

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

**О. В. Заморський
О. М. Павловський**

ОСНОВИ ЦИФРОВОЇ СХЕМОТЕХНІКИ

ПРАКТИКУМ

НАВЧАЛЬНИЙ ПОСІБНИК

*Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою
«Комп'ютерно-інтегровані системи та технології в приладобудуванні»
спеціальностей*

*151 «Автоматизація та комп'ютерні технології»
174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»*

Київ
КПІ ім. Ігоря Сікорського
2023

Рецензенти	Галаган Р.М. – канд. техн. наук, доцент, доцент кафедри автоматизації та систем неруйнівного контролю КПІ ім. Ігоря Сікорського Головач С.В. – канд. техн. наук, головний конструктор напрямку АТ «Елміз»
Відповідальний редактор	Васильковська І. О., канд. техн. наук, ст. викладач кафедри комп'ютерно-інтегрованих оптичних і навігаційних систем КПІ ім. Ігоря Сікорського

*Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
(протокол № 7 від 27.04.2023 р.)
за поданням Вченої ради Приладобудівного факультету
(протокол № 4/23 від 24.04.2023 р.)*

ОСНОВИ ЦИФРОВОЇ СХЕМОТЕХНІКИ ПРАКТИКУМ

Навчальний посібник містить практичні вправи і завдання з дисципліни «Основи цифрової схемотехніки», довідкові теоретичні відомості та методичні рекомендації для їх виконання.

Видання призначене для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спеціальностей 151 «Автоматизація та комп'ютерні технології» і 174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка» та може використовуватись викладачами при проведенні практичних занять та студентами при самостійній роботі за іншими освітніми програмами споріднених спеціальностей.

Реєстр. № НП 22/23-662 Обсяг 7,41 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056

<https://kpi.ua>

Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© О. В. Заморський, 2023
© О. М. Павловський, 2023
© КПІ ім. Ігоря Сікорського, 2023

ЗМІСТ

ВСТУП	5
1. ПРАКТИКУМ 1. СИСТЕМИ ЧИСЛЕННЯ.....	9
1.1. Вступ. Двійкові числа як структурні елементи цифрової логіки	9
1.1.1. Логіка висловлювань	9
1.1.2. Цифрова логіка і двійкові числа.....	11
1.2. Позиційні системи числення	12
1.2.1. Двійкова позиційна система числення	12
1.2.2. Вісімкова і шістнадцяткова позиційні системи числення	14
1.3. Арифметичні операції над двійковими числами	17
1.3.1. Двійкові числа зі знаком, додатковий код (доповнення)	17
1.3.2. Операції додавання і віднімання двійкових чисел	17
1.4. Вправи і завдання	20
2. ПРАКТИКУМ 2. ЦИФРОВА ЛОГІКА	23
2.1. Основні положення цифрової логіки	23
2.1.1. Аналітичне і табличне представлення логічних функцій	23
2.1.2. Правила обчислень логічних виразів. Закон Моргана	26
2.2. Перетворення і спрощення логічних виразів	27
2.2.1. Аналітичне перетворення логічних виразів	27
2.2.2. Матрично-графічне представлення логічних функцій. Карти Карно ...	28
2.3. Вправи і завдання	34
3. ПРАКТИКУМ 3. ЛОГІЧНІ ФУНКЦІЇ	37
3.1. Довідкові теоретичні відомості	37
3.1.1. Функції пустої множини аргументів і функції одного аргументу	37
3.1.2. Функції двох аргументів	38
3.1.3. Перемикальні властивості логічних функцій	42
3.2. Вправи і завдання	44
4. ПРАКТИКУМ 4. БАЗОВІ ЛОГІЧНІ ЕЛЕМЕНТИ	45
4.1. Вступ. Цифрові елементи	45
4.1.1. Цифрові схеми. Дискретні сигнали	45
4.1.2. Технології виробництва цифрових інтегральних мікросхем	47
4.2. Базові логічні елементи	49
4.2.1. Визначення	49

4.2.2. Основні логічні елементи	49
4.2.3. Універсальні логічні елементи	50
4.2.4. Виключні логічні елементи	51
4.2.4. Еквівалентні логічні схеми	53
4.2.5. Прості логічні схеми застосування базових логічних елементів	55
4.3. Вправи і завдання	56
4.4. Додаток	58
5. ПРАКТИКУМ 5. КОМБІНАЦІЙНІ ЦИФРОВІ ПРИСТРОЇ	59
5.1. Вступ	59
5.1.1. Комбінаційна і послідовнісна логіка	59
5.1.2. Інтегральні мікросхеми цифрової логіки	60
5.2. Сімейство інтегральних мікросхем пристрою цифрової логіки	62
5.2.1. Мультиплексори. Стандартна логічна схема селектора даних	62
5.2.2. Відмінності логічних схем мікросхем селектора даних	63
5.2.3. Аналогові мультиплексори/демультиплексори з цифровим керуванням ..	66
5.2.4. Сімейство мікросхем селектора даних (мультиплексора 2 до 1)	68
5.3. Комбінаційні цифрові пристрої	69
5.3.1. Мультиплексори	69
5.3.2. Демультимплексори	71
5.3.3. Шифратори і дешифратори	72
5.3.4. Компаратори і суматори	74
5.4. Вправи і завдання	75
6. ПРАКТИКУМ 6. ІМПУЛЬСНІ ПРИСТРОЇ	79
6.1. Довідкові відомості	79
6.1.1. Імпульсні пристрої, тактові сигнали і одиничний імпульс	79
6.1.2. Мультивібратори	80
6.1.3. Схеми мультивібраторів на основі мікросхеми таймера 555	83
6.1.4. Практичні схеми реалізації генераторів тактових імпульсів	85
6.2. Вправи і завдання	88
7. ПРАКТИКУМ 7. ПОСЛІДОВНІСНІ ЦИФРОВІ ПРИСТРОЇ	91
7.1. Довідкові відомості	91
7.1.1. Тригери	91
7.1.2. Лічильники	101
7.1.3. Регістри	104
7.2. Вправи і завдання	106
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	107

ПЕРЕДМОВА

Навчальний посібник «Основи цифрової схемотехніки» (практикум) призначений для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спеціальностей «Автоматизація та комп'ютерно-інтегровані технології» та «Автоматизація, комп'ютерно-інтегровані технології та робототехніка».

Навчальний посібник складається з семи тематичних практикумів, кожен з яких містить довідкові теоретичні відомості, приклади вирішення практичних задач та варіанти практичних вправ і завдань для самостійного вирішення.

Перший практикум «Системи числення» складається з двох частин. У першій, вступній частині, двійкові числа 1 і 0 розглядаються як *структурні елементи цифрової логіки*. Через логічні висловлювання визначаються поняття логічної змінної, логічної операції і логічної функції. На вербальних прикладах розглянуті всі три основні логічні функції – І, АБО, НЕ. Зазначаються основні логічні елементи цифрової схемотехніки та їх фізична реалізація, інтегральні мікросхеми. Визначаються еквівалентні значення «істинності» і «хибності» бінарних логічних змінних – вербальні ТАК (TRUE) і НІ (FALSE), цифрові 1 і 0, фізичні ВИСОКИЙ (HIGH) і НИЗЬКИЙ (LOW) рівні сигналу. Розглядаються впорядковані набори двійкових значень 1 і 0 логічних змінних функції – таблиці істинності функції. В другій частині практикуму двійкові числа 1 і 0 розглядаються, власне, як *символьні знаки двійкової позиційної системи числення*. Подаються способи і приклади перетворення чисел із однієї системи чисел в іншу (двійкова, вісімкова, десяткова, шістнадцяткова системи числення). Розглядаються двійкові числа в додатковому коді з повним і неповним доповненням. Наведені приклади здійснення простих арифметичних операцій з двійковими числами.

Два наступні практикуми, «Цифрова логіка» та «Логічні функції», присвячені булевій алгебрі або алгебрі логіки, аналітичній основі цифрової схемотехніки.

У першій частині другого практикуму «Цифрова логіка» розглядаються основи цифрової логіки – аналітичні і табличні представлення логічних функцій, основні логічні операції кон'юнкції, диз'юнкції та заперечення, ідеальні нормальні кон'юнктивні та диз'юнктивні форми запису функції, правила і закони обчислення логічних функцій. У другій частині практикуму подаються

прикладі перетворення (спрощення) виразів логічних функцій аналітичним способом із застосуванням правил і законів алгебри логіки та матрично-табличним способом із застосуванням карт Карно.

У третьому практикумі «Логічні функції» розглядаються повні сімейства логічних функцій пустої множини аргументів (логічних змінних) та множин одного і двох аргументів. Додатково до основних логічних функцій визначаються функції констант, тотожності, еквіваленції та імплікації, а також несуттєві (фіктивні) аргументи та втрачені функції розглянутих сімейств. Вказані перемикальні властивості логічних функцій, подані еквівалентні перемикальні електричні схеми логічних функцій пустої множини аргументів, одного аргументу й основних логічних функцій двох аргументів. Розглянута схема реалізації перемикальної функції на основі транзисторного ключа за технологією резистивно-транзисторної логіки (РТЛ).

Четвертий практикум «Базові логічні елементи» складається з двох частин. У першій, вступній частині, розглядаються загальні питання символічного представлення логічних схем, способів отримання цифрових (дискретних) сигналів, технологій виробництва інтегральних мікросхем цифрової логіки – транзистор-транзисторної логіки (ТТЛ) та комплементарної структури метал-оксид-напівпровідник (КМОП). В другій частині практикуму послідовно розглядаються всі базові логічні елементи – основні, універсальні та виключні. Подані їх аналітичні та табличні визначення, символічні схеми за різними стандартами та еквівалентні перемикальні схеми за технологією РТЛ. Для універсальних та виключних елементів показані еквівалентні схеми на основних логічних елементах. Вказуються інтегральні мікросхеми серії 74LS00 реалізації базових логічних елементів за стандартною схемою. Завершують практикум приклади побудови еквівалентних логічних схем через перетворення (спрощення) виразів логічної функції, реалізованої заданими схемами на основі базових логічних елементів, та приклади побудови простих схем практичного застосування базових логічних елементів. У додатку до практикуму для основних логічних елементів подані їх еквівалентні схеми на універсальних логічних елементах.

У п'ятому практикумі «Комбінаційні цифрові пристрої» подається класифікація цифрових пристроїв за принципом побудови логічних схем (комбінаційні і послідовнісні) та класифікація практично застосованих комбінаційних цифрових пристроїв за їх функціональним призначенням (мультиплексори і демультимплексори, шифратори і дешифратори, компаратори і суматори). На прикладі селектора даних (мультиплексора 2 до 1) розглядається сімейство інтегральних мікросхем цього пристрою. Вказується на типові відмінності логічних

схем мікросхем такого сімейства від стандартної логічної схеми, які характерні для всіх промислово реалізованих сімейств інтегральних мікросхем інших цифрових пристроїв. Розглядаються відмінності логічних схем сімейства мікросхем за їх входами (наявність тригерів Шмітта, додаткових керуючих сигналів) та виходами (вихід з двома і трьома станами, вихід з відкритим колектором, інверсний вихід). Для вибраного сімейства інтегральних мікросхем селектора даних показані символічні схеми мікросхем з призначенням контактів, логічні схеми та таблиці істинності мікросхем. У вибраному сімействі мікросхем селектора даних розглянуті також комбіновані схеми, що поєднують комбінаційні і послідовні цифрові пристрої (селектори даних з пам'яттю) та аналогові і цифрові пристрої (аналогові мультіплексори/демультиплексори з цифровим керуванням). Наведено приклад практичної реалізації цифроаналогового перетворювача (ЦАП) на основі аналогового мультіплексора/демультиплексора і двійково вагомих резисторів. Розглянута стандартна логічна схема розподільовача даних (декодера або демультиплексора). Для мультіплексорів і демультиплексорів наведені табличні дані інтегральних мікросхем, присутніх на ринку. Наведено приклад спільного практичного застосування шифратора і дешифратора в схемі управління з цифрової клавіатури семисегментним світлодіодним індикатором. Вказано на тенденцію припинення практичного застосування спеціалізованих інтегральних мікросхем шифраторів, дешифраторів, суматорів через широке застосування цифрових пристроїв з програмним управлінням – мікроконтролерів та програмованих користувачем вентиляційних матриць (ПКВМ). Для вивчення цифрових пристроїв та виконання практичних завдань рекомендовано ознайомлення з технічною документацією від виробника інтегральних мікросхем цифрових пристроїв.

Шостий практикум «Імпульсні цифрові пристрої» присвячений важливим для цифрової схемотехніки електронним пристроям, що працюють у переривчастому (імпульсному, дискретному) режимі, виконуючи функцію формування, генерування та перетворення імпульсів прямокутної форми. Визначаються основні характеристики послідовності прямокутних тактових імпульсів та імпульсного сигналу при його проходженні через цифровий пристрій. Розглядаються схеми мультівібраторів на біполярних n - p - n транзисторах – астабільних (генераторів тактових імпульсів), бістабільних (тригерів) і моностабільних (одновібраторів) мультівібраторів. Подаються схеми астабільних і моностабільних мультівібраторів на основі широко застосованої інтегральної мікросхеми таймера 555. Розглядаються практичні схеми реалізації генераторів тактових імпульсів на інтегральних мікросхемах 74HC04 і 74LS00 дискретних базових логіч-

них елементів НЕ (NOT) і І-НЕ (NAND) відповідно. Практикум завершується практичними схемами застосування спеціалізованої інтегральної мікросхеми 74LS124 генераторів тактових сигналів, керованих напругою.

Сьомий практикум «Послідовнісні цифрові пристрої» структурно і методично побудовано, як і п'ятий практикум «Комбінаційні цифрові пристрої». Подаються логічні та символічні схеми, таблиці істинності та часові діаграми базових послідовних цифрових пристроїв – асинхронних та синхронних тригерів. Розглядаються всі, практично застосовані, способи активації синхронних тригерів – рівнями синхронізуючого тактового сигналу (високим та низьким), перепадами тактового сигналу (фронтами тактового імпульсу), тактовим імпульсом (по схемі керуючий-керований тригер). Наведені приклади схем побудови послідовнісних цифрових пристроїв з застосуванням інтегральних мікросхем тригерів (схема запобігання брязкоту контактів, лічильники, регістри).

Практикум супроводжується значним ілюстративним матеріалом та значною кількістю внутрішніх перехресних посилань на формули, рисунки та виклади з інших розділів, підрозділів та параграфів, нумерація яких, як правило, укладена в дужки та інтуїтивно зрозуміла. В кожному з семи практикумів нумерація формул, рисунків і параграфів починається заново (спочатку), при цьому перша цифра нумерації вказує на порядковий номер практикуму. Для формул і рисунків друга цифра вказує на їх порядковий номер. Наприклад, посилання (6.3) вказує на третю формулу з шостого практикуму, а посилання (Рис. 2.11. д) вказує на фрагмент д) одинадцятого рисунку з другого практикуму. Нумерація параграфів починається заново (спочатку) в кожному підрозділі розділу практикумів, складається з одного порядкового числа і закінчується символічним знаком °, проставляється на початку першого абзацу тексту параграфу жирним шрифтом, наприклад, як 4°. При цьому в посиланні на текст параграфу, яке складається з чотирьох цифр, наприклад (5.1.2.4°), цифри по порядку вказують на номер практикуму, розділу, підрозділу, параграфу.

Практикум «Основи цифрової схемотехніки» може бути корисним також для студентів і викладачів інших приладобудівних спеціальностей та використовуватись в освітньому процесі за іншими освітніми програмами.

ПРАКТИКУМ 1.

СИСТЕМИ ЧИСЛЕННЯ

1.1. ВСТУП. ДВІЙКОВІ ЧИСЛА ЯК СТРУКТУРНІ ЕЛЕМЕНТИ ЛОГІКИ

1.1.1. Логіка висловлювань

1°. Якщо розглянути *висловлювання* – розповідні речення, які можуть бути *істинними* або *хибними*, наприклад, такі – ОПАЛЕННЯ В БУДИНКУ ВІМКНУТО (висловлювання A) та ВІКНА В КІМНАТІ ЗАЧИНЕНІ (висловлювання B), то висновок ВЗИМКУ В КІМНАТІ ТЕПЛО (висловлювання F) можна отримати через *логічне висловлювання* **ЯКЩО** ОПАЛЕННЯ В БУДИНКУ ВІМКНУТО (висловлювання A істинне) **І** ВІКНА В КІМНАТІ ЗАЧИНЕНІ (висловлювання B істинне), **ТО** ВЗИМКУ В КІМНАТІ ТЕПЛО (висловлювання F істинне).

Висловлювання F є істинним лише тоді, коли обидва висловлювання A і B істинні. Якщо одне з висловлювань A і B хибне або обидва висловлювання A і B хибні, то і висловлювання F є хибним.

Висловлювання A і B є *вхідними логічними змінними*, а висловлювання F є *вихідною логічною змінною* логічного висловлювання. Логічні змінні висловлювань приймають одне з двох словесних значень – ТАК (TRUE) або НІ (FALSE), які визначають істинність або хибність висловлювань.

У розглянутому логічному висловлюванні сполучник **І** пов’язує вхідні змінні A і B між собою, виконуючи *операцію логічного множення* **І** над ними. Сполучник **ЯКЩО**, **ТО** пов’язує вхідні змінні A і B з вихідною змінною F і разом зі сполучником **І** утворює *логічну функцію* **І** – **ЯКЩО** A **І** B істинні, **ТО** F істинне.

Всі *комбінації* логічних значень вхідних логічних змінних (висловлювання A і висловлювання B) та відповідні логічні значення вихідної логічної змінної (висловлювання F) представлені на Рис. 1.1.

ЯКЩО A **І** B , **ТО** F
Логічна функція: $A \times B = F$
Логічні змінні: A, B, F
Висловлювання A :
ОПАЛЕННЯ ВІМКНУТО
Висловлювання B :
ВІКНА ЗАЧИНЕНІ
Висловлювання F :
В КІМНАТІ ТЕПЛО

КОМБІНАЦІЇ ЗНАЧЕНЬ ЛОГІЧНИХ ЗМІННИХ			
№ комбінації	Висловлювання A	Висловлювання B	Висловлювання F
1	НІ (FALSE)	НІ (FALSE)	НІ (FALSE)
2	НІ (FALSE)	ТАК (TRUE)	НІ (FALSE)
3	ТАК (TRUE)	НІ (FALSE)	НІ (FALSE)
4	ТАК (TRUE)	ТАК (TRUE)	ТАК (TRUE)

Рис. 1.1. Комбінації значень логічних змінних для логічного висловлювання (1.1.1.1°)

2°. Якщо часткою НЕ заперечити розглянуті висловлювання A і B (1.1.1.1°), то для отриманих висловлювань – ОПАЛЕННЯ В БУДИНКУ НЕ ВІМКНУТО (висловлювання $\bar{A}$ – заперечення висловлювання A) та ВІКНА В КІМНАТІ НЕ ЗАЧИНЕНІ (висловлювання $\bar{B}$ – заперечення висловлювання B) можна отримати висновок ВЗИМКУ В КІМНАТІ НЕ ТЕПЛО (висловлювання $\bar{F}$ – заперечення висловлювання F) через *логічне висловлювання* ЯКЩО ОПАЛЕННЯ В БУДИНКУ НЕ ВІМКНУТО (висловлювання $\bar{A}$ істинне, A – хибне) АБО ВІКНА В КІМНАТІ НЕ ЗАЧИНЕНІ (висловлювання $\bar{B}$ істинне, B – хибне), ТО ВЗИМКУ В КІМНАТІ НЕ ТЕПЛО (висловлювання $\bar{F}$ істинне, F – хибне).

Висловлювання $\bar{F}$ є істинним тоді, коли хоча б одне із висловлювань $\bar{A}$ і $\bar{B}$ істинне або обидва висловлювання $\bar{A}$ і $\bar{B}$ істинні. Висловлювання $\bar{F}$ є хибним лише тоді, коли обидва висловлювання $\bar{A}$ і $\bar{B}$ хибні (Рис. 1.2).

В логічному висловлюванні частка НЕ виконує *операцію логічного заперечення* НЕ над однією змінною (в прикладі – над кожною із змінних A, B, F), утворюючи *логічну функцію* НЕ. Сполучник АБО пов’язує вхідні змінні $\bar{A}$ і $\bar{B}$ між собою, виконуючи *операцію логічного додавання* АБО над ними. Сполучник ЯКЩО, ТО пов’язує вхідні змінні $\bar{A}$ і $\bar{B}$ з вихідною змінною $\bar{F}$ і разом зі сполучником АБО утворює *логічну функцію* АБО – ЯКЩО $\bar{A}$ АБО $\bar{B}$ істинне, ТО $\bar{F}$ істинне.

ЯКЩО $\bar{A}$ АБО $\bar{B}$, ТО $\bar{F}$

Логічна функція: $\bar{A} + \bar{B} = \bar{F}$

Логічні змінні: A, B, F

Висловлювання $\bar{A}$:
ОПАЛЕННЯ НЕ ВІМКНУТО

Висловлювання $\bar{B}$:
ВІКНА НЕ ЗАЧИНЕНІ

Висловлювання $\bar{F}$:
В КІМНАТІ НЕ ТЕПЛО

КОМБІНАЦІЇ ЗНАЧЕНЬ ЛОГІЧНИХ ЗМІННИХ			
№ комбінації	Висловлювання $\bar{A}$	Висловлювання $\bar{B}$	Висловлювання $\bar{F}$
1	TAK (TRUE)	TAK (TRUE)	TAK (TRUE)
2	TAK (TRUE)	HI (FALSE)	TAK (TRUE)
3	HI (FALSE)	TAK (TRUE)	TAK (TRUE)
4	HI (FALSE)	HI (FALSE)	HI (FALSE)

Рис. 1.2. Комбінації значень логічних змінних для логічного висловлювання (1.1.1.2°)

3°. Наведені в прикладах (1.1.1.1°, 1.1.1.2°) логічні висловлювання розповідними реченнями визначають істинність протилежних один одному логічних висновків. Проте ці різні логічні висловлювання стверджують насправді одне і те саме, тобто, є *рівносильними* (2.1.2.3°).

4°. Для зручного запису логічних висловлювань використовуються *символьні знаки*. В наведених прикладах *букви* латинського алфавіту A, B і F позначають логічні висловлювання – логічні змінні, а математичні *символи* $\times, +, \bar{}$ позначають логічні операції – логічного множення І, логічного додавання АБО, логічного заперечення НЕ відповідно. Математичний *символ* $=$ пов’язує вхідні змінні A і B з вихідною змінною F , утворюючи логічну функцію (Рис. 1.1, 1.2).

Для логічних змінних, які є бінарними (двійковими), зручними для позначення їх значень істинності і хибності є *двійкові цифри 1 і 0* (1.1.2.3°).

1.1.2. Цифрова логіка і двійкові числа

1°. *Цифрова схемотехніка* базується на *логічних елементах* (Практикум 4), які виконують певні логічні функції (Практикум 3). Логічні функції пов'язують вихідні змінні з входними змінними логічних елементів через логічні операції (Практикум 2). Бінарні логічні змінні приймають лише два можливі значення, *істинне* або *хибне*. Для логічних висловлювань (1.1.1.1°, 1.1.1.2°) еквівалентами істинного і хибного логічних значень відповідно є словесні значення **ТАК (TRUE)** і **НІ (FALSE)**.

2°. Фізичним втіленням логічних елементів є *цифрові інтегральні мікросхеми* (Практикуми 4÷7), в яких електричний сигнал на вході і виході логічного елемента змінюється стрибкоподібно, *дискретним* чином, від одного сталого рівня до іншого при зміні значення логічної змінної. Фізичними еквівалентами логічних значень **ТАК (TRUE)** і **НІ (FALSE)** є певні *значення дискретних рівнів електричного сигналу*, наприклад, +5 В і 0 В. Конкретні значення рівнів сигналу залежать від схемотехнічних рішень логічних елементів та визначаються типом інтегральних мікросхем (Практикум 4). Незалежно від типу інтегральних мікросхем для цих значень прийняті загальні найменування – **ВИСОКИЙ (HIGH)** і **НИЗЬКИЙ (LOW)** рівень сигналу.

3°. Логічні функції вивчає спеціальний розділ математики – *алгебра логіки* (*булева алгебра*), по законам якої виконуються логічні операції. Символьними еквівалентами логічних значень **ТАК (TRUE)** і **НІ (FALSE)** та значень рівнів сигналів **ВИСОКИЙ (HIGH)** і **НИЗЬКИЙ (LOW)** є цифри 1 і 0. Над *числовими значеннями* 1 і 0 логічних змінних в алгебрі логіки виконуються логічні операції. Еквівалентні значення логічних, фізичних і числових змінних логічних елементів для звичної *позитивної логіки* (4.1.1.3°) представлені на Рис. 1.3.

ЕКВІВАЛЕНТНІ ЗНАЧЕННЯ ЗМІННИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ				
ЛОГІЧНІ		ФІЗИЧНІ		ЧИСЛОВІ
Істинність	ТАК (TRUE)	+5 В	ВИСОКИЙ (HIGH) рівень сигналу	1
Хибність	НІ (FALSE)	0 В	НИЗЬКИЙ (LOW) рівень сигналу	0

Рис. 1.3. Еквівалентні значення змінних логічних елементів цифрових пристроїв

Отже, в цифровій схемотехніці для визначення одного з двох значень логічних змінних та для оцінки одного з двох дискретних станів входних і вихідних сигналів логічних елементів застосовується *двійкові числа 1 і 0*.

4°. Двійкові числа 1 і 0 є *структурними* елементами *цифрової логіки* (Практикум 2), які дозволяють природно впорядкувати множину значень логічних змінних. Наприклад, для логічних висловлювань, наведених вище в 1.1.1.1° (Рис. 1.1) і 1.1.1.2° (Рис. 1.2) впорядковані *набори* значень 1 і 0 логічних змінних представлені на Рис. 1.4. а) і Рис. 1.4. б) відповідно. Варто зазначити, що ці набори, як і логічні висловлювання, для яких вони складені, *еквівалентні* (2.1.2.3°).

ЯКЩО $A \vee B$, ТО F
 $A \vee B = F$

N	A	B	F
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

а)

ЯКЩО $\bar{A} \wedge \bar{B}$, ТО $\bar{F}$
 $\bar{A} + \bar{B} = \bar{F}$

N	$\bar{A}$	$\bar{B}$	$\bar{F}$
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

б)

Впорядкування набору розмірністю $n=2$
 N - номер набору

$(AB)_2 = N_{10}$	N	A	B	F
$00_2 = 0_{10}$	0	0	0	x
$01_2 = 1_{10}$	1	0	1	x
$10_2 = 2_{10}$	2	1	0	x
$11_2 = 3_{10}$	3	1	1	x

в)

Рис. 1.4. Двійкові значення 1 і 0 змінних логічних функцій та їх впорядковані набори

Число вхідних змінних n називають *розмірністю* набору. Число наборів складає $n_N = 2^n$. *Нумерація* набору $N = 0, \dots, 2^n - 1$ починається з нуля і співпадає з впорядкованим двійковим числом значень вхідних змінних $(x_2 x_1)_2$, переведеним в десяткове: $00_2 = 0_{10}$, $01_2 = 1_{10}$, $10_2 = 2_{10}$, $11_2 = 3_{10}$ (Рис. 1.4. в).

Впорядкований набір двійкових значень 1 і 0 змінних логічної функції називається *таблицею істинності* функції. Таблиця істинності з цифровими значеннями змінних функції – один з основних способів визначення логічної функції, в тому числі при визначенні комбінацій значень змінних на входах і виходах цифрових логічних елементів.

5°. Числа 1 і 0 цифрової логіки не варто плутати з числами 1 і 0 *двійкової позиційної системи числення* (1.2.1.1°), яка послуговує для представлення (кодування) чисел.

1.2. ПОЗИЦІЙНІ СИСТЕМИ ЧИСЛЕННЯ

1.2.1. Двійкова позиційна система числення

1°. Двійкову систему числення, в якій для кодування числа використовуються в якості числових знаків лише два символи, цифри 0 і 1, називають системою числення з *основою* 2. В позиційних системах числення один і той же знак (цифра або буква) означає різне число залежно від його *розряду* (позиції знаку в коді числа). Розряд має *чисельну вагу*, яка кратна основі системи числення. Вага розряду представляється числом звичної десяткової системи числення (Рис. 1.5).



Рис. 1.5. Вага розрядів в восьмирозрядному двійковому числовому коді

2°. Якщо число A_b системи числення з основою b ($b = 2$ для двійкової системи числення) представити як код з n числових знаків a_i з порядковими номерами їх розрядів (позицій в коді числа) i , які відраховуються від 1 до n справа наліво від молодшого до старшого розряду:

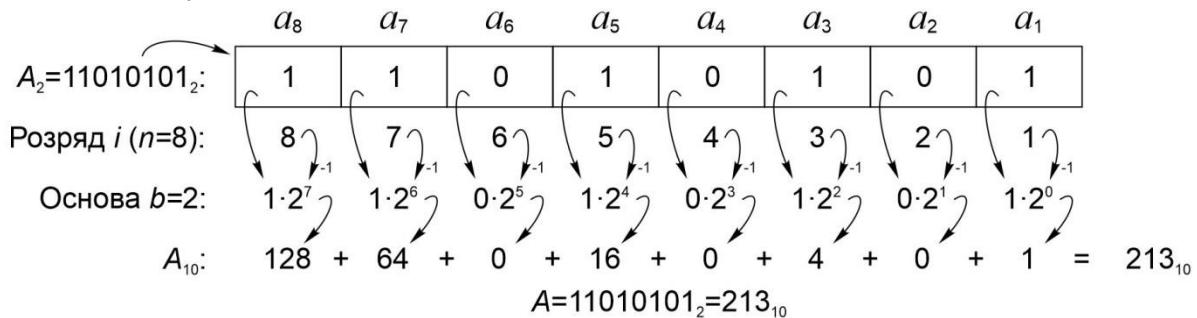
$$A_b \sim a_n \dots a_i \dots a_3 a_2 a_1; \quad (1.1.)$$

то число A_b можна перетворити в число A_{10} десятикової системи числення як

$$A_b \rightarrow A_{10} = \sum_{i=1}^n a_i b^{i-1} = a_1 + a_2 b + a_3 b^2 + \dots + a_i b^{i-1} + \dots + a_n b^{n-1}. \quad (1.2.)$$

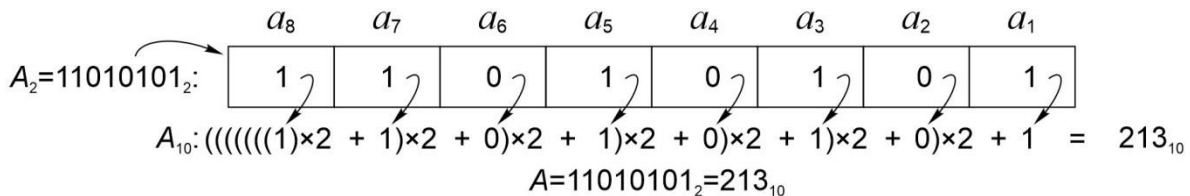
Тоді двійкове число A_2 перетворюється в десятикове число A_{10} (Рис. 1.6) як

$$A_2 \rightarrow A_{10} = \sum_{i=1}^n a_i 2^{i-1} = a_1 + 2a_2 + 4a_3 + \dots + 2^{i-1}a_i + \dots + 2^{n-1}a_n; \quad (1.3.)$$


 Рис. 1.6. Приклад перетворення двійкового числа 11010101_2 в десятикове

3°. Перетворення (1.2) і (1.3) можна представити в вигляді, який має назву *схеми Горнера*. Для двійкового числа схема Горнера виглядає (Рис. 1.7) як

$$A_2 \rightarrow A_{10} = (((((a_n \cdot 2 + a_{n-1}) \cdot 2 + a_{n-2}) \dots) \cdot 2 + a_2) \cdot 2 + \dots + a_1. \quad (1.4.)$$


 Рис. 1.7. Перетворення двійкового числа 11010101_2 в десятикове по схемі Горнера

4°. Для перетворення десятикового числа A_{10} в число A_b системи числення з основою b необхідно ділити десятикове число A_{10} на основу b з залишками a_i . Порядковий номер i відраховується від першої операції ділення ($i = 1, a_i = a_1$) поки частка c_{n-1} не стане менше основи b ($c_{n-1} < b$). Тоді наступне ділення з

порядковим номером n ($i = n, a_i = a_n$) буде останнім та дасть частку $c_n = 0$ і залишок $a_n = c_{n-1}$. При цьому число A_b записується як запис (1.1).

При перетворенні десяткового числа A_{10} в двійкове число A_2 (Рис. 1.8) завжди неділима частка і останній залишок рівні 1 ($c_{n-1} = a_n = 1$), тому для такого перетворення запис (1.1.) можна представити як

$$A_{10} \rightarrow A_2 \sim 1a_{n-1} \dots a_i \dots a_3a_2a_1. \quad (1.5.)$$

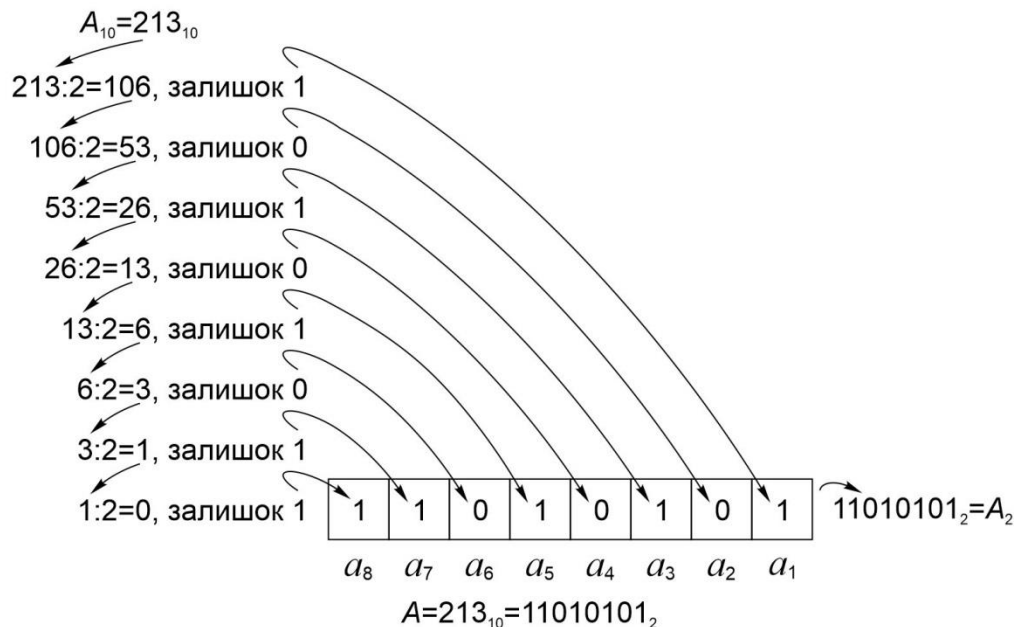


Рис. 1.8. Перетворення десяткового числа 213_{10} в двійкове

5°. Під розрядністю цифрових пристроїв зазвичай розуміється розрядність чисел двійкової системи числення. Один розряд двійкової системи числення називають *бітом*. 8 біт утворюють один *байт*, 16 біт або 2 байти утворюють 16-ти розрядне *слово*, 4 байти – 32-х розрядне, 8 байт – 64-х розрядне. Байтами і байтовими словами оперують сучасні цифрові пристрої, від простих 8-ми розрядних мікроконтролерів до потужних персональних комп'ютерів з 64-ти розрядними процесорами. Для компактного запису багаторозрядних двійкових чисел застосовуються вісімкова та шістнадцяткова (зазвичай) системи числення.

1.2.2. Вісімкова і шістнадцяткова позиційні системи числення

1°. Якщо в двійковому числі виділити по три або чотири розряди справа наліво (від молодшого розряду до старшого), то отримуються так звані двійкові *триплети* a_i^8 або *квартети* a_i^{16} відповідно (Рис. 1.9).

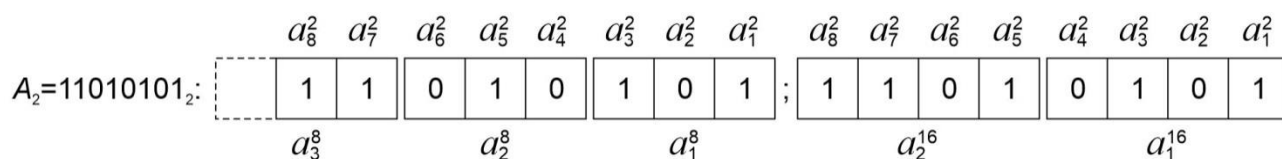


Рис. 1.9. Триплети і квартети восьмирозрядного двійкового числа 11010101_2

2°. Якщо застосувати вісімкову позиційну систему числення з основою $b = 8$, в якій для кодування числа використовується 8 символів – цифри 0, 1, 2, 3, 4, 5, 6, 7, то будь-який двійковий триплет з 3-х знаків (1.2.2.1°, Рис. 1.9) можна замінити на один знак вісімкової системи (Рис. 1.10).

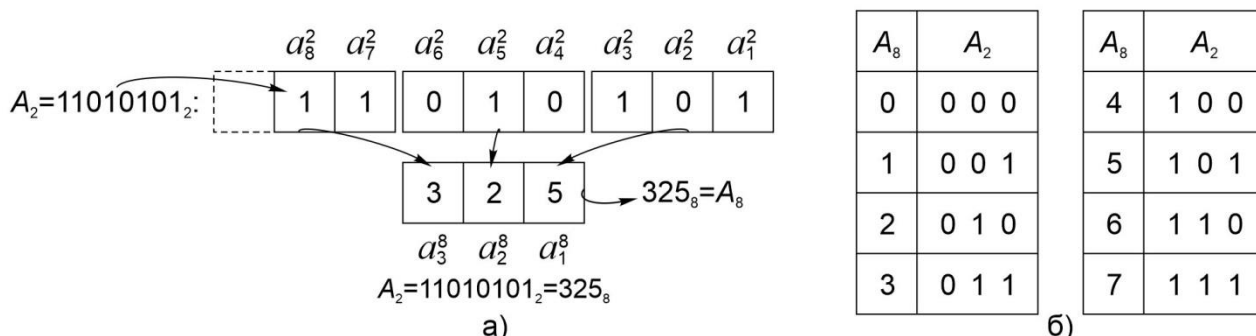


Рис. 1.10. Вісімкова система числення, де б) перетворення двійкового числа 11010101₂ в вісімкове, а) кодова таблиця вісімкових (A_8) і двійкових (A_2) чисел

При перетворенні двійкового числа A_2 в вісімкове A_8 (Рис. 1.10. а) і навпаки, вісімкового числа A_8 в двійкове A_2 , використовуються відповідні значення з кодової таблиці (Рис. 1.10. б).

Проте, жодне з поширених 8-ми, 16-ти, 32-х та 64-х розрядних двійкових чисел не розбивається на повноцінні триплети. Наприклад, триплет a_3^8 8-ми розрядного двійкового числа складається лише з 2-х знаків (Рис. 1.9, Рис. 1.10. а). Вказаний недолік вісімкової системи числення практично обмежив її сучасне застосування. Свого часу вісімкові числа ефективно застосовувались при кодуванні команд для 8-ми розрядних персональних обчислювальних машин.

3°. В шістнадцятковій позиційній системі числення з основою $b = 16$ використовується шістнадцять символів як числових знаків – десять цифр 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 і шість букв А, В, С, D, E, F, які відповідно визначають числа 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15.

Кодові таблиці, в яких одному шістнадцятковому знаку відповідає чотири двійкових знаки (Рис. 1.11), застосовуються при перетворенні двійкового числа A_2 в шістнадцяткове A_{16} (Рис. 1.12. а) і навпаки, шістнадцяткового числа A_{16} в двійкове A_2 , (Рис. 1.12. б). З таблиць вибираються відповідні значення чисел.

A_{16}	A_2	A_{16}	A_2	A_{16}	A_2	A_{16}	A_2
0	0 0 0 0	4	0 1 0 0	8	1 0 0 0	C	1 1 0 0
1	0 0 0 1	5	0 1 0 1	9	1 0 0 1	D	1 1 0 1
2	0 0 1 0	6	0 1 1 0	A	1 0 1 0	E	1 1 1 0
3	0 0 1 1	7	0 1 1 1	B	1 0 1 1	F	1 1 1 1

Рис. 1.11. Кодова таблиця шістнадцяткових (A_{16}) і двійкових (A_2) чисел

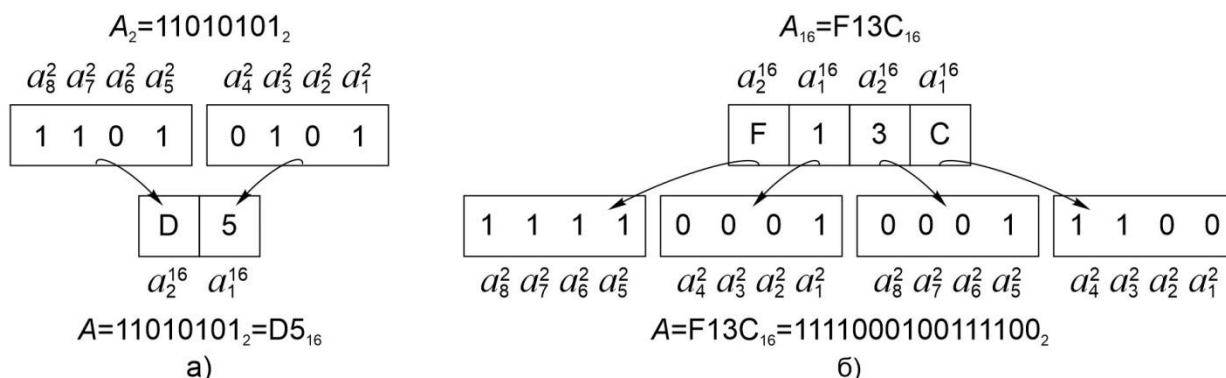


Рис. 1.12. Перетворення чисел, де а) однобайтового двійкового числа 11010101_2 в шістнадцяткове, б) шістнадцяткового числа $A13C_{16}$ в двобайтове двійкове

4°. Шістнадцяткове число A_{16} перетворюється в десяткове число A_{10} згідно (1.2) як (Рис. 1.13. а):

$$A_{16} \rightarrow A_{10} = \sum_{i=1}^n a_i 16^{i-1} = a_1 + 16a_2 + \dots + 16^{i-1}a_i + \dots + 16^{n-1}a_n. \quad (1.6.)$$

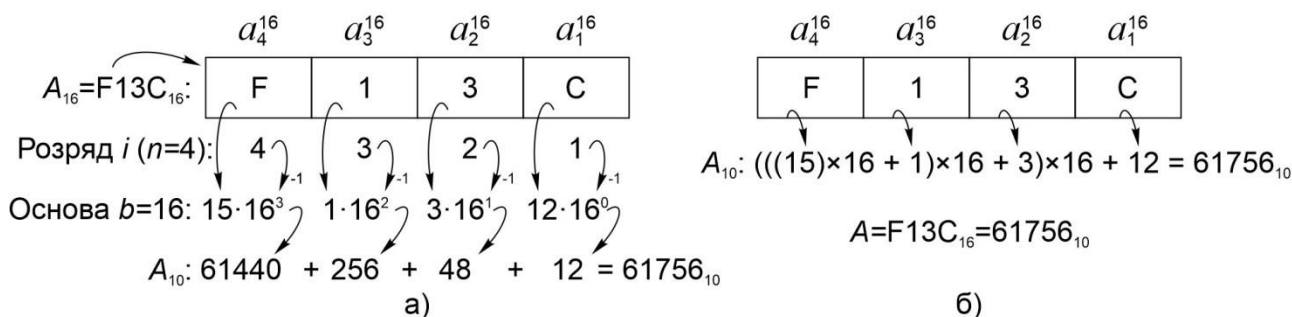


Рис. 1.13. Перетворення шістнадцяткового числа $F13C_2$ в десяткове:
 а) згідно (1.6); б) згідно (1.7)

Перетворення шістнадцяткового числа A_{16} в десяткове число A_{10} можна здійснити по схемі Горнера (1.2.1.3°) як (Рис. 1.13. б):

$$A_{16} \rightarrow A_{10} = (((...((a_n \cdot 16 + a_{n-1}) \cdot 16 + a_{n-2}) \dots) \cdot 16 + a_2) \cdot 16 + \dots + a_1. \quad (1.7.)$$

5°. Перетворення десяткового числа A_{10} в шістнадцяткове число A_{16} відбувається по тій же методиці, що і перетворення десяткового числа A_{10} в двійкове число A_2 (1.2.1.4°) (Рис. 1.14).

