідовного з’єднання логічних елементів 2АБО та НЕ (рис. 2.11,а).

Це з’єднання реалізує функцію 2АБО-НЕ, яка носить назву стрілки Пірса й умовно позначається , як показано на рис. 2.11,б.

Роботу схеми 2АБО-НЕ описує таблиця істинності (табл. 2.12).

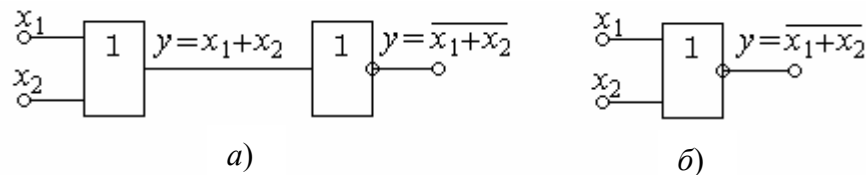


Рисунок 2.11 – Двовходова схема 2АБО-НЕ: а) послідовне з'єднання логічних елементів 2АБО та НЕ; б) умовне позначення

Часова діаграма роботи схеми 2АБО-НЕ показана на рис. 2.12, з якої видно наступне.

Таблиця 2.12

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	0

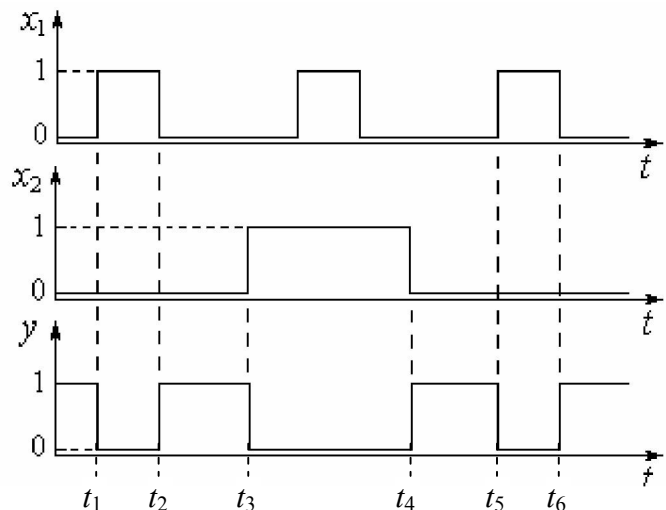


Рисунок 2.12 – Часова діаграма схеми 2АБО-НЕ

Якщо обидва входи нульові ($x_1 = 0$ та $x_2 = 0$), значення функції $y = 1$ одиничне.

Коли ж або $x_1 = 1$, або $x_2 = 1$, або обидва аргументи одиничні, то значення функції $y = 0$.

Якщо порівняти роботу схем 2АБО (рис. 2.10) та 2АБО-НЕ (рис. 2.12), то можна побачити, що вся різниця полягає лише в інверсії вихідного сигналу y .

Аналогічно можна отримати логічну функцію 2І-НЕ за допомогою двох логічних функцій бульового базису 2І та заперечення НЕ (рис. 2.13). Функція 2І-НЕ носить назву “штрих Шеффера”.

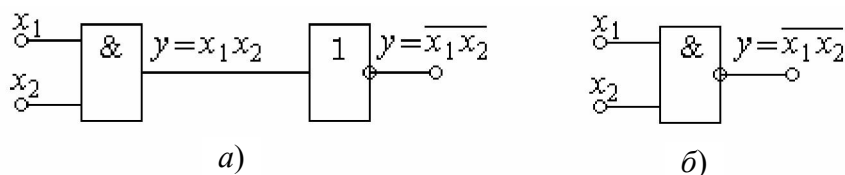


Рисунок 2.13 – Двовходова схема 2І-НЕ: а) послідовне з'єднання логічних елементів 2І та НЕ; б) умовне позначення

Елемент Шеффера будується послідовним з'єднанням двох логічних елементів, що реалізують функції 2І та НЕ. Тому таблиця істинності (табл. 2.13) є проінвертованою таблицею істинності (табл. 2.10) функції 2І.

Часова діаграма роботи логічної схеми 2І-НЕ (рис. 2.14) утворюється інверсією вихідного сигналу y логічної схеми 2І (див. рис. 2.8).

Таблиця 2.13

x_1	x_2	y
1	1	0
0	1	1
1	0	1
0	0	1

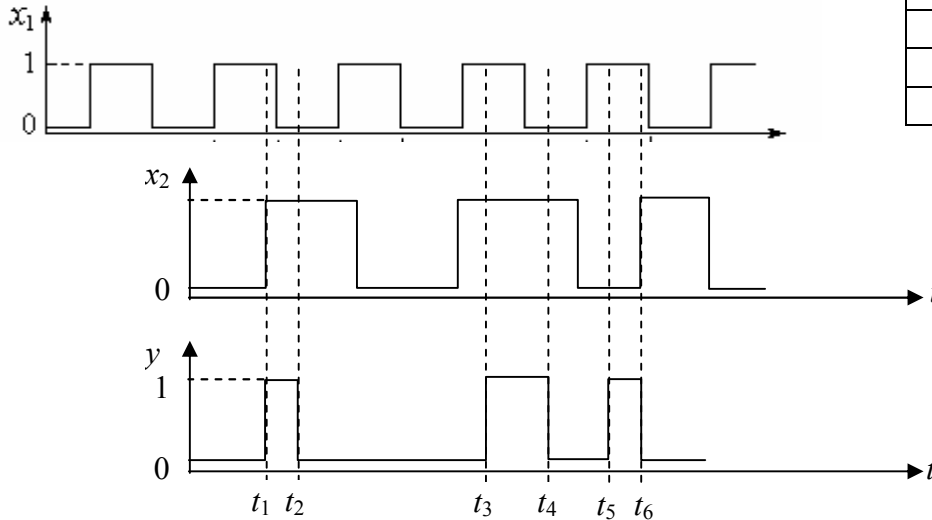


Рисунок 2.14 – Часова діаграма схеми 2І-НЕ

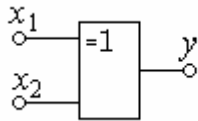


Рисунок 2.15 – Умовне позначення логічного елемента «виняткове АБО»

У цифрових функціональних пристроях комбінаційного типу багато застосовується логічний елемент, що реалізує функцію «виняткове АБО» (рис. 2.15). Цю функцію іноді називають «сума за mod 2».

Якщо обидва входи одиничні ($x_1 = 1$ та $x_2 = 1$), тільки тоді значення функції $y = 0$ нульове.

Коли ж або $x_1 = 0$, або $x_2 = 0$, або обидва аргументи нульові, то значення функції $y = 1$.

Логічна функція «виняткове АБО» описується формулою

$$x_1 \oplus x_2 = x_1 \bar{x}_2 \vee \bar{x}_1 x_2 = (x_1 \vee x_2)(\bar{x}_1 \vee \bar{x}_2). \quad (2.37)$$

Символ \oplus означає, що змінні x_1 та x_2 пов'язані логічною функцією «виняткове АБО».

Її робота полягає в тому, що $y = 1$ тоді і тільки тоді, коли вхідні змінні різні $x_1 \neq x_2$. Коли ж усі змінні однакові $x_1 = x_2$, тобто $x_1 = 0$ та $x_2 = 0$ або $x_1 = 1$ та $x_2 = 1$, лише тоді $y = 0$ (табл. 2.14).

Таблиця 2.14

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

Виняткове АБО має властивості комутативності, асоціативності, дистрибутивності відносно кон'юнкції. Для неї також справедливі аксіоми бульової алгебри. На основі властивостей і аксіом функції «виняткове АБО» можна отримати функції елементарного базису.

Часова діаграма роботи схеми, яка реалізує логічну функцію «виняткове АБО», показана на рис. 2.16.

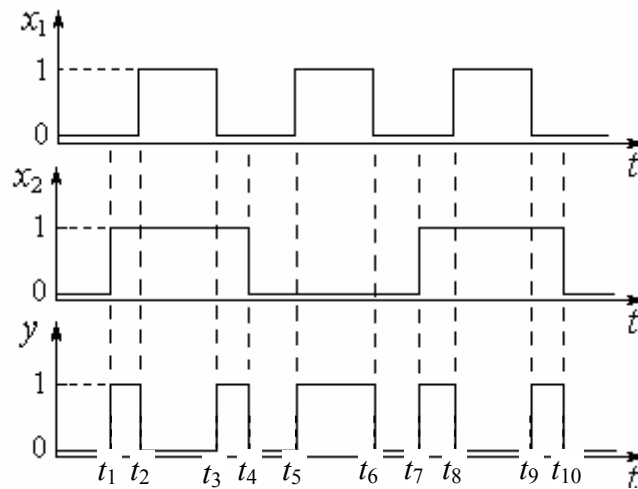


Рисунок 2.16 – Часова діаграма «виняткове АБО»

З цієї діаграми видно, функція $y = 1$ набуває одиничного значення лише тоді, коли обидва аргументи різні $x_1 \neq x_2$.

Контрольні питання

- 2.1. Що таке «цифровий автомат»?
- 2.2. Які цифрові автомати називають синхронними?
- 2.3. Які цифрові автомати називають асинхронними?
- 2.4. Що таке комбінаційний автомат?
- 2.5. Що таке послідовнісний автомат?
- 2.6. Які сигнали називають цифровими?
- 2.7. Яка система числення називається двійковою?
- 2.8. Що таке логічна функція?
- 2.9. Який набір логічних функцій вважається повним?
- 2.10. Які існують форми зображення логічних функцій?
- 2.11. Що таке логічна функція НЕ?
- 2.12. Що таке логічна функція І?
- 2.13. Що таке логічна функція АБО?
- 2.14. Функція стрілки Пірса?
- 2.15. Функція елемента Шеффера?
- 2.16. Функція елемента «виняткове АБО»?

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко В.Д. – Одеса: ОНАЗ ім. О.С.Попова, 2004. – Ч. 2. – С. 183 – 198.

2. Титце У. Полупроводниковая схемотехника: Справочное руководство / У. Титце, К. Шенк. – М.: Мир, 1982. – С. 100 – 102.

3. Хоровиц П. Искусство схемотехники / П. Хоровиц, У. Хилл. – М.: Мир, 1983. – Т. 1. – С. 499 – 515.

3. СХЕМОТЕХНІКА ІНТЕГРАЛЬНИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ

3.1. Класифікація та основні параметри логічних елементів

Залежно від схемотехнічного виконання базового логічного елемента (ЛЕ) сучасні цифрові схеми можна поділити на наступні типи:

ТТЛ – транзисторно-транзисторна логіка (універсальна, стандартна);

ТТЛШ – транзисторно-транзисторна логіка з діодом Шоттки;

КМОН – з комплементарними МОН-транзисторами;

ІІЛ – елементи інтегральної інжекційної логіки;

ЕЗЛ – емітерно-зв'язана логіка;

Елементи на МОН-транзисторах із заслоном Шоттки на основі арсеніду галію.

За принципом побудови активного елемента інтегральні схеми (ІС) поділяються на біполярні та польові.

За способом передавання інформації – на синхронні та асинхронні.

За типом інформаційних сигналів – на потенційні (основний тип), імпульсні та імпульсно-потенційні.

Потенційними називаються ЛЕ, для яких вхідні та вихідні сигнали задаються певними рівнями напруги («низька» – «висока»). Причому на тривалість затримки сигналів на вході (виході) не накладається ніяких обмежень. В імпульсних ЛЕ значення логічної змінної визначено інтервалом дії тактового сигналу. Вихідні сигнали визначаються або наявністю, або відсутністю тактового імпульсу.

Основне розповсюдження отримали потенційні логічні елементи, що мають значні переваги над іншими.

За технологічними та конструктивними ознаками ІС випускаються серіями. Серія – це сукупність ІС різного функціонального призначення, що мають загальні електричні та експлуатаційні характеристики, виконані за єдиною технологією, поєднані одним конструктивним рішенням корпусу.

Ступінь інтеграції – показник складності ІС – оцінюється числом елементів, розміщених у одному кристалі або на підкладці:

– мала інтегральна схема (МІС)	до 100;
– середня ІС (СІС)	101...1000;
– велика ІС (ВІС)	1001...10000;
– надвелика ІС (НВІС)	понад 10000.

Основною статичною характеристикою ЛЕ є передатна характеристика – це залежність вихідної напруги $U_{\text{вх}}$ від напруги на одному з входів за сталих напруг на інших, рівних U^0 або U^1 .

За видом передатної характеристики розрізняють інвертуючі (рис. 3.1,а) та неінвертуючі ЛЕ (рис. 3.1,б).

Передатні характеристики мають три чітко виражені ділянки.

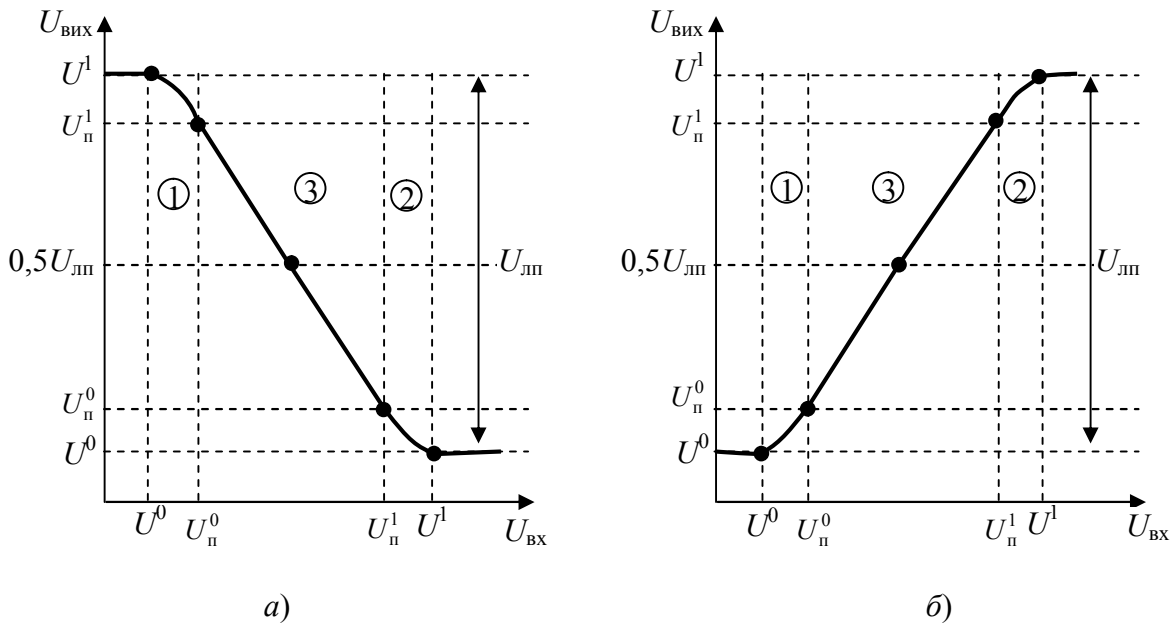


Рисунок 3.1 – Передатні характеристики ЛЕ: а) для інвертуючого ЛЕ; б) для неінвертуючого ЛЕ

Передатні характеристики мають три чітко виражені ділянки:

Ділянка 1. Стан високої вихідної напруги $U_{\text{вих}}^1$ для інвертуючого ЛЕ та низької вихідної напруги $U_{\text{вих}}^0$ для неінвертуючого ЛЕ. Рівень U^1 відповідає логічній «1», а рівень U^0 – логічному «0».

Ділянка 2. Стан $U_{\text{вих}}^0$ для інвертуючого ЛЕ та високої вихідної напруги $U_{\text{вих}}^1$ для неінвертуючого ЛЕ.

Ділянка 3 – це перехід з одного стану в інший: $U^1 - U^0$ або $U^0 - U^1$.

Межі ділянок визначаються точками одиничного підсилення

$$\frac{dU_{\text{вих}}}{dU_{\text{вх}}} = 1. \quad (3.1)$$

Вхідні напруги, що визначають межі ділянок, називаються *порогами перемикання* $U_{\text{п}}^0$ та $U_{\text{п}}^1$.

Різниця напруг логічної «1» та логічного «0» називають логічним перепадом

$$U_{\text{лп}} = U^1 - U^0. \quad (3.2)$$

Завадостійкість визначає припустиму напругу завад на входах МС та безпосередньо зв'язана з її передатною характеристикою. Залежно від тривалості завади розрізняють *статичну* та *динамічну завадостійкість*.

Якщо на вході діє низька напруга U^0 , то небезпечні завади, що мають позитивну полярність, тому що вони підвищують вхідну напругу. При цьому робоча точка на передатній характеристиці може зміститися в область перемикання, що призведе до збою у роботі. Максимально припустиму напругу позитивної завади можна визначити за передатною характеристикою як

$$U_{\text{зав}}^0 = U_{\text{п}}^0 - U^0. \quad (3.3)$$

Якщо на вході діє висока напруга U^1 , то небезпечна завада негативної полярності, що знижує рівень вхідної напруги. Максимальне значення завади за високим рівнем

$$U_{зав}^1 = U^1 - U_{п}^1. \quad (3.4)$$

Статична завадостійкість слугує основним показником захищеності мікросхеми від завад.

Динамічна завадостійкість вища статичної, тому що за короткочасних завад відбувається вплив паразитних ємностей та інерційність процесів у мікросхемі. Якщо тривалість імпульсу завади зменшується настільки, що стає меншим час перемикання ЛЕ, то припустима амплітуда імпульсної завади збільшується.

Вхідна характеристика – це залежність вхідного струму $I_{вх}$ від напруги на вході за сталих напруг на решті входах. Для ЛЕ на біполярних транзисторах за цією характеристикою визначають вхідні струми для низького та високого рівнів вхідної напруги. При $U_{вх} = U^0$ струм витікає з даного входу й при $I_{вх}^0 \leq 0$; при $U_{вх} = U^1$ струм втікає в цей вхід $I_{вх}^1 \geq 0$.

Вихідна характеристика – це залежність вихідної напруги від струму навантаження $I_{вих}$. Визначається також для низького рівня $U_{вих}^0 = f(I_{вих}^0)$ та для високого $U_{вих}^1 = f(I_{вих}^1)$, де $I_{вих}^0$, $I_{вих}^1$ – вихідні струми низького й високого рівнів.

Коефіцієнт об'єднання за входом $K_{об}$ визначається числом входів ЛЕ. ЛЕ масового виробництва виробляються з 2, 3, 4 та 8 входами. Збільшення входів, зазвичай, знижує швидкодію. Якщо виникає необхідність у збільшенні числа входів, то слід використати кілька однотипних ЛЕ, з'єднавши їх з урахуванням законів бульової алгебри.

Споживана потужність ЛЕ оцінюється як середня споживана потужність у статичному режимі:

$$P_{сер} = 0,5E_{ж}(I_{дж}^0 + I_{дж}^1), \quad (3.5)$$

де $I_{дж}^0$, $I_{дж}^1$ – струм, споживаний від джерела живлення, при $U_{вх} = U_{вих}^0$ та $U_{вих}^1$ відповідно; $E_{ж}$ – напруга джерела живлення.

Потужність, споживану додатково у процесі перемикання, називають *динамічною*. Вона пропорційна частоті перемикання ЛЕ, у зв'язку з чим її визначають за заданою частотою, що близька до максимальної.

Для оцінки часових властивостей мікросхем існує кілька параметрів. Швидкодію логічного елемента оцінюють середнім часом затримки розповсюдження сигналу $t_{зт\ сер}$, що означає середній час затримки виконання логічної операції:

$$t_{зт\ сер} = 0,5(t_{зт}^{0,1} + t_{зт}^{1,0}), \quad (3.6)$$

де $t_{зт}^{0,1}$, $t_{зт}^{1,0}$ – час затримки розповсюдження сигналу при переходах $U^0 \rightarrow U^1$ та $U^1 \rightarrow U^0$ відповідно.

Значення затримки розповсюдження оцінюється, звичайно, на рівні 0,5 від напруги логічного перепаду $U_{\text{лп}}$ (рис. 3.2). Інколи оцінку ведуть за часом затримки ввімкнення та вимкнення, які вимірюються на рівнях 0,1 або 0,9 від логічного перепаду $U_{\text{лп}}$.

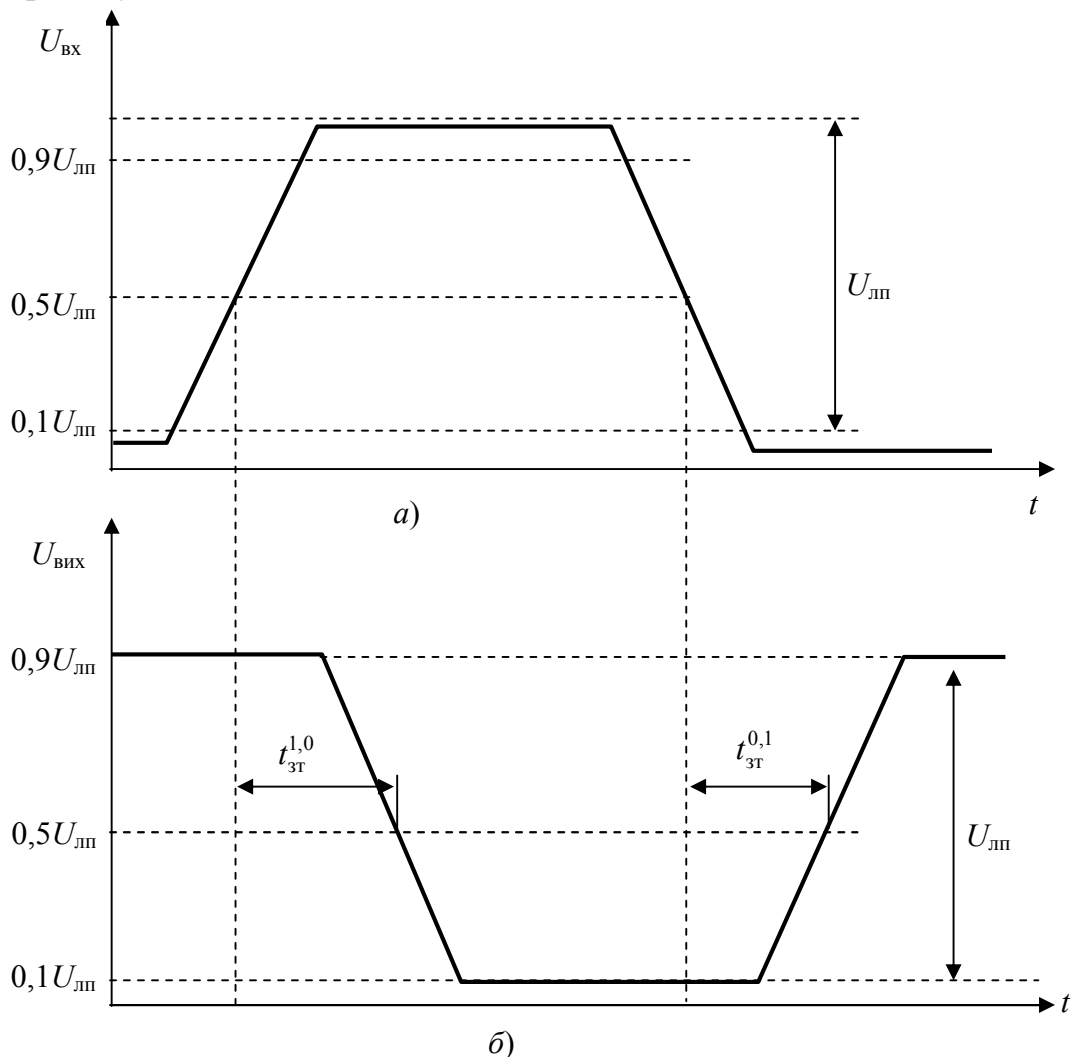


Рисунок 3.2 – Оцінка затримки сигналів: а) вхідний імпульс; б) вихідний імпульс з інверсією

Оцінку швидкодії послідовнісних пристроїв (тригерів, лічильників, регістрів) ведуть за максимальною частотою перемикання, часом затримки управляючих сигналів і т.п. Зменшити час затримки ЛЕ у певних межах можна шляхом збільшення струму, споживаного від джерела живлення, тому що при цьому зменшується час перезаряду паразитних ємностей. Тому існує зворотній зв'язок між часом затримки та споживаною потужністю: чим більший споживаний струм, тим менша середня затримка.

Для порівняння за швидкодією ЛЕ різних типів використовують параметр, називаний *енергією* (роботою) *перемикання*

$$A = P_{\text{сер}} t_{\text{зт сер}} \quad (3.7)$$

Для більшості серій цифрових мікросхем енергія перемикання знаходиться у межах 1...500 пДж для пристроїв малого ступеня інтеграції й 0,1...1,0 пДж для ЛЕ у ВІС та НВІС.

До інших показників можна віднести надійність та стійкість мікросхем до механічних та кліматичних впливів. Ці показники в інтегральних мікросхемах виключно значні. Ймовірність безвідмовної роботи за 15000 годин може скласти 99,8%. Мікросхеми можуть працювати за більших механічних навантажень (вібрації, удари тощо) у широкому температурному діапазоні (від -10 до $+70^{\circ}\text{C}$ для МС широкого застосування та від -60 до $+120^{\circ}\text{C}$ – спеціального) та значній вологості.

3.2. Транзисторно-транзисторна логіка (ТТЛ)

Схемотехніка ТТЛ має найбільше число різновидів. Мікросхеми ТТЛ на світовому ринку стабільно посідають перше місце протягом декількох десятиліть.

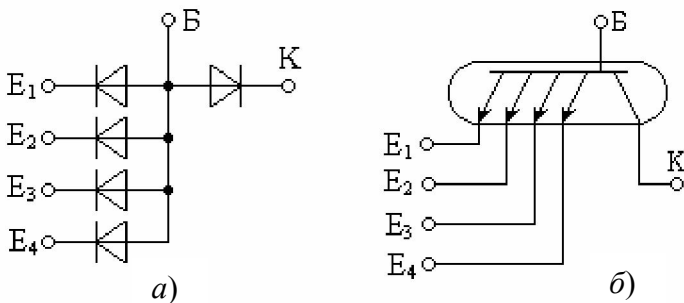


Рисунок 3.4 – Багатоемітерний транзистор: а) діодна збірка – еквівалент БЕТ; б) умовне позначення БЕТ

Особливістю мікросхем ТТЛ є наявність у вхідному колі *багатоемітерного транзистора*. Багатоемітерний транзистор (БЕТ) являє собою інтегральний елемент, який має декілька (до 8) емітерних переходів. БЕТ у мікросхемах ТТЛ вмикають на вході, реалізуючи при цьому функцію діодних збірок (рис. 3.4,а).

Умовне позначення БЕТ показано на рис. 3.4,б.

Типова двовходова схема 2І-НЕ показана на рис. 3.5. Число емітерів БЕТ дорівнює числу входів.

Рівнем логічного «0» є потенціал нульового проводу, тобто фізичний нуль, рівнем логічної «1» є напруга живлення $+E_K$.

Схема містить багатоемітерний транзистор $VT1$, який сумісно з $R1$ виконує логічну функцію 2І, фазорозщеплюючий каскад на $VT2$ та складний інвертор на транзисторах $VT3$ та $VT4$.

Фазорозщеплюючий каскад здійснює відкриття лише одного транзистора: або $VT3$, або $VT4$, забезпечуючи саме тим на виході відповідно або логічну одиницю, або логічний нуль.

При цьому складний інвертор, у якому один із транзисторів $VT3$ або $VT4$ завжди закритий, не споживає енергію.

Схема працює наступним чином.

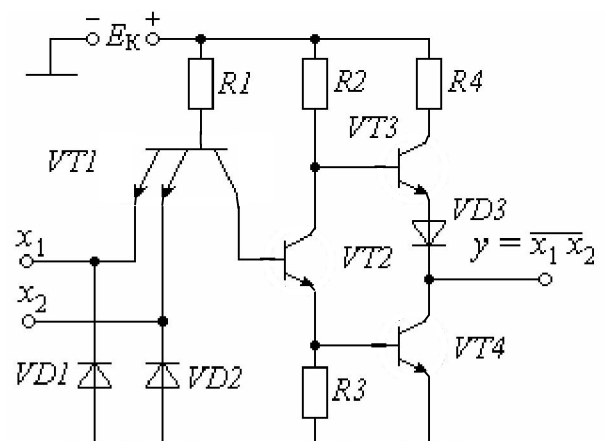


Рисунок 3.5 – Двовходова схема ТТЛ

Вхідне коло $R1$, $VT1$ є несиметричним розгалужуючим з'єднанням: одне коло містить лише один перехід (емітерний перехід $VT1$), а інше – три (колекторний $VT1$, емітерний $VT2$ та емітерний $VT4$). Через цю несиметричність струм бази $VT1$ може протікати тільки по одному колу: або через емітери, або через колектор $VT1$.

Напряга живлення E_K сумісно з опором резистора $R1$ створює джерело струму бази $VT1$, який тече завжди або по колах емітерів, або по колу колектора в залежності від стану входів x_1 та x_2 .

При обох одиничних входах $x_1 = 1$ та $x_2 = 1$ емітерні переходи $VT1$ закриті, через що струм бази $VT1$ через його колекторний перехід тече в базу транзистора $VT2$, відкриваючи його. Тоді емітерний струм транзистора $VT2$, втікаючи в базу $VT4$, відкриває і його.

Отже, при всіх одиничних входах транзистори $VT2$ та $VT4$ завжди відкриті. При цьому низький потенціал колектора $VT2$ закриває транзистор $VT3$.

Таким чином, при всіх одиничних входах закритий транзистор $VT3$ відключає вихід y від напруги E_K , а відкритий транзистор $VT4$ підключає вихід y до нуля.

Так, при всіх одиничних входах $x_1 = 1$ та $x_2 = 1$ вихід $y = 0$.

Якщо хоча б до одного входу надходить логічний "0", тобто $x_1 = 0$; $x_2 = 1$, або $x_1 = 1$; $x_2 = 0$, або $x_1 = 0$; $x_2 = 0$, то хоча б один або всі емітерні переходи $VT1$ відкриті і вони відгалужують на себе струм бази $VT1$. Бази $VT2$ і $VT4$ знеструмлені, через що транзистори $VT2$ та $VT4$ закриті. Одиничний потенціал колектора $VT2$ відкриває транзистор $VT3$.

Отже, закритий транзистор $VT4$ відключає вихід y від нуля, а відкритий транзистор $VT3$ підключає y до напруги E_K , тобто до логічної "1".

Таким чином, при хоча б одному вході під логічним "0" на виході є логічна "1" $y = 1$.

Схема здійснює логічну функцію 2І-НЕ і може бути описана таблицею істинності (табл. 3.1).

Щодо діодів $VD1 - VD3$, то $VD1$ та $VD2$ захищають входи схеми від переполюсовки напруги, а діод $VD3$ в емітері транзистора $VT3$ сприяє його закриттю.

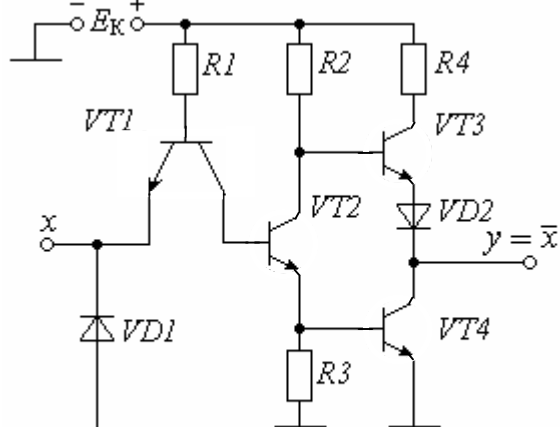


Рисунок 3.6 – ТТЛ інвертор

Таблиця 3.1

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	0

Емітери БЕТ $VT1$ можна з'єднувати між собою, зменшуючи при цьому кількість переходів (діодів) на вході мікросхеми. Нарешті, якщо з'єднати всі емітери між собою, то можна отримати одноемітерний транзистор.

Типовий інвертор ТТЛ або логічна схема НЕ показана на рис. 3.6.

Порівнюючи схему 2І-НЕ (рис. 3.5) зі схемою інвертора (рис. 3.6), переконуємося в тому, що вони відрізняються лише кількістю входів

(емітерів $VT1$). Тому принципи дії цих схем збігаються.

Щодо діодів $VD1$, $VD2$ у схемі (рис. 3.5) та $VD1$ у схемі (рис. 3.6), то вони необхідні для наступного.

Вхідні кола розглянутих схем мають розподілені ємності та індуктивності. Тому при дії на вході високочастотних сигналів можуть з'явитися “дренькоти” зі значними від’ємними викидами, які є згубними для схеми. З метою знищення цих викидів кожний вхід ТТЛ елемента з’єднують із загальною шиною через діод VD , як показано на рис. 3.6. Цей діод зменшує від’ємний викид до $-0,7$ В, підвищуючи, саме тим, завадостійкість мікросхеми та час її безвідмовної експлуатації.

