

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

**О.М. Павловський  
І.О. Васильковська**

# **ОСНОВИ ЦИФРОВОЇ СХЕМОТЕХНІКИ КОМП'ЮТЕРНИЙ 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»  
спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології»

Електронне мережне навчальне видання

Київ  
КПІ ім. Ігоря Сікорського  
2022

Рецензенти: *Галаган Р.М.* - канд. техн. наук, доцент, доцент кафедри автоматизації та систем неруйнівного контролю КПП ім. Ігоря Сікорського  
*Головач С.В.* - канд. техн. наук, головний конструктор напрямку АТ «Елміз»

Відповідальний редактор: *Лакоза С.Л.* - канд. техн. наук, доцент, доцент кафедри комп'ютерно-інтегрованих оптичних та навігаційних систем КПП ім. Ігоря Сікорського

*Гриф надано Методичною радою КПП ім. Ігоря Сікорського  
(протокол № 1 від 02.09.2022 р.)  
за поданням Вченої ради Приладобудівного факультету  
(протокол № 9/22 від 29.08.2022 р.)*

Навчальний посібник містить теоретичні відомості та практичні рекомендації до виконання комп'ютерних практикумів з дисципліни «Основи цифрової схемотехніки».

Під час виконання комп'ютерних практикумів будуть розглянуті цифрові пристрої, що базуються на використанні логічних елементів, такі як комбінаційні схеми(шифратори та дешифратори, мультиплексори та демультимплексори, компаратори, суматори), та мікросхеми із пам'яттю(тригери та регістри) та інші пристрої побудовані на їх базі. Поступове ускладнення матеріалу призведе до більшої зацікавленості здобувачів та буде спонукати до подальшого, більш глибокого опанування основних моментів дисципліни.

Навчальний посібник призначений для здобувачів ступеня бакалавра за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології», буде також корисним студентам споріднених технічних спеціальностей.

Реєстр. № НП 22/23-057 Обсяг 2,77 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.М. Павловський, 2022  
© І.О. Васильковська, 2022  
© КПП ім. Ігоря Сікорського, 2022

## ЗМІСТ

|                                                                                    |    |
|------------------------------------------------------------------------------------|----|
| ВСТУП .....                                                                        | 4  |
| Комп'ютерний практикум № 1. Знайомство із базовими логічними елементами.....       | 5  |
| Комп'ютерний практикум № 2. Формування і перетворення логічних послідовностей..... | 17 |
| Комп'ютерний практикум № 3. Шифратори і дешифратори.....                           | 28 |
| Комп'ютерний практикум № 4. Мультиплексори та демюльтиплексори.....                | 36 |
| Комп'ютерний практикум №5. Компаратори та суматори.....                            | 45 |
| Комп'ютерний практикум №6. Тригери та регістри.....                                | 52 |
| Список рекомендованих джерел.....                                                  | 61 |

## ВСТУП

У сучасному світі використання електронних пристроїв перейшло на абсолютно новий рівень, за статистикою 60-70% всіх об'єктів із якими взаємодіє сучасний фахівець – електронні прилади. І з кожним роком такий відсоток буде збільшуватись. Не дивлячись на те, що людство фізично повністю не може позбутися аналогових електронних пристроїв, цифрові пристрої поступово замінюють їх у всіх сферах, науки, промисловості, бізнесу розваг, тощо.

Таким чином, метою даного посібника є ознайомлення із основними схемотехнічними рішеннями, що використовуються у сучасній цифровій електроніці. Починаючи від основних логічних елементів, поступово ускладнюючи і комбінуючи отримані схеми студент зможе більш глибоко опанувати логіку побудови таких пристроїв та усвідомити основні принципи їх вдосконалення, використання та розроблення.

У розділах посібника, які для простоти освоєння сформовані у вигляді окремих комп'ютерних практикумів, будуть розглянуті питання створення, використання, дослідження та вдосконалення основних схемотехнічних рішень цифрових електронних схем, таких як логічні вентиля, комбінаційні схеми, схеми з пам'яттю та інші.

У посібнику пропонується реалізовувати приклади та завдання у середовищі схемотехнічного моделювання MicroCAP, такий підхід дозволяє швидко та безпечно провести моделювання та дослідження запропонованих схем. Для більш глибокого засвоєння матеріалу здобувачам буде запропоновано повторити проведені дослідження та моделювання із реальними мікросхемами, використовуючи плати для безпечного монтажу, що дозволить наочно зв'язати теорію і практичну складову освітнього компоненту.

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 1

## ЗНАЙОМСТВО ІЗ БАЗОВИМИ ЛОГІЧНИМИ ЕЛЕМЕНТАМИ

**Мета роботи:** дослідити моделі базових логічних елементів при дії різних логічних рівнів на їх входах.

### 1.1. Теоретичні відомості

У Булевій алгебрі, на якій базується вся цифрова техніка, електронні елементи повинні виконувати ряд певних дій. Це так званий логічний базис. Ось три основні дії:

АБО (OR) - логічне додавання (диз'юнкція);

І (AND)- логічне множення (кон'юнкція);

НЕ (NOT) - логічне заперечення (інверсія, інвертування).

У цифровій електроніці такі дії реалізуються за допомогою відповідних базових логічних елементів, які зараз розглянемо детально. Найпростіший логічний елемент - це інвертор (логічний елемент НЕ, inverter). Інвертор виконує найпростішу логічну функцію - інвертування, тобто зміну рівня вхідного сигналу на протилежний. Інвертор має всього один вхід і один вихід. Умовне графічне позначення інвертора на схемах показано на рис. 1.1.

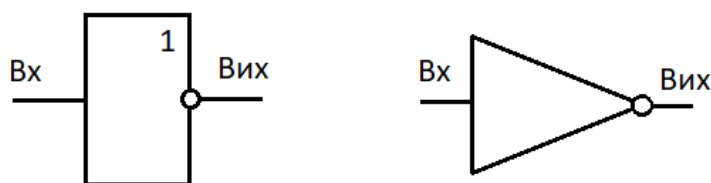


Рис. 1.1. Графічне позначення інвертора на електричних принципових схемах

Принцип дії логічних елементів зазвичай описується або часовими діаграмами, або таблицею істинності. Таблиця істинності для інвертора виглядає наступним чином (Табл. 1.1).

Дві основні області застосування інверторів - це зміна полярності сигналу і зміна полярності фронту сигналу. Тобто з позитивного вхідного

сигналу інвертор робить негативний вихідний сигнал і навпаки, а з позитивного фронту вхідного сигналу - негативний фронт вихідного сигналу.

Таблиця 1.1 – Таблиця істинності інвертора

| Вхід | Вихід |
|------|-------|
| 0    | 1     |
| 1    | 0     |

Ще одне важливе застосування інвертора - буферизація сигналу (з інверсією), тобто збільшення навантажувальної здатності сигналу. Це буває потрібно в тому випадку, коли якийсь сигнал треба подати на багато входів, а вихідний струм джерела сигналу недостатній.

Тепер розглянемо логічні елементи, що реалізують функції логічного множення та логічного додавання. Відтак, на рисунку 1.2 а показано умовне графічне зображення логічного елементу І, а на рис. 1.2. б – логічного елементу АБО.

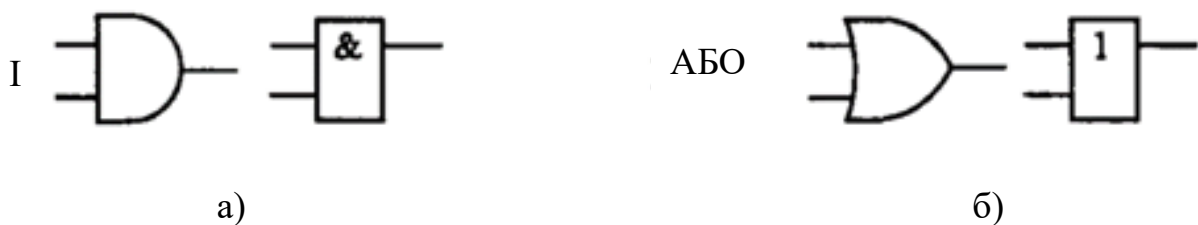


Рис. 1.2. Графічне позначення логічних елементів: а) Логічне І; Логічне АБО

Дуже часто зазначені логічні елементи комплексуються із інвертором, таким чином утворюючи логічні елементи із інвертованим вихідним сигналом. Ці елементи показані на рис. 1.3.

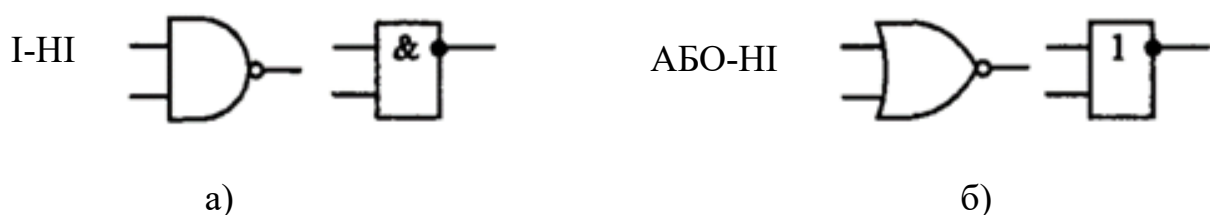


Рис. 1.3. Графічне позначення логічних елементів: а) Логічне І-НІ; б) Логічне АБО-НІ

Таблиця істинності для таких елементів буде виглядати наступним чином (Табл. 1.2.):

Таблиця 1.2. Таблиця істинності логічних елементів АБО, І та їх інверсних похідних

| Вх.1 | Вх.2 | АБО | АБО-НІ | І | І-НІ |
|------|------|-----|--------|---|------|
| 0    | 0    | 0   | 1      | 0 | 1    |
| 0    | 1    | 1   | 0      | 0 | 1    |
| 1    | 0    | 1   | 0      | 0 | 1    |
| 1    | 1    | 1   | 0      | 1 | 0    |

Треба зазначити, що входи у таких елементів взаємозамінні і абсолютно рівноправні.

Комбінуючи базові логічні елементи можна утворити схеми з новими властивостями. А тому у цьому комп'ютерному практикумі необхідно розглянути ще два елементи, які хоча не є базовими проте їх використання також набуло широкого застосування. Це виключне АБО та виключне АБО-НІ (XOR та XNOR). Утворення таких елементів як комбінації базових елементів розглянемо у наступному комп'ютерному практикумі.

Позначення та таблиця істинності приведені на рис. 1.4. та табл. 1.3. відповідно.