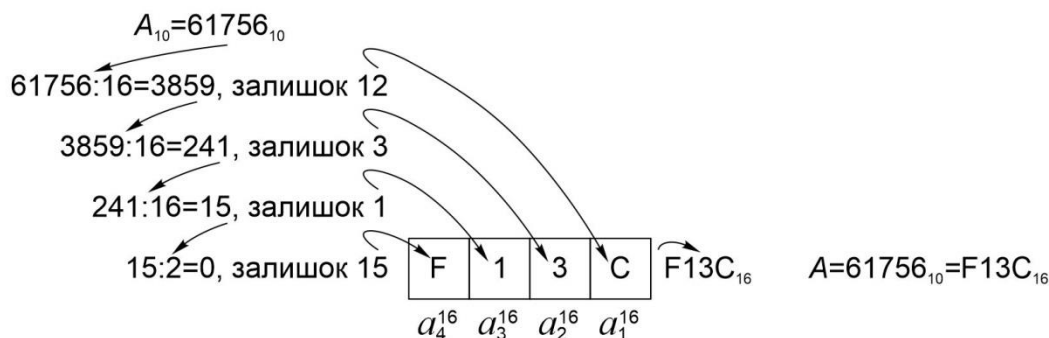


Рис. 1.14. Перетворення десяткового числа 61756_{10} в шістнадцяткове

1.3. АРИФМЕТИЧНІ ОПЕРАЦІЇ НАД ДВІЙКОВИМИ ЧИСЛАМИ

1.3.1. Двійкові числа зі знаком, додатковий код (доповнення)

1°. В двійковій системі числення для чисел зі знаком на знак числа вказує *знаковий розряд* – старший розряд запису числа. Цифра 0 в знаковому розряді вказує на *позитивні* числа, а цифра 1 на *негативні*. Максимальне десяткове число, яке можна записати n розрядним двійковим числом, враховуючи знаковий розряд, складає $2^{n-1} - 1$. Наприклад, для числа розрядів $n = 4$ це десяткове число $2^3 - 1 = 7$ (записи 4-х розрядних чисел зі знаком показані на рисунку 1.15).

A_{10}	a_4^2	a_3^2	a_2^2	a_1^2
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0
	1	1	1	1
-1	1	1	1	0
-2	1	1	0	1
-3	1	1	0	0
-4	1	0	1	1

a_4^2

– знаковий розряд

0

– позитивні числа

1

– негативні числа

A_{10}	a_4^2	a_3^2	a_2^2	a_1^2
4	0	1	0	0
3	0	0	1	1
2	0	0	1	0
1	0	0	0	1
0	0	0	0	0
	0	0	0	0
-1	1	1	1	1
-2	1	1	1	0
-3	1	1	0	1
-4	1	1	0	0

а)
б)

Рис. 1.15. Чотирьох розрядні двійкові числа зі знаком, негативні числа яких представлені в додатковому коді як а) доповнення до 1, б) доповнення до 2

2°. Для зручності здійснення арифметичних операцій над числами негативні числа записуються в *додатковому коді з неповним доповненням* (доповнення до 1) або з *повним доповненням* (доповнення до 2 – Рис. 1.15. б). Для запису негативного числа з доповненням до 1 в рівному по модулю позитивному числі 0 замінюється на 1, а 1 на 0 (Рис. 1.15. а). При додаванні 1 (+1) до негативного числа з доповненням до 1 отримується негативне число з доповненням до 2 (Рис. 1.15. б).

1.3.2. Операції додавання і віднімання двійкових чисел

1°. Правила додавання двійкових чисел (Рис. 1.16. а) прості та ідентичні правилам додавання десяткових чисел. Нижче наведені приклади проведення операцій додавання двійкових чисел (Рис. 1.16. б).

а)
$$\begin{array}{r} 0 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ + 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 0 \\ + 1 \\ \hline 1 \end{array} \quad \begin{array}{r} 1 \\ 1 \\ + 1 \\ \hline 0 \end{array} \leftarrow \text{Перенесення 1}$$

б)
$$\begin{array}{r} 10 \\ + 101 \\ \hline 111 \end{array} \left(\begin{array}{l} 2_{10} \\ + 5_{10} \\ \hline 7_{10} \end{array} \right) \quad \begin{array}{r} 1000 \\ + 11 \\ \hline 1011 \end{array} \left(\begin{array}{l} 8_{10} \\ + 3_{10} \\ \hline 11_{10} \end{array} \right) \quad \begin{array}{r} 1010 \\ + 101 \\ \hline 1101 \end{array} \left(\begin{array}{l} 10_{10} \\ + 3_{10} \\ \hline 13_{10} \end{array} \right) \quad \begin{array}{r} 1101 \\ + 1101 \\ \hline 1101 \end{array} \left(\begin{array}{l} 6_{10} \\ + 7_{10} \\ \hline 13_{10} \end{array} \right)$$

Рис. 1.16. Додавання двійкових чисел, де а) правила додавання, б) приклади

2°. Правила віднімання двійкових чисел (Рис. 1.17. а) також прості та ідентичні правилам віднімання десяткових чисел. Нижче наведені приклади проведення операцій віднімання двійкових чисел (Рис. 1.17. б).

а)
$$\begin{array}{r} 0 \\ - 0 \\ \hline 0 \end{array} \quad \begin{array}{r} 1 \\ - 0 \\ \hline 1 \end{array}$$

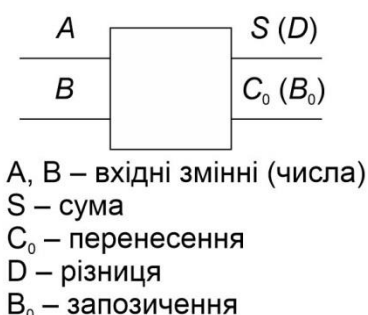
б)
$$\begin{array}{r} 1101 \\ - 1001 \\ \hline 100 \end{array} \left(\begin{array}{l} 13_{10} \\ - 9_{10} \\ \hline 4_{10} \end{array} \right) \quad \begin{array}{r} 101 \\ - 10 \\ \hline 11 \end{array} \left(\begin{array}{l} 5_{10} \\ - 2_{10} \\ \hline 3_{10} \end{array} \right)$$

Рис. 1.17. Віднімання двійкових чисел, де а) правила додавання, б) приклади

3°. Правила додавання двійкових чисел (Рис. 1.16. а) можна представити в вигляді таблиці істинності логічних функцій (1.1.2.4°), в якій доданки є вхідними змінними A і B та яка має дві вихідні змінні (функції) – суму S і перенесення C_0 (Рис. 1.18. а). Фізично такі логічні функції реалізуються одним логічним елементом (Рис. 1.18. б), названим неповним суматором, який є основою багаторозрядних суматорів (Практикум 5).

N	A	B	S	C_0
0	0	0	0	0
1	0	1	1	0
2	1	0	1	0
3	1	1	0	1

а)



б)

N	A	B	D	B_0
0	0	0	0	0
1	0	1	1	1
2	1	0	1	0
3	1	1	0	1

в)

Рис. 1.18. Таблиці істинності логічних елементів додавання і віднімання, де а) таблиця додавання, б) логічний елемент в) таблиця для віднімання

4°. Правила віднімання двійкових чисел (Рис. 1.17. а) теж можна представити в вигляді таблиці істинності логічних функцій (Рис. 1.18. в) та реалізувати фізично логічним елементом (Рис. 1.18. б). Проте, для спрощення схемотехнічних рішень зручно мати один універсальний елемент. Тому операції віднімання теж виконують суматори – логічні елементи, в яких реалізована операція дода-

вання. При цьому двійкові числа записуються в додатковому коді з доповненням (1.3.1.2°) по певним правилам. Нижче наведені приклади проведення операцій віднімання і додавання над двійковими числами з доповненнями (Рис. 1.19).

Знаковий розряд 0 - позитивне число
1 - негативне число

Циклічний зсув

Нехтується

а) б) в) г) д) е)

Рис. 1.19. Операції над двійковими числами з додатковим кодом, де а-в) з доповненням до 1, г-е) з доповненням до 2

5°. Отже, операції віднімання двійкових позитивних чисел та операції додавання і віднімання двійкових негативних чисел в цифрових пристроях виконуються через операцію додавання. При цьому виконуються наступні правила. Двійкові числа, які при операції додавання зберігають знак «мінус» (-), записуються в додатковому коді з доповненням (1.3.1.2°). Далі відбувається операція по правилам додавання двійкових чисел (1.3.2.1°). Якщо при додаванні чисел з доповненням до 1 відбувається перенесення зі знакового розряду, то здійснюється *циклічний зсув* – до отриманої суми додається 1 (Рис. 1.19. б). При додаванні чисел з доповненням до 2 перенесенням зі знакового розряду нехтується (Рис. 1.18. д).

1.4. ВПРАВИ І ЗАВДАННЯ

1.4.1. Вправи для виконання

1°. Перевести десяткове число A_{10} в двійкове A_2 (варіанти).

Варіант	A_{10}	Варіант	A_{10}	Варіант	A_{10}	Варіант	A_{10}
1.	1009 ₁₀	6.	1040 ₁₀	11.	1087 ₁₀	16.	1117 ₁₀
2.	1218 ₁₀	7.	1297 ₁₀	12.	1404 ₁₀	17.	1484 ₁₀
3.	1151 ₁₀	8.	1362 ₁₀	13.	1277 ₁₀	18.	1091 ₁₀
4.	1548 ₁₀	9.	1456 ₁₀	14.	1372 ₁₀	19.	1530 ₁₀
5.	1013 ₁₀	10.	1493 ₁₀	15.	1231 ₁₀	20.	1493 ₁₀

2°. Перевести двійкове число A_2 в десяткове A_{10} (варіанти).

Варіант	A_2	Варіант	A_2	Варіант	A_2	Варіант	A_2
1.	01101100 ₂	6.	10101100 ₂	11.	11000110 ₂	16.	11100010 ₂
2.	11010011 ₂	7.	10010011 ₂	12.	00111001 ₂	17.	10011101 ₂
3.	11110000 ₂	8.	01010011 ₂	13.	00011101 ₂	18.	01000001 ₂
4.	10001111 ₂	9.	01001000 ₂	14.	10101011 ₂	19.	11011001 ₂
5.	00110110 ₂	10.	11001100 ₂	15.	11001010 ₂	20.	10011011 ₂

3°. Перевести десяткове число A_{10} в шістнадцяткове A_{16} (варіанти).

Варіант	A_{10}	Варіант	A_{10}	Варіант	A_{10}	Варіант	A_{10}
1.	1091 ₁₀	6.	1277 ₁₀	11.	1151 ₁₀	16.	1372 ₁₀
2.	1530 ₁₀	7.	1009 ₁₀	12.	1548 ₁₀	17.	1231 ₁₀
3.	1493 ₁₀	8.	1218 ₁₀	13.	1013 ₁₀	18.	1456 ₁₀
4.	1087 ₁₀	9.	1117 ₁₀	14.	1493 ₁₀	19.	1297 ₁₀
5.	1404 ₁₀	10.	1484 ₁₀	15.	1040 ₁₀	20.	1362 ₁₀

4°. Перевести двійкове число A_2 в шістнадцяткове A_{16} (варіанти).

Варіант	A_2	Варіант	A_2	Варіант	A_2	Варіант	A_2
1.	00011101 ₂	6.	10011011 ₂	11.	10001111 ₂	16.	11110000 ₂
2.	10101011 ₂	7.	11000110 ₂	12.	11011001 ₂	17.	01010011 ₂
3.	01000001 ₂	8.	00111001 ₂	13.	11100010 ₂	18.	01001000 ₂
4.	10101100 ₂	9.	01101100 ₂	14.	10011101 ₂	19.	11001100 ₂
5.	10010011 ₂	10.	11010011 ₂	15.	00110110 ₂	20.	11001010 ₂

5°. Перевести шістнадцяткове число A_{16} в десяткове A_{10} (варіанти).

Варіант	A_{16}	Варіант	A_{16}	Варіант	A_{16}	Варіант	A_{16}
1.	A5 F6 ₁₆	6.	13 0B ₁₆	11.	3B 06 ₁₆	16.	05 55 ₁₆
2.	05 0C ₁₆	7.	20 1E ₁₆	12.	A4 46 ₁₆	17.	10 00 ₁₆
3.	FF 13 ₁₆	8.	71 9C ₁₆	13.	6F 5A ₁₆	18.	AA 11 ₁₆
4.	12 13 ₁₆	9.	AB BA ₁₆	14.	A2 10 ₁₆	19.	C1 53 ₁₆
5.	E1 1E ₁₆	10.	09 04 ₁₆	15.	90 1E ₁₆	20.	20 23 ₁₆

6°. Перевести шістнадцяткове число A_{16} в двійкове A_2 (варіанти).

Варіант	A_{16}	Варіант	A_{16}	Варіант	A_{16}	Варіант	A_{16}
1.	3B 06 ₁₆	6.	6F 5A ₁₆	11.	FF 13 ₁₆	16.	90 1E ₁₆
2.	13 0B ₁₆	7.	A2 10 ₁₆	12.	12 13 ₁₆	17.	AB BA ₁₆
3.	20 1E ₁₆	8.	AA 11 ₁₆	13.	E1 1E ₁₆	18.	09 04 ₁₆
4.	71 9C ₁₆	9.	C1 53 ₁₆	14.	05 55 ₁₆	19.	A5 F6 ₁₆
5.	A4 46 ₁₆	10.	20 23 ₁₆	15.	10 00 ₁₆	20.	05 0C ₁₆

7°. Виконати арифметичні операції (1.8.) над 4-х розрядними двійковими числами зі знаком: Операції віднімання виконати через операцію додавання з записом чисел, де необхідно, в додатковому коді з доповненням до 1. Операції віднімання повторити з записом чисел в додатковому коді з доповненням до 2.

$$A_2 + B_2; A_2 - B_2; (-A_2) - (-B_2). \quad (1.8.)$$

Числа A_2 і B_2 для вправи представлені в десятковій системі числення як A_{10} і B_{10} (варіанти).

Варіант	A_{10}	B_{10}
1.	4	3
2.	3	4
3.	2	5
4.	3	3
5.	2	2
6.	6	1
7.	4	2
8.	2	4

1.4.2. Контрольне завдання

1°. Виконати вправи 1.4.1.1° ÷ 1.4.1.7° для одного варіанту. Варіант вибирається за номером по списку групи. Для чергового номеру, що першим перевершує число варіантів, вибирається варіант 1, і так далі.

2°. По результатам виконання вправ оформити звіт. На початку звіту вказується номер практикуму і його тема, номер групи і прізвище виконавця. Виконані вправи повинні містити умови завдання та його розв'язання за відповідною до даного практикуму нумерацією.

ПРАКТИКУМ 2. ЦИФРОВА ЛОГІКА

2.1. ОСНОВНІ ПОЛОЖЕННЯ ЦИФРОВОЇ ЛОГІКИ

2.1.1. Аналітичне і табличне представлення логічних функцій

1°. Аналіз і синтез цифрових логічних схем проводиться з застосуванням математичного апарату *алгебри логіки (булевої алгебри)*.

В булевій алгебрі будь-які логічні висловлювання (Практикум 1) представляються аналітично математичними формулами – *логічними виразами*, в яких висловлювання (*вхідні і вихідна змінна*) позначаються зазвичай *буквами латинського алфавіту* (великими і/або малими, без індексації і/або з індексацією), а словесні сполучники та/чи частки (*логічні операції*) умовно позначаються *символьними знаками*. Наприклад, для розглянутого логічного висловлювання (1.1.1.1°) аналітичний логічний вираз може виглядати як

$$F = A \times B, \quad \text{або} \quad y = x_1 \wedge x_2, \quad (2.1.)$$

а для логічного висловлювання (1.1.1.2°) як

$$\bar{F} = \bar{A} + \bar{B}, \quad \text{або} \quad \neg y = \neg x_1 \vee \neg x_2, \quad (2.2.)$$

де $y = F$ – вихідна змінна (*функція*), $x_1 = A, x_2 = B$ – вхідні змінні (*аргументи функції*), $\times (\wedge), + (\vee), \bar{F} (\neg)$ – символічне позначення логічних операцій (Рис. 2.1).

ОСНОВНІ ЛОГІЧНІ ОПЕРАЦІЇ					
№	Найменування операції	Булева алгебра	Символ операції	Приклади логічних виразів	Логічна функція
1	Логічне множення	Кон'юнкція	$\times, \wedge, \cdot, \&, \text{б/с}$	$A \times B, A \wedge B, A \cdot B \cdot C, ABC$	I (AND)
2	Логічне додавання	Диз'юнкція	$+, \vee$	$A+B, A \vee B \vee C$	АБО (OR)
3	Логічне заперечення	Інверсія	$\bar{}, \neg$	$\bar{A}, \neg B, \overline{A \wedge B}, \bar{A} + \bar{B} + \bar{C}$	НЕ (NOT)

Рис. 2.1. Основні логічні операції

2°. Операції *кон'юнкції, диз'юнкції і інверсії є основними* логічними операціями булевої алгебри (Рис. 2.1). Будь-які інші логічні операції є похідними, їх можна здійснити через комбінації основних операцій кон'юнкції, диз'юнкції і інверсії.

Логічна функція I (AND), яка виконує операцію кон'юнкції (логічного множення), є істинною лише тоді, коли істинними є всі її аргументи, і хибною у всіх інших випадках – коли хоча б один її аргумент хибний (Рис. 2.2. а). Логічна функція АБО (OR), яка виконує операцію диз'юнкції (логічного додавання), є

істинною тоді, коли хоча б один її аргумент істинний, і хибною лише у випадку, коли всі її аргументи хибні (Рис. 2.2. б). Логічна функція НЕ (NOT), яка виконує операцію інверсії (логічного заперечення), є істинною тоді, коли її аргумент хибний, і хибною, коли її аргумент істинний (Рис. 2.2. в). Результат унарної операції інверсії $\bar{A}$ (над одним операндом A) називають *доповненням* до операнду (до аргументу функції) A .

3°. Крім словесних логічних висловлень (1.1.1.1°, 1.1.1.2°) і аналітичних логічних формул (2.1, 2.2) логічні функції можна визначити табличним способом – *таблицями істинності*, які є впорядкованими наборами (1.1.2.4°) двійкових чисел 1 і 0 (Рис. 2.2) – істинних і хибних значень змінних логічної функції.

Кон'юнкція $A \times B = F$

N	A	B	F
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

Функція І (AND)

а)

Диз'юнкція $A + B = F$

N	A	B	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1

Функція АБО (OR)

б)

Інверсія $\bar{A} = F$

N	A	F
0	0	1
1	1	0

Функція НЕ (NOT)

в)

Нумерація наборів N

N	A	B
$0_{10} = (00)_2$	0	0
$1_{10} = (01)_2$	0	1
$2_{10} = (10)_2$	1	0
$3_{10} = (11)_2$	1	1

г)

Рис. 2.2. Таблиці істинності основних логічних функцій з нумерацією наборів N , де а) функція І (AND), б) функція АБО (OR), в) функція НЕ (NOT). г) нумерація наборів N

Оскільки десяткове число номеру набору рівне двійковому числу впорядкованих значень вхідних змінних (Рис. 2.2. г), нумерація наборів в таблицях істинності зазвичай опускається.

4°. Для будь-якого аналітичного логічного виразу (логічної функції), жорстко дотримуючись послідовності операцій (Рис. 2.3. а), можна скласти таблицю істинності (Рис. 2.3. б),.

Послідовність операцій

Пріоритет	Операція	
1	$\bar{}$	Інверсія одного операнду
2	()	Операція в дужках
3	$\times$	Кон'юнкція
4	$+$	Диз'юнкція

а)

Функція $F = \bar{A} \times B + A \times \bar{B}$

N	A	B	$\bar{A}$	$\bar{B}$	$\bar{A} \times B$	$A \times \bar{B}$	$\bar{A} \times B + A \times \bar{B} = F$
0	0	0	1	1	0	0	0
1	0	1	1	0	1	0	1
2	1	0	0	1	0	1	1
3	1	1	0	0	0	0	0

б)

Рис. 2.3. Таблиця істинності логічної функції, де а) послідовність виконання операцій, б) приклад складання таблиці істинності для функції $F = \bar{A} \times B + A \times \bar{B}$

5°. Для будь-якої заданої таблиці істинності можна визначити аналітичний вираз логічної функції в *нормальній диз'юнктивній формі* (Рис. 2.4. а) та/або в *нормальній кон'юнктивній формі* (Рис. 2.4. б).

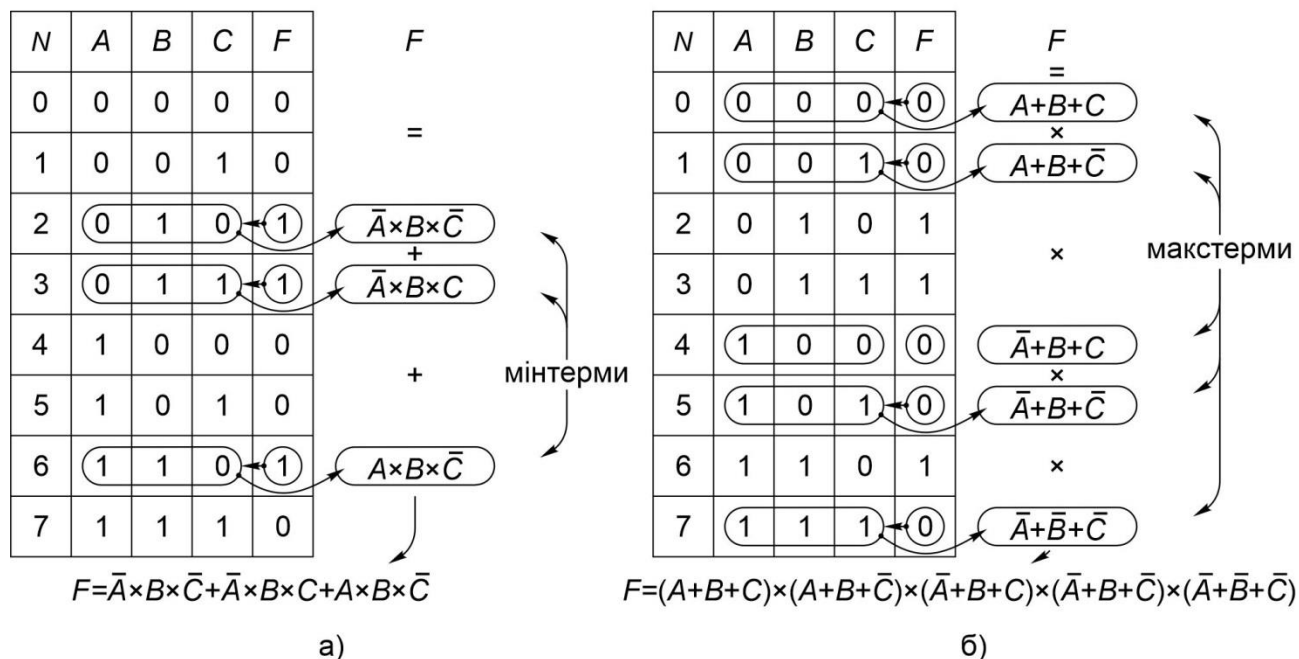


Рис. 2.4. Приклад визначення виразу логічної функції з таблиці істинності, де
а) нормальна диз'юнктивна форма представлення логічної функції,
б) нормальна кон'юнктивна форма представлення логічної функції

Ідеальна нормальна диз'юнктивна форма запису функції отримується як сума (диз'юнкція) *мінтермів* – добутків (кон'юнкцій) всіх вхідних змінних (аргументів функції або їх доповнень) для наборів, в яких функція приймає значення 1. Наприклад, для заданої таблиці істинності (Рис. 2.4) функція приймає значення 1 в наборах 2, 3 і 6. Тоді функція записується в ідеальній нормальній диз'юнктивній формі як диз'юнкція мінтермів наборів 2, 3 і 6 (Рис. 2.4. а) так:

$$F = \bar{A} \times B \times \bar{C} + A \times B \times \bar{C} + \bar{A} \times B \times C. \quad (2.3.)$$

Ідеальна нормальна кон'юнктивна форма запису функції отримується як добуток (кон'юнкція) *макстермів* – сум (диз'юнкцій) всіх вхідних змінних (аргументів функції або їх доповнень) для наборів, в яких функція приймає значення 0. Для заданої таблиці істинності (Рис. 2.4.) функція приймає значення 0 в наборах 0, 1, 4, 5 і 7. Тоді функція записується в ідеальній нормальній кон'юнктивній формі як кон'юнкція макстермів наборів 0, 1, 4, 5 і 7 (Рис. 2.4. б) так:

$$F = (A + B + C) \times (A + B + \bar{C}) \times (\bar{A} + B + C) \times (\bar{A} + B + \bar{C}) \times (\bar{A} + \bar{B} + \bar{C}). \quad (2.4.)$$

Кон'юнкції (2.3) і диз'юнкції (2.4) ідеальних форм запису завжди містять повний набір вхідних змінних (всіх аргументів функції або їх доповнень).

В наведеному прикладі для коротшого запису функції з меншим числом операцій варто надати перевагу нормальній диз'юнктивній формі запису (2.3.).

2.1.2. Правила обчислень логічних виразів. Закон Моргана

1°. Аналітичні вирази логічних функцій в ідеальних нормальних формах запису (2.1.1.5°) можна *спростити* або *перетворити*. Кон'юнкції і диз'юнкції спрощених виразів містять, як правило, не повний набір аргументів функцій і їх доповнень. Такі кон'юнкції і диз'юнкції називають *елементарними*. Спрощення може значно скоротити запис виразу функції, особливо при кількості аргументів функції три і більше та при значній кількості задіяних мінтермів чи макстермів. Спрощення може бути бажаним для зменшення кількості логічних елементів при фізичній реалізації логічної функції в цифрових пристроях. Враховуючи промислово досягнутий високий рівень інтеграції активних і пасивних елементів (транзисторів і резисторів тощо) на одному кристалі мікросхеми для виконання логічних операцій, спрощення виразів логічних функцій для цифрової схемотехніки не є самоціллю. Важливим може бути перетворення виразів логічних функцій до виду зручного і зрозумілого для полегшення вивчення, аналізу та діагностики цифрових схем. Для спрощення та перетворення логічних аналітичних виразів необхідні знання правил і законів алгебри логіки.

2°. З таблиць істинності (Рис. 2.2) випливають постулати (Рис. 2.5) операцій логічного множення (кон'юнкції), логічного додавання (диз'юнкції) і логічного заперечення (інверсії) над двійковими значеннями логічних змінних кожної з основних функцій І (AND), АБО (OR), НЕ (NOT) відповідно.

Функція І (AND)	Функція АБО (OR)	Функція НЕ (NOT)
$0 \times 0 = 0$ (П.1.)	$0 + 0 = 0$ (П.5.)	$\bar{0} = 1$ (П.9.)
$0 \times 1 = 0$ (П.2.)	$0 + 1 = 1$ (П.6.)	$\bar{1} = 0$ (П.10.)
$1 \times 0 = 0$ (П.3.)	$1 + 0 = 1$ (П.7.)	
$1 \times 1 = 1$ (П.4.)	$1 + 1 = 1$ (П.8.)	

Рис. 2.5. Проведення логічних операцій над двійковими змінними логічних функцій

3°. Правила обчислень для однієї логічної змінної A при виконанні логічних операцій кожної з основних логічних функцій І (AND), АБО (OR), НЕ (NOT) зведені окремо (Рис. 2.6).

Функція І (AND)	Функція АБО (OR)	Функція НЕ (NOT)
$A \times 0 = 0$ (П.11.)	$A + 0 = A$ (П.16.)	$\bar{\bar{A}} = A$ (П.21.)
$A \times 1 = A$ (П.12.)	$A + 1 = 1$ (П.17.)	
$A \times A = A$ (П.13.)	$A + A = A$ (П.18.)	
$A \times \bar{A} = 0$ (П.14.)	$A + \bar{A} = 1$ (П.19.)	
$\bar{A} \times \bar{A} = \bar{A}$ (П.15.)	$\bar{A} + \bar{A} = \bar{A}$ (П.20.)	

Рис. 2.6. Правила обчислень для однієї логічної змінної логічних функцій

4°. Для булевої алгебри також справедливі *асоціативні, комутативні і дистрибутивні* правила, відомі зі звичайної алгебри як *сполучний закон, закон переміщення та розподільний закон* відповідно (Рис. 2.7).

Закон	Функція І (AND)	Функція АБО (OR)
Асоціативний	$(A \times B) \times C = (A \times C) \times B$ (П.22.)	$(A + B) + C = (A + C) + B$ (П.25.)
Комутативний	$A \times B = B \times A$ (П.23.)	$A + B = B + A$ (П.26.)
Дистрибутивний	$A \times (B + C) = A \times B + A \times C$ (П.24.)	$A + B \times C = (A + B) \times (A + C)$ (П.27.)

Рис. 2.7. Асоціативний, комутативний та дистрибутивний закони булевої алгебри

5°. З попереднього переліку постулатів, правил і законів слідують правила обчислень, подані нижче без зведення для певних функцій (Рис. 2.8).

$A \times (\bar{A} + B) = A \times B$ (П.28.)	$A \times B + A \times \bar{B} = A$ (П.30.)	$A + \bar{A} \times B = A + B$ (П.33.)
$\bar{A} \times (A + B) = \bar{A} \times B$ (П.29.)	$\bar{A} \times B + A \times B = \bar{A} + B$ (П.31.)	$\bar{A} + A \times B = \bar{A} + B$ (П.34.)
	$\bar{A} \times \bar{B} + A \times B = \bar{A} + \bar{B}$ (П.32.)	

Рис. 2.8. Правила обчислень при виконанні логічних операцій

6°. Важливим законом алгебри логіки є закон Моргана (Рис. 2.9), який можна сформулювати через два постулати: доповнення добутку змінних дорівнює сумі доповнень цих змінних (Рис. 2.9, П.35); доповнення суми змінних дорівнює добутку доповнень цих змінних (Рис. 2.9, П.36).

$$\overline{A \times B} = \bar{A} + \bar{B} \quad (\text{П.35.}) \qquad \overline{A + B} = \bar{A} \times \bar{B} \quad (\text{П.36.})$$

Рис. 2.9. Закон Моргана

2.2. ПЕРЕТВОРЕННЯ І СПРОЩЕННЯ ЛОГІЧНИХ ВИРАЗІВ

2.2.1. Аналітичне перетворення логічних виразів

1°. Аналітичне перетворення виразів логічних функцій здійснюється з застосуванням постулатів, правил та законів булевої алгебри (Рис. 2.5÷Рис. 2.9, П1÷П36).

Перетворення логічних виразів часто здійснюється для спрощення кінцевого виразу логічної функції. Наприклад, вираз (2.3) в ідеальній нормальній диз'юнктивній формі запису логічної функції, таблиця істинності якої представлена на Рис. 2.4, можна перетворити з застосуванням законів і правил булевої алгебри (П.22, П.23, П.24, П.19 і П.34) та отримати такий спрощений вираз:

$$\begin{aligned}
 F &= \bar{A} \times B \times \bar{C} + \bar{A} \times B \times C + A \times B \times \bar{C} = \\
 &= (\bar{A} + A) \times (B \times \bar{C}) + \bar{A} \times B \times C = B \times \bar{C} + \bar{A} \times B \times C = \\
 &= B \times (\bar{C} + \bar{A} \times C) = B \times (\bar{C} + \bar{A}) = \bar{A} \times B + B \times \bar{C}.
 \end{aligned} \tag{2.5.}$$

2°. Кон'юнкції в отриманому виразі (2.5) не містять повний набір змінних (аргументів функції або їх доповнень) порівняно з початковим виразом (2.3). Такі кон'юнкції називаються *елементарними*.

Аналогічно (2.5.), перетворивши початковий вираз (2.4) в ідеальній нормальній кон'юнктивній формі запису логічної функції, можна отримати спрощений вираз:

$$\begin{aligned}
 F &= (A + B + C) \times (A + B + \bar{C}) \times (\bar{A} + B + C) \times (\bar{A} + B + \bar{C}) \times (\bar{A} + \bar{B} + \bar{C}) = \\
 &= (A + B + C \times \bar{C}) \times (\bar{A} + B + C \times \bar{C}) \times (\bar{A} + \bar{B} + \bar{C}) = \\
 &= (A + B) \times (\bar{A} + B) \times (\bar{A} + \bar{B} + \bar{C}) = (A \times \bar{A} + B) \times (\bar{A} + \bar{B} + \bar{C}) = \\
 &= B \times (\bar{A} + \bar{B} + \bar{C}) = \bar{A} \times B + B + \bar{C}
 \end{aligned} \tag{2.6}$$

Після перетворення виразу в ідеальній нормальній кон'юнктивній формі запису (2.4) спрощений вираз (2.6) також містить елементарні кон'юнкції, рівносильні отриманим в спрощеному виразі (2.5) після перетворення виразу в ідеальній нормальній диз'юнктивній формі запису (2.3). Це є доказом еквівалентності функцій, представлених в різних формах запису – диз'юнктивній (2.3) та кон'юнктивній (2.4).

2.2.2. Матрично-графічне представлення логічних функцій. Карти Карно

1°. Логічні функції з числом аргументів (вхідних змінних) два або більше (розмірність наборів таблиці істинності $n \geq 2$) можна представити *матрично-графічним* способом – *картами Карно*.

Карти Карно ефективно застосовуються для отримання спрощених виразів логічних функцій. Для складання карт Карно логічних функцій застосовуються графічні шаблони (Рис. 2.10). Шаблон для функції вибирається залежно від кількості n аргументів функції і містить n_N клітинок по кількості $n_N = 2^n$ наборів в таблиці істинності функції (1.1.2.4°, 2.1.1.3°).

	$\bar{B}$	B
$\bar{A}$	F_0	F_1
A	F_2	F_3

$n = 2, n_N = 4$

а)

	$\bar{B}$		B	
$\bar{A}$	F_0	F_1	F_3	F_2
A	F_4	F_5	F_7	F_6
	$\bar{C}$	C	$\bar{C}$	

$n = 3, n_N = 8$

б)

	$\bar{B}$		B		
$\bar{A}$	F_0	F_2	F_6	F_4	$\bar{D}$
	F_1	F_3	F_7	F_5	D
A	F_9	F_{11}	F_{15}	F_{13}	D
	F_8	F_{10}	F_{14}	F_{12}	$\bar{D}$
	$\bar{C}$	C	$\bar{C}$		

$n = 4, n_N = 16$

в)

Рис. 2.10. Шаблони карт Карно для функцій з числами аргументів n і наборів $N = 2^n$, де: а) $n = 2, n_N = 4$; б) $n = 3, n_N = 8$; в) $n = 4, n_N = 16$

В кожному із клітинок карти Карно записуються двійкові значення F_i функції, де i – номер набору N таблиці істинності, $i = N = 0, 1, \dots, 2^n - 1$. Значення 1 функції в клітинках карти Карно є істинними значеннями мінтермів функції при її записі в нормальній диз'юнктивній формі (2.1.1.5°). Кожний мінтерм до спрощення містить всі аргументи функції ($A, \dots$) або їх доповнення ($\bar{A}, \dots$).

2°. На Рис. 2.11 представлений детальний схемо-графічний порядок заповнення карти Карно для функції з двома аргументами (Рис. 2.11. а, б), детальна графічна схема визначення запису найпростішого виразу заданої функції (Рис. 2.11. в, г, д) та кінцевий результат опрацювання карти Карно (Рис. 2.11. е).

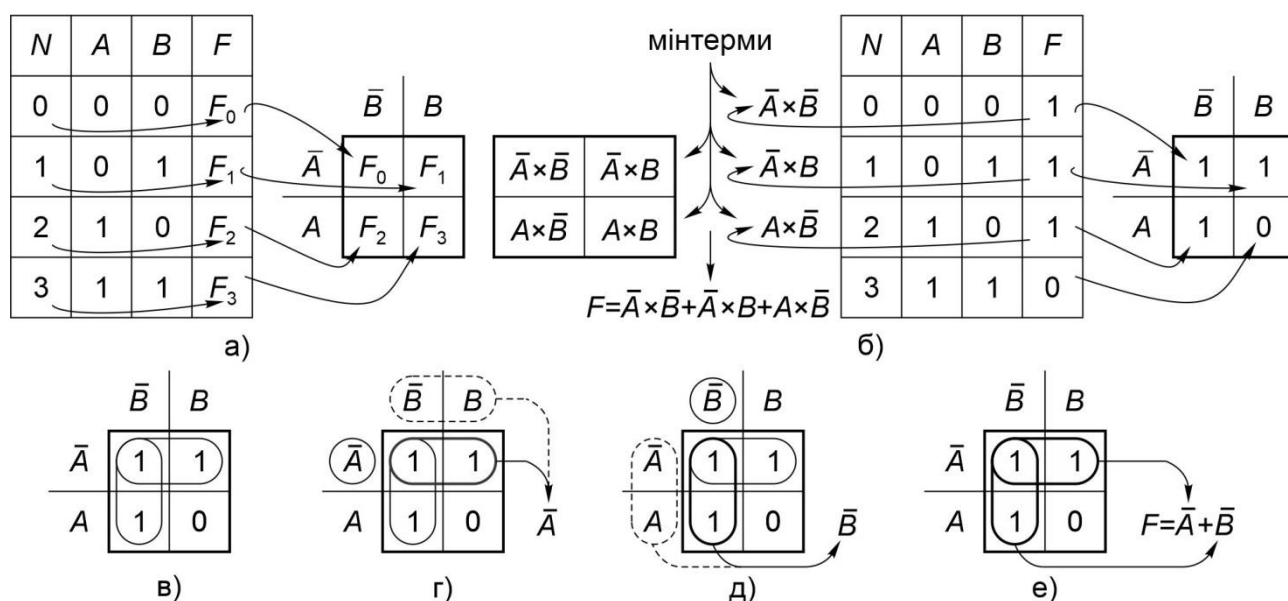


Рис. 2.11. Опрацювання карти Карно логічної функції з числом аргументів $n = 2$

Порядок опрацювання карт Карно наступний. Після заповнення карти (Рис. 2.11. б) згідно шаблону (Рис. 2.11. а) графічно виділяються групи з істинними значеннями заданої функції (значення 1), які є сусідніми по горизонталі і вертикалі (Рис. 2.11. в). По діагоналі групи не виділяються. Одна виділена група може перекривати елементи (мінтерми) іншої виділеної групи. Для мінтермів цих груп визначаються змінні, які приймають значення 0 і 1 – $\bar{B}$ і B (Рис. 2.11. г) та $\bar{A}$ і A (Рис. 2.11. д), якими нехтується. Такі змінні є *несуттєвими* (фіктивними) змінними для виділених в групу мінтермів. *Суттєвими* для виділених в групу мінтермів є лише ті змінні, які входять в ці мінтерми без доповнень або лише своїми доповненнями – $\bar{A}$ (Рис. 2.11. г) та $\bar{B}$ (Рис. 2.11. д). Після опрацювання таким чином карти Карно отримується найкоротший вираз (Рис. 2.11. е) заданої логічної функції. Опрацювання карт Карно для функцій з двома аргументами достатньо просте. Застосування карт Карно найбільш ефективно для спрощення виразів функцій з трьома і чотирма аргументами. Для функцій з більшим числом аргументів карти Карно значно ускладнюються.

3°. Для карт Карно з числом аргументів три і більше необхідно дотримуватись наступних додаткових правил. Число елементів в виділених групах має бути кратним числу 2 – 2, 4, 8, 2^n . Групи повинні містити максимальну кількість можливих елементів. Групи можуть перекриватись, тобто, один і той же елемент може входити в різні групи. Графічно виділені групи можуть утворювати квадрат чи прямокутник. Допускається загортання групи з одного краю карти на інший (протилежний).

На Рис. 2.12 представлено опрацювання карт Карно для визначення найпростіших виразів логічних функцій з трьома аргументами. При заповненні карт Карно застосовано шаблон (Рис. 2.10. б).

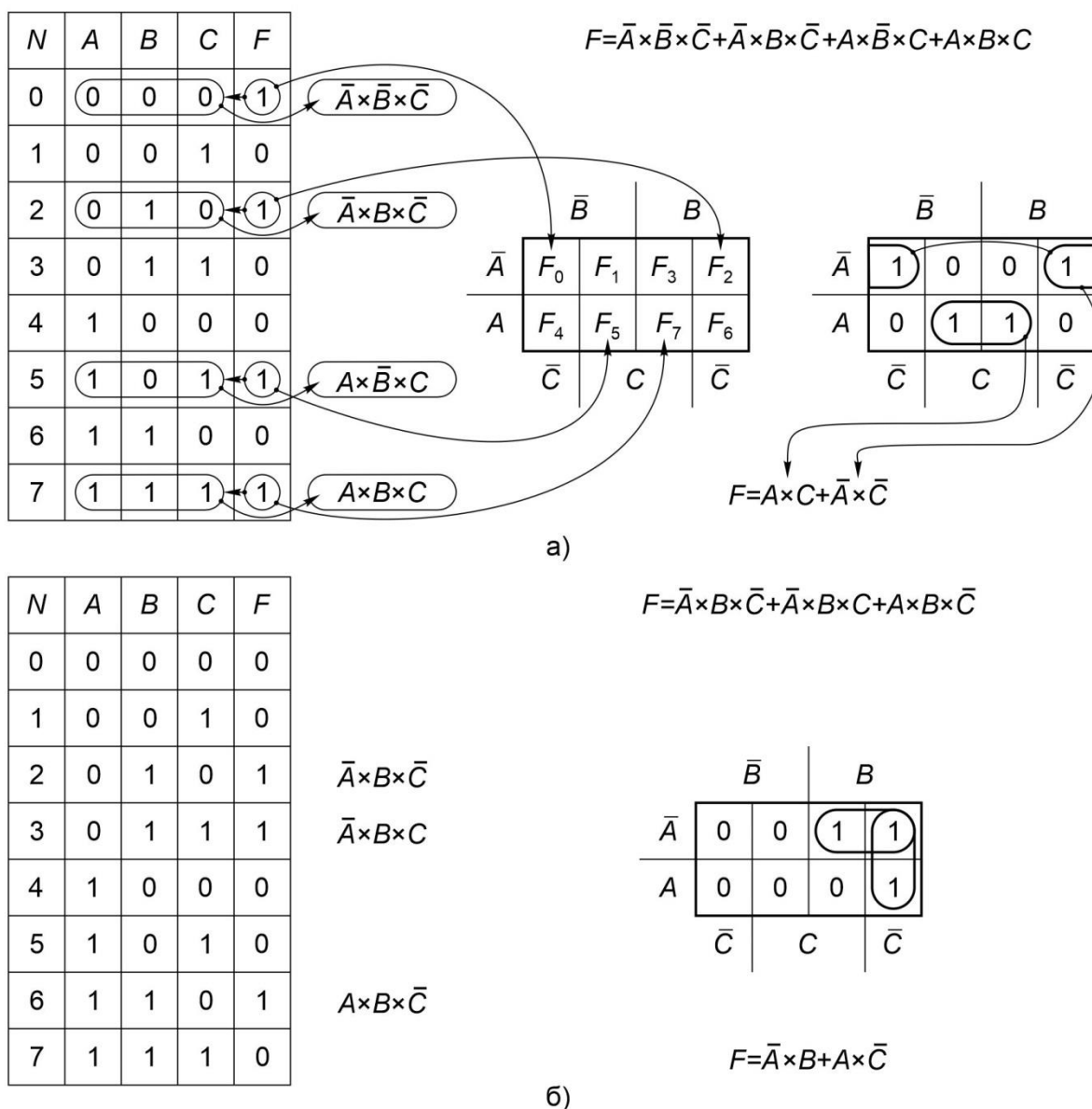


Рис. 2.12. Карты Карно для функций з трьома аргументами, де а) і б) приклади

В першому наведеному прикладі (Рис. 2.12 а) одна з виділених груп загорнута по горизонталі з однієї сторони на іншу. Після опрацювання карти Карно заданої функції (Рис. 2.12) встановлено, що змінна B є для функції несуттєвою.

4°. Інший приклад (Рис. 2.12 б) демонструє без додаткової пояснювальної графіки зрозуміле опрацювання карти Карно заданої вище функції (2.1.1.5°, Рис. 2.4). Спрощений вираз функції, отриманий з застосуванням карти Карно (Рис. 2.12 б) співпадає зі спрощеним виразом, отриманим аналітичним перетворенням з застосуванням правил булевої алгебри (2.2.1.1°, 2.5) виразу в ідеальній нормальній диз'юнктивній формі запису (2.1.1.5°, 2.3), отриманого з таблиці істинності (2.1.1.5°, Рис. 2.4). Порівнюючи ці два способи отримання найпростіших виразів логічної функції, можна дійти висновку, що матрично-графічний спосіб (застосування карт Карно) більш наглядний, тому може бути простішим і надійнішим ніж аналітичний спосіб. Враховуючи те, що тут, при застосуванні карти Карно (Рис. 2.12), визначення мінтермів і виразу в ідеальній нормальній диз'юнктивній формі запису носить довідковий, а не обов'язковий характер, достатньо лише таблиці істинності функції, це стає більш очевидним.