Щодо вихідного кола, то через його двотактовість слід особливо попередити, що з’єднання елементів по виходах не припустиме.

Організація двотактового виходу в ТТЛ елементах робить неможливим паралельне з’єднання виходів декількох мікросхем, бо якщо з’єднані інвертори при цьому перебувають у різноманітних станах, то через відкриті транзистори $VT3$ одного з них (рис. 3.5, 3.6) та $VT4$ іншого потече неприпустимо великий струм, який може вивести з ладу з’єднані мікросхеми.

Цю проблему можна вирішити за рахунок використання логічного елемента з так званим *відкритим колектором*.

Схема логічного елемента з відкритим колектором створена тим, що з елементів (рис. 3.5, 3.6) вилучений транзистор $VT3$, а колектор $VT4$ виведений на вихід.

Спрощена схема логічного елемента з відкритим колектором та її умовне позначення показані на рис. 3.7. Схема має у своєму складі логічний елемент (в нашому випадку це схема 2І-НЕ), вихід якої підключений до бази транзистора VT (рис. 3.7,а). Колектор транзистора є виходом y .

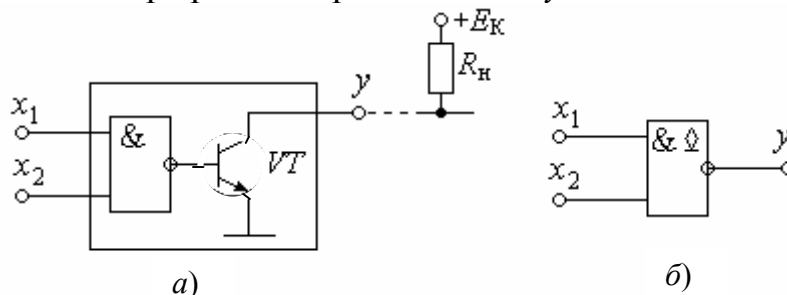


Рисунок 3.7 – Логічний елемент з відкритим колектором: а) спрощена схема; б) умовне позначення

Такий вихідний каскад (без опору навантаження в колекторному колі) не може сам собою сформувати на виході y високий рівень напруги. Для цього до виходу y зовнішнім монтажем підключається опір R_n , який носить назву *підтягуючого*. Замість резистора R_n можна підключати будь-яке зовнішнє навантаження: реле, світлодіод та ін.

Логічні елементи з відкритим колектором дозволяють підводити до вихідного транзистора напругу $E_k = 30$ В. За допомогою логічних елементів з відкритим колектором можна збільшувати кількість входів.

Крім схемотехнічних можливостей логічні елементи з відкритим колектором здатні реалізувати додатково і логічні операції завдяки тому, що вони допускають паралельне з'єднання аналогічних виходів на одне спільне навантаження. Таке об'єднання виходів називається *монтажною* або *провідниковою* логікою, бо вона утворюється в результаті зовнішнього монтажу.

На рис. 3.8 показана схема на логічних елементах 2І-НЕ з відкритим колектором, з'єднання яких на виході y разом з R_n утворюють *монтажне І*.

На об'єднаному виході y (рис. 3.8) рівень буде одиничним ($y = 1$) лише тоді, коли усі виходи y_i будуть одиничними: $y_1 = 1$; $y_2 = 1$; $y_3 = 1$.

Якщо хоча б один із елементів має на своєму виході низький потенціал $y_i = 0$, то на об'єднаному виході y рівень буде нульовим $y = 0$.

Отже виходи y_i разом з R_n утворюють *монтажне І*:

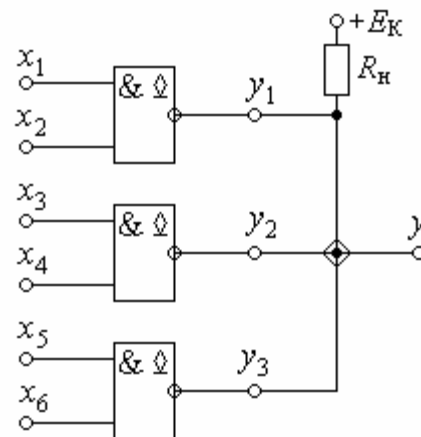


Рисунок 3.8 – Схема монтажного І

$$y = y_1 \cdot y_2 \cdot y_3 = \overline{x_1 x_2} \cdot \overline{x_3 x_4} \cdot \overline{x_5 x_6} \quad (3.9)$$

Принцип монтажного І значно використовується у ВІС пам'яті та програмованих логічних матрицях (ПЛМ), а також для побудови двонапрямлених числових шин.

Транзисторно-транзисторна логіка має низку переваг, а саме:

- досить мала споживана потужність, що досягається за рахунок ускладнення інвертора парою транзисторів $VT3$ та $VT4$ (див. рис. 3.5), які працюють у фазоінверсному режимі під дією керуючого фазорозщеплювача $VT2$;
- у статичному режимі один із транзисторів $VT3$ або $VT4$ завжди відкритий, тому вихідний опір, а тому й стала часу вихідного кола є досить малими, що сприяє підвищенню швидкодії;
- у динамічному режимі, коли на час переключення одночасно відкриті обидва транзистори $VT3$ і $VT4$ малий вихідний опір логічного елемента сприяє швидкому перезаряджанню ємності навантаження та паразитній ємності монтажу, що забезпечує високу навантажувальну здатність логічного елемента ТТЛ.

До недоліків серій ТТЛ відноситься порівняно низька швидкодія, тому що в статичному режимі окремі транзистори, що складають схему ТТЛ, знаходяться у стані глибокого насичення. Це насичення забезпечує надлишковий вхідний струм, який зміщує емітерний перехід у прямому напрямі і приводить до накопичення рухомих носіїв у базах транзисторів. Після зникнення вхідного струму для переходу транзистора у стан відсікання потрібний тривалий час для подолання перехідного процесу розсмоктування накопичених носіїв, коли протягом десятків наносекунд транзистор залишається відкритим.

Отже, для підвищення швидкодії треба запобігти насиченню транзисторів. Ці недоліки ліквідовані в ТТЛ елементах з діодом Шотткі* (ТТЛШ).

3.3. Транзисторно-транзисторна логіка з діодом Шотткі (ТТЛШ)

Більш швидкодіючими порівняно із серією ТТЛ є логічні елементи на діодах та транзисторах Шотткі (ТТЛШ). Мікросхеми ТТЛШ мають ту саму структуру електричної схеми, що й ТТЛ, тільки замість діодів та транзисторів використовуються напівпровідникові прилади з бар'єром Шотткі.

В діодах Шотткі VD (рис. 3.9,а) використаний перехід “метал-напівпровідник”. Пряма напруга цього переходу становить 0,35 ... 0,40 В, що значно менше, ніж у звичайних діодів (0,7 В).

Для зменшення насичення паралельно до колекторного переходу біполярного транзистора VT підключається діод Шотткі, чим і утворюється *транзистор Шотткі* (рис. 3.9,б).

При закритому транзисторі VT потенціал колектора є позитивним відносно бази, через що діод VD перебуває під зворотною напругою і не впливає на роботу ключа.

Коли ж транзистор VT відкритий, то діод VD знаходиться під прямою напругою і відгалужує частину струму бази VT у колекторне коло. Струм бази зменшується, через що зменшується накопичення рухомих носіїв заряду в базі. Тому при переході транзистора у стан відсікання буде зменшений час розсмоктування накопичених носіїв у базі.

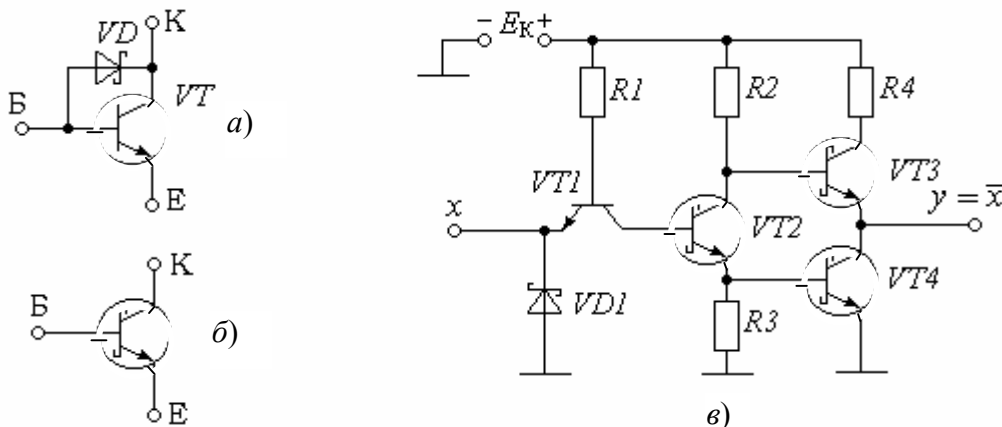


Рисунок 3.9 – Типовий інвертор ТТЛШ: а) БТ з діодом Шотткі; б) транзистор Шотткі; в) схема інвертора

Інвертор ТТЛШ (рис. 3.9,в) працює також як звичайна ТТЛ.

Струм бази $VT1$ може протікати або по вхідному колу x , або по колу бази $VT2$.

При нульовій вхідній напрузі ($x = 0$) струм бази $VT1$ тече по вхідному колу x і не втікає в базу $VT2$. Транзистори $VT2$ і $VT4$ закриті, через що вихід у

* Шотткі (Schottky) – німецький фізик, який відкрив у 1914 р. фізичне явище, що виникає при проходженні струму через контакт “метал-напівпровідник”, так званий бар'єр Шотткі.

відірваний від нуля і через відкритий $VT3$ підключений до $+E_K$, тобто до логічної «1» $y = 1$.

Коли ж напруга на вході x дорівнює логічній «1» ($x = 1$), то струм бази $VT1$ тече через його колекторний перехід у базу транзистора $VT2$, відкриваючи його. Відкривається і $VT4$, підключаючи вихід y до нуля і утворюючи тим самим на виході логічний «0» ($y = 0$).

Суттєве зменшення часу розсмоктування надлишкового заряду у схемі з використанням транзисторів Шотткі скорочує час затримки до 3 ... 5 нс.

Проте слід зауважити, що на відміну від ТТЛ елементи ТТЛШ мають нижчу завадостійкість унаслідок малого порога відкривання транзисторів Шотткі і зменшення їхнього насичення.

3.4. Емітерно-зв'язана логіка (ЕЗЛ)

Особливістю мікросхем транзисторної логіки з емітерними зв'язками є ненасичений режим роботи транзисторів, що забезпечує їхню високу швидкодію.

Базовою схемою ЛЕ емітерно-зв'язаної логіки (ЕЗЛ) є перемикач струму, який побудований за схемою диференційного підсилювача (рис. 3.10).

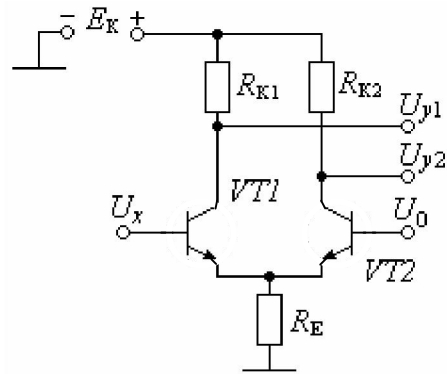


Рисунок 3.10 – Базова схема ЕЗЛ

Поріг переключення задається зовнішньою опорною напругою U_0 . За будь-якого цього порогу відкриті транзистори не насичені. В результаті обом транзисторам $VT1$ та $VT2$ не потрібний час на розсмоктування надлишкового заряду неосновних носіїв у базах, і через що швидкодія досить висока.

Однак разом з тим слід визначити незручність безпосереднього спраження логічних елементів ЕЗЛ з іншими ЛЕ, яка зумовлена від'ємними логічними рівнями. Крім того, ЛЕ ЕЗЛ споживають досить велику потужність.

3.5. Інтегральна інжекційна логіка (I^2L)

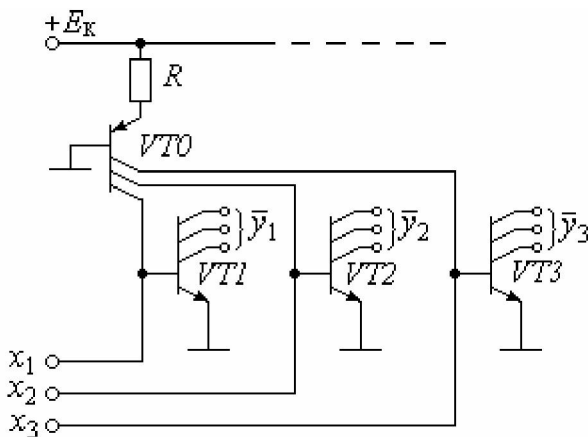


Рисунок 3.11 – Базові елементи I^2L

Логічні елементи інтегральної інжекційної логіки (I^2L) реалізують на базі транзисторних структур $n-p-n$ та $p-n-p$ типів, які з'єднують за схемою, що показана на рис. 3.11.

Базовими елементами I^2L є каскади-інвертори на транзисторах $VT1$, $VT2$, $VT3$..., які можуть мати декілька колекторів-виходів \bar{y} , що виконані в об'ємі загальної p -бази, аналогічно емітерам багатоемітерного транзистора в ТТЛ.

Каскад на транзисторі $VT0$ типу $p-n-p$ називається *інжектором* і здійснює інжекційне живлення інверторів $VT1, VT2, VT3 \dots$. Кількість колекторів транзистора $VT0$ може досягати 10...20 і навіть більше. Тому таку ж кількість каскадів може живити інжектор.

Щодо живлення, то воно охоплює тільки бази $VT1, VT2, VT3 \dots$. Колектори ж цих транзисторів залишаються відкритими.

Резистор R сумісно з напругою живлення E_K створює джерело струму, тобто обмежує струм, який розгалужується по базах $VT1, VT2, VT3 \dots$, здійснюючи їхнє живлення. Без резистора R струми живлення могли б зростати необмежено.

Навантажувальна здатність ЛЕ $I^2Л$ визначається числом колекторів транзисторів $VT1, VT2, VT3 \dots$.

Інвертори $I^2Л$ працюють наступним чином.

Якщо на входах x_1, x_2, x_3 діють логічні одиниці, то бази транзисторів $VT1, VT2, VT3 \dots$ знаходяться під струмами. Виходи $\bar{y}_1, \bar{y}_2, \bar{y}_3$ через насичені транзистори підключаються до загальної шини і тому мають логічні "0".

Коли ж на входах x_1, x_2, x_3 будуть логічні "0", то струми колекторів інжектора відгалужуються на ці відповідні входи і бази $VT1, VT2, VT3 \dots$ знеструмлені. Ці транзистори перейдуть у режим відсікання і відімкнуть виходи $\bar{y}_1, \bar{y}_2, \bar{y}_3$ від загальної шини. На виходах установляться логічні "1".

Переваги інжекційної логіки наступні:

- живляться не колектори, а бази, через що напруга живлення досить низька і не перевищує 1...1,5 В;

- висока економічність через відсутність у логічних елементах резисторів, на яких завжди розсіюється потужність;

- відкриті колектори зумовлюють насичення транзисторів навіть у режимі мікрострумів, що теж зберігає енергію;

- багатоколекторні транзистори розширюють функціональні можливості і значно полегшують побудову складних логічних схем;

- інтегральна інжекційна логіка вдало поєднує мініатюризацію біполярної структури з досить незначним споживанням потужності на одиницю площі кристалу і дозволяє розмістити на площі, яку займає один елемент ТТЛ, близько десяти аналогічних елементів $I^2Л$.

Поруч з цими значними перевагами $I^2Л$ -елементи мають низьку завадостійкість до запираючих завад (не більше 0,1 В). Тому $I^2Л$ -елементи використовують частіше у внутрішніх вузлах ВІС, НВІС і СВІС.

3.6. Логічні елементи на польових структурах

Головним конкурентом мікросхем ТТЛ за широтністю використання є ЛЕ на *польових* МОН- і КМОН-структурах. Це зумовлено дуже незначним споживанням вхідного струму, великим вхідним опором, повним гальванічним розв'язанням вхідного та вихідного кіл, малим споживанням потужності, досить високою густиною розміщення елементів. Мікросхеми МОН- або

КМОН-структур мають широкий діапазон напруг живлення (від 3 до 15 В), що дозволяє виконувати просте поєднання з мікросхемами ТТЛ та операційними підсилювачами.

Польові транзистори забезпечили на початку 70-х років створення ВІС та СВІС. Велике практичне застосування при розробці логічних схем на МДН- та КМОН-структурах набули польові транзистори (ПТ). Польові транзистори розподіляються за способом формування каналу і видом носіїв. Наприклад, базовий ЛЕ на МОН-транзисторах з індукованим каналом *n*- або *p*-типів носять назву відповідно *n*МОН- і *p*МОН-структури.

*n*МОН-структури мають низку переваг порівняно з *p*МОН. Вони мають більшу швидкодію, бо носіями струму в *n*-структурах є електрони, що мають більшу рухомість ніж дірки, які відіграють таку ж роль в *p*-структурах. Зауважимо, що ПТ *p*-типу відкриваються *негативною* напругою на заслоні відносно витоку, а *n*МОН-транзистори – *позитивною*. Тому для реалізації ЛЕ позитивної логіки зручніше використовувати *n*-канальні транзистори, а для негативної логіки – *p*-канальні.

З цих причин логічні елементи *n*МОН зустрічаються частіше, ніж *p*МОН-структур.

3.6.1. МОН-логіка

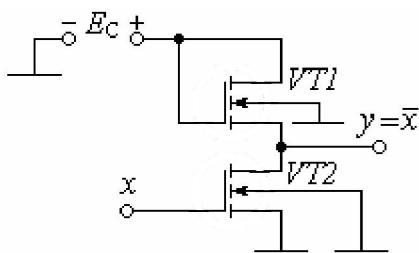


Рисунок 3.12 – Інвертор на *n*МОН-транзисторах з індукованими каналами

МОН-логіка вигідно відрізняється від ТТЛ зменшеним споживанням енергії.

Інвертор на *n*МОН-структурі показаний на рис. 3.12.

Транзистор *VT1* відіграє роль опора навантаження, тобто він має бути певною мірою відкритим. Тому затвор транзистора *VT1* з'єднується з колом живлення $+E_C$.

Коли на вхід схеми поданий низький рівень напруги, тобто $x = 0$, то транзистор *VT2* закритий, через що вихід *y* відірваний від нуля і на виході інвертора установлюється високий рівень $y = 1$. Якщо $x = 1$, то транзистор *VT2* відкритий, через що вихід *y* підключений до нуля і тому на виході $y = 0$.

Так інвертор здійснює реалізацію функції НЕ.

Для створення *схеми І* використовують транзистори *VT2* та *VT3* (рис. 3.13). Кількість таких транзисторів збігається з числом *i* входів x_i .

Як і в інверторі, транзистор *VT1* є опором навантаження.

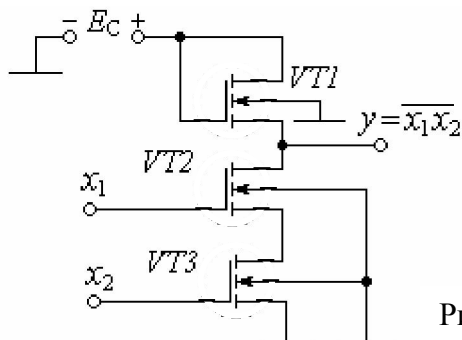


Рисунок 3.13 – Схема І-НЕ

Таблиця 3.2

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

Роботу схеми віддзеркалює таблиця істинності (табл. 3.2).

Якщо хоч би один із входів x_1 або x_2 нульовий, то або $VT2$, або $VT3$, або обидва транзистори закриті і відривають вихід y від нуля. На виході $y = 1$. Лише, коли обидва входи одиничні $x_1 = 1$ та $x_2 = 1$, то обидва транзистори $VT2$ та $VT3$ відкриті і підключають вихід y до нуля, тобто $y = 0$. Так здійснюється реалізація функції І-НЕ.

Для створення схеми АБО слід $VT2$ та $VT3$ з'єднати паралельно відповідно по стоках та витоках (рис. 3.14).

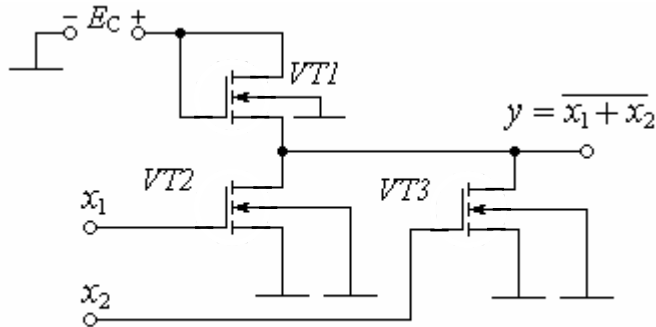


Рисунок 3.14 – Схема АБО-НЕ

Таблиця 3.3

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	0

Роботу схеми віддзеркалює таблиця істинності (табл. 3.3).

Якщо обидва входи одиничні $x_1 = 1$ та $x_2 = 1$, то обидва транзистори $VT2$ та $VT3$ відкриті і підключають вихід y до нуля, тобто $y = 0$.

Коли хоча б один із входів або x_1 , або x_2 , або обидва одиничні, то або $VT2$, або $VT3$, або обидва транзистори відкриті і підключають вихід y до нуля. На виході $y = 0$. Так здійснюється реалізація функції АБО-НЕ.

Основною перевагою МОН-логіки за ТТЛ є зменшене споживання потужності, бо при $x = 0$ транзистори $VT2$ та $VT3$ закриті і схема не використовує струму.

Недоліком МОН-логіки є певне споживання потужності, бо при одиничних входах транзистори $VT2$ та $VT3$ відкриті і схема споживає струм.

Цей недолік усунений в КМОН-логіці.

3.6.2. КМОН-логіка

Літера “К” розшифровується як “комплементарний”.

Комплементарною називається пара послідовно з'єднаних транзисторів з протилежними типами провідностей: p -каналом та n -каналом.

Перевагами КМОН-логіки є практична відсутність споживання енергії і простота виготовлення.

Щодо відсутності споживання енергії, то вона забезпечується тим, що в комплементарній парі за будь-якого стану її входу один із двох транзисторів обов'язково закритий, через що комплементарна пара не споживає струму.

Простота виготовлення зумовлена тим, що мікросхеми КМОН-логіки вільні від резисторів, діодів, складних багатоємітерних транзисторів і містять лише МОН-транзистори з індукованим каналом. Ці транзистори мають досить високу порогову напругу затвору (до 4 В), що забезпечує певну завадостійкість.

Підвищення завадостійкості пояснюється існуванням порогової напруги заслону, яка досягає кількох вольт. Тому зона невизначеності КМОН-логіки значно більше за схеми ТТЛ. Так при напрузі живлення + 10 В зона невизначеності становить 2 В, тобто втричі більше за схеми ТТЛ.

Крім того, мікросхеми КМОН-серій живляться підвищеною напругою (до + 15 В). Це дає можливість рознести пороги перемикання також на декілька вольт. Результатом є те, що при напрузі живлення + 10 В рівень логічного «0» становить $U_x^0 = 0,5$ В, а рівень логічної «1» $U_x^1 = 9,5$ В. Це означає, що логічний елемент КМОН-логіки не чутливий до амплітуди завади, яка не перевищує $U_{mз} < U_x^1 - U_x^0 = 9,5 - 0,5 = 9,0$ В, тобто завадостійкість КМОН-логіки майже у шість разів вище за ТТЛ.

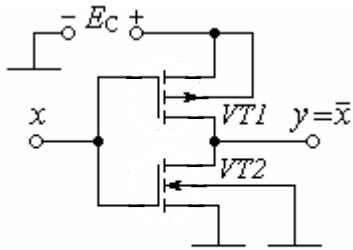


Рисунок 3.15 – КМОН-інвертор

Найпростішим елементом КМОН-логіки є інвертор. Він складається лише з однієї комплементарної пари.

Схема інвертора показана на рис. 3.15.

До складу інвертора надходять два МОН-транзистори з різними провідностями: $VT1$ з p -каналом та $VT2$ з n -каналом. Ці транзистори і створюють комплементарну пару, в якій входом x є з'єднання заслонів, а виходом y – з'єднання стоків.

Транзистори відкриваються, якщо між заслоном та прошарком прикладена одинична напруга, та закриваються, якщо ця напруга нульова.

Інвертор працює наступним чином.

Якщо на вході діє рівень логічного «0» ($x = 0$), то транзистор $VT1$ буде відкритим, бо напруга «затвор-прошарок» цього транзистора одинична (затвор нульовий, прошарок одиничний).

Щодо транзистора $VT2$, то він закритий, бо напруга «затвор-прошарок» для нього дорівнює нулю.

Закритий транзистор $VT2$ відключає вихід y від нульової шини, тобто від логічного «0», а відкритий $VT1$ підключає вихід y до $+E_c$, тобто до одиниці ($y = 1$).

Якщо на вході логічна «1» ($x = 1$), то транзистор $VT1$ закриється, бо напруга «затвор-прошарок» для нього нульова, а транзистор $VT2$ відкриється через те, що напруга «затвор-прошарок» для нього одинична.

Закритий транзистор $VT1$ відключає вихід y від напруги $+E_c$, тобто від логічної «1», а відкритий транзистор $VT2$ підключає вихід y до нульової шини, тобто до логічного «0» ($y = 0$).

Так здійснюється функція НЕ.

В обох випадках і при $x = 0$, і при $x = 1$ один із транзисторів або $VT1$, або $VT2$ обов'язково закритий. Через це логічні елементи КМОН-логіки практично не споживають струму.

Таким чином, можна дійти висновку, що у статистичному режимі схеми на КМОН-структурах практично не споживають потужності. Практично тому, що в момент, коли рівень вихідної напруги змінюється від 0 до 1, має місце коротка тривалість часу, коли обидва транзистори $VT1$ та $VT2$ відкриті і від кола

живлення $+ E_C$ споживається помітний струм. Однак це явище впливає на споживану потужність тільки на занадто високих частотах.

Розглянемо мікросхеми для реалізації останніх двох функцій бульового базису.

На рис. 3.16 показана схема І-НЕ на КМОН-структурах, тобто на комплементарних парах МОН-транзисторів.

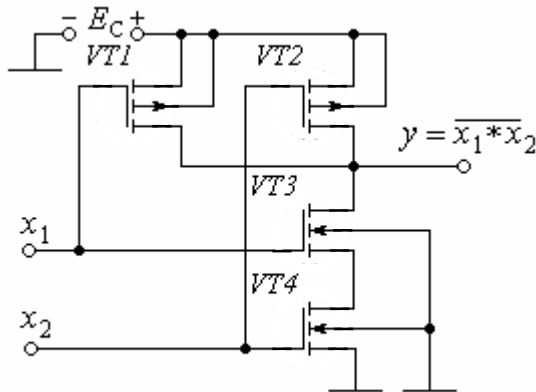


Рисунок 3.16 – Схема 2І-НЕ

Таблиця 3.4

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

Комплементарні пари створюють ті транзистори, затвори яких з'єднані. В цих парах за будь-якого стану затвору один із транзисторів обов'язково закритий, бо транзистори комплементарної пари мають протилежні провідності. Тому комплементарні пари не споживають струму.

Тут комплементарними парами є $VT1$ та $VT3$ і $VT2$ та $VT4$.

Роботу схеми віддзеркалює таблиця істинності (табл. 3.4).

Якщо хоча б один із входів x_1 або x_2 нульовий, то або $VT3$, або $VT4$, або обидва транзистори закриті, бо напруга “затвор-прошарок” цих транзисторів нульова. Закриті $VT3$, $VT4$ відривають вихід y від нуля.

Проте напруга “затвор-прошарок” транзисторів $VT1$ та $VT2$ одинична, через що вони відкриті і підключають вихід y до $+ E_C$, тобто одиниці. На виході $y = 1$.

Коли ж обидва входи одиничні і $x_1 = 1$, і $x_2 = 1$, то напруга “затвор-прошарок” транзисторів $VT1$ та $VT2$ нульова, через що вони закриті і відривають вихід y від $+ E_C$, тобто від одиниці.

Проте напруга “затвор-прошарок” транзисторів $VT3$ та $VT4$ одинична, через що вони відкриті і підключають вихід y до нуля, тобто $y = 0$. Так здійснюється реалізація функції 2І-НЕ.

На рис. 3.17 показана схема 2АБО-НЕ.

Тут комплементарними парами є $VT1$ та $VT2$ і $VT3$ та $VT4$.

Роботу схеми віддзеркалює таблиця істинності (табл. 3.5).

При обох нульових входах ($x_1 = 0$; $x_2 = 0$) напруга “затвор-прошарок” транзисторів $VT1$ та $VT4$ нульова, через що вони закриті і відривають вихід y від нуля.

Проте напруга “затвор-прошарок” транзисторів $VT2$ та $VT3$ одинична, через що вони відкриті і підключають вихід y до $+ E_C$, тобто $y = 1$.

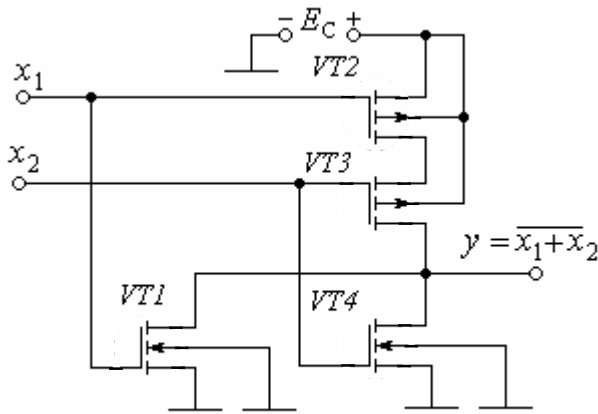


Рисунок 3.17 – Схема 2АБО-НЕ

Таблиця 3.5

x_1	x_2	y
0	0	1
1	0	0
0	1	0
1	1	0

Якщо хоча б один із входів x_1 або x_2 одиничний, то або $VT1$, або $VT4$, або обидва транзистори відкриті, бо напруга “затвор-прошарок” цих транзисторів одинична. Відкриті $VT1$, $VT4$ підключають вихід y до нуля, тобто $y = 0$.

Так здійснюється функція 2АБО-НЕ.

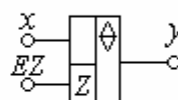
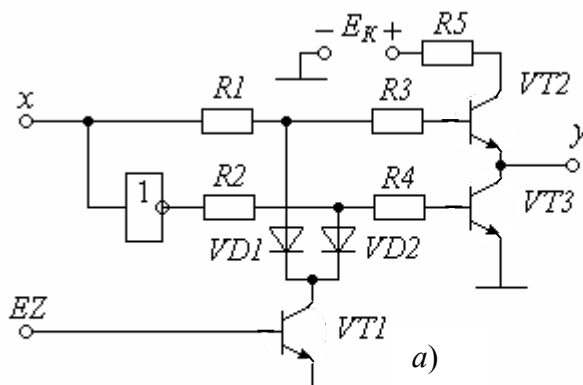
Кожна серія мікросхем має значну кількість комбінацій логічних схем, що робить кожну серію гнучкою при проектуванні цифрових приладів.

3.7. Логічний елемент з трьома станами

Логічний елемент з трьома станами або тристановий драйвер, або тристановий буфер має на виході, крім звичайних двох станів “0” і “1”, третій, який носить назву Z -стану. Третій стан – це стан великого вихідного опору, який становить сотні кілоом, тобто практично є діелектриком.

Потреба створення елементів з трьома вихідними станами виникла при організації процесорних систем із загальними шинами обміну даними між пристроями, які входять до складу системи. По одній і тій самій шині можна як приймати, так і передавати дані.

Одна зі схем тристанового драйвера (далі: драйвер), тобто елемента з трьома станами показана на рис. 3.18,а, а її умовне позначення – на рис. 3.18,б. Третій стан визначається входом EZ , який називається керуючим.



Таблиця 3.6 – Стани драйвера

x	EZ	y
X	1	Z
1	0	1
0	0	0

Рисунок 3.18 – Елемент з трьома вихідними станами (драйвер):
а) схема; б) умовне позначення

Роботу схеми віддзеркалює таблиця істинності (табл. 3.6), у якій хрестиком X позначена незалежність вихідного рівня у від стану входу x.

Ключ на $VT1$ забезпечує третій стан, а перемикач на $VT2, VT3$ здійснює переключення виходу у до логічних або нуля, або одиниці.

Драйвер працює наступним чином.

Якщо $EZ = 0$, то транзистор $VT1$ закритий і не впливає на роботу схеми, яка є наступною.