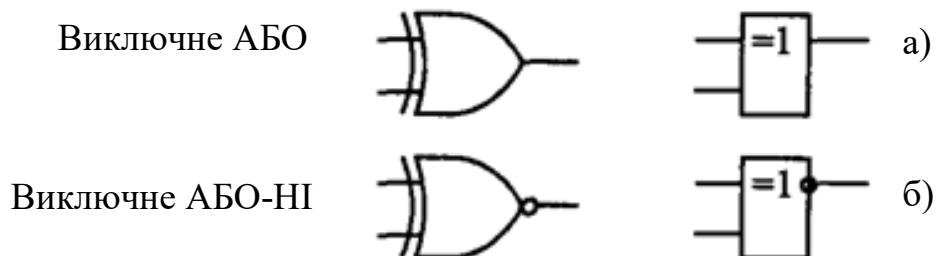


Рис. 1.4. Графічне позначення логічних елементів: а) Виключне АБО;  
б) Виключне АБО -НІ

Таблиця 1.3. Таблиця істинності логічних елементів Виключне АБО та Виключне АБО -НІ

| Вх.1 | Вх.2 | Виключне АБО | Виключне АБО -НІ |
|------|------|--------------|------------------|
| 0    | 0    | 0            | 1                |
| 0    | 1    | 1            | 0                |
| 1    | 0    | 1            | 0                |
| 1    | 1    | 0            | 1                |

Таким чином, логічна одиниця на виході виключного АБО з’являється лише тоді, коли активний сигнал (лог.1) присутній на одному і лише на одному із входів.

Мікросхеми базових логічних елементів дуже часто випускаються у корпусах формфактору DIP-14 або SO-14. DIP (або DIL dual in-line package), прямокутні корпуси, що зазвичай виготовляються із кремнію або пластику та мають два ряди виводів (ніжок), розташованих паралельно по довшим сторонам корпуса. Корпуси SO або SOIC (Small-Outline Integrated Circuit) призначені для поверхневого монтажу, що значно скорочує площу, яку займає мікросхема на друкованій платі у порівнянні із мікросхемами формфактору DIP, проте потребує значно більших навичок і додаткового обладнання для її використання. Тому в даних лабораторних роботах для побудови і дослідження схем будемо використовувати мікросхеми саме у DIP корпусах. На рисунку 1.5 показано загальний вигляд мікросхеми логічного елементу 74HC00 I- НІ (NAND) у DIP та SOIC корпусах відповідно. Цифра біля типу корпуса вказує на кількість виводів. Відповідно, DIP-14 вказує на те, що мікросхема має 14 виводів по 7 з кожної довшої сторони, що видно на рис. 1.4.а.





Рис. 1.5. Загальний вигляд мікросхем логічного елемента І-НІ 74HC00:  
а) у корпусі DIP-14; б) у корпусі SOIC-14

Для того, щоб визначити принцип підключення мікросхеми для її подальшого використання, необхідно ознайомитись із її технічним описом, т.з. Datasheet. Цей документ описує всі основні характеристики схеми, як механічні, так електричні та часові. Треба зауважити, що в залежності від виробника мікросхеми, Datasheet може мати різний вигляд. Наприклад, виводи для підключення можуть бути показані у вигляді функціональної схеми, або у вигляді принципової, як показано на рис. 1.6., проте такі відмінності не протирічають один одному.

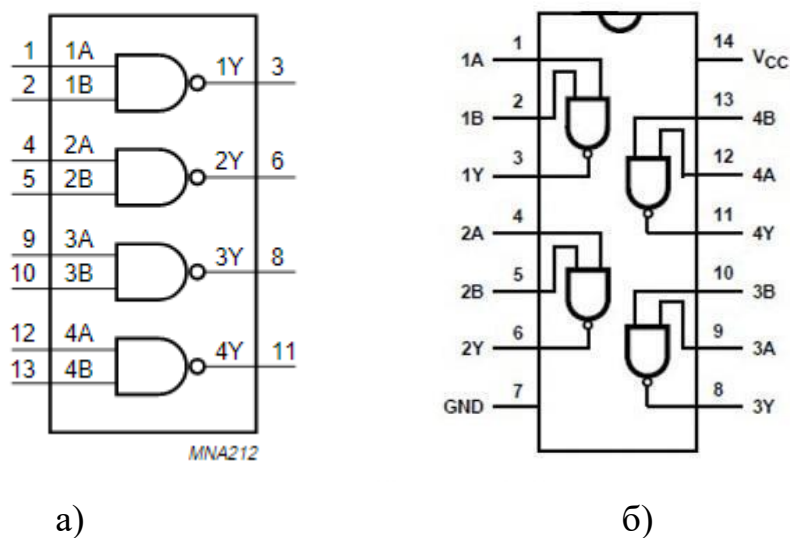


Рис. 1.6. Дані із технічного опису мікросхеми 74HC00:  
а) функціональна схема; б) принципова схема

Для того, щоб порівняти дані, що наведені у Datasheet із реальною мікросхемою, у нашому випадку дізнатись яка ніжка перша, яка друга і яка остання на корпусах мікросхем наносять спеціальні позначення, т.з. «ключ». Так, на рис. 1.5.а на корпусі мікросхеми присутня виїмка, яка і є ключем. Така ж виїмка показана на принциповій схемі 1.5.б. Відповідно, перша за рахунком ніжка буде знаходитись ліворуч від ключа, далі нумерація ніжок відбувається проти годинникової стрілки. Для мікросхем SOIC, ключ має вигляд втисненого кола, що і позначає початок нумерації ніжок.

## 1.2. Завдання для виконання

### 1.2.1. Завдання 1. Інвертор

У середовищі MicroCAP відкрити новий робочий простір File->New -> Schematic File. На робочому полі встановити інвертор, який знаходиться у папці Components / Digital Primitives/ Standard Gates / Inverter (Тип моделі обрати D\_00). Генератор прямокутного сигналу (Components / Analog Primitives/Waveform Sources/Pulse Source), тип моделі – Square, і резистор опором 10кОм. Зібрати схему як показано на рис. 1.7.

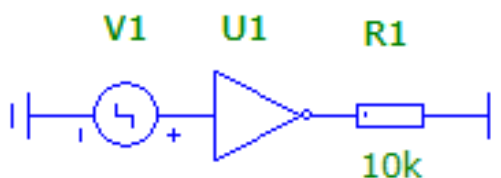


Рис. 1.7. Приклад реалізації схеми із інвертором

Зняти часові діаграми з генератора сигналу і з опору. Переконайтесь у виконанні інвертором заявлених функцій. Зняти і записати напругу рівнів логічного нуля та логічної одиниці на виході (опорі).

Нагадаємо, що для побудови часових діаграму у середовищі MicroCAP необхідно виконати команду Analysis/Transient (<Alt> + <1>). У вікні налаштувань в колонці Y Expression вказати значення V(V1) та V(R1).

Час моделювання задати таким чином, щоб виводилося 5-20 циклів зміни логічного рівня джерела V1. У результаті, таблиця налаштувань

моделювання має виглядати як показано на рис. 1.8., а результат моделювання – на рис. 1.9.

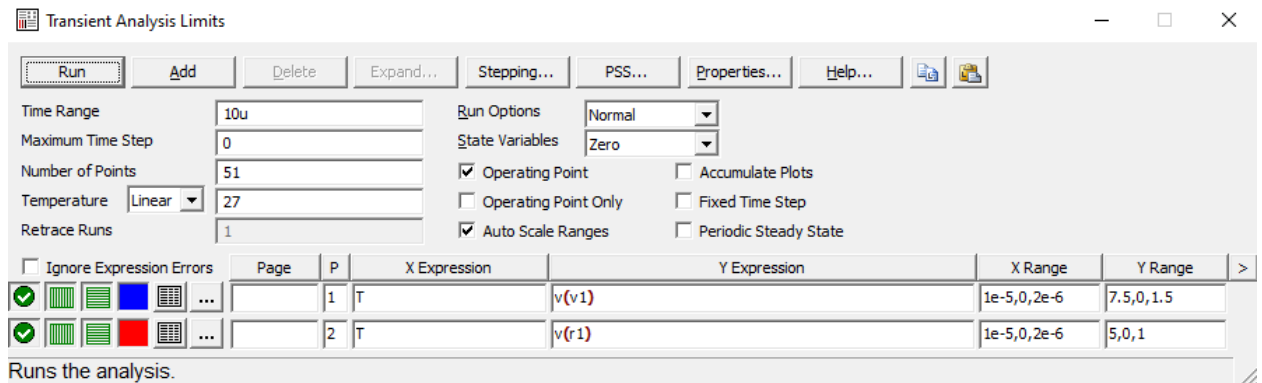


Рис. 1.8. Налаштування часового аналізу

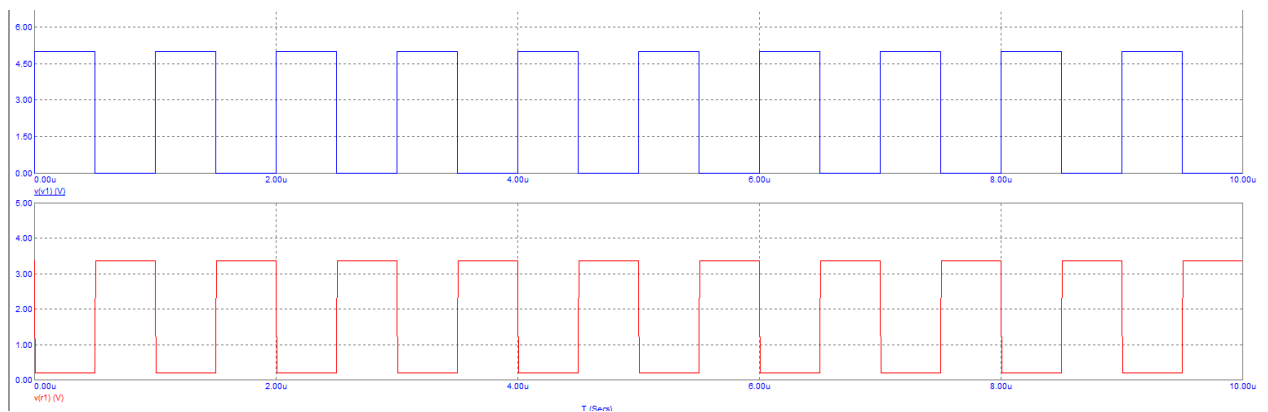


Рис. 1.9. Результат моделювання за завданням з п.п. 1.2.1

Тепер дослідимо вплив гармонічного сигналу на інвертор. Для цього треба змінити джерело прямокутного сигналу на два, Components / Analog Primitives/Waveform Sources/Sine Sources із параметрами амплітуди  $1V + N_{\text{варіанту}}/5$  та частотою 1кГц, та Battery із номінальною напругою 2.5В. Після заміни схема має виглядати як показано на рис. 1.10 (Індекси елементів можуть відрізнятись).

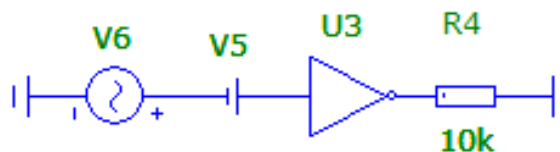


Рис. 1.10. Схема для дослідження інвертора при гармонічному сигналі

Вивести графіки напруг вхідного та вихідного сигналу. Час моделювання вибрати таким, щоб відображалось 5-10 змін стану інвертора.