5°. В деякій спеціальній літературі для визначення функції трьох, чотирьох і більше аргументів замість її порівняно громіздкої таблиці істинності (або порівняно довгого запису функції в ідеальній нормальній диз'юнктивній чи кон'юнктивній формі) застосовується лаконічна *числова форма* представлення функції. Наприклад, представлення функції F в числовій формі

$$F(ABCD) = \Sigma(0, 8, 10, 15) \quad (2.7.)$$

читається так: логічна функція F аргументів A, B, C, D є диз'юнкцією (логічною сумою Σ) кон'юнкцій з наборів (1.1.2.4°, Рис. 1.4, Рис. 2.2) за номерами 0, 8, 10, 15. Або по іншому: логічна функція F аргументів A, B, C, D є істинною в наборах за номерами 0, 8, 10, 15.

6°. Нехай, для прикладу, задана складна логічної функції чотирьох аргументів A, B, C, D , яка є істинною в наборах за номерами 4, 5, 9, 12, 13, 14, 15:

$$F(ABCD) = \Sigma(4, 5, 9, 12, 13, 14, 15) \quad (2.8.)$$

При записі (2.8.) логічної функції можна одразу заповнити її карту Карно (Рис. 2.13) для подальшого опрацювання, застосувавши шаблон (Рис. 2.10. в), який складено для належно впорядкованих наборів таблиці істинності.

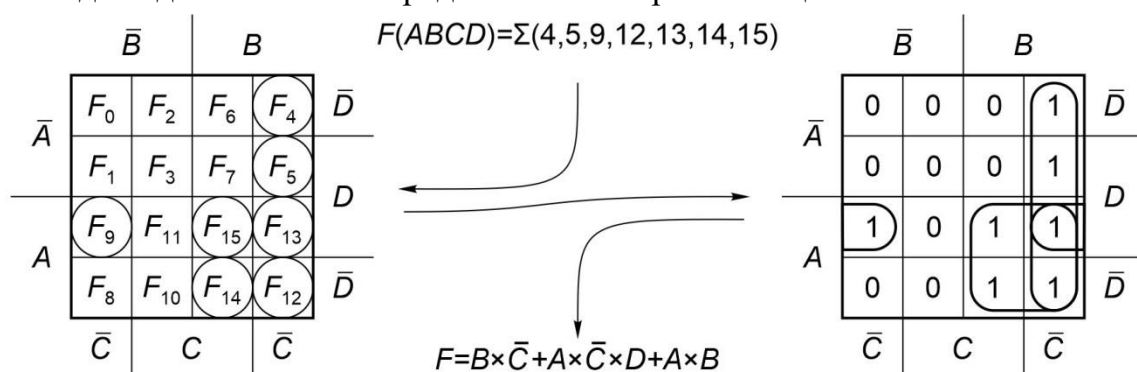


Рис. 2.13. Карта Карно логічної функції $F(ABCD) = \Sigma(4, 5, 9, 12, 13, 14, 15)$

Детальне матрично-графічне опрацювання карти Карно заданої функції представлено на рисунку нижче (Рис. 2.14. в). На рисунку також для наочності показано переведення запису функції в числовій формі в запис функції в ідеальній нормальній диз'юнктивній формі (Рис. 2.14. а), а також складання таблиці істинності з мінтермами по запису функції в числовій формі (Рис. 2.14. б).

$$F(ABCD)=\Sigma(4,5,9,12,13,14,15)$$

$$n = 4, n_N = 16$$

$$F = \bar{A} \times B \times \bar{C} \times \bar{D} + \bar{A} \times B \times \bar{C} \times D + A \times \bar{B} \times \bar{C} \times D + A \times B \times \bar{C} \times \bar{D} + A \times B \times \bar{C} \times D + A \times B \times C \times \bar{D} + A \times B \times C \times D$$

а)

N	A	B	C	D	F
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	0
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	0
8	1	0	0	0	0
9	1	0	0	1	1
10	1	0	1	0	0
11	1	0	1	1	0
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

$$\bar{A} \times B \times \bar{C} \times \bar{D} = F_4$$

$$\bar{A} \times B \times \bar{C} \times D = F_5$$

$$A \times \bar{B} \times \bar{C} \times D = F_9$$

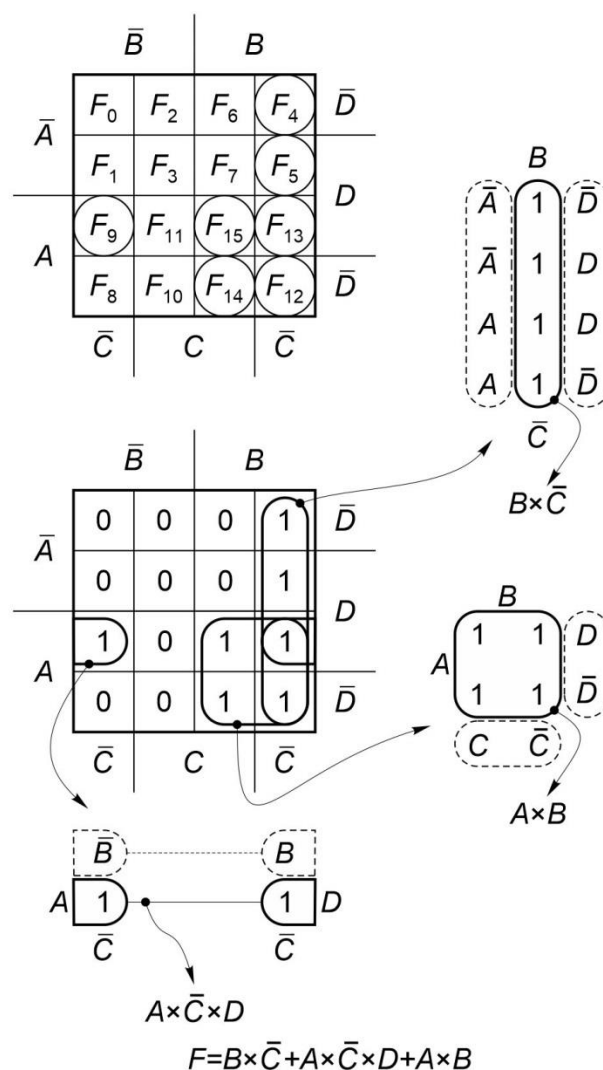
$$A \times B \times \bar{C} \times \bar{D} = F_{12}$$

$$A \times B \times \bar{C} \times D = F_{13}$$

$$A \times B \times C \times \bar{D} = F_{14}$$

$$A \times B \times C \times D = F_{15}$$

б)



в)

Рис. 2.14. Визначення логічної функції чотирьох аргументів по її числовому запису, де а) запис функції в ідеальній нормальній диз'юнктивній формі, б) таблиця істинності функції з мінтермами, в) опрацювання карти Карно функції

В наведеному прикладі в карті Карно графічно виділено три групи елементів (істинних значень 1 функції), – одна група з двома і дві групи з чотирма

елементами. Група з двома елементами загорнута по горизонталі з однієї сторони на іншу. Всі три групи перекривають одна одну, тобто мають спільний елемент (елементи). Для виділених груп несуттєвими є аргументи, в яких елементи групи охоплюють сам аргумент і його доповнення.

Елементи вертикальної прямокутної групи з чотирьох елементів охоплюють аргументи A і D з їх доповненнями $\bar{A}$ і $\bar{D}$, тому для цієї групи ці аргументи несуттєві. Група охоплює також аргумент B і доповнення $\bar{C}$ до аргументу, які складають елементарну кон'юнкцію $B \times \bar{C}$ для цієї групи істинних значень 1 функції.

Елементи горизонтальної прямокутної групи з двох елементів охоплюють аргумент B з його доповненням $\bar{B}$ – для цієї групи аргумент B несуттєвий. Група охоплює також аргументи A і D і доповнення $\bar{C}$ до аргументу, які складають елементарну кон'юнкцію $A \times \bar{C} \times D$ для цієї групи.

Нарешті, елементи квадратної групи з чотирьох елементів охоплюють аргументи C і D з їх доповненнями $\bar{C}$ і $\bar{D}$, тому для цієї групи ці аргументи несуттєві. Група охоплює також аргументи A і B , які складають елементарну кон'юнкцію $A \times B$ для цієї групи.

Після опрацювання карти Карно запис функції є диз'юнкцією з трьох елементарних кон'юнкцій, дві з яких містять по два аргументи або їх доповнення, а одна – три аргументи або доповнення (Рис. 2.14. в):

$$F = B \times \bar{C} + A \times \bar{C} \times D + A \times B. \quad (2.9.)$$

Для порівняння, початковий запис функції в ідеальній нормальній формі є диз'юнкцією семи повних кон'юнкцій, які містять всі чотири аргументи або їх доповнення (Рис. 2.14. в):

$$F = \bar{A} \times B \times \bar{C} \times \bar{D} + \bar{A} \times B \times \bar{C} \times D + A \times \bar{B} \times \bar{C} \times D + A \times B \times \bar{C} \times \bar{D} + A \times B \times \bar{C} \times D + A \times B \times C \times \bar{D} + A \times B \times C \times D. \quad (2.10.)$$

Порівнюючи вирази (2.9.) і (2.10.) можна дійти висновку, що початковий запис функції в ідеальній нормальній формі містить 31 логічну операцію, а спрощений – всього 7 операцій.

2.3. ВПРАВИ І ЗАВДАННЯ

2.3.1. Вправи для виконання

1°. Скласти вирази в ідеальній нормальній диз'юнктивній і кон'юнктивній формі запису логічної функції $F(ABCD)$ чотирьох аргументів A, B, C, D . Функція $F(ABCD)$ задана в числовій формі запису (варіанти).

Варіант	$F(ABCD)$
1.	$\Sigma(1, 5, 9, 13, 15)$
2.	$\Sigma(0, 6, 8, 9, 13, 14)$
3.	$\Sigma(1, 3, 4, 5, 12, 13)$
4.	$\Sigma(2, 3, 6, 7, 10, 11, 14, 15)$
5.	$\Sigma(4, 7, 10, 11, 15)$
6.	$\Sigma(2, 5, 7, 10, 13, 15)$
7.	$\Sigma(1, 2, 5, 9, 10, 13)$
8.	$\Sigma(1, 3, 6, 9, 11, 14, 15)$
9.	$\Sigma(0, 2, 3, 4, 7, 9, 10, 11, 13, 14)$
10.	$\Sigma(0, 3, 4, 7, 8, 11, 12, 15)$
11.	$\Sigma(1, 2, 5, 6, 11, 15)$
12.	$\Sigma(2, 3, 10, 11, 14, 15)$
13.	$\Sigma(1, 3, 5, 7, 8, 11, 15)$
14.	$\Sigma(1, 6, 8, 9, 12, 13)$
15.	$\Sigma(1, 3, 5, 7, 9, 11, 13, 15)$
16.	$\Sigma(0, 4, 6, 12, 14)$
17.	$\Sigma(3, 6, 9, 13, 14)$
18.	$\Sigma(0, 1, 4, 5, 6, 7, 8, 11, 12)$
19.	$\Sigma(1, 5, 6, 7, 9, 12, 13, 14, 15)$
20.	$\Sigma(0, 2, 3, 6, 7, 14)$

Чи варто з двох складених виразів надати перевагу одному з них для запису функції? Якщо так, то чому?

2°. Скласти вираз в нормальній формі запису логічної функції $F(ABCD)$ чотирьох аргументів A, B, C, D . Спростити вираз шляхом перетворення, засто-

сувавши правила і закони алгебри логіки. Функція $F(ABCD)$ задана в числовій формі запису (варіанти).

Варіант	$F(ABCD)$
1.	$\Sigma(1, 3, 5, 7, 9, 11, 13, 15)$
2.	$\Sigma(1, 3, 6, 9, 11, 14, 15)$
3.	$\Sigma(0, 1, 4, 5, 6, 7, 8, 11, 12)$
4.	$\Sigma(1, 2, 5, 6, 11, 15)$
5.	$\Sigma(2, 3, 10, 11, 14, 15)$
6.	$\Sigma(0, 4, 6, 12, 14)$
7.	$\Sigma(0, 2, 3, 4, 7, 9, 10, 11, 13, 14)$
8.	$\Sigma(3, 6, 9, 13, 14)$
9.	$\Sigma(4, 7, 10, 11, 15)$
10.	$\Sigma(1, 3, 4, 5, 12, 13)$
11.	$\Sigma(0, 6, 8, 9, 13, 14)$
12.	$\Sigma(1, 5, 9, 13, 15)$
13.	$\Sigma(1, 2, 5, 9, 10, 13)$
14.	$\Sigma(2, 3, 6, 7, 10, 11, 14, 15)$
15.	$\Sigma(0, 2, 3, 6, 7, 14)$
16.	$\Sigma(0, 3, 4, 7, 8, 11, 12, 15)$
17.	$\Sigma(1, 5, 6, 7, 9, 12, 13, 14, 15)$
18.	$\Sigma(2, 5, 7, 10, 13, 15)$
19.	$\Sigma(1, 6, 8, 9, 12, 13)$
20.	$\Sigma(1, 3, 5, 7, 8, 11, 15)$

3°. Скласти вираз в нормальній формі запису логічної функції $F(ABCD)$ чотирьох аргументів A, B, C, D . Спростити вираз шляхом складання і обробки карти Карно. Функція $F(ABCD)$ задана в числовій формі запису (варіанти).

Варіант	$F(ABCD)$
1.	$\Sigma(1, 6, 8, 9, 12, 13)$
2.	$\Sigma(0, 4, 6, 12, 14)$
3.	$\Sigma(1, 2, 5, 9, 10, 13)$
4.	$\Sigma(1, 5, 6, 7, 9, 12, 13, 14, 15)$

Варіант	$F(ABCD)$
5.	$\Sigma(0, 2, 3, 6, 7, 14)$
6.	$\Sigma(1, 3, 6, 9, 11, 14, 15)$
7.	$\Sigma(1, 2, 5, 6, 11, 15)$
8.	$\Sigma(1, 3, 5, 7, 8, 11, 15)$
9.	$\Sigma(2, 5, 7, 10, 13, 15)$
10.	$\Sigma(2, 3, 6, 7, 10, 11, 14, 15)$
11.	$\Sigma(3, 6, 9, 13, 14)$
12.	$\Sigma(0, 2, 3, 4, 7, 9, 10, 11, 13, 14)$
13.	$\Sigma(1, 3, 5, 7, 9, 11, 13, 15)$
14.	$\Sigma(0, 3, 4, 7, 8, 11, 12, 15)$
15.	$\Sigma(2, 3, 10, 11, 14, 15)$
16.	$\Sigma(0, 1, 4, 5, 6, 7, 8, 11, 12)$
17.	$\Sigma(4, 7, 10, 11, 15)$
18.	$\Sigma(1, 3, 4, 5, 12, 13)$
19.	$\Sigma(1, 5, 9, 13, 15)$
20.	$\Sigma(0, 6, 8, 9, 13, 14)$

2.3.2. Контрольне завдання

1°. Виконати вправи 2.3.1.1° ÷ 2.3.1.3° для одного варіанту. Варіант вибирається за номером по списку групи. Для чергового номеру, що першим перевершує число варіантів, вибирається варіант 1, і так далі.

2°. По результатам виконання вправ оформити звіт. На початку звіту вказується номер практикуму і його тема, номер групи і прізвище виконавця. Виконані вправи нумеруються відповідно нумерації даного практикуму. Вправи повинні містити умови завдання і його розв'язання, а також таблицю істинності заданої функції. Перед здачею звіту перевірити по таблиці істинності правильність результатів, отриманих при виконанні вправ.

ПРАКТИКУМ 3. ЛОГІЧНІ ФУНКЦІЇ

3.1. ДОВІДКОВІ ТЕОРЕТИЧНІ ВІДОМОСТІ

3.1.1. Функції пустої множини аргументів і функції одного аргументу

1°. Логічні функції можна представити *вербально* логічними висловлюваннями (1.1.1.1°), *аналітично* математичними виразами (2.1.1.1°) та *таблично* таблицями істинності (2.1.1.3°). Таблиця істинності складається з впорядкованих *наборів* двійкових значень вхідних змінних (аргументів функції) і значення вихідної змінної (функції). Для числа наборів $n_N = 2^n$ (n – *розмірність* набору, число аргументів) загальне число функцій складає $n_F = 2^{n_N}$ (Рис. 3.1).

ЧИСЛО ФУНКЦІЙ n_F ДЛЯ ЧИСЛА АРГУМЕНТІВ n							
Число аргументів функції	n	0	1	2	3	4	5
Число наборів	$n_N = 2^n$	1	2	4	8	16	32
Число логічних функцій	$n_F = 2^{n_N}$	2	4	16	256	65 536	4 294 967 296

Рис. 3.1. Загальне число n_F функцій для числа n аргументів функції

2°. Таблиці істинності функцій $F_i(\emptyset)$ без аргументів ($n = 0$, $\emptyset$ – пуста множина аргументів) складаються з одного набору ($n_N = 2^n = 2^0 = 1$), по яким можна визначити дві функції ($n_F = 2^{n_N} = 2^1 = 2$).

Дві функції $F_i(\emptyset)$ без аргументів (Рис. 3.2), $F_0(\emptyset) = 0$ і $F_1(\emptyset) = 1$, не залежать від змінних, завжди мають постійні значення 0 і 1 відповідно. Функція $F_0(\emptyset) = 0$ називається *нульовою* або *абсолютно хибною функцією*, а функція $F_1(\emptyset) = 1$ – *одиначною* або *абсолютно істинною функцією*.

$F_0(\emptyset)=0$			$F_1(\emptyset)=1$		
№	F	Абсолютно хибна функція (Нульова функція) Константа 0	№	F	Абсолютно істинна функція (Одиначна функція) Константа 1
0	0		0	1	

Рис. 3.2. Функції $F_i(\emptyset)$ пустої множини аргументів ($n = 0$, $n_N = 1$, $n_{F(\emptyset)} = 2$)

3°. Таблиці істинності функцій $F_i(A)$ з одним аргументом A складаються з двох наборів, по яким можна визначити чотири функції (Рис. 3.3).

Дві з чотирьох функцій, $F_0(A) = F_0(\emptyset) = 0$ і $F_3(A) = F_1(\emptyset) = 1$, які не залежать від змінної A , називаються *втраченими функціями* для змінної A , а

змінна A називається *не суттєвою (фіктивною) змінною* для цих функцій. Ці функції є *не втраченими функціями* пустої множини аргументів (Рис. 3.2).

Абсолютно хибна функція (Нульова функція) Константа 0 $F_0(A)=0$	Функція заперечення (Інверсія) Функція НЕ $F_1(A)=\bar{A}$	Функція повторення (Тотожність) Функція A $F_2(A)=A$	Абсолютно істинна функція (Одинична функція) Константа 1 $F_3(A)=1$
№	№	№	№
A	A	A	A
F	F	F	F
0	0	0	0
1	1	1	1

$F=F_0(A)=A \times \bar{A}=0=F_0(\emptyset)$

A

 – не суттєва (фіктивна) змінна A

A

 – суттєва змінна A

$F=F_1(A)=\bar{A}$

$F=F_2(A)=A$

F

 – втрачена функція $F(A)$

F

 – не втрачена функція $F(A)$

$F=F_3(A)=A + \bar{A}=1=F_3(\emptyset)$

Рис. 3.3. Функції $F_i(A)$ одного аргументу A ($n = 1$, $N = 2$, $N_{F(A)} = 4$)

Функція $F_2(A) = A$, значення якої завжди *тотожні* значенню змінної A , називається *функцією повторення*. Функція $F_2(A)$ істинна, якщо істинна змінна A , функція $F_2(A)$ хибна, якщо хибна змінна A .

Функція $F_1(A) = \bar{A}$ називається *функцією заперечення*. Функція $F_1(A)$ істинна, якщо змінна A хибна, функція $F_1(A)$ хибна, якщо змінна A істинна. Функцію заперечення (інверсії) виконує логічний елемент НЕ (NOT) – *інвертор* (4.2.2.3°).

4°. Для будь-якої функції F_j будь-якої множини аргументів $n = 0, 1, 2, \dots$ завжди знайдеться *інверсна функція* F_k так, що

$$F_j = \overline{F_k}, \quad F_k = \overline{F_j} \quad \text{при} \quad j + k = 2^{n_N} - 1 = n_F - 1, \quad (3.1.)$$

де j, k – індекси функції належно впорядкованих наборів таблиць істинності.

Наприклад, для функцій $F_i(\emptyset)$ пустої множини аргументів:

$$j + k = 1; \quad F_0(\emptyset) = \overline{F_1(\emptyset)}. \quad (3.2.)$$

Для функцій $F_i(A)$ одного аргументів:

$$j + k = 3; \quad F_0(A) = \overline{F_3(A)}; \quad F_1(A) = \overline{F_2(A)}. \quad (3.3.)$$

Для функцій $F_i(A, B)$ двох аргументів:

$$j + k = 15; \quad F_0(A, B) = \overline{F_{15}(A, B)}; \quad F_1(A, B) = \overline{F_{14}(A, B)}; \dots \quad (3.4.)$$

3.1.2. Функції двох аргументів

1°. Таблиці істинності функцій $F_i(A, B)$ з двома аргументами A і B складаються з чотирьох наборів, по яким можна визначити шістнадцять функцій (Рис. 3.4), з яких шість функцій є втраченими для двох аргументів (3.1.2.2°).

Нульова функція (Константа 0) $F=F_0(A,B)=0$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	0	2	1	0	0	3	1	1	0	Функція АБО-НЕ (Інверсія диз'юнкції) $F=F_1(A,B)=\overline{A+B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	0	2	1	0	0	3	1	1	0	Функція заборони (Інверсія об. імплікації) $F=F_2(A,B)=\overline{A \leftarrow B} = \overline{A} \times B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	1	2	1	0	0	3	1	1	0	Функція НЕ (Інверсія) $F=F_3(A,B)=\overline{A}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	1	2	1	0	0	3	1	1	0
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	0																																																																																
2	1	0	0																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	0																																																																																
2	1	0	0																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	1																																																																																
2	1	0	0																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	1																																																																																
2	1	0	0																																																																																
3	1	1	0																																																																																
Функція заборони (Інверсія імплікації) $F=F_4(A,B)=\overline{A \rightarrow B} = A \times \overline{B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	0	2	1	0	1	3	1	1	0	Функція НЕ (Інверсія) $F=F_5(A,B)=\overline{B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	0	2	1	0	1	3	1	1	0	Функція не еквівалентності (Не еквіваленція) $F=F_6(A,B)=\overline{A \leftrightarrow B} = A \oplus B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	1	2	1	0	1	3	1	1	0	Функція І-НЕ (Інверсія кон'юнкції) $F=F_7(A,B)=\overline{A \times B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>0</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	1	2	1	0	1	3	1	1	0
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	0																																																																																
2	1	0	1																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	0																																																																																
2	1	0	1																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	1																																																																																
2	1	0	1																																																																																
3	1	1	0																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	1																																																																																
2	1	0	1																																																																																
3	1	1	0																																																																																
Функція І (Кон'юнкція) $F=F_8(A,B)=A \times B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	0	2	1	0	0	3	1	1	1	Функція еквівалентності (Еквіваленція) $F=F_9(A,B)=A \leftrightarrow B = \overline{A \oplus B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	0	2	1	0	0	3	1	1	1	Функція повторення (Тотожність) $F=F_{10}(A,B)=B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	1	2	1	0	0	3	1	1	1	Функція проходження (Імплікація) $F=F_{11}(A,B)=A \rightarrow B = \overline{A} + B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	1	2	1	0	0	3	1	1	1
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	0																																																																																
2	1	0	0																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	0																																																																																
2	1	0	0																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	1																																																																																
2	1	0	0																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	1																																																																																
2	1	0	0																																																																																
3	1	1	1																																																																																
Функція повторення (Тотожність) $F=F_{12}(A,B)=A$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	0	2	1	0	1	3	1	1	1	Функція проходження (Обернена імплікація) $F=F_{13}(A,B)=A \leftarrow B = A + \overline{B}$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	0	2	1	0	1	3	1	1	1	Функція АБО (Диз'юнкція) $F=F_{14}(A,B)=A+B$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	0	1	0	1	1	2	1	0	1	3	1	1	1	Одинична функція (Константа 1) $F=F_{15}(A,B)=1$ <table> <tr><th>№</th><th>A</th><th>B</th><th>F</th></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td><td>1</td></tr> <tr><td>2</td><td>1</td><td>0</td><td>1</td></tr> <tr><td>3</td><td>1</td><td>1</td><td>1</td></tr> </table>	№	A	B	F	0	0	0	1	1	0	1	1	2	1	0	1	3	1	1	1
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	0																																																																																
2	1	0	1																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	0																																																																																
2	1	0	1																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	0																																																																																
1	0	1	1																																																																																
2	1	0	1																																																																																
3	1	1	1																																																																																
№	A	B	F																																																																																
0	0	0	1																																																																																
1	0	1	1																																																																																
2	1	0	1																																																																																
3	1	1	1																																																																																

A

,

B

 – не суттєві (фіктивні) змінні A, B

F

 – втрачена функція F(A)

Рис. 3.4. Функції $F_i(A, B)$ двох аргументів A і B ($n = 2$, $n_N = 4$, $n_F = 16$)

2°. Нульова і одинична функції, $F_0(A, B) = F_0(\emptyset) = 0$ і $F_{15}(A, B) = F_1(\emptyset) = 1$, не залежать від змінних A і B – змінні A і B є не суттєвими (фіктивними) змінними для цих функцій. Функції $F_0(A, B)$ і $F_{15}(A, B)$ є інверсними одна до одної (3.4). Ці функції, $F_0(A, B)$ і $F_{15}(A, B)$, які є втраченими функціями для множини двох аргументів A і B , є функціями пустої множини аргументів (3.1.1.2°).

Для функцій $F_3(A, B) = F_1(A) = \bar{A}$ і $F_{12}(A, B) = F_2(A) = A$, змінна B є не суттєвою, а для функцій $F_5(A, B) = F_1(B) = \bar{B}$ і $F_{10}(B) = F_2(B) = B$ не суттєвою є змінна A . Функція $F_3(A, B)$ інверсна до функції $F_{12}(A, B)$, а функція $F_5(A, B)$ інверсна до функції $F_{10}(A, B)$. Ці чотири функції є втраченими для двох змінних A і B – це функції однієї змінної (3.1.1.3°).

Отже, з шістнадцяти функцій двох аргументів шість функцій втрачені, не втраченими функціями двох аргументів є десять функцій (3.1.2.3°, 3.1.2.4°).

3°. Функція $F_8(A, B) = A \times B$ називається *функцією логічного множення* або *кон'юнкцією*. Функція істинна лише тоді, коли істинні обидві змінні A і B , і хибною у всіх інших випадках – коли обидві змінні A і B або хоча б одна з них хибні. Функцію логічного множення (кон'юнкцію) виконує логічний елемент І (AND) (4.2.2.1°).

Функція $F_{14}(A, B) = A + B$ називається *функцією логічного додавання* або *диз'юнкцією*. Функція істинна тоді, коли обидві змінні A і B або хоча б одна з них істинні, і хибна лише тоді, коли хибні обидві змінні A і B . Функцію логічного додавання (диз'юнкцію) виконує логічний елемент АБО (OR) (4.2.2.2°).

Функція $F_9(A, B) = A \leftrightarrow B = \bar{A} \times \bar{B} + A \times B = \overline{A \oplus B}$ називається *функцією еквівалентності* або *еквіваленцією*. Функція істинна тоді, коли обидві змінні A і B істинні або хибні, і хибна тоді, коли одна із змінних A і B хибна, а інша істинна. Функцію логічної еквівалентності (еквіваленцію) виконує логічний елемент виключне АБО-НЕ (XNOR) (4.2.4.2°).

Функція $F_{11}(A, B) = A \rightarrow B = \bar{A} \times \bar{B} + \bar{A} \times B + A \times B = \bar{A} + B$ називається *функцією проходження від A до B* або *імплікацією*. Функція хибна тоді, коли змінна A істинна, а змінна B хибна, і істинна при інших комбінаціях істинності змінних A і B .

Функція $F_{13}(A, B) = A \leftarrow B = \bar{A} \times \bar{B} + A \times \bar{B} + A \times B = A + \bar{B}$ називається *функцією проходження від B до A* або *оберненою імплікацією*. Функція хибна тоді, коли змінна A хибна, а змінна B істинна, і істинна при інших комбінаціях істинності змінних A і B .

До цих розглянутих функцій інші п'ять не втрачених функцій двох аргументів є інверсними (3.1.1.4°).

4°. Функція $F_7(A, B) = \overline{A \times B} = \bar{A} + \bar{B}$ інверсна до функції $F_8(A, B)$ і називається *функцією заперечення логічного множення* або *інверсією кон'юнкції*. Функція істинна тоді, коли обидві змінні A і B або хоча б одна з них хибні, і хибна лише тоді,

коли істинні обидві змінні A і B . Функцію заперечення логічного множення (інверсію кон'юнкції) виконує логічний елемент І-НЕ (NAND) (4.2.3.1°).

Функція $F_1(A, B) = \overline{A \times B} = \overline{A} \times \overline{B}$ інверсна до функції $F_{14}(A, B)$ і називається *функцією заперечення логічного додавання або інверсією диз'юнкції*. Функція істинна тоді, коли обидві змінні A і B хибні, і хибна тоді, коли обидві змінні A і B або хоча б одна з них істинні. Функцію заперечення логічного додавання (інверсію диз'юнкції) виконує логічний елемент АБО-НЕ (NOR) (4.2.3.2°).

Функція $F_6(A, B) = \overline{A \leftrightarrow B} = A \times \overline{B} + \overline{A} \times B = A \oplus B$ інверсна до функції $F_9(A, B)$, називається *функцією заперечення еквівалентності або інверсією еквіваленції*. Функція істинна тоді, коли одна із змінних A і B хибна, а інша істинна, і хибна тоді, коли обидві змінні A і B істинні або хибні. Функцію заперечення еквівалентності (інверсію еквіваленції) виконує логічний елемент виключне АБО (XOR) (4.2.4.1°).

Функція $F_4(A, B) = \overline{A \rightarrow B} = A \times \overline{B}$ інверсна до функції $F_{11}(A, B)$ і називається *функцією заборони проходження від A до B або інверсією імплікації*. Функція істинна тоді, коли змінна A істинна, а змінна B хибна, і хибна при інших комбінаціях істинності змінних A і B .

Функція $F_2(A, B) = \overline{A \leftarrow B} = \overline{A} \times B$ інверсна до функції $F_{13}(A, B)$ і називається *функцією заборони проходження від B до A або інверсією оберненої імплікації*. Функція істинна тоді, коли змінна A хибна, а змінна B істинна, і хибна при інших комбінаціях істинності змінних A і B .

6°. Функції пустої множини, одного і двох аргументів зведені на Рис. 3.5.

ФУНКЦІЇ $F(\emptyset)$ ПУСТОЇ МНОЖИНИ АРГУМЕНТІВ		
Абсолютно істинна функція (константа 1)	$F=F_1(\emptyset)=1$	
Абсолютно хибна функція (константа 0)	$F=F_0(\emptyset)=0$	
ФУНКЦІЇ $F(A)$ ОДНОГО АРГУМЕНТУ (знак + відзначає реалізований логічний елемент)		
Функція повторення (тотожність)	$F=F_2(A)=A$	
Функція заперечення (інверсія)	$F=F_1(A)=\overline{A}$	+
ФУНКЦІЇ $F(A, B)$ ДВОХ АРГУМЕНТІВ (знаки + відзначають реалізовані логічні елементи)		
Функція логічного множення (кон'юнкція)	$F=F_8(A, B)=A \times B$	+
Функція інверсії кон'юнкції	$F=F_7(A, B)=\overline{A \times B}=\overline{A} + \overline{B}$	+
Функція логічного додавання (диз'юнкція)	$F=F_{14}(A, B)=A + B$	+
Функція інверсії диз'юнкції	$F=F_1(A, B)=\overline{A + B}=\overline{A} \times \overline{B}$	+
Функція проходження від A до B (імплікація)	$F=F_{11}(A, B)=A \rightarrow B=\overline{A} + B$	
Функція інверсії імплікації	$F=F_4(A, B)=\overline{A \rightarrow B}=A \times \overline{B}$	
Функція проходження від B до A (обернена імплікація)	$F=F_{13}(A, B)=A \leftarrow B=A + \overline{B}$	
Функція інверсії оберненої імплікації	$F=F_2(A, B)=\overline{A \leftarrow B}=\overline{A} \times B$	
Функція еквівалентності (еквіваленція)	$F=F_9(A, B)=A \leftrightarrow B=\overline{A \oplus B}=\overline{A} \times \overline{B} + A \times B$	+
Функція інверсії еквіваленції	$F=F_6(A, B)=\overline{A \leftrightarrow B}=A \oplus B=\overline{A} \times B + A \times \overline{B}$	+

Рис. 3.5. Не втрачені функції пустої множини аргументів та одного і двох аргументів

3.1.3. Перемикальні властивості логічних функцій

1°. Із чотирнадцяти не втрачених функцій пустої множини аргументів та функцій одного і двох аргументів сім функцій (відзначені на Рис. 3.5 знаком +) реалізовані базовими логічними елементами (Рис. 3.6), які є основою цифрової схемотехніки (Практикум 4).

ФУНКЦІЯ	ВИРАЗ	ЛОГІЧНИЙ ЕЛЕМЕНТ
Функція заперечення (інверсія)	$F = \bar{A}$	НЕ (NOT)
Функція логічного множення (кон'юнкція)	$F = A \times B$	І (AND)
Функція інверсії кон'юнкції	$F = \overline{A \times B}$	І-НЕ (NAND)
Функція логічного додавання (диз'юнкція)	$F = A + B$	АБО (OR)
Функція інверсії диз'юнкції	$F = \overline{A + B}$	АБО-НЕ (NOR)
Функція еквівалентності (еквіваленція)	$F = \overline{A \oplus B}$	Виключне АБО-НЕ (XNOR)
Функція інверсії еквіваленції	$F = A \oplus B$	Виключне АБО (XOR)

Рис. 3.6. Функції, реалізовані логічними елементами

2°. Логічні елементи можна технічно реалізувати завдяки перемикальним властивостям логічних функцій. Фізичну інтерпретацію двійкових значень логічних змінних 1 і 0, як електричних сигналів високого і низького рівня відповідно (1.1.2.3°, Рис. 1.3), можна отримати застосуванням контактних електричних схем (Рис. 3.7, Рис. 3.8).

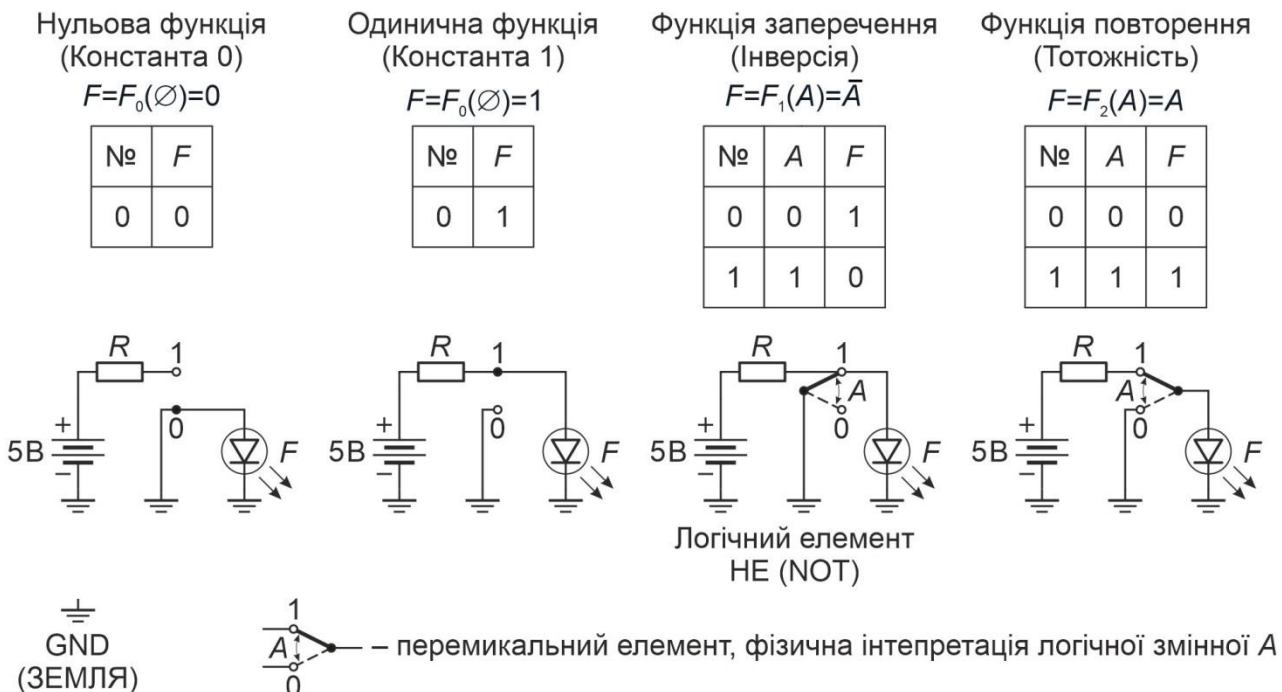


Рис. 3.7. Еквівалентні перемикальні схеми логічних функцій пустої множини аргументів і одного аргументу

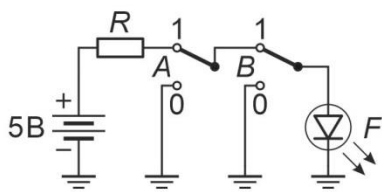
На еквівалентних перемикальних схемах (Рис. 3.7, Рис. 3.8) істинному значенню ТАК (TRUE) логічної функції відповідає ВИСОКИЙ (HIGH) рівень елект-

ричного сигналу на аноді світлодіода (світло випромінюється), а хибному значенню HI (FALSE) – НИЗЬКИЙ (LOW) рівень сигналу (світло не випромінюється). ВИСОКИЙ (HIGH) рівень електричного сигналу на аноді світлодіода забезпечується при його підключенні (перемиканні) до джерела живлення 5 В, а НИЗЬКИЙ (LOW) рівень сигналу – при підключенні (перемиканні) анода світлодіода до загального провідника ЗЕМЛЯ (GND).

Функція логічного множення
(Кон'юнкція)

$$F=F_8(A,B)=A \times B$$

№	A	B	F
0	0	0	0
1	0	1	0
2	1	0	0
3	1	1	1

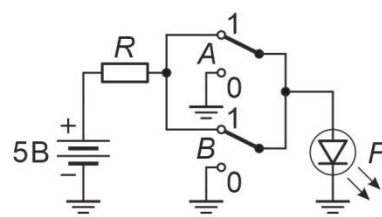


Логічний елемент І (AND)

Функція логічного додавання
(Диз'юнкція)

$$F=F_{14}(A,B)=A+B$$

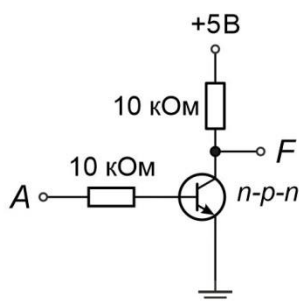
№	A	B	F
0	0	0	0
1	0	1	1
2	1	0	1
3	1	1	1



Логічний елемент АБО (OR)

Рис. 3.8. Еквівалентні перемикальні схеми логічних функцій двох аргументів, які виконують основні операції кон'юнкції і диз'юнкції

3°. Можна скласти еквівалентну перемикальну схему для будь-якої функції. На практиці перемикальні функції логічних елементів реалізуються зазвичай на біполярних або польових транзисторах (Практикум 4). На Рис. 3.9 показана найпростіша схема реалізації перемикальної функції на біполярному *n-p-n* транзисторі.



Функція заперечення (інверсія)

$$F=F_1(A)=\bar{A}$$

Рівні електричного сигналу	
A	F
ВИСОКИЙ (HIGH)	НИЗЬКИЙ (LOW)
НИЗЬКИЙ (LOW)	ВИСОКИЙ (HIGH)

Логічний елемент НЕ (NOT)

Рис. 3.9. Реалізація перемикальної функцій на основі транзисторного ключа

Технологія побудови цифрових схем на основі простих транзисторних ключів (Рис. 3.9) отримала назву резистивно-транзисторної логіки (РТЛ).

3.2. ВПРАВИ І ЗАВДАННЯ

3.2.1. Вправи для виконання

1°. Визначити не втрачену логічну функцію і суттєві аргументи функції, перетворивши заданий вираз функції з двома аргументами (варіанти).

2°. Скласти еквівалентну перемикальну схему логічної функції, вираз якої заданий (варіанти).

3°. Варіанти до вправ 3.2.1.1° і 3.2.1.2° зведені в одну таблицю.

Варіант	Вираз функції для вправи 1	Вираз функції для вправи 2
1.	$\overline{A + B} + \overline{A \leftarrow B}$	$\overline{A} \times B$
2.	$(A \leftrightarrow B) \times (A + B)$	$A \times B + C$
3.	$\overline{A \leftarrow B} + \overline{A \rightarrow B}$	$A \times (B + C) \times D$
4.	$\overline{A + B} + \overline{A \rightarrow B}$	$\overline{A} \times \overline{B}$
5.	$(A \leftarrow B) \times (A \rightarrow B)$	$A \times (B + C \times D)$
6.	$(A \rightarrow B) \times \overline{A \times B}$	$A + \overline{B}$
7.	$(A \leftarrow B) + (A \rightarrow B)$	$A + B \times C + D$
8.	$\overline{(A \leftrightarrow B)} \times \overline{A \leftarrow B}$	$(A + B) \times C + D$
9.	$A \times B + \overline{A \leftarrow B}$	$A \times B + C \times D$
10.	$\overline{(A \leftrightarrow B)} \times \overline{A \times B}$	$(A + B) \times (C + D)$
11.	$A \times B + \overline{A \rightarrow B}$	$(A \times B + C) \times D$
12.	$(A \leftarrow B) \times (A + B)$	$\overline{A} + \overline{B}$

3.2.2. Контрольне завдання

1°. Виконати вправи 3.2.1.1° і 3.2.1.2° для одного варіанту. Варіант вибирається за номером по списку групи. Для чергового номеру, що першим перевершує число варіантів, вибирається варіант 1, і так далі.

2°. По результатам виконання вправ оформити звіт.

ПРАКТИКУМ 4. БАЗОВІ ЛОГІЧНІ ЕЛЕМЕНТИ

4.1. ВСТУП. ЦИФРОВІ ЕЛЕМЕНТИ

4.1.1. Цифрові схеми. Дискретні сигнали

1°. Символічні представлення *цифрових елементів* на схемах цифрових пристроїв схожі на «чорні скриньки» з одним або кількома входами і виходами (Рис. 4.1. а), про які відомо лише єдине – їх функція (Рис. 4.1. б, е), яка пов’язує вихідні змінні з входними змінними елементів (Рис. 4.1. в, г, д).

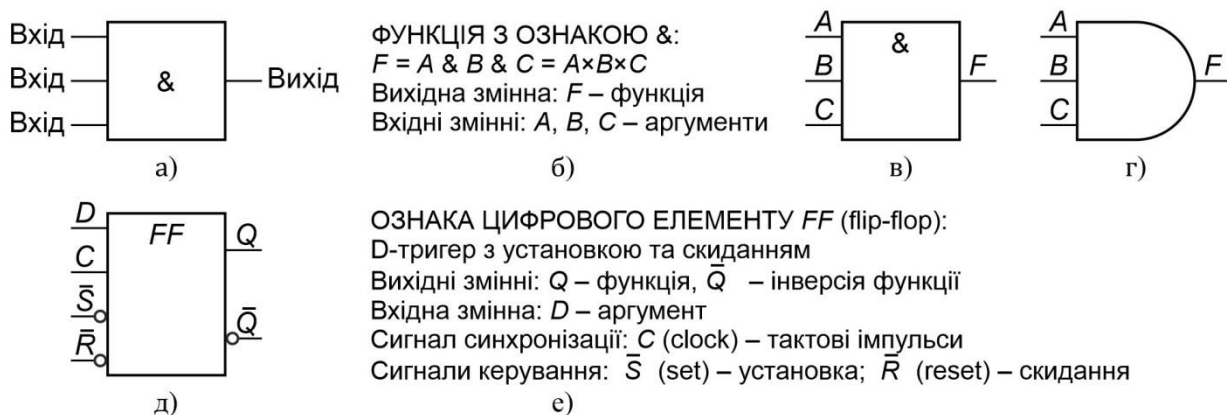


Рис. 4.1. Приклади символічного представлення цифрових елементів на схемах, де
 а) умовна «чорна скринька» з трьома входами і одним виходом та ознакою &,
 б) функція кон'юнкції I (AND), ознака якої &, в, г) схема логічного елементу I (AND):
 в) згідно стандарту Міжнародної електротехнічної комісії (ІЕК);
 г) згідно поширеного в спеціальній літературі стандарту milspec (США);
 д, е) D-тригер з установкою та скиданням (flip-flop)

В цифрових елементах еквівалентами бінарних цифрових значень 1 і 0 істинності і хибності змінних логічних функцій є дискретні рівні електричного сигналу. Значення фізичних величин цих рівнів сигналу залежать від технології виробництва інтегральних мікросхем (4.1.1.2°), в яких фізично втілені цифрові логічні елементи.