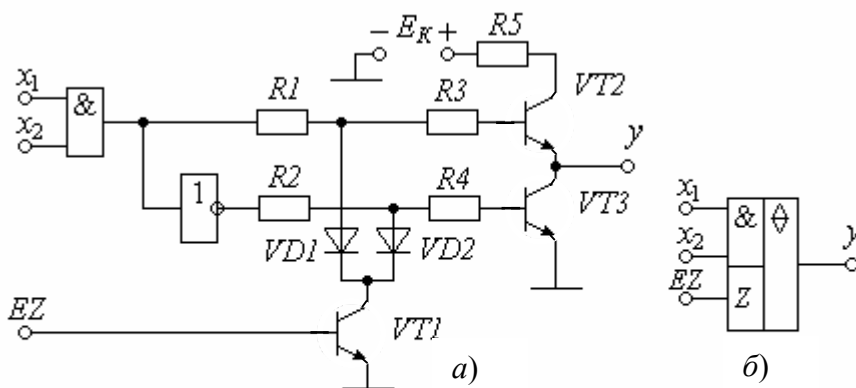
При $x = 0$ транзистор $VT2$ закритий і відриває вихід у від $+E_K$, тобто від логічної одиниці. Проте через наявність інвертора 1 транзистор $VT3$ відкритий і підключає вихід у до землі, тобто до логічного "0" ($y = 0$).

При $x = 1$ стан транзисторів $VT2, VT3$ змінюється на протилежний, через що $y = 1$.

Коли ж $EZ = 1$, то транзистор $VT1$ відкривається і через діоди $VD1, VD2$ закорочує кола баз $VT2, VT3$ на нуль. Обидва транзистори $VT2, VT3$ закриваються, і на виході у встановлюється високий опір.

Якщо виходу будь-якого логічного елемента треба надати третій стан, то до входу цього елемента треба підключити вхід x драйвера, як це показано на рис. 3.19. Роботу схеми пояснює таблиця істинності (табл. 3.7).

Бувають випадки, коли активним сигналом, тобто таким, який змінює стан логічного елемента на Z, приймається логічний "0". Це характерно для мікропроцесорної техніки, якщо керуючі сигнали вибираються за законами негативної логіки.



Таблиця 3.7 – Стани елемента I з драйвером

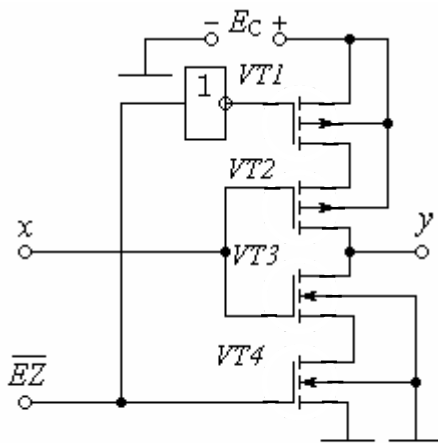
x_1	x_2	EZ	y
X	X	1	Z
0	0	0	0
1	0	0	0
0	1	0	0
1	1	0	1

Рисунок 3.19 – Логічний елемент I з трьома вихідними станами:
а) схема; б) умовне позначення

На рис. 3.20 показана схема тристанового драйвера на КМОН-структурах з інверсним керуючим входом \overline{EZ} .

Схема є КМОН-інвертором на транзисторах $VT2, VT3$, який доповнений n- і p-МОН ключами $VT1$ та $VT4$ відповідно. Ці ключі забезпечують Z-стан наступним чином.

Роботу схеми віддзеркалює таблиця істинності (табл. 3.8), у якій хрестиком X позначена незалежність вихідного рівня у від стану входу x.



Таблиця 3.8 –
Стани драйвера

x	\overline{EZ}	y
X	0	Z
1	1	1
0	1	0

Рисунок 3.20 – Драйвер на КМОН-структурах

У стан Z схема переходить тоді, коли $\overline{EZ} = 0$, через що обидва ключі $VT1$ і $VT4$ закриті. При цьому закритий ключ $VT1$ обриває коло живлення, тобто логічну “1”, а закритий $VT4$ обриває коло логічного “0”. Так вихід інвертора y відключений і від нульового, і від одиничного рівнів, тобто має високий вихідний опір.

Якщо на вхід \overline{EZ} подано логічну “1”, то обидва ключі $VT4$ і $VT1$ відкриті, і схема виконує функцію інвертора.

Хоча тристанові драйвери на КМОН-структурах мають швидкодію меншу за TTL драйверів, у стані Z вони забезпечують практично ідеальну розв’язку, бо вихідний опір більше 10 МОм.

Елементи з трьома станами виходу розроблені спеціально для використання в якості вихідного керованого буфера для підключення цифрових блоків до магістральних.

Буфери з трьома станами виходу носять назву *шинних драйверів*.

3.8. Узгодження логічних мікросхем

Узгодження логічних мікросхем передбачає забезпечення спільної роботи мікросхем різноманітних серій, наприклад, TTL і КМОН.

Мікросхеми розглянутих серій відрізняються багатьма ознаками. При проектуванні цифрових пристроїв досить часто треба розв’язувати задачі, які зв’язані з сумісним застосуванням різноманітних серій мікросхем, що викликає значні труднощі.

Узгодження різних типів мікросхем пов’язане в першу чергу з необхідністю узгодження вхідних та вихідних рівнів напруги, тобто рівнів логічного нуля та логічної одиниці за різних напруг живлення; вхідних та вихідних струмів; забезпечення заданих рівнів завадостійкості. Для розробки питань узгодження треба враховувати основні параметри тієї чи іншої серії.

Типові електричні параметри базових ЛЕ надані в табл. 3.9.

Таблиця 3.9 – Параметри логічних елементів

Тип ЛЕ	$U_{\text{вх}}^1$	$U_{\text{вх}}^0$	$I_{\text{вих}}^1$	$I_{\text{вих}}^0$	$I_{\text{вх}}^1$	$I_{\text{вх}}^0$	$E_{\text{ж}}, \text{В}$
ТТЛ	2,5...4,0	$\leq 0,4$	0,5...1,0	≤ 20	$\leq 0,15$	≤ 2	+5
І ² Л	0,7...0,9	$\leq 0,1$	0	$\leq 0,02$	0	$\leq 0,05$	+1,5
ЕЗЛ	0,75...0,9	1,8	5...20	$\leq 0,2$	$\leq 0,2$	0	-5,2
p-МОН	-1	≥ -8	2	≤ 3	10^{-3}	10^{-3}	-12
n-МОН	≥ 8	≤ 1	3	≤ 5	10^{-3}	10^{-3}	+12
КМОН	$\geq 0,95 E_{\text{ж}}$	$\leq 0,5$	2	≤ 3	10^{-3}	10^{-3}	5...15

Таблиця містить наступні основні параметри :

$U_{\text{вх}}^1$ – вхідна напруга логічної “1”;

$U_{\text{вх}}^0$ – вхідна напруга логічного “0”;

$I_{\text{вих}}^1$ – вихідний струм логічної “1”, який може бути відданий у навантаження;

$I_{\text{вих}}^0$ – вихідний струм логічного “0”, який можна відібрати від навантаження;

$I_{\text{вх}}^1$ – вхідний струм логічної “1”;

$I_{\text{вх}}^0$ – вхідний струм логічного “0”;

$E_{\text{ж}}$ – напруга живлення.

Пристрої, за допомогою яких виконується узгодження різноманітних серій, називають *перетворювачами* або *трансляторами рівнів*. Особливою відзнакою перетворювачів є те, що рівні їхніх вхідних та вихідних сигналів завжди різні. На практиці особливий інтерес викликають перетворювачі рівнів найбільш часто використовуваних мікросхем, наприклад, ТТЛ та КМОН.

Для логічних елементів КМОН вхідний струм практично відсутній при зміні $U_{\text{вх}}$ від 0 до 15 В (струм зворотно зміщеного витоку $< 10^{-5}$ мкА), а вихідний струм відкритих транзисторів обмежується на рівні декількох міліампер.

Щодо логічних елементів ТТЛ, то їхній вихідний струм може досягати 16 мА. Тому елементи ТТЛ і КМОН можна з'єднувати тільки через схеми узгодження, що враховують зазначену різницю струмів.

При узгодженні ТТЛ-КМОН або КМОН-ТТЛ можуть зустрічатися наступні варіанти:

- узгодження ТТЛ-КМОН за однакової напруги живлення (+ 5 В);
- узгодження ТТЛ-КМОН, якщо напруга живлення КМОН серії значно більша за мікросхеми серії ТТЛ;
- узгодження КМОН-ТТЛ за однакової напруги живлення (+ 5 В);
- узгодження КМОН-ТТЛ, якщо напруга живлення КМОН серії значно більша за мікросхеми серії ТТЛ.

У разі *єдиного кола живлення* + 5 В для мікросхем серій ТТЛ та КМОН узгодження виконується без додаткових елементів. До виходу логічного елемента ТТЛ можна підключити вхід елемента КМОН.

Оскільки на вході КМОН струм надто малий, то узгодження забезпечується автоматично. Для підвищення завадостійкості на виході ЛЕ ТТЛ між його виходом і джерелом напруги $+E_{ж}$ слід включити резистор з опором до 5 кОм в залежності від типу мікросхеми ТТЛ.

Коли напруга живлення КМОН серії *значно більша*, ніж мікросхем ТТЛ, наприклад, $E_{КМОН} = 10 \text{ В}$, а $E_{ТТЛ} = 5 \text{ В}$, то узгодження ТТЛ-КМОН треба виконувати за схемою, яка показана на рис. 3.21.

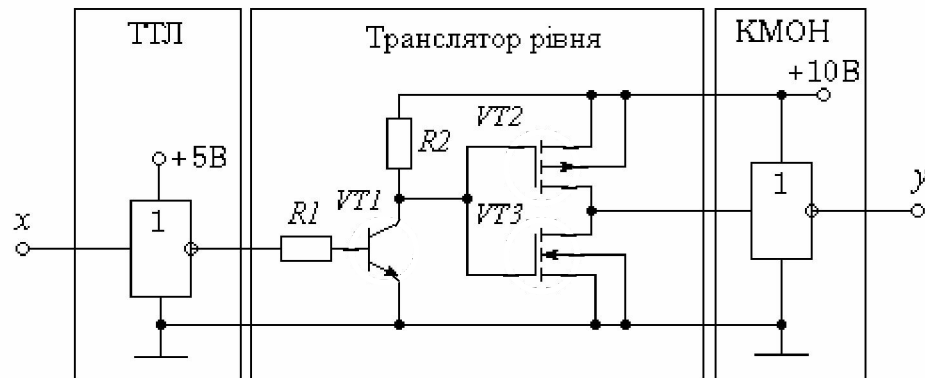


Рисунок 3.21 – Узгодження мікросхем ТТЛ-КМОН за різних напруг живлення

Замість транзистора $VT1$ у схемі транслятора можна використовувати логічний елемент з відкритим колектором. Опір резистора $R2$ в цьому випадку не має принципового значення і вибирається в межах 10 ... 100 кОм.

Узгодження мікросхем КМОН-ТТЛ, якщо напруга живлення КМОН-серії *значно більша*, ніж мікросхем ТТЛ, виконується, як показано на схемі (рис. 3.22).

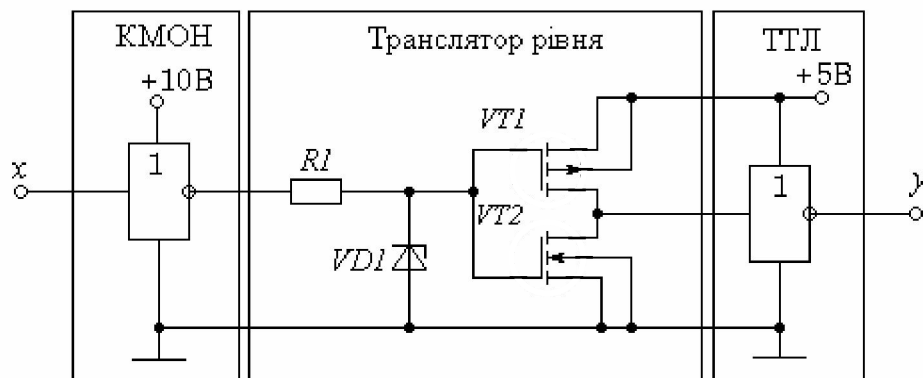


Рисунок 3.22 – Узгодження мікросхем КМОН-ТТЛ за різних напруг живлення

У цьому випадку треба виконувати узгодження елементів різноманітних серій як за струмом, так і за напругою.

Транслятор рівня містить обмежувач напруги на резисторі $R1$ та стабілітроні $VD1$ з напругою стабілізації 5,1 В. Цей обмежувач має знизити напругу до 5,1 В, щоб транзистор $VT1$ надійно заклався.

Щодо інших мікросхем, то якщо вони мають однакові рівні відповідно вхідних та вихідних напруг, через що узгодження не потрібне.

3.9. Правила схемного включення елементів

Обмеження за навантажувальною здатністю елементів визначає число входів елементів аналогічної серії, яке можна підключити до виходу даного елемента. При підвищенні навантаження зверх певного вихідні параметри не гарантуються.

Невикористані входи схем мають бути підключеними. В ТТЛ- і ТТЛШ-серіях сигнали від непідключених входів сприймаються як логічні “1”. Якщо залишити входи непідключеними, то виникаючі при цьому додаткові заряди бази затримують на деякий час переключення логічного елемента по інших працюючих входах.

В ТТЛ і ТТЛШ невикористані x -входи або поєднують з іншими використаними входами (рис. 3.23,а), або підключають до джерела логічної “1”.

Таким джерелом може бути вихід логічного елемента, наприклад, НЕ, вхід якого підключений до нульового потенціалу (рис. 3.23,б), або резистор R з опором 1 кОм, підключений до джерела напруги + 5 В (рис. 3.23,в). До такого резистора можна підключати до 20 невикористаних входів.

У КМОН-елементах непідключеним не можна залишати жодного входу, бо при цьому відповідний транзистор залишиться без напруги “затвор-прошарок” і схема стане непрацездатною.

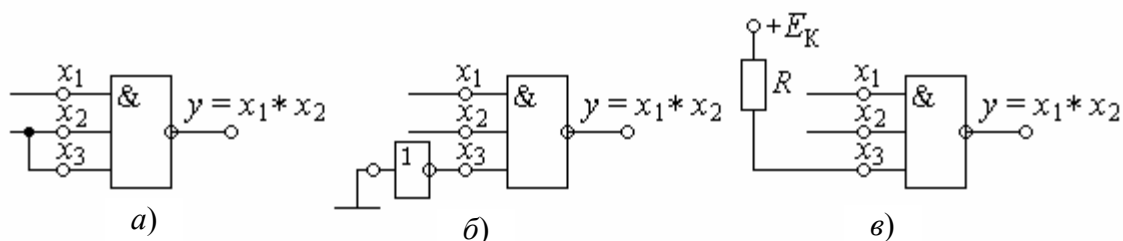


Рисунок 3.23 – Підключення схеми ТТЛ або ТТЛШ з використанням входом x_3 :
а) з поєднаними входами; б) з підключенням до виходу логічного елемента;
в) з підключенням до джерела напруги через резистор

У серіях КМОН невикористані входи можна підключати до джерела напруги безпосередньо без резисторів. Як і в ТТЛ-елементах, невикористані входи можна поєднувати з робочими.

Невикористані входи схем АБО в будь-яких серіях треба підключати до логічного “0”.

Якщо деякі елементи, що входять до складу корпусу мікросхеми, не використовуються, то на їхні входи треба подавати такий потенціал, за якого на виході устанавлюється логічна “1”. При такому стані елемент споживає меншу потужність і його можна використовувати як джерело логічної “1”.

До найважливіших проблем цифрової схемотехніки є питання захисту елементів від *завад* та *шумів*. Значну роль у цій проблемі відіграють конструкція пристрою і монтаж окремої мікросхеми. Причин появи внутрішніх і зовнішніх завад та шумів є досить багато. Основними джерелами їхнього виникнення є кола живлення і заземлення, вхідні й вихідні кола сигналу.

Внутрішній власний шум (крім теплового) можна зменшити раціональним розміщенням елементів на платі, фільтрацією та екрануванням.

Ситуація ускладнюється, коли сигнали треба приймати або передавати за межами плати. Досить складна вона і тоді, коли сама плата має велику площу. В цих випадках треба секціонувати шини “земля” так, щоб вузли великої потужності мали б свою землю, а малої потужності – свою. Ці шини мають об’єднуватись в одній і саме тій точці, яка заземляється. Крім того необхідно зменшувати паразитну індуктивність земельних шин збільшенням їхньої площі та їх скороченням.

Завади зростають і при передаванні цифрових сигналів на відстань. Вони стають більш інтенсивними внаслідок збільшення погонної реактивності провідників. Інтенсивність завад визначається також властивостями лінії зв’язку.

Для узгодження лінії значної довжини рекомендовано як передавачі на передавальному кінці (рис. 3.24) використовувати логічні елементи з відкритим колектором (стоком), а приймачем може бути ЛЕ на тригері Шмітта (про тригер Шмітта буде йти мова далі).

При цьому резистор навантаження $R1$ слід включати на приймальному кінці. При занадто високих рівнях завад (> 2 В) треба включати фільтруючі кола, наприклад, $R2, C, R3$.

Вищу завадостійкість можна отримати при застосуванні симетричної лінії. Такою лінією може бути скручена пара однотипних проводів. Для симетричної лінії передавач і приймач повинні мати диференційні вихід і вхід відповідно.

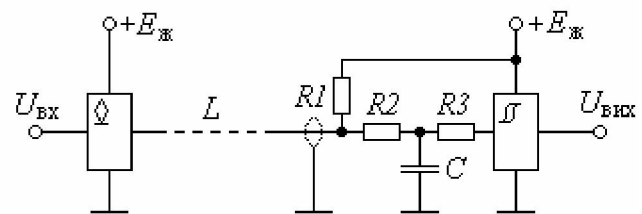


Рисунок 3.24 – Захист від завад при передаванні сигналу на відстань

Захистити цифрові схеми від завад та шумів можна за допомогою *синхронізуючих сигналів*. Синхронізація допомагає уникнути небажаних процесів переключення від завад x_3 (рис. 3.25,а).

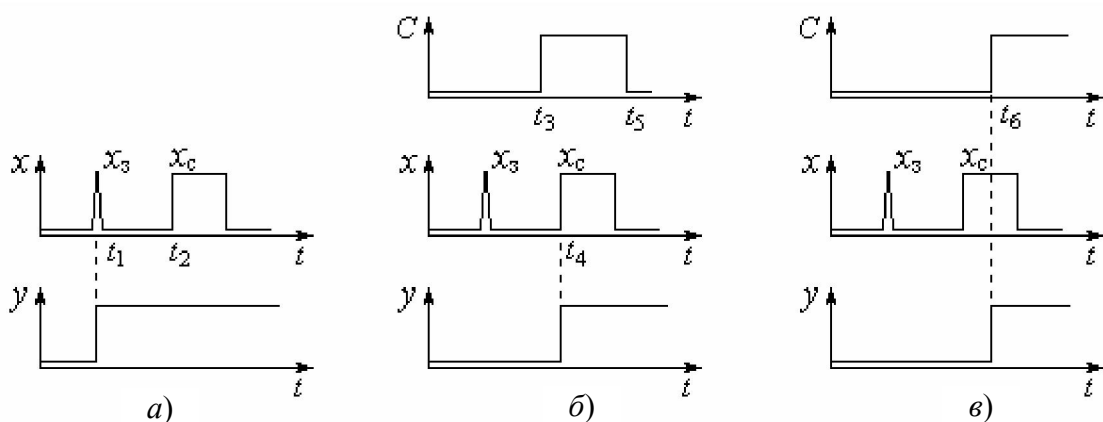


Рисунок 3.25 – Прийом цифрових сигналів: а) без синхронізації; б) з синхронізацією рівнем; в) з синхронізацією перепадом

Алгоритмом прийому сигналів без синхронізації є

$$y = x, \quad (3.10)$$

тобто, що на вході ЛЕ, то й на його виході.

Незважаючи на те, що сигнал x_c (рис. 3.25,а) з'являється лише в момент t_2 , ЛЕ спрацює в момент t_1 від завади x_3 . Таким чином, система без синхронізації ніяк не захищена від завад.

Цей суттєвий недолік значною мірою усунений у системі з синхронізацією.

Існують два способи синхронізації: рівнем та перепадом.

При синхронізації *рівнем* спрацьовування ЛЕ можливе протягом усього часу ($t_3 \dots t_5$) дії дозволяючого рівня синхросигналу C (рис. 3.25,б).

Алгоритмом прийому сигналів із синхронізацією рівнем є функція логічного множення рівня синхронізації C та входу x :

$$y = C \wedge x = C \cdot x = 1, \quad (3.11)$$

тобто ЛЕ може спрацювати тільки під час ($t_3 \dots t_5$) дії дозволяючого рівня синхросигналу C . Тому, якщо завада x_3 знаходиться поза межами цього часу, то вона не сприймається пристроєм. ЛЕ спрацює тільки тоді, коли з'явиться сигнал x_c у момент t_4 . Оскільки час ($t_3 \dots t_5$) дозволу прийому обмежений, то спрацьовування від завад зменшене.

Так, синхронізація рівнем підвищує завадостійкість.

Ще вищу завадостійкість забезпечує синхронізація *перепадом* (рис. 3.25,в).

Алгоритмом прийому сигналів із синхронізацією перепадом є функція логічного множення *похідної* рівня синхронізації C та входу x :

$$y = \frac{dC}{dt} \wedge x = \frac{dC}{dt} \cdot x = 1, \quad (3.12)$$

тобто ЛЕ може спрацювати тільки в момент t_6 дії фронту C . За межами цього фронту ЛЕ нечутливий до будь-яких завад.

Оскільки час дозволу спрацьовування скорочений до одного моменту, то завадостійкість висока.

При використанні обох методів синхронізації треба на сигнальних входах x_c ставити логічні елементи з порогом спрацьовування, наприклад, тригери Шмітта.

Виходи мікросхем, які передають сигнали *за межі плати* або *цифрової системи* треба буферувати за допомогою спеціальних мікросхем-буферів.

Це допомагає уникнути явища інтерференції завади і корисного сигналу, а також придушує викиди струму на неузгодженій лінії.

Зменшити вплив завад та шумів значною мірою допомагає раціональне розміщення шин напруг живлення та земель приладу. Ці питання досить складні і їм присвячено багато книг та статей. Але ж, для кожного випадку, для кожного приладу завадостійкість залежить від знання та інженерної ерудиції розробника схем.

Контрольні питання

- 3.1. Що таке цифровий елемент?
- 3.2. Які цифрові мікросхеми називають потенційними, а які імпульсними?
- 3.3. Яка логіка називається позитивною, а яка негативною?
- 3.4. Що таке логічний елемент?
- 3.5. Що таке багатомітерний транзистор і де він використовується?
- 3.6. Які переваги та недоліки мікросхем ТТЛ?
- 3.7. Що таке логічний елемент з відкритим колектором? В якому разі він використовується?
- 3.8. Що таке провідна логіка?
- 3.9. Які переваги набувають мікросхеми, в яких використовується напівпровідникові прилади з заслоном Шотткі?
- 3.10. Що таке МОН- або МДН-логіка?
- 3.11. Що таке *n*-МОН або *p*-МОН структури? Яка з них набула більшого поширення?
- 3.12. Що таке КМОН-інвертор?
- 3.13. Що таке цифровий елемент?
- 3.14. Які цифрові мікросхеми називають потенційними, а які імпульсними?
- 3.15. Яка логіка називається позитивною, а яка негативною?
- 3.16. Що таке багатомітерний транзистор і де він використовується?
- 3.17. Які переваги та недоліки мікросхем ТТЛ?
- 3.18. Що таке логічний елемент з відкритим колектором? В якому разі він використовується?
- 3.19. Що таке провідна логіка?
- 3.20. Які переваги набувають мікросхеми, в яких використовується напівпровідникові прилади з бар'єром Шотткі?
- 3.21. Що таке МОН- або МДН-логіка?
- 3.22. Що таке *n*-МОН або *p*-МОН структури? Яка з них набула більшого поширення?
- 3.23. Що таке КМОН-інвертор?
- 3.24. Мікросхеми якої логіки мають більшу завадостійкість?
- 3.25. Що таке тристановий драйвер?
- 3.26. Чому треба виконувати узгодження логічних мікросхем різних серій?
- 3.27. Що таке перетворювач або транслятор рівня?
- 3.28. Які правила включення логічних схем І, АБО за наявності невикористаних входів?
- 3.29. Що таке синхронізація рівнем та перепадом?

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко. – Одеса: ОНАЗ ім. О.С. Попова, 2004. – Ч. 2. – С. 199 – 227.
2. Титце У. Полупроводниковая схемотехника: Справочное руководство / У. Титце, К. Шенк. – М.: Мир, 1982. – С. 106 – 115.
3. Хоровиц П. Искусство схемотехники / П. Хоровиц, У. Хилл. – М.: Мир, 1983. – Т. 1. – С. 508 – 526.
4. Рицар Б.Є. Цифрова техніка / Рицар Б.Є. – К.: НМК ВО, 1990. – 371 с.

4. ПОСЛІДОВНІСНІ ПРИСТРОЇ

Послідовнісні пристрої – це цифрові автомати з пам'яттю. Вони характеризуються деяким числом внутрішніх станів. Під дією вхідних сигналів послідовнісні пристрої переходять з одного стану в інший. Новий стан залежить від комбінації діючих сигналів на його входах та попереднього стану, в якому знаходився автомат. Тому послідовнісні пристрої складаються з комбінаційної частини й елементів пам'яті – запам'ятовувачів інформації.

Функцію найпростішого запам'ятовувача, що може запам'ятовувати один біт інформації (0 або 1), виконує тригер.

4.1. Тригери

Тригери – це елементи з двома стійкими станами. Тригери – це найбільш поширені функціональні елементи цифрових систем. Тригери застосовуються в лічильниках імпульсів, регістрах, запам'ятовувачах, розподільвачах сигналів, накопичувальних суматорах тощо. Тригери мають самостійне застосування, наприклад, у приладах керування, виконуючи важливі функції логічного перетворення та схову інформації.

Тригерами називають логічну схему з додатним зворотним зв'язком, яка має два стійких стани, і під дією зовнішніх сигналів переключається в будь-який стан та знаходиться в цьому стані необмежений час після припинення дії вхідних сигналів.

Тригер, що може запам'ятовувати один біт двійкової інформації (логічний нуль “0” або логічна одиниця “1”), носить назву двостанового запам'ятовувача. Як правило, двостанові тригери мають два виходи, за рівнем яких можна судити про стан тригера. Перший з виходів має назву *прямого* і позначається Q , другий – *інверсний* \bar{Q} .

Перший із станів тригера приймається за нульовий і носить назву *скинутого*. При цьому його вихід Q знаходиться під нульовим потенціалом, тобто $Q = 0$, а $\bar{Q} = 1$.

Другий стан тригера – одиничний, коли $Q = 1$, а $\bar{Q} = 0$ і його називають *встановленим*.

Переключення тригера в одиничний стан носить назву *встановлення*, а в стан 0 – *скидання* або *скиду*.

Узагальнена структурна схема тригера (рис. 4.1) складається з двох частин: елемента пам'яті – власне двостанового тригера T та схеми керування СК, на яку надходять зовнішні сигнали керування роботою схеми.

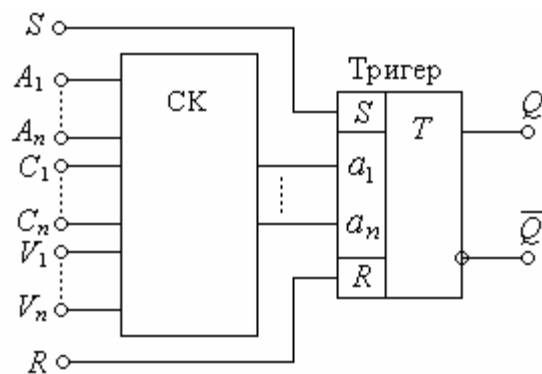


Рисунок 4.1 – Узагальнена структурна схема тригера

Входи A_1, \dots, A_n називають *інформаційними* або *логічними*. До них подають двійкову інформацію у вигляді логічних нуля або одиниці. За допомогою схеми управління вхідна інформація перетворюється в комбінації сигналів 00, 01, 10, 11, що діють безпосередньо на входах a_1, \dots, a_n власно тригера.

Схема тригера, крім логічних, може мати тактові входи *синхронізації* (C_1, \dots, C_n), *керуючі* входи (V_1, \dots, V_n), а також *настановні* S - та R -входи безпосереднього *встановлення* або *скидання* тригера.

Вхід S – вхід встановлення *одиночного* стану або *одиночний вхід*.

Вхід R – вхід встановлення *нульового* стану або *нульовий вхід*.

Тактові або синхронізуючі сигнали, які подаються до входів C_1, \dots, C_n , визначають у СК момент запису чи зчитування стану, тобто переключення тригера.

Класифікація тригерів може бути надана за багатьма ознаками: за способом схемної реалізації, за логікою функціонування, за способом запису інформації.

Згідно з ознакою за способом схемної *реалізації* визначають основні властивості тригерних приладів:

– тип використовуваного запам'ятовувача (статичний, динамічний, статично-динамічний);

– вид керуючого сигналу, з яким може працювати тригер: імпульсний (або сигнал обмеженої тривалості) і потенційний (або сигнал безмежної тривалості; імпульсно-потенційний);

– наявність або відсутність у структурній схемі вузлів, перетворюючих потенційні вхідні сигнали в імпульсні.

У відповідності з цими ознаками тригери поділяються на *статичні*, *квазістатичні*, *імпульсно-статичні*, *динамічні*.

Функціональна ознака, тобто логіка функціонування тригера визначає той чи інший тип тригера за видом характеристичного рівняння, що в загальному випадку записується як

$$Q^{n+1} = f(Q^n, A_1^n, A_2^n, \dots, A_n^n) \quad (4.1)$$

і відбиває стан виходу Q^{n+1} тригера на нинішній момент t_{n+1} в залежності від комбінації сигналів, діючих на входах тригера A_1, \dots, A_n , та його стану Q^n на попередній момент t_n . Назва тригера надається за значенням його інформаційних входів, яким замість символів A_1, \dots, A_n присвоюються символи D, J, K, T та інші.

В інтегральній схемотехніці, здебільшого, зустрічаються двовходові RS - і JK -тригери, універсальні – $D(RS)$, $JK(RS)$ -тригери. Всі типи тригерів, яких досить багато, можуть бути побудовані в базисі різних логічних елементів.

За способом запису інформації розрізняють два типи тригерів: *асинхронні* та *синхронні*.

До асинхронних відносяться тригери, які мають лише інформаційні входи без входу синхронізації. Запис інформації в них здійснюється в будь-який час

безпосередньо з надходженням інформаційного сигналу, який подається на інформаційний вхід тригера.

Відмінною особливістю синхронних тригерів є наявність додаткового тактового (синхронізуючого) входу C для подачі синхронізуючих імпульсів, які здійснюють запис інформації. У той самий час синхронні тригери поділяються на тригери одноктактової та багатотактової дії. В тригерах одноктактової дії спрацьовування відбувається з надходженням кожного синхроімпульсу. В тригерах багатотактової дії запис інформації закінчується наприкінці дії n -го тактового імпульсу.

У тригерах розрізняють входи *статичні* та *динамічні*.

На статичні входи подають потенційні цифрові сигнали для переключення тригера, тобто такі сигнали x , що визначені рівнями $x = 0$ або $x = 1$.

Динамічні входи керуються імпульсними сигналами, часова тривалість яких нескінченно мала, тобто при зміні потенційного сигналу з одиниці на нуль (зрізу), або навпаки, при зміні на вході сигналу з нуля на одиницю (фронт).

4.1.1. Параметри тригерів

Параметри тригера можна поділити на *функціональні* та *схемотехнічні*.

До *функціональних* відносяться параметри, яким повинні задовольняти тригери при їхньому конкретному застосуванні. Це такі параметри:

- функціональний тип тригера;
- спосіб запису інформації в тригер;
- спосіб керування записом інформації в тригер;
- число тактових, інформаційних, дозволяючих та установлюючих входів;
- еквівалент навантаження тригера по тактовому входу;
- навантажувальна здатність тригера по виходу;
- швидкодія тригера;
- функціональна надійність тригера.

До *схемотехнічних* параметрів мають відношення такі параметри:

- апаратні витрати;
- потужність споживання.

Функціональний тип тригера припускає вибір готового або розробку нового тригера, характеристичне рівняння якого повинне задовольняти необхідним вимогам за логікою дії.

Способів запису інформації в тригер два: асинхронний або синхронний. При цьому треба враховувати, в якому режимі тригер може працювати. Наприклад, асинхронний тригер працює у синхронному режимі, або навпаки.

Спосіб керування записом інформації припускає вибір одного з типів тригерів, який необхідно мати за логікою праці в конкретному випадку.

Число тактових, інформаційних та інших входів визначається конкретикою його застосування. В загальному випадку тригер може мати безліч входів або в найпростішому випадку може бути лише один або два інформаційних входи.