Визначити напругу вхідного сигналу, при якому відбувається переключення логічного елемента.

Тепер замінимо модель інвертора тригером Шмітта. За допомогою пошуку (Ctrl+Shift+F) знайдемо модель 74НС14. Для порівняння результатів, пропонується зробити копію схеми, що показана на рис. 1.10, і лише після цього проводити заміну інвертора на тригер Шмітта.

Вивести графіки напруг вхідного та вихідного сигналу. Час моделювання вибрати таким, щоб відображалось 5-10 змін стану інвертора. Визначити напруги вхідного сигналу, при якому відбувається переключення тригера Шмітта.

Результати моделювання вивести одразу для двох схем (інвертор і тригер Шмітта) як показано на рис. 1.11.

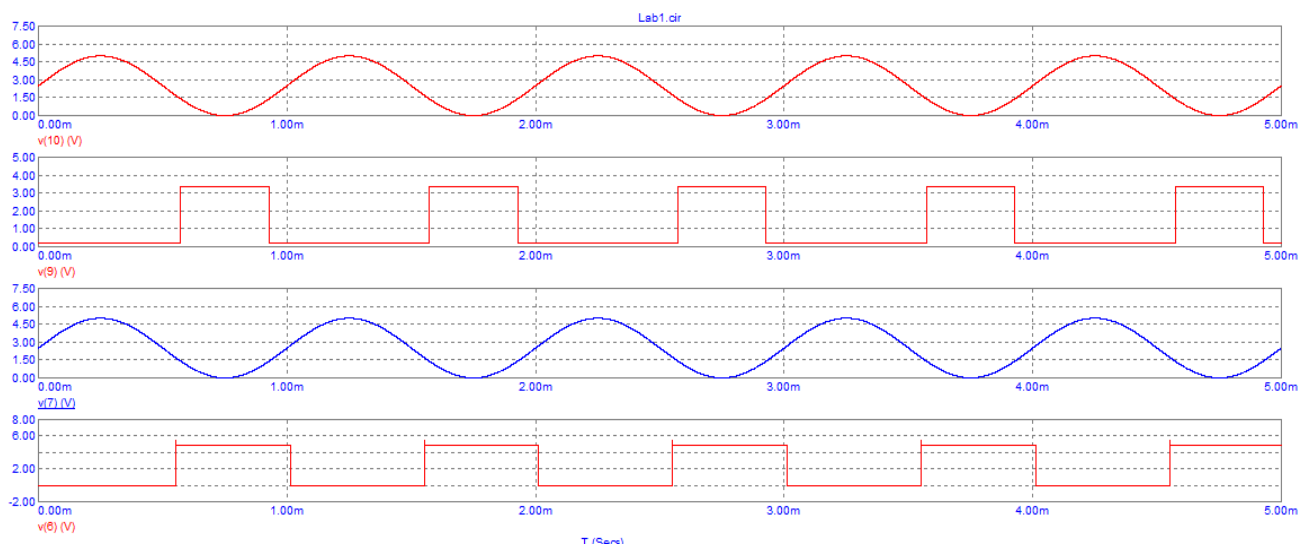


Рис. 1.11. Результат моделювання впливу гармонічного сигналу на інвертори

Отримані результати порівняти та зробити висновки.

### 1.2.3. Завдання 2. Дослідження логічних елементів

Перед початком моделювання із таблиці 1.4. необхідно обрати логічні елементи відповідно до номера варіанту. Варіант визначається за списком студентів, 11-й за списком - знову 1й-варіант, 12 - 2-ий і т.д. У MicroCAP дані моделі мікросхем знаходяться за адресою: Components / Digital Primitives/ Standard Gates.

Приклад схеми підключення логічного елемента у системі MicroCAP представлено на рис. 1.12.

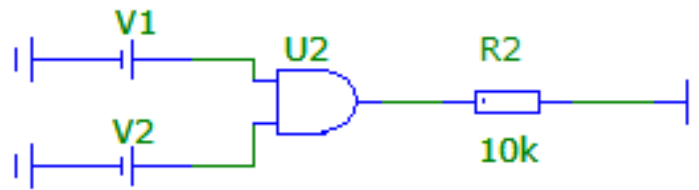
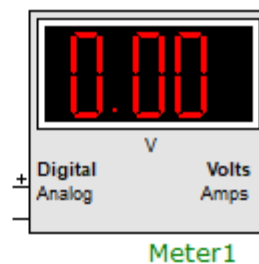


Рис. 1.12. Приклад реалізації схеми із логічним елементом

На входи логічного елемента встановити постійні джерела напруги (Battery). Для використовуваних моделей логічних елементів, напругу рівня логічного нуля приймемо  $U_0 = 0$  В, а напругу рівня логічної одиниці  $U_1 = +5$  В. Для зручності зміни напруги, поставити маркер у властивості Show (Показати) джерела напруги. Також рекомендується для швидкого перемикання рівнів на входах схеми встановити перемикачі (рис. 1.13. а), а на виході - графічний індикатор у вигляді цифрового чи аналогового вольтметра (рис. 1.13 б). Ці компоненти пропонується знайти самостійно, використовуючи меню пошуку.



а)



б)

Рис. 1.13. Додаткові компоненти: а) Animated Digital Switch; б) Animated Meter

Запустити аналіз перехідних процесів, відображаючи значення на входах логічного елемента та на його виході. Подаючи через джерела напруги рівні логічного нуля і одиниці, для заданого логічного елемента побудувати таблицю істинності. Приклад таблиці істинності для елементів із двома входами показаний в табл 1.5, а для елементів із трьома входами – в таблиці 1.6. відповідно.

Ознайомитись із Datasheet на логічний елемент (Модель логічного елемента вказує викладач).

Таблиця 1.4. Таблиця з варіантами завдання

| Варіант | Логічний елемент       |
|---------|------------------------|
| 1       | And2 (2I)              |
| 2       | And3 (3I)              |
| 3       | Nand2 (2I-НІ)          |
| 4       | Nand3 (3I-НІ)          |
| 5       | Nor2 (2АБО-НІ)         |
| 6       | Nor3 (3АБО-НІ)         |
| 7       | Or2 (2АБО)             |
| 8       | Or3 (3АБО)             |
| 9       | Xnor (Виключне АБО-НІ) |
| 10      | Xor (Виключне АБО)     |

Таблиця 1.5. Таблиця для заповнення для логічного елемента із двома входами

| Вхід 1 | Вхід 2 | Вихід |
|--------|--------|-------|
| 0      | 0      |       |
| 0      | 1      |       |
| 1      | 0      |       |
| 1      | 1      |       |

Таблиця 1.6. Таблиця для заповнення для логічного елемента із трьома входами

| Вхід 1 | Вхід 2 | Вхід 3 | Вихід |
|--------|--------|--------|-------|
| 0      | 0      | 0      |       |
| 0      | 0      | 1      |       |
| 0      | 1      | 0      |       |
| 0      | 1      | 1      |       |
| 1      | 0      | 0      |       |
| 1      | 0      | 1      |       |
| 1      | 1      | 0      |       |
| 1      | 1      | 1      |       |

### 1.3. Зміст звіту

1. Схеми логічного елемента побудовані в середовищі MicroCAP за своїм номером варіанту.
2. Для інвертора. Графіки перехідних процесів із позначенням напруги низького та високого рівнів.
3. Для інвертора при впливі графонічного сигналу схеми і графіки перехідних процесів (часового аналізу) для звичайного інвертора і тригера Шмітта. Вказати напруги спрацювання логічних елементів.
4. Таблиці результатів вимірів 1.5. або 1.6.
5. Зображення, що підтверджують проведені експерименти для різних вхідних значень (2-3 зображення) із таблиці істинності.
6. Ознайомитися з еквівалентними схемами РТЛ, ДТЛ і ТТЛ логіками, побудувати відповідні схеми (одну на вибір).
7. Висновки по роботі промодельованих логічних схем і відповідність їх теоретичним даним.

#### **1.4. Контрольні запитання**

1. Назвіть базові логічні елементи, які їх функції?
2. Що таке інвертор та для чого вони застосовуються?
3. Опишіть основні властивості тригера Шмітта.
4. Які корпуси мікросхем найбільш розповсюджені для базових логічних елементів, у чому їхня відмінність?
5. Що таке ключ на корпусі мікросхеми та як визначити номер ніжки мікросхеми?
6. У чому різниця аналогового сигналу та цифрового?
7. Чи можна подавати аналоговий сигнал на цифровий логічний елемент?
8. Як створити логічний елемент із трьома входами, якщо наявні лише двовходові логічні елементи?



# КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 2

## ФОРМУВАННЯ І ПЕРЕТВОРЕННЯ ЛОГІЧНИХ ПОСЛІДОВНОСТЕЙ

**Мета роботи:** освоїти перетворення логічних виразів, вивчити закони алгебри логіки і підтвердити правильність перетворень моделями в MicroCAP.

### 2.1. Теоретичні відомості

#### 2.1.1. Умовні позначення логічних операцій

Оскільки майже всі цифрові пристрої складаються із елементів, що виконують базові логічні функції, а їхні послідовності бувають доволі складними, то попередньо, при створенні таких пристроїв формують їхню математичну або логічну модель. Для спрощення запису таких виразів використовуються умовні позначення:

|                                   |                                             |
|-----------------------------------|---------------------------------------------|
| $\neg A, \bar{A}$                 | НІ (НЕ) $A$ (заперечення, інверсія)         |
| $A \wedge B, A \cdot B$           | $A$ і $B$ (логічне множення, кон'юнкція)    |
| $A \vee B, A + B$                 | $A$ або $B$ (логічне додавання, диз'юнкція) |
| $A \rightarrow B$                 | імплікація (проходження)                    |
| $A \leftrightarrow B, A \equiv B$ | еквіваленція (еквівалентність)              |

- Операцію імплікації можна виразити через «АБО» і «НІ»:

$$A \rightarrow B = \neg A \vee B \text{ або в інших позначеннях } A \rightarrow B = \bar{A} + B$$

- Операцію еквіваленції також можна виразити через «АБО» і «НІ»:

$$A \leftrightarrow B = \neg A \wedge \neg B \vee A \wedge B \text{ або в інших позначеннях } A \leftrightarrow B = \bar{A} \cdot \bar{B} + A \cdot B$$

Дуже часто, записані логічні вирази можуть бути спрощені. Під спрощенням формули, що не містить операцій імплікації і еквіваленції, розуміють рівносильне перетворення, яке приводить до формули, що містить у порівнянні з вихідною менше число операцій кон'юнкції і диз'юнкції, або містить меншу кількість входів змінних.