2°. Якщо в аналоговій електроніці вихідний сигнал змінюється *безперервно* при *поступовій* зміні входного сигналу (Рис. 4.2. а, б), то в цифровій електроніці сигнали змінюються *стрибкоподібно*, *дискретним* чином від одного сталого рівня сигналу до іншого (від рівня a до рівня b і навпаки – від рівня b до рівня a на Рис. 4.2. в). Ці рівні мають загальну назву – ВИСОКИЙ (HIGH) і НИЗЬКИЙ (LOW) рівні сигналу, незалежно від значень їх фізичних величин. На схемах та в таблицях істинності застосовується їх аббревіатура В (Н) і Н (L).

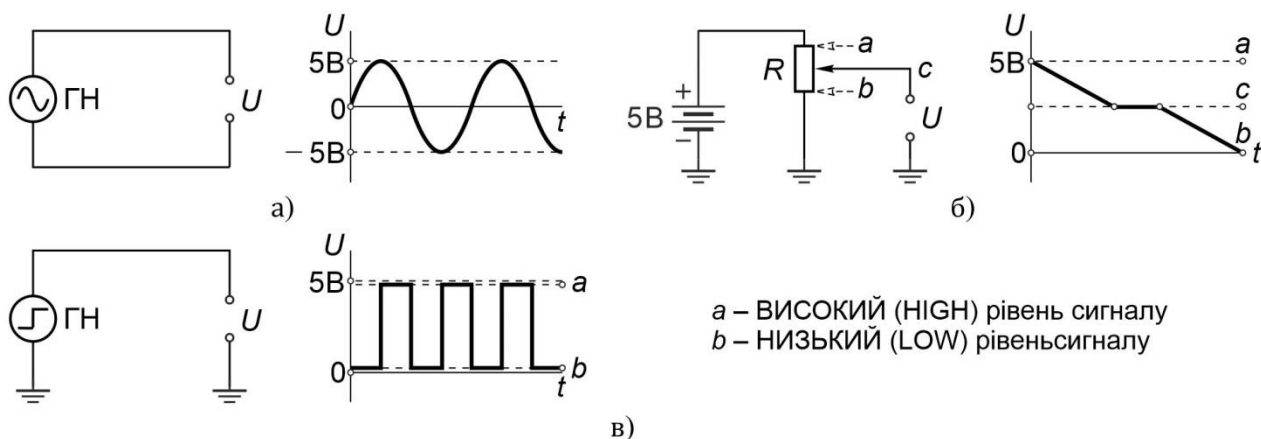


Рис. 4.2. Електричні сигнали, де а) аналоговий сигнал генератора гармонічних сигналів, а) аналоговий сигнал резистивного регулятора постійного струму, в) цифровий сигнал генератора тактових імпульсів (астабільного мультівібратора)

3°. Зазвичай ВИСОКИЙ (HIGH) рівень сигналу є фізичним еквівалентом цифрового значення 1, яким в алгебрі логіки визначається істинність логічної змінної, а НИЗЬКИЙ (LOW) рівень сигналу є еквівалентом цифрового значення 0 хибності логічної змінної. Таку логіку прийнято називати *позитивною логікою*. В цифровій схемотехніці також реалізуються схеми при яких ВИСОКИЙ (HIGH) рівень сигналу відповідає значенню 0, а НИЗЬКИЙ (LOW) рівень сигналу значенню 1, зазвичай це сигнали керування цифровими елементами. Такі сигнали прийнято позначати на схемах кружечком на вході цифрового елемента, при цьому буквений символ сигналу має символ інверсії (Рис. 4.1. е). Таку логіку прийнято називати *негативною логікою*.

4°. Цифровий сигнал можна отримати ручним способом механічним перемикачем (Рис. 4.3. а), проте, може бути невизначеність рівня сигналу через брязкіт контактів перемикача (Рис. 4.3. б).

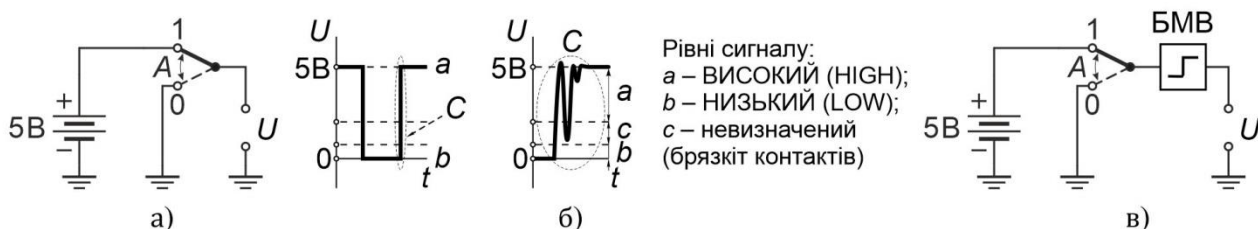


Рис. 4.3. Отримання цифрового сигналу ручним способом механічним перемикачем, де а) схема, б) невизначеність сигналу через брязкіт контактів перемикача, в) схема нейтралізації брязкоту контактів бістабільним мультівібратором (БМВ)

Для нейтралізації не визначення рівня сигналу застосовується додатковий елемент (Рис. 4.3. в) – *фіксатор* для усунення брязкоту контактів, *бістабільний мультівібратор* (Практикум 6) або *тригер* (Практикум 7, 7.1.1.3°, Рис. 7.2 д).

Також цифровий сигнал можна отримати ручним способом механічною кнопкою (Рис. 4.4. а).

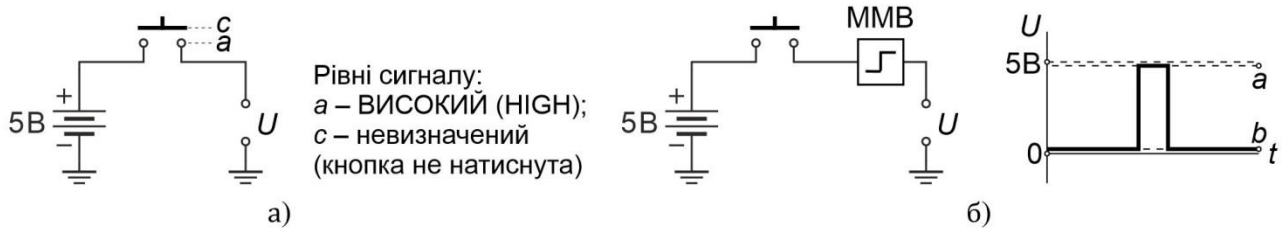


Рис. 4.4. Отримання цифрового сигналу ручним способом механічним кнопкою, де а) схема, б) схема з моностабільним мультівібратором (ММВ)

При застосуванні механічної кнопки також може бути невизначеність сигналу при не натиснутій кнопці (Рис. 4.4. б). Для нейтралізації не визначення рівня сигналу застосовується додатковий елемент (Рис. 4.4. в) – *одновібратор*, який ще називають *моностабільним мультівібратором* (Практикум 6).

Третій тип мультівібратора, що працює в режимі вільних коливань, який названо *астабільним мультівібратором* (Рис. 4.2. в), виконує функцію *генератора тактових імпульсів* (Практикум 6).

4.1.2. Технології виробництва цифрових інтегральних мікросхем

1°. Технології виробництва цифрових інтегральних мікросхем отримали свої назви від схемотехнічної реалізації логічних функцій. Так, технологія побудови логічних елементів на основі транзисторних ключів (3.1.3.3°, Рис. 3.9) отримала назву резисторно-транзисторної логіки (РТЛ). Технологія РТЛ, як і наступна реалізована технологія діодно-транзисторної логіки (ДТЛ) на основі діодів, біполярних транзисторів і резисторів, вважаються застарілими.

2°. Найбільш широке застосування отримала технологія транзистор-транзисторної логіки (ТТЛ) на основі біполярних транзисторів і резисторів, та технологія комплементарної структури метал-оксид-напівпровідник (КМОП) на основі польових транзисторів. Елементи ТТЛ зазвичай працюють при напрузі живлення 5 В, тоді як сучасні елементи КМОП можуть працювати при напрузі, що вимірюється в долях вольта, а деякі елементи КМОП можуть потребувати напругу до 15 В. В сучасних цифрових пристроях, особливо для широкого застосування, застосовуються елементи КМОП з напругою живлення 3,3 В.

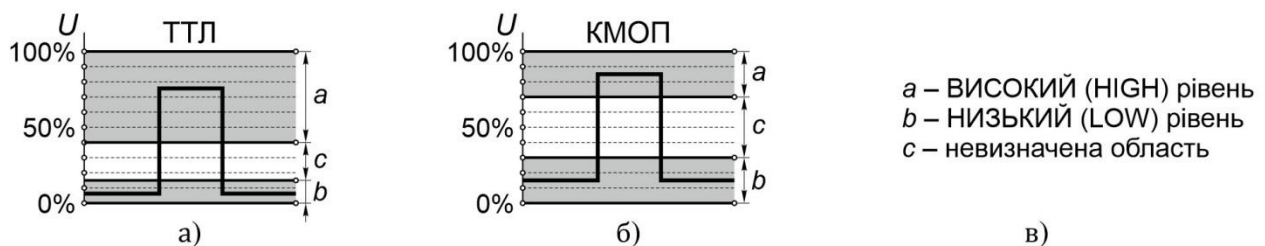


Рис. 4.5. Логічні рівні інтегральних мікросхем в процентах від напруги живлення U , де а) технологія ТТЛ, б) технологія КМОП

Особливістю технології КМОП у порівнянні з технологією ТТЛ є дуже мале енергоспоживання у статичному режимі. В більшості випадків можна вважати, що енергія елементами КМОП споживається лише під час перемикання логічних станів.

3°. Промисловим стандартом інтегральних мікросхем, які реалізують логічні функції за технологією ТТЛ, є серія 7400, що запроваджена фірмою Texas Instruments (середина 1960-х, Даллас, США), а за технологією КМОП – серія 4000, що запроваджена фірмою RCA Corporation (1968, Нью Йорк, США).

Серії інтегральних мікросхем 7400 і 4000 містять сотні найменувань цифрових пристроїв, починаючи з базових логічних елементів. Серії 7400 і 4000 містять підсерії за призначенням, за типом розвинутої технології, тощо, що відображається в найменуванні мікросхеми, наприклад, для серії 7400 – прописними буквами (префікс, після цифр 74, суфікс). Найменування SN74LS08N мікросхеми означає стандартну за призначенням мікросхему, що вироблена фірмою Texas Instruments (префікс SN) за технологією ТТЛ Шоттки зі зниженою споживаною потужністю (букви LS – Low-power Schottky) в DIP-корпусі (суфікс N) для комерційного призначення (цифри 74, для військового призначення – цифри 54), яка містить чотири логічних елементи І (AND) (цифри 08).

Конкретна мікросхема супроводжується *технічним паспортом* (Data Sheet), що містить вичерпну технічну інформацію безпосередньо від виробника і яким необхідно послуговуватися при застосуванні мікросхеми.

В сучасних цифрових пристроях застосовуються інтегральні мікросхеми для поверхневого монтажу, в тому числі серії 7400 за технологією КМОП. Мікросхеми серій 7400 і 4000 в DIP-корпусах (Рис. 4.6) залишаються зручними для макетування і налаштування схем в процесі практичних і лабораторних робіт.

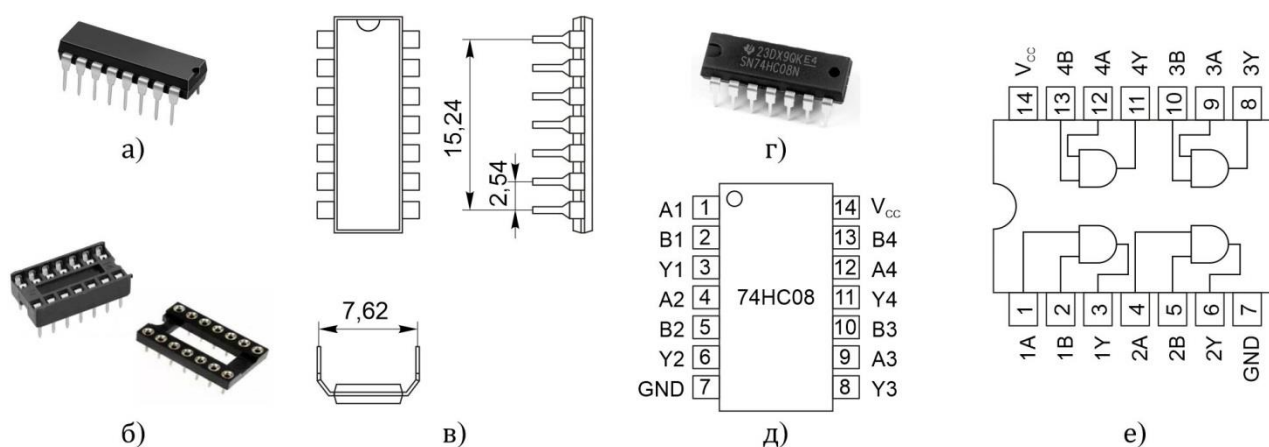


Рис. 4.6. Інтегральна мікросхема в корпусі DIP 14 (14 контактів), де а) DIP -корпус, б) макетні монтажні панелі, в) монтажні розміри, в) мікросхема SN74HC08N за технологією КМОП 4-х логічних елементів І (AND), г) призначення контактів мікросхеми SN74HC08N (pinout v.1), д) схема SN74HC08N (pinout v.2)

4.2. БАЗОВІ ЛОГІЧНІ ЕЛЕМЕНТИ

4.2.1. Визначення

1°. Із чотирнадцяти не втрачених функцій пустої множини, одного і двох аргументів (Практикум 3) сім функцій виконуються *логічними елементами*, які прийнято називати *базовими логічними елементами*. Всі базові логічні елементи реалізовані фізично інтегральними мікросхемами (4.1.2.3°, Рис. 4.6).

Три з семи базових логічних елементів, на основі яких можна побудувати будь-яку цифрову логічну схему – НЕ (NOT), І (AND), АБО (OR), прийнято називати *основними логічними елементами* (4.2.2.1°÷4.2.2.34°, Рис. 4.7).

Інші два базових логічних елементи, І-НЕ (NAND) та АБО-НЕ (NOR), прийнято називати *універсальними логічними елементами* (4.2.3.1°÷4.2.3.3°, Рис. 4.8) завдяки їх властивостям – будь-яку цифрову логічну схему можна побудувати на основі одного з цих двох логічних елементів.

Ще два базових логічних елементи прийнято називати *виключними логічними елементами* – виключне АБО (XOR) та виключне АБО-НЕ (XNOR), які доповнюють перші п'ять для зручного практичного вирішення різноманітних логічних задач (4.2.4.1°÷4.2.4.3°, Рис. 4.9).

2°. Інтегральні мікросхеми чотирьох базових логічних елементів, І (AND) та АБО (OR), І-НЕ (NAND) та АБО-НЕ (NOR), можуть мати більше двох входів (вхідних змінних, аргументів функції) для кожного елементу. Тоді запис такого логічного елементу здійснюється, наприклад, як 4І-НЕ (4-input NAND), що означає логічний елемент І-НЕ (NAND) з чотирма входами (вхідними змінними). Логічні елементи 4І-НЕ (4-input NAND) реалізовано, наприклад, інтегральною мікросхемою 74LS20 (Dual 4-input NAND Gate – два логічних елементи 4І-НЕ).

4.2.2. Основні логічні елементи

1°. Основний логічний елемент НЕ (NOT) виконує функцію інверсії (операцію логічного заперечення) одного аргументу A і має логічний вираз функції:

$$F = \bar{A}. \quad (4.4.)$$

2°. Основний логічний елемент І (AND) виконує функцію кон'юнкції (операцію логічного множення). Для двох аргументів A і B (два входи логічного елементу) логічний вираз функції І (AND):

$$F = A \times B. \quad (4.1.)$$

Для чотирьох аргументів A, B, C, D (чотири входи логічного елементу), наприклад, логічний вираз функції 4І-НЕ (4-input NAND):

$$F = A \times B \times C \times D. \quad (4.2.)$$

3°. Основний логічний елемент АБО (OR) виконує функцію диз'юнкції (операцію логічного додавання). Для двох аргументів A і B (два входи логічного елемента) логічний вираз функції АБО (OR):

$$F = A + B. \quad (4.3.)$$

4°. Символьні схеми за стандартами ІЕК і milspek основних логічних елементів НЕ (NOT), І (AND), АБО (OR), їх таблиці істинності, перемикальні еквівалентні схеми, схеми реалізації по технологіям РТЛ і ТТЛ та схеми реалізації в інтегральних мікросхемах зведені у таблиці (Рис. 4.7).

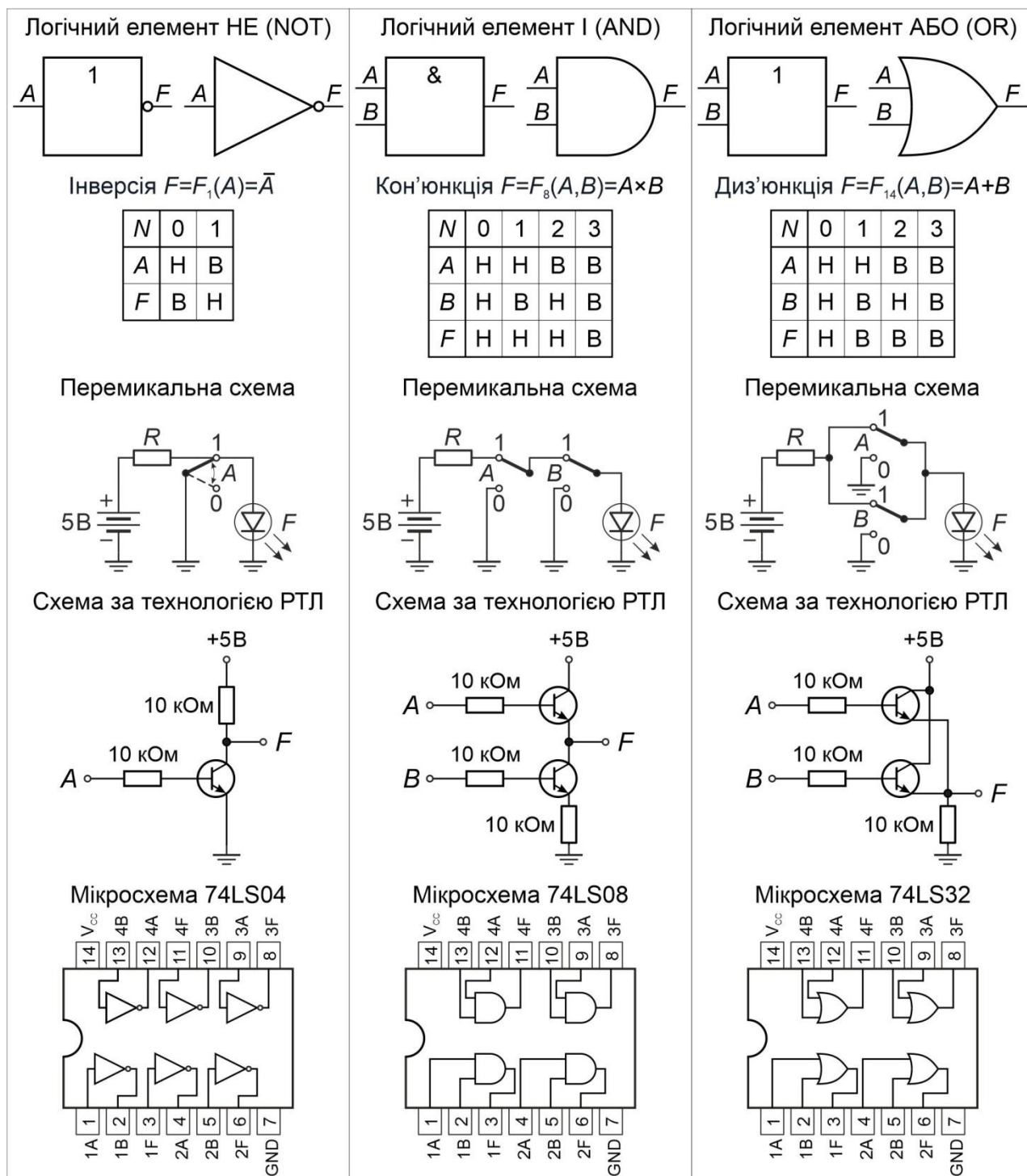


Рис. 4.7. Основні логічні елементи НЕ (NOT), І (AND) і АБО (OR)

4.2.3. Універсальні логічні елементи

1°. Схеми, таблиці істинності та реалізація універсальних логічних елементів І-НЕ (NAND) та АБО-НЕ (NOR) зведені у таблиці (Рис. 4.8).

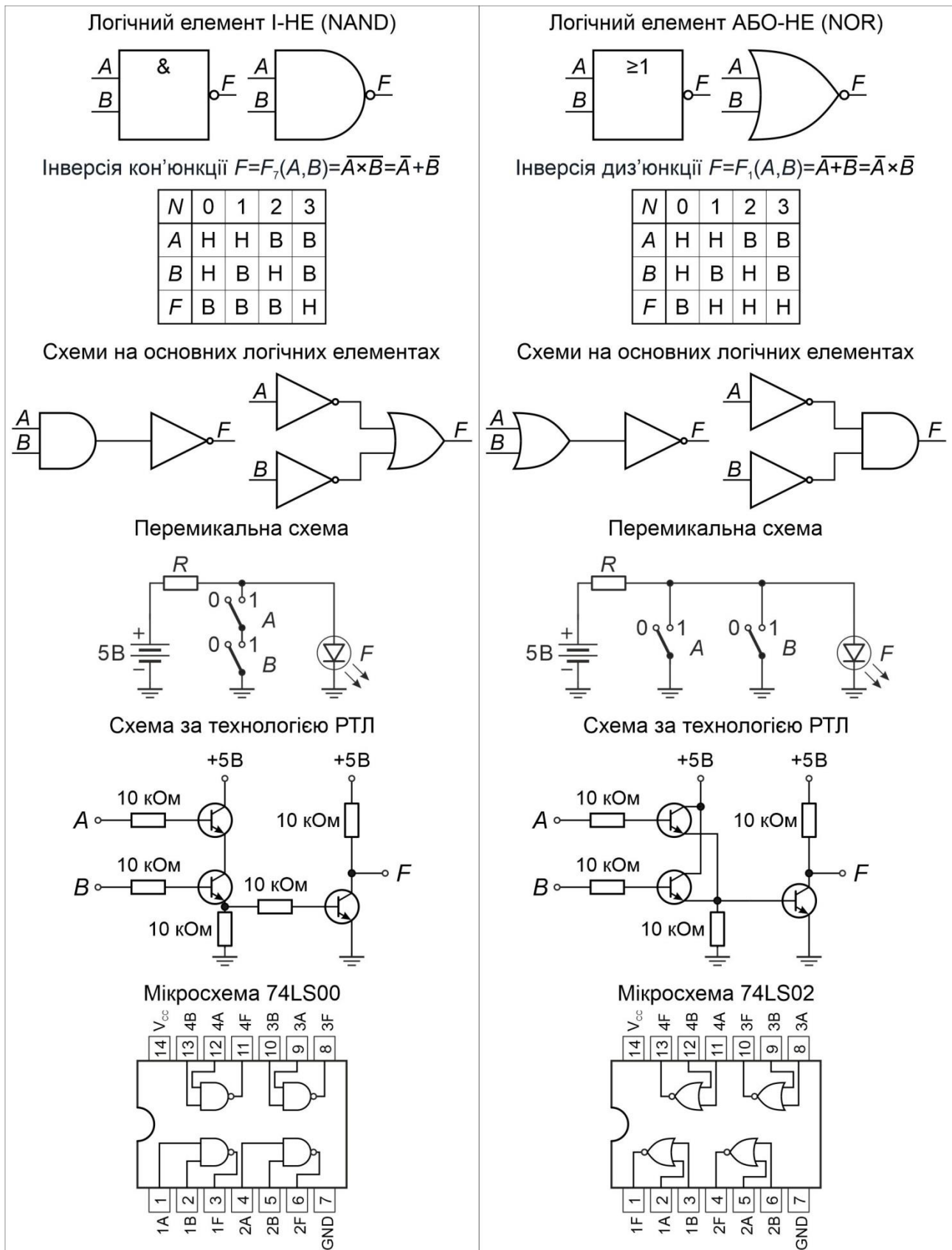


Рис. 4.8. Універсальні логічні елементи І-НЕ (NAND) та АБО-НЕ (NOR)

2°. Універсальний логічний елемент І-НЕ (NAND) виконує функцію інверсії кон'юнкції (операцію заперечення логічного множення). Для аргументів A і B (два входи логічного елементу) логічний вираз функції І-НЕ (NAND):

$$F = \overline{A \times B} = \overline{A} + \overline{B}. \quad (4.5.)$$

3°. Універсальний логічний елемент АБО-НЕ (NOR) виконує функцію інверсії диз'юнкції (операцію заперечення логічного додавання). Для аргументів A і B (два входи логічного елементу) логічний вираз функції АБО-НЕ (NOR):

$$F = \overline{A + B} = \overline{A} \times \overline{B}. \quad (4.6.)$$

4.2.4. Виключні логічні елементи

1°. Схеми, таблиці істинності та реалізація логічних елементів виключного АБО (XOR) та виключного АБО-НЕ (XNOR) зведені у таблиці (Рис. 4.9).

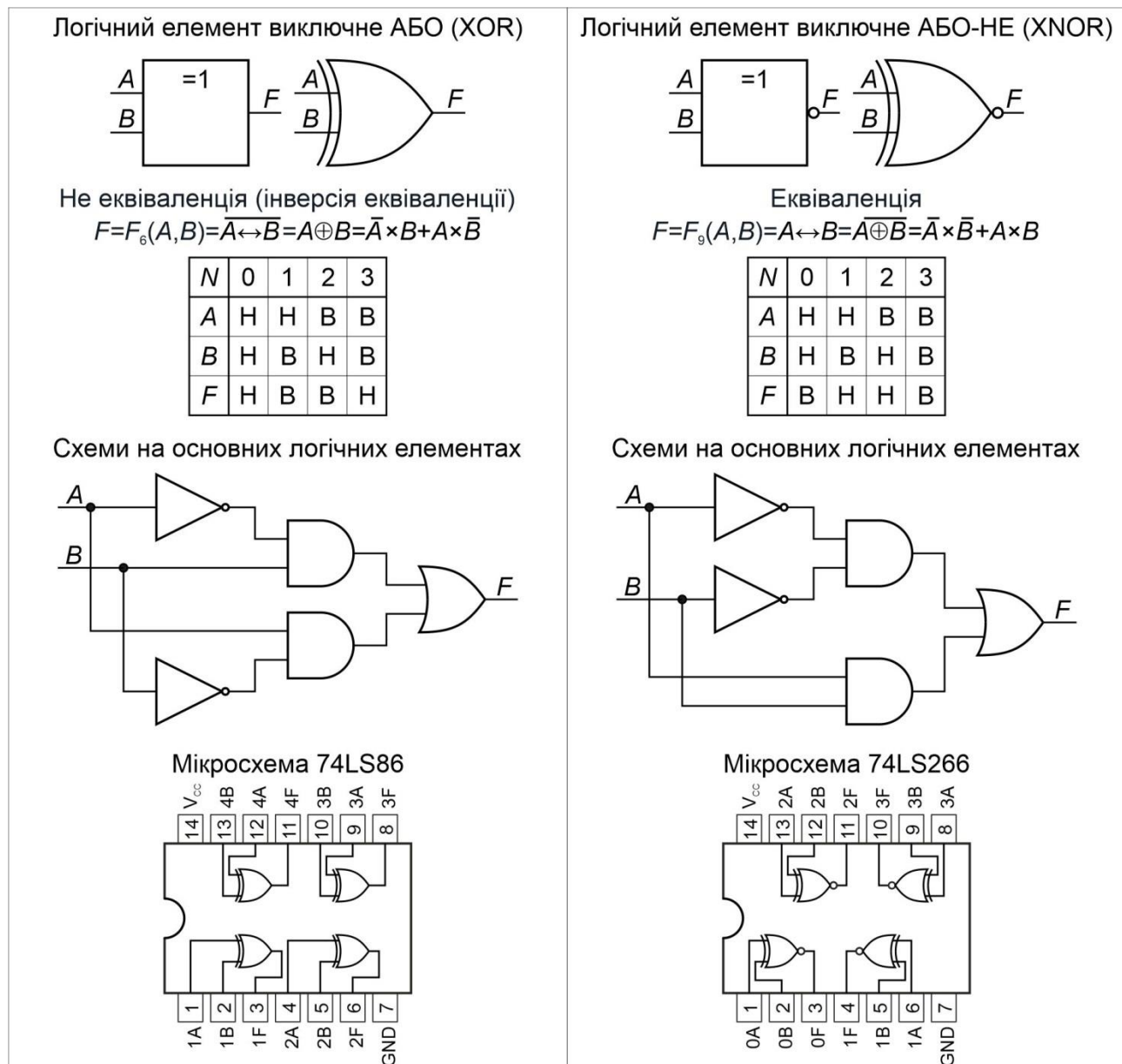


Рис. 4.9. Логічні елементи виключне АБО (XOR) та виключне АБО-НЕ (XNOR)

2°. Логічний елемент виключне АБО (XOR) виконує функцію інверсії еквівалентності (операцію заперечення еквівалентності). Для аргументів A і B (два входи логічного елементу) логічний вираз функції АБО (XOR):

$$F = A \oplus B = \overline{A \leftrightarrow B} = A \times \overline{B} + \overline{A} \times B. \quad (4.7.)$$

3°. Логічний елемент виключне АБО-НЕ (XNOR) виконує функцію еквівалентності (операцію еквівалентності). Для аргументів A і B (два входи логічного елементу) логічний вираз функції АБО-НЕ (XNOR):

$$F = \overline{A \oplus B} = A \leftrightarrow B = \overline{A} \times \overline{B} + A \times B. \quad (4.8.)$$

4.2.5. Еквівалентні логічні схеми

1°. З будь-якого виразу логічної функції, записаного в ідеальній нормальній кон'юнктивній або диз'юнктивній формі чи з елементарними кон'юнкціями або диз'юнкціями (Практикум 3), впливає схема на основних логічних елементах І (AND), АБО (OR), НЕ (NOT). Наприклад – схеми виключних логічних елементів АБО (XOR) та АБО-НЕ (XNOR), які впливають з виразів їх функцій в ідеальній нормальній кон'юнктивній формі, що показано на Рис. 9. Ці вирази можна перетворити і отримати інший еквівалентний вираз і побудувати іншу *еквівалентну логічну схему* на інших логічних елементах. Наприклад, для функції (4.7) логічного елементу виключного АБО (XOR) на основі правил і законів алгебри логіки (Практикум 2) можна зробити наступні перетворення:

$$\begin{aligned} F = A \oplus B &= \overline{A \leftrightarrow B} = A \times \overline{B} + \overline{A} \times B = A \times \overline{B} + A \times \overline{A} + \overline{B} \times B + \overline{A} \times B = \\ &= A \times (\overline{A} + \overline{B}) + B \times (\overline{A} + \overline{B}) = A \times (\overline{\overline{A} \times \overline{B}}) + B \times (\overline{\overline{A} \times \overline{B}}) = \\ &= \overline{\overline{A} \times \overline{B}} \times (A + B). \end{aligned} \quad (4.9.)$$

Еквівалентні схеми логічного елементу виключного АБО (XOR) до і після перетворення виразу його функції показані на Рис. 10.

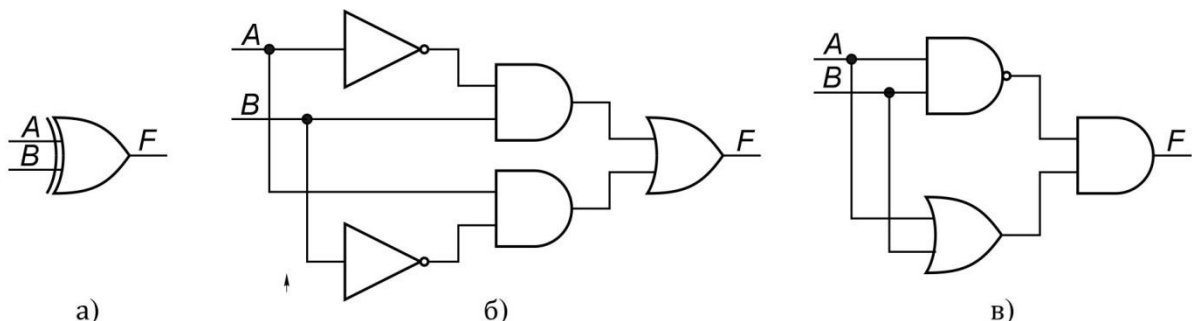


Рис. 4.10. Еквівалентні схеми логічного елементу виключного АБО (XOR), де а) символічна схема логічного елементу виключного АБО (XOR), б) еквівалентна схема на основних логічних елементах, в) спрощена еквівалентна схема

Якщо еквівалентна схема логічного елементу виключного АБО (XOR) на основних логічних елементах потребує по два елементи НЕ (NOT) та І (AND) і

один елемент АБО (OR) – всього п'ять, то еквівалентна схема, після перетворення логічного виразу функції, потребує лише по одному основному елементу І (AND) та АБО (OR) і один універсальний елемент АБО-НЕ (NOR), всього – три.

2°. Практичний інтерес мають еквівалентні схеми на універсальних логічних елементах І-НЕ (NAND) та АБО-НЕ (NOR), коли для будь-якої функції можна скласти схему лише на основі одного з цих двох елементів. Нижче (Рис. 4.11) показані еквівалентні схеми на універсальних логічних елементах І-НЕ (NAND) та АБО-НЕ (NOR) для розглянутого вище прикладу – для функції логічного елементу виключного АБО (XOR).

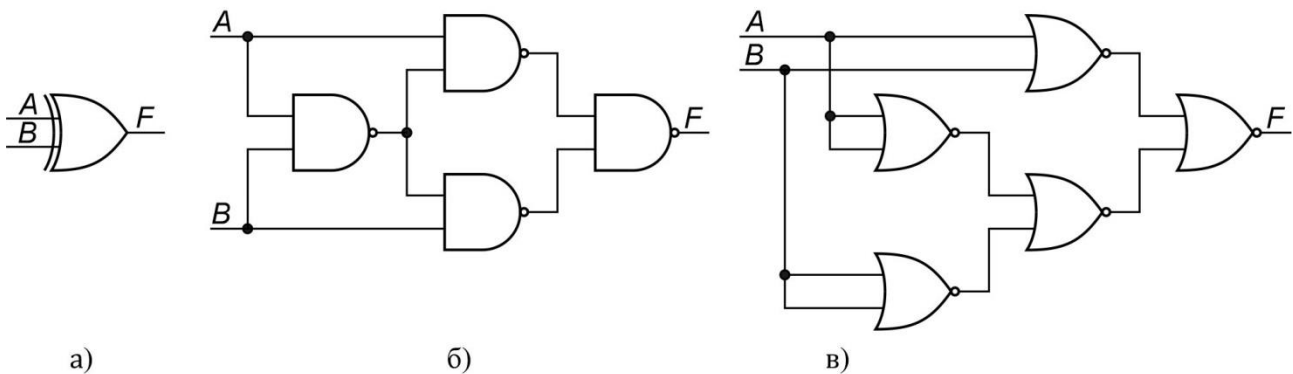


Рис. 4.11. Еквівалентні схеми логічного елементу виключного АБО (XOR) на основі одного універсального логічного елементу: а) І-НЕ (NAND); б) АБО-НЕ (NOR)

3°. В процесі аналізу логічних схем може виникнути зворотна задача – скласти вираз булевої функції заданої логічної схеми та, при необхідності і можливості, цей вираз спростити. Такий процес можна продемонструвати для еквівалентних логічних схем логічного елементу виключного АБО (XOR) на основі універсальних логічних елементів. Булева функція схеми на універсальному логічному елементі І-НЕ (NAND) (Рис. 4.11. б) з подальшим перетворенням:

$$\begin{aligned}
 F &= \overline{\overline{A \times (\overline{A \times B})} \times \overline{B \times (\overline{A \times B})}} = A \times (\overline{A \times B}) + B \times (\overline{A \times B}) = \\
 &= A \times (\overline{A} + \overline{B}) + B \times (\overline{A} + \overline{B}) = A \times \overline{A} + A \times \overline{B} + B \times \overline{A} + B \times \overline{B} = \\
 &= 0 + A \times \overline{B} + \overline{A} \times B + 0 = A \times \overline{B} + \overline{A} \times B = \overline{A \leftrightarrow B} = A \oplus B
 \end{aligned}
 \tag{4.10.}$$

Булева функція схеми на логічному елементі АБО-НЕ (NOR) (Рис. 4.11. в) з подальшим перетворенням:

$$\begin{aligned}
 F &= \overline{\overline{A + B} + \overline{(\overline{A + A}) + (\overline{B + B})}} = \overline{\overline{A + B} + \overline{\overline{A} + \overline{B}}} = \\
 &= (A + B) \times (\overline{\overline{A} + \overline{B}}) = A \times \overline{\overline{A}} + A \times \overline{\overline{B}} + B \times \overline{\overline{A}} + B \times \overline{\overline{B}} = \\
 &= 0 + A \times \overline{B} + \overline{A} \times B + 0 = A \times \overline{B} + \overline{A} \times B = \overline{A \leftrightarrow B} = A \oplus B
 \end{aligned}
 \tag{4.11.}$$

На початку складання виразу функції заданої складної логічної схеми зручно логічну операцію, яку виконує кожний логічний елемент, виділяти окремими дужками, навіть коли цього не потребують правила алгебри логіки.

4.2.6. Прості логічні схеми застосування базових логічних елементів

1°. Будь-яку вербальну функцію (Практикум 1), наприклад – ЯКЩО НА ВУЛИЦІ ТЕПЛО (висловлювання A) АБО ОПАЛЕННЯ В БУДИНКУ ВІМКНУТО (висловлювання B) та ВІКНА В КІМНАТІ ЗАЧИНЕНІ (висловлювання C), ТО В КІМНАТІ ТЕПЛО (висловлювання F), можна виразити булевою функцією (4.12) та скласти її логічну схему на базових логічних елементах (Рис. 4.12).

$$F = A + B \times C. \quad (4.12.)$$

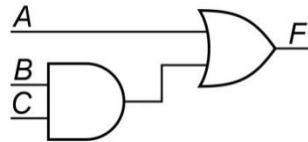


Рис. 4.12. Логічна схема на базових логічних елементах I (AND) та АБО (OR) логічного висловлення (4.2.3.1°)

2°. Інший приклад простої схеми застосування базових логічних елементів – керуванням дозволом (забороною) проходження сигналів, яке можна здійснити на основних логічних елементах I (AND) та АБО (OR) (Рис. 4.11) або універсальних елементах I-НЕ (NAND) та АБО-НЕ (NOR). На один із входів логічного елемента подається сигнал A , якому дозволяється або забороняється проходження, наприклад, послідовність тактових сигналів з генератора прямокутних імпульсів (ГН), а на інший – сигнал керування B , який встановлює час дозволу τ_d і час заборони τ_z проходження вхідного сигналу A на вихід F .

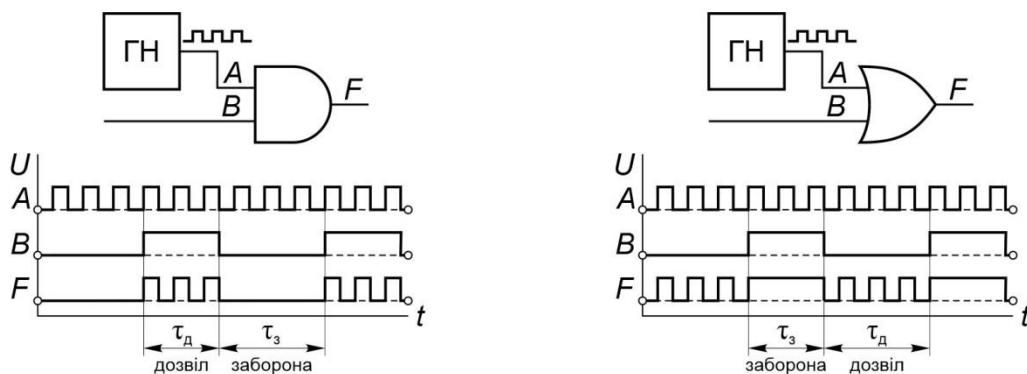


Рис. 4.13. Схеми керуванням дозволом (забороною) проходження сигналів

3°. Розглянуті вище логічні схеми на кількох логічних елементах (4.2.5.1°, Рис. 4.10; 4.2.5.2°, Рис. 4.11; 4.2.6.1°, Рис. 4.12) є комбінацією базових логічних елементів для виконання простих функцій. По такому ж принципу будуються комбінаційні схеми для виконання більш складних функцій, розповсюджених у цифрових пристроях. Реалізовані по логічним комбінаційним схемам цифрові пристрої отримали назву *комбінаційних пристроїв* (Практикум 5).

4.3. ВПРАВИ І ЗАВДАННЯ

4.3.1. Вправи для виконання

1°. Для заданої схеми (варіанти представлені в зведеній таблиці для вправ) скласти вираз логічної функції.

2°. Спростити вираз логічної функції, отриманий при виконанні вправи 4.3.1.1°, до виразу, що містить найменшу кількість елементарних логічних операцій інверсії, кон'юнкції та диз'юнкції, застосувавши постулати, правила та закони алгебри логіки (Практикум 2). Побудувати еквівалентну спрощену схему на основних логічних елементах відповідно до отриманого спрощеного виразу логічної функції.

3°. Реалізувати виконання логічної функції заданої схеми одним універсальним логічним елементом, застосувавши інтегральні мікросхеми серії 7400 (варіанти), для чого побудувати схему з відповідно пов'язаними входами і виходами інтегральної мікросхеми (мікросхем). При виконанні вправи варто дотримуватись наступних рекомендацій:

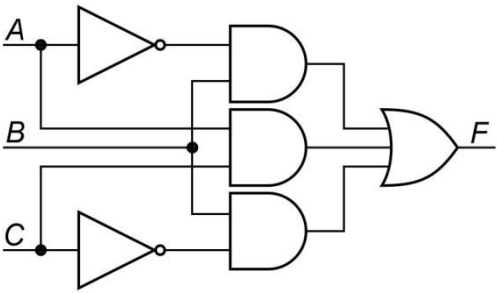
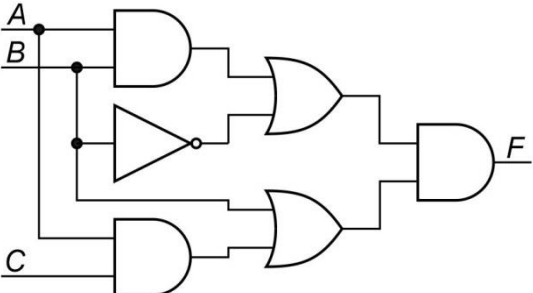
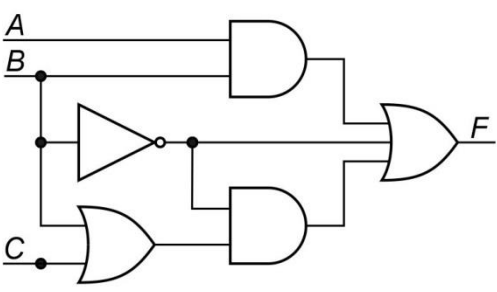
а) ознайомитись з технічним паспортом на інтегральну мікросхему на сайті виробника мікросхеми (вибрати самостійно), наприклад, вибравши інтегральну мікросхему логічних елементів фірми Texas Instruments (США) за посиланням <https://www.ti.com/logic-voltage-translation/logic-gates/overview.html>;

б) на схемі виходи не задіяних логічних елементів інтегральної мікросхеми (мікросхем) варто заземлити підключенням до земляного полюсу GND джерела живлення мікросхеми, що є обов'язковим (!) для мікросхем, виготовлених по технології КМОП;

в) якщо в інтегральній мікросхемі з трьома (чотирма) входами універсального логічного елемента є не задіяні входи, то для коректного виконання логічних операцій необхідно на не задіяних контактах логічного елемента АБО-НЕ (NOR) забезпечити НИЗЬКИЙ (LOW) рівень сигналу через їх підключення до земляного полюсу GND джерела живлення мікросхеми, а на не задіяних контактах логічного елемента І-НЕ (NAND) – ВИСОКИЙ (HIGH) рівень сигналу через їх підключення до позитивного полюсу джерела живлення мікросхеми;

г) при побудові еквівалентної схеми на одному універсальному логічному елементі І-НЕ (NAND) чи АБО-НЕ (NOR) можна скористатися еквівалентними логічними схемами з додатку до цього практикуму.