Еквівалент навантаження тригера по тактовому входу n_c показує, яке число тригерів з боку тактового входу C може бути підключено до типового елемента без порушення його працездатності.

Для тригерів, які виконані на логічних елементах, параметр n_c визначається числом зв'язків S_c тактового входу C з рештою елементів тригера.

Навантажувальна здатність по виходу n_Q показує, яке число елементів можна підключити до виходу тригера без відхилення вихідної напруги від сталого логічного рівня, і визначається за формулою

$$n_Q = n_e - S_Q, \quad (4.2)$$

де n_e – навантажувальна здатність елемента, на якому виконаний тригер;

S_Q – число зв'язків виходу тригера.

Швидкодія тригера показує максимальну частоту проходження вхідних сигналів f_{\max} , за якої тригер повністю виконує свої функції. В загальному випадку f_{\max} будь-якого тригера визначається формулою

$$f_{\max} = \frac{1}{\tau_{i \min} - \tau_{п \min}} = \frac{1}{t_d}, \quad (4.3)$$

де $\tau_{i \min}$ – мінімальна тривалість сигналу, за якого працює тригер;

$\tau_{п \min}$ – мінімальна тривалість паузи між сигналами запису інформації;

$t_d = \tau_{i \min} - \tau_{п \min}$ – мінімальний часовий інтервал між двома вхідними імпульсами.

Вимога функціональної надійності припускає схемну реалізацію тригера, в якому немає небезпечних змагань фронтів імпульсів.

Число логічних елементів у тригері $n_{сп}$ зумовлює *потужність споживання тригера*.

Апаратні трати обраховують витрати за числом корпусів інтегральних мікросхем.

Схемотехнічні параметри допомагають виконувати оптимальний вибір тригера. Слід зауважити, що вибір типу тригера здійснюється за розглядом як схемотехнічних, так і функціональних параметрів. Тут слід звернути увагу на те, що в деяких випадках різні типи тригерів можуть виконувати одну й ту ж функцію. Наприклад, лічильну функцію виконують тригери D -; JK -типів та інші.

4.1.2. Асинхронний RS-тригер

Асинхронним RS-тригером називають логічний прилад з двома стійкими станами, що має два інформаційних входи S та R . Якщо на S -вхід поданий рівень логічної "1" ($S = 1$), а на R – логічний "0" ($R = 0$), тригер RS типу встановлюється, тобто набуває одиничний стан, за якого на виході Q потенціал досягає одиничного рівня напруги ($Q = 1$), а при $S = 0$ і $R = 1$ тригер набуває нульового стану, в якому $Q = 0$.

Тригер RS -типу має найпростішу структуру, яку умовно назвемо базовою. Реалізувати RS тригер можна на будь-яких логічних елементах бульового базису.

На рис. 4.2,*а* показана базова схема *RS*-тригера, яка реалізована на двох логічних елементах 2АБО-НЕ (стрілка Пірса), а на рис. 4.2,*б* – умовне позначення *RS*-тригера.

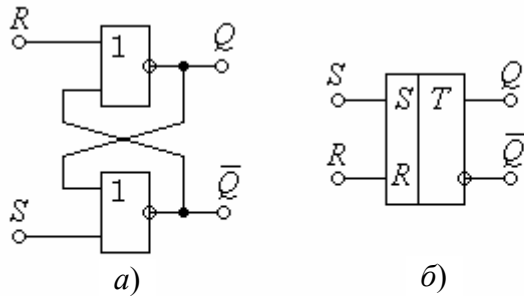


Рисунок 4.2 – Асинхронний *RS*-тригер: *а*) схема; *б*) умовне позначення

Таблиця 4.1 – Стани *RS*-тригера

Q^n	R	S	Q^{n+1}
0	0	0	0
1	0	0	1
X	1	0	0
X	0	1	1
X	1	1	–

Роботу *RS*-тригера віддзеркалює *таблиця переходів* (табл. 4.1), у якій Q^n – попередній стан тригера, а Q^{n+1} – теперішній стан тригера. Хрестиком X позначена незалежність від стану.

З таблиці видно наступне.

При обох нульових входах $R = 0$ та $S = 0$ тригер знаходиться в режимі зберігання, тобто він не спрацьовує, а зберігає попередню інформацію (попередній стан) Q^n : або $Q^n = 0$, або $Q^n = 1$. Якщо тригер був скинутий ($Q^n = 0$) до подачі сигналів $R = 0$; $S = 0$, то скинутим він і залишиться, тобто $Q^{n+1} = 0$. Коли ж тригер до подачі сигналів $R = 0$; $S = 0$ перебував у зведеному (одиночному) стані ($Q^n = 1$), то зведеним він і залишиться, тобто $Q^{n+1} = 1$.

Якщо на нульовий вхід R подати логічну “1” ($R = 1$), а на одиничний вхід S – логічний “0” ($S = 0$), то незалежно від попереднього стану тригер переходить в нульовий стан ($Q = 0$, $\bar{Q} = 1$). *Надалі стан тригера будемо визначати значенням тільки прямого виходу Q* . Дійсно, при $R = 1$ рівень прямого виходу становить $Q = 0$. Тоді до обох входів нижньої схеми АБО-НЕ (рис. 4.2,*а*) надходять логічні “0” ($Q = 0$ та $S = 0$), через що на виході $\bar{Q} = 1$. Так при $R = 1$; $S = 0$ тригер скидається, тобто в нього *записується* логічний “0” (нульовий стан).

Якщо на одиничний вхід S подати логічну “1” ($S = 1$), а на нульовий вхід R – логічний “0” ($R = 0$), то незалежно від попереднього стану тригер устанавлюється в одиничний стан $Q = 1$. Дійсно, при $S = 1$ рівень інверсного виходу становить $\bar{Q} = 0$. Тоді до обох входів верхньої схеми АБО-НЕ надходять логічні “0” ($\bar{Q} = 0$ та $R = 0$), через що на виході $Q = 1$. Так при $S = 1$; $R = 0$ тригер встановлюється, тобто в нього *записується* логічна “1” (одиничний стан).

Якщо ж на входах R і S діють одночасно одиничні сигнали ($S = 1$; $R = 1$), то стан тригера стає невизначеним, що позначено в табл. 4.1 значком “–”. За такої комбінації входних сигналів обидва елементи 2АБО-НЕ мають на своїх виходах рівні логічного “0”, що не визначає стан тригера. Така комбінація

вхідних сигналів для RS -тригера є забороненою і тому її треба уникати, щоб запобігти можливих збоїв у цифровій системі.

Базовий RS -тригер (рис. 4.2) належить до *асинхронних*, через що особливість його функціонування полягає в тому, що він змінює свій стан безпосередньо в момент подачі сигналів на входи S або R .

Принцип дії RS -тригера пояснюється часовою діаграмою роботи, яка показана на рис. 4.3.

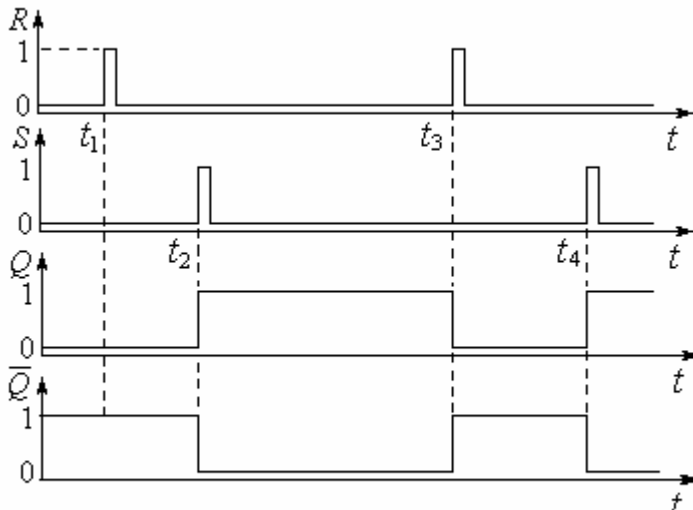


Рисунок 4.3 – Часова діаграма роботи асинхронного RS -тригера

Нехай до моменту t_1 тригер знаходиться у нульовому стані ($Q = 0$; $\bar{Q} = 1$).

У момент t_1 надходження одиничного імпульсу на R -вхід на вході S залишається нульовий рівень, через що згідно з табл. 4.1 тригер зберігає попередню інформацію, тобто $Q^{n+1} = 0$. Цей стан тригера не зміниться до того моменту t_2 , коли на вхід S надійде одиничний імпульс.

У момент t_2 надходження одиничного імпульсу на S -вхід на R -вході залишається нульовий рівень, через що згідно з табл. 4.1

тригер встановлюється, тобто в нього записується логічна «1». Незважаючи на те, що імпульс на S -вході зник, тригер залишається зведеним. Так, він пам'ятає, що в нього записали логічну «1». Цей стан тригера може зберігатись, скільки завгодно довго, доки на R -вхід надійде одиничний імпульс у момент t_3 .

У момент t_3 надходження одиничного імпульсу на R -вхід на S -вході залишається нульовий рівень, через що згідно з табл. 4.1 тригер скидається, тобто в нього записується логічний «0». Незважаючи на те, що імпульс на R -вході зник, тригер залишається скинутим. Так він пам'ятає, що в нього записали логічний «0». Цей стан тригера може зберігатись, скільки завгодно довго, доки на S -вхід надійде черговий одиничний імпульс у момент t_4 і т.д.

З діаграми роботи (рис. 4.3) видно, чим зумовлена назва *асинхронний*, а саме: тригер спрацьовує безпосередньо у будь-які моменти надходження сигналів до настановних входів S та R .

RS -тригер може бути побудований також на логічних елементах 2І-НЕ (штрих Шеффера), якщо в кола R - і S -входів включити інвертори (рис. 4.4). Роботу тригера віддзеркалює таблиця переходів (табл. 4.2).

Порівнюючи таблиці переходів (табл. 4.1) та (табл. 4.2), переконуємось у тому, що тригер на елементах Шеффера (2І-НЕ) працює так само, як і тригер на стрілках Пірса (елементах 2АБО-НЕ).

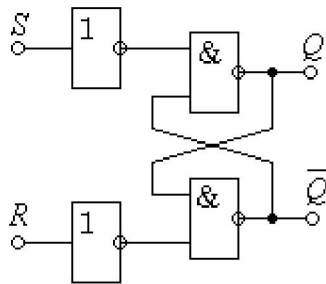


Рисунок 4.4 – Асинхронний RS -тригер на елементах 2I-HE

Таблиця 4.2

Q^n	R	S	Q^{n+1}
0	0	0	0
1	0	0	1
X	1	0	0
X	0	1	1
X	1	1	–

RS -тригер можна побудувати без інверторів у колах R та S на двох елементах Шеффера (2I-HE), якщо керувати тригер інверсними сигналами \bar{R} та \bar{S} . Такий тригер носить назву інверсного $\bar{R}\bar{S}$ -тригера (рис. 4.5).

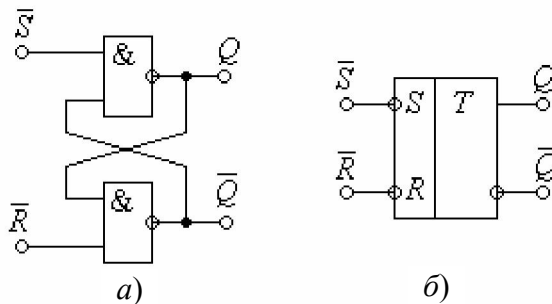


Рисунок 4.5 – Інверсний асинхронний $\bar{R}\bar{S}$ -тригер: а) схема; б) умовне позначення

Таблиця 4.3 – Стани інверсного асинхронного $\bar{R}\bar{S}$ -тригера

Q^n	\bar{R}	\bar{S}	Q^{n+1}
0	1	1	0
1	1	1	1
X	0	1	0
X	1	0	1
X	0	0	–

З таблиці переходів (табл. 4.3) інверсного RS -тригера видно, що активним сигналом для цієї схеми приймається низький рівень логічного “0”.

Інверсний RS -тригер змінює стан на протилежний в наступних випадках: встановлюється при $\bar{R} = 1$; $\bar{S} = 0$; скидається при $\bar{R} = 0$; $\bar{S} = 1$. Для інверсного тригера забороненим є стан, коли на обох інверсних входах діє рівень логічного “0”, тобто $\bar{R} = 0$; $\bar{S} = 0$.

У решті робота тригера з інверсними входами \bar{R} і \bar{S} така сама, як і тригера з прямими входами R і S .

4.1.3. Синхронний RS -тригер

Синхронний RS -тригер позначається як CRS -тригер. Він відрізняється від асинхронного тригера лише тим, що сигнали R і S тільки готують тригер до спрацьовування, а стан тригера змінюється лише тоді, коли надійде синхронізуючий сигнал C .

Синхронний CRS -тригер можна отримати з асинхронного, якщо доповнити його синхронізуючим входом C . CRS -тригер можна реалізувати, наприклад, на базі RS -тригера на чотирьох логічних елементах 2I-HE, як показано на рис. 4.6,а.

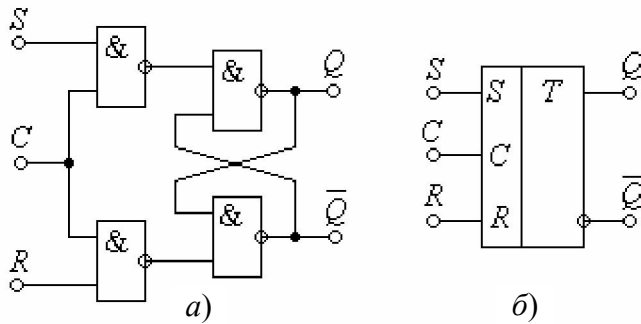


Рисунок 4.6 – $C\overline{R}\overline{S}$ -тригер:
а) схема; б) умовне позначення

Таблиця 4.4 – Стани синхронного RS -тригера

Q^n	R	S	C	Q^{n+1}
0	0	0	⌈	0
1	0	0	⌈	1
X	1	0	⌈	0
X	0	1	⌈	1
X	0	0	⌈	–

Умовне позначення CRS -тригера (рис. 4.6,б) схоже на тригер RS типу (див. рис. 4.2,б). Уся різниця полягає в наявності додаткового входу синхронізації C . Роботу тригера віддзеркалює таблиця переходів (табл. 4.4).

Принцип дії CRS -тригера полягає в тому, що стани входів R і S лише готують тригер до спрацьовування, а встановлення або скид в залежності від станів R - і S -входів здійснюється лише тоді, коли з'явиться імпульс “⌈” (фронт) на вході C .

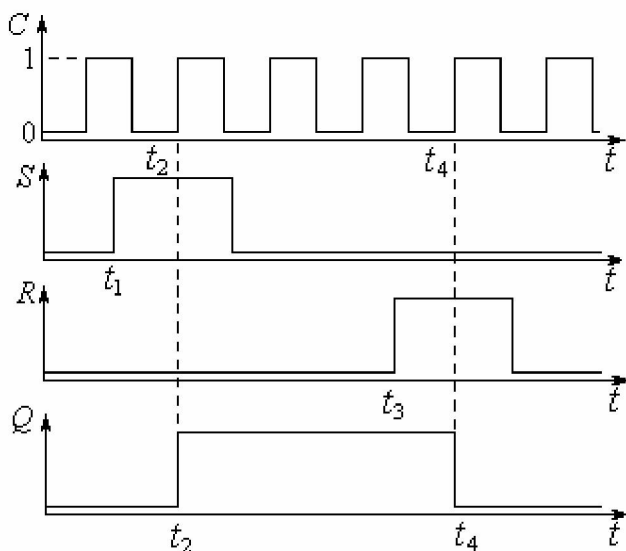


Рисунок 4.7 – Часова діаграма роботи $C\overline{R}\overline{S}$ -тригера

Отже, входи R і S цього тригера відіграють роль керуючих процесом спрацьовування тригера, а саме ж спрацьовування відбувається тільки з надходженням синхроімпульсу C . Часова діаграма роботи CRS -тригера показана на рис. 4.7.

Нехай тригер знаходиться в нульовому стані. Якщо на вхід S надходить логічна “1” у будь-який час t_1 , то тригер у цей момент не спрацьовує. Він встановиться тільки при надходженні синхронізуючого імпульсу на вхід C у момент t_2 .

Аналогічна послідовність надходження і при скиді тригера. Незважаючи на те, що одиничний сигнал надходить до входу R у момент t_3 , тригер скинеться лише в момент t_4 , коли на синхровході C з'явиться фронт, тобто спочатку подається сигнал скиду R , а потім синхросигнал C скидає тригер.

Синхронний CRS -тригер, схема якого показана на рис. 4.6, можна перетворити в асинхронний, якщо на синхровхід C подати одиничний потенціал на весь час роботи.

4.1.4. D-тригер

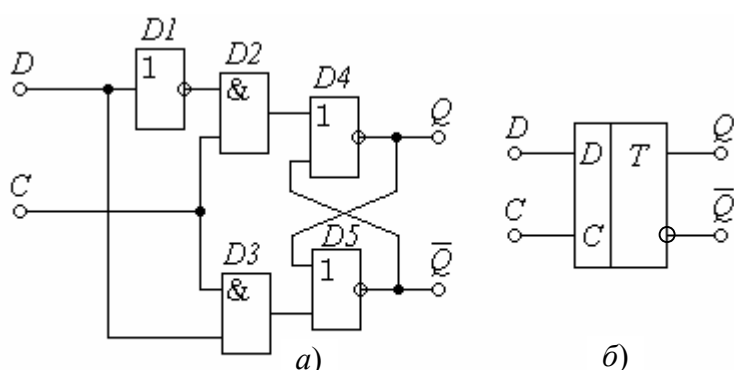
D-тригер або тригер-затримка (названий так від англійського *Delay* – затримка) – це елементарний автомат, який має один інформаційний вхід *D* та вхід синхросигналу *C*. *D*-тригер є синхронним. *D*-вхід готує тригер до спрацьовування, а саме спрацьовування здійснюється фронтом синхросигналу *C*, тобто фронтом *C* тригер копіює стан входу *D*.

D-тригер, як і будь-який синхронний, має два види запуску: роздільний та лічильний.

У роздільному запуску *D*-вхід підключений до якогось зовнішнього пристрою, а в лічильному – до інверсного виходу саме цього тригера.

Наявність лише одного інформаційного входу *D* є перевагою цього тригера перед двовходовими, бо зменшується кількість міжкаскадних зв'язків.

Схема *D*-тригера та його умовне позначення показані на рис. 4.8.



Таблиця 4.5 –
Стани *D*-тригера

Q^n	<i>D</i>	<i>C</i>	Q^{n+1}
0	X	0	0
1	X	1	1
0	0	Г	0
1	0	Г	1
X	0	Г	0
X	1	Г	1

Рисунок 4.8 – *D*-тригер: а) схема; б) умовне позначення

Тут організація *D*- та *C*-входів здійснюється відповідно на інверторі *D1* та схемах І *D2* та *D3*. Схеми АБО *D4* та *D5* утворюють *RS*-тригер.

Закон функціонування *D*-тригера наданий в таблиці переходів (табл. 4.5). З цієї таблиці видно, що і при *C* = 0, і при *C* = 1 тригер зберігає свій попередній стан, тобто зберігає інформацію.

Спрацьовує ж тригер тільки від фронту синхросигналу *C*, коли його рівень змінюється від 0 до 1 (“Г”).

D-тригер не сприймає зміну на вході *D* при постійних сигналах *C* = 0 та *C* = 1, а також при зрізі, коли синхросигнал *C* змінюється від 1 до 0.

У відсутності синхроімпульсів на вході *C* тригер перебуває у стані зберігання інформації Q^n і на зміну сигналу *D*-входу не реагує. Останнє зумовлює високу завадостійкість тригера за інформаційним входом *D*.

D-тригер у роздільному запуску працює наступним чином.

Нехай до моменту t_1 (рис. 4.9) тригер був скинутий ($Q^n = 0$). При *D* = 0 на виході інвертора *D1* (рис. 4.8) діє логічна одиниця, яка готує до спрацьовування схему І *D2*.

Якщо *C* = 0, то схеми І *D2* та *D3* заперті і можуть спрацьовувати лише в моменти t_1, t_2, t_3 і т.д., коли на *C*-вхід надійдуть фронти (рис. 4.9).

У момент t_1 з надходженням фронту *C* на виході *D2* діє логічна “1”, яка встановлює на виході *D4* нуль ($Q = 0$).

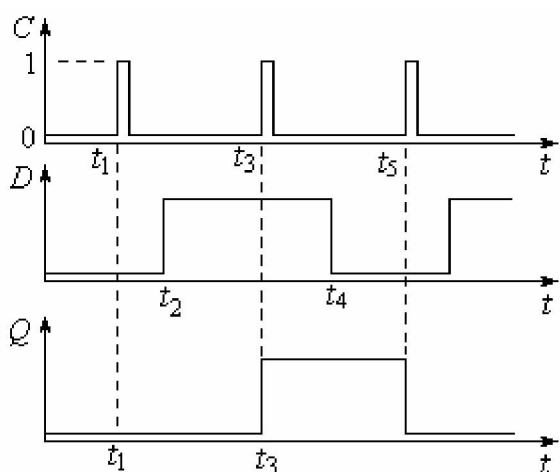


Рисунок 4.9 – Часова діаграма роботи D-тригера в роздільному запуску

Проте при $D = 1$ схема $D3$ підготовлена до спрацьовування і в момент t_3 фронтом C на її виході встановлюється одиниця, яка схемою $D5$ забезпечує $\bar{Q} = 0$. Тоді на обох входах $D4$ логічні «0», через що $Q = 1$. Отже при $D = 1$ з надходженням фронту C тригер переходить у стан $Q = 1$, а $\bar{Q} = 0$, тобто одиничний (встановлюється). Так інформаційний вхід $D = 1$ встановлює тригер ($Q = 1$) тільки з надходженням фронту C .

Таким чином, щоб записати в D -тригер логічний “0”, треба до D -входу підвести нуль і подати на C -вхід фронт. Щоб записати логічну “1”, слід до D -входу підвести одиницю і подати на C -вхід фронт.

Слід зауважити, що для роботи з D -тригером у роздільному запуску слід виконувати часові погодження надходження сигналів на його входи. Щоб копіювання стану входу D пройшло без помилок, необхідно установлювати рівні на вході D до надходження фронту (або зрізу) синхроімпульсу C .

Це означає, що всі зміни стану D -входу разом з перехідними процесами мають закінчитися за деякий час до приходу фронту (або зрізу) синхроімпульсу C . Цей час носить назву *часової підготовки*. Тому всі зміни рівнів на входах тригера можна починати не раніше закінчення цього часу.

Бувають випадки, коли необхідні D -тригери, які синхронізуються низьким рівнем, тобто встановлення та скид тригера виконується під час дії на синхровході C зрізу, тобто переходу синхронізуючого імпульсу на вході C з одиниці в нуль. Такий тригер має умовне позначення, яке показано на рис. 4.10, і носить назву *D-тригера зі зворотним динамічним входом*.

Розглянемо лічильний запуск D -тригера.

У лічильному запуску D -тригер з кожним імпульсом на синхровході C змінює свій стан на протилежний.

Це здійснюється з'єднанням входу D з інверсним виходом \bar{Q} .

Проте на входах і виході $D3$ нулі, через що стан $D5$ не змінюється. Отже $Q = 0$, а $\bar{Q} = 1$, тобто тригер перебуває в нульовому стані.

Так, інформаційний вхід $D = 0$ скидає тригер ($Q = 0$) тільки з надходженням фронту C . (У момент t_1 нульовий стан підтверджується).

Нехай в момент t_2 до входу D надійшов одиничний сигнал ($D = 1$).

Тригер на нього не реагує, бо в момент t_2 стан $C = 0$ запирає схеми $D2$ та $D3$. На їхніх виходах нулі, які не змінюють станів $D4, D5$.

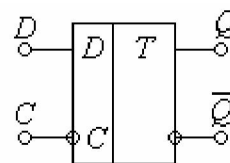


Рисунок 4.10 – D-тригер зі зворотним динамічним входом

Схема D -тригера з лічильним запуском та часова діаграма його роботи показані на рис. 4.11.

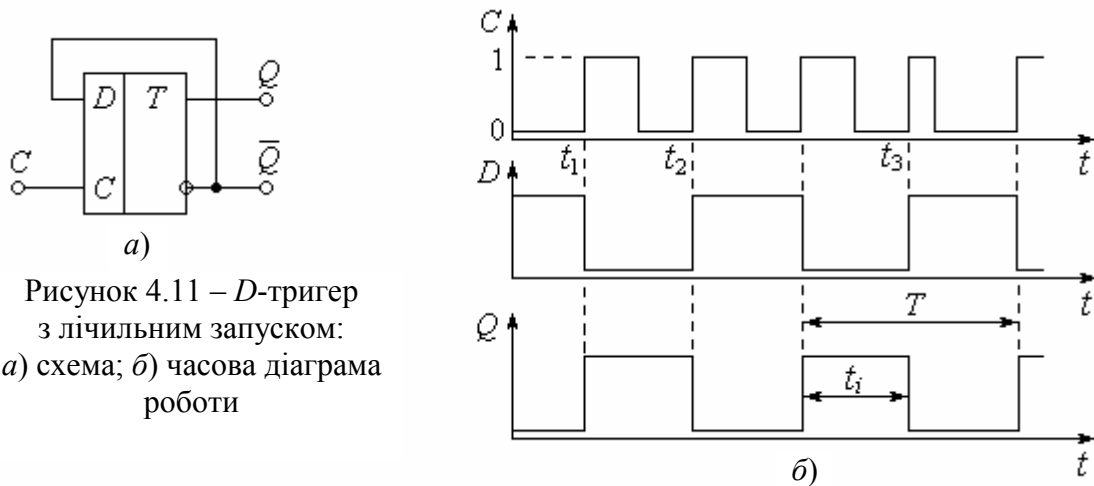


Рисунок 4.11 – D -тригер з лічильним запуском:
а) схема; б) часова діаграма роботи

D -тригер з лічильним запуском працює наступним чином.

Нехай до моменту t_1 тригер перебував у нульовому стані ($Q = 0$, $\bar{Q} = 1$). Оскільки вхід D закорочений з інверсним виходом \bar{Q} , то в нульовому стані тригера $D = \bar{Q} = 1$, тобто стан входу D одиничний і підготовлює тригер до взведення. В момент t_1 фронтом C тригер зводиться ($Q = 1$, $\bar{Q} = 0$).

Починаючи з моменту t_1 тригер в одиничному стані, через що $D = 0$, тобто стан входу D нульовий і підготовлює тригер до скиду, який відбувається фронтом C у момент t_2 і т.д.

З діаграми роботи видно, що в лічильному запуску з кожним фронтом C стан тригера змінюється на протилежний.

Результатом роботи D -тригера з лічильним запуском є наступне:

- поділ частоти слідкування вхідних імпульсів C на два;
- забезпечення щільності вихідних імпульсів, яка дорівнює двом.

Щодо ділення частоти на два, то з діаграми роботи видно, що період вихідних імпульсів Q вдвічі більший за період вхідних імпульсів C .

Забезпечення ж щільності два пояснюється наступним.

Щільністю імпульсів є відношення їхнього періоду T до тривалості t_i :

$$N = \frac{T}{t_i}. \quad (4.4)$$

Як видно з діаграми, тривалість імпульсів t_i зберігається навіть тоді, коли щільність імпульсів C відрізняється від двох (момент t_3). Тому щільність два вихідних імпульсів тригера з лічильним запуском гарантована.

4.1.5. JK - та MS -тригери

4.1.5.1. JK -тригер

JK -тригер має два інформаційні входи J і K та вхід синхросигналу C .

JK -тригер, як і CRS -тригер є синхронним, але має ту перевагу, що на відміну від CRS -тригера не має заборонених (непевних) станів. Ця перевага

стала основою для побудови цілої низки синхронних тригерів в інтегральному виконанні.

Найбільшого практичного застосування в інтегральній схемотехніці дістали синхронні одноступеневі *JK*-тригери.

Одноступеневий синхронний *JK*-тригер будується за аналогією синхронного *CRS*-тригера.

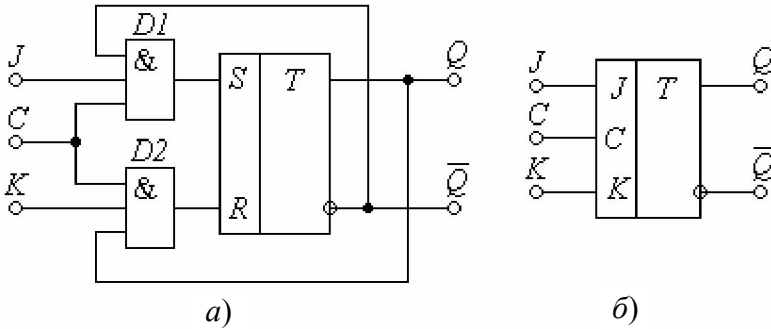


Рисунок 4.12 – *JK*-тригер:
а) схема; б) умовне позначення

Схема *JK*-тригера (рис. 4.12) побудована на *RS*-тригері і двох логічних схемах збігу *D1* та *D2*.

У початковому (нульовому) стані тригера ($Q = 0$; $\bar{Q} = 1$) з виходу \bar{Q} до входу *D1* подається одиничний рівень, підготовлюючи саме тим установлення (введення) тригера. Тригер установиться,

коли $J = 1$, $K = 0$ і до входу *C* надійде фронт “Г”.

У введеному (одиничному) стані тригера ($Q = 1$; $\bar{Q} = 0$) з виходу *Q* одиничний рівень подається до входу *D2*, готуючи саме тим скид тригера. Тригер скинеться, коли $K = 1$, $J = 0$ і до входу *C* надійде фронт “Г”.

Роботу тригера видно з табл. 4.6 та часової діаграми роботи в роздільному запуску (рис. 4.13).

З табл. 4.6 видно, що одноступеневий *JK*-тригер спрацьовує лише за наявності фронту “Г” на вході синхронізації *C*. За відсутності синхроімпульсу, коли $C = 0$, входи *J* і *K* блокуються і тригер залишається в попередньому стані Q^n .

Комбінація сигналів на *J*- і *K*-входах визначає стан, в якому буде тригер після надходження фронту на вхід *C*.

Таблиця 4.6 – Стани *JK*-тригера

n	J^n	K^n	C	Q^{n+1}
0	X	X	0	0
1	X	X	0	1
0	0	0	Г	0
1	0	0	Г	1
X	0	1	Г	0
X	1	0	Г	1
X	1	1	Г	\bar{Q}^n

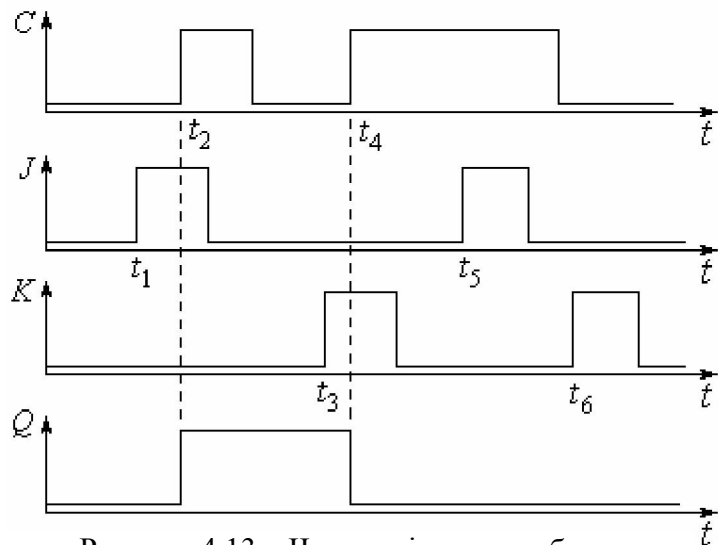


Рисунок 4.13 – Часова діаграма роботи одноступеневого *JK*-тригера

Так, до моменту t_1 (рис. 4.13) тригер знаходиться в

нульовому стані ($Q = 0$).

У момент t_1 комбінація $J = 1$; $K = 0$ підготовлює тригер до переходу в одиничний стан ($Q = 1$), але це відбудеться тільки в момент t_2 з надходженням фронту C .

У момент t_3 комбінація $J = 0$; $K = 1$ готує тригер до переходу в нульовий стан ($Q = 0$), який установиться фронтом C у момент t_4 .

Після моменту t_4 , коли на вході C немає фронтів, як би не змінювались сигнали на входах J і K (моменти t_5 ; t_6), тригер не спрацює.

З табл. 4.6 видно, що при $J = 1$ та $K = 1$ стан JK -тригера змінюється на протилежний з приходом кожного синхроімпульсу C , тобто здійснюється лічильний запуск. Отже, для здійснення лічильного запуску слід подати логічну одиницю одночасно на входи J і K ($J = 1$ і $K = 1$). У цьому випадку одноступеневий JK -тригер працює в лічильному запуску, здійснюючи поділ частоти на два.