Рівносильні перетворення логічних формул мають те ж призначення, що і перетворення формул у звичайній алгебрі. Вони служать для спрощення формул або приведення їх до певного виду, використовуючи основні закони алгебри логіки.

Так само як у звичайній математиці, в алгебрі логіки є жорстка послідовність операцій. Пріоритет операцій зазначено в таблиці 2.1.

Таблиця 2.1. Пріоритет операцій алгебри логіки

| Пріоритет                                                                      | Операція                                                    |
|--------------------------------------------------------------------------------|-------------------------------------------------------------|
| 1                                                                              | Операція з одним операндом (унарна операція) – НІ, інверсія |
| 2                                                                              | Кон'юнкція - І                                              |
| 3                                                                              | Диз'юнкція - АБО                                            |
| 4                                                                              | Сума по модулю два – виключне АБО                           |
| <b>* Дужки змінюють пріоритет операцій, такі дії виконуються в першу чергу</b> |                                                             |

Операції одного рангу виконуються зліва направо в порядку написання логічного виразу. Алгебра логіки лінійна і для неї справедливий принцип суперпозиції.

Для спрощення логічних виразів користуються тими самими законами, що і у алгебрі, проте адаптованих під двійкову систему числення. Із-за цього у таких законів з'являються додаткові властивості. Отже, основні закони для перетворення логічних виразів приведемо в таблиці 2.2.

Таблиця 2.2. Правила перетворення логічних виразів (закони алгебри логіки)

| Закон                  | Для І                          | Для АБО                |
|------------------------|--------------------------------|------------------------|
| подвійного заперечення | $\overline{\overline{A}} = A$  |                        |
| виключення третього    | $A \cdot \overline{A} = 0$     | $A + \overline{A} = 1$ |
| виключення констант    | $A \cdot 1 = A; A \cdot 0 = 0$ | $A + 0 = A; A + 1 = 1$ |
| повторення             | $A \cdot A = A$                | $A + A = A$            |

Таблиця 2.2. Продовження

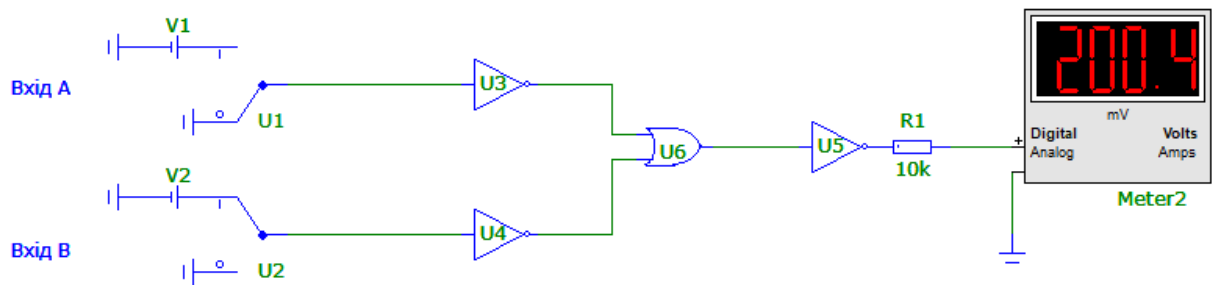
| Закон           | Для І                                                | Для АБО                                              |
|-----------------|------------------------------------------------------|------------------------------------------------------|
| поглинання      | $A \cdot (A + B) = A$                                | $A + A \cdot B = A$                                  |
| переміщувальний | $A \cdot B = B \cdot A$                              | $A + B = B + A$                                      |
| сполучний       | $A \cdot (B \cdot C) = (A \cdot B) \cdot C$          | $A + (B + C) = (A + B) + C$                          |
| розподільний    | $A + B \cdot C = (A + B) \cdot (A + C)$              | $A \cdot (B + C) = A \cdot B + A \cdot C$            |
| де Моргана      | $\overline{A \cdot B} = \overline{A} + \overline{B}$ | $\overline{A + B} = \overline{A} \cdot \overline{B}$ |

*Наведемо приклад.* Спростимо вираз  $\overline{\overline{A} + \overline{B}}$ . Застосувавши закон де Моргана, звільняємося від загальної інверсії, отримаємо  $\overline{\overline{A}} \cdot \overline{\overline{B}}$ , після, застосувавши закон подвійного заперечення  $A \cdot B$ . Перевіркою результату, служить побудова таблиць істинності для таких виразів, вони будуть співпадати.

У результаті застосування закону де Моргана ми значно спростили початковий вираз. Окрім цього це дозволяє сформулювати цікавий висновок: якщо інвертувати всі входи і вихід логічного елемента АБО, отримаємо логічний елемент І, і навпаки. Такий підхід дуже часто використовується для заміни логічних елементів при збірці електронних схем та пристроїв.

Тепер побудуємо ці вирази в середовищі MicroCAP і впевнимися в правильності наведених перетворень. У якості логічних рівнів на входах візьмемо рівні напруг, що відповідають 5В – лог. 1, та 0В – лог. 0. Для швидкого дослідження схеми будемо використовувати інструменти, що показані на рис. 1.13. Загальний вигляд зібраних в середовищі MicroCAP схем, що відповідають досліджуваним виразам, показаний на рис. 2.1.

### Початковий вираз



### Еквівалентний вираз

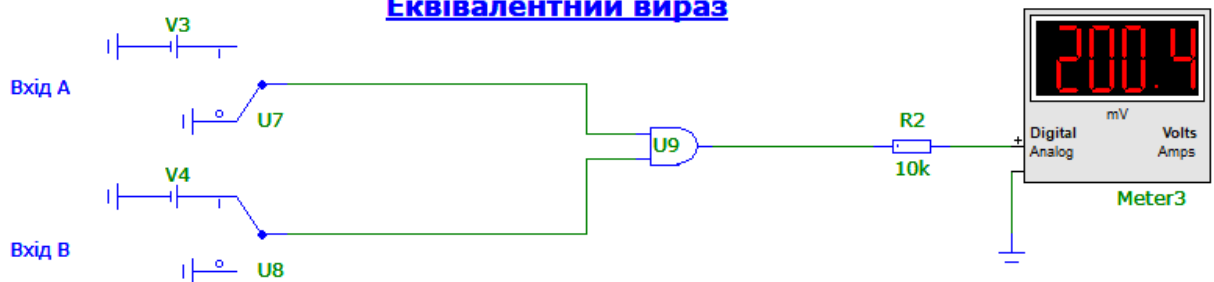


Рис. 2.1. Результат моделювання у MicroCAP

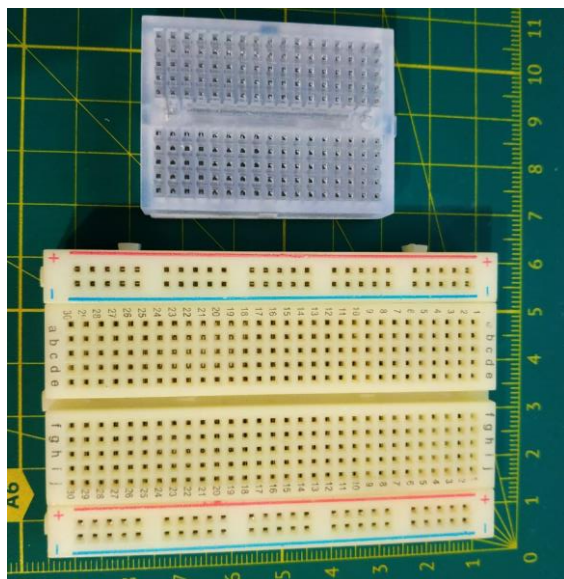
Для відображення значень на цифрових вольтметрах необхідно і достатньо виконати моделювання перехідних процесів Analysis/Transient (<Alt> + <1>), причому параметри аналізу у вікні налаштувань можна не змінювати.

#### **2.1.2. Знайомство із елементною базою**

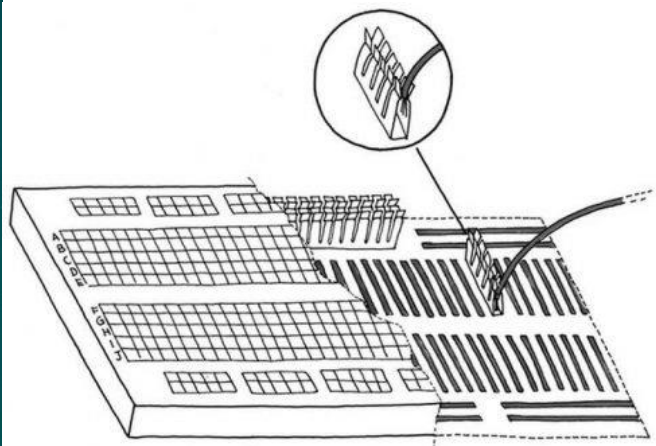
Як було вказано в минулому практикумі, для збірки схем будемо використовувати мікросхеми у корпусах форм-фактору DIP-14 (такий корпус зручний для швидкого монтажу і дослідження). Також для зручної роботи із мікросхемами та іншим приладдям будемо використовувати макетні плати для безпечного монтажу, т.з. бредборди. Загальний вигляд таких макетних плат показано на рис. 2.2 а, а конструкція на рис. 2.2.б.

Така макетна плата складається з пластмасової основи, в якій є набір струмопровідних контактних роз'ємів. Контактні роз'єми підпружинені і тому надійно фіксують дрiт, контакт елемента або ніжку мікросхеми. Залежно від конструкції макетної плати контактні роз'єми поєднуються в рядки, зазвичай,

по 5 штук. В результаті утворюється п'ятиконтактний роз'єм. Кожен із роз'ємів дозволяє підключати до нього виводи електронних компонентів або струмопровідних провідників діаметром, як правило, від 0,4 до 0,7 мм. Роз'єми розраховані на 50 000 циклів підключення/відключення. Існують макетні плати різних розмірів: 170, 400, 830, 2800, 3220 отворів (pin), в яких отвори розташовані на стандартній відстані 2,54 мм по горизонталі та вертикалі, що дозволяє підключати мікросхеми у корпусі DIP. По середині макетна плата розділена на дві рівні незв'язані частини, що спрощує комутацію схем. Як показано на рис. 2.2.а, по боках макетної плати розташовані контактори, що поєднуються у дві паралельні лінії (позначені червоним і синім кольором). Ці контакти зазвичай використовуються як лінії живлення.



а)



б)

Рис. 2.2. Макетні плати для безпечного монажу: а) загальний вигляд і різновиди макетних плат; б) конструктивна схема

Для комутації схеми застосовуються ніжки використовуваних радіодеталей і дроти-перемички, які бувають гнучкі та жорсткі, для зручності монтажу (рис. 2.3.).

Для забезпечення живлення зручно використовувати адаптер живлення для таких макетних плат, загальний вигляд якого показано на рис. 2.4.