4°. Варіанти до вправ 4.3.1.1° і 4.3.1.3° зведені в одну таблицю. В таблиці рекомендовано застосувати серію мікросхем 74LS00, де позначення LS між цифрами вказує на технологію виробництва інтегральної мікросхеми – Low-power Schottky (технологія ТТЛШ зі зниженою споживаною потужністю). При виконанні вправи 4.3.1.3° може бути застосована мікросхема, виготовлена по іншій технології, при умові, що ця технологія буде вказана (розшифрована).

Схема	Інтегральна мікросхема	Варіант
	74LS00	1
	74LS02	2
	74LS10	3
	74LS20	4
	74LS27	5
	74LS03	6
	74LS02	7
	74LS03	8
	74LS10	9
	74LS20	10
	74LS27	11
	74LS00	12
	74LS00	13
	74LS02	14
	74LS10	15
	74LS20	16
	74LS27	17
	74LS03	18

4.3.2. Контрольне завдання

1°. Виконати вправи 4.3.1.1°, 4.3.1.2° і 4.3.1.3° для одного варіанту. Варіант вибирається за номером по списку групи. Для чергового номеру, що першим перевіряє число варіантів, вибирається варіант 1, і так далі.

2°. По результатам виконання вправ оформити звіт. На початку звіту вказується номер практикуму і його тема, номер групи і прізвище виконавця. Виконані вправи повинні містити умови завдання та його розв'язання за відповідною до даного практикуму нумерацією.

4.4. ДОДАТОК

4.4.1. Еквівалентні схеми основного логічного елементу НЕ (NOT)

1°. Еквівалентні схеми логічного елементу НЕ (NOT) на універсальних логічних елементах І-НЕ (NAND) і АБО-НЕ (NOR) представлені на Рис. 4.14.



Рис. 4.14. Еквівалентні логічні схеми логічної функції інверсії $\bar{A}$

2°. Вирази функції інверсії $\bar{A}$ відповідно до еквівалентних схем (Рис. 4.14):

$$\bar{A} = \overline{A \times A}; \quad \bar{A} = \overline{A + A}. \quad (4.13.)$$

4.4.1. Еквівалентні схеми основного логічного елементу І (AND)

1°. Еквівалентні схеми логічного елементу І (AND) на універсальних логічних елементах І-НЕ (NAND) і АБО-НЕ (NOR) представлені на Рис. 4.15.

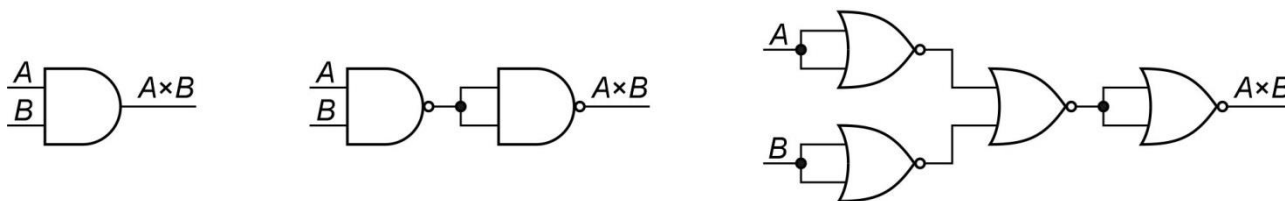


Рис. 4.15. Еквівалентні логічні схеми логічної функції кон'юнкції $A \times B$

2°. Вирази функції кон'юнкції $A \times B$ відповідно до схем (Рис. 4.15):

$$A \times B = \overline{\overline{A + A + B + B + A + A + B + B + A + A + B + B + A + A + B + B}}; \quad (4.14.)$$

4.4.1. Еквівалентні схеми основного логічного елементу АБО (OR)

1°. Еквівалентні схеми логічного елементу АБО (OR) на універсальних логічних елементах І-НЕ (NAND) і АБО-НЕ (NOR) представлені на Рис. 4.16.

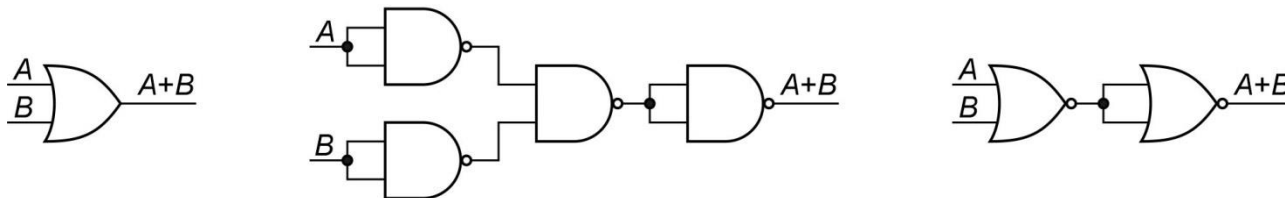


Рис. 4.16. Еквівалентні логічні схеми логічної функції диз'юнкції $A + B$

2°. Вирази функції диз'юнкції $A + B$ відповідно до схем (Рис. 4.16):

$$A + B = \overline{\overline{A \times A \times B \times B \times A \times A \times B \times B \times A \times A \times B \times B \times A \times A \times B \times B}}; \quad (4.15.)$$

$$A + B = \overline{\overline{A + B + A + B}}.$$

ПРАКТИКУМ 5. КОМБІНАЦІЙНІ ЦИФРОВІ ПРИСТРОЇ

5.1. ВСТУП

5.1.1. Комбінаційна і послідовнісна логіка

1°. Логічні схеми, побудовані шляхом з'єднання (*комбінації*) логічних елементів для виконання певної логічної задачі, в яких поточний стан на виході схеми визначається поточним станом на входах схеми і ніяк не залежить від попередніх станів на вході, отримали назву схем *комбінаційної логіки*. Комбінаційна логіка полягає в тому, що при зміні станів на вході логічної схеми отримується негайний відгук на виході, попередній стан будь-якого входу не запам'ятовується. Цифрові пристрої, реалізовані по схемі комбінаційної логіки, називають *комбінаційними цифровими пристроями*.

2°. Комбінаційні цифрові пристрої не володіють властивістю зберігання інформації. Таку властивість мають логічні схеми, побудовані на логічних елементах по *послідовнісній логіці*, яка полягає в тому, що поточний стан на виході схеми визначається *послідовністю* зміни станів на вході в певному порядку. Схеми послідовнісної логіки дозволяють зберегти отриманий на вході біт (біти) інформації, і в подальшому отримати його (їх) на виході. Цифрові пристрої, реалізовані по схемі послідовнісної логіки, називають *послідовнісними цифровими пристроями* (Практикум 7).

3°. До комбінаційних цифрових пристроїв, що розглядаються далі у цьому практикумі, відносяться *мультиплексори* (розділ 5.2 і підрозділ 5.3.1) і *демультиплексори* (підрозділ 5.3.2), *шифратори* (5.3.3.1°) і *дешифратори* (5.3.3.2°), *компаратори* (5.3.4.1°) і *суматори* (5.3.4.2°).

4°. Логічні схеми комбінаційної і послідовнісної логіки, що розповсюдженні в цифровій схемотехніці, фізично реалізовані інтегральними мікросхемами, які застосовуються на практиці. Предметом розгляду у цьому практикумі є інтегральні мікросхеми комбінаційної логіки та їх застосування. Інтегральні мікросхеми послідовнісної логіки розглядаються в Практикумі 7.

5°. При опрацюванні цього і подальших практикумів (Практикуми 5, 6, 7), предметом розгляду яких є цифрові пристрої, реалізовані інтегральними мікросхемами, варто також послуговуватися технічними паспортами безпосередньо від виробника мікросхем (Data Sheet – технічний паспорт), які містять вичерпну інформацію про цифровий пристрій, що розглядається.

5.1.2. Інтегральні мікросхеми цифрової логіки

1°. Вище відзначалося те, що найбільше поширення отримали інтегральні мікросхеми, виготовлені по технологіям ТТЛ і КМОП (4.1.2.2°). Якщо на початку застосування мікросхеми ТТЛ мали перевагу по швидкодії, а мікросхеми КМОП мали перевагу по низькій споживаній потужності, то розвиток технології КМОП, направлений в першу чергу на підвищення швидкодії, створив тенденцію на поступове витіснення мікросхем ТТЛ мікросхемами КМОП.

2°. Започаткована фірмою Texas Instruments (США) серія мікросхем 7400 тривалий час була стандартом лише мікросхем ТТЛ. Сучасна продукція серії 7400 містить підсерії поліпшеної ТТЛ, КМОП, біполярних КМОП (BiCMOS) та поліпшених КМОП (Advanced CMOS). Наприклад, комбінаційний цифровий пристрій, два мультиплексори 4 до 1, який реалізовано інтегральною мікросхемою SN74ALS153 за поліпшеною технологією ТТЛ Шоттки зі зниженою споживаною потужністю (ALS – Advanced Low Power Shottky Logic), має аналогічну реалізацію мікросхемами SN74HC153 за технологією КМОП високої швидкодії (HC – High-Speed CMOS Logic) та SN74AC153 за поліпшеною технологією КМОП (AC – Advanced CMOS Logic). Ці мікросхеми, виготовлені за різними технологіями, мають однакові логічні схеми і функціональне призначення, номер серії 153 і призначення контактів. Проте, є винятки (5.1.2.3°), тому варто завжди послуговуватися технічним паспортом на мікросхему безпосередньо від виробника.

3°. Однакові за функціональним призначенням і номером серії інтегральні мікросхеми цифрової логіки можуть мати схему, відмінну від стандартної схеми, коли на входах логічної схеми застосовуються тригери Шмітта (Schmitt-Triger Input). Наприклад, вказані вище мультиплексори SN74ALS153, SN74HC153, SN74AC153 (5.1.1.2°) мають однакову логічну схему. А аналогічні за функціональним призначенням і номером в серії (153) мультиплексори SN74HCS153 за технологією КМОП високої швидкодії мають схему з тригером Шмітта на входах (HCS – High-Speed CMOS Logic with Schmitt-Triger Input).

Наявність логічних схем з тригерами Шмітта на вході характерна для інших пристроїв цифрової логіки. При цьому інтегральним мікросхемам можуть бути присвоєні різні серійні номери, на відміну від розглянутого вище прикладу. Наприклад, мікросхема SN74HCS00, в якій реалізовано чотири логічних елементи І-НЕ (NAND) з двома входами, має функціональний аналог – мікросхему SN74HCS132 з тригерами Шмітта на входах.

Тригер Шмітта має аналоговий вхід і застосовується, наприклад, для нейтралізації брязкоту контактів. Далі розглядається ще один тип мікросхем цифрової комбінаційної логіки, що має аналогові входи і виходи (підрозділ 5.2.3).

4°. Однакові за функціональним призначенням мікросхеми цифрової логіки за своїми виходами також мають схеми, відмінні від стандартної логічної схеми. Стандартною вважається логічна схема, на виходах якої фізично може бути два стани цифрового сигналу – ВИСОКИЙ (HIGH) і НИЗЬКИЙ (LOW) рівні сигналу, стани 2S (2-State Output). Обидва стани є активними, тобто, при обох рівнях сигналу через вихідний ланцюг може протікати струм. Наприклад, згадані вище мікросхеми SN74HCS00 і SN74HCS132 чотирьох логічних елементів I-HE (NAND) з двома входами (5.1.1.3°) мають два вихідні стани (2S).

Вихід логічної схеми може мати третій стан, стан 3S (3-State Output), коли на виході утримується *високоімпедансний стан*, який є пасивним – струм через вихідний ланцюг не протікає, мікросхема відносно її виходу сприймається як діелектрик. Високоімпедансний третій стан отримується залежно від рівня додаткового до стандартної схеми керуючого сигналу на вході, який може позначатися на схемі, наприклад, як $\bar{S}$ (S – selection – вибір мікросхеми для дозволу передачі даних). Знак інверсії керуючого сигналу означає, що активні рівні вихідних сигналів будуть забезпечені при НИЗЬКОМУ (LOW) рівні керуючого сигналу $\bar{S}$. При ВИСОКОМУ (HIGH) рівні керуючого сигналу $\bar{S}$ на виході мікросхеми встановиться третій, високоімпедансний стан (3S).

До входу цифрових пристроїв підключеним може бути один і лише один вихід цифрового пристрою з двома станами (2S), тоді як кількість підключених виходів з трьома станами (3S) обмежується лише організованою логікою. Ланцюг 3S-виходів утворює так звану шину даних або адресну шину, на якій в поточний момент активним є вихід, мікросхема якого в цей час вибрана керуючим сигналом. Такий режим роботи цифрових пристроїв називають *мультиплексним режимом роботи*. Наприклад, всі розглянуті вище мікросхеми мультиплексорів 4 до 1, мікросхема SN74ALS153 та її аналоги (5.1.1.2°, 5.1.1.3°), мають три вихідних стани (3S). Мультиплексори 4 до 1 в серії 7400 реалізовані також мікросхемою SN74ALS352 з двома вихідними станами (2S), яка функціонально за стандартною логічною схемою є аналогом мікросхеми SN74ALS153 (3S).

5°. Інтегральні мікросхеми цифрової логіки можуть мати так звані *виходи з відкритим колектором* (OK – Open-Collector Output). В таких мікросхемах з ОК-виходами, як і в мікросхем з 2S-виходами, можливі два стани на виході, проте, активним є лише стан НИЗЬКОГО (LOW) рівня сигналу, при якому можливе протікання струму, що відповідає логічному цифровому значенню 0 на виході мікросхеми. Другий стан, що відповідає логічному цифровому значенню 1 на виході мікросхеми, є пасивним станом, при якому вихід мікросхеми відключається від під'єднаних до нього входів. Високий стан на виході таких мікросхем

можна забезпечити підключенням навантаження між виходами мікросхеми і джерела живлення, наприклад, резистора R з величиною опору порядку сотні Ом. Виходи мікросхем з ОК-виходами можуть об'єднуватися з ОК-виходами інших мікросхем на вході іншого цифрового пристрою, що надає їм властивість, споріднену з властивостями мікросхем з 3S-виходами. При цьому немає потреби в додатковому сигналі керування на вході мікросхем з ОК-виходами.

6°. Можливі стани на виході інтегральних мікросхем цифрової логіки з 2S-, 3S- і ОК-виходами можна продемонструвати еквівалентними перемикальними схемами (Рис. 5.1. а, б, в).

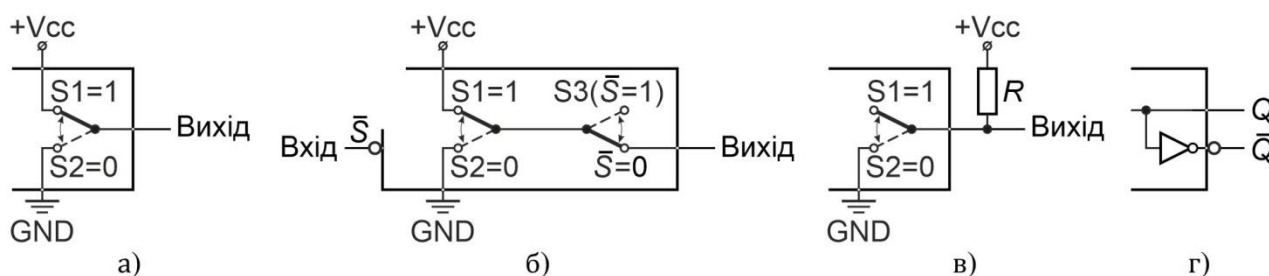


Рис. 5.1. Еквівалентні перемикальні схеми станів виходу інтегральних мікросхем, де а) стандартна логічна схема 2S (S1 – стан логічної 1, S2 – стан логічного 0), б) схема 3S (S3 – високоімпедансний стан), в) схема ОК (R – зовнішнє навантаження для отримання стану логічної 1), та г) вихід з доповненням (істинний і інверсний)

7°. Інтегральні мікросхеми цифрової логіки можуть мати схему з двома виходами, однаково функціонально пов'язаними з входами, з тією різницею, що другий вихід є логічним доповненням першого, тобто, другий вихід є інвертованим відносно першого (Рис. 5.1. г). Така логічна схема притаманна послідовним цифровим пристроям (Практикум 7), і іншим.

5.2. СІМЕЙСТВО ІНТЕГРАЛЬНИХ МІКРОСХЕМ ПРИСТРОЮ ЦИФРОВОЇ ЛОГІКИ НА ПРИКЛАДІ СЕЛЕКТОРА ДАНИХ (МУЛЬТИПЛЕКСОРА 2 ДО 1)

5.2.1. Мультиплексори. Стандартна логічна схема селектора даних

1°. Мультиплексорами або селекторами даних прийнято називати комбінаційні цифрові пристрої, які виконують функцію керування цифровим способом перемикачів. Вказані комбінаційні цифрові пристрої завжди мають один вихід, n адресних (селекторних) входів та 2^n інформаційних входів (входи з даними). Інтегральні мікросхеми можуть містити кілька таких пристроїв.

2°. В деякій спеціальній літературі назви *селектор* і *мультиплексор* розрізняються. Під *селектором* розуміється однополюсний двопозиційний цифровий перемикач (5.2.1.3°), а під мультиплексором – багатопозиційні цифрові перемикачі (5.3.1.2°). До цього класу комбінаційних цифрових пристроїв відноситься

також пристрій, у якого кількість адресних входів $n = 0$ та, відповідно, $2^n = 1$ інформаційних входів при одному інформаційному виході. Згідно загальної термінології (5.2.1.1°) пристрій міг би називатись селектором або мультиплексором 1 до 1 (1-Line to 1-Line). Назва «селектор» немає сенсу, оскільки вибору (селекції) даних на вході у пристрою немає, а «мультиплексор» припускає «багато» вибору. Такий пристрій названо *двоспрямованим перемикачем* (Bilateral Switch) і реалізовано мікросхемою CD4066B (5.2.3.2°), в якій сигнал може передаватися з входу на вихід і в зворотному напрямку. Тому пристрій міг би ще назватися мультиплексором/демультиплексором 1 до 1, що не має сенсу.

3°. При кількості адресних входів $n = 1$ та, відповідно, $2^n = 2$ інформаційних входів отримується цифровий двопозиційний перемикач – селектор даних або мультиплексор 2 до 1 (2-Line to 1-Line Selector/Multiplexer), стандартна логічна схема якого показана на Рис. 5.2. а). Мікросхема SN74LS157 (Рис. 5.2 б) містить чотири таких селектори (Quadruple 2-Line to 1-Line Selector/Multiplexer).

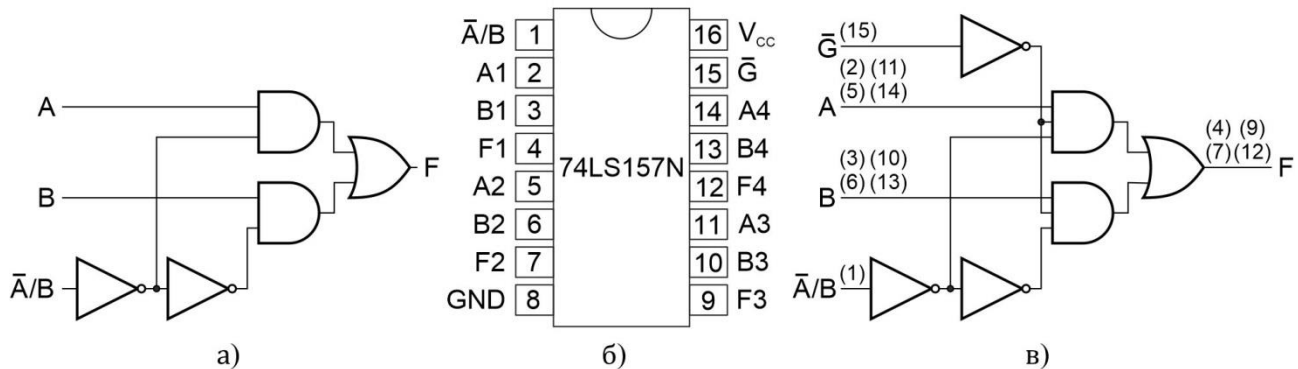


Рис. 5.2. Інтегральна мікросхема SN74LS157 селектора або мультиплексора 2 до 1, де а) стандартна логічна схема селектора, б) призначення контактів мікросхеми, в) логічна схема реалізації одного з чотирьох селекторів мікросхеми

4°. Далі, на прикладі селектора (мультиплексора 2 до 1), розглядається утворене сімейство інтегральних мікросхем цього одного цифрового пристрою. Наявність таких сімейств мікросхем, доступних на ринку, характерна для інших пристроїв цифрової логіки, що завжди варто враховувати при проектуванні.

5.2.2. Відмінності логічних схем інтегральних мікросхем селектора даних

1°. Власне, серія 7400 не містить мікросхем селектора даних (мультиплексора 2 до 1 по стандартній логічній схемі з 2S-виходами (Рис. 5.2.а).

Логічна схема (Рис. 5.2. в) базової мікросхеми SN74LS157 (Рис. 5.2. б) цього сімейства окрім стандартного селекторного (адресного) сигналу $\bar{A}/B$ вибору інформаційного входу А або В містить додатково ще один селекторний (стробіруючий) сигнал $\bar{G}$. Таблиця істинності і варіант символічних схем мікросхем SN74LS157 і SN74LS158 цього сімейства показані на Рис. 5.3. Логічна схема мік-

росхеми SN74LS158 відрізняється від логічної схеми мікросхеми SN74LS157 лише інвертованими виходами.



Рис. 5.3. Інтегральні мікросхеми селектора даних або мультиплексора 2 до 1, де
а) символічна схема мікросхем SN74LS157 і SN74LS257, б) таблиця істинності,
в) символічна схема мікросхем SN74LS158 і SN74LS258

2°. Призначення контактів, логічні схеми, таблиці істинності наступних мікросхем, SN74LS257 і SN74LS258, ті самі, що і у мікросхем SN74LS157 і SN74LS158, за винятком і з урахуванням того, що в логічній схемі селекторний сигнал $\bar{G}$ на вході цих мікросхем виконує функцію контролю виходів мікросхеми (output control) при підключенні її виходів до шини даних. Тобто, мікросхеми SN74LS257 і SN74LS258 мають логічні схеми з 3S-виходами, при цьому у мікросхемі SN74LS258, як і в мікросхемі SN74LS158, сигнали на виході інвертовані. Таблиця істинності і варіант символічних схем мікросхем SN74LS257 і SN74LS258 цього сімейства показані на Рис. 5.3.

3°. Наступні мікросхеми, SN74ALS857 і SN74LS399, окрім стандартної функції селектора даних (мультиплексора 2 до 1), виконують додаткові функції. Логічна схема мікросхеми SN74ALS857 має 3S-виходи, SN74LS399 – 2S-виходи.

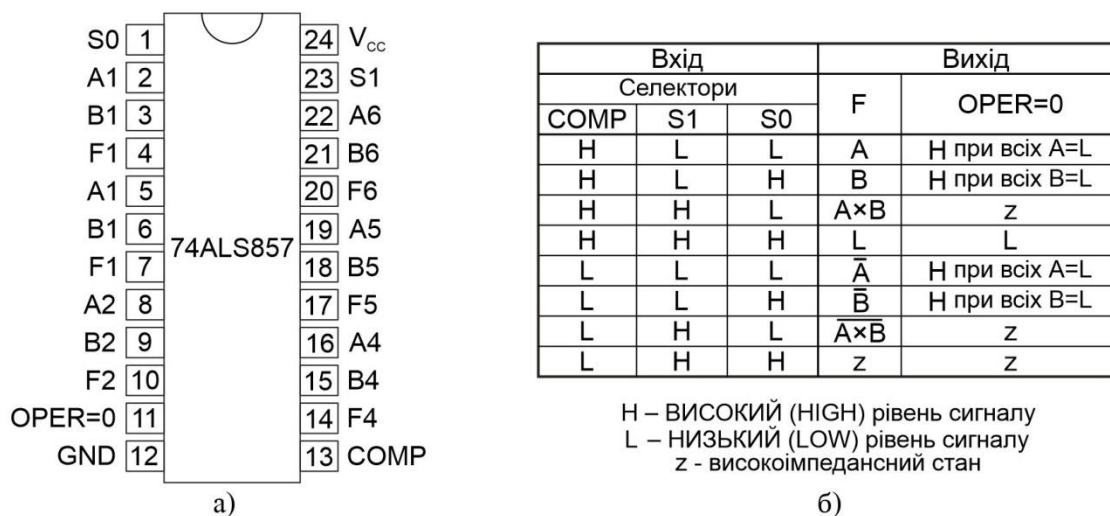
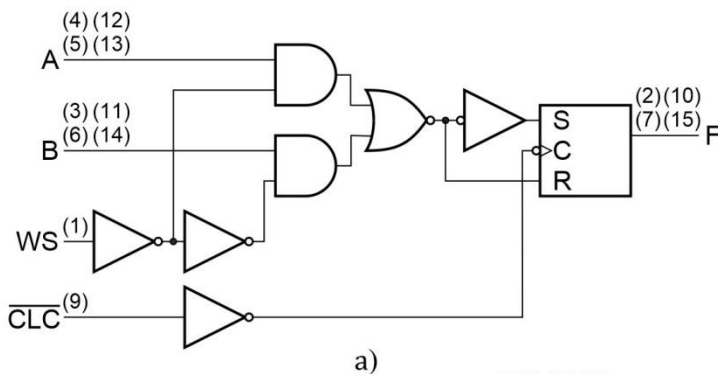


Рис. 5.4. Інтегральна мікросхема SN74ALS857 мультиплексора 2 до 1, де
а) призначення контактів мікросхеми, б) таблиця істинності мікросхеми

Мікросхема SN74ALS857 містить шість мультиплексорів 2 до 1, три входних селекторних сигнали та один вихідний операційний сигнал (OPER=0), який застосовується при каскадному з'єднанні мультиплексорів для збільшення кількості операндів. Завдяки наявності трьох входних селекторних сигналів (COMP, S1, S0), залежно від їх логічного значення (режиму роботи мультиплексора) окрім стандартної функції мультиплексора, селекції істинних значень одного з двох операндів, A і B, мікросхема виконує додаткові функції (має додаткові режими роботи). А саме – селекція інверсних значень операндів, а також виконання логічних операцій I (AND) і I-HE (NAND) над операндами. Також мікросхема виконує функцію очищення даних на виході при ВИСОКИХ (HIGH) рівнях всіх селекторних сигналів. Призначення контактів мікросхеми SN74ALS258 і її таблиця істинності показані на Рис. 5.4.

4°. Логічна схема мікросхеми SN74LS399 (Рис. 5.5) об'єднує схеми комбінаційної і послідовної логіки. Окрім стандартної функції селектора даних (мультиплексора 2 до 1), додатково виконується функція збереження даних, яка притаманна послідовним цифровим пристроям (тригерам, див. Практикум 7). Логічна схема мікросхеми SN74LS399 складається з послідовно з'єднаних стандартних схем мультиплексора 2 до 1 з істинним і інвертованим виходами і синхронного RS-тригера (Практикум 7) з істинним виходом.



Вхід		Вихід
CLC	WS	F
↑	L	A
↑	H	B
L	x	F ₀

г)

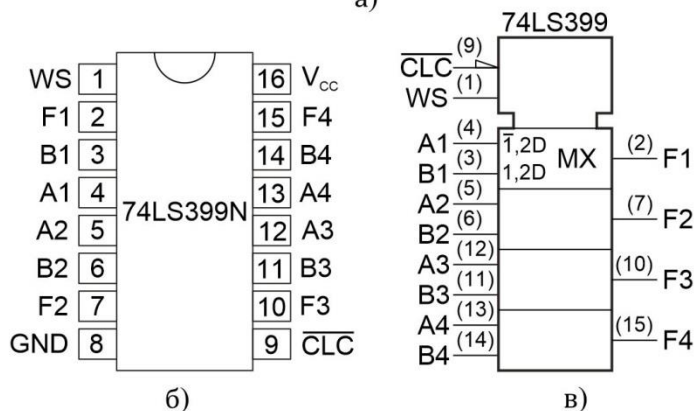
Поточні дані: A і B - на вході; F - на виході
(F=A або F=B).

Збережені дані: F₀ - на виході
(F₀=A₀ або F₀=B₀).

Вхідний селективний сигнал вибору даних:
WS - word select.

Вхідний сигнал синхронізації зберігання даних:
CLC - clock.

Стан сигналів на входах і виході:
H – ВИСОКИЙ (HIGH) рівень сигналу;
L – НИЗЬКИЙ (LOW) рівень сигналу;
x – наявний рівень сигналу не має значення;
↑ - наростання імпульсу сигналу синхронізації
(встановлення поточних даних на виході
F=A або F=B).



б)

в)

д)

Рис. 5.5. Інтегральна мікросхема SN74LS399 чотирьох мультиплексорів 2 до 1 зі зберіганням даних, де а) логічна схема мікросхеми, б) призначення контактів мікросхеми, в) символна схема мікросхеми, г) таблиця істинності мікросхеми, д) прийняті скорочення і позначення

5°. На ринку зустрічаються радянські аналоги інтегральних мікросхем серії 7400 за технологією ТТЛ і серії 4000 за технологією КМОП. Наприклад, базова серія 155 радянських мікросхем, аналог серії 7400, відзначається значною споживаною потужністю, яка була знижена запровадженням серії 555, аналогом серії 74LS00 за технологією ТТЛ Шоттки зі зниженою споживаною потужністю. Радянськими аналогами розглянутих вище інтегральних мікросхем селекторів даних (мультиплексорів 2 до 1) SN74LS157 і SN74LS158, SN74LS257 і SN74LS258 є радянські мікросхеми 555КП16 і 555КП18, 555КП11 і 555КП14 відповідно.

Радянська технологія виробництва мікросхем копіювала західну і не здобула особливого власного розвитку і поліпшень. При застосуванні радянських мікросхем цифрової логіки для визначення їх аналогів необхідно скористатися відповідними довідниками в пошуковому режимі.

5.2.3. Аналогові мультиплексори/демультиплексори з цифровим управлінням

1°. З серії інтегральних мікросхем 4000 за технологією КМОП, започаткованої фірмою RCA Corporation (США), аналогом мультиплексора 2 до 1 є мікросхема CD4053B (Рис. 5.6), в якій реалізовані три однополюсні двоканальні аналогові мультиплексори/демультиплексори з цифровим управлінням (2-Channel Analog Multiplexer/Demultiplexer with Logic-Level Conversion). Такі мікросхеми дво-спрямованих аналогових комутаторів з цифровим управлінням відіграють важливу роль в схемотехніці, пов'язуючи аналогові і цифрові пристрої.

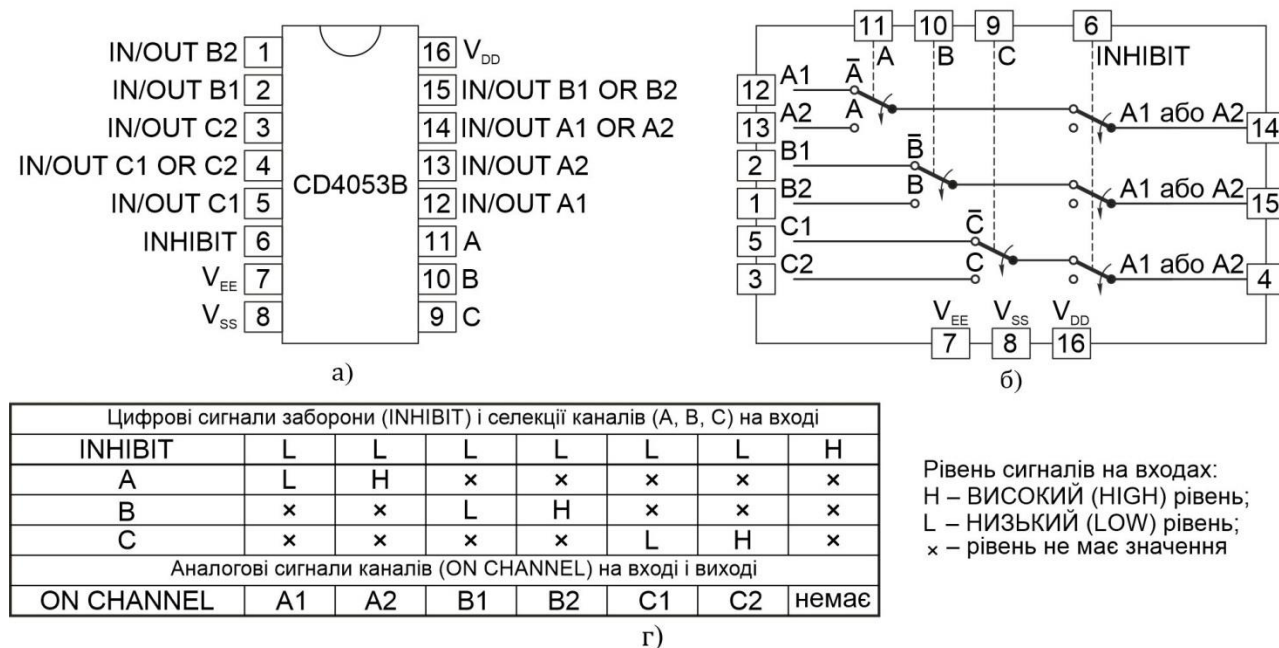


Рис. 5.6. Інтегральна мікросхема CD4053B – двопозиційний аналоговий мультиплексор/демультиплексор з цифровим управлінням, де а) призначення контактів, б) еквівалентна функціональна схема, в) таблиця істинності

Мікросхема CD4053B як і інші мікросхеми мультиплексорів/демультиплексорів серії – CD4052B (1 полюс, 4 канали) і CD4051B (1 полюс, 8 каналів), є аналоговими комутаторами з цифровим керуванням, які мають низький опір увімкнення та дуже низький струм споживання у вимкненому стані. Ці схеми мультиплексора/демультиплексора споживають надзвичайно низьку потужність, незалежно від логічного стану керуючих сигналів.

2°. Застосуванням з цієї ж серії 4000 вище згаданої мікросхеми CD4066B найпростішого аналогового двоспрямованого перемикача з цифровим управлінням (5.2.1.2°), можна розглянути взаємодію цифрових і аналогових пристроїв на прикладі реалізації простого чотирьохрозрядного цифроаналогового перетворювача (ЦАП) на так званих двійково вагомих резисторах (Рис. 5.7).

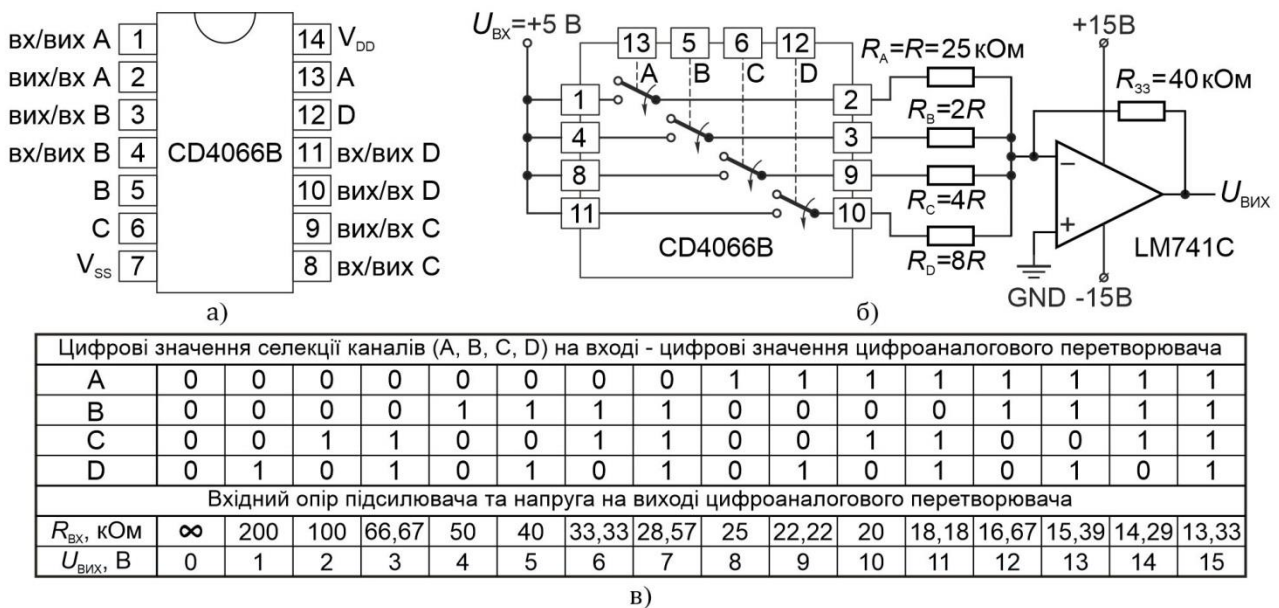


Рис. 5.7. Інтегральна мікросхема CD4066B чотирьох аналогових перемикачів з цифровим управлінням та її застосування, де а) призначення контактів мікросхеми, б) схема чотирьохрозрядного ЦАП з мікросхемою CD4066B, в) таблиця істинності

ЦАП на мікросхемі CD4066B і операційному підсилювачі, наприклад, LM741C, та двійково вагомих резисторах реалізується по схемі інверсного операційного підсилювача з негативним зворотнім зв'язком (Рис. 5.7. б). По такій схемі, залежно від напруги $U_{вх}$ і опору $R_{вх}$ на вході підсилювача та опору R_{33} в ланцюгу зворотного зв'язку, напруга $U_{вих}$ на виході ЦАП визначається як

$$U_{вих} = -U_{вх} \cdot (R_{33}/R_{вх}), \quad (5.1.)$$

Вхідний опір $U_{вх}$ залежить від двійкових значень 0 і 1 цифрових селекторних сигналів А, В, С і D на входах та значень опору двійково вагомих резисторів R_A, R_B, R_C і R_D на виходах мікросхеми CD4066B як

$$1/R_{вх} = A/R_A + B/R_B + C/R_C + D/R_D = A/R_A + B/2R + C/4R + D/8R. \quad (5.2.)$$

При $U_{вх} = 5\text{В}$ отримується чотирьохрозрядний ЦАП, що має діапазон від 0 до 15 В з роздільною здатністю 1 LSB/В (Рис. 5.7. в).

5.2.4. Сімейство мікросхем селектора даних (мультиплексора 2 до 1)

1°. Із сімейства цифрових перемикачів, селекторів даних або мультиплексорів, які мають завжди один інформаційний вихід і 2^n інформаційних входів при числі n адресних входів, вище розглянута реалізація цифрових пристроїв для випадків, коли $n = 0$ ($2^0 = 1$) та $n = 1$ ($2^1 = 2$). В першому випадку це найпростіший аналоговий двоспрямований перемикач з цифровим управлінням CD4066В (5.2.1.2°, 5.2.3.2°, Рис. 5.7). В другому – це селектори даних або мультиплексори 2 до 1. Шість розглянутих інтегральних мікросхем таких селекторів, SN74LS157 і SN74LS158 (5.2.2.1°, Рис. 5.3), SN74LS257 і SN74LS258 (5.2.2.2°, Рис. 5.3), SN74ALS857 (5.2.2.3°, Рис. 5.4), SN74LS399 (5.2.2.4°, Рис. 5.5), відрізняються за схемами реалізації, мають різні серійні номери за серією 7400, і є суто цифровими. Ще одна розглянута мікросхема за схемою і функціонально суттєво відрізняється від названих – це однополюсні двоканальні аналогові мультиплексори/демультиплексори з цифровим управлінням (мультиплексори 2 до 1 / демультиплексори 1 до 2) CD4053В.

2°. Окрім функціонального призначення та реалізованої логічної схеми, при проектуванні цифрових пристроїв можуть бути важливими електричні параметри інтегральної мікросхеми. Ідеальний на вході прямокутний імпульс в процесі проходження через елементи логічної схеми мікросхеми змінює свої параметри та характеристики (Практикум 6), що впливає на швидкодія пристрою, реалізованого мікросхемою. Не менш важливий параметр – споживана мікросхемою потужність. Вказані електричні параметри, швидкодія та споживана потужність, визначаються технологією виготовлення мікросхеми (Рис. 5.8).

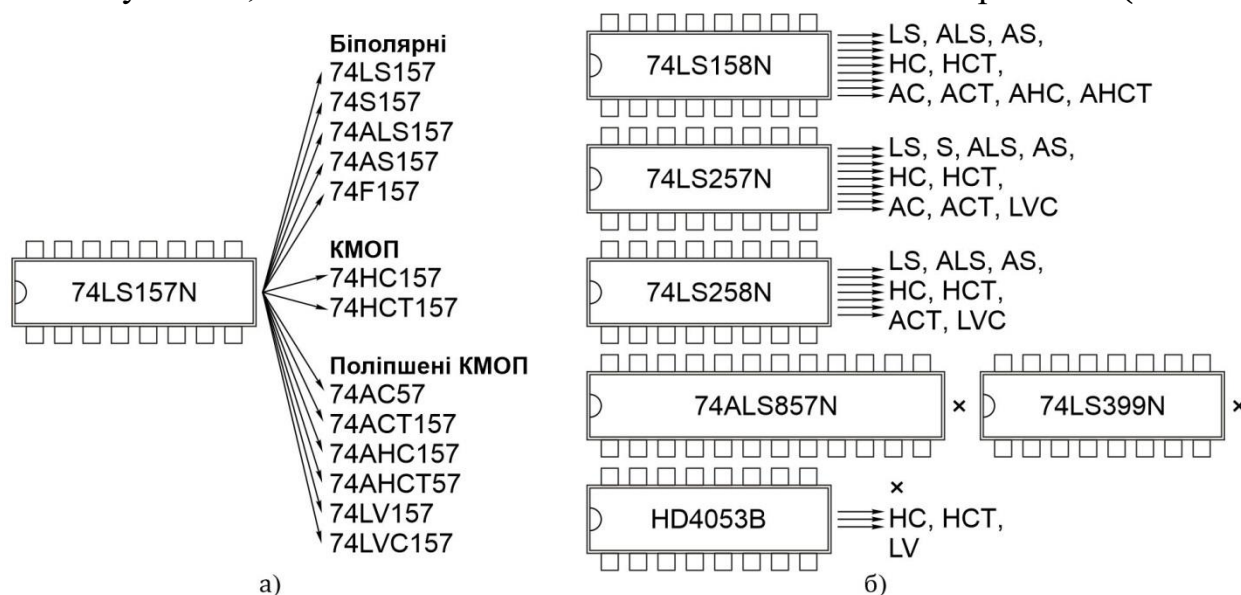


Рис. 5.8. Технології виготовлення інтегральних мікросхем селектора даних або мультиплексора 2 до 1 фірми Texas Instruments (Даллас, США), з каталогу 2007 року, де технології а) базової мікросхеми SN74LS157, б) інших мікросхем сімейства

Опис тієї чи іншої технології можна знайти в спеціальній літературі. Електричні і інші технічні характеристики пристрою цифрової логіки за тією чи іншою технологією виготовлення вказані в технічному паспорті на пристрій безпосередньо від виробника. Електричні характеристики пристрою за однією і тією ж технологією можуть дещо відрізнятися, залежно від виробника.

3°. Номенклатура інтегральних мікросхем одного сімейства розширюється також за рахунок їх різного конструктивного виконання (Рис. 5.9).



Рис. 5.9. Конструктивні виконання мультимплексора 2 до 1 на прикладі базової інтегральної мікросхеми сімейства – мікросхеми 74LS157 фірми Texas Instruments

Номенклатура інтегральних мікросхем одного сімейства збільшується також за рахунок їх різного конструктивного виконання (Рис. 5.9).

5.3. КОМБІНАЦІЙНІ ЦИФРОВІ ПРИСТРОЇ

5.3.1. Мультимплексори (продовження)

1°. В попередньому підрозділі практикуму розглянуто сімейство інтегральних мікросхем на прикладі селектора даних, мультимплексора 2 до 1. Дані про інтегральні мікросхеми мультимплексора 2 до 1 подані в таблиці (Таблиця 5.1).

Таблиця 5.1. Інтегральні мікросхеми мультимплексорів 2 до 1

Мультимплексори	Тип	Примітки
74LS157	(2 до 1)×4	Вихід 2S (5.2.1.3°, Рис.5.2) (5.2.2.1°, 5.2.2.2°, Рис.5.3)
74LS158	(2 до 1)×4	Інверсний вихід 2S (5.2.2.1°, 5.2.2.2°, Рис.5.3)
74LS257	(2 до 1)×4	Вихід 3S (5.2.2.1°, 5.2.2.2°, Рис.5.3)
74LS258	(2 до 1)×4	Інверсний вихід 3S (5.2.2.1°, 5.2.2.2°, Рис.5.3)
74ALS857	(2 до 1)×6	Вихід 3S, додаткові функції на виході (5.2.2.3°, Рис.5.4)
74LS399	(2 до 1)×4	Вихід 2S, з пам'яттю (5.2.2.4°, Рис.5.5)
HD4053B	(2 до 1)×3	Аналогові сигнали, цифрове керування (5.2.3.1°, Рис.5.6)

2°. Промислово реалізовані також сімейства інтегральних мікросхем мультимплексорів 4 до 1, 8 до 1, 16 до 1. Довідкові дані про мікросхеми кожного з цих сімейств представлені далі в таблиці. Логічна схема певної мікросхеми, її відмінність від стандартної логічної схеми мультимплексора, таблиця істинності, конструктивне виконання, електричні параметри мікросхеми тощо вичерпано представлені в технічному паспорті (datasheet) на цю мікросхему (Рис. 5.10).