На базі одноступеневих синхронних JK -тригерів можна комутацією зовнішніх виводів будувати різні типи тригерів, наприклад, D -тригер (рис. 4.14).

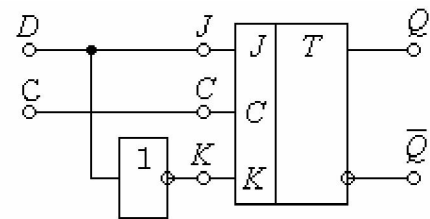
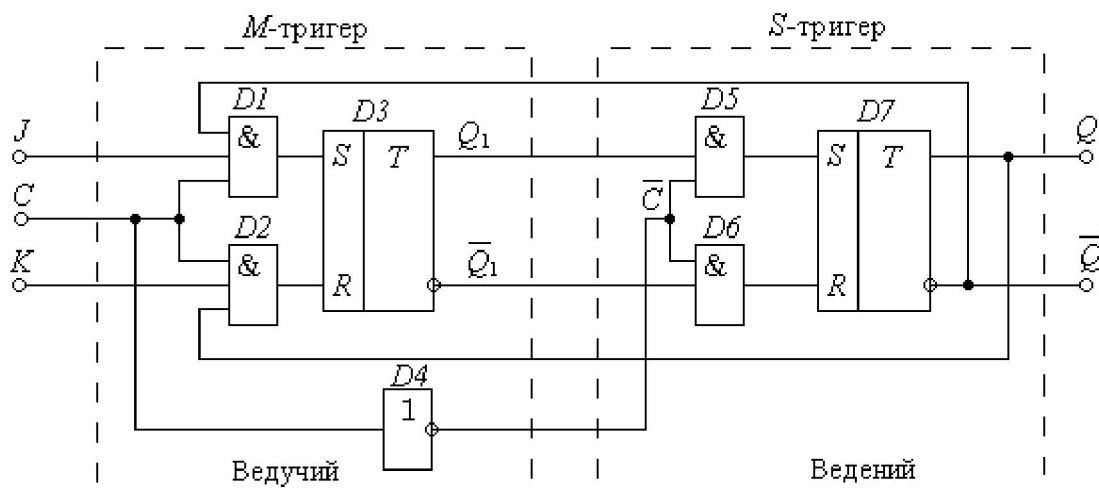


Рисунок 4.14 – D -тригер на базі JK -тригера

4.1.5.2. MS -тригер

MS -тригер містить два каскадно з'єднанні RS -тригери: основний і допоміжний. Звідси назва MS (скорочено від англійських слів *master* – господар і *slave* – раб, невільник). Ці тригери працюють за принципом “ведучий – ведений” (рис. 4.15).

Рисунок 4.15 – MS -тригер



У MS -тригері для приймання (запису) і передавання (зчитування) інформації використовуються два фрагменти: і фронт, і зріз синхросигналу. Тригери цього типу ще називають тригерами з внутрішньою затримкою або двоступеневими синхронними тригерами.

Основною перевагою двоступеневих синхронних тригерів перед одноступеневими є їхня підвищена завадостійкість.

Ця перевага досягається тим, що спрацьовування *MS*-тригера в цілому відбувається двома етапами:

- при переході синхроімпульсу у стан $C = 1$ здійснюється запис інформації тільки в *M*-тригер відповідно станам входів *J* і *K* з одночасним блокуванням (запиранням) інформаційних входів *S*-тригера;

- при зворотному переході у стан $C = 0$ синхроімпульсу стан *M*-тригера записується в *S*-тригер з одночасним блокуванням (запиранням) інформаційних входів *M*-тригера.

Отже, на відміну від одноступеневого тригера, в якому *запис* інформації та її *передавання* збігаються, в двоступеневому тригері вони рознесені в часі, причому, під час запису інформації в один тригер блокуються інформаційні входи іншого, вилучаючи вплив завад.

Двоступеневий синхронний *JK*-тригер побудований на базі каскадного з'єднання двох одноступеневих синхронних *RS*-тригерів та схеми блокування на логічному елементі *D4* (рис. 4.15).

Перший тригер (*D1*, *D2*, *D3*) називається *M*, а другий – *S* (*D5*, *D6*, *D7*). Особливістю запуску *S*-тригера є те, що в його коло синхросигналу включений інвертор *D4*, який не припускає передачу стану тригера *M* в *S* під час запису. Наявність інвертора *D4* забезпечує рознесення активних рівнів спрацьовування *M*- і *S*-тригерів.

Так, для *M*-тригера активним сигналом запису інформації є $C = 1$, а для *S*-тригера – $\bar{C} = 1$, тобто $C = 0$.

Тоді стан *M*-тригера змінюється в момент переходу синхросигналу *C* з нуля в одиницю (фронтом імпульсу), а стан *S*-тригера – в момент переходу синхросигналу з одиниці в нуль (зрізом імпульсу). У цілому, таким чином, *MS*-тригер запускається *зрізом*.

В *MS*-тригері існує небезпека хибного спрацьовування через так звані *гонки* між моментами запису в *M*- і *S*-тригери. Запобігання цьому недоліку здійснюється вибором порогів спрацьовування інвертора *D4* та елементів *D1* і *D2*.

Співвідношення згаданих порогів ілюструє часова діаграма роботи тригера (рис. 4.16).

Тут:

$U_{1,2}$ – поріг спрацьовування елементів *D1* і *D2*;

U_4 – поріг спрацьовування інвертора *D4*;

$U_{\text{вих } 1,2}$ – вихідна напруга елементів або *D1*, або *D2*.

Спрацьовування *MS*-тригера без впливу завад вихідних кіл Q та \bar{Q} відбувається

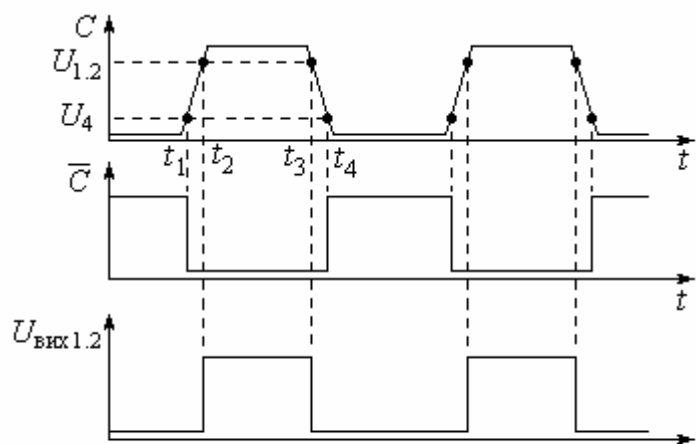


Рисунок 4.16 – Часова діаграма формування синхросигналів *MS*-тригера

наступним чином.

Поріг U_4 спрацьовування інвертора $D4$ нижчий за поріг $U_{1.2}$ спрацьовування елементів $D1$ і $D2$, тобто $U_4 < U_{1.2}$. Тому на початку появи напруги синхроімпульсу C входи S і R тригера $D3$ закриті (заблоковані) елементами $D1$ і $D2$ і лишатимуться заблокованими, доки напруга C не перевищить поріг $U_{1.2}$.

Коли напруга C в момент t_1 досягне порогу U_4 , спрацьовує інвертор $D4$, через що синхроімпульс \bar{C} стає нульовим $\bar{C} = 0$. Тоді елементи $D5$ і $D6$ заблоковані і закривають входи S - і R -тригера $D7$. При $\bar{C} = 0$ стан S -тригера ні в якому разі змінитися не може.

В момент t_2 напруга C досягає порогу $U_{1.2}$ спрацьовування елементів $D1$ і $D2$, через що розблоковуються елементи $D1$ і $D2$ і на одному з їхніх виходів, в залежності від стану входів J і K , з'являється напруга $U_{\text{вих } 1.2}$, яка забезпечує спрацьовування M -тригера. При цьому S -тригер спрацювати не може, бо синхроімпульс $\bar{C} = 0$ блокує його входи.

В момент t_3 напруга $U_C < U_{1.2}$, через що блокуються входи M -тригера і тому його стан ні в якому разі змінитися не може.

В момент t_4 напруга U_C стає менша за поріг спрацьовування інвертора $D4$, через що його вихідна напруга стає одиничною ($\bar{C} = 0$), розблоковуючи входи S -тригера. Тоді стан M -тригера переписується в S -тригер.

Таким чином, запис в один з M - або S -тригерів відбувається лише тоді, коли другий заблокований. При цьому вихідні кола Q та \bar{Q} функціонально ізольовані від ведучого M -тригера, чим зумовлене підвищення завадостійкості.

Роботу двоступеневого синхронного JK -тригера або MS -тригера можна пояснити за допомогою часової діаграми роботи, яка показана на рис. 4.17.

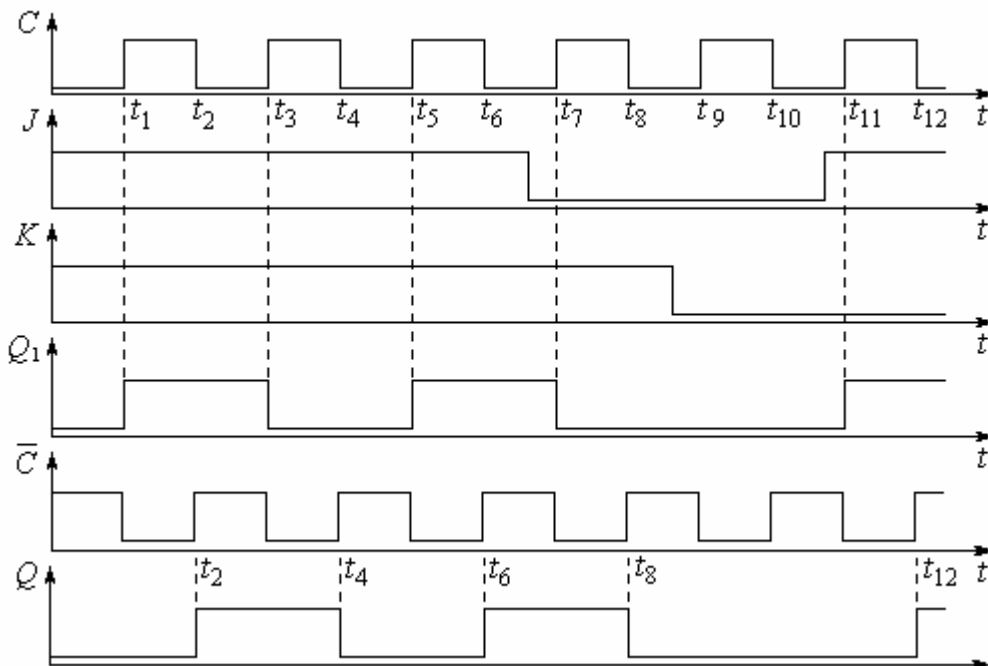


Рисунок 4.17 – Часова діаграма роботи MS -тригера

При $J = 1; K = 1$ в інтервалі моментів $t_1 \dots t_5$ тригер працює у лічильному запуску, аналогічно випадку одноступеневого JK -тригера (див. табл. 4.5). Різниця лише в тому, що в двоступеневому тригері зміна стану тригера в лічильному режимі відбувається за зрізом, а не за фронтом синхроімпульсів C (моменти $t_2; t_4, t_6 \dots$).

Якщо, наприклад, $J = 0$ і $K = 1$ ($t_7 - t_8$), то тригер має перейти у стан $Q^{n+1} = 0$.

Це виконується в момент t_8 , коли синхросигнал на вході C переходить з одиниці в нуль, бо лише в цей момент відбувається зміна або збереження стану S -тригера під дією синхроімпульсу \bar{C} .

Запис інформації в M -тригер (тобто в MS -тригер) здійснюється в момент t_7 , коли синхросигнал C на вході MS -тригера переходить з нуля в одиницю, а зчитування – в момент t_8 , тобто на наступному такті періоду синхроімпульсів.

Якщо $J = 0; K = 0$ ($t_9 - t_{10}$), то тригер перебуває в режимі зберігання попереднього біта інформації.

Якщо на момент t_{11} стани входів $J = 1; K = 0$, то M -тригер переходить у стан 1, здійснюючи при цьому запис 1 в JK -тригер. В наступний такт t_{12} у стан 1 переходить S -тригер і при цьому на Q -виході MS -тригера встановлюється рівень логічної "1", що відповідає процесу зчитування одиничної інформації з виходу MS -тригера.

Важливою перевагою двоступеневого JK -тригера є те, що інформаційні входи за період синхроімпульсів функціонально ізольовані від вихідних кіл. Така своєрідна властивість MS -тригера забезпечує стійке переключення тригера. Завдяки таким властивостям на двоступеневих JK -тригерах можна будувати завадостійкі цифрові пристрої.

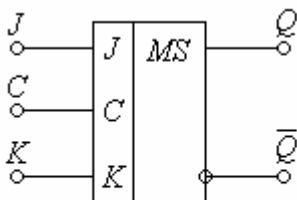


Рисунок 4.18 – Умовне позначення MS -тригера

Умовне позначення MS -тригера показане на рис. 4.18.

У мікросхемотехніці широке застосування має універсальний JK -тригер, що крім інформаційних J - та K -входів і входу синхронізації C , має також несинхронізовані входи S і R або \bar{S} і \bar{R} . Під час переключення тригера з одного стану в інший під дією сигналів на інформаційних входах J і K та синхронізуючому вході C , на входи S і R (\bar{S} і \bar{R}) слід подати пасивні рівні, тобто $S = 0; R = 0$ або $\bar{S} = 1; \bar{R} = 1$.

На базі двоступеневого JK -тригера можна будувати інші схеми тригерів шляхом комутації зовнішніх входів J, K, C, S та R .

4.1.6. Підвищення завадостійкості тригерів

Для підвищення завадостійкості тригерів захист необхідно не тільки по входах, а й по виходах. Це пояснюється тим, що виходи тригера через кола зворотного зв'язку зв'язані із входами.

Дійсно, якщо на лінії зв'язку, підключеної до виходу Q тригера (див. рис. 4.2,а), з'явиться одинична завада, то вона впливатиме і на вхід

протилежного елемента тригера, що може викликати переключення усього тригера, як від звичайного вхідного сигналу.

Тому, якщо тригер будь-якого типу $D1$ (рис. 4.19) працює на лінію, де можливі завади, то його слід підключати до лінії через буфер $D2$, яким може бути RS -тригер.

Оскільки на входах буферного тригера завжди присутній або R -, або S -сигнал, то цей тригер не може запам'ятовувати заваду, яка діє на виході Q або \bar{Q} . Після кінця дії завади буферний тригер обов'язково повертається в попередній стан. При запусках будь-яких тригерів по входах R і S потрібно зауважити, що вони є *настановними* і тому мають пріоритет у своєму впливі на стан тригера при порівнянні з усіма іншими входами. Тобто входи R і S устанавлюють стан тригера в залежності від власних рівнів незалежно від сигналів, що діють у цей час на других входах, в тому числі і на синхровході C , коли він є. Отже входи R і S є *асинхронними*. Із закінченням дії асинхронного сигналу устанавлений ним стан тригера залишається до приходу фронту C -сигналу. За цим фронтом запускаються по C -входах тільки від занадто стрімких фронтів. Напруга ж завад тригера спрацює вже у відповідності з діючими рівнями на керуючих входах.

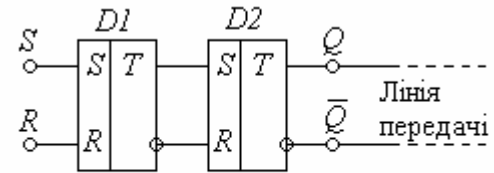


Рисунок 4.19 – Тригер $D1$ з буфером $D2$

Переходячи до *синхронних* тригерів, слід зауважити, що вони, джерелом яких можуть бути контакти, тумблери, кнопки, реле, імпульсні сигнали різноманітних електронних схем, сигнали від інших цифрових приладів тощо, здебільшого відрізняється за своїми характеристиками від сигналів, які потрібні для запуску тригера.

Це дає змогу шляхом формування сигналів зробити тригер нечутливим до завад.

Перш за все вхідний сигнал повинен мати полярність і амплітуду, які відповідають рівням сигналів означених елементів цифрового приладу. Ці умови легко виконати, якщо ввести відповідні подільники напруги або підсилювачі. Для взаємних переходів між типовими рівнями сигналів випускають спеціальні узгоджувачі.

Досить значними умовами є такі, що пред'являють вимоги до часових параметрів сигналів. Такими суттєвими вимогами є забезпечення *тривалості фронтів* вхідних сигналів та усунення впливу *дренькоту контактів*. Тут не обійти застосування формувачів.

4.2. Формувачі тривалості фронтів

Формувачі тривалості фронтів призначені для підвищення крутизни фронтів, тобто зменшення їхньої тривалості. Це необхідно, наприклад, для надійного запуску тригерів по входах синхронізації C , бо вхід C реагує на fronti лише тієї тривалості, яка не перевищує певного часу.

До тривалості фронтів критичні не тільки тригери, а й, наприклад, схеми збігу. Дійсно, якщо на двох входах схеми 2І діють відповідно два одиничні прямокутні імпульси з різними тривалостями фронтів, то на межі цих імпульсів з'явиться непередбачений логічний нуль на час, який дорівнює згаданій різниці тривалості фронтів. Це явище називають *змаганням фронтів* або *гонками*.

Отже, в багатьох випадках треба зменшити тривалість фронтів (зрізів).

Взагалі, за цілою низкою обставин крутизну фронтів вхідних імпульсів необхідно мати близькою до необхідної.

Для перетворення пологих фронтів на круті найчастіше використовують *тригер Шмітта*.

Тригер Шмітта – це несиметричний тригер, який є пороговим пристроєм. Він змінює свою вихідну напругу стрибком, коли вхідна напруга досягає заданого порогу. Тому він використовується для формування з вхідної напруги, яка змінюється повільно, прямокутних імпульсів вихідної напруги з крутими фронтами.

Умовне позначення тригера Шмітта показано на рис. 4.20,а, а його типова передавальна характеристика $U_{\text{вих}} = f(U_{\text{вх}})$ – на рис. 4.20,б.

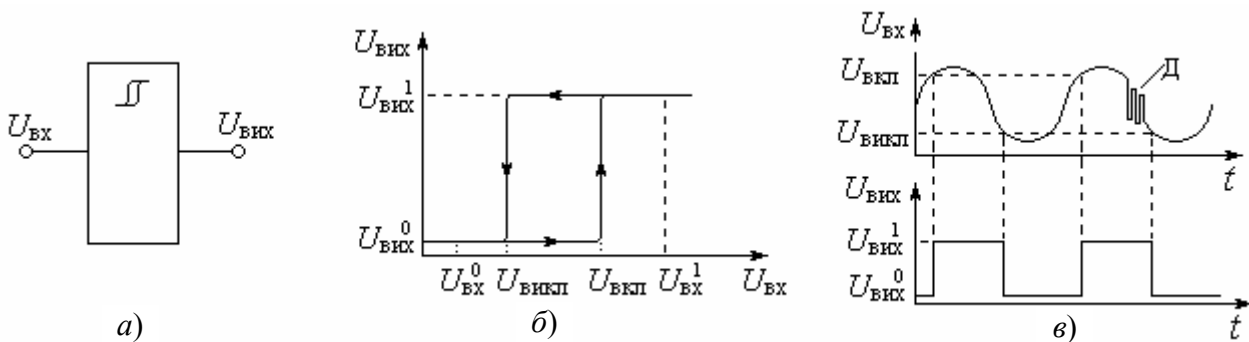


Рисунок 4.20 – Тригер Шмітта:

а) умовне позначення; б) передавальна характеристика; в) діаграма роботи

Основною властивістю тригера Шмітта є наявність двох *рознесених* порогів: верхній поріг включення $U_{\text{вкл}}$ та нижній поріг виключення $U_{\text{вкл}}$ (рис. 4.20,б; 4.20,в).

Коли вхідна напруга $U_{\text{вх}}$ перевищить верхній поріг $U_{\text{вкл}}$, то вихідна напруга $U_{\text{вих}}$ стрибком підвищується до рівня логічної одиниці $U_{\text{вих}}^1$. Коли ж $U_{\text{вх}}$ стане менше нижнього порогу $U_{\text{вкл}}$, то вихідна напруга $U_{\text{вих}}$ стрибком зменшиться від $U_{\text{вих}}^1$ до рівня логічного нуля $U_{\text{вих}}^0$.

Отже, прямокутна передавальна характеристика тригера Шмітта дозволяє використовувати його за формувач прямокутної вихідної напруги з повільної вхідної.

Слід зауважити, що переходи тригера з одного сталого стану в інший відбуваються за різних значень вхідної напруги $U_{\text{вкл}}$ та $U_{\text{вкл}}$. Різниця напруг пороїв включення та виключення називається *гістерезисом переключення*.

Гістерезис захищає тригер від дренькоту Д (рис. 4.20,в), а також від повторного спрацьовування, коли на вхідний сигнал накладені високочастотні шуми (в межах гістерезису).

Щодо схеми тригера Шмітта, то одна з них показана на рис. 4.21.

Вона побудована на операційному підсилювачі А, який охоплений від'ємним зворотним зв'язком (ЗЗ) (резистори R_3, R_1) для отримання необхідного вихідного рівня, та додатним ЗЗ (резистори R_4, R_2) для забезпечення гістерезису. Пороги включення та виключення встановлюються напругою $U_{\text{вкл}}$.

Тригер працює наступним чином.

У початковому стані при $U_{\text{вх}} = 0$ вихідна напруга теж нульова $U_{\text{вих}} = U_{\text{вих}}^0$ (див. рис. 4.20,б).

З повільним підвищенням вхідної напруги $U_{\text{вх}}$ тригер залишається в цьому нульовому стані доки $U_{\text{вх}}$ не перевищить поріг включення $U_{\text{вкл}}$.

При $U_{\text{вх}} > U_{\text{вкл}}$ вихідна напруга, завдяки додатному ЗЗ, стрибком змінюється на одиничну $U_{\text{вих}} = U_{\text{вих}}^1$, формуючи *крутий фронт*, не зважаючи на *повільну* вхідну напругу. Подальше збільшення $U_{\text{вх}}$ не впливає на вихідну напругу, залишаючи її одиничною $U_{\text{вих}}^1$. Вихідна одинична напруга $U_{\text{вих}}^1$ підвищує потенціал неінвертуючого входу $U_{\text{н}}$. Це підвищення $U_{\text{н}}$ підтримує тригер в одиничному стані навіть коли $U_{\text{вх}}$ стане менше за поріг $U_{\text{вкл}}$, бо при підвищеному $U_{\text{н}}$ зменшується поріг до $U_{\text{вкл}}$.

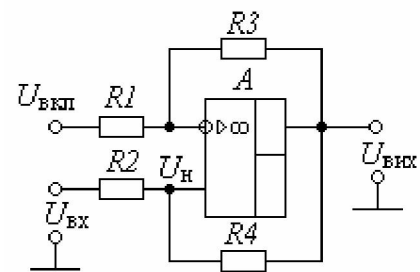


Рисунок 4.21 – Тригер Шмітта

Коли вхідна напруга зменшиться до $U_{\text{вх}} < U_{\text{вкл}}$, то вихідна напруга стрибком змінюється на нульову, збільшуючи при цьому поріг включення до $U_{\text{вкл}}$.

Таким чином, тригер Шмітта формує прямокутну вихідну напругу з повільної вхідної. Ширина петлі гістерезису $U_{\text{вкл}} - U_{\text{вкл}}$ визначається співвідношенням опорів R_4, R_2 . Збільшення R_4 звужує петлю гістерезису $U_{\text{вкл}} - U_{\text{вкл}}$.

Щодо застосування тригера Шмітта, то воно досить широке, але зводиться до необхідності формування прямокутної напруги, придушення дренькоту та завад у межах гістерезису.

Одним із прикладів застосування тригера Шмітта є запуск будь-якого тригера по синхровходу від зовнішнього джерела сигналу (рис. 4.22).

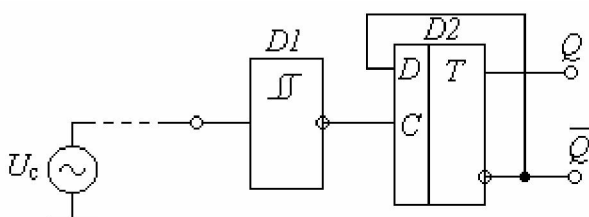


Рисунок 4.22 – Запуск D-тригера від віддаленого джерела сигналу

Сигнал від джерела U_c на трасі набуває завад і втрати крутизни фронтів, через що надійний запуск послідовнісного пристрою по синхровходу стає неможливим. Тому слід мати за правило, що будь-який послідовнісний пристрій має запускатися по синхровходу тільки

від місцевого тригера. Таким місцевим тригером і є тригер Шмітта $D1$, через який запускається по синхровходу C тригер $D2$ (рис. 4.22).

Тригери Шмітта використовують і в інших схемах для надійної роботи приладів, особливо при надходженні вхідного сигналу від зовнішніх приладів, або від іншої плати.

4.3. Методи боротьби з дренькотом контактів

Дренькотом контактів називають процес вібрації контактів, які викликають удари при вмиканні. Після першого доторкання контакт розмикається, після чого знов замикається і так декілька разів.

Дренькіт спостерігається у будь-яких контактах: реле, тумблерів, кнопок. Частота комутації дренькоти лежить в межах 100 Гц ... 10 кГц. Якщо пристрій запускається від згаданих контактів, то на такі повторювання він може реагувати як на окремі вхідні сигнали. Наприклад, додаток одиниці у лічильник може повторюватися декілька разів.

Ліквідувати вплив дренькоти контактів можна за допомогою тригера Шмітта (рис. 4.22).

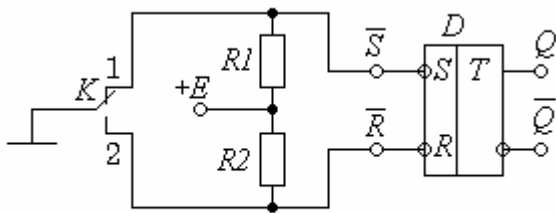


Рисунок 4.23 – Схема захисту від дренькоти контактів

Якщо контакт перекидаючий, то доцільно використати $\overline{R}\overline{S}$ -тригер (рис. 4.23).

При спрацьовуванні реле рухомий контакт K починає рухатися від контакту 1 до контакту 2. За час цього руху на обох входах тригера резистори $R1$ та $R2$ підтримують високі рівні, через що тригер зберігає попередній стан, в якому він був до початку руху контакту.

При першому доторканні точки 2 тригер, інерційність якого на порядок менша за контакт, перекидається в новий стан. Відрив контакту при дренькоті не перекидає тригер назад, бо потенціали входів \overline{S} і \overline{R} залишаються одиничними, зберігаючи стан тригера.

4.4. Лічильники імпульсів

Лічильники імпульсів – це пристрої, які виконують операцію підраховування числа імпульсів, що надійшли до їхніх входів.

У загальному випадку лічильником є пристрій, який може переходити з одного стану в інші під дією вхідних імпульсів, які належить лічити. З надходженням вхідних імпульсів лічильник перебирає свої стани у визначеному для даної схеми порядку. Тому, якщо лічильник має лічити до 10 імпульсів, то він повинен мати 10 різноманітних станів. При цьому кожний 10-й імпульс повинен повертати схему до початкового стану.

4.4.1. Параметри лічильника

Число станів є основним параметром лічильника і носить назву *коефіцієнта перерахунку* K_n або *модуля лічби* M .

Найпростішим лічильником є тригер з лічильним входом, який виконує лічбу імпульсів за модулем $M = 2$, тобто він має два стани 0 та 1, які змінюються по чергово під дією вхідних імпульсів.

У загальному виді модуль лічби двійкового лічильника визначається формулою

$$M = 2^m, \quad (4.5)$$

де m – число двійкових розрядів лічильника.

Отже, модуль лічби визначає число стійких станів лічильника (у тому числі і нульовий стан) або кількість імпульсів, яку треба підвести до входу лічильника, щоб він повернувся у початковий стан, утворюючи при цьому на своєму виході імпульс переповнення.

Поруч з модулем лічби лічильник характеризується його *місткістю*

$$K = M - 1, \quad (4.6)$$

тобто максимальним числом одиниць, яке він може накопичити (підрахувати).

4.4.2. Класифікація лічильників

Різноманітність типів лічильників дозволяє будувати їхні схеми на синхронних та асинхронних тригерах RS -; D -; JK -типів, на регістрах, на кінцевих схемах тощо. У цифрових пристроях використовується значне число типів лічильників, які відрізняються цілою низкою параметрів. Найбільш значного поширення набули лічильники, що побудовані на синхронних тригерах різних типів.

Для заданого модуля лічби M кількість тригерів, яка необхідна для побудови лічильників, визначається з умови найближчого більшого цілого числа формулою

$$m = \text{int} [\log_2 M]. \quad (4.7)$$

Залежно від модуля лічби M лічильники бувають *двійковими* (за модулем $M = 2^m$) і з *довільним модулем* ($M \neq 2^m$), в яких число m округлене до більшого цілого числа. За цією класифікаційною ознакою лічильники можуть працювати у двійковому та іншому кодах. Назву лічильникам, як правило, дають за видом кодування його станів, тобто за кодом, в якому працює лічильник.

За цільовим призначенням лічильники поділяють на три типи:

- підсумовувальні лічильники;
- віднімальні лічильники;
- реверсивні лічильники.

У *підсумовувальному лічильнику* з кожним вхідним імпульсом число, що містить лічильник, *зростає на одиницю*, а у *віднімальному* – *зменшується на одиницю*. Отже, підсумовувальний лічильник виконує прямий, а віднімальний – обернений підрахунок числа одиниць, що надійшли до його входу.

Реверсивні лічильники працюють в режимі або прямої, або оберненої лічби.

За способом устанавлення стану розрядів лічильники підрозділяються на *синхронні* та *асинхронні*.

До синхронних лічильників відносяться такі, в яких процес устанавлення будь-якого нового стану розрядів виконується одночасно в усіх розрядах лічильника. В асинхронних лічильниках стани розрядів устанавлюються не одночасно, а послідовно.

За способом утворення сигналів переносу, лічильники поділяються на три групи: з *послідовним*, *паралельним* та *послідовно-паралельним переносом*. За цією ознакою лічильники відрізняються способами подачі вхідних імпульсів.

У послідовному лічильнику вхідні імпульси подаються тільки на вхід першого тригера, а у паралельному – одночасно на синхровходи тригерів усіх розрядів. Різновидом паралельних лічильників є кільцеві лічильники, що будуються на базі реєстрів зсуву.

Послідовно-паралельні лічильники будують за принципом послідовного з'єднання кількох паралельних лічильників.

Одним з основних часових параметрів лічильника є роздільна здатність та час реєстрації. Ці параметри характеризують швидкодію лічильників.

Роздільна здатність лічильників – це мінімальний період надходження вхідних імпульсів, за якого забезпечується надійна робота лічильника.

Час реєстрації – це максимальний часовий інтервал між моментами закінчення подачі вхідного імпульсу та моментом устанавлення числа на виходах розрядів лічильника.

4.4.3. Послідовні лічильники

Послідовні лічильники будуються, як правило, на базі декількох тригерів, кожен з яких працює як лічильник за модулем 2. При цьому послідовне з'єднання тригерів виконується таким чином, що вихід тригера i -го розряду підключається безпосередньо до лічильного входу наступного тригера $i + 1$.

Послідовні лічильники відносяться до класу асинхронних, бо стани розрядів в таких пристроях устанавлюються послідовно після приходу чергового фронту чи зрізу синхроімпульсу C . Послідовні лічильники будуються на базі D - або JK -тригерів.

На рис. 4.24 показана схема послідовного підсумовувального лічильника, розрядні тригери якого виконані на базі синхронних D -тригерів, працюючих у лічильному режимі.

Тут C – лічильний вхід;

R – вхід скиду лічильника;

P – вихід переповнення лічильника.

Лічильний режим роботи тригерів забезпечується з'єднанням інформаційного D -входу з інверсним виходом власного тригера.