Рис. 2.3. Дроти-перемички, що використовуються для безпачних макетних плат

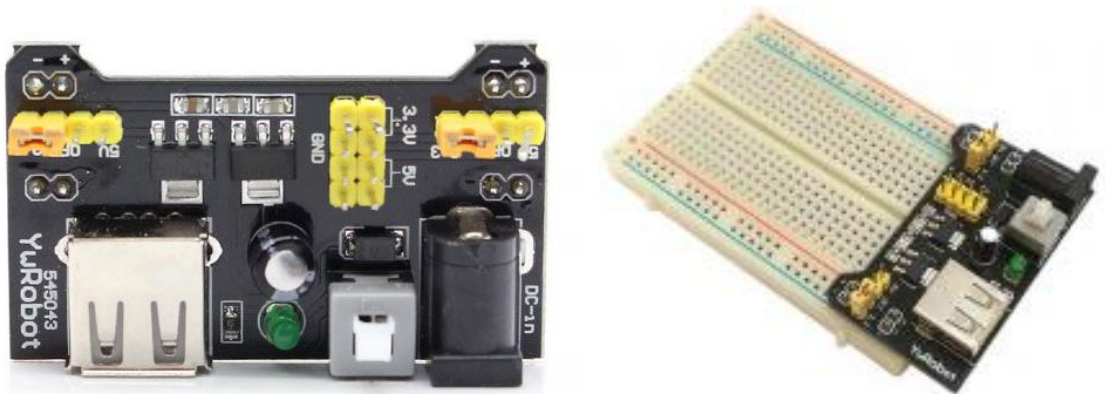


Рис. 2.4. Адаптер живлення для безпачних макетних плат

Адаптер приєднується до бокових ліній живлення і має два вбудовані лінійні стабілізатори напруги на 5В та 3.3В із можливістю перемикання, що є стандартними напругами живлення більшості використовуваних у курсі мікросхем та елементів. Максимальний струм становить 100мА. Вхідна напруга варіюється у межах 6-9В, зовнішнє джерело живлення приєднується через стандартний роз'єм 2.1x5.5мм.

## 2.2. Завдання для виконання

Спростити вираз використовуючи закони алгебри логіки. Для перевірки правильності перетворення, у середовищі MicroCAP побудувати дві схеми, початкову та спрощену, та порівняти їх таблиці істинності. Рівні напруг на вході взяти 0В і 5В відповідно для низького і високого рівня.

1. Вкажіть, який логічний вираз рівносильний виразу  $\neg(A \vee \neg B \vee C)$  ?

- 1)  $\neg A \vee B \vee \neg C$
- 2)  $A \wedge \neg B \wedge C$
- 3)  $\neg A \vee \neg B \vee \neg C$
- 4)  $\neg A \wedge B \wedge \neg C$

2. Який логічний вираз рівносильний виразу  $\neg(A \wedge B) \wedge \neg C$  ?

- 1)  $\neg A \vee B \vee \neg C$
- 2)  $(\neg A \vee \neg B) \wedge \neg C$
- 3)  $(\neg A \vee \neg B) \wedge C$
- 4)  $\neg A \wedge \neg B \wedge \neg C$

3. Вкажіть, який логічний вираз рівносильний виразу  $\neg(\neg A \wedge B)$  ?

- 1)  $A \vee \neg B$
- 2)  $\neg A \vee B$
- 3)  $B \wedge \neg A$
- 4)  $A \wedge \neg B$

4. Який логічний вираз рівносильний виразу  $\neg(A \vee \neg B)$  ?

- 1)  $A \vee B$
- 2)  $A \wedge B$
- 3)  $\neg A \vee \neg B$
- 4)  $\neg A \wedge B$

5. Який логічний вираз рівносильний виразу  $\neg(\neg A \vee \neg B) \wedge C$  ?

- 1)  $(A \vee \neg B) \vee C$
- 2)  $A \wedge B \wedge C$
- 3)  $(A \rightarrow \neg B) \vee C$
- 4)  $\neg(A \vee \neg B) \vee C$

6. Який логічний вираз рівносильний виразу  $A \wedge \neg(\neg B \wedge \neg C)$ ?
- 1)  $A \wedge B \wedge C$
  - 2)  $A \vee B \vee \neg C$
  - 3)  $A \wedge (B \vee C)$
  - 4)  $(A \vee \neg B) \wedge \neg C$
7. Який логічний вираз рівносильний виразу  $\neg(A \vee B) \wedge \neg C$ ?
- 1)  $(A \vee B) \wedge \neg C$
  - 2)  $(A \wedge B) \wedge C$
  - 3)  $(\neg A \wedge \neg B) \wedge \neg C$
  - 4)  $(A \vee B) \wedge C$
8. Який логічний вираз рівносильний виразу  $\neg(A \vee \neg B) \wedge \neg C$ ?
- 1)  $A \vee B \wedge C$
  - 2)  $\neg(A \wedge B) \wedge C$
  - 3)  $\neg(A \vee C) \vee B$
  - 4)  $\neg(A \vee C) \wedge B$
9. Який логічний вираз рівносильний виразу  $\neg(\neg A \wedge B) \wedge \neg C$ ?
- 1)  $(A \wedge B) \wedge \neg C$
  - 2)  $(A \vee B) \vee C$
  - 3)  $(A \wedge \neg B) \vee \neg C$
  - 4)  $(A \vee \neg B) \wedge \neg C$
10. Який логічний вираз рівносильний виразу  $\neg(A \vee B) \rightarrow C$ ?
- 1)  $\neg A \wedge B \wedge C$
  - 2)  $A \vee B \vee C$
  - 3)  $\neg(A \vee B) \vee C$
  - 4)  $\neg A \vee \neg B \vee \neg C$
11. Який логічний вираз рівносильний виразу  $\neg(\neg A \vee \neg B) \wedge C$ ?
- 1)  $\neg A \vee B \vee \neg C$
  - 2)  $(\neg A \vee \neg B) \wedge \neg C$
  - 3)  $(A \vee B) \wedge C$
  - 4)  $A \wedge B \wedge C$



12. Який логічний вираз рівносильний виразу  $A \wedge \neg(B \vee \neg C)$ ?
- 1)  $A \wedge \neg B \wedge C$
  - 2)  $A \vee \neg B \vee \neg C$
  - 3)  $A \wedge \neg B \wedge \neg C$
  - 4)  $A \vee \neg B \vee C$
13. Який логічний вираз рівносильний виразу  $\neg(\neg A \vee B) \wedge C$ ?
- 1)  $\neg A \wedge B \wedge \neg C$
  - 2)  $(A \wedge \neg B) \vee C$
  - 3)  $(A \wedge B) \vee C$
  - 4)  $A \wedge \neg B \wedge C$
14. Який логічний вираз рівносильний виразу  $\neg(A \wedge \neg B \wedge \neg C)$ ?
- 1)  $\neg A \vee B \vee C$
  - 2)  $\neg A \vee B \vee \neg C$
  - 3)  $\neg A \wedge B \wedge C$
  - 4)  $A \wedge B \wedge \neg C$
15. Який логічний вираз рівносильний виразу  $\neg(\neg A \wedge \neg B) \wedge C$ ?
- 1)  $\neg A \vee B \vee \neg C$
  - 2)  $A \wedge B \wedge C$
  - 3)  $(A \vee B) \wedge C$
  - 4)  $(\neg A \wedge \neg B) \vee \neg C$
16. Який логічний вираз рівносильний виразу  $\neg(\neg A \wedge (\neg B \vee C))$ ?
- 1)  $\neg A \vee \neg B \vee C$
  - 2)  $A \wedge \neg B \wedge \neg C$
  - 3)  $A \vee B \wedge \neg C$
  - 4)  $A \wedge \neg B \wedge C$
17. Який логічний вираз рівносильний виразу  $\neg(A \vee \neg B \wedge C)$ ?
- 1)  $\neg A \wedge B \wedge \neg C$
  - 2)  $\neg A \wedge B \vee \neg C$
  - 3)  $\neg A \wedge (B \vee C)$
  - 4)  $\neg A \wedge B \vee \neg A \wedge \neg C$

18. Який логічний вираз рівносильний виразу  $\neg A \wedge \neg(\neg B \vee \neg\neg C) \vee D$ ?

- 1)  $\neg A \wedge \neg B \vee C \vee D$
- 2)  $\neg A \wedge \neg B \wedge \neg C \vee D$
- 3)  $\neg A \wedge B \wedge \neg C \vee D$
- 4)  $\neg A \wedge B \wedge C \wedge D$

19. Який логічний вираз рівносильний виразу  $\neg(A \vee \neg B) \wedge \neg C \wedge D$ ?

- 1)  $A \wedge \neg B \wedge C \wedge \neg D$
- 2)  $A \vee \neg B \wedge C \wedge D$
- 3)  $\neg A \vee B \vee \neg C \vee D$
- 4)  $\neg A \wedge B \wedge \neg C \wedge D$

20. Який логічний вираз рівносильний виразу  $\neg(\neg B \wedge \neg C) \wedge \neg A$ ?

- 1)  $\neg A \wedge (B \wedge C)$
- 2)  $\neg A \wedge \neg B \wedge C$
- 3)  $\neg A \vee B \vee \neg C$
- 4)  $\neg A \wedge (B \vee C)$

Варіант визначається за списком студентів, 21-й за списком - знову 1й-варіант, 22 - 2-ий і т.д.

**Додаткове завдання.** Використовуючи доступний набір мікросхем логіки, зібрати спрощену схему свого варіанту на макетній платі. У якості індикатору рівня на виході використовувати світлодіоди. Живлення мікросхем логіки встановити на рівні 5В. В якості вхідного сигналу логічної 1 використовувати напругу 3.3В.

При відсутності необхідного елемента в наборі мікросхем логіки, замінити на наявний, використовуючи закони бульової алгебри для заміни.

### 2.3. Зміст звіту

- 1. Дві схеми (початкова і спрощена) побудовані в середовищі MicroCAP за своїм номером варіанту.
- 2. Таблиці істинності для двох схем та 2-3 результати моделювання, що підтверджують правильність таблиць істинності.
- 3. Фото зібраної схеми на макетній платі за своїм варіантом.
- 4. Висновки по роботі.

## 2.4. Контрольні запитання

1. Яка пріоритетність операцій операндів бульової алгебри?
2. Які закони бульової алгебри ви знаєте?
3. Про що говорить закон де Моргана?
4. Яким чином можна замінити логічне АБО на логічне І?
5. Яким чином можна замінити логічне І на логічне АБО?
6. Як виразити еквіваленцію використовуючи «АБО» і «НІ»?
7. Як виразити імплікацію використовуючи «АБО» і «НІ»?
8. Для чого потрібна макетна плата для безпечного монтажу, які її переваги та недоліки?
9. Які відмінності між моделлю у середовищі MicroCAP та зібраною схемою ви помітили? Відповідь обґрунтуйте.

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 3

### ШИФРАТОРИ І ДЕШИФРАТОРИ

**Мета роботи:** дослідити роботу шифраторів та дешифраторів на базових логічних елементах, підтвердити працездатність побудованих комбінаційних пристроїв.