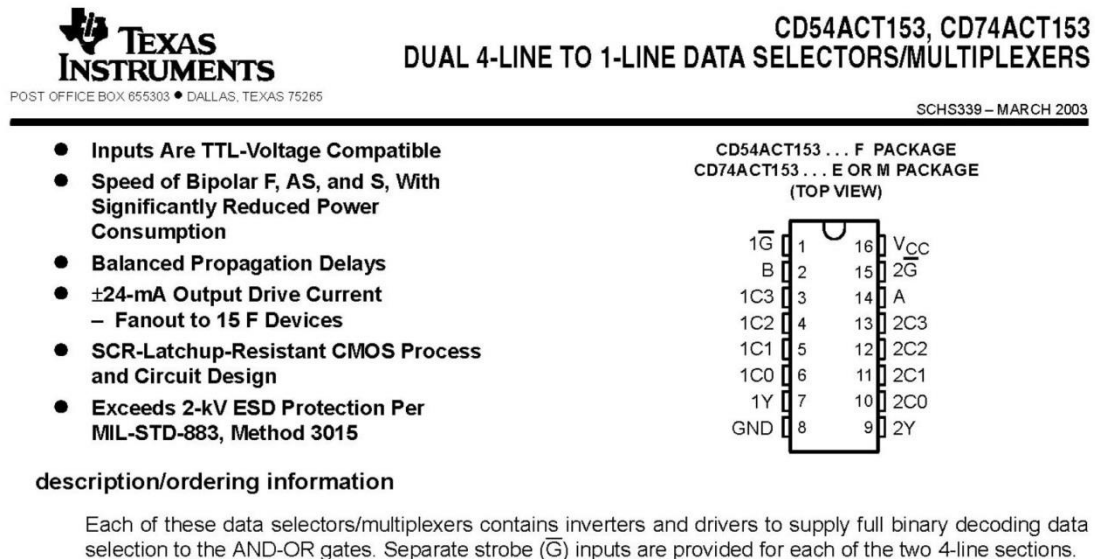


Рис. 5.10. Фрагмент титульної сторінки технічного паспорта (datasheet) на інтегральну мікросхему CD74ACT153 мультиплексора 4 до 1 від виробника мікросхеми (фрагмент вилучено з цифрової копії документу лише для наглядної ілюстрації)

Перед практичним застосуванням будь-яких інтегральних мікросхем, зокрема мультиплексорів (Таблиця 5.2), необхідно ознайомитись з технічним паспортом на мікросхему.

Таблиця 5.2. Інтегральні мікросхеми мультиплексорів 4 до 1, 8 до 1, 16 до 1

Мікросхема	MS	Функція згідно з технічним паспортом (datasheet)
74LS153	4 до 1	Dual 2-state 4-line to 1-line data selectors/multiplexers
74LS253	4 до 1	Dual 3-state 4-line to 1-line data selectors/multiplexers
74HC4052	4 до 1	Dual 4-channel analog multiplexer/demultiplexer
74HC4352	4 до 1	Dual 4-channel analog multiplexer/demultiplexer with latch
74LS151	8 до 1	2-state 1-of-8 line data selector/multiplexer
74LS251	8 до 1	3-state 1-of-8 line data selector/multiplexer
74HCT354	8 до 1	3-state 8-input multiplexer/register with transparent latches
74HCT356	8 до 1	3-state 8-input multiplexer/register, non-transparent data latch
74HCT4051	8 до 1	8-channel analog multiplexer/demultiplexer
74HCT4351	8 до 1	8-channel analog multiplexer/demultiplexer with latch
DM74150	16 до 1	2-state 1-of-16 line data selector/multiplexer
74AS250	16 до 1	3-state 1-of-16 line data selector/multiplexer
74HC4067	16 до 1	16-channel analog multiplexer/demultiplexer

5.3.2. Демультимплексори

3°. Демультимплексорами або розподільниками даних прийнято називати комбінаційні цифрові пристрої, які виконують функцію керованих цифровим способом перемикачів. Демультимплексори завжди мають один вхід (вхід з даними), n адресних (селекторних) входів та 2^n інформаційних виходів (виходи з даними). Коли на адресні входах встановлюється кодова послідовність, що відповідає номеру одного із виходів, інформаційний вхід перемикається до цього виходу. Такі комбінаційні цифрові пристрої називають ще *декодерами*.

Промислово реалізовані сімейства інтегральних мікросхем демультимплексорів при кількості адресних входів $n = 2, 3, 4$ та, відповідно, при кількості інформаційних виходів $2^n = 4, 8, 16$. На відміну від селекторів (мультиплексорів), в назві типу яких вказують відношення кількості інформаційних входів до кількості інформаційних виходів (2 до 1, 4 до 1, 8 до 1, 16 до 1), в назві типу декодерів (демультимплексорів) вказують відношення кількості адресних входів до кількості інформаційних виходів (2 до 4, 3 до 8, 4 до 10, 4 до 16).

4°. Для одного адресного входу $n = 1$, на відміну від мультиплексора, реалізація інтегральною мікросхемою демультимплексора відсутня. При $n = 2$ адресних (селекторних) входів та $2^n = 4$ інформаційних виходів (виходи з даними) отримується демультимплексор 2 до 4 (2-to-4 Line), стандартна логічна схема якого показана на Рис. 5.11. а). Мікросхема SN74LS139 (Рис. 5.11 б) містить два таких декодери (Dual 2-to-4 line decoder/demultiplexer).

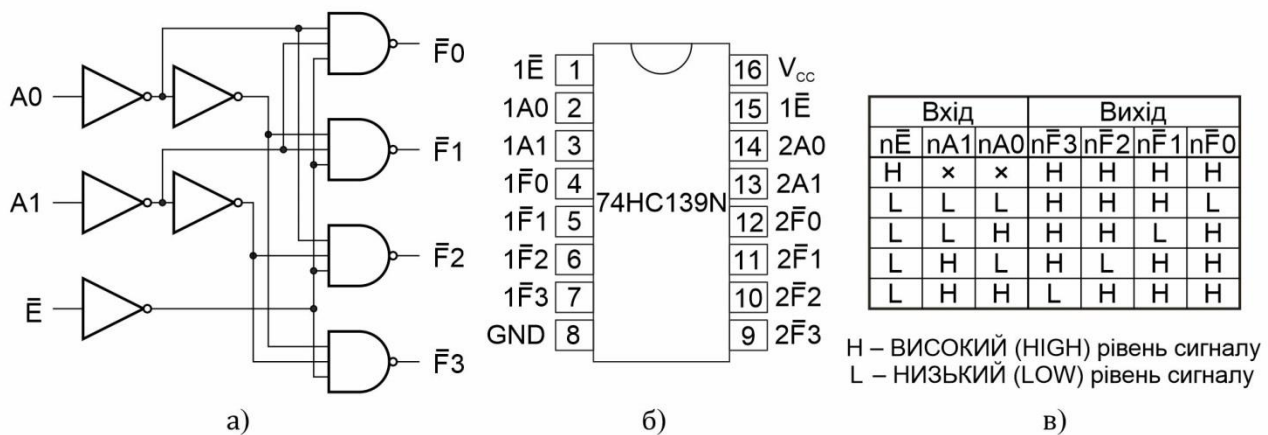


Рис. 5.11. Інтегральна мікросхема SN74LS139 декодера або демультимплексора 2 до 4, де а) стандартна логічна схема декодера, б) призначення контактів мікросхеми, в) таблиця істинності

5°. Промислово реалізовані сімейства інтегральних мікросхем декодерів (демультимплексорів) 2 до 4, 3 до 8, 4 до 10, 4 до 16 представлені в Таблиці 5.3. До інтегральних мікросхем демультимплексорів можна віднести також представлені вище мікросхеми двоспрямованих (мультиплексори/демультимплексори) аналого-

вих комутаторів з цифровим управлінням CD4053B (5.2.3.1°, Рис. 5.6), CD4066B (5.2.3.2°, Рис. 5.7), 74НС4052, 74НС4352, 74НСТ4051, 74НСТ4351, 74НС4067 (5.3.1.2°, Таблиця 5.2).

Таблиця 5.3. Інтегральні мікросхеми демультиплексорів 4/1, 8/1, 10/1, 16/1

Мікросхема	DMS	Функція згідно з технічним паспортом (datasheet)
74LS139	2 до 4	Dual 2-state 2-to-4 line decoder/demultiplexer
74LS155	2 до 4 3 до 8	Dual 2-state 2-to-4 line decoder/demultiplexer: dual 2-to-4 line decoder; dual 1-to-4 line demultiplexer; 3-to-8 line decoder; 1-to-8 line demultiplexer
74LS156	2 до 4 3 до 8	Dual 2-to-4 line decoder/demultiplexer, open-collector output: dual 2-to-4 line decoder; dual 1-to-4 line demultiplexer; 3-to-8 line decoder; 1-to-8 line demultiplexer
74LS138	3 до 8	2-state 3-to-8 line decoder/demultiplexer
74НС137	3 до 8	2-state 3-to-8 line decoder/demultiplexer with address latches
74LS238	3 до 8	2-state 3-to-8 line decoder/demultiplexer, inverting
74НС237	3 до 8	2-state 3-to-8 line decoder/demultiplexer with address latches, inverting
74LS42	4 до 10	2-state 4-line BCD to 10-line Decimal decoders
74НС154	4 до 16	3-state 4 to 16 line decoder/demultiplexer
SN74159	4 до 16	4 to 16 line decoder/demultiplexer with open-collector output
74НС4514	4 до 16	2-state 4-to-16 line decoder/demultiplexer with input latches
74НС4515	4 до 16	2-state 4-to-16 line decoder/demultiplexer with input latches, inverting

5.3.3. Шифратори і дешифратори

1°. *Шифраторами* прийнято називати комбінаційні цифрові пристрої, які створюють вихідний код з одного активного чисельного входу. При кількості виходів n (утворюють код) шифратори мають зазвичай 2^n входів (число входів, яке може бути закодовано на виході).

2°. *Дешифратори* функціонують протилежно шифратору. При кількості входів n (утворюють код) дешифратори мають зазвичай 2^n виходів (число виходів, яке може бути закодовано на вході). Залежно від кодової комбінації на вході активним стає лише один вихід, що відповідає цій комбінації. Дешифратори, як і демультиплексори, ще називають *декодерами*. Якщо демультиплексор направляє вхідні дані на один вихід залежно від адресної кодової комбінації на вході, то дешифратор встановлює активний рівень сигналу на одному з виходів залежно від кодової комбінації на вході, при цьому на інших виходах утримується пасивний рівень сигналу. Залежно від логічної схеми реалізації

дешифратора, активним на виході може бути як високий (позитивна логіка), так і низький (негативна логіка) рівень сигналу. Низький активний рівень вихідного сигналу на символній схемі мікросхеми позначається кружком. Той чи інший активний рівень сигналу впливає на схему підключення навантаження. На практиці частота застосування високого і низького активних рівнів сигналу (позитивної і негативної логіки) однакова.

3°. Спільне застосування шифраторів і дешифраторів можна навести прикладом управління з цифрової клавіатури семисегментним світлодіодним індикатором (Рис. 5.12).

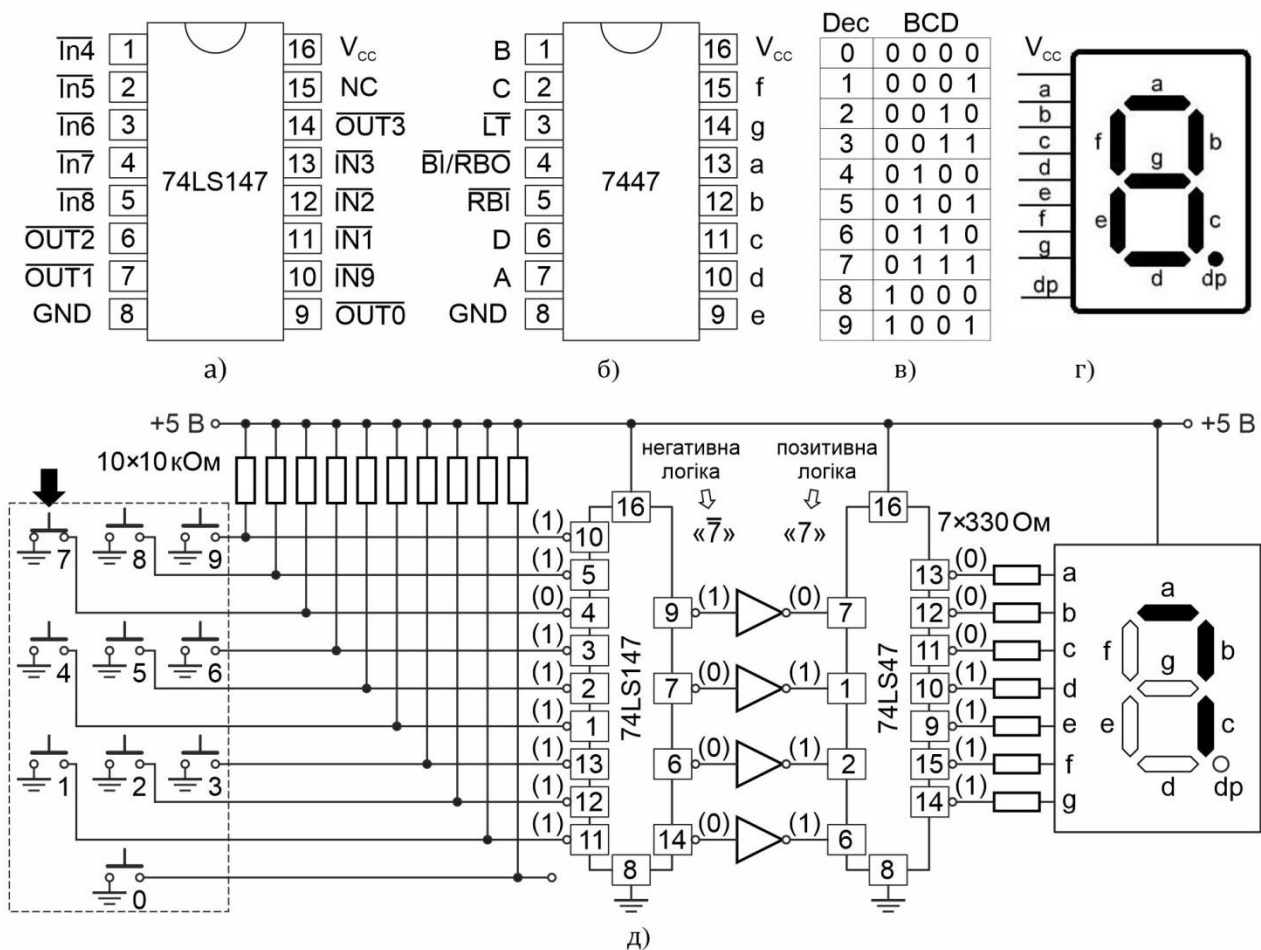


Рис. 5.12. Застосування інтегральних мікросхем шифратора і дешифратора для управління з цифрової клавіатури семи сегментним світлодіодним індикатором, де а), б) призначення контактів мікросхем 74LS147 і 74LS47, шифратора і дешифратора відповідно, в) двійково-десятковий код BCD (binary-coded decimal), г) символна схема семисегментного світлодіодного індикатора, в) схема управління з цифрової клавіатури семисегментним світлодіодним індикатором

Шифратор, реалізований інтегральною мікросхемою 74LS147 (Рис. 5.12. а), шифрує десяткову цифру, в наведеному прикладі цифру 7 (Рис. 5.12. д), отриману натисненням клавіші клавіатури, в двійково-десятковий код (Рис. 5.12. в), застосовуючи негативну логіку. Сигнали негативної логіки на виході шифратора

перетворюються в сигнали позитивної логіки через застосування інверторів. Перетворений таким чином двійково-десятковий код десяткової цифри передається на вхід дешифратора, реалізованого мікросхемою 74LS47 (Рис. 5.12. б). Дешифратор перетворює двійково-десятковий код в код управління семисегментним світлодіодним індикатором (Рис. 5.12. г).

4°. Як і мультиплексори та демультиплексори, шифратори і дешифратори реалізовані сімействами інтегральних мікросхем, з якими можна ознайомитись з довідкової літератури або з каталогів продукції від виробника інтегральних мікросхем цифрових логічних пристроїв.

5.3.3. Компаратори і суматори

1°. *Компараторами* називають комбінаційні цифрові пристрої, які порівнюють два двійкових числа, що подаються на вхід компаратора. На одному з виходів компаратора, відповідному, встановлюється логічна 1, якщо числа на вході рівні. Якщо перше число на вході компаратора більше другого числа, або навпаки, друге число більше першого, логічна 1 встановлюється на одному з двох інших інформаційних виходів компаратора. Отже, залежно від значень двійкових чисел на вході, логічна 1 може бути встановлена на одному інформаційному виході, на інших інформаційних виходах компаратора при цьому встановлюється логічний 0.

Промислово реалізовані сімейства інтегральних мікросхем компараторів для 4-х і 8-ми бітних двійкових чисел. Для порівняння чисел більшої розрядності застосовується відповідна схема підключення двох і більше компараторів меншої розрядності.

2°. Двійковими *суматорами* називають комбінаційні цифрові пристрої, що виконують операції додавання і віднімання двійкових чисел, які подаються на вхід суматора.

Інтегральні мікросхеми двійкових суматорів, як і деякі інші спеціалізовані мікросхеми комбінаційних цифрових пристроїв (шифратори, дешифратори), в теперішній час майже не застосовуються через широке застосування цифрових пристроїв з програмним управлінням. Як правило, функції цих спеціалізованих мікросхем практично успішно виконують мікроконтролери, втому числі такі, ціна яких на ринку порівняно низька. Зважаючи на низьку швидкодію виконання функцій мікроконтролером порівняно з спеціалізованими мікросхемами, при необхідності мати високу швидкодію рекомендується практичне застосування інших цифрових пристроїв з програмним управлінням – програмованих користувачем вентиляційних матриць (ПКВМ).

5.4. ВПРАВИ І ЗАВДАННЯ

5.4.1. Вправи для виконання

1°. Ознайомитись з технічними паспортами (datasheet) інтегральних мікросхем комбінаційних цифрових пристроїв, в тому числі комбінованих цифро-аналогових пристроїв (варіанти представлені в зведеній таблиці для вправ).

2°. Побудувати схему практичного застосування комбінаційних цифрових пристроїв, в тому числі комбіновану схему цифроаналогового перетворювача (варіанти представлені в зведеній таблиці для вправ). При необхідності провести необхідні розрахунки для визначення номінальних значень фізичних величин напруги й опору для аналогових компонентів схеми. Перевірити працездатність схеми, для чого скласти загальну впорядковану таблицю (таблицю істинності) вхідних значень змінних – цифрових і аналогових для побудованої схеми комбінаційного пристрою.

3°. При наявності технічної можливості рекомендується побудову і перевірку працездатності схеми комбінаційного пристрою згідно вибраного (заданого) варіанту вправи 5.4.1.2° здійснити з застосуванням програмного середовища Micro-Cap 12 або іншого програмного застосунку – симулятора аналогових і цифрових електронних схем з вбудованим редактором схем. Рекомендації по вибору в каталозі Micro-Cap 12 основних компонентів для побудови схеми вибраного (заданого) варіанту вправи 5.4.1.2° представлені в зведеній таблиці для вправ.

4°. Варіанти вправ 5.4.1.1° і 5.4.1.2° з урахуванням рекомендації 5.4.1.3° зведені в одну таблицю.

Варіант	Вправа 1 (5.4.1.1°)	Посилання
1.	<i>Інтегральна мікросхема 74LS147</i> – 10-Line to 4-Line Priority Encoder – Шифратор десяткових цифр в двійково-десятковий код.	Рис. 5.12. (5.3.3.3°)
	<i>Інтегральна мікросхема 74LS47</i> – BCD-to-Seven-Segment Decoders/Drivers – Дешифратор двійково-десятьового коду в код управління семисегментним світлодіодним індикатором	(6.1.2.5°) (Рис. 6.6)
2.	<i>Інтегральна мікросхема CD4066</i> – CMOS Quad Bilateral Switch – Аналоговий мультиплексор/демультиплексор – 4-х каналний аналоговий двоспрямований перемикач з цифровим управлінням	(6.1.2.2°) (Рис. 6.3)
	<i>Інтегральна мікросхема LM741</i> – Operational Amplifier – Операційний підсилювач	(6.1.2.3°) (Рис. 6.4)

Варіант	Вправа 2 (5.4.1.2°)	Приклад
1.	Схема управління з цифрової клавіатури семисегментним світлодіодним індикатором з застосуванням інтегральних мікросхем шифратора 74LS147 та дешифратора 74LS47	Рис. 5.12. (5.3.3.3°)
2.	Схема чотирьохрозрядного цифроаналогового перетворювача (ЦАП) на двійково вагомих резисторах та інтегральних мікросхемах аналогового мультиплексора/демультиплексора CD4066 і операційного підсилювача LM741. Вихідні дані: $R_{зз} = 8 \text{ кОм}$; $U_{вх} = 5 \text{ В}$; $U_{вих} = 0 \div 1 \text{ В}$; $\Delta U_{вих} = 0,1 \text{ В}$; Розрахувати: номінальні значення двійково вагомих резисторів R_A ; R_B ; R_C ; R_D .	Рис. 5.7. (5.2.3.1°)
Варіант	Рекомендації (5.4.1.3°)	Застосунок
1.	Вибір основних компонентів схеми – Component/Find Component/74LS147 – Component/Find Component/74LS47 – Component/Find Component/Animated Seven Segment Display	Micro-Cap 12
2.	Вибір основних компонентів схеми – Component/Find Component/CD4066 (схема одного каналу CD4066 з чотирьох типових каналів) – Component/Find Component/LM741C Component/Analog Library/Opamp/General/L-/LM358-/LM741C	Micro-Cap 12

5.4.2. Контрольне завдання

1°. Виконати вправи 5.4.1.1° і 5.4.1.2° для одного варіанту. Варіант вибирається за парним чи непарним номером по списку групи. При можливості, вправу 5.4.1.2° виконати з урахуванням рекомендацій 5.4.1.3°.

2°. По результатам виконання вправ оформити звіт. При виконанні вправи 5.4.1.2° з застосуванням симулятора електронних схем (5.4.1.3°) для захисту звіту підготувати відповідний демонстраційний файл, наприклад, в форматі CIR при застосуванні програмного середовища Micro-Cap 12.

5.4.3. Посилання на каталоги цифрових інтегральних мікросхем

1°. При опрацюванні цього і подальших практикумів (Практикуми 5, 6, 7), предметом розгляду яких є цифрові пристрої, реалізовані інтегральними мікросхемами, варто послуговуватися технічними паспортами (datasheet) безпосередньо від виробника мікросхем. Для прикладу, нижче подаються доступні посилання на веб-ресурси деяких виробників інтегральних мікросхем.

- [1]. Електронний каталог інтегральних мікросхем фірми Texas Instruments (США) доступний при виборі вкладки «Products» за посиланням:
<https://www.ti.com/>

- [2]. Технічна документація на інтегральні мікросхеми цифрових пристроїв корпорації ON Semiconductor (США) доступна при виборі продукції на розгалужених вкладках за посиланням:

<https://www.onsemi.com/design/resources/technical-documentation>

2°. Нижче подаються, для прикладу, посилання на електронний каталог та на доступний в форматі PDF технічний паспорт (datasheet) цифрових інтегральних мікросхем фірми Texas Instruments, які можуть бути корисними при виконанні вправ 5.4.1.1° і 5.4.1.2°.

- [1]. Інтегральна мікросхема 74LS147 10-Line to 4-Line Priority Encoder – шифратор десяткових цифр в двійково-десятковий код. Доступні посилання:

<https://www.ti.com/sitesearch/en-us/docs/universalsearch.tsp?langPref=en-US&searchTerm=74HC147&nr=3#q=74HC147&numberOfResults=25>

https://www.ti.com/lit/ds/symlink/cd74hc147.pdf?ts=1678807434432&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FCD74HC147%252Fpart-de-tails%252FCD74HC147E%253Futm_source%253Dgoogle%2526utm_medium%253Dcpc%2526utm_campaign%253Ddocb-tistore-promo-asc_opn_en-cpc-storeic-google-ww%2526utm_content%253DDevice%2526ds_k%253DCD74HC147E%2526DCM%253Dyes%2526gclid%253DCj0KCQjwtsCgBhDEARIsAE7RYh1ZJ5Pra-sK8vK-ozQ8K8KxblhdkvaH7ehyJBtUcScWBAyWL9M_pFgaAijXEALw_wcB%2526gclsrc%253Daw.ds

- [2]. Інтегральна мікросхема 74LS47 BCD-to-Seven-Segment Decoders/Drivers – дешифратор двійково-десятьового коду в код управління семисегментним світлодіодним індикатором. Доступні посилання:

<https://www.ti.com/sitesearch/en-us/docs/universalsearch.tsp?langPref=en-US&searchTerm=74ls47&nr=14#q=74ls47&numberOfResults=25>

https://www.ti.com/lit/ds/symlink/sn74ls47.pdf?ts=1678789058068&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253D7%2Bsegemnt%2Bdecoder%2526nr%253D6

- [3]. Інтегральна мікросхема CD4066 CMOS Quad Bilateral Switch – 4-х каналний аналоговий двоспрямований перемикач з цифровим управлінням (найпростіший аналоговий мультиплексор/демультиплексор). Доступні посилання:

<https://www.ti.com/product/CD4066B?keyMatch=CD4066>

https://www.ti.com/lit/ds/symlink/cd4066b.pdf?ts=1678778879959&ref_url=https%253A%252F%252Fwww.google.com%252F

- [4]. Інтегральна мікросхема LM741 Operational Amplifier – операційний підсилювач. Доступні посилання:

<https://www.ti.com/sitesearch/en-us/docs/universalsearch.tsp?langPref=en-US&searchTerm=lm741&nr=112#q=lm741&numberOfResults=25>

https://www.ti.com/lit/ds/symlink/lm741.pdf?ts=1678791988752&ref_url=https%253A%252F%252Fwww.ti.com%252Fsitesearch%252Fen-us%252Fdocs%252Funiversalsearch.tsp%253FlangPref%253Den-US%2526searchTerm%253Dlm741%2526nr%253D112

ПРАКТИКУМ 6. ІМПУЛЬСНІ ПРИСТРОЇ

6.1. ДОВІДКОВІ ВІДОМОСТІ

6.1.1. Імпульсні пристрої, тактові сигнали і одиничний імпульс

1°. *Імпульсні* електронні пристрої – пристрої, що працюють в переривистому (імпульсному, дискретному) режимі. В цифровій схемотехніці важливу роль відіграють пристрої, які виконують функцію формування, генерування та перетворення імпульсів прямокутної форми. Зокрема, *генератори тактових сигналів* (генератори прямокутних коливань) застосовуються для забезпечення переміщення біту інформації через логічні елементи та тригери зі швидкістю, яка визначається частотою тактового генератора, а *генератори одиночних імпульсів* – для керування елементами цифрових пристроїв.

2°. Параметри періодичної послідовності тактових сигналів (Рис. 6.1) – *амплітуда імпульсу* U_m , *період* T і *частота* f тактових сигналів, *тривалість імпульсу* τ і *тривалість паузи* $\tau_{\text{п}}$, *прогальність* або *шпаруватість* s , і *коефіцієнт заповнення* q :

$$T = \tau + \tau_{\text{п}}; \quad f = \frac{1}{T}; \quad s = \frac{T}{\tau_{\text{п}}} = \frac{1}{q}; \quad q = \frac{T}{\tau} = \frac{1}{s}. \quad (6.1.)$$

Послідовність імпульсів, для якої $\tau = \tau_{\text{п}}$ і $s = 2$ називається *меандром*.

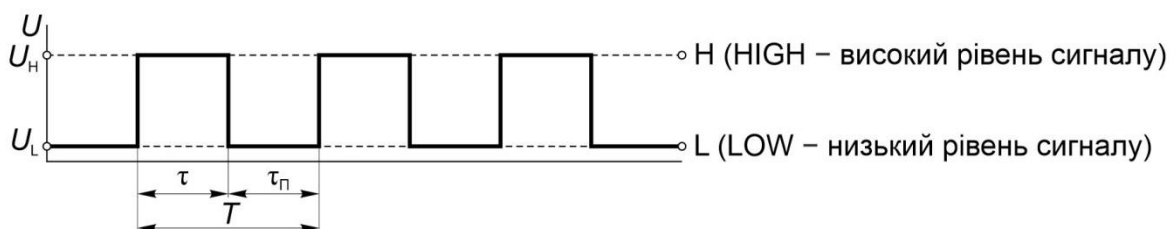


Рис. 6.1. Періодична послідовність тактових імпульсів (меандр, $\tau = \tau_{\text{п}}$)

3°. При проходженні через схему імпульс може змінювати свої параметри (Рис. 6.2 а). Основні характеристики вихідного імпульсу при цьому (Рис. 6.2 б) – *час наростання* $\tau_{\text{н}}$ і *час спадання* $\tau_{\text{с}}$ імпульсу, *тривалість* $\tau_{\text{т}}$ і *час затримки* $\tau_{\text{з}}$ імпульсу. Час наростання $\tau_{\text{н}}$ визначається як інтервал часу, протягом якого амплітуда імпульсу наростає від 10 % до 90 % свого максимального значення, а час спадання $\tau_{\text{с}}$ – як інтервал часу, протягом якого амплітуда імпульсу спадає від 90 % до 10 % максимального значення. Наприклад, на виході базових логічних елементів серії 74LS00 час наростання і час спадання імпульсу зазвичай складає кільканадцять наносекунд. Тривалість $\tau_{\text{т}}$ імпульсу визначається як ін-

тервал часу між двома значеннями величиною 50 % максимального значення імпульсного сигналу. Час затримки τ_z визначається як інтервал часу від початку імпульсного сигналу ($t = 0$) і його величиною 10 % максимального значення.

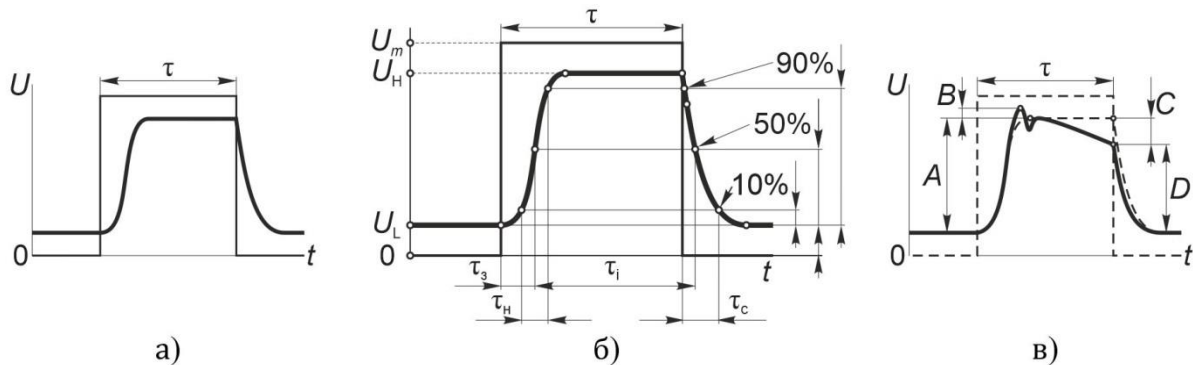


Рис. 6.2. Характеристики імпульсу, де а) вхідний (1) і вихідний (2) імпульси, б) вихідний імпульс і його характеристики, в) викид і нахил на вершині імпульсу

Можливі *викид* $E(\%)$ і *нахил* $I(\%)$ в верхній частині імпульсу вимірюються у процентах (Рис. 6.2 в):

$$E(\%) = \frac{B}{A} \cdot 100\%; \quad I(\%) = \frac{C}{D} \cdot 100\%. \quad (6.2.)$$

Граничні значення характеристик вихідних імпульсів цифрових інтегральних мікросхем вказуються в технічних паспортах на інтегральну мікросхему.

6.1.2. Мультивібратори

1°. В Практикумі 4 показані деякі приклади застосування мультивібраторів і визначені три типи мультивібраторів за принципом формування імпульсних сигналів – моностабільні, бістабільні, астабільні мультивібратори (4.1.1.4°).

Астабільні мультивібратори, або *генератори прямокутних коливань*, формують на виході послідовність імпульсів прямокутної форми в режимі автоколивань, без впливу зовнішніх сигналів (6.1.2.2°).

Бістабільні мультивібратори, або *тригери*, формують на виході імпульси прямокутної форми під впливом двох послідовних вхідних сигналів. Перший вхідний сигнал формує фронт, другий – спадання вихідного імпульсу (6.1.2.3°). В цифровій схемотехніці широко застосовується тригер Шмітта (6.1.2.4°).

Моностабільні мультивібратори, або *одновібратори*, формують на виході одиночний прямокутний імпульс під впливом сигналу на вході (6.1.2.5°).

В електронній техніці, в тому числі цифровій, використовуються різноманітні варіанти схем мультивібраторів, які різняться між собою за типом використовуваних елементів (транзисторні, тиристорні, мікроелектронні, інші), видами зв'язку між підсилювальними елементами, способами регулювання тривалості і частоти генерованих імпульсів, інше.

2°. Одними з найпоширеніших генераторів імпульсів прямокутної форми є *релаксаційні* генератори електричних прямокутних коливань, астабільні мультивібратори, які ще в деякій спеціальній літературі називають *несинхронізованими* мультивібраторами. Такий мультивібратор являє собою двокаскадний резистивний підсилювач з додатнім зворотним зв'язком (Рис. 6.3. а).

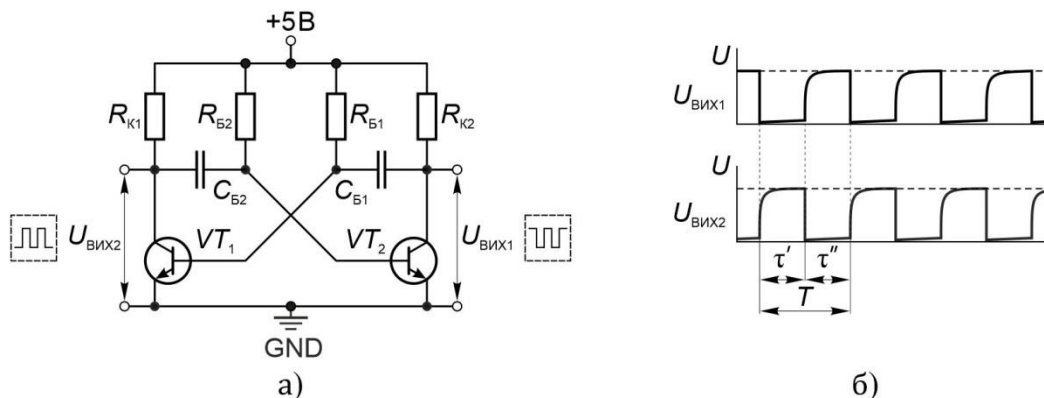


Рис. 6.3. Астабільний мультивібратор, де а) схема, б) вихідні сигнали

Не акцентуючи увагу на складній взаємодії струмів в симетричних електричних ланцюгах мультивібратора, принцип його роботи можна пояснити наступним чином. Кожен з двох, пов'язаних між собою, біполярних *n-p-n* транзисторів мультивібратора працює в режимі транзисторного ключа, тобто, виконує перемикальну функцію (3.1.3.3°). Внаслідок того, що колектор одного транзистора пов'язаний з базою іншого транзистора через конденсатор, транзистори керують один одним так, що в момент відкриття одного транзистора інший миттєво закривається, і навпаки – перший транзистор миттєво закривається при відкритті другого. Періодична чергова послідовність відкриття і закриття транзисторів генерує на двох симетричних виходах мультивібратора дві послідовності імпульсів, за рівнями сигналу обернених одна до одної. При НИЗЬКОМУ (LOW) рівні електричного сигналу однієї вихідної послідовності імпульсів інша вихідна послідовність матиме ВИСОКИЙ (HIGH) рівень сигналу і навпаки, при ВИСОКОМУ (HIGH) рівні сигналу першої вихідної послідовності друга послідовність матиме НИЗЬКИЙ (LOW) рівень сигналу (Рис. 6.3. б). Період відкриття і закриття транзисторів та, відповідно, період T послідовності імпульсів визначаються швидкістю заряду та розряду конденсаторів. Отже, можливе регулювання періоду T підбором номінальних значень опорів і ємності симетричних RC ланцюгів мультивібратора (Рис. 6.3. а) як:

$$\begin{aligned} \tau' &= C_{B1} R_{B1} \ln 2; & \tau'' &= C_{B2} R_{B2} \ln 2; & T &= \tau' + \tau''; \\ \text{при } \left. \begin{aligned} C_{B1} &= C_{B2} = C \\ R_{B1} &= R_{B2} = R \end{aligned} \right\} & \Rightarrow \tau' &= \tau'' & \Rightarrow T &= 2CR \ln 2. \end{aligned} \quad (6.3.)$$

Тут, вихід 2 мультивібратора вважається прямим, а вихід 1 інверсним.

3°. Бістабільний мультивібратор (тригер) на біполярних n - p - n транзисторах, як і астабільний мультивібратор, будується по симетричній схемі з двома входами і двома виходами – прямим і інверсним (Рис. 6.4. а).

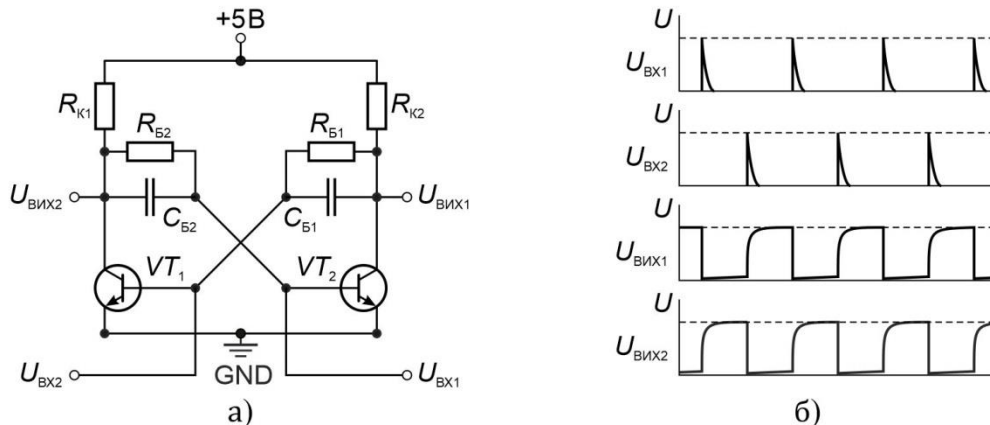


Рис. 6.4. Бістабільний мультивібратор, де а) схема, б) входні і вихідні сигнали

Тут, вихід 2 мультивібратора вважається прямим, а вихід 1 інверсним. При появі пускового імпульсу на вході 1 формується фронт імпульсу прямої послідовності на виході 2 і спадання імпульсу інверсної послідовності на вході 1. І навпаки, при появі пускового імпульсу на вході 2 формується фронт імпульсу на виході 1 і спадання імпульсу на вході 2 (Рис. 6.4. б). Отже, залежно від керуючих сигналів на входах симетричного бістабільного мультивібратора, на його виходах встановлюється один з двох стабільних рівнів сигналу, ВИСОКИЙ (HIGH) або НИЗЬКИЙ (LOW) рівень сигналу.

4°. Для цифрової схемотехніки значний практичний інтерес представляє тригер Шмітта (5.1.2.3°). На Рис. 6.5 показаний бістабільний мультивібратор на біполярних n - p - n транзисторах, побудований по несиметричній схемі з одним входом і виходом, – тригер Шмітта.

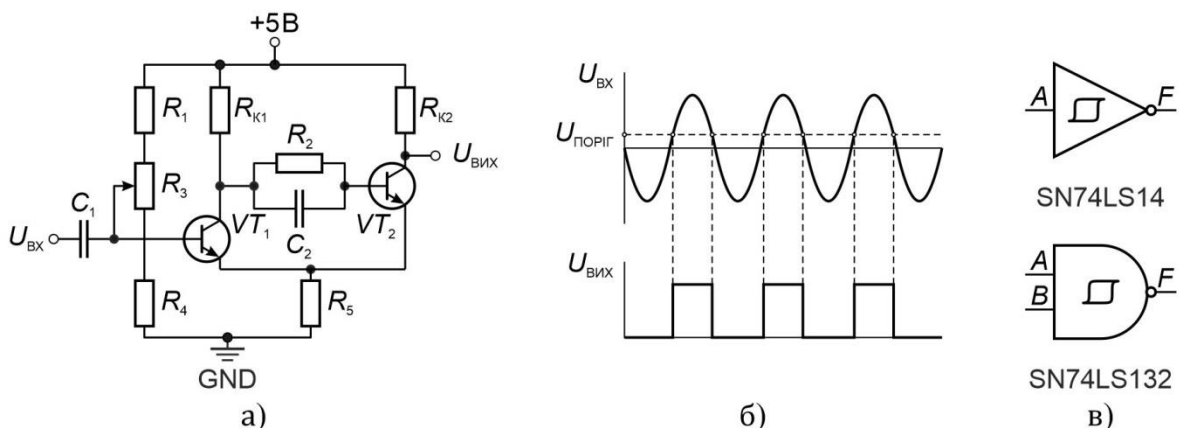


Рис. 6.5. Тригер Шмітта, де а) схема, б) входний і вихідний сигнали, в) схеми логічних елементів НЕ (NOT) і І-НЕ (NAND) з тригером Шмітта та їх інтегральні мікросхеми

На виході тригера Шмітта отримуються прямокутні імпульси при входньому сигналі довільної форми. Схема тригера Шмітта (Рис. 6.5. а) побудована

так, що в імпульс прямокутної форми перетворюються ті сигнали, поточне значення яких перевершує деяке задане порогове значення $U_{\text{ПОРІГ}}$ (Рис. 6.5. б), яке можна встановити, наприклад, застосуванням потенціометра R_3 (Рис. 6.5. а).

Для багатьох пристроїв цифрової логіки, реалізованих інтегральними мікросхемами, характерна наявність логічних схем з тригерами Шмітта на вході (Рис. 6.5. в), що є важливим для багатьох практичних застосувань, коли до цифрових пристроїв необхідно підключити пристрої з аналоговим виходом. На логічних елементах з тригером Шмітта на вході (Рис. 6.5. в) можлива реалізація генератора тактових імпульсів (6.1.4.1°, 6.1.4.2°).

5°. Моностабільний мультивібратор (одновібратор) на біполярних n - p - n транзисторах можна побудувати по симетричній схемі, як астабільний мультивібратор, так і по несиметричній схемі, як тригер Шмітта (Рис. 6.6. а).

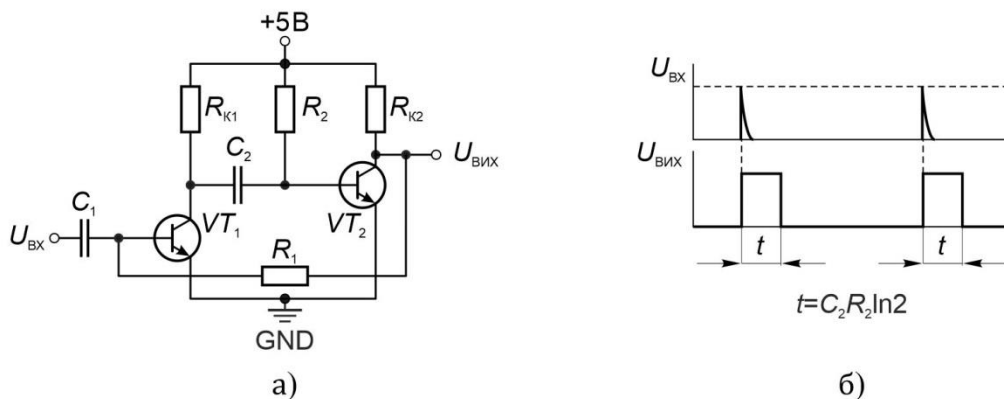


Рис. 6.6. Моностабільний мультивібратор, де а) схема, б) входні і вихідні сигнали

На виході моностабільного мультивібратора генерується прямокутний імпульс при появі на вході керуючого сигналу (Рис. 6.6. б), тому такий мультивібратор в спеціальній літературі ще називають чекаючим (standby) мультивібратором. Моностабільні мультивібратори реалізовані сімействами інтегральних мікросхем, наприклад, серій SN74LS121, SN74LS122, SN74LS123, SN74LS221, SN74LS423, SN74HC4538.