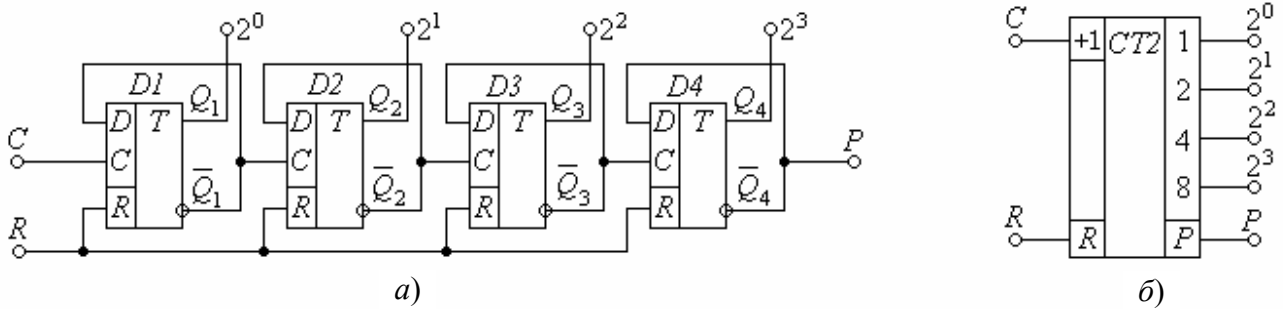


Рисунок 4.24 – Послідовний підсумовувальний лічильник: а) схема; б) умовне позначення

Тригери, що входять до складу лічильників, повинні мати входи скиду R , які з'єднуються між собою, й утворюють вхід скиду лічильника R . Перед початком роботи або за необхідності на вхід скиду подається одиничний імпульс, за допомогою якого всі тригери скидаються, після чого треба забезпечити наявність на цьому вході нульового потенціалу.

Кожний тригер лічильника є двійковим розрядом, який характеризується так званим *ваговим* коефіцієнтом. Ваговий коефіцієнт визначається як 2^{n-1} , де n – порядковий номер тригера (розряду).

Так, перший тригер має ваговий коефіцієнт $2^0 = 1$, другий тригер з ваговим коефіцієнтом $2^1 = 2$, третій тригер має ваговий коефіцієнт $2^2 = 4$, четвертий $2^3 = 8$ і т.д.

Вхід C першого тригера носить назву *підсумовувального входу* лічильника. Принцип дії підсумовувального лічильника ілюструє часова діаграма його роботи, яка показана на рис. 4.25 за умови, що наступний тригер запускається фронтом інверсного виходу \bar{Q} попереднього тригера.

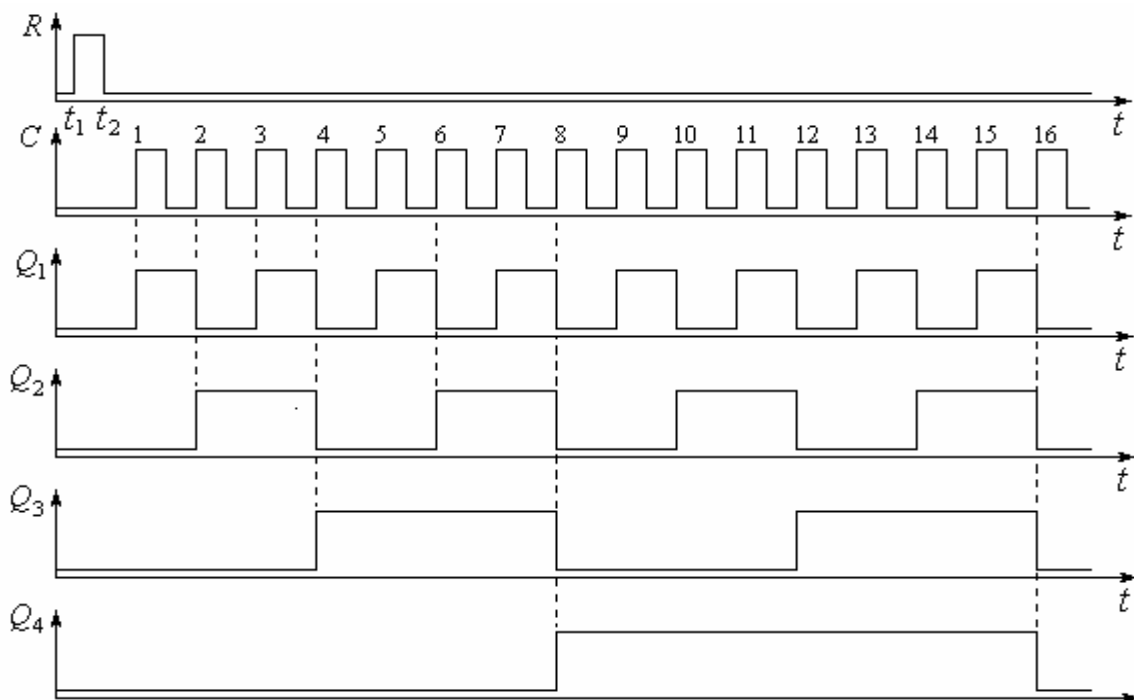


Рисунок 4.25 – Часова діаграма роботи 4-розрядного підсумовувального лічильника
Підсумовувальний лічильник працює наступним чином.

До початку лічби необхідно скинути лічильник, тобто установити всі тригери в нульовий стан. Це здійснюється подаванням на вхід R короткочасного (інтервал моментів $t_1 \dots t_2$) одиничного імпульсу. При цьому всі тригери, незалежно від їхніх попередніх станів, скидаються. Вміст лічильника дорівнює нулю (табл. 4.7)

Лише після скінчення імпульсу скиду t_2 можна починати лічбу, тобто подавати імпульси C .

З приходом першого фронту імпульсу лічби 1 встановлюється перший тригер $D1$, який є самим молодшим розрядом лічильника. На його виході Q_1 з'являється 1, а потенціал на інверсному виході \overline{Q}_1 змінюється з 1 на 0, тобто формується зріз. Тому стан тригера $D2$ не змінюється, бо тригер спрацьовує від фронту. На виходах лічильника Q_1, Q_2, Q_3 і Q_4 фіксується число 0001.

Другий вхідний імпульс своїм фронтом 2 скидає тригер $D1$. На виході \overline{Q}_1 тригера $D1$ з'являється 1, яка є так званим імпульсом переносу. При цьому фронт \overline{Q}_1 встановлює тригер $D2$. На виходах лічильника з'являється число 0010.

Фронт третього імпульсу встановлює перший тригер $D1$, не змінюючи при цьому одиничний стан другого тригера $D2$, тобто вміст лічильника дорівнює числу 0011.

З приходом четвертого імпульсу переходять у стан нуля обидва тригери $D1$ та $D2$, а виникаючий при цьому імпульс переносу \overline{Q}_2 встановлює тригер $D3$. Отже на виходах лічильника з'явиться число 0100.

Заповнення розрядів лічильника з кожним імпульсом буде продовжуватися доти, поки лічильник не відряхує максимальне число 1111 на п'ятнадцятому імпульсі, що надходить на вхід C першого тригера $D1$. При цьому всі тригери є встановленими.

Шістнадцятий імпульс переводить своїм фронтом тригер $D1$ у нульовий стан, а імпульси переносу $\overline{Q}_1, \overline{Q}_2$ та \overline{Q}_3 скидають тригери $D2, D3$ та $D4$, повертаючи лічильник у початковий стан 0000.

При поверненні тригера $D4$ у нульовий стан 16-м імпульсом на виході \overline{Q}_4 виникає так званий імпульс переповнення P , який призначений для встановлення наступного розряду $D5$ (якщо він є) число 10000, тобто 16, що фіксується лічильником.

Таблиця 4.7 – Стани 4-розрядного підсумовувального лічильника

Номер вхідного імпульсу	Виходи				Число в лічильнику
	Q_4	Q_3	Q_2	Q_1	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	1	0	1	0	10
11	1	0	1	1	11
12	1	1	0	0	12
13	1	1	0	1	13
14	1	1	1	0	14
15	1	1	1	1	15
16	0	0	0	0	0

З надходженням кожного вхідного імпульсу на підсумовувальний вхід лічильника його вміст збільшується на одиницю, як показано в таблиці станів (табл. 4.7).

За напрямом лічби лічильники бувають як *підсумовувальними*, так і *віднімальними*. Зміна напрямку лічби на протилежний називається *реверсом* лічби.

На рис. 4.26 показана схема віднімального 4-розрядного двійкового лічильника.

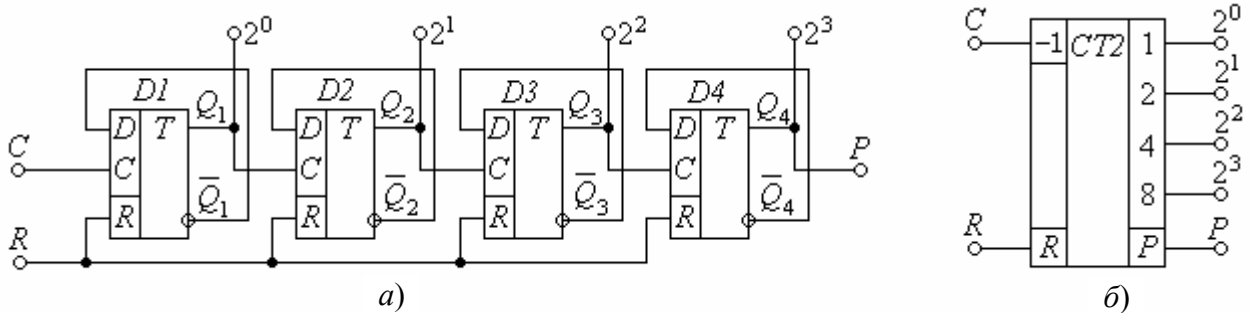


Рисунок 4.26 – Послідовний піднімальний лічильник:
а) схема; б) умовне позначення

Порівнюючи рис. 4.25 з 4.26, переконуємося в тому, що схема віднімального лічильника (рис. 4.26) відрізняється від підсумовувального лише тим, що запуск наступного тригера здійснюється не з інверсного виходу \bar{Q}_i попереднього тригера, а з прямого Q_i .

Принцип дії віднімального лічильника ілюструє часова діаграма його роботи, яка показана на рис. 4.27.

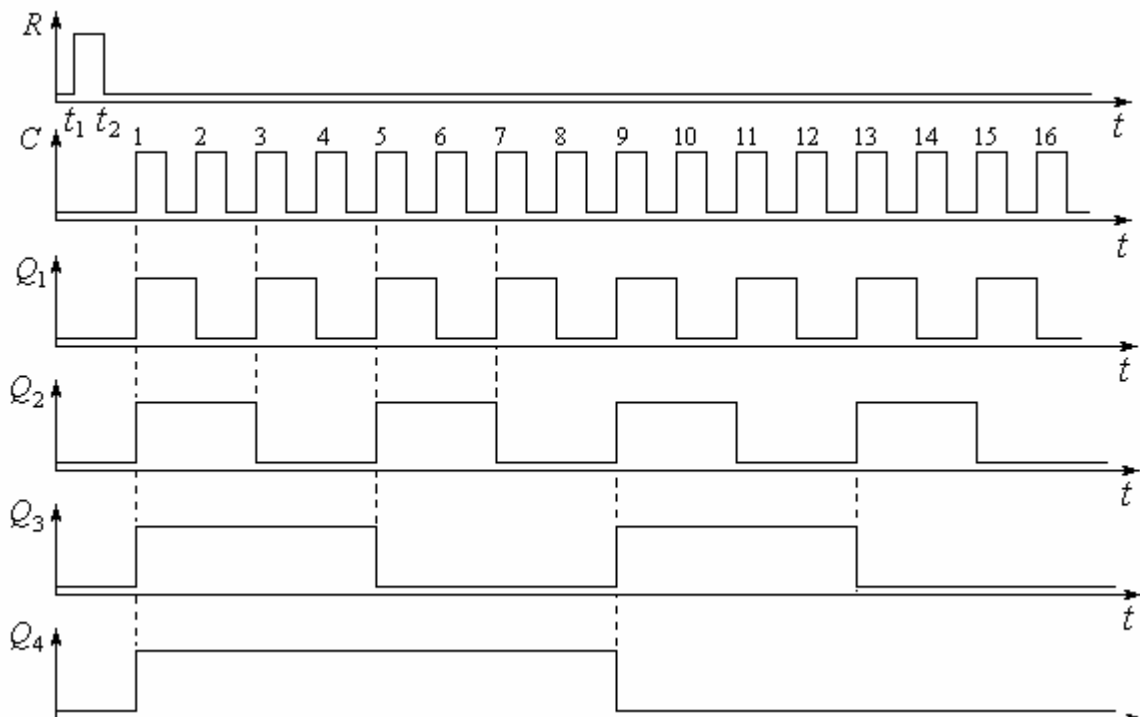


Рисунок 4.27 – Часова діаграма роботи 4-розрядного віднімального лічильника
Віднімальний лічильник працює наступним чином.

До початку лічби необхідно скинути лічильник, тобто установити всі тригери в нульовий стан. Це здійснюється подачею на вхід R короткочасного (інтервал моментів $t_1 \dots t_2$) одиничного імпульсу. При цьому всі тригери, незалежно від їхніх попередніх станів, скидаються.

Лише після скінчення імпульсу скиду t_2 можна починати лічбу, тобто подавати імпульси C .

З приходом першого фронту імпульсу лічби 1 встановлюється перший тригер $D1$, який є самим молодшим розрядом лічильника.

З встановленням тригера $D1$ на його прямому виході Q_1 з'являється фронт, яким встановлюється наступний тригер $D2$. Одночасно фронт виходу Q_2 встановлює тригер $D3$ і в цей самий момент надходження першого імпульсу фронтом Q_3 встановлюється четвертий тригер $D4$.

Таким чином, у віднімальному лічильнику з надходженням першого імпульсу встановлюються всі тригери (у нашому випадку – чотири). Вміст лічильника становить 1111, тобто 15. Так, з урахуванням позичення двійкового числа 10000 виконується операція віднімання двійкових чисел $10000 - 0001 = 1111$, тобто $16 - 1 = 15$.

Походження числа 15 пояснюється математичною дією $16 - 1 = 15$, де 16 – число, яке позичене з наступного розряду з ваговим коефіцієнтом $2^5 = 16$.

Далі з кожним черговим вхідним імпульсом вміст лічильника зменшується на одиницю (табл. 4.8).

Другий вхідний імпульс C скине перший тригер $D1$. Інші тригери не спрацьовують, бо на виході Q_1 зріз. Вміст лічильника становить 1110, тобто $16 - 2 = 14$.

Вхідний імпульс 3 встановить тригер $D1$, фронт виходу Q_1 якого скине тригер $D2$. Інші тригери не спрацьовують, бо на виході Q_2 зріз. У лічильнику залишиться число 1101, тобто $16 - 3 = 13$ і т.д.

З кожним вхідним імпульсом C вміст лічильника зменшується на одиницю.

Після 15 імпульсу C у лічильнику залишиться встановлений лише перший тригер $D1$. Вміст лічильника становить 0001.

16 імпульс скидає тригер $D1$, після чого лічильник обнуляється, а імпульс переповнення P передається в наступний розряд.

Для підвищення завадостійкості послідовних лічильників їх виконують на двотактових тригерах MS -типів.

Таблиця 4.8 – Стани 4-розрядного віднімального лічильника

Номер вхідного імпульсу	Виходи				Число в лічильнику
	Q_4	Q_3	Q_2	Q_1	
0	0	0	0	0	0
1	1	1	1	1	15
2	1	1	1	0	14
3	1	1	0	1	13
4	1	1	0	0	12
5	1	0	1	1	11
6	1	0	1	0	10
7	1	0	0	1	9
8	1	0	0	0	8
9	0	1	1	1	7
10	0	1	1	0	6
11	0	1	0	1	5
12	0	1	0	0	4
13	0	0	1	1	3
14	0	0	1	0	2
15	0	0	0	1	1
16	0	0	0	0	0

На рис. 4.28 показана схема підсумовувального лічильника за модулем 16, який побудований на чотирьох двоступеневих *JK*-тригерах, тобто на тригерах *MS*-типу.

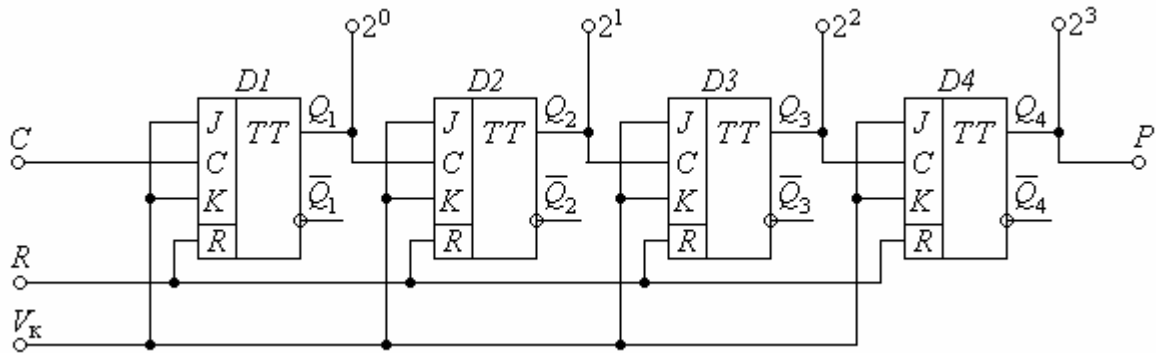


Рисунок 4.28 – Підсумовувальний лічильник на *JK*-тригерах

Тут *C* – вхід лічильника; *R* – вхід скиду лічильника;

V_k – керуючий вхід; *P* – вихід переповнення.

Якщо на керуючому вході $V_k = 0$, то вхід *C* лічильника запертий, через що лічильник знаходиться в режимі зберігання інформації і вхідні імпульси *C* не сприймає.

Коли ж $V_k = 1$, то вхід *C* лічильника відкритий для вхідних імпульсів *C*, які треба лічити.

Роботу лічильника пояснюють часова діаграма (рис. 4.29) і таблиця станів (див. табл. 4.7).

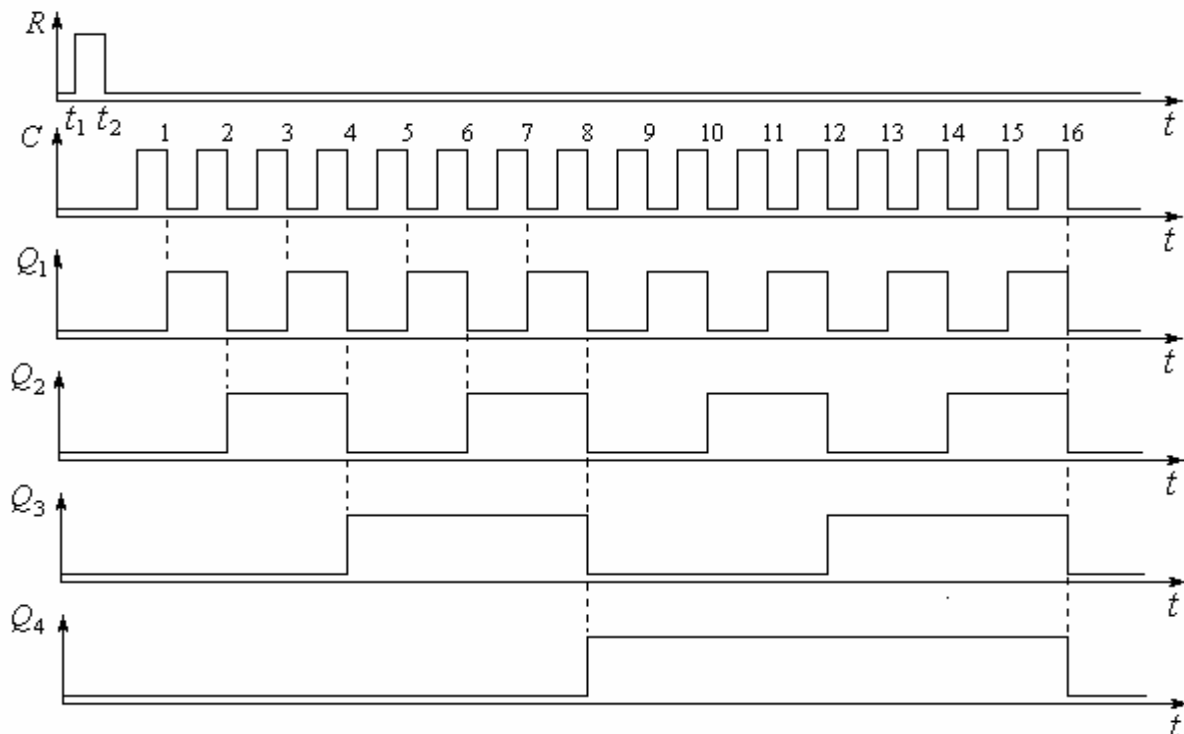


Рисунок 4.29 – Часова діаграма роботи підсумовувального лічильника на *JK*-тригерах

Діаграма роботи побудована за тих умов, що двоступеневий тригер *MS*-типу встановлюється в одиничний стан у два етапи, тобто з приходом фронту синхроімпульсу встановлюється його ведучий тригер *M*, а від зрізу того ж

імпульсу спрацьовує ведений тригер S . Тому в цілому MS -тригер запускається від зрізу.

Лічильник працює наступним чином.

У нульовий стан, в якому $Q_1 = 0$, $Q_2 = 0$, $Q_3 = 0$, $Q_4 = 0$ лічильник скидається по входу R (інтервал моментів $t_1 \dots t_2$).

З приходом фронту імпульсу C вихідний рівень Q_1 тригера $D1$ не змінюється, бо фронтом запускається (встановлюється або скидається) лише ведучий тригер M , а ведений тригер S спрацьовує теж від фронту, але він надходить лише після закінчення імпульсу, тобто в цілому перший тригер, як і решта тригерів, запускається від зрізу. Тому надалі розглядається запуск будь-якого тригера від зрізу на його вході C .

Після закінчення першого імпульсу, тобто з надходженням його зрізу 1, на виході $Q_1 = 1$, тобто фронт. Тому тригер $D2$ залишається в попередньому стані. Вміст лічильника становить 0001, тобто 1. Так фіксується надходження на вхід лічильника одного імпульсу.

З надходженням імпульсу 2 зрізом C скидається перший тригер $D1$. На виході Q_1 є зріз, через що встановлюється тригер $D2$. Вміст лічильника становить 0010, тобто 2. Так фіксується надходження на вхід лічильника двох імпульсів і т.д. З кожним зрізом на вході C лічильника його вміст збільшується на одиницю.

Після надходження зрізу 15 встановлені всі тригери. Вміст лічильника дорівнює 1111, тобто 15.

Зріз 16 скидає перший тригер $D1$. На виході Q_1 зріз, через що скидається тригер $D2$. Зрізом Q_2 скидається тригер $D3$, а виходом Q_3 скидається тригер $D4$. Так 16 імпульс скидає всі чотири тригери і через вихід Q_4 передає одиницю P до наступного розряду $2^5 = 16$. З урахуванням розряду 2^5 число в лічильнику дорівнює 10000, тобто 16. Якщо тригера п'ятого розряду немає, то число в лічильнику після 16 імпульсу становить 0000. Так фіксується надходження на вхід лічильника 16 імпульсів.

Послідовні *віднімальні* лічильники будуються за такими самими схемами, але при з'єднанні розрядів використовуються інверсні виходи попередніх тригерів (рис. 4.30).

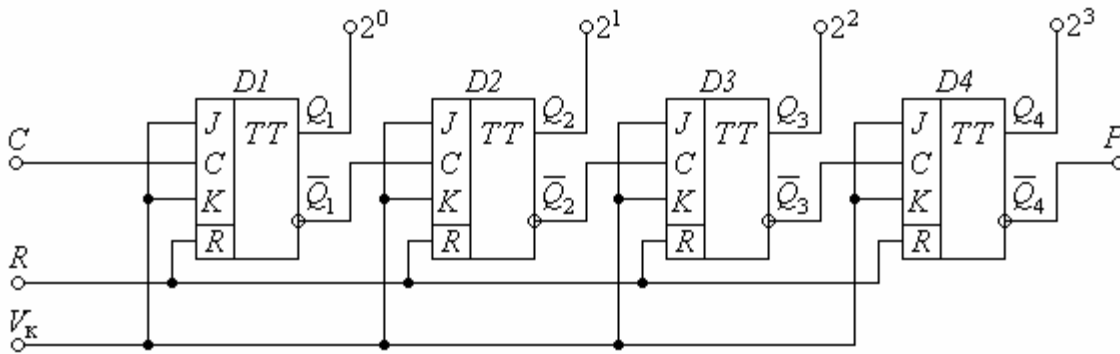


Рисунок 4.30 – Віднімальний лічильник на JK -тригерах

За такої комутації виходів імпульси переносу від розряду до розряду будуть утворюватися при переході відповідального тригера зі стану 0 у стан 1.

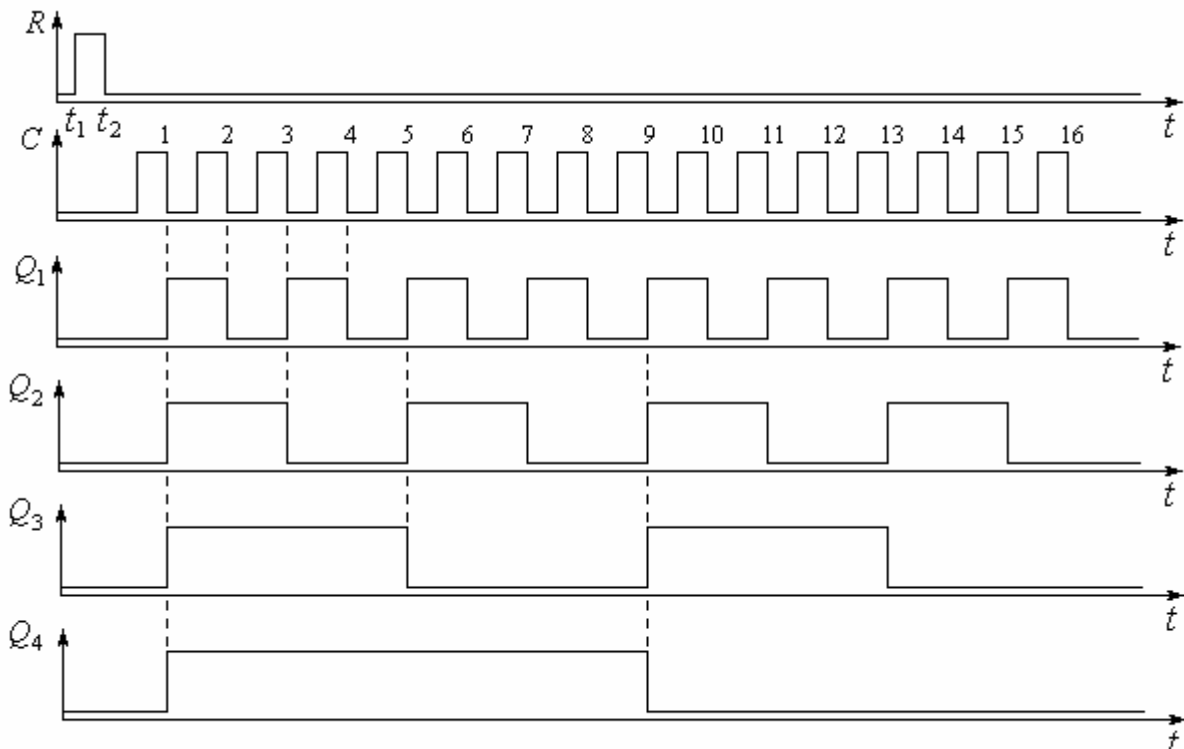


Рисунок 4.31 – Часова діаграма роботи віднімального лічильника на JK -тригерах

Роботу віднімального послідовного лічильника ілюструють часова діаграма, яка показана на рис. 4.31, та таблиця станів (див. табл. 4.7).

Початковий стан віднімального лічильника визначається вихідним числом 0000, яке забезпечується скидом лічильника в нульовий стан по входу R в інтервалі моментів $t_1 \dots t_2$.

З приходом першого імпульсу на вхід C лічильника встановлюється перший тригер $D1$. На його виході $Q_1 = 1$, тобто фронт. Тоді на виході \bar{Q}_1 зріз, яким встановлюється другий тригер $D2$.

При цьому зрізом \bar{Q}_2 встановлюється третій тригер $D3$, а зрізом \bar{Q}_3 встановлюється четвертий тригер $D4$.

Таким чином, після надходження першого імпульсу на вхід C віднімального лічильника встановлюються всі його тригери. При цьому всі прямі виходи тригерів, Q_1, Q_2, Q_3, Q_4 будуть одиничними.

Так, з урахуванням позички 16 виконується операція віднімання двійкових чисел $10000 - 0001 = 1111$, тобто $16 - 1 = 15$.

Другий імпульс на вході C скине перший тригер $D1$, на інверсному виході \overline{Q}_1 якого створюється фронт, яким другий тригер $D2$ не запускається. Тому всі тригери, крім першого, залишаються в одиничному стані: $Q_1 = 0; Q_2 = 1; Q_3 = 1; Q_4 = 1$. Так виконується операція віднімання двійкових чисел $1111 - 0001 = 1110$, тобто $16 - 2 = 14$ і т.д.

З кожним вхідним імпульсом вміст лічильника зменшується на одиницю, тобто здійснюється віднімання.

Після п'ятнадцятого імпульсу стан лічильника буде $Q_1 = 1; Q_2 = 0; Q_3 = 0; Q_4 = 0$ і вміст лічильника дорівнюватиме 0001.

Шістнадцятий імпульс скидає перший тригер $D1$, повертаючи лічильник у початковий стан: $Q_1 = 0; Q_2 = 0; Q_3 = 0; Q_4 = 0$.

Таким чином, на виходах віднімального лічильника буде фіксуватися число в оберненому коді, як показано в таблиці станів (табл. 4.7).

Якщо треба відзначити максимальне число станів, то використовується назва лічильника за модулем $M = 2^4 = 16$.

4.4.4. Швидкодія лічильників

Швидкодія лічильника визначається частотою слідкування вхідних імпульсів

$$f_{\text{сл}} = \frac{1}{\tau_i + t_{\text{уст}}}, \quad (4.8)$$

де τ_i – тривалість лічильного імпульсу;

$t_{\text{уст}}$ – час установалення числа в лічильнику.

Для схеми лічильника, що має n розрядів, час установалення числа визначається як

$$t_{\text{уст max}} = n \cdot \tau_t, \quad (4.9)$$

де τ_t – тривалість перехідних процесів.

Після підстановки (4.9) у (4.8) отримуємо формулу, яка визначає максимальну частоту слідкування вхідних імпульсів без урахування часу на зчитування числа з його виходів

$$f_{\text{сл max}} = \frac{1}{\tau_{i \text{ min}} + n \tau_t}. \quad (4.10)$$

Якщо час на зчитування числа визначити, як $\tau_{\text{зч}}$, то формула (4.10) буде мати закінчений вигляд

$$f_{\text{сл max}} = \frac{1}{\tau_{i \text{ min}} + n \tau_t + \tau_{\text{зч}}}. \quad (4.11)$$

Послідовні лічильники використовуються також як *подільники частоти*.

Дійсно, з часової діаграми роботи (рис. 4.31), видно, що на виході першого тригера Q_1 лічильника частота вхідних імпульсів C поділяється на 2, на виході другого тригера Q_2 імпульси надходять з частотою, яка у чотири рази менше за вхідні C . Кожний четвертий вхідний імпульс змінює стан виходу третього тригера Q_3 , тобто частота слідування імпульсів на виході Q_3 у вісім разів нижча за вхідну C . З виходу Q_4 останнього тригера знімаються імпульси, частота яких у 16 разів менша за вхідну C . Якщо лічильник використовується як подільник частоти, то сигнал знімається лише з одного виходу.

Основним параметром подільника частоти є коефіцієнт поділення 2^n , де n – порядковий номер тригера.

Максимальний коефіцієнт поділення частоти дорівнює модулю лічби $M = 2^m$ і вихід такого подільника береться зі старшого розряду лічильника.

Гранична частота слідування лічильних імпульсів у послідовних лічильниках визначається максимальною частотою переключення тригера першого розряду за формулою

$$f_{\text{п max}} = \frac{1}{\tau_{\text{ч}} + \tau_{\text{т}}}. \quad (4.12)$$

Розглянуті асинхронні послідовні лічильники мають загальний недолік – значний час реєстрації підрахованої кількості вхідних імпульсів, тобто мають відносно малу швидкодію. Цей недолік зумовлений втратою часу на послідовне формування імпульсу переносу в кожному розряді лічильника. У найбільш несприятливому випадку перенос, що виник у молодшому розряді, викличе по черзі переноси в усіх інших розрядах лічильника і час установалення числа при цьому дорівнюватиме $n\tau_{\text{т}}$.

Зменшення часу установалення числа, тобто реєстрації вмісту лічильника, досягається при застосуванні паралельних лічильників або лічильників з паралельним переносом біта з молодшого розряду в старші.

4.4.5. Паралельні лічильники

Паралельні – це синхронні лічильники, в яких лічильні імпульси надходять одночасно на синхровходи C усіх тригерів.

Перевагою паралельних лічильників є підвищена швидкодія.

Синхронні лічильники будуються, як правило, на базі двоступеневих багатовходових RS -, D - та JK -тригерів. Число інформаційних входів тригерів паралельних лічильників за модулем $M = 2^m$ дорівнює $(m - 1)$. Так, для чотирирозрядного двійкового лічильника кожен тригер повинен мати потроєні J - і K -входи.