#### 3.1. Теоретичні відомості

Як було сказано раніше, більшість цифрових пристроїв у своїй структурі мають логічні елементи поєднані за певним принципом. Поєднання логічних елементів дає певну комбінацію логічних дій, що утворює ряд нових пристроїв. Такі пристрої отримали назву комбінаційні схеми.

Комбінаційні схеми виконують більш складні функції, ніж прості логічні елементи. Їх входи об'єднані у функціональні групи і не є повністю взаємозамінними. Наприклад, будь-які два входи логічного елемента І-НЕ абсолютно спокійно можна поміняти місцями, від цього вихідний сигнал ніяк не зміниться. А для комбінаційних мікросхем це неможливо, бо кожен вхід має свою особливу функцію.

Об'єднує комбінаційні мікросхеми з логічними елементами те, що і ті, й інші не мають внутрішньої пам'яті. Тобто рівні їх вихідних сигналів завжди однозначно визначаються поточними рівнями вхідних сигналів і ніяк не пов'язані з попередніми значеннями. Будь-яка зміна вхідних сигналів обов'язково змінює стан вихідних сигналів. Саме тому логічні елементи іноді також називають комбінаційними мікросхемами на відміну від послідовних (або послідовносних) мікросхем, які мають внутрішню пам'ять і керуються не рівнями вхідних сигналів, а їх послідовностями.

##### 3.1.1. Шифратори

Шифратор – цифрова логічна комбінаційна мікросхема, що перетворює номер вхідного сигналу у вихідний двійковий код (шифрує номер вхідного сигналу). Кількість входів і виходів шифратора описується співвідношенням

$2^N$  – входів і  $N$ - виходів, що відповідає розрядності вихідного коду. Тобто, наприклад, якщо шифратор має 8 входів, то виходів у нього буде три, що дорівнює всім можливим варіантам двійкового коду.

Зауважимо, що під входним сигналом мається наявність логічної 1 на одному і лише одному із входів шифратора. Інакше, активним завжди є тільки один вихід дешифратора, причому номер цього виходу (і відповідного йому сигналу) однозначно визначається входним кодом.

Мікросхеми дешифраторів позначаються на схемах буквами CD (від англійського Coder), див. рис. 3.1. *Вхід  $x_0$  умовно не показаний.*

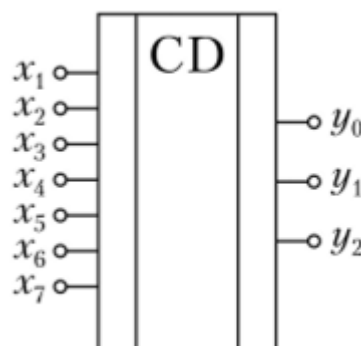


Рис. 3.1. Позначення дешифратора

Найпростіший варіант шифратора 2 в 1, що має відповідно два входи і один вихід представляє собою тривіальну конструкцію, що представлена на рис. 3.2.

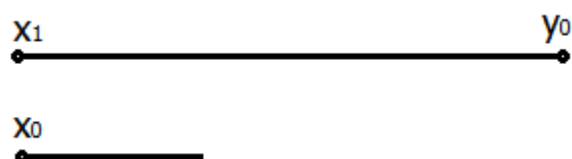


Рис. 3.2. Приклад реалізації шифратора 2 в 1

При приході сигналу на вхід  $x_0$  на виході  $y_0$  отримуємо значення логічного 0, при приході сигналу на  $x_1$  на виході маємо код 1, що відповідає номеру входу. Звісно, така конструкція не використовується.

Тепер розглянемо шифратор 4 в 2, схеми на логічних елементах АБО та позначення приведені на рис. 3.3.

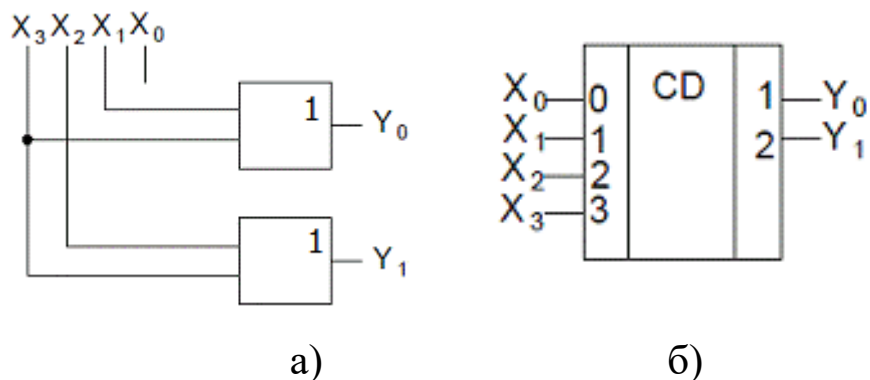


Рис. 3.3. Шифратор 4 в 2: а) Принципова схема на логічних елементах; б) Позначення

Таблиця істинності для такого шифратора буде мати вигляд як показано в табл. 3.1.:

Таблиця 3.1. Таблиця істинності шифратора 4 в 2

| Вхідні значення |                |                |                | Вихідний код   |                |
|-----------------|----------------|----------------|----------------|----------------|----------------|
| X <sub>0</sub>  | X <sub>1</sub> | X <sub>2</sub> | X <sub>3</sub> | Y <sub>1</sub> | Y <sub>0</sub> |
| 1               | 0              | 0              | 0              | 0              | 0              |
| 0               | 1              | 0              | 0              | 0              | 1              |
| 0               | 0              | 1              | 0              | 1              | 0              |
| 0               | 0              | 0              | 1              | 1              | 1              |

Для реалізації шифратора 8 в 3 зручно використовувати чотиривходові логічні елементи АБО - така схема представлена на рис. 3.4. Необхідно відмітити, що логіка збільшення кількості виходів дешифратора залишається такою самою при подальшому збільшенні розрядності вихідного коду.

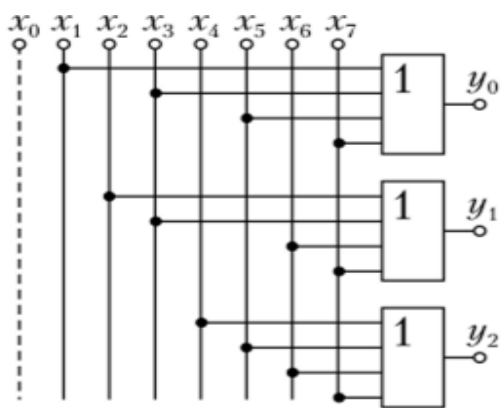


Рис. 3.4. Приклад реалізації шифратора 8 в 3

Таблицю істинності (див.табл. 3.2.) запишемо у більш компактному вигляді у табл.3.3:

Таблиця 3.3. Таблиця істинності шифратора 8 в 3

| № входу | Вихідний код   |                |                |
|---------|----------------|----------------|----------------|
|         | Y <sub>2</sub> | Y <sub>1</sub> | Y <sub>0</sub> |
| 0       | 0              | 0              | 0              |
| 1       | 0              | 0              | 1              |
| 2       | 0              | 1              | 0              |
| 3       | 0              | 1              | 1              |
| 4       | 1              | 0              | 0              |
| 5       | 1              | 0              | 1              |
| 6       | 1              | 1              | 0              |
| 7       | 1              | 1              | 1              |

### 3.1.2. Дешифратори

Дешифратор - це цифрова логічна мікросхема, що перетворює вхідний двійковий код в номер вихідного сигналу (дешифрує код). Пристрій має N входів та  $2^N$  виходів, де N розрядність дешифратора, що відповідає розрядності вхідного коду. На схемах позначається як мікросхема із позначенням DC (Decoder), див. рис.3.5.

Найпростіший однорозрядний дешифратор можна реалізувати одним інвертором, як показано на рис. 3.5.



Рис. 3.5. Приклад реалізації найпростішого дешифратора 1 в 2

Відповідно, коли на вхід приходить логічний 0, то сигнал (лог. 1) з'являється на виході Y<sub>0</sub>, коли на вхід подається лог.1, то сигнал з'явиться на виході Y<sub>1</sub>. Таким чином відбувається дешифрування двійкового коду.

Більш складну схему двобітного дешифратора 2 в 4 можна реалізувати, використовуючи логічні елементи І та інвертори. Така схема і відповідне позначення дешифратора показані на рис. 3.6., а таблиця істинності приведена в табл. 3.4.

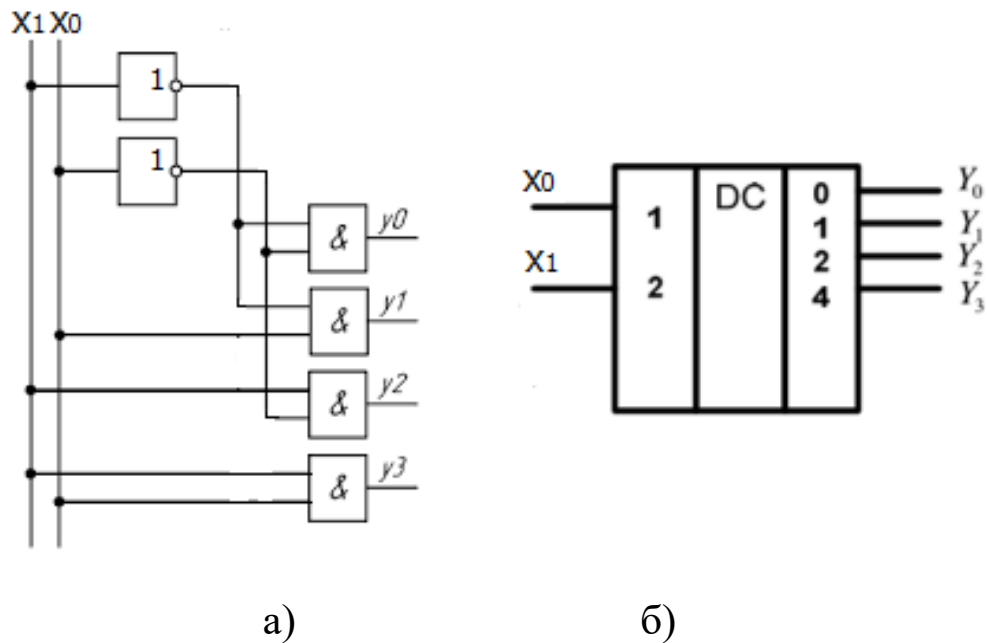


Рис. 3.6. Дешифратор 2 в 4: а) Принципова схема на логічних елементах; б) Позначення

Таблиця 3.4. Таблиця істинності дешифратора 2 в 4

| Вхідний код |       | Вихідні значення |       |       |       |
|-------------|-------|------------------|-------|-------|-------|
| $X_1$       | $X_0$ | $Y_3$            | $Y_2$ | $Y_1$ | $Y_0$ |
| 0           | 0     | 0                | 0     | 0     | 1     |
| 0           | 1     | 0                | 0     | 1     | 0     |
| 1           | 0     | 0                | 1     | 0     | 0     |
| 1           | 1     | 1                | 0     | 0     | 0     |

Збільшимо розрядність дешифратора до 3, отримавши схему на рис. 3.7. та таблицю істинності 3.5.