6.1.3. Схеми мультивібраторів на основі мікросхеми таймера 555

1°. Поширеною в практичному застосуванні є багатофункціональна інтегральна мікросхема таймера NE555 (Рис. 6.7) розробки фірми Signetics (1972, США). Функціонально схема мікросхеми таймера NE555 містить два аналогові компаратори, тригер, розрядний транзистор та вихідний каскад (Рис. 6.7. б).

На основі таймера NE555 реалізуються, серед інших схем, схеми астабільного мультивібратора – генератора послідовних тактових імпульсів (Рис. 6.8. а) та моностабільного мультивібратора – генератора одиночних імпульсів, які виникають під впливом входного сигналу (Рис. 6.8. б).

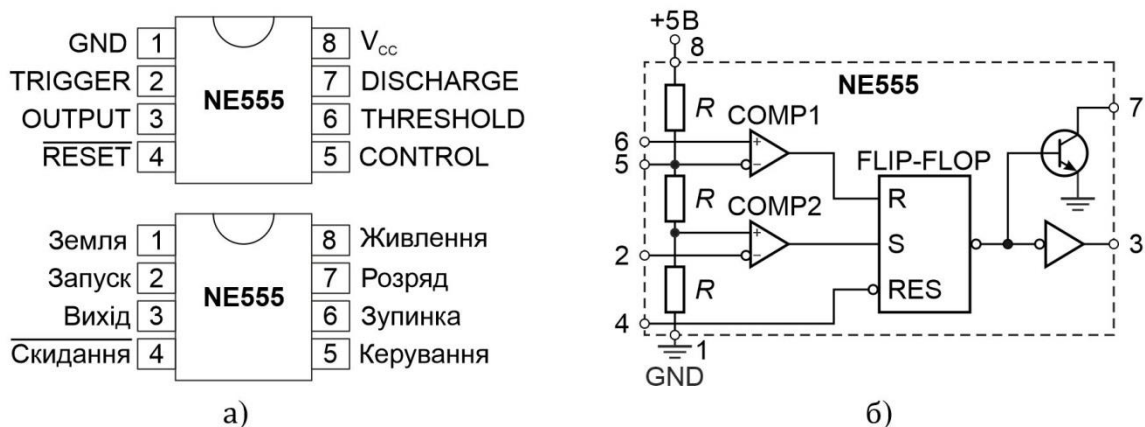


Рис. 6.7. Інтегральна мікросхема таймера NE555, де а) призначення контактів мікросхеми, б) функціональна схема

2°. Схема астабільного мултивібратора на основі таймера NE555 окрім самої мікросхеми містить діод, резистори та конденсатори (Рис. 6.8. а). Частота f , коефіцієнт заповнення q і шпаруватість s тактових сигналів (6.1.1.2°) мултивібратора за схемою на основі таймера NE555 (Рис. 6.8. а) визначаються як

$$f = \frac{1}{(R_A + R_B)C_{3B} \ln 2}, \quad q = \frac{R_A}{R_A + R_B}, \quad s = \frac{1}{q}, \quad (6.4.)$$

де частота отримується в герцах, якщо опір і ємність задані в омах і фарадах відповідно. З (6.4.) випливає, що на виході мултивібратора меандр (6.1.1.2°) отримується коли $s = 2$ при $R_A = R_B$.

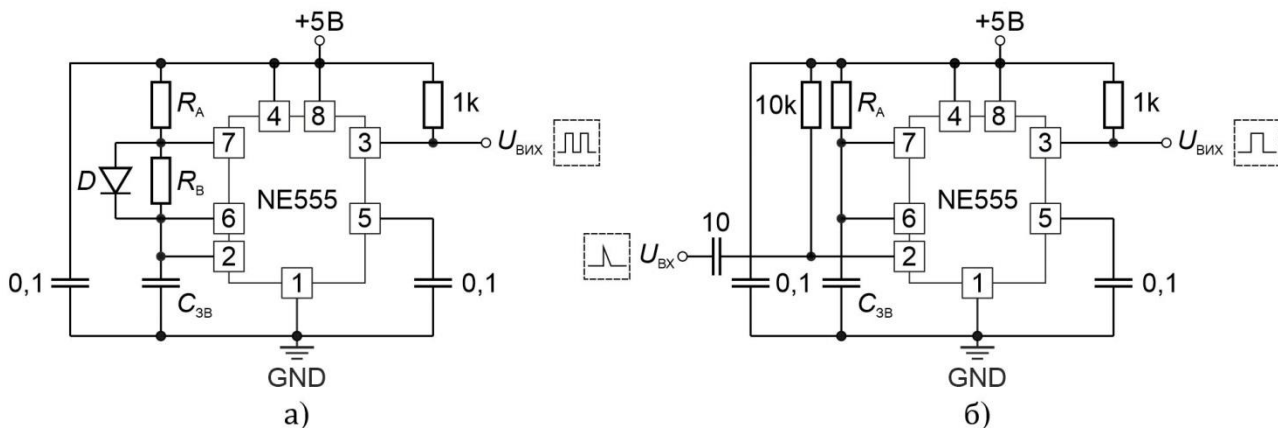


Рис. 6.8. Схеми мултивібраторів на основі інтегральної мікросхеми таймера NE555, де а) астабільний мултивібратор – генератор тактових імпульсів, б) моностабільний мултивібратор – генератор одиничних керувальних імпульсів

3°. Схема моностабільного мултивібратора на основі таймера NE555 представлена на Рис. 6.8. б). Тривалість імпульсу t (Рис. 6.6. б) мултивібратора за схемою на основі таймера NE555 (Рис. 6.8 б) визначаються як

$$t = 1,1R_A C_{3B}, \quad (6.5.)$$

де час отримується в секундах, якщо опір і ємність задані в омах і фарадах відповідно.

6.1.4. Практичні схеми реалізації генераторів тактових сигналів

1°. Найпростіші практичні схеми генераторів прямокутних тактових сигналів можна скласти на дискретних логічних елементах – на основі базових логічних елементів інвертора НЕ (NOT), реалізованих, наприклад, інтегральною мікросхемою 74НС04 (Рис. 6.9. а).

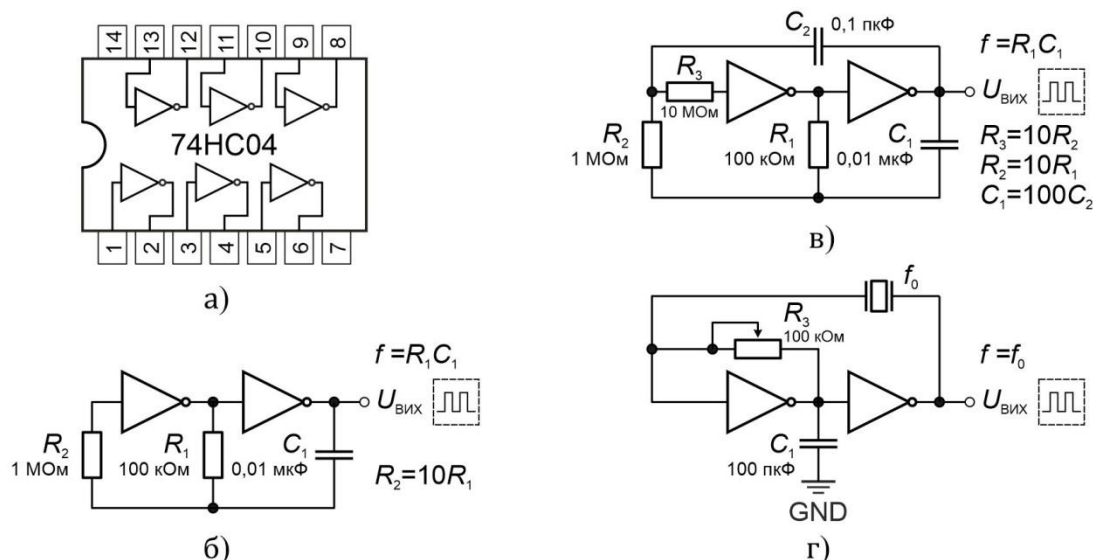


Рис. 6.9. Генератори тактових сигналів на логічних елементах інвертора НЕ (NOT), де а) інтегральна мікросхема 74НС04 інверторів НЕ (NOT), б) ÷ г) схеми генераторів

Схема генератора тактових сигналів з застосуванням мікросхеми 74НС04 за технологією КМОП (Рис. 6.9. а) складається з двох інверторів НЕ (NOT) і являє собою RC генератор коливань або кварцовий генератор коливань. В першому випадку частота f тактових сигналів визначається постійною часу $R_1 C_1$ генераторів (Рис. 6.9. б, в), в другому – частотою кварцу, наприклад, $f_0 = 20$ МГц (Рис. 6.9. г). Перша схема (Рис. 6.9. б) має певні недоліки – можливий не запуск генератора та можливе відхилення частоти тактових сигналів від розрахункових. Друга схема (Рис. 6.9. в) за рахунок введення додаткового RC контуру не має таких недоліків. Третя схема (Рис. 6.9. в) з кварцом застосовується тоді, коли необхідно забезпечити високу стабільність частоти вихідних тактових сигналів.

2°. Схеми генераторів прямокутних тактових сигналів можна скласти також на дискретних логічних елементах І-НЕ (NAND), реалізованих, наприклад, інтегральною мікросхемою 74LS00 за технологією ТТЛ (Рис. 6.10. а).

Схема генератора тактових сигналів з застосуванням мікросхеми 74 LS00 складається з двох логічних елементів І-НЕ (NAND) і являє собою RC генератор коливань. Частота f тактових сигналів визначається постійною часу $R_1 C_1$ генератора (Рис. 6.9. б) і при певному виборі номінальних значень опору R_1 і ємності

C_1 може досягти максимального значення біля 2 МГц. Тут (Рис. 6.9. б) схема доповнена вхідним сигналом $U_{ДЗ}$ дозволу появи вихідної послідовності тактових сигналів, який з'являється при ВИСОКОМУ (HIGH) рівні сигналу $U_{ДЗ}$. Можна застосувати стандартну схему генератора без керуючого сигналу на вході, тоді цей перший, керуючий, вхід необхідно підключити до другого входу цього ж логічного елементу I-HE (NAND).

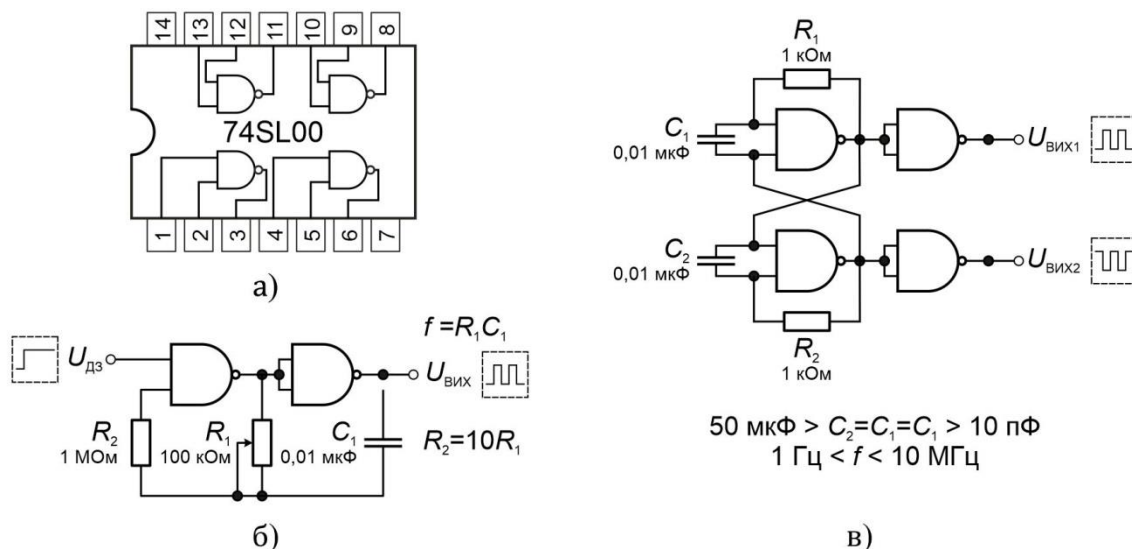


Рис. 6.10. Генератори тактових сигналів на логічних елементах I-HE (NAND), де а) інтегральна мікросхема 74LS00 логічних елементів I-HE (NAND), б) схема генератора тактових сигналів з керуючим сигналом дозволу роботи, г) схеми генератора з прямою і інверсною послідовністю вихідних тактових сигналів

Друга схема (Рис. 6.10. в) генератора тактових сигналів з застосуванням мікросхеми 74LS00 базових логічних елементів I-HE (NAND) побудована по схемі SR тригера (Практикум 7) з резисторами зворотного зв'язку і RC контуром. Такий генератор генерує на виході пряму і інверсну послідовності тактових сигналів. Відношення C_1/C_2 (C_2/C_1) номінальних значень ємностей визначає коефіцієнт заповнення q (шпаруватість s) послідовності тактових сигналів.

3°. Серія 7400 інтегральних мікросхем цифрових пристроїв містить спеціалізовану мікросхему 74LS124 з двома генераторами тактових сигналів, керованих напругою – Dual Voltage Controlled Oscillators (Рис. 6.11. а), які вирізняються високою стабільністю частоти тактових сигналів в широкому діапазоні – від 0,12 Гц до 30 МГц.

Частота тактових сигналів генераторів мікросхеми 74LS124 визначається лише RC контуром – зовнішнім конденсатором ($C_{ЗВ}$ на Рис. 6.11. а) і внутрішнім резистором. Кожен з двох генераторів має два входи для регулювання частоти – вхід «регулювання частоти» ($f_{\text{регулятор}}$ на Рис. 6.11. а) і вхід «діапазон частоти» ($f_{\text{діапазон}}$), а також вхід керування генератором (U). При ВИСОКОМУ (HIGH)

рівні сигналу U забороняється проходження сигналу на вихід генератора, генератор запускається і працює при НИЗЬКОМУ (LOW) рівні сигналу U на вході. Живлення мікросхеми 74LS124 здійснюється окремо для генератора і сигналів керування на вході для забезпечення взаємної ізоляції і запобігання шкідливого впливу на логічні рівні сигналів.

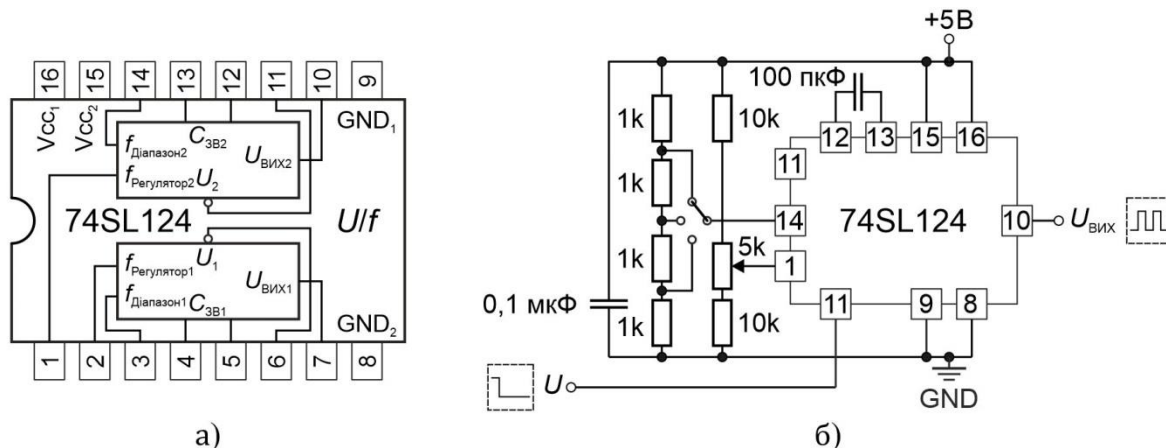


Рис. 6.11. Інтегральна мікросхема 74LS124 генератора тактових сигналів, керованих напругою, де а) функціональна схема мікросхеми, б) практична схема генератора тактових сигналів з застосуванням мікросхеми 74LS124

Схема генератора тактових сигналів з застосуванням мікросхеми 74LS124 (Рис. 6.11. б) містить два зовнішні ланцюга, які дозволяють змінювати частоту на виході генератора. Один ланцюг може змінювати частоту дискретно за допомогою позиційного перемикача, а інший безперервно за допомогою потенціометра. Частота f_0 послідовності тактових сигналів визначається ємністю зовнішнього конденсатора $C_{ЗВ}$ для мікросхеми 74LS124 як

$$f_0 = \frac{1 \cdot 10^{-4}}{C_{ЗВ}}, \quad (6.6.)$$

а для мікросхеми 74S124 як

$$f_0 = \frac{5 \cdot 10^{-4}}{C_{ЗВ}}, \quad (6.7.)$$

де частота f_0 отримується в герцах, якщо ємність $C_{ЗВ}$ задана в фарадах.

Генератор тактових сигналів з застосуванням мікросхеми 74LS124 може працювати в режимі кварцового генератора, якщо зовнішній конденсатор замінити на кристал кварцу.

6.2. ВПРАВИ І ЗАВДАННЯ

6.2.1. Вправи для виконання

1°. Побудувати схему мультивібратора – генератора прямокутних сигналів (варіанти). Визначити номінальні значення опору і ємності RC-ланцюгів мультивібратора. Побудувати вихідні графічні характеристики мультивібратора, визначити частоту і коефіцієнт заповнення послідовності тактових сигналів або час тривалості імпульсу (для моностабільного мультивібратора). Зробити висновки.

2°. Ознайомитись з технічними паспортами (datasheet) інтегральних мікросхем, застосованих в пристроях при виконанні вправи 6.2.1.1° (Вправа 1).

3°. При наявності технічної можливості рекомендується побудову і перевірку працездатності схеми імпульсного пристрою згідно вибраного (заданого) варіанту вправи 6.2.1.1° здійснити з застосуванням програмного середовища Micro-Cap 12 або іншого програмного застосунку – симулятора аналогових і цифрових електронних схем з вбудованим редактором схем.

4°. Варіанти вправи 6.2.1.1° (Вправа 1) зведені в таблиці. Приклад визначення номінальних значень опору і ємності RC-ланцюгів мультивібратора показаний для Варіанту 1.

Варіант	Пристрій	Посилання на приклад
1.	Астабільний мультивібратор (генератор) на біполярних $n-p-n$ транзисторах <i>Приклад визначення номінальних значень опору і ємності RC-ланцюгів мультивібратора:</i> $R_{K1} = R_{K2} = R_K = 510 \text{ Ом};$ $R_{B1} = R_{B2} = R_B = 18 \text{ кОм};$ $R_B \gg R_K;$ $C_{B1} = C_{B2} = C_B = 10 \text{ нФ}.$ <i>Примітка:</i> для миттєвого збудження симетричного мультивібратора, при запуску його роботи в програмному середовищі Micro-Cap 12, рекомендується встановити початковий заряд одного з конденсаторів через відповідне встановлення його атрибутів, наприклад, як 10N IC=2, що означає початковий заряд 2 В конденсатора ємністю 10 нФ.	(6.1.2.2°) (Рис. 6.3)
2.	Бістабільний мультивібратор (тригер) на біполярних $n-p-n$ транзисторах	(6.1.2.3°) (Рис. 6.4)
3.	Тригер Шмітта на біполярних $n-p-n$ транзисторах	(6.1.2.4°) (Рис. 6.5)

Варіант	Пристрій	Посилання на приклад
4.	Моностабільний мултивібратор (одновібратор) на біполярних <i>n-p-n</i> транзисторах	(6.1.2.5°) (Рис. 6.6)
5.	Астабільний мултивібратор (генератор) на інтегральній мікросхемі NE555	(6.1.3.1°) (6.1.3.2°) (Рис. 6.8.а)
6.	Моностабільний мултивібратор (одновібратор) на інтегральній мікросхемі NE 555	(6.1.3.1°) (6.1.3.3°) (Рис. 6.8.б)
7.	Генератор тактових сигналів на інтегральній мікросхемі 74НС04 логічних елементів НЕ (NOT)	(6.1.4.1°) (Рис. 6.9)
8.	Генератор тактових сигналів на інтегральній мікросхемі 74LS00 логічних елементів І-НЕ (NAND)	(6.1.4.2°) (Рис. 6.10)
9.	Генератор тактових сигналів на інтегральній мікросхемі 74LS124	(6.1.4.2°) (Рис. 6.11)

6.2.2. Контрольне завдання

1°. Виконати вправи 6.2.1.1° і 6.2.1.2° для одного варіанту. Варіант вибирається за номером по списку групи. Для чергового номеру, що першим перевершує число варіантів, вибирається варіант 1, і так далі. При можливості, вправу 6.2.1.1° виконати з урахуванням рекомендацій 6.2.1.3°.

2°. По результатам виконання вправ оформити звіт. При виконанні вправи 6.2.1.1° з застосуванням симулятора електронних схем (6.2.1.3°) для захисту звіту підготувати відповідний демонстраційний файл, наприклад, в форматі CIR при застосуванні програмного середовища Micro-Cap 12.

6.2.3. Технічні паспорти (datasheet) цифрових інтегральних мікросхем

1°. Нижче подаються, для прикладу, посилання на доступний в форматі PDF технічний паспорт (datasheet) цифрових інтегральних мікросхем фірми Texas Instruments, які можуть бути корисними при виконанні вправи 6.2.1.1°.

- [1]. Інтегральна мікросхема таймера NE555. Доступне посилання:
https://www.ti.com/lit/ds/symlink/se555.pdf?ts=1680055124031&ref_url=https%253A%252F%252Fwww.google.com%252F
- [2]. Інтегральна мікросхема SN74НСТ04 логічних елементів НЕ (NOT). Доступне посилання:
https://www.ti.com/lit/ds/scls042f/scls042f.pdf?ts=1680040592815&ref_url=https%253A%252F%252Fwww.google.com%252F
- [3]. Інтегральна мікросхема SN74LST00 логічних елементів І-НЕ (NAND). Доступне посилання:
<https://www.ti.com/lit/ds/symlink/sn74ls00.pdf>

- [4]. Інтегральна мікросхема SN74LS629 осцилятора, керованого напругою (подібна генератору тактових сигналів SN74LS124). Доступне посилання:

<https://www.ti.com/lit/ds/symlink/sn74ls624.pdf?ts=1680073423823>

ПРАКТИКУМ 7.

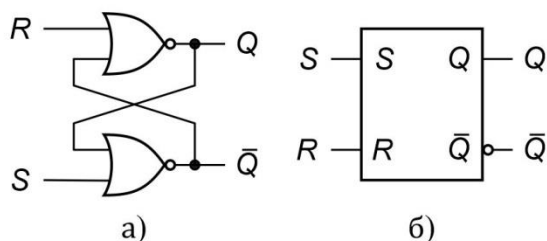
ПОСЛІДОВНІСНІ ЦИФРОВІ ПРИСТРОЇ

7.1. ДОВІДКОВІ ТЕОРЕТИЧНІ ВІДОМОСТІ

7.1.1. Тригери

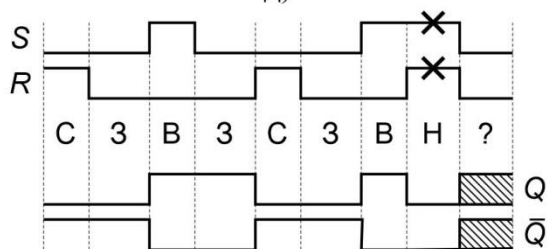
1°. Вище зазначалось (Практикум 5), що цифрові пристрої, схеми яких побудовані на логічних елементах, розділяються на дві гілки – комбінаційні пристрої (5.1.1.1°) і послідовнісні пристрої (5.1.1.2°). На відміну від комбінаційних пристроїв (Практикум 5), послідовнісні пристрої володіють властивістю зберігання інформації. Ці цифрові пристрої виконують певну *послідовність* операцій для отримання, збереження та подальшої видачі біта (бітів) інформації за певною логічною схемою, звідси їх назва – *послідовнісні* цифрові пристрої.

2°. Найпростішим елементарним послідовнісним пристроєм вважається *тригер з встановленням і скиданням* (Set-Reset Flip-Flop) – SR-тригер, який в спеціальній літературі ще називають *прозорою засувкою* (transparent latch). Стандартна логічна схема SR-тригера реалізується на універсальних логічних елементах АБО-НЕ (NOR) (Рис. 7.1 а) та І-НЕ (NAND) (Рис. 7.1 в).

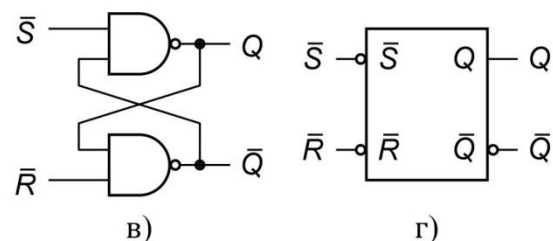


Стан	S	R	Q	$\bar{Q}$
Зберігання (З)	0	0	Q	$\bar{Q}$
Скидання (С)	0	1	0	1
Встановлення (В)	1	0	1	0
Не застосовується (Н)	1	1	0	0

д)

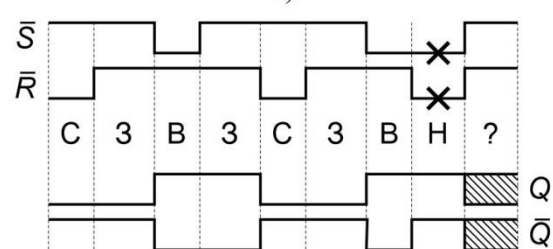


е)



Стан	$\bar{S}$	$\bar{R}$	Q	$\bar{Q}$
Не застосовується (Н)	0	0	1	1
Встановлення (В)	0	1	1	0
Скидання (С)	1	0	0	1
Зберігання (З)	1	1	Q	$\bar{Q}$

е)



ж)

Рис. 7.1. SR-тригер на елементах АБО-НЕ (NOR) та І-НЕ (NAND), де а) і в) логічні схеми, б) і г) символічні схеми, д) і е) таблиці істинності, д) і е) часові діаграми

Символьні схеми SR -тригера на основі універсальних логічних елементів АБО-НЕ (NOR) та І-НЕ (NAND) показані на Рис. 7.1 б) та Рис. 7.1 г) відповідно. Встановлення і скидання логічного значення 1 на виході Q тригера (логічного значення 0 на інверсному виході $\bar{Q}$) відбувається при ВИСОКОМУ (HIGH) рівні відповідних вхідних сигналів встановлення S і скидання R для SR -тригера на логічних елементах АБО-НЕ (NOR), при НИЗЬКОМУ (LOW) рівні – для SR -тригера на логічних елементах І-НЕ (NAND), на що вказують символи (кружки) на схемі тригера (Рис. 7.1 г) та символи інверсії на позначеннях входів $\bar{S}$ і $\bar{R}$.

Певним недоліком SR -тригера вважається те, що при певній послідовності комбінацій значень вхідних змінних на виході виникає нестійкий, отже, невідомий, стан (Рис. 7.1 є, ж). Тому комбінації $S=R=1$ і $\bar{S}=\bar{R}=0$ вхідних станів SR -тригерів не застосовуються (Рис. 7.1 д, е).

3°. Для практичного застосування найпростіші SR -тригери реалізовані інтегральною мікросхемою 74LS279 (Рис. 7.2 а). Запобігання брязкоту контактів (4.1.1.4°, Рис. 4.3, Рис. 7.2 г) – одне з таких застосувань (Рис. 7.2 д).

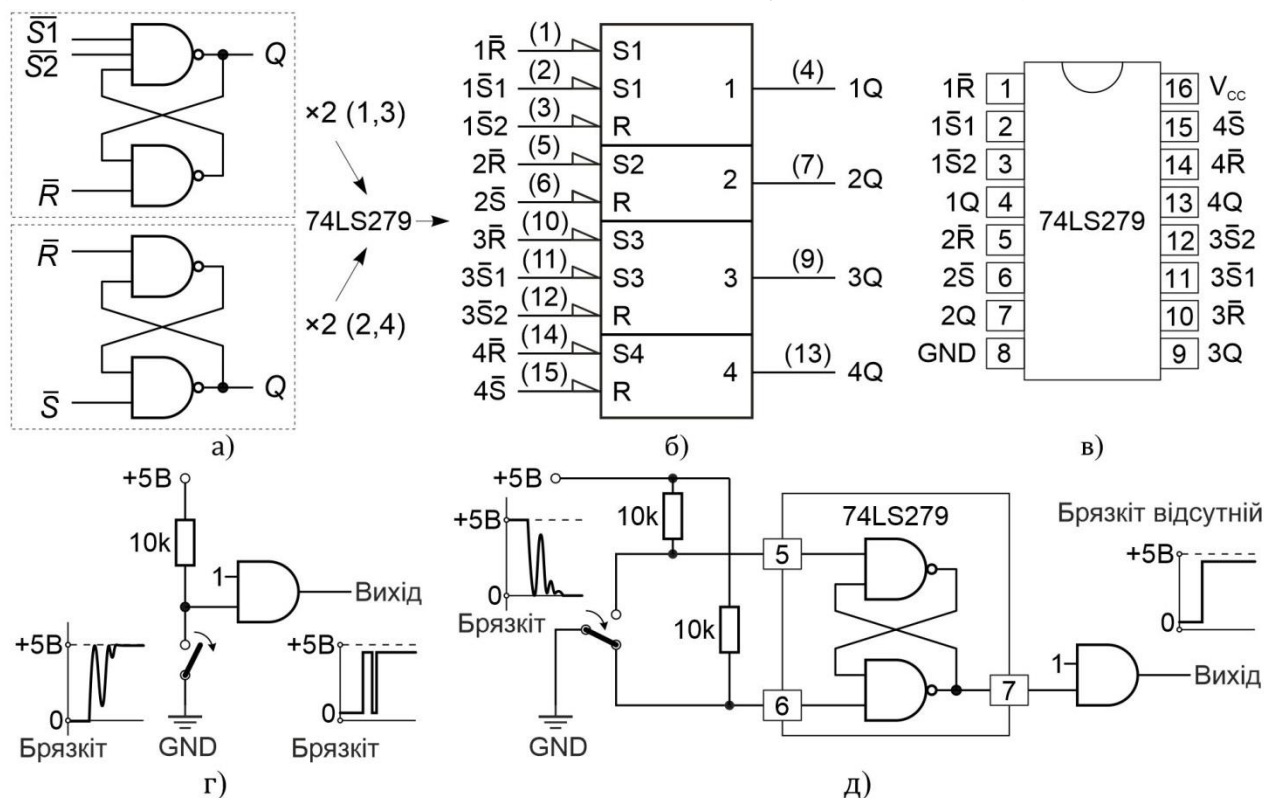


Рис. 7.2. Реалізація SR -тригера інтегральною мікросхемою 74LS279, де а) логічні схеми, б) варіант символічної схеми мікросхеми згідно стандарту ANSI/IEEE 91-1984, в) схема призначення контактів мікросхеми, г) шкідливий ефект брязкоту контактів, д) схема запобігання брязкоту контактів застосуванням одного SR -тригера інтегральної мікросхеми 74LS279

4°. Найпростіші SR -тригери (7.1.1.2°) ще називають *асинхронними* або *статичними* тригерами. SR -тригери, що мають додатковий *синхронізуючий* вхід

тактових сигналів C (CLC – Clock) називають *синхронними* або *тактованими* SR-тригерами. Стан на виходах синхронних SR-тригерів може змінюватися лише в *режимі активації* тригера, синхронно з тактовими сигналами C .

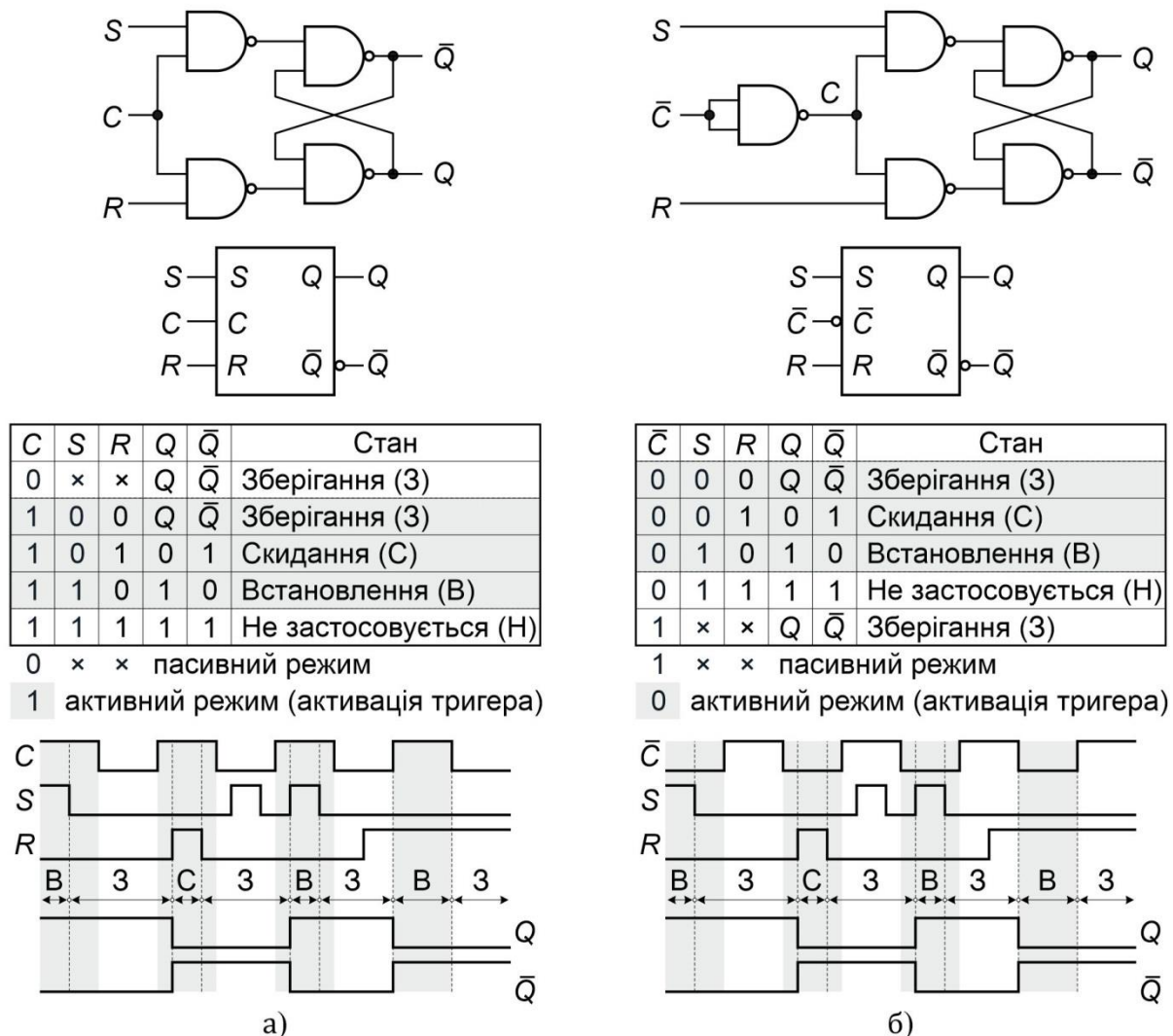


Рис. 7.3. Логічні та символічні схеми, таблиці істинності та часові діаграми синхронних SR-тригерів на логічних елементах І-НЕ (NAND), де а) активація тригера тактовим сигналом ВИСОКОГО (HIGH) рівня, б) активація тригера тактовим сигналом НИЗЬКОГО (LOW) рівня

Тактовий вхід C синхронного тригера, активація якого відбувається при ВИСОКОМУ (HIGH) рівні синхронізуючого тактового сигналу (Рис. 7.3 а), називають *потенціальним прямим* входом. Активація синхронного тригера може відбуватися по іншій схемі, при НИЗЬКОМУ (LOW) рівні синхронізуючого тактового сигналу $\bar{C}$ (Рис. 7.3 б). Тактовий вхід такого тригера називають *потенціальним інверсним* входом, на що вказує символ інверсії (кружок) на схемі тригера та символ інверсії на позначенні входу $\bar{C}$.

5°. Більш гнучкими схемами керування синхронними тригерами вважаються схеми активації тригера *перепадом* тактових сигналів (активація *фронтом* тактового імпульсу), для чого в схему синхронного тригера (Рис. 7.3) дода-

ється *елемент затримки* – генератор імпульсів (Рис. 7.4). Логічна схема генератора, на виході якого генерується імпульс в момент *наростання* тактового сигналу на вході (*позитивний перепад* тактового сигналу), показана на Рис. 7.4 а). Логічна схема генератора, на виході якого генерується імпульс в момент *спадання* тактового сигналу (*негативний перепад*), показана на Рис. 7.4 б). Тривалість генерованого імпульсу визначається часом затримки проходження тактового сигналу через інвертор і складає одиниці наносекунд.

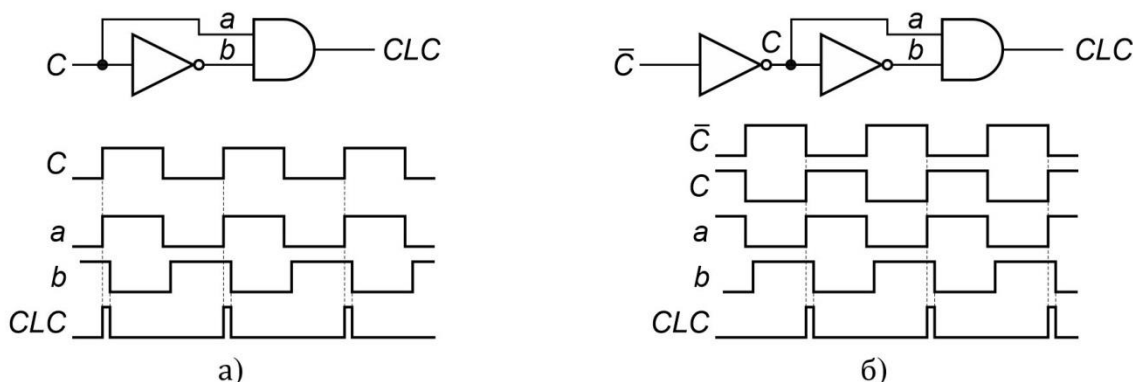


Рис. 7.4. Логічні схеми, таблиці істинності та часові діаграми генераторів імпульсів, де а) тактові імпульси генеруються при наростанні тактового сигналу на вході (позитивний перепад тактового сигналу), б) тактові імпульси генеруються при спаданні тактового сигналу на вході (негативний перепад тактового сигналу)

Логічна та символічна схеми, таблиця істинності та часові діаграми синхронного *SR*-тригера на логічних елементах І-НЕ (NAND), що активується *позитивним перепадом* тактових сигналів (*наростаючим фронтом* тактового імпульсу) показана на Рис. 7.5.

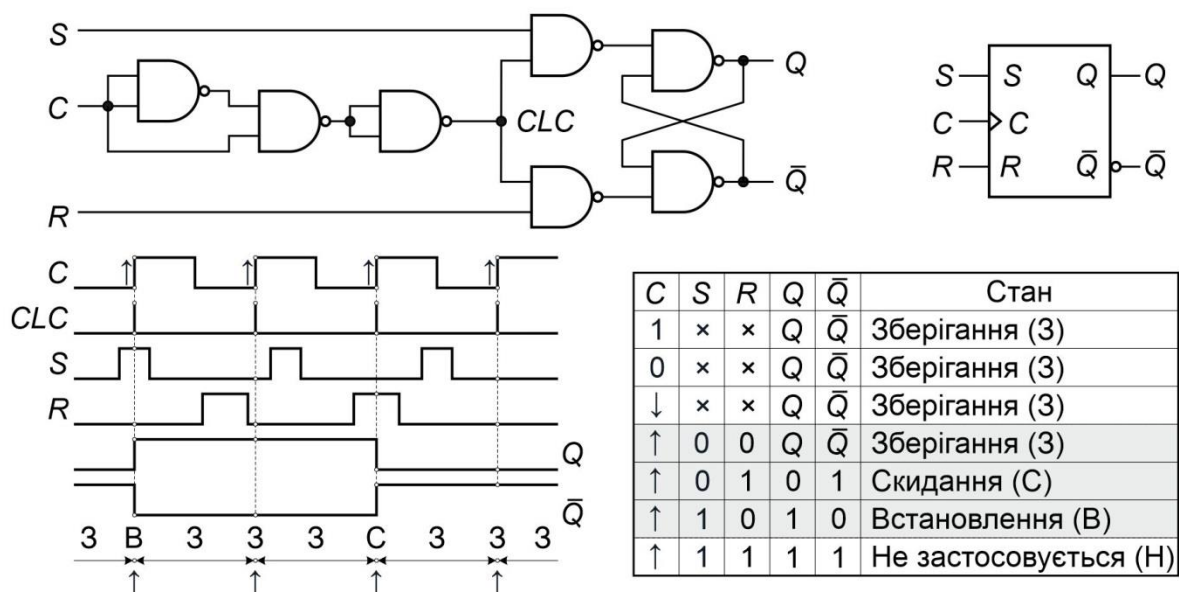


Рис. 7.5. Логічна та символічна схеми, часові діаграми та таблиця істинності синхронного *SR*-тригера на елементах І-НЕ (NAND), що активується позитивним перепадом тактових сигналів (наростаючим фронтом тактового імпульсу)

Логічна та символна схеми, таблиця істинності та часові діаграми синхронного SR-тригера на логічних елементах І-НЕ (NAND), що активується *негативним перепадом* тактових сигналів (*спадаючим фронтом* тактового імпульсу) показана на Рис. 7.6.

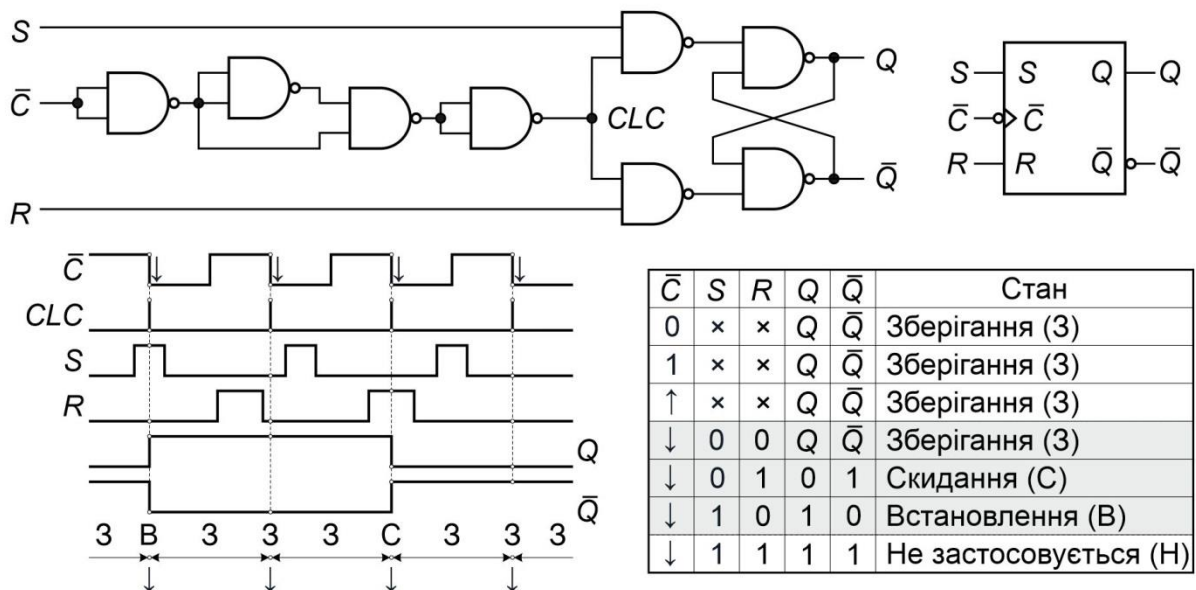


Рис. 7.6. Логічна та символна схеми, часові діаграми та таблиця істинності синхронного SR-тригера на елементах І-НЕ (NAND), що активується негативним перепадом тактових сигналів (спадаючим фронтом тактового імпульсу)

6°. Ще одна схема керування тригером, яка широко застосовується в цифровій схемотехніці, – схема активації тригера *тактовим імпульсом*. Активація тригера відбувається певним рівнем вхідного синхронізуючого тактового сигналу – ВИСОКИМ (HIGH) або НИЗЬКИМ (LOW). Проте, для зміни рівня сигналу на виході тригера необхідно, щоб відбулося як наростання так і падіння тактового сигналу при активації ВИСОКИМ (HIGH) рівнем сигналу або, відповідно, падіння і наростання тактового сигналу при активації НИЗЬКИМ (LOW) рівнем сигналу. Логічна схема такого тригера (Рис. 7.7) будується на основі двох стандартних схем синхронного тригера (7.1.1.2°, Рис. 7.3), одна з яких утворює умовно керуючий тригер, а інша – умовно керований (master-slave flip-flop).