Схема паралельного лічильника за модулем $M = 2^4 = 16$ на синхронних двоступеневих JK -тригерах з потроєними входами J і K показана на рис. 4.32.

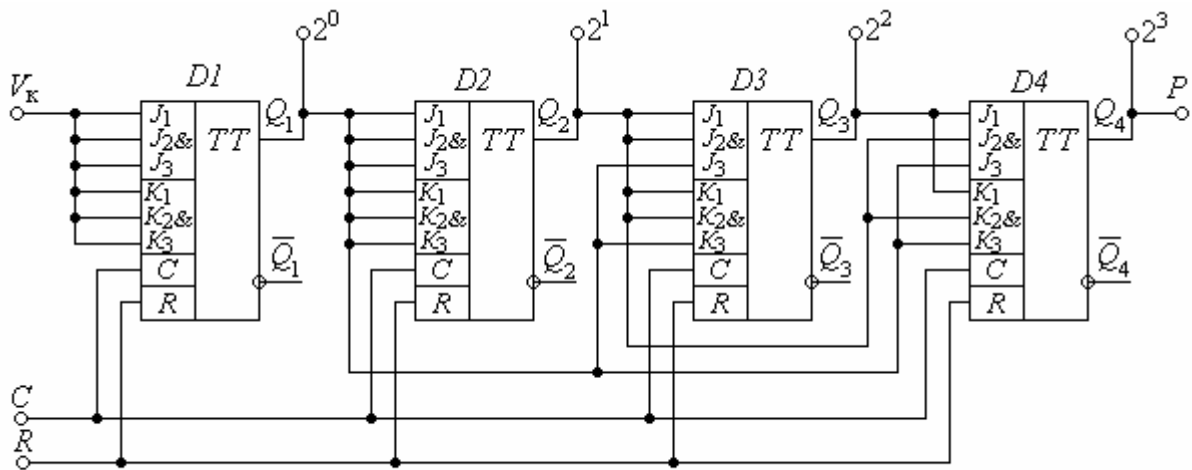


Рисунок 4.32 – Схема паралельного лічильника за модулем 16

Тут C – вхід лічильника;
 R – вхід скиду лічильника;
 V_k – керуючий вхід;
 P – вихід переповнення.

Якщо на керуючому вході $V_k = 0$, то вхід C лічильника запертий, через що лічильник перебуває в режимі зберігання інформації і вхідні імпульси C не сприймає.

Коли ж $V_k = 1$, то вхід C лічильника відкритий для вхідних імпульсів C , які треба лічити.

Схема працює наступним чином.

З приходом першого лічильного імпульсу одночасно на всі входи C чотирьох тригерів встановлюється лише перший тригер $D1$, бо входи J і K тригерів $D2$, $D3$ і $D4$ мають нульові потенціали. Вміст лічильника дорівнює числу 0001.

Одиничний потенціал з виходу Q_1 надходить до інформаційних входів J і K другого тригера $D2$, дозволяючи йому змінити свій стан з приходом наступного імпульсу C .

Другий лічильний імпульс перекине тригер $D2$ в одиничний стан і одночасно скине перший тригер $D1$. Тому вміст лічильника буде дорівнювати двійковому числу 0010. При цьому третій $D3$ і четвертий $D4$ тригери свій стан не змінюють.

Лише після третього лічильного імпульсу, коли в лічильнику встановлюється число 0011, тобто $Q_1 = 1$; $Q_2 = 1$, дозвіл на зміну стана отримує третій тригер $D3$. Цей дозвіл надходить до його входів J і K з виходів Q_1 та Q_2 .

Четвертий імпульс C встановить третій тригер $D3$ і скине тригери $D1$ і $D2$.

Після сьомого імпульсу число в лічильнику зміниться на 0111, тобто $Q_1 = 1$; $Q_2 = 1$; $Q_3 = 1$. Через це четвертий тригер $D4$ отримує дозвіл на встановлення.

Восьмий імпульс C встановлює четвертий тригер у стан 1 і скидає в нуль перші три тригери. При цьому число в лічильнику дорівнюватиме 1000.

Подальші спрацьовування лічильника будуть відбуватися з кожним імпульсом C , збільшуючи число на одиницю доти, поки лічильник не відрахує максимальне число 1111.

Шістнадцятий імпульс переводить своїм зрізом усі тригери в стан 0, повертаючи лічильник у початковий стан 0000. Часова діаграма підсумовувального паралельного лічильника не відрізняється від часової діаграми послідовного лічильника, яка показана на рис. 4.29.

Для побудови віднімального паралельного лічильника досить у розглянутій схемі (рис. 4.32) змінити виходи в тригерах усіх розрядів з прямих Q_i на інверсні \overline{Q}_i .

Швидкодія роботи паралельних лічильників значно вища, ніж послідовних. Час установлення числа дорівнює тривалості переключення одного тригера

$$t_{\text{уст max}} = \tau_t. \quad (4.13)$$

З урахуванням (4.13) максимальна частота слідкування вхідних імпульсів визначається формулою

$$f_{\text{сл max}} = \frac{1}{\tau_i + \tau_t}, \quad (4.14)$$

де τ_i – тривалість імпульсу.

Недоліками паралельних лічильників є наступні:

– застосування тригерів з великим числом інформаційних входів зумовлює значну складність, що приводить до зниження надійності;

– суттєва залежність навантажувальної здатності розрядних тригерів по виходах від числа розрядів лічильника; найбільш навантаженим є перший тригер $D1$.

Тому розрядність паралельних лічильників не перевершує чотирьох.

Для отримання великих значень модуля лічби M застосовують каскадне з'єднання лічильників. Для цього їх розбивають на групи. Такі лічильники носять назву лічильників з груповим переносом або паралельно-послідовних лічильників. Вони швидкодіючі, але при цьому не потребують багаточисленних інформаційних входів і тому отримали значне розповсюдження.

4.4.6. Паралельно-послідовні лічильники

У *паралельно-послідовних лічильниках* всі розряди розбивають на групи і використовують два види зв'язків між розрядами; у середині групи – паралельний зв'язок, а між групами – послідовний. За допомогою такого з'єднання розрядів забезпечується одночасне переключення розрядів у групах і послідовне переключення між групами.

На рис. 4.33 показана схема 4-розрядного паралельно-послідовного лічильника на синхронних MS -тригерах JK -типу. Лічильник поділений на дві групи по два розряди в кожній: перша група на тригерах $D1, D2$ та друга – на тригерах $D3, D4$.

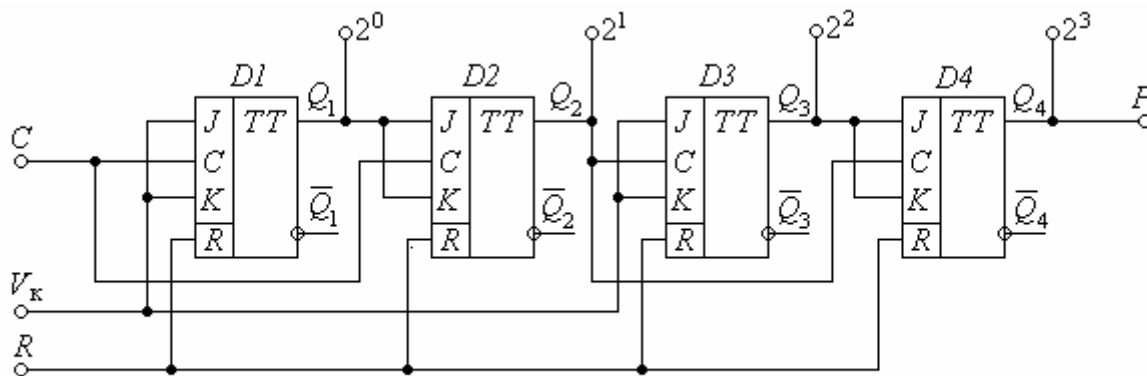


Рисунок 4.33 – Схема паралельно-последовного лічильника за модулем 16

Тут C – вхід лічильника;

R – вхід скиду лічильника;

V_k – керуючий вхід;

P – вихід переповнення, яким передається двійкове число 10000, тобто 16 до наступного розряду.

Якщо на керуючому вході $V_k = 0$, то вхід C лічильника запертий, через що лічильник перебуває в режимі зберігання інформації і вхідні імпульси C не сприймає.

Коли ж $V_k = 1$, то вхід C лічильника відкритий для вхідних імпульсів C , які треба лічити.

Процес переключення розрядів у групі відповідає послідовності роботи розрядів паралельних лічильників, а процес спрацьовування груп відповідає порядку переключення лічильників послідовного типу.

Швидкодія паралельно-последовного лічильника визначається максимальною частотою слідування імпульсів

$$f_{\text{слmax}} = \frac{1}{\tau_u + t_{\text{уст}}} = \frac{1}{\tau_c + n_{\text{гр}} \cdot \tau_t}, \quad (4.15)$$

де $n_{\text{гр}}$ – число груп у лічильнику.

Швидкодія паралельно-последовних лічильників декілька менша за паралельні, але набагато більша за послідовні лічильники.

Як видно з рис. 4.33, не зважаючи на підвищення швидкодії, паралельно-последовні лічильники не потребують потроєння інформаційних входів, як паралельні, через що забезпечують достатню надійність.

4.4.7. Реверсивні лічильники

Реверсом лічильників називається зміна напрямку лічби. Реверс здійснюється перемиканням прямих та інверсних виходів попереднього тригера, з якого запускається наступний.

Реверсивними є лічильники, які можуть виконувати як додавання, так і віднімання імпульсів, що надходять на вхід C . Схема керування реверсивним лічильником дозволяє класифікувати їх за двома ознаками:

- з одним лічильним і двома керуючими входами;
- з двома лічильними входами.

Реверсивні лічильники бувають двох типів.

Реверсивні лічильники *першого типу* мають керуючі входи і виконують додавання або віднімання імпульсів у залежності від активного сигналу на його керуючих входах. Коли одиничний рівень на вході дозволу додавання, то лічильник підсумовує імпульси, тобто збільшує на одиницю свій вміст з приходом кожного синхроімпульсу.

При подачі високого рівня на *вхід дозволу віднімання* лічильник віднімає одиницю, змінюючи свій вміст на одиницю з кожним синхроімпульсом на вході лічби *C*.

У реверсивних лічильниках *другого типу* використовуються два входи синхросигналу: по одному з них (+1) подаються імпульси для підсумовування, а по іншому (-1) – для віднімання. Для таких лічильників не треба створювати додаткові кола керування.

На рис. 4.34 показана схема реверсивного лічильника першого типу.

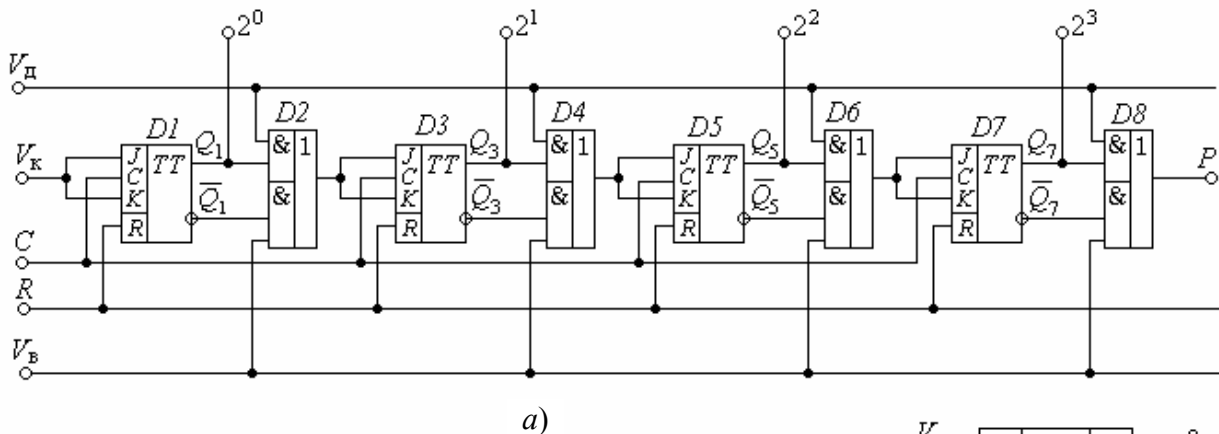
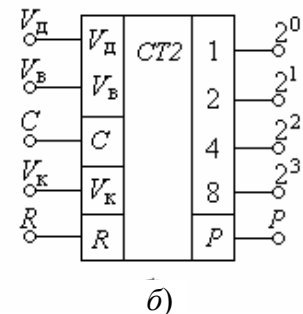


Рисунок 4.34 – Реверсивний лічильник з двома керуючими входами: а) схема; б) умовне позначення



Тут *C* – лічильний вхід;

R – вхід скиду;

V_д – вхід дозволу додавання;

V_в – вхід дозволу віднімання;

V_к – керуючий вхід;

P – вихід переповнення.

Якщо на керуючому вході $V_k = 0$, то вхід *C* лічильника запертий, через що лічильник знаходиться в режимі зберігання інформації і вхідні імпульси *C* не сприймає.

Коли ж $V_k = 1$, то вхід *C* лічильника відкритий для вхідних імпульсів *C*, які треба лічити.

Вхід *R* короткочасним імпульсом $R = 1$ скидає лічильник у нуль.

Режим входів дозволу $V_d = 1$ та $V_b = 0$ забезпечує режим додавання, в якому з кожним вхідним імпульсом C вміст лічильника збільшується на одиницю.

Режим $V_d = 0$ та $V_b = 1$ забезпечує режим віднімання, в якому з кожним вхідним імпульсом C вміст лічильника зменшується на одиницю.

Реверс лічильника здійснюється перемиканням прямих Q_i або інверсних \overline{Q}_i виходів за допомогою чотирьох логічних схем 2І-2АБО: $D2; D4; D6; D8$.

Реверсування досягається передачею або сигналу переносу з прямих виходів Q_i , або сигналу позички з інверсних виходів \overline{Q}_i тригерів.

Керуючий вхід $V_k = 1$ забезпечує лічильний режим лічильника, а $V_k = 0$ закриває вхід лічильника, чим здійснює зберігання попередньої інформації.

Лічильник працює наступним чином.

Режим додавання устанавлюється подаванням високого рівня логічної "1" на вхід дозволу додавання та низького рівня – на вхід дозволу віднімання ($V_d = 1, V_b = 0$). При цьому нижні кон'юнктори елементів $D2, D4, D6, D8$ закриті і не пропускають сигнали позички з *інверсних* виходів \overline{Q}_i , забороняючи саме тим режим *віднімання*. Верхні ж кон'юнктори елементів $D2, D4, D6, D8$ відкриті і пропускають сигнали переносу з *прямих* виходів Q_i до лічильних входів C тригерів, створюючи саме тим режим *додавання*.

Вихід переповнення P здійснює в режимі додавання передачу в наступний розряд числа 10000, тобто 16, після 16 імпульсу C та позичку цього числа в режимі віднімання після першого імпульсу C .

Режим віднімання устанавлюється подачею високого рівня логічної "1" на вхід дозволу віднімання та низького рівня – на вхід дозволу додавання ($V_b = 1; V_d = 0$). При цьому верхні кон'юнктори елементів $D2, D4, D6, D8$ закриті і не пропускають сигнали переносу з *прямих* виходів Q_i , забороняючи тим самим режим *додавання*. Нижні ж кон'юнктори елементів $D2, D4, D6, D8$ відкриті і пропускають сигнали позички з *інверсних* виходів \overline{Q}_i до лічильних входів C тригерів, створюючи саме тим режим *віднімання*.

Примітка. Одночасна подача одиниць на входи V_d та V_b заборонена.

Схема реверсивного лічильника з двома входами (синхровходами) та одним керуючим входом показана на рис. 4.35.

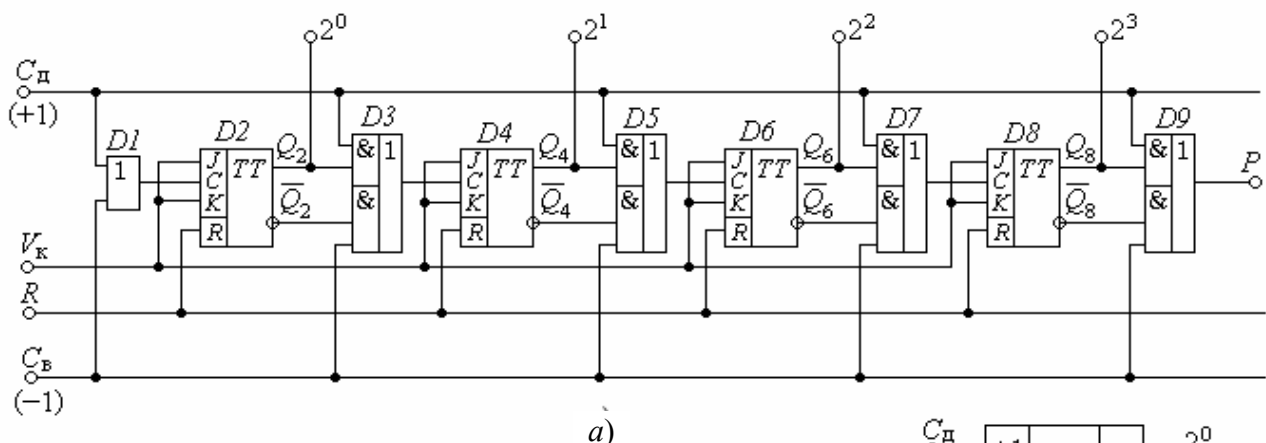
Тут C_d – шина додавання;

C_b – шина віднімання;

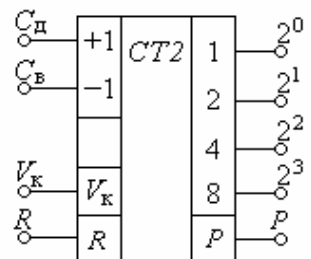
V_k – керуючий вхід;

R – вхід скиду;

P – вихід переповнення.



а)



б)

Рисунок 4.35 – Реверсивний лічильник з двома синхровходами: а) схема; б) умовне позначення

Шини C_d та C_b створюють синхровходи, до яких надходять вхідні імпульси. У режимі додавання імпульси слід подавати на шину C_d , а в режимі віднімання – на шину C_b . При цьому вільна шина повинна мати нульовий рівень.

Лічильник працює наступним чином.

Якщо на керуючому вході $V_k = 0$, то вхід C лічильника запертий, через що лічильник знаходиться в режимі зберігання інформації і вхідні імпульси C не сприймає.

Коли ж $V_k = 1$, то вхід C лічильника відкритий для вхідних імпульсів C , які треба лічити.

Реверс здійснюється за допомогою п'яти логічних схем 2І-2АБО: D_1, D_3, D_5, D_7, D_9 .

Тригери D_2, D_4, D_6, D_8 надають відповідно схемам D_3, D_5, D_7, D_9 дозвіл на проходження імпульсів з шин або додавання C_d , або – віднімання C_b .

Для здійснення режиму додавання необхідно на шину C_b подати нульовий рівень, а до шини C_d підвести імпульси, які треба лічити.

При $C_b = 0$ нижні кон'юнктори схем D_3, D_5, D_7, D_9 будуть закриті і тому не пропустять сигнали позички з інверсних виходів $\overline{Q_2}, \overline{Q_4}, \overline{Q_6}, \overline{Q_8}$. Верхні ж кон'юнктори схем D_3, D_5, D_7, D_9 будуть пропускати імпульси до входів C , якщо попередній тригер перебуває в одиничному стані.

У решті схема працює так само, як і звичайний двійковий підсумовувальний лічильник.

Для здійснення режиму віднімання необхідно на шину C_d подати нульовий рівень, а до шини C_b підвести імпульси, які треба лічити.

У решті схема працює так само, як і звичайний двійковий віднімальний лічильник.

Крім розглянутих схемних рішень можуть бути й інші. У схемотехніці використовується досить значна кількість різноманітних схем, на яких реалізовані реверсивні лічильники.

4.4.8. Лічильники з довільним модулем лічби

У цифровій схемотехніці дуже часто виникає необхідність будувати лічильники з довільним модулем лічби $M \neq 2^m$. Так, наприклад, до таких лічильників відносяться досить поширені десяткові, де $M = 10$. Такі лічильники використовуються частіше у тих випадках, коли з машиною спілкується людина, бо людина звикла до десяткової системи числення.

Розрядність лічильника за модулем $M \neq 2^m$ визначається за умови:

$$2^{m-1} < M < 2^m. \quad (4.16)$$

Так, для десяткового лічильника з модулем лічби $M = 10$ потрібно $m = 4$ тригера, бо $2^3 < 10 < 2^4$. Отже, лічильник з модулем $M = 10$ має чотири розряди і при появі на його виходах числа $1010_2 = 10_{10}$ він повинен скидається в нуль.

Такий лічильник має невикористані надлишкові стани. Дійсно, для двійкового чотирирозрядного лічильника $M = 2^4 = 16$. Тоді число надлишкових станів визначається як $n = 16 - 10 = 6$.

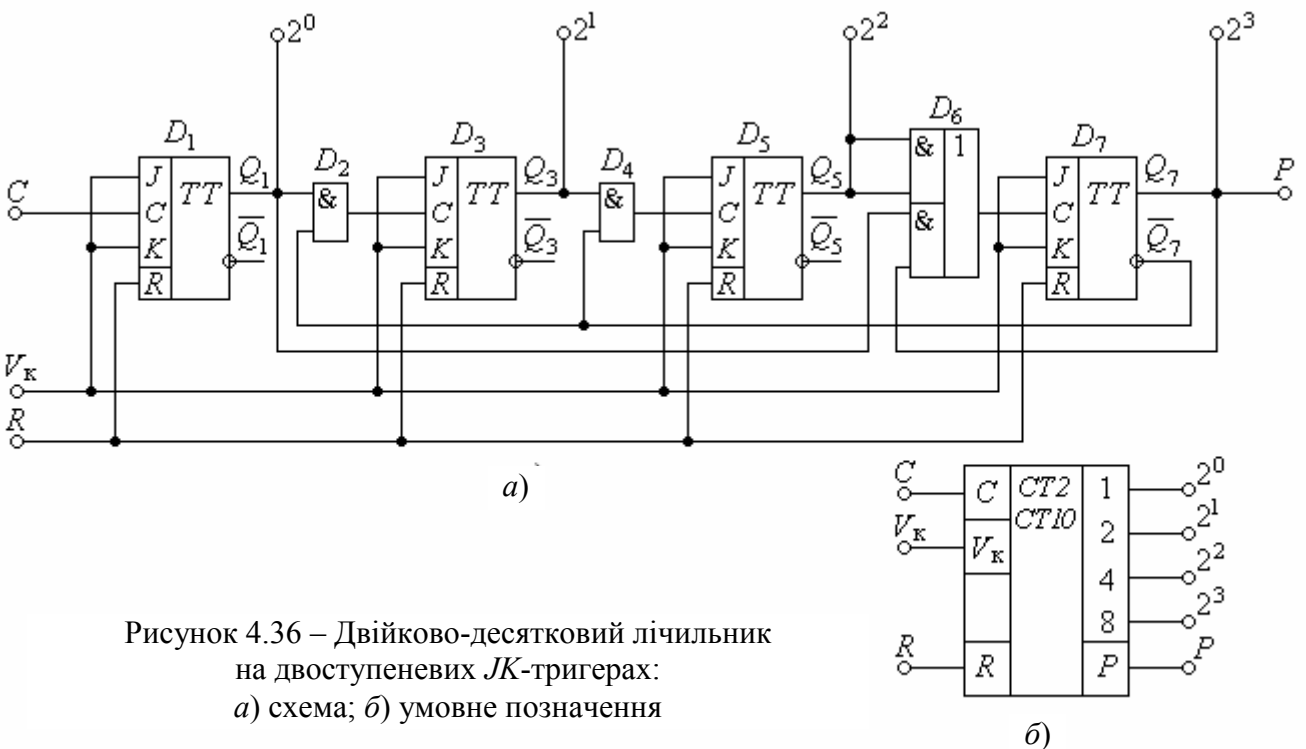


Рисунок 4.36 – Двійково-десятковий лічильник на двоступеневих *JK*-тригерах:
а) схема; б) умовне позначення

Щоб позбутися цих надлишкових для десяткового лічильника станів, застосовують зворотні зв'язки між виходами останнього тригера лічильника і входами тригерів тих розрядів, які у двійковому коді складають число надлишкових станів, тобто у нашому прикладі $n = 6_{10} = 0110_2$. Отже, сигнал зворотного зв'язку слід подавати на відповідні входи тригерів другого і третього розрядів. У синхронних лічильниках такий принцип побудови внутрішніх зв'язків називається *блокуванням переносу*.

На рис. 4.36 показана схема асинхронного (послідовного) двійково-десятькового підсумовувального лічильника, який побудований на синхронних двотактових *JK*-тригерах.

Тут зворотний зв'язок здійснюють логічні елементи *D2, D4, D6*. Елементи *D2* та *D4* забороняють устанавлення тригерів *D3* та *D5* після восьмого імпульсу, а *D6* забороняє встановлення тригера *D7* до восьмого імпульсу.

Схема працює наступним чином.

Як видно з часової діаграми (рис. 4.37) і таблиці станів (табл. 4.9), підрахунок числа імпульсів до восьмого включно виконується в двійковому коді: 0000, 0001, 0010, ... 1000.

Дійсно, доки тригер старшого розряду *D7* знаходиться у нульовому стані, на його виході $\overline{Q_7} = 1$. Це дає дозвіл на проходження імпульсів через логічні схеми 2І *D2*

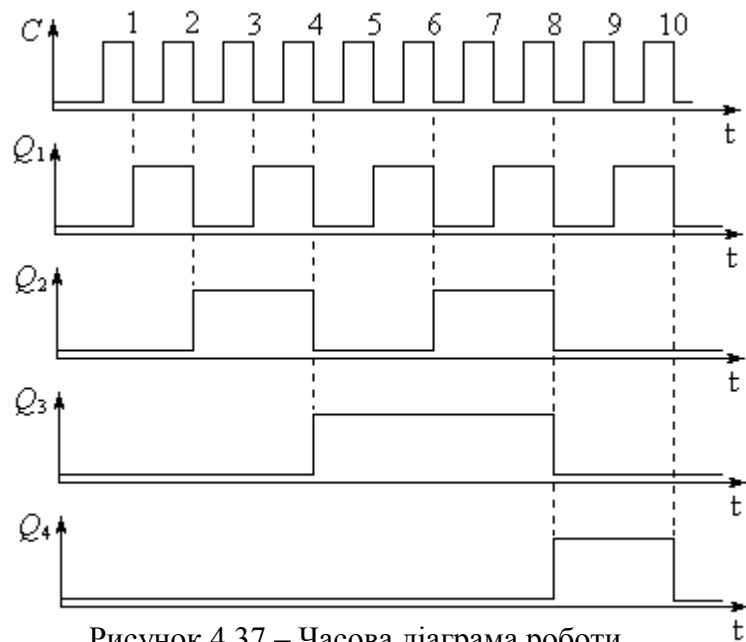


Рисунок 4.37 – Часова діаграма роботи двійково-десятькового лічильника

Таблиця 4.9 – Стани тригерів двійково-десятькового лічильника

Число вхідних імпульсів	Виходи				Число в лічильнику
	Q_4	Q_3	Q_2	Q_1	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9
10	0	0	0	0	0

та *D4* з тригерів *D1* та *D3* відповідно на входи *C* тригерів *D3* та *D5*.

Після надходження восьмого імпульсу повертаються в нульовий стан тригери перших трьох розрядів *D1, D3* та *D5* і встановлюється тригер *D7*.

При встановленому тригері *D7* спрацьовування логічних елементів *D2* та *D4* заборонене подачею з виходу $\overline{Q_7}$ логічного нуля, але з'являється дозвіл на спрацьовування нижнього логічного елемента 2І-АБО *D6*.

Після дев'ятого імпульсу встановлюється тригер *D1*, який готує до проходження імпульсу скиду через схему *D6* на тригер *D7*.

Десятий лічильний імпульс скидає перший тригер *D1*, який скидає тригер *D7* у нульовий стан. При цьому тригери *D3* і *D5* свій стан не змінюють і таким чином усі тригери лічильника перебувають в нульовому стані.

Імпульс переповнення лічильника з'являється після кожного десятого вхідного синхроімпульсу.

З приходом одинадцятого імпульсу лічильник починає працювати з початку.

Роботу послідовного підсумовувального двійково-десятькового лічильника можна пояснити за допомогою станів тригерів (табл. 4.9).

У схемах лічильників передбачено вхід скиду R , завдяки якому можна скинути усі тригери в нульовий стан.

Принципів побудови лічильників з довільним модулем лічби багато. Але усі вони основані на вилученні надлишкових станів двійкового лічильника.

4.4.9. Збільшення розрядності лічильників. Подільники частоти

На практиці часто виникає необхідність використання модуля лічби понад 64. Це необхідно для лічильників великої місткості та подільників частоти з великим коефіцієнтом ділення.

Дійсно, лічильник з модулем лічби M одночасно є подільником частоти з таким самим коефіцієнтом ділення M тому, що на виході останнього тригера лічильника сигнал буде мати частоту в M разів меншу за вхідну.

Проте промисловість випускає мікросхеми різноманітних лічильників, але з обмеженим модулем лічби. Їхній максимальний модуль лічби не перевищує $M = 64$.

Для збільшення модуля лічби стандартні лічильники з'єднують каскадно.

Для каскадного з'єднання двох або більшого числа лічильників треба вибрати такі стандартні мікросхеми, які мають виходи переповнення P .

Коли на вхід лічильника надійде кількість імпульсів, яка дорівнює M , то на виході P з'являється імпульс переповнення, який передається в наступний лічильник. Так модуль лічби n каскадно з'єднаних лічильників дорівнює M^n .

На рис. 4.38 показане каскадне з'єднання двох лічильників $D1$ і $D2$ з модулем лічби $M = 16$ кожного. Спільний модуль лічби становить $M_c = M^2 = 16^2 = 256$.

Схема працює наступним чином.

На п'ятнадцятому імпульсі, що подається на вхід C першого лічильника $D1$, в ньому буде число 1111.

Шістнадцятий імпульс повертає всі розряди $D1$ до нульового стану і формує імпульс переповнення на виході P , який використовується як вхідний імпульс другого лічильника $D2$. У лічильнику $D2$ буде число $1111 + 0001 = 10000_2 = 16_{10}$.

Таким чином, вхідним синхроімпульсом для спрацьовування другого лічильника є кожний шістнадцятий вхідний імпульс.

Каскадне з'єднання також може бути подільником частоти. При цьому метою є не тільки збільшення модуля лічби. Наприклад, в цифрових

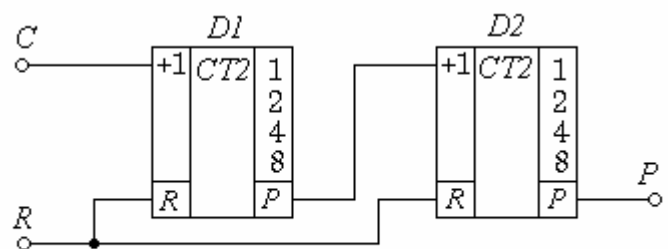


Рисунок 4.38 – Каскадне з'єднання лічильників

годинниках основним елементом є подільник частоти на 60. Для реалізації такого подільника потрібно послідовно з'єднати лічильник за модулем 10 та лічильник за модулем 6. Так виникає потреба в лічильниках з довільним модулем лічби.

Для цього випускаються стандартні інтегральні схеми лічильників, які мають входи D_n установлювання необхідного модуля лічби (коефіцієнта ділення). Умовне позначення таких лічильників показано на рис. 4.39.

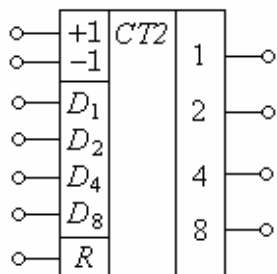


Рисунок 4.39 –
Умовне позначення
реверсивного
лічильника з
регульованим
модулем лічби

Установлюючи на входах $D1, D2, D3, D4$ двійкове число, можна отримувати модуль лічби лічильника (коефіцієнта ділення подільника частоти) від 2 до 16 з дискретністю в одиницю. Щодо серійних подільників частоти, то вони мають лише лічильний вхід та вихід переносу без виводів – виходів тригерів лічильника.

4.5. Регістри

4.5.1. Класифікація регістрів

Регістри називаються послідовнісні пристрої, які виконують функцію приймання, запам'ятовування і передавання інформації. Інформація в регістрі зберігається за видом числа (слова), зображеного комбінацією сигналів 0 та 1.

Кожному розряду числа, що записане в регістр, відповідає свій розряд, побудований, на базі тригерів *RS*-, *D*- або *JK*-типу.

На регістрах можна виконувати операції перетворення інформації з одного виду в інший, наприклад, послідовного коду в паралельний. Регістри можуть використовуватися для виконання деяких логічних операцій, наприклад, логічне порозрядне множення або ділення.