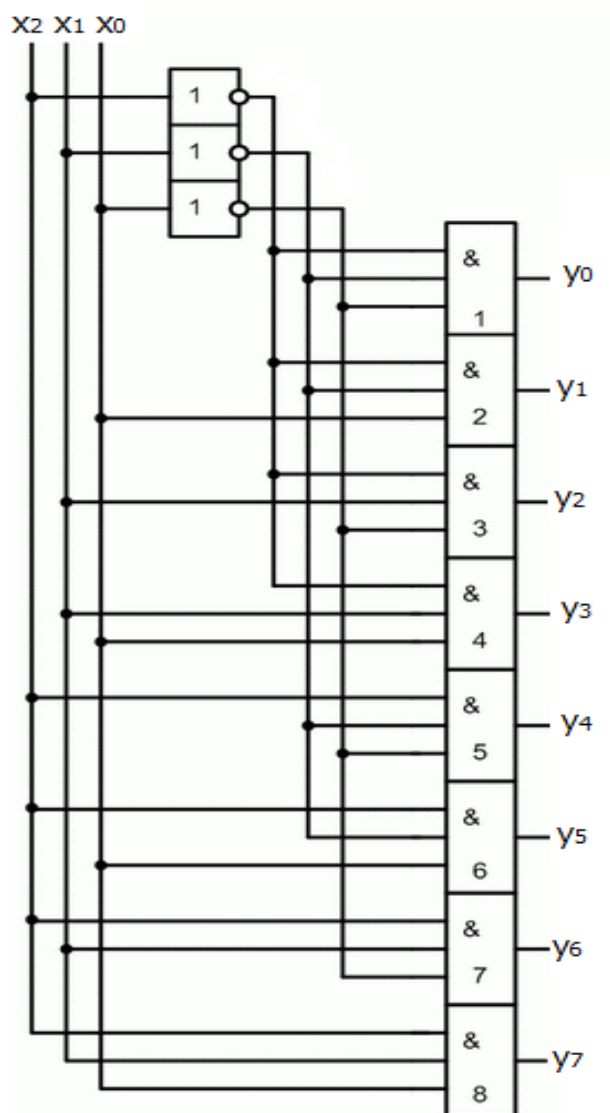


Рис. 3.7. Приклад реалізації дешифратора 3 в 8.

Таблиця 3.5. Таблиця істинності дешифратора 3 в 8

| Вхідний код    |                |                | № виходу |
|----------------|----------------|----------------|----------|
| X <sub>2</sub> | X <sub>1</sub> | X <sub>0</sub> |          |
| 0              | 0              | 0              | <b>0</b> |
| 0              | 0              | 1              | <b>1</b> |
| 0              | 1              | 0              | <b>2</b> |
| 0              | 1              | 1              | <b>3</b> |
| 1              | 0              | 0              | <b>4</b> |
| 1              | 0              | 1              | <b>5</b> |
| 1              | 1              | 0              | <b>6</b> |
| 1              | 1              | 1              | <b>7</b> |

### 3.2. Завдання для виконання

*Для непарних варіантів.* У середовищі MicroCAP побудувати шифратор 8 в 3, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

*Для парних варіантів.* У середовищі MicroCAP побудувати дешифратор 3 в 8, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

Використовуючи макетні плати та набір мікросхем логіки побудувати пристрій по своєму варіанту, розрядність пристрою – 2. Впевнитись в його працездатності.

У якості індикатору рівня на виході схеми використовувати світлодіоди. Живлення мікросхем логіки встановити на рівні 5В. В якості вхідного сигналу логічної 1 використовувати напругу 3.3В.

При відсутності необхідного елемента в наборі мікросхем логіки, замінити на наявний, використовуючи закони бульової алгебри для заміни.

Приклад зібраного шифратора 4 в 2 на логічних мікросхемах АБО-НІ та інверторах показано на рис. 3.8.

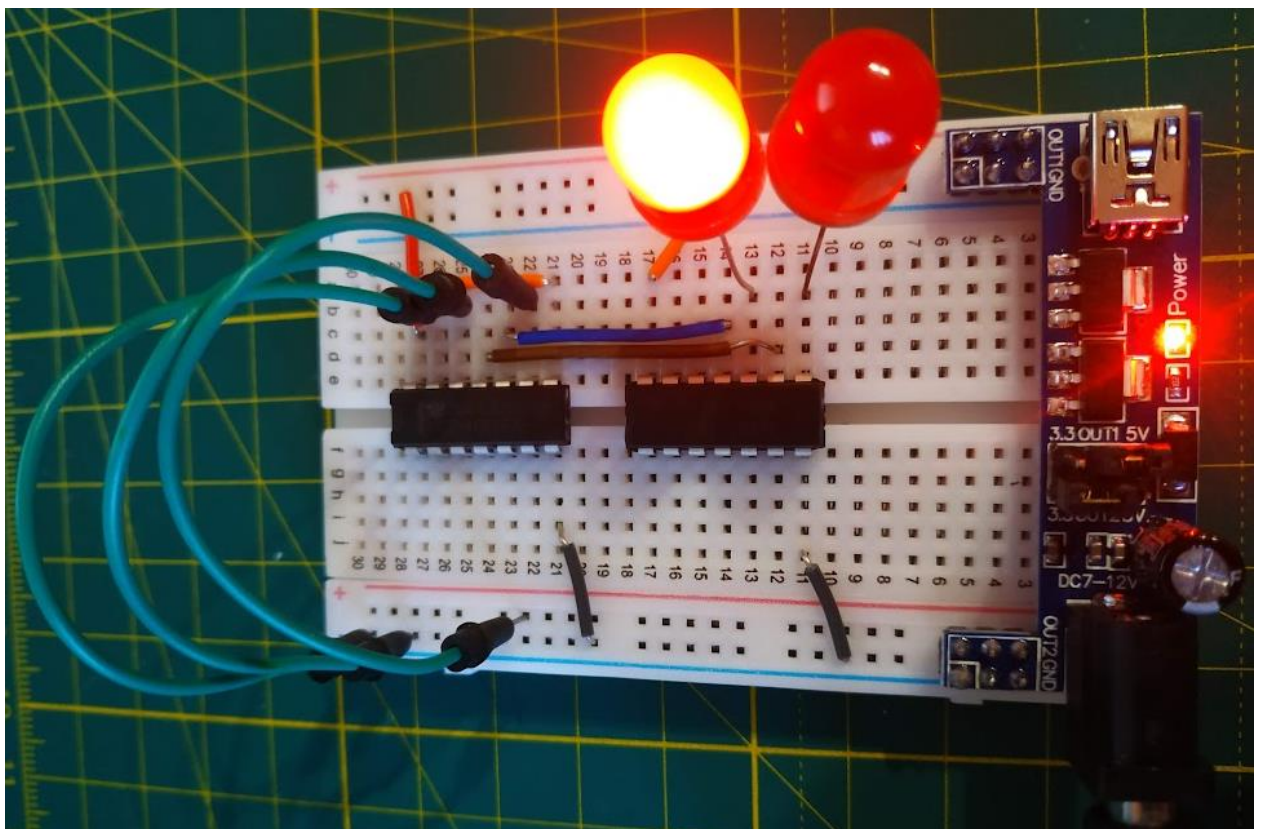


Рис. 3.8. Приклад зібраного шифратора 4 в 2

### **3.3. Зміст звіту**

1. Схема шифратора 8 в 3 чи дешифратора 3 в 8 згідно свого варіанту.
2. Діаграми, що доводять працездатність схеми, або 2-3 варіанти вихідних значень (в залежності від використання індикативних елементів).
3. Таблиця істинності для схеми свого варіанту.
4. Фото зібраної схеми на макетній платі за своїм варіантом.
5. Висновки по роботі.

### **3.4. Контрольні запитання**

1. Які функції виконує шифратор?
2. Які функції виконує дешифратор?
3. Чи можна реалізувати схему шифратора використовуючи інші логічні елементи окрім АБО?
4. Яка логіка зміни схеми шифратора реалізованого на логічних елементах АБО при збільшенні розрядності?
5. Чи може на входах шифратора одночасно бути більше ніж одна логічна 1?
6. Чи може на входах дешифратора одночасно бути більше ніж одна логічна 1?
7. Чи можна змінити індексацію входів або виходів шифратора або дешифратора, що зміниться?

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 4

### МУЛЬТИПЛЕКСОРИ ТА ДЕМУЛЬТИПЛЕКСОРИ

**Мета роботи:** ознайомитись із принципами роботи мультиплексорів та демультимплексорів, побудувати схеми на базових логічних елементах та підтвердити їх працездатність.

#### 4.1. Теоретичні відомості

##### 4.1.1. Мультиплексори

Наступним кроком ускладнення комбінацій логічних елементів є такі комбінаційні схеми як мультиплексори та демультимплексори.

Отже, мультиплексор (англ. multiplexer) - цифрова комбінаційна мікросхема, яка має 1 вихід, N адресних (керуючих) входів і  $2^N$  інформаційних входів.

Принцип його роботи наступний. Коли на адресні входи приходить код, що відповідає одному із номерів інформаційних входів, то цей і тільки цей вхід під'єднується до виходу. Значення, що прийшли на інші інформаційні входи ігноруються. На відміну від шифраторів тут під інформаційним сигналом (сигнал, що приходить на інформаційні входи) мається на увазі будь-яка послідовність логічних значень (напр. 1101001).

На принципових схемах мультиплексор позначається як MUX, MS, або MX (рис. 4.1).

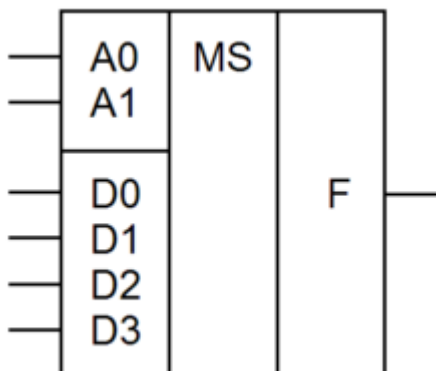


Рис. 4.1. Спрощене позначення мультиплексора на принципових електричних схемах

Одразу розглянемо схему мультиплексора на базових логічних елементах при  $N=2$ . Треба зазначити, що такий мультиплексор іноді називають 4 в 1. Схема представлена на рисунку 4.2.

Іноді кількість інформаційних входів може бути менша ніж  $2^N$ , тоді такий мультиплексор називається неповним.