Входи S та R керуючого тригера називають *синхронними входами*. Залежно від рівнів сигналу на цих входах, рівні сигналів на виході керуючого тригера змінюються синхронно з тактовим сигналом C на вході. Керований тригер має додаткові входи встановлення $\overline{PRE}$ та скидання $\overline{CLR}$ (Рис. 7.7), які називають *асинхронними входами*. На відміну від синхронних входів встановлення S та скидання R , асинхронні входи встановлення $\overline{PRE}$ та скидання $\overline{CLR}$, залежно від їх рівнів сигналу, змінюють рівні сигналів на виході тригера незалежно від синхронізуючого входу тактових сигналів C , тобто, асинхронно. Входи $\overline{PRE}$ та $\overline{CLR}$ є інверсними, на що вказує, як прийнято, кружок на символній схемі.

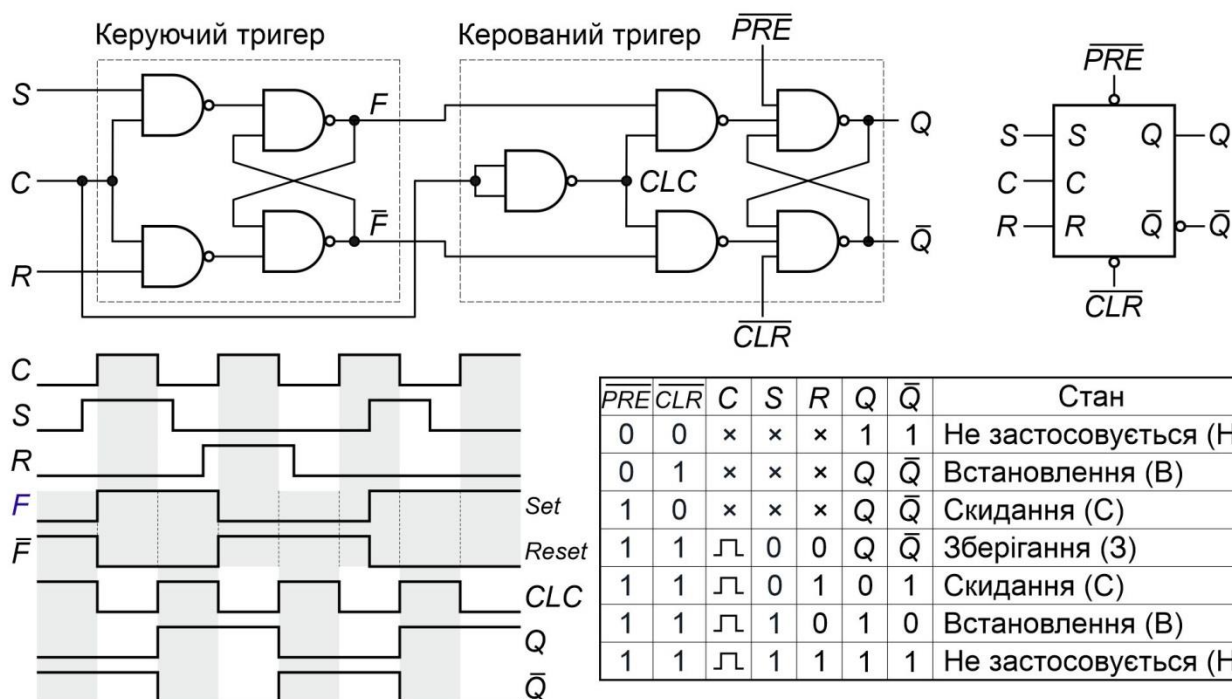


Рис. 7.7. Логічна та символна схеми, часові діаграми та таблиця істинності синхронного SR-тригера на елементах І-НЕ (NAND), що активується тактовим імпульсом, – тригер по схемі керуючий-керований (master-slave flip-flop)

7°. Модифікацією SR-тригера з одним входом є D-тригер, в якого входи S та R узгоджені через інвертор (Рис. 7.8). Входу S SR-тригера відповідає вхід D D-тригера, а входу R SR-тригера відповідає вхід $\overline{D}$ D-тригера. У D-тригера неможлива комбінація, за якою рівні станів вхідних сигналів однакові – $D \neq \overline{D}$, на відміну від SR-тригера, для якого при певній послідовності комбінацій вхідних станів, починаючи з комбінації $S=R=1$ або $\overline{S}=\overline{R}=0$, на виході виникає нестійкий, отже, невизначений, стан (7.1.1.2°, Рис. 7.1 є, ж). Отже, у D-тригерів відсутній недолік можливої невизначеності вихідних станів, притаманний SR-триграм.

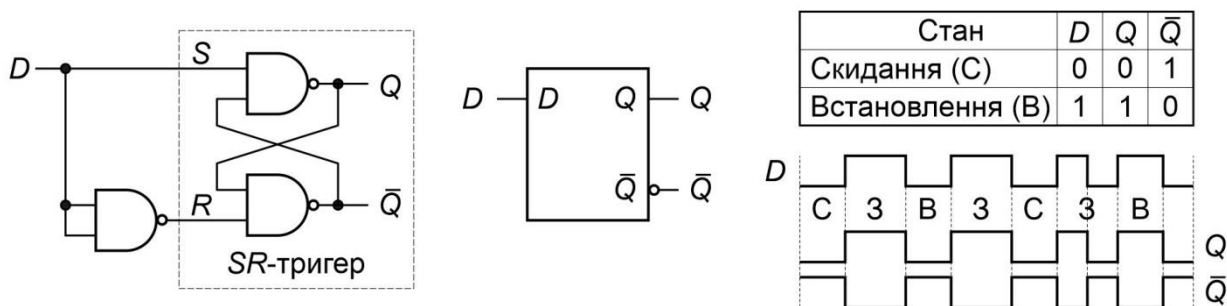
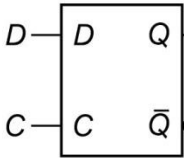
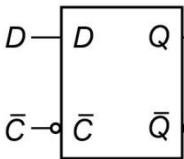
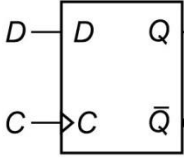
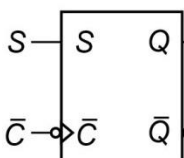


Рис. 7.8. Логічна та символна схеми, таблиця істинності та часові діаграми D-тригера на логічних елементах І-НЕ (NAND)

8°. Логічні схеми синхронних D-тригерів, стан на виходах яких змінюється синхронно з тактовими сигналами C на вході в режимі активації тригера, будуються аналогічно побудові схеми асинхронного D-тригера (Рис. 7.8) – в

стандартні схеми синхронних SR -тригерів (7.1.1.5°÷7.1.1.7°, Рис. 7.3÷Рис. 7.7) вводиться інвертор для узгодження входу D D -тригера з входами S та R SR -тригера. Тому далі показані лише символічні схеми синхронних D -тригерів, а їх логічні схеми не розглядаються. Деякі символічні схеми та таблиці істинності синхронних D -тригерів зведені в таблиці (Рис. 7.9).

АКТИВАЦІЯ СИНХРОННОГО D -ТРИГЕРА РІВНЕМ ТАКОВОГО СИГНАЛУ C						
Потенціальний прямий вхід (позитивний рівень активації)		C	D	Q	$\bar{Q}$	Стан
		0	x	Q	$\bar{Q}$	Зберігання
		1	0	0	1	Скидання
		1	1	1	0	Встановлення
Потенціальний інверсний вхід (негативний рівень активації)		$\bar{C}$	D	Q	$\bar{Q}$	Стан
		0	0	0	1	Скидання
		0	1	1	0	Встановлення
		1	x	Q	$\bar{Q}$	Зберігання

АКТИВАЦІЯ СИНХРОННОГО D -ТРИГЕРА ПЕРЕПАДАМИ ТАКОВОГО СИГНАЛУ C						
Позитивний перепад сигналу (активація наростаючим фронтом)		C	D	Q	$\bar{Q}$	Стан
		0	x	Q	$\bar{Q}$	Зберігання
		1	x	Q	$\bar{Q}$	Зберігання
		↓	x	Q	$\bar{Q}$	Зберігання
		↑	0	0	1	Скидання
		↑	1	1	0	Встановлення
Негативний перепад сигналу (активація спадаючим фронтом)		$\bar{C}$	D	Q	$\bar{Q}$	Стан
		0	x	Q	$\bar{Q}$	Зберігання
		1	x	Q	$\bar{Q}$	Зберігання
		↑	x	Q	$\bar{Q}$	Зберігання
		↓	0	0	1	Скидання
		↓	1	1	0	Встановлення

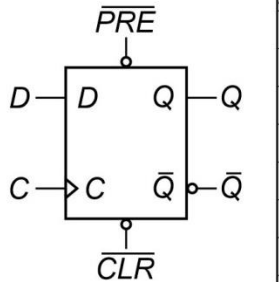
Позитивний перепад сигналу синхронізації (активація наростаючим фронтом) Додаткові вхідні сигнали асинхронного скидання та встановлення		$\overline{PRE}$	$\overline{CLR}$	C	D	Q	$\bar{Q}$	Стан
		0	0	x	x	1	1	Не застосовується
		0	1	x	x	1	0	Ас. встановлення
		1	0	x	x	0	1	Ас. скидання
		1	1	0	x	Q	$\bar{Q}$	Зберігання
		1	1	1	x	Q	$\bar{Q}$	Зберігання
		1	1	↓	x	Q	$\bar{Q}$	Зберігання
		1	1	↑	0	0	1	Скидання
		1	1	↑	1	1	0	Встановлення

Рис. 7.9. Символьні схеми, таблиці істинності та інтегральні мікросхеми синхронних D -тригерів

Асинхронні входи будь-яких синхронних тригерів мають небажану комбінацію значень $\overline{PRE} = \overline{CLR} = 0$, яка практично не застосовується. Проте, асинхронні входи часто застосовуються на практиці для повного скидання значень лічильників (7.2.1.1°) та регістрів (7.3.1.1°), які зібрані з масиву тригерів.

9°. Модифікацією SR -тригера є також JK -тригер, входи якого J і K відповідають входам S і R SR -тригера з тією різницею, що логічна схема JK -тригера містить додаткові ланцюги перехресного зворотного зв'язку між входами і виходами (Рис. 7.10). Завдяки такій схемі при комбінації вхідних значень $J = K = 1$ відбувається *переключення* (інверсія) станів на виходах JK -тригера, на відміну від SR -тригера, в якого при комбінації вхідних значень $S = R = 1$ на виході тригера в подальшому виникає невизначений стан (7.1.1.2°, Рис. 7.1 є, ж). Така модифікована схема, синхронного JK -тригера, не може бути активована рівними вхідних сигналів, а лише їх перепадами або тактовим імпульсом. Логічна та символна схеми, часові діаграми та таблиця істинності синхронного JK -тригера на елементах І-НЕ (NAND), що активується позитивним перепадом (наростаючим фронтом) тактового імпульсу, представлені на Рис. 7.10.

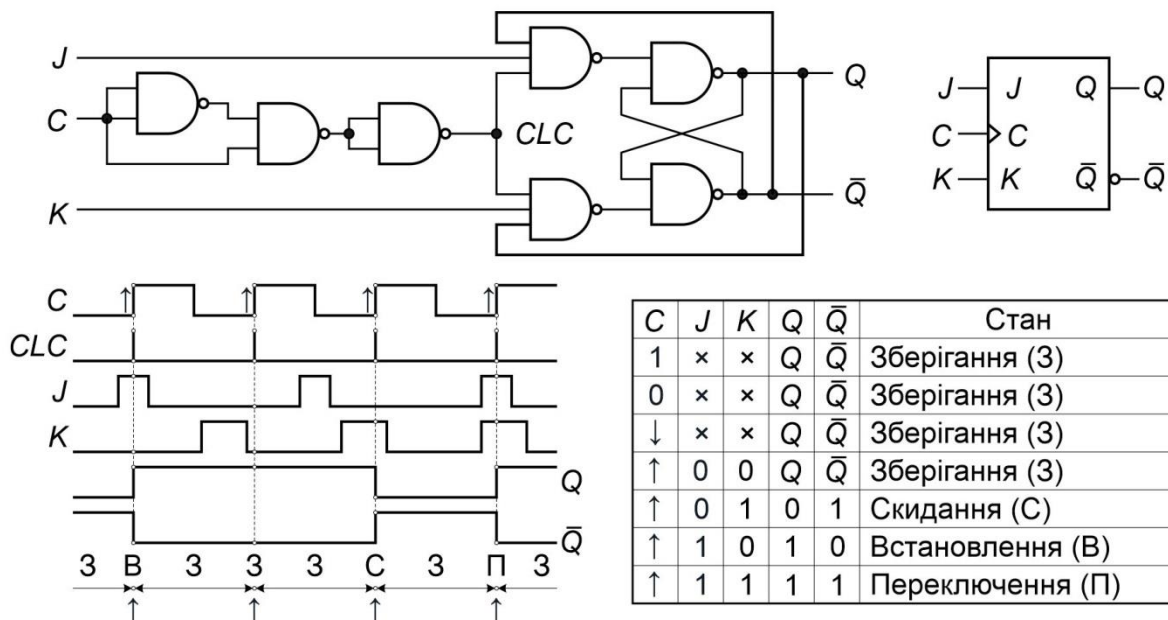


Рис. 7.10. Логічна та символна схеми, часові діаграми та таблиця істинності синхронного JK -тригера на елементах І-НЕ (NAND), що активується позитивним перепадом тактових сигналів (наростаючим фронтом тактового імпульсу)

Символьна схема та таблиця істинності синхронного JK -тригера, що активується негативним перепадом (спадаючим фронтом) тактового імпульсу, представлені на Рис. 7.11.

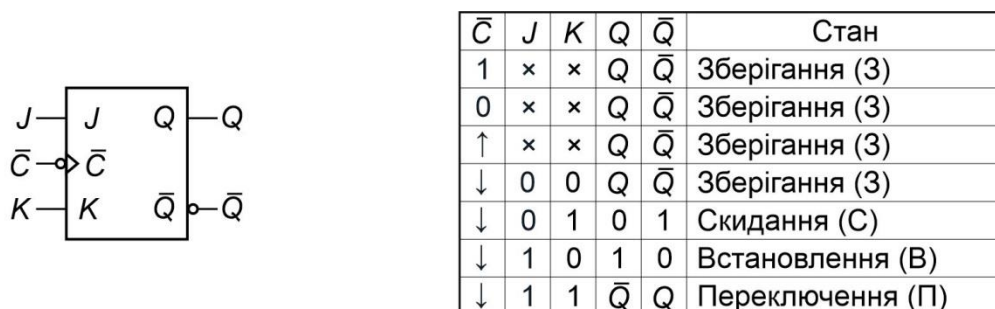


Рис. 7.11. JK -тригер, що активується спадаючим фронтом тактового імпульсу

10°. В цифровій схемотехніці синхронні *JK*-тригери, що активуються тактовим імпульсом (7.1.1.6°) застосовуються рідко. Широке застосування знайшли *JK*-тригери з синхронними та асинхронними входами – синхронні *JK*-тригери, що активуються перепадами (фронтами) тактового імпульсу (7.1.1.9°) та мають додаткові асинхронні входи встановлення та скидання вихідних станів тригера (Рис. 7.12).

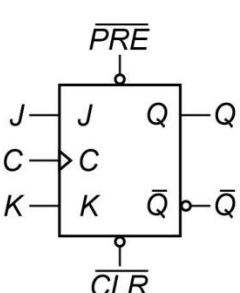
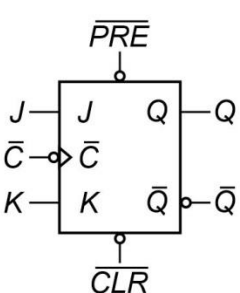
JK-ТРИГЕРИ З СИНХРОННИМИ ТА АСИНХРОННИМИ ВХОДАМИ							
Синхронна активація позитивним перепадом тактового сигналу (наростаючим фронтом тактового імпульсу) 	$\overline{PRE}$	$\overline{CLR}$	C	J	K	Q	$\bar{Q}$
	0	0	x	x	x	1	1
	0	1	x	x	x	1	0
	1	0	x	x	x	0	1
	1	1	0	x	x	Q	$\bar{Q}$
	1	1	1	x	x	Q	$\bar{Q}$
	1	1	↓	x	x	Q	$\bar{Q}$
	1	1	↑	0	0	Q	$\bar{Q}$
	1	1	↑	0	1	1	0
	1	1	↑	1	0	0	1
Синхронна активація негативним перепадом тактового сигналу (спадаючим фронтом тактового імпульсу) 	$\overline{PRE}$	$\overline{CLR}$	$\bar{C}$	J	K	Q	$\bar{Q}$
	0	0	x	x	x	1	1
	0	1	x	x	x	1	0
	1	0	x	x	x	0	1
	1	1	0	x	x	Q	$\bar{Q}$
	1	1	1	x	x	Q	$\bar{Q}$
	1	1	↑	x	x	Q	$\bar{Q}$
	1	1	↓	0	0	Q	$\bar{Q}$
	1	1	↓	0	1	1	0
	1	1	↓	1	0	0	1
Стан							
Не застосовується							
Ас. встановлення							
Ас. скидання							
Зберігання							
Зберігання							
Зберігання							
Зберігання							
Скидання							
Встановлення							
Переключення							

Рис. 7.12. *JK*-тригери з синхронними та асинхронними входами, що активуються в синхронному режимі перепадами (фронтами) тактового імпульсу

Поширені застосування *JK*-тригерів з синхронними та асинхронними входами – в схемах лічильників і зсувних регістрів (див. наступні підрозділи).

11°. Тригери різних типів реалізовані сімействами інтегральних мікросхем. Від найпростішого *SR*-тригера, реалізованого інтегральною мікросхемою 74LS279 (7.1.1.3° Рис. 7.2) до *JK*-тригерів. Цифрові пристрої, схеми яких будуються на основі тригерів – лічильники та регістри, також реалізовані спеціалізованими інтегральними мікросхемами. Цей чинник та розвиток мікропроцесорних технологій привели до припинення промислового виготовлення та практичного вжитку мікросхем за деякими серійними номерами. Наприклад, фірма Texas Instruments припинила випуск інтегральних мікросхем *JK*-тригерів серії

7400 за технологією ТТЛ за такими серійними номерами 72, 70, 78, 110, 75, 111, 113, 114, 276, 376, підтримуючи випуск сімейств *JK*-тригерів за різними технологіями за такими серійними номерами 73, 107, 109, 112 (Рис. 7.12).

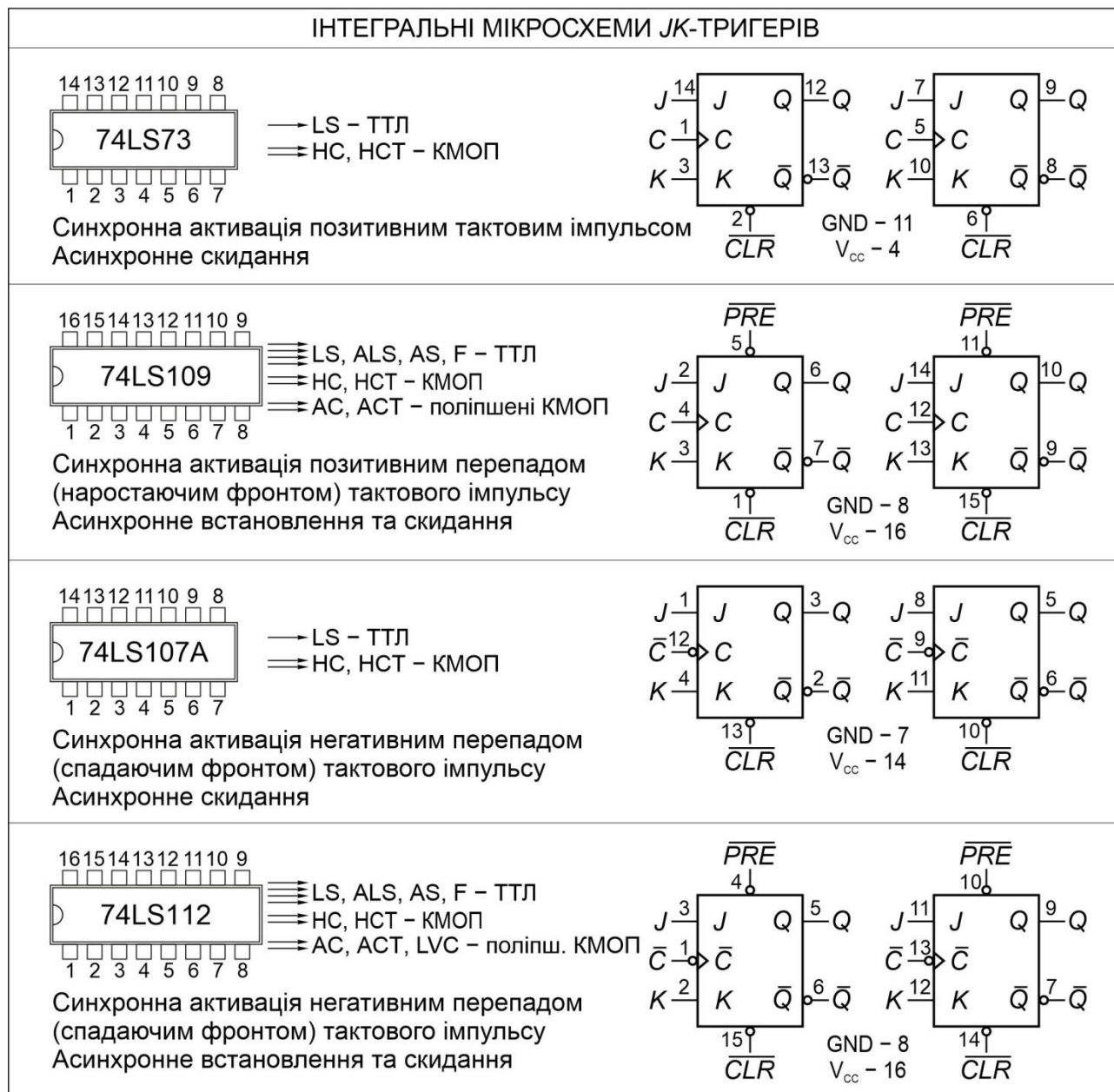


Рис. 7.13. *JK*-тригери з синхронними та асинхронними входами, що активуються в синхронному режимі перепадами (фронтами) тактового імпульсу

При застосуванні мікросхем *JK*-тригерів за номерами 74LS73 та 74LS107 необхідно звернути увагу на наступне. Ці мікросхеми функціонально ідентичні, в синхронному режимі активуються позитивним тактовим імпульсом та мають один асинхронний вхід скидання. Особливість мікросхеми 74LS73 – в нестандартних призначеннях її контактів живлення (Рис. 7.13). Мікросхема 74LS107A (з суфіксом А) в синхронному режимі функціонує іншим способом – активується негативним перепадом (спадаючим фронтом) тактового імпульсу (Рис. 7.13).

7.1.2. Лічильники

1°. Складні цифрові схеми містять *лічильники* – цифрові пристрої, призначені для підрахування числа певних подій або часових інтервалів, чи впорядкованих подій в хронологічному порядку. Лічильники широко застосовуються також як *подільники частоти*. В деяких випадках лічильники застосовуються для виконання непрямих функцій, як елементи пам'яті чи елементи адресації. Розрізняють *двійкові* та *десяткові* лічильники. Схеми лічильників можна зібрати на дискретних елементах – тригерах. Проте, промислово реалізовані серії інтегральних мікросхем різноманітних лічильників та подільників частоти, *синхронних* та *асинхронних* за принципом активації.

Як приклад, схеми лічильника по модулю 16 та, одночасно, подільника частоти можна реалізувати на дискретних елементах – на чотирьох елементах *JK*-тригерів. Схема такого *асинхронного* лічильника та подільника частоти при реалізації на дискретних елементах *JK*-тригерів показана на Рис. 7.14, а при реалізації на інтегральній мікросхемі 74LS93 – на Рис. 7.15. Такий лічильник по модулю 16 та подільник частоти ще називають *двійковим чотирьох розрядним лічильником* (4 bit binary counter) або *подільником частоти на 2, 4, 8, 16*.

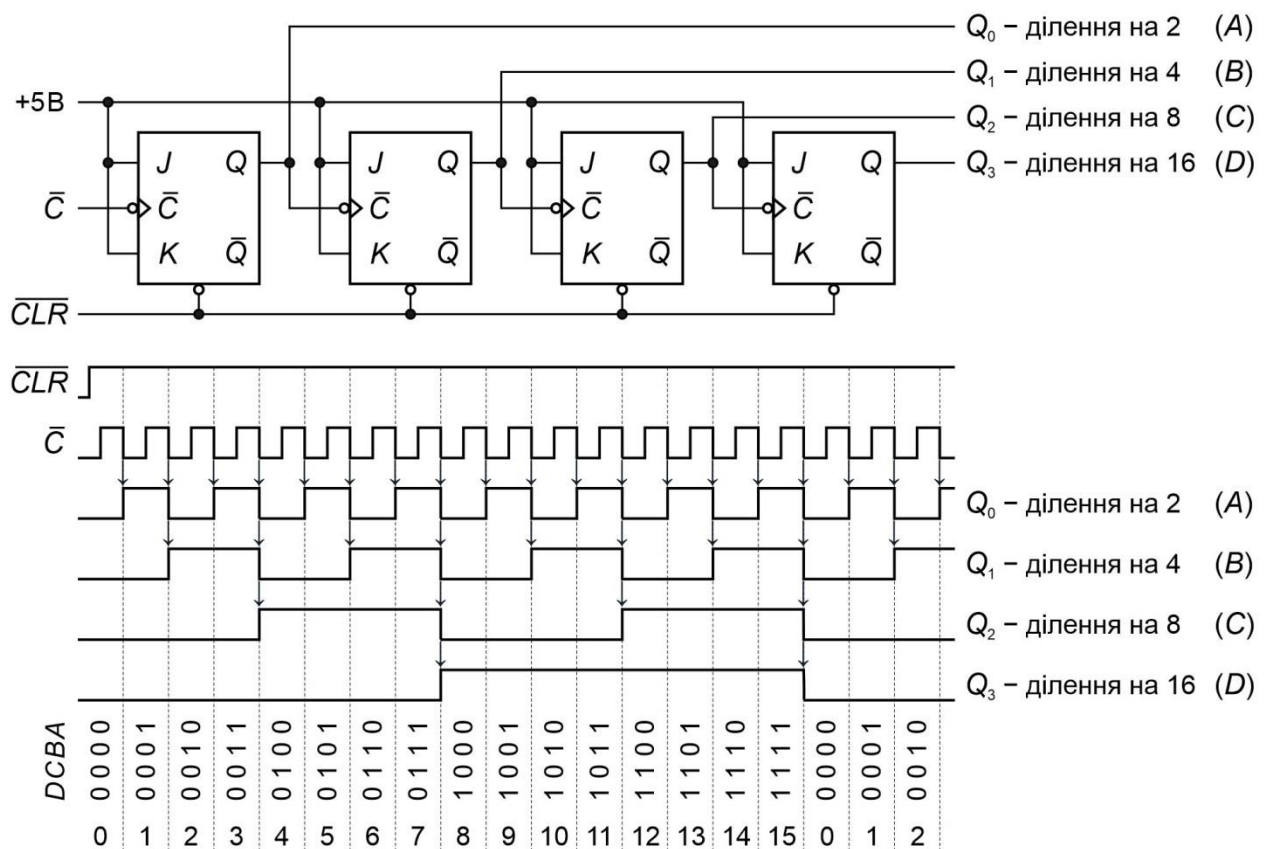


Рис. 7.14. Структурна схема та часові діаграми асинхронного двійкового чотирьох розрядного лічильника та подільника частоти на 2, 4, 8, 16 на дискретних елементах *JK*-тригерів

2°. Для того, щоб схема асинхронного лічильника та подільника частоти, яка реалізується на інтегральній мікросхемі 74LS93, була еквівалентна схемі, яка реалізована на дискретних *JK*-тригерах (Рис. 7.14), необхідно з'єднати вихід Q_A (контакт 12) з входом C_2 (контакт 1) мікросхеми 74LS93 (Рис. 7.15). Окрім цього, асинхронний вхідний сигнал $\overline{CLR}$ скидання вихідних станів лічильника для на інтегральній мікросхемі 74LS93 повинен мати високий рівень активації, для чого в схему необхідно ввести додатково інвертор (Рис. 7.15).

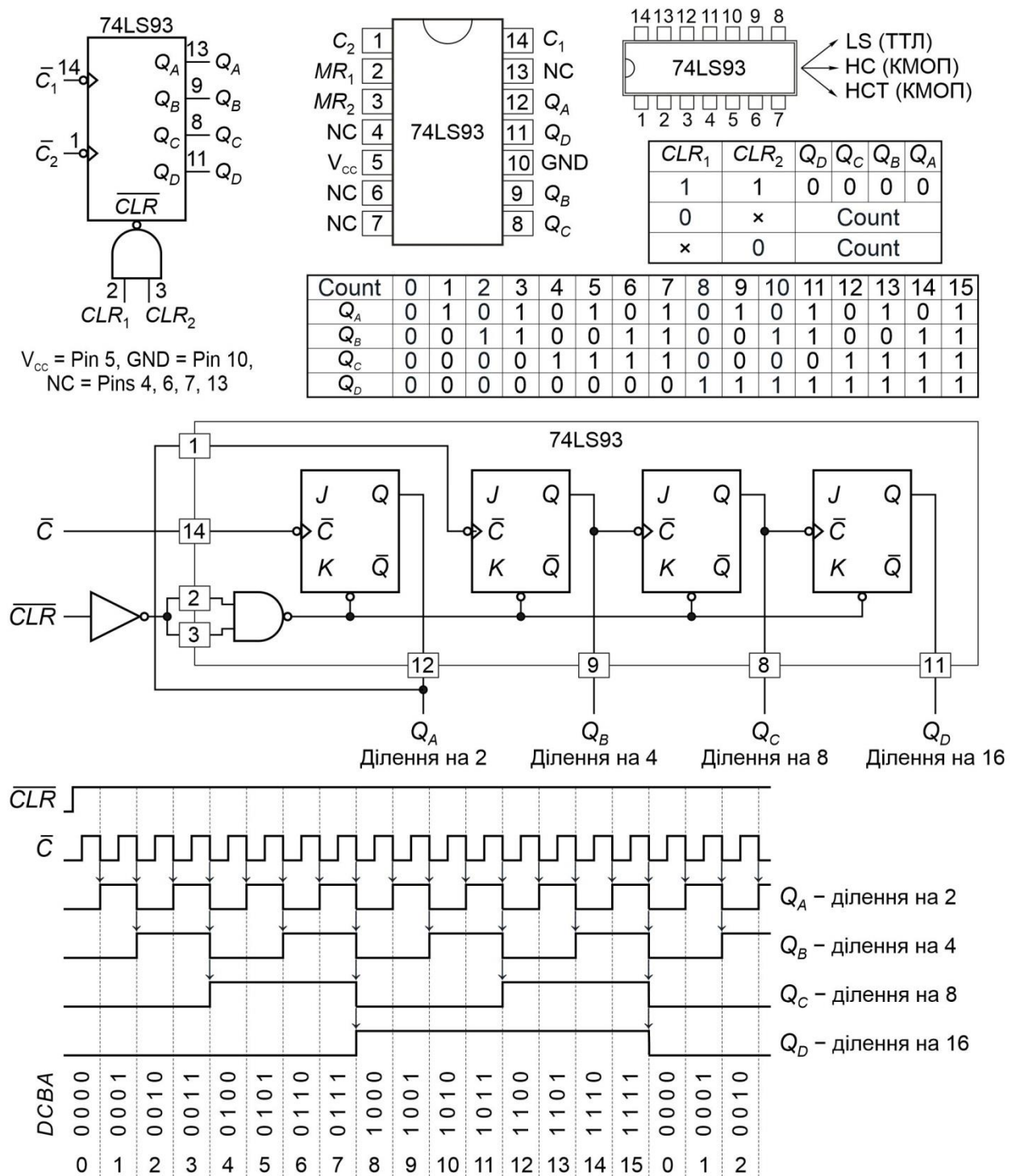


Рис. 7.15. Асинхронний двійковий чотирьох розрядний лічильник та подільник частоти, реалізований інтегральною мікросхемою 74LS93

3°. В асинхронних лічильниках та подільниках частоти зміна вихідних станів не відбувається одночасно, тобто, асинхронно, через затримку сигналів при їх проходженні через елементи схеми. В високочастотних цифрових пристроях застосовуються лише синхронні лічильники та подільники частоти. Розглянутий вище двійковий чотирьох розрядний лічильник або подільник частоти на 2, 4, 8, 16, що працює в синхронному режимі, також можна реалізувати на дискретних елементах – на чотирьох елементах *JK*-тригерів та двох універсальних базових логічних елементах І (AND). Схема такого *синхронного* лічильника та подільника частоти показана на Рис. 7.16.

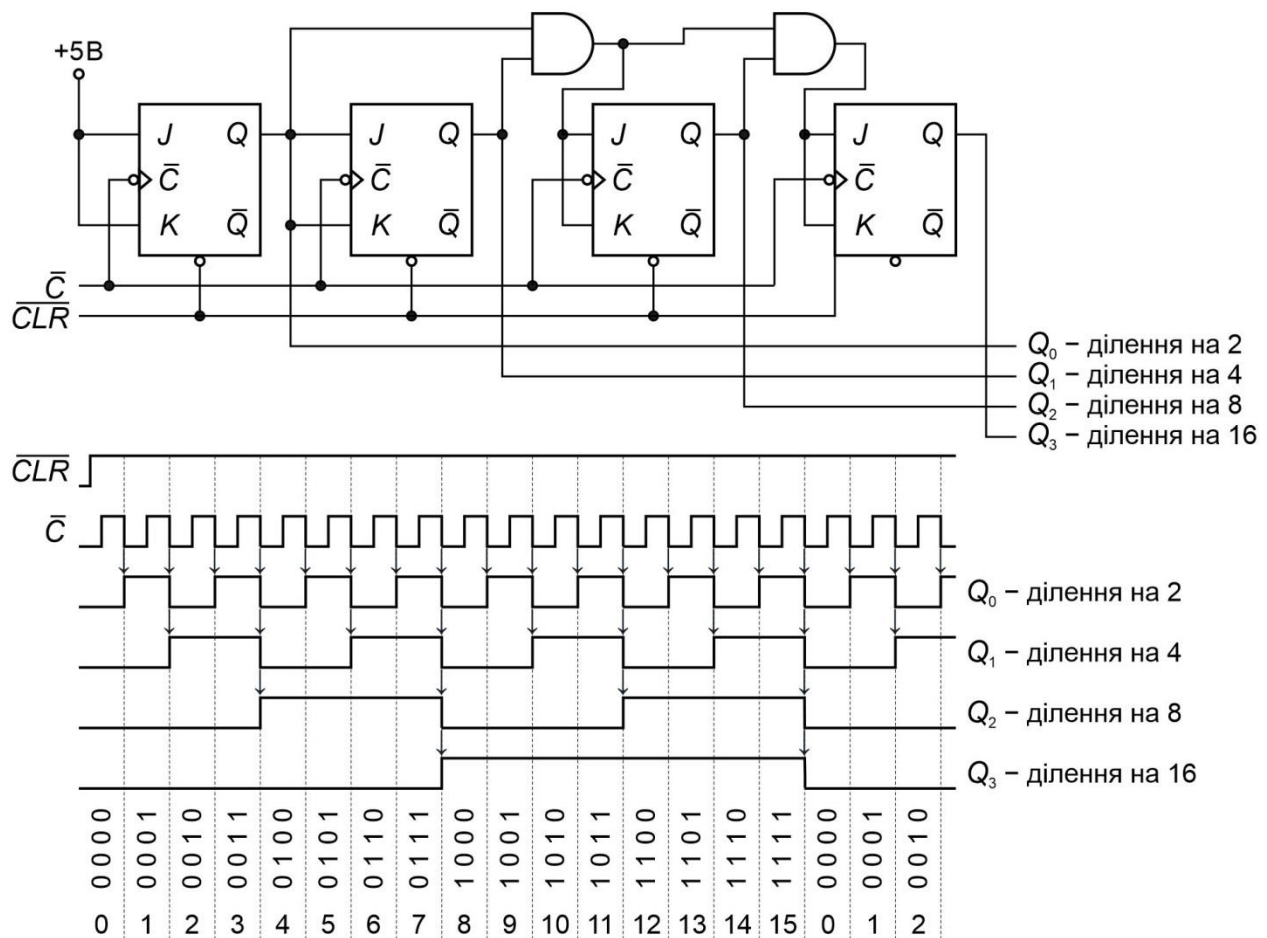


Рис. 7.16. Структурна схема та часові діаграми синхронного двійкового чотирьох розрядного лічильника та подільника частоти на 2, 4, 8, 16 на дискретних елементах *JK*-тригерів та універсальних базових логічних елементах І (AND)

Синхронний двійковий чотирьох розрядний лічильник або подільник частоти на 2, 4, 8, 16 також можна реалізувати інтегральною мікросхемою 74LS193. Інтегральна мікросхема 74LS193 є багатофункціональною. Рахунок можна почати в прямому або в зворотному напрямку, для чого мікросхема містить два синхронізуючих входи тактових сигналів прямого і зворотного рахунку. Рахунок можна почати з будь-якого числа, для чого мікросхема містить чотири інформаційних входи паралельного завантаження стартового числа для відліку.

7.1.3. Регістри

1°. В цифровій схемотехніці для тимчасового зберігання послідовності цифрових даних, що переміщуються, застосовуються послідовнісні цифрові пристрої – *реєстри*. Простий приклад застосування реєстрів показаний на Рис. 7.17 (7.1.3.2°). *Зсувні реєстри* застосовуються при необхідності зсуву побітно направо чи наліво переміщуваної послідовності цифрових даних. Зсувні реєстри можуть переміщувати дані як *паралельно*, так і *послідовно*, а також перетворювати послідовні дані в паралельні, та навпаки – паралельні в послідовні. Реєстри та зсувні реєстри можна реалізувати на дискретних елементах, тригерах, та на спеціалізованих інтегральних мікросхемах, інформацію про які можна знайти в спеціалізованій літературі, в тому числі рекомендованій, або в каталогах від виробника мікросхем.

2°. У наведеному в Практикумі 5 прикладі управління з цифрової клавіатури семисегментним світлодіодним індикатором з застосуванням комбінаційних цифрових пристроїв, шифраторів і дешифраторів (5.3.3.3°, Рис. 5.12), індикатор висвітлює десяткову цифру лише при утриманні натиснутою відповідної клавіші цифрової клавіатури. При відпусканні клавіші індикатор перестає висвітлювати цифру через те, що комбінаційні цифрові пристрої, шифратори і дешифратори, не володіють елементами пам'яті, на відміну від послідовнісних цифрових пристроїв, якими є регістри. При введенні в розглянуту схему (Рис. 5.12) регістра, світлодіодний індикатор висвітлюватиме цифру після відпускання відповідної, попередньо натиснутої, клавіші (Рис. 7.17).

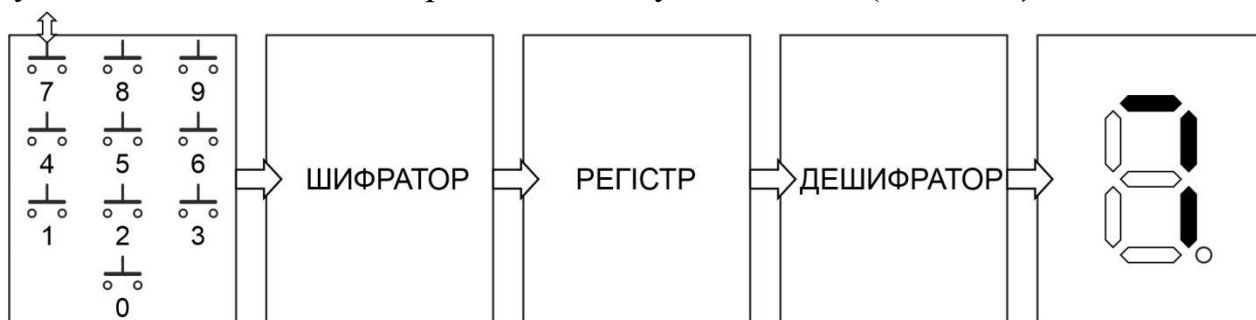


Рис. 7.17. Схема зі шифратором і дешифратором для управління з цифрової клавіатури семи сегментним світлодіодним індикатором з застосуванням регістра для запам'ятовування і подальшого висвітлювання цифри на індикаторі після відпускання відповідної, попередньо натиснутої, клавіші

При реалізації цифрового пристрою управління не одним семисегментним світлодіодним індикатором, а дисплеєм з матрицею таких індикаторів, застосовуються зсувні регістри. При цьому логічна схема ускладнюється через необхідність введення додаткових цифрових елементів для управління матрицею індикаторів дисплея в мультиплексному режимі (див. *драйвери*, 7.1.3.3°).

3°. Вище зазначалось, що будь-які цифрові пристрої можна реалізувати на основі дискретних елементів. Для комбінаційних цифрових пристроїв такими елементами є базові логічні елементи, а для послідовнісних – тригери, які в свою чергу також реалізовані на базових логічних елементах. Проте, що теж зазначалось вище, на практиці зазвичай застосовуються спеціалізовані інтегральні мікросхеми – шифратори та дешифратори, лічильники та регістри тощо. Також зазначалась тенденція широкого застосування цифрових пристроїв з програмним управлінням – мікроконтролерів та програмованих користувачем вентиляційних матриць (5.3.3.2°). Така тенденція викликала поступове припинення застосування спеціалізованих інтегральних мікросхем, в першу чергу комбінаційних цифрових пристроїв. Наприклад, мікроконтролери порівняно низької вартості успішно виконують функції суматорів та дешифраторів. При цьому необхідно враховувати, що практична реалізація цих функцій мікроконтролером передбачає виконання певних дії – складання відповідної програми для мікроконтролера.

Тут варто зазначити також тенденцію на широке застосування комбінованих інтегральних мікросхем – *драйверів*, призначених для керування певними цифровими або цифро-аналоговими пристроями. Логічні схеми драйверів складаються з комбінаційних та послідовнісних цифрових пристроїв. Таким чином, один спеціальний драйвер замінює собою кілька або низку спеціалізованих цифрових інтегральних мікросхем. Наприклад, при застосуванні спеціальних інтегральних мікросхем MM5450 (LED Display Drivers) і ICM7217 (4-Digit LED Display, Programmable Up/Down Counter) драйверів дисплеїв з семисегментними світлодіодними індикаторами відпадає потреба в спеціалізованих інтегральних мікросхемах шифраторів та дешифраторів, мультиплексорів та демультиплексорів, зсувних регістрів та лічильників (мікросхема ICM7217 окрім драйвера дисплея містить лічильник, рахунок якого, власне, може висвітлити дисплей). Враховуючи спеціальні призначення драйверів, інформацію про їх застосування, схеми підключення, режими роботи тощо варто отримувати безпосередньо з технічних паспортів (datasheets) від виробника інтегральних мікросхем драйверів.

7.2. ВПРАВИ І ЗАВДАННЯ

7.2.1. Вправи для виконання

1°. Самостійно, в пошуковому режимі, ознайомитись з технічним паспортом (datasheet) інтегральної мікросхеми послідовнісного цифрового пристрою – асинхронного чотирьох розрядного лічильника (варіанти). Проаналізувати логічну схему лічильника, визначити типи застосованих тригерів та способи їх активації. На основі символічної схеми мікросхеми побудувати практичну схему лічильника та навести приклад часової діаграми.

2°. Варіанти вправи зведені в таблиці.

Варіант	Інтегральна мікросхема лічильника
1.	74LS293
2.	74LS393

7.2.2. Контрольне завдання

1°. Виконати вправу 7.2.1.1° для одного варіанту. Варіант вибирається за непарним або парним номером по списку групи.

2°. По результатам виконання вправ оформити звіт. Рекомендується виконання вправи здійснювати з застосуванням симулятора електронних схем, наприклад, програмного середовища Micro-Cap 12, при наявності такої технічної можливості. При цьому, для захисту звіту підготувати відповідний демонстраційний файл, наприклад, в форматі CIR при застосуванні програмного середовища Micro-Cap 12.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Roger Tokheim, Patrick Hoppe. *Digital Electronics: Principles and Applications*. 9th edition in New York, USA: McGraw Hill LCC, 2022, 576 p.
2. Paul Scherz, Simon Monk. *Practical Electronics for Inventors*. 4th edition in New York, USA: McGraw Hill LCC, 2016.
3. Павловський О. М. Основи цифрової схемотехніки. Комп'ютерний практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерно-інтегровані системи та технології в приладобудуванні» спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології» / О. М. Павловський, І.О. Васильковська; КПІ ім. Ігоря Сікорського.– Київ : КПІ ім. Ігоря Сікорського, 2022. – 62 с.

**О. В. Заморський
О. М. Павловський**

ОСНОВИ ЦИФРОВОЇ СХЕМОТЕХНІКИ ПРАКТИКУМ

Реєстр. № НП 22/23-662 Обсяг 7,41 авт. арк.

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056

<https://kpi.ua>

Свідectво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.