Основною класифікаційною відзнакою, за якою відрізняють регістри, є спосіб запису і зчитування двійкової інформації. За цією відзнакою розрізняють три типи регістрів: послідовні, паралельні та паралельно-послідовні.

У *послідовних регістрах* запис і зчитування інформації здійснюються послідовно за часом, тобто почергово. Вони можуть мати як послідовні виходи, так і паралельні. Інформація записується шляхом послідовного зсуву числа синхроімпульсами. Тому регістри послідовного типу носять назву *регістрів зсуву*.

У *паралельних регістрах*, які мають паралельні входи та виходи, запис інформації виконуються одночасно в усіх розрядах за один такт керування. Такі регістри називають *регістрами пам'яті*.

Паралельно-послідовні регістри мають або паралельний вхід та послідовний вихід, або послідовний вхід та паралельний вихід. В перших регістрах інформація записується одночасно по паралельних входах, а зчитується почергово, в других – записується почергово, а зчитується паралельно. Паралельно-послідовні регістри можуть бути як регістрами зсуву, так і регістрами пам'яті.

За способом приймання та передавання інформації послідовні, паралельні та послідовно-паралельні регістри поділяються на чотири типи:

- *реєстри типу SISO* – з послідовним входом та послідовним виходом;
- *реєстри типу SIPO* – з послідовним входом та паралельним виходом;
- *реєстри типу PISO* – з паралельним входом та послідовним виходом;
- *реєстри типу PIPO* – з паралельними входом і виходом.

Найбільш універсальними є реєстри, які мають у своєму складі разом послідовні і паралельні входи й виходи. Такі реєстри називаються реєстрами з послідовно-паралельним прийманням інформації та послідовно-паралельним передаванням.

4.5.2. Послідовні реєстри

Послідовні реєстри, або *реєстри зсуву* призначені для виконання операцій зсуву двійкової інформації під дією зовнішніх синхроімпульсів.

Основною класифікаційною відзнакою реєстрів зсуву є спосіб керування зсувом числа. Ця відзнака припускає поділ реєстрів зсуву за числом тактових синхроімпульсів, необхідних для виконання операції зсуву інформації на один розряд.

Бувають реєстри зсуву *однотактової* та *багатотактової* дій.

Особливість перших полягає в тому, що зсув числа на один розряд виконується за допомогою одного синхроімпульсу.

У багатотактових реєстрах зсув числа на один розряд здійснюється, як мінімум, за два імпульси зсуву.

У свою чергу обидва види реєстрів класифікуються за трьома ознаками: видом електричного зв'язку між розрядами, напрямом зсуву та способом приймання і передавання інформації.

За видом електричного зв'язку між розрядами послідовні реєстри поділяються на *однопровідні*, *двопровідні* та реєстри зі *змішаним типом зв'язків*. Однопровідні реєстри будуються на тригерах *D*-типу, двопровідні – на *RS*- або *JK*-тригерах, а треті – на сумісно використаних тригерах *RS*- і *D*- або *JK*- і *D*- типів.

За напрямом зсуву реєстри відрізняються за трьома ознаками:

– реєстри, що виконують зсув числа від молодших до старших розрядів називаються *реєстрами зсуву вправо*;

– реєстри, що виконують зсув числа від старших до молодших розрядів називаються *реєстрами зсуву вліво*;

– реєстри, що виконують зсув як вправо, так і вліво – *реверсивні реєстри зсуву*.

Основне призначення реєстрів послідовної дії – це зсув інформації, яка подається на вхід у двійковому коді. В якості таких реєстрів можна використовувати типи *SISO* та *SIPO*, тобто реєстри з послідовними входами.

На рис. 4.40 показана схема та умовне позначення чотирирозрядного однотактового реєстра зсуву вправо на синхронних *D*-тригерах.

Реєстр має один інформаційний вхід *D*, на який надходить інформація у вигляді послідовності імпульсів, і тактовий вхід *C*, на який подаються синхронізуючі імпульси.

На *паралельних* виходах реєстра $Q_1 \dots Q_4$ інформація з'являється *одночасно*, а на *послідовному* виході, яким є Q_4 , тільки по черзі по одному біту.

Перший тригер $D1$ відповідає за молодший розряд кожного слова інформації, а $D4$ – за старший, якщо вважати, що інформація надходить, починаючи з молодшого розряду.

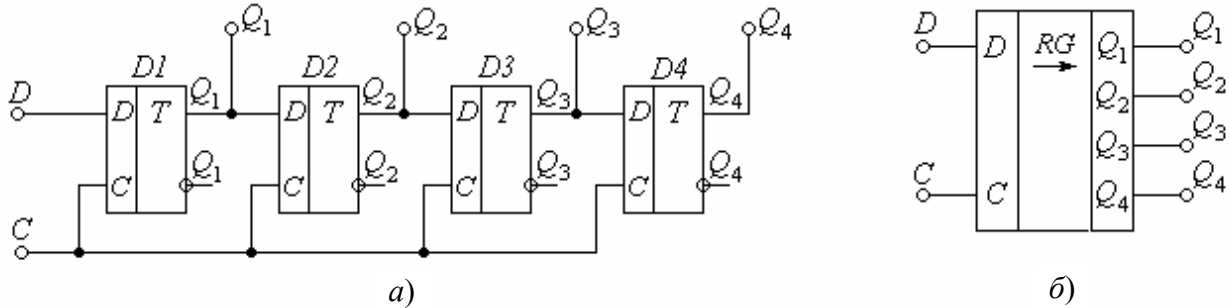


Рисунок 4.40 – Однотактовий реєстр зсуву вправо: а) схема; б) умовне позначення

З надходженням черги синхронізуючих імпульсів одночасно на тактові входи C усіх D -тригерів з кожним фронтом C вихід кожного наступного тригера набуватиме стану попереднього тригера.

Реєстр, що має чотири тригера або чотири розряди, може прийняти інформацію, яка складається з чотирьох бітів.

Для запису і зчитування інформації у даному реєстрі потрібно мати чотири такти синхроімпульсів (за числом розрядів реєстра). Зчитування інформації з реєстра, яка після припинення подачі тактових імпульсів зберігатиметься на його виходах Q_1 , Q_2 , Q_3 , і Q_4 , можна виконати в паралельному коді одночасно з чотирьох означених виходів або в послідовному коді, з виходу останнього тригера $D4$.

Принцип дії реєстра зсуву вправо ілюструє часова діаграма його роботи (рис. 4.41).

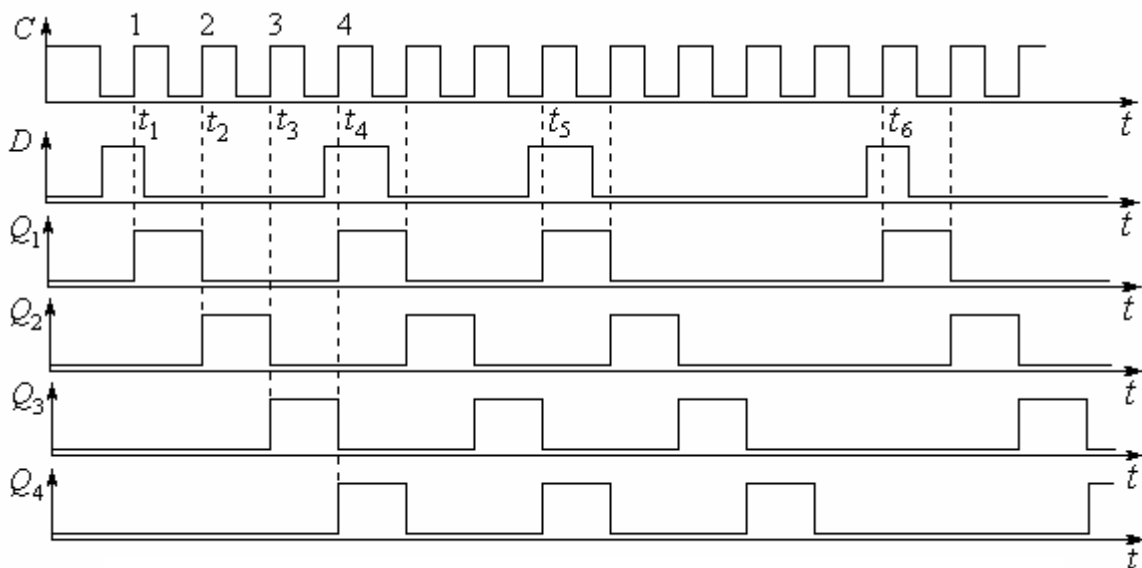


Рисунок 4.41 – Часова діаграма роботи реєстра зсуву вправо

Реєстр зсуву вправо працює наступним чином.

Нехай на момент t_1 на інформаційний вхід D регістра надійшла одиниця. Вона готує до встановлення тільки тригер першого молодшого розряду $D1$. Входи D тригерів $D2$, $D3$ і $D4$ залишаються під нульовими потенціалами.

Фронт першого синхроімпульсу C в момент t_1 встановлює тригер $D1$ і на його виході буде $Q_1 = 1$. Ця інформація передається до входу D тригера $D2$ й одиничний потенціал на вході D готує $D2$ до встановлення.

Другий синхроімпульс в момент t_2 встановлює тригер $D2$ і змінює потенціал на його виході Q_2 з нуля на одиницю.

З приходом третього синхроімпульсу в момент t_3 на виході Q_3 з'являється рівень логічної "1".

Четвертий синхроімпульс в момент t_4 встановлює тригер $D4$ і на його виході Q_4 потенціал дорівнює 1.

Таким чином, у розглянутій схемі (рис. 4.40) здійснюється зсув логічної «1», що надійшла на вхід D регістра з молодшого розряду до старшого, тобто вправо.

Аналогічним чином записуються та зсуваються логічні "1", що надійшли до входу D регістра в моменти t_4 , t_5 , t_6 .

Такий регістр носить назву регістра зсуву *вправо*.

Для збільшення місткості регістрів використовують їхнє каскадування (рис. 4.42), в якому вихід останнього розряду попереднього регістра $D1$ з'єднують зі входом D наступного регістра $D2$ та об'єднують шину C . При цьому ті ж самі виходи наступного регістра Q_1 ; Q_2 ; Q_3 ; Q_4 мають у з'єднанні іншу розрядність Q_5 ; Q_6 ; Q_7 ; Q_8 , тобто після розряду Q_4 буде розряд Q_5 , потім Q_6 і т.д.

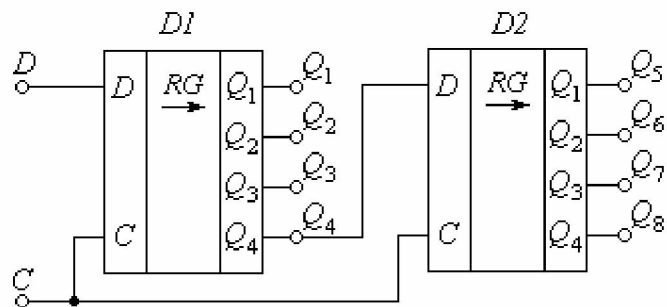


Рисунок 4.42 – Каскадування регістрів зсуву вправо

Розглядаючи типи регістрів, слід зауважити на те, що зсув інформації може бути як вправо, так і вліво. Це залежить від того, як надходить інформація послідовного двійкового числа на вхід D регістра: починаючи з молодшого або зі старшого розряду.

На рис. 10.43 показана схема чотирирозрядного регістра зсуву вліво.

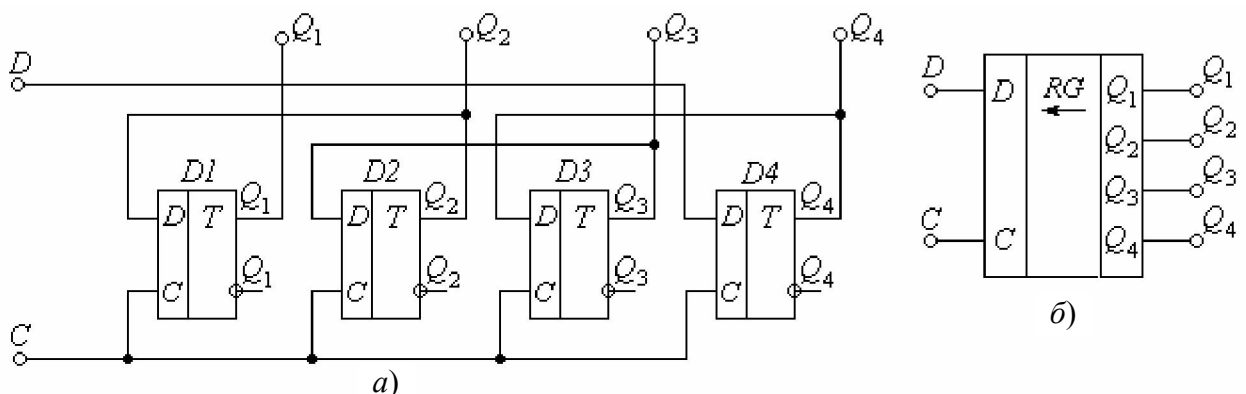


Рисунок 4.43 – Однотактовий регістр зсуву вліво: а) схема; б) умовне позначення

Для здійснення зсуву вліво вхід D регістра надходить до останнього тригера $D4$, а вихід кожного наступного тригера з'єднаний з D -входом попереднього.

Принцип дії регістра зсуву вліво ілюструє часова діаграма його роботи (рис. 4.44).

Регістр зсуву вліво працює наступним чином.

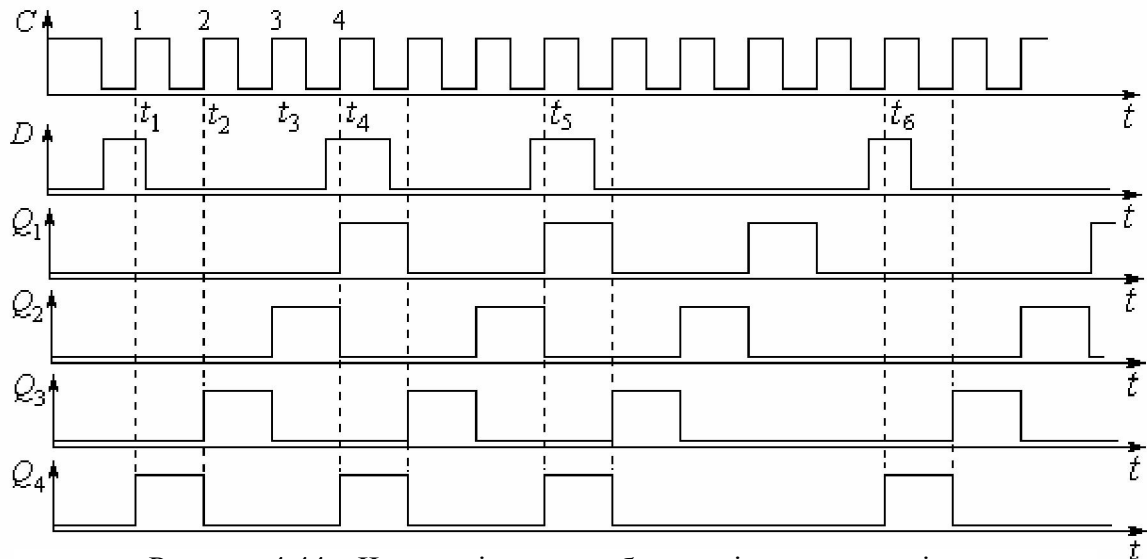


Рисунок 4.44 – Часова діаграма роботи регістра зсуву вліво

Нехай на момент t_1 до інформаційного входу D регістра надійшла одиниця. Вона готує до встановлення тільки тригер старшого розряду $D4$. Входи D тригерів $D1$, $D2$, $D3$ залишаються під нульовими потенціалами.

Фронт першого синхроімпульсу C в момент t_1 встановлює тригер старшого розряду $D4$ і на його виході буде $Q_4 = 1$. Ця інформація передається до входу D тригера $D3$ і одиничний потенціал на вході D готує $D3$ до встановлення.

Другий синхроімпульс в момент t_2 встановлює тригер $D3$ і змінює потенціал на його виході Q_3 з нуля на одиницю.

З приходом третього синхроімпульсу в момент t_3 на виході Q_2 з'являється рівень логічної одиниці.

Четвертий синхроімпульс в момент t_4 встановлює тригер $D1$ і на його виході Q_1 потенціал дорівнює 1.

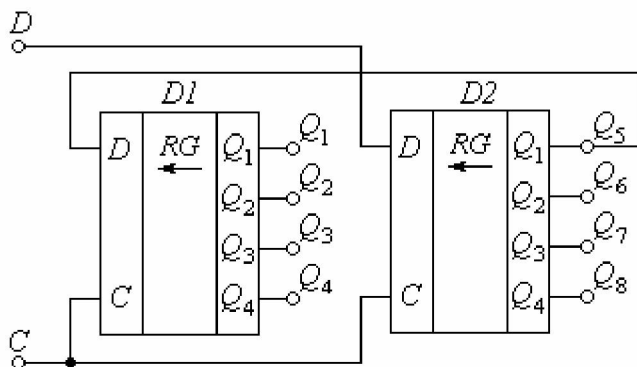


Рисунок 4.45– Каскадування регістрів зсуву вліво

Таким чином, у розглянутій схемі (рис. 4.44) здійснюється зсув логічної "1", що надійшла на вхід D регістра зі старшого розряду до молодшого, тобто вліво.

Аналогічним чином записуються та зсуваються логічні одиниці, що надійшли до входу D регістра в моменти t_4 , t_5 , t_6 .

Щодо каскадування регістрів зсуву вліво (рис. 4.45), то збіль-

шення їхньої місткості здійснюється нарощуванням *молодших* розрядів, тобто підключенням до регістра $D2$ регістра $D1$. При цьому вхід D регістра $D1$ підключається до виходу *молодшого* розряду Q_5 регістра $D2$.

Перевагою обох розглянутих одноктактових регістрів зсуву є відносна простота, бо на один двійковий розряд крім єдиного тригера нічого більше не потрібно.

Недоліком є явище міжкаскадних “гонок”, яке полягає в тому, що запис інформації в наступний розряд (тригер) і спрацювання попереднього тригера, з якого переписується інформація, відбуваються *одночасно*. Це може привести до того, що попередній тригер вже спрацював, а переписати з нього інформацію наступний тригер ще не встиг. Тому функціональна надійність *одноктактового* регістра зсуву в деяких випадках (наприклад, наявність завад) може бути недостатньою. Цей недолік зумовлений *спільною* шиною синхроімпульсів C , через що запуск усіх тригерів здійснюється *одночасно*.

Цей недолік усунений в регістрах зсуву *багатотактової* дії, в яких кожний тригер запускається своїм окремим синхроімпульсом C_i . Тому і запис інформації наступним тригером з попереднього, і спрацювання попереднього тригера рознесені за часом. Завдяки цьому функціональна надійність *багатотактового* регістра зсуву вища за *одноктактового*.

Схема багатотактового трирозрядного регістра зсуву вправо на тригерах D -типу показана на рис. 446.

Крім розрядних тригерів $D1$, $D2$, $D3$ регістр містить *додатковий* тригер $D0$ зі своїм синхровходом C для запису інформації D .

Кожний з розрядних тригерів $D1$, $D2$, $D3$ запускається *своїм* синхровходом відповідно C_1, C_2, C_3 , через що запис і зчитування інформації рознесені за часом і не зумовлюють розрядних гонок.

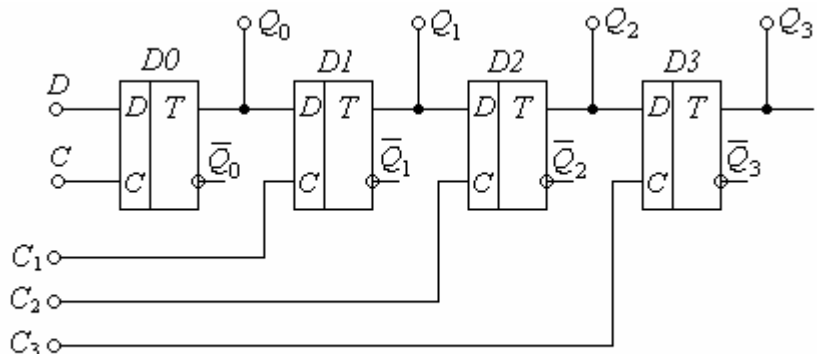


Рисунок 4.46 – Багатотактовий регістр зсуву вправо

Запис відбувається з приходом першого синхроімпульсу на вхід синхронізації C_1 першого тригера $D1$. Регістр (рис. 4.46) – трирозрядний, бо має три основних тригери: $D1$, $D2$, і $D3$.

Зсув відбувається в наступному такті за допомогою послідовно сформованих синхроімпульсів, що такт за тактом з’являються на синхровходах C_1, C_2, C_3 .

Перший розряд інформації зсувається до тригера $D3$ за три такти. Загальне число синхроімпульсів для зсуву першого розряду інформації дорівнює 4 (1 запис + 3 зсуви). Другий розряд інформації другим синхроімпульсом C записується у додатковий тригер $D0$, а за допомогою 2-х синхроімпульсів (C_0 та C_1) пересувається до тригера $D1$.

Пересув другого розряду інформації виконується за дією трьох синхроімпульсів, а третього – чотирьох синхроімпульсів.

Аналіз кількості імпульсів дозволяє дійти висновку, що для запису повного n -розрядного слова або зсуву n -розрядів у регістрах даного типу потрібно мати імпульси зсуву:

$$n * n = n^2. \quad (4.17)$$

Підвищити швидкодію регістрів багатотактової дії можна шляхом розбивки регістра на окремі групи.

Недоліком регістрів зсуву багатотактової дії є не тільки мала швидкодія, а й складна схема керування зсувом. Однак наявність багатотактової схеми зсуву виключає явище міжкаскадних “тонок” і збільшує функціональну надійність регістра.

Існують *двонаправлені* або *реверсивні* регістри зсуву, які здатні зсувати записану інформацію і вправо, і вліво.

Промисловістю випускаються також універсальні регістри, які мають і паралельні, і послідовний входи.

На рис. 4.47 показане умовне позначення універсального регістра.

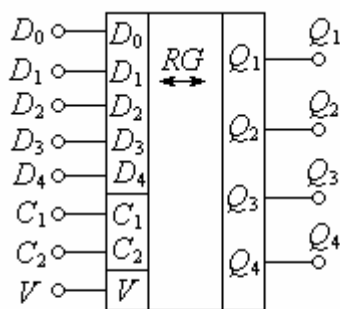


Рисунок 4.47 – Умовне позначення універсального регістра зсуву

Тут:
 D_0 – послідовний вхід, використання якого зумовлює зсув;

$D_1... D_4$ – паралельні входи, через які інформація записується паралельно;

C_1 – вхід керування типом регістра (паралельний або регістр зсуву);

C_2 – тактовий вхід;

V – керуючий вхід, рівнем якого (0 або 1) установлюється напрям зсуву;

$Q_1...Q_4$ – паралельні виходи.

Інформацію з виходів можна знімати або в паралельному коді з виходів $Q_1...Q_4$, або в послідовному коді з виходу Q_4 .

Щодо застосування регістрів зсуву, то воно досить різноманітне. На їхній базі можна будувати функціональні вузли. Зсув інформації на один розряд вправо або вліво виконує арифметичні операції відповідно або *ділення*, або *множення* на два.

Регістр зсуву може виконувати функцію лічильника, якщо на його послідовний вхід D подавати чергу одиниць. На базі регістрів зсуву можна будувати цифрові лінії затримки.

Регістри зсуву, що утримують лише одну одиницю, можуть виконувати роль лічильника, який відображує число імпульсів, які надійшли на його вхід, станом одиниці на лінійній шкалі. Це, наприклад, лампочка, яка світиться, вказуючи номер поверху в ліфті. Якщо на вхід зсуву такого лічильника подати імпульси тактового генератора, то можна отримати прилад, який розподіляє опорні тактові імпульси по декількох фазах синхронізації.

У системах радіозв'язку та радіолокації регістри зсуву застосовують для побудови радіозамків, двійкових кореляторів та інших пристроїв складного оброблення радіосигналів.

4.5.3. Паралельні регістри

Паралельні регістри або *регістри пам'яті* – це багаторозрядні регістри типу *PIPO* паралельної дії з паралельними входами та паралельними виходами, в яких кількість розрядів визначається числом тригерів, на яких будується регістр.

Регістри пам'яті – це *накопичувальні регістри*. Їхнє основне призначення – збирання двійкової інформації невеликого обсягу.

Перший молодший розряд числа записується та зберігається у першому молодшому розряді регістра, другий – у другому розряді і т.д. Старший розряд регістра приймає старший розряд числа.

Однофазні паралельні регістри одноктактової дії можна побудувати, наприклад, на *D*-тригерах, кількість яких залежить від числа входів регістра.

На рис. 4.48 показана схема 4-розрядного одноктактового регістра пам'яті, яка побудована на чотирьох синхронних *D*-тригерах.

Запис числа $A (a_1, a_2, a_3, a_4)$ виконується за наявності синхроімпульсу C без попереднього скидання тригерів у стан нуля.

Регістри пам'яті будуються на тригерах різноманітних типів.

Кожен розряд двійкового числа $A (a_1, a_2, a_3, a_4)$ подається на інформаційний вхід D окремого тригера $D_1 \dots D_4$.

Інформація записується в регістр тільки з появою фронту синхроімпульсу C , а зчитувати її з виходів Q кожного тригера можна в будь-який час після завершення процесу запису.

Якщо на вхід синхроімпульсів C поданий рівень логічного "0", то регістр перебуває у стані збереження записаної інформації.

Синхронізуються регістри рівнями 1 або 0, фронтом чи зрізом синхросигналу, залежно від застосованого тригера. Інформацію про полярність імпульсу синхронізації можна отримати у довідниках. В нашому випадку при застосуванні *D*-тригерів синхронізація відбувається фронтом імпульсу C .

Існують регістри, в яких зчитування інформації з виходів також синхронізується. Однак при цьому запис та зчитування обов'язково рознесені за часом.

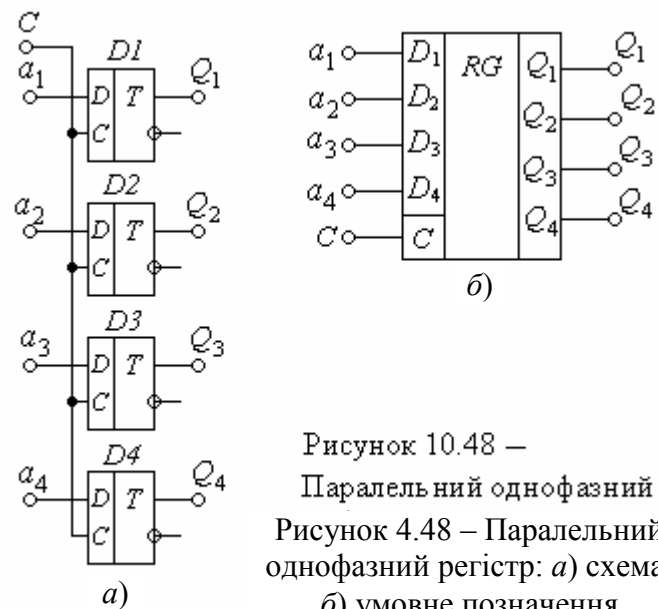


Рисунок 10.48 –
Паралельний однофазний
Рисунок 4.48 – Паралельний
однофазний регістр: а) схема;
б) умовне позначення

Досить часто виникає задача запису двох і більше чисел. В цьому разі треба мати додаткову ємність регістра, додаткові інформаційні входи і додаткові входи для синхроімпульсів.

Класифікація регістрів пам'яті здійснюється за трьома ознаками:

- однофазні паралельні регістри однотокової дії;
- однофазні паралельні регістри двотокової дії;
- парафазні паралельні регістри.

Однофазні паралельні регістри однотокової дії досить економічні та надійні. Швидкодія цих регістрів більша приблизно в два рази за однофазні регістри двотокової дії, але при цьому ж порівнянні апаратні втрати на реалізацію однотокових регістрів більші.

Серед регістрів *двотокової дії* найбільш економічними за числом компонентів вважаються регістри, які побудовані на синхронних тригерах *CSR*-типів або двотокових *D*-тригерах.

Регістри даного типу мають обов'язково вхід *R* (або *V*) установлення тригерів у стан 0. Тому для запису числа треба сформувати два синхросигнали: перший подається на вхід *R* (або *V*) для скидання всіх тригерів, а другий – на вхід *C* для запису інформації, що присутня на інформаційних входах *D*. Умовне позначення такого регістра показано на рис. 4.49.

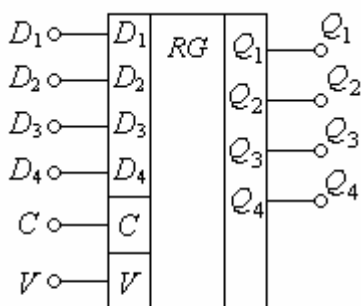


Рисунок 4.49 – Умовне позначення однофазного регістра двотокової дії

Парафазні паралельні регістри за принципом дії є однотоковими і будуються на синхронних тригерах *CSR*-типу. Число подається на регістр одночасно по двох каналах: прямому та інверсному.

Синхроімпульс запису *C* відкриває одночасно вентилі запису прямого та інверсного входів. Попереднє скидання тригерів такого регістра в нуль вимагає додаткового часу, що зменшує швидкодію регістра. Парафазні паралельні регістри не відрізняються широкою різноманітністю схем. Усі парафазні паралельні регістри є однотоковими і виконуються на однакових тригерах. При порівнянні

однофазних та парафазних регістрів слід відзначити, що однофазні вважаються найбільш ефективними в інтегральному виконанні тому, що мають у два рази менше інформаційних входів.

Розрядність регістрів нарощують збільшенням числа тригерів у схемі регістра. Розрядність регістрів можна збільшити їхнім каскадуванням.

Контрольні питання

- 4.1. Що таке тригер?
- 4.2. Який тригер називається синхронним, а який – асинхронним?
- 4.3. Що таке інформаційні та керуючі входи тригера?
- 4.4. Що таке *RS*-тригер і на яких елементах його можна реалізувати?
- 4.5. Що таке *D*-тригер і на яких елементах його можна реалізувати?

- 4.6. Роздільний запуск *D*-тригера.
- 4.7. Лічильний запуск *D*-тригера.
- 4.8. Одноступеневий *JK*-тригер.
- 4.9. Двоступеневий тригер. Від яких фронтів вхідного імпульсу спрацьовує кожний ступінь двоступеневих тригерів?
- 4.10. Тригер Шмітта.
- 4.11. Захист тригерів буфером від дії завад.
- 4.12. Захист тригерів від дренькоту контактів, що запускають тригери.
- 4.13. Що таке лічильник?
- 4.14. Що таке модуль лічби?
- 4.15. Які лічильники називаються послідовними?
- 4.16. Що таке реверсивний лічильник?
- 4.17. Як перетворити підсумовувальний лічильник у віднімальний?
- 4.18. Підсумовувальний та віднімальний лічильники на *D*-тригерах.
- 4.19. Підсумовувальний та віднімальний лічильники на *JK* –тригерах.
- 4.20. Що таке подільник частоти?
- 4.21. Паралельні лічильники та їхні переваги і недоліки за порівнянням з послідовними.
- 4.22. Паралельно-послідовні лічильники.
- 4.23. Реверсивні лічильники з двома керуючими входами.
- 4.24. Реверсивні лічильники з двома синхровходами.
- 4.25. Лічильники з довільним модулем лічби та їх застосування.
- 4.26. Збільшення розрядності лічильників.
- 4.27. Призначення регістрів зсуву.
- 4.28. Призначення паралельних регістрів.
- 4.29. Регістри зсуву вправо.
- 4.30. Регістри зсуву вліво.
- 4.31. Паралельні регістри.

Рекомендована література

1. Воробйова О.М. Основи схемотехніки: У двох частинах: навч. посіб. / О.М. Воробйова, В.Д. Іванченко. – Одеса: ОНАЗ ім. О.С.Попова, 2004. – Ч. 2. – С. 228 – 275.
2. Титце У. Полупроводниковая схемотехника: Справочное руководство / У. Титце, К. Шенк. – М.: Мир, 1982. – С. 97 – 387.
3. Хоровиц П. Искусство схемотехніки / П. Хоровиц, У. Хилл. – М.: Мир, 1983. – Т. 1. – С. 540 – 569.
4. Рицар Б.Є. Цифрова техніка / Рицар Б.Є. – К.: НМК ВО, 1990. – 371 с.

Навчальне видання

Воробйова Олена Михайлівна
Савицька Маргарита Павлівна
Флейта Юрій Вікторович

ЦИФРОВІ ПРИСТРОЇ

Навчальний посібник

Частина 1

Редактор Кодрул Л.А.

Верстання Гардиман Ж.А.