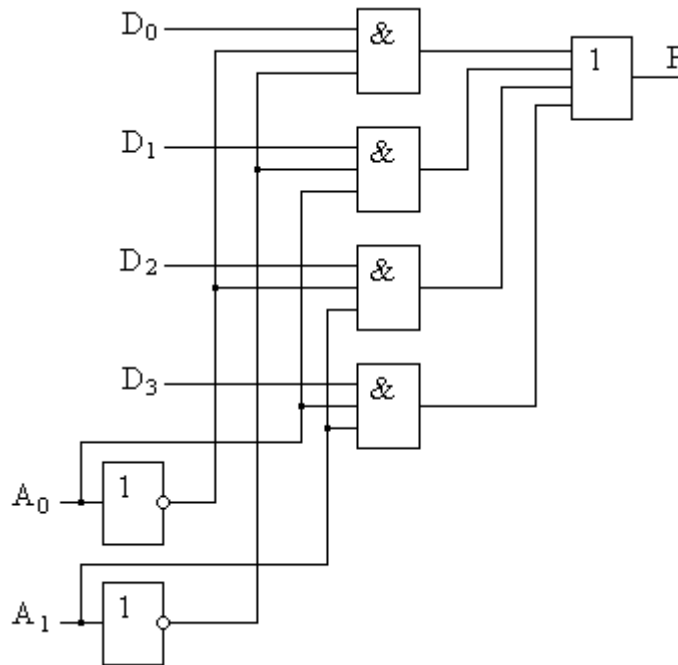


Рис. 4.2. Мультиплексор 4 в 1

Якщо уважно проаналізувати схему, то легко помітити, що основна частина нагадує дешифратор із доданим додатковим входом і в якого всі виходи зібрані на логічних елементах АБО. Тому часто мультиплексор представляють у вигляді схеми із використанням дешифратора, як показано на рис. 4.3.

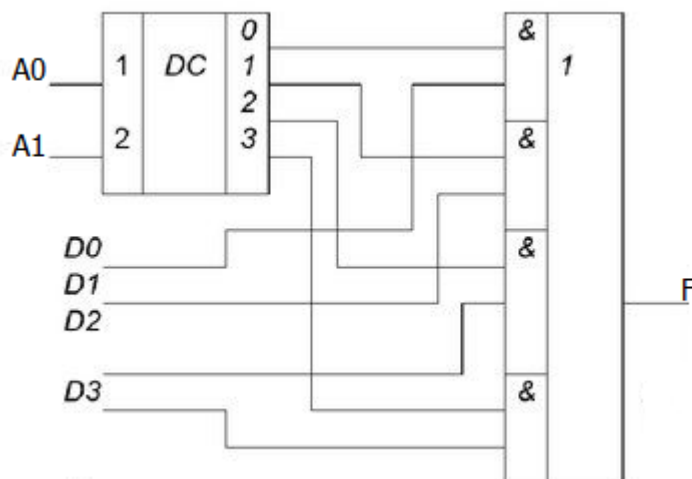


Рис. 4.3. Мультиплексор 4 в 1 на базі дешифратора

Таблиця істинності мультиплексора 4 в 1 буде виглядати наступним чином (див. табл 4.1.):

Таблиця 4.1. Таблиця істинності мультиплексора 4 в 1

| Адресні входи |    | Інформаційні входи |          |          |          | Вихід    |
|---------------|----|--------------------|----------|----------|----------|----------|
| A1            | A0 | D3                 | D2       | D1       | D0       | F        |
| 0             | 0  | x                  | x        | x        | Сигнал 1 | Сигнал 1 |
| 0             | 1  | x                  | x        | Сигнал 2 | x        | Сигнал 2 |
| 1             | 0  | x                  | Сигнал 3 | x        | x        | Сигнал 3 |
| 1             | 1  | Сигнал 4           | x        | x        | x        | Сигнал 4 |

Часто в конструкцію реальних мікросхем додають додатковий вхід Е (Enable), що блокує або дозволяє роботу схеми. Такий вхід, окрім своїх безпосередніх функцій, може використовуватись при комплексуванні мультиплексорів для збільшення кількості входів, як показано на рис. 4.4.

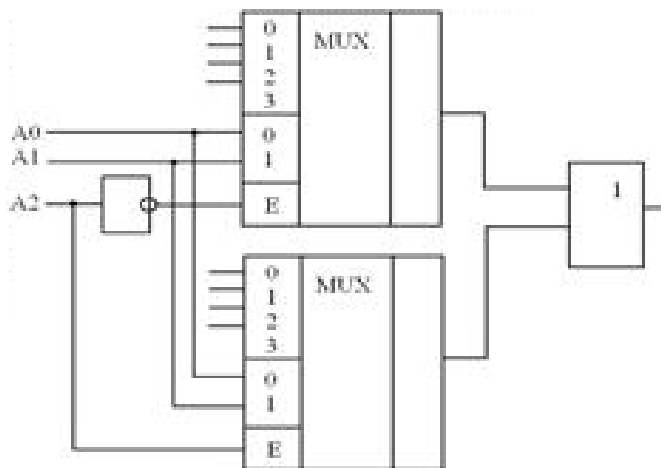


Рис. 4.4. Приклад комплексування мультиплексорів для збільшення кількості інформаційних входів.

Принцип складання схем мультиплексорів більшої розрядності (8 в 1, 16 в 1 т.п.) залишається незмінним, тому тут наведений не буде.

#### 4.1.2. Демультимплексори

Демультимплексор (demultiplexer) - цифрова комбінаційна мікросхема, яка має 1 інформаційний вхід, N адресних (керуючих) входів і  $2^N$  виходів.

Принцип його роботи наступний: коли на адресні входи приходить кодова послідовність, що відповідає одному із номерів виходів, то із цим виходом зв'язується інформаційний вхід.

На принципових схемах демультимплексор позначається як DMX, DMS, DX (рис. 4.5).

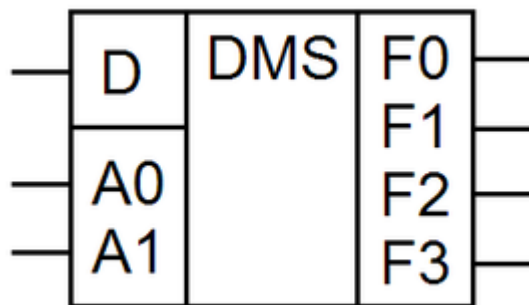


Рис. 4.5. Спрощене позначення демультимплексора на принципових електричних схемах

Розглянемо схему демультимплексора на базових логічних елементах при  $N=2$ . Схему складемо із використанням логічних елементів І та інверторів, як показано на рисунку 4.6.

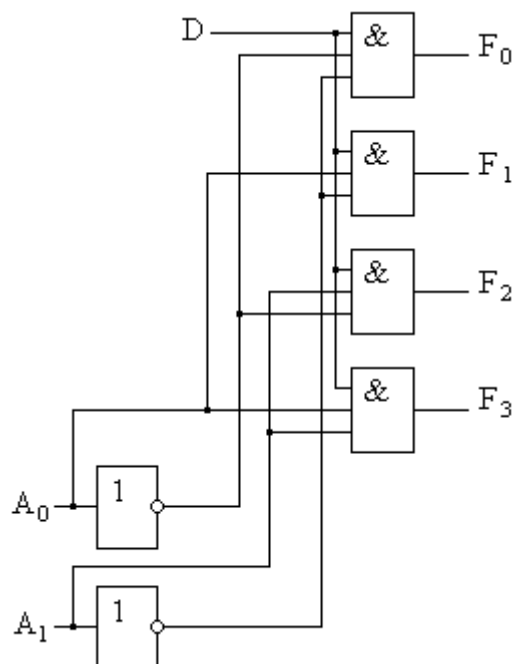


Рис. 4.6. Демультимплексор 1 в 4

Таблиця істинності буде мати наступний вигляд:

Таблиця 4.2. Таблиця істинності демультимплексора 1 в 4

| Інформаційний<br>вхід | Адресні<br>входи |    | Виходи |        |        |        |
|-----------------------|------------------|----|--------|--------|--------|--------|
|                       | A1               | A0 | F3     | F2     | F1     | F0     |
| <b>D</b>              | A1               | A0 | F3     | F2     | F1     | F0     |
| <b>Сигнал</b>         | 0                | 0  | х      | х      | х      | Сигнал |
| <b>Сигнал</b>         | 0                | 1  | х      | х      | Сигнал | х      |
| <b>Сигнал</b>         | 1                | 0  | х      | Сигнал | х      | х      |
| <b>Сигнал</b>         | 1                | 1  | Сигнал | х      | х      | х      |

Аналізуючи схему на рисунку 4.6. видно, що якщо подати постійний сигнал на інформаційний вхід D, то він ніяким чином не буде впливати на стан логічних елементів, що входять у схему. У такому випадку демультимплексор буде використовуватись у якості дешифратора. Представниками таких мікросхем є SN74155N та SN74HC138N.

Якщо порівняти рис. 4.2 та рис. 4.6., то видно, що основна частина, яка реалізована за допомогою двох інверторів та логічних елементів І не змінюється. Отже, в промисловості використовують такий підхід, щоб реалізувати універсальні мікросхеми мультимплексорів/демультимплексорів, як наприклад, 16 каналний CD74HC4067, загальний вигляд якого показано на рис. 4.7.

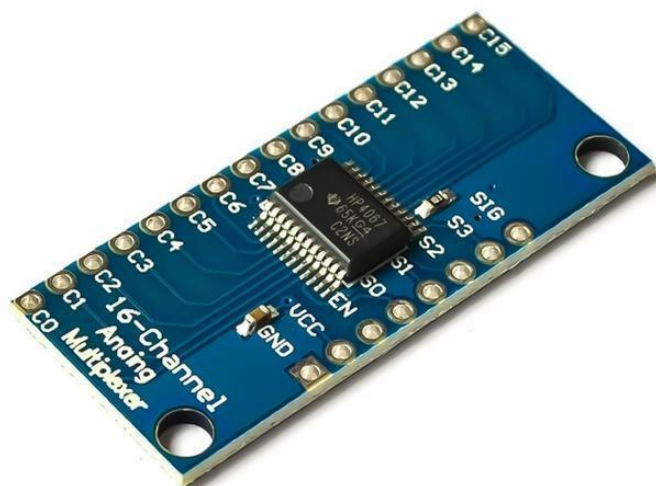


Рис. 4.7. 16 каналний мультимплексор/демультимплексор CD74HC4067



### 4.1.3. Джерело програмованого сигналу

Для зручного подання логічних рівнів на входи схем зручно використовувати джерело програмованого логічного сигналу Stim. Такі блоки можуть мати 1, 2, 4, 8 або 16 виходів, відповідно формуючи сигнал необхідної розрядності. Якщо необхідна інша кількість виходів, то обирається варіант із більшою кількістю контактів, але застосовуються не всі. Загальний вигляд джерел програмованого сигналу Stim показано на рис. 4.8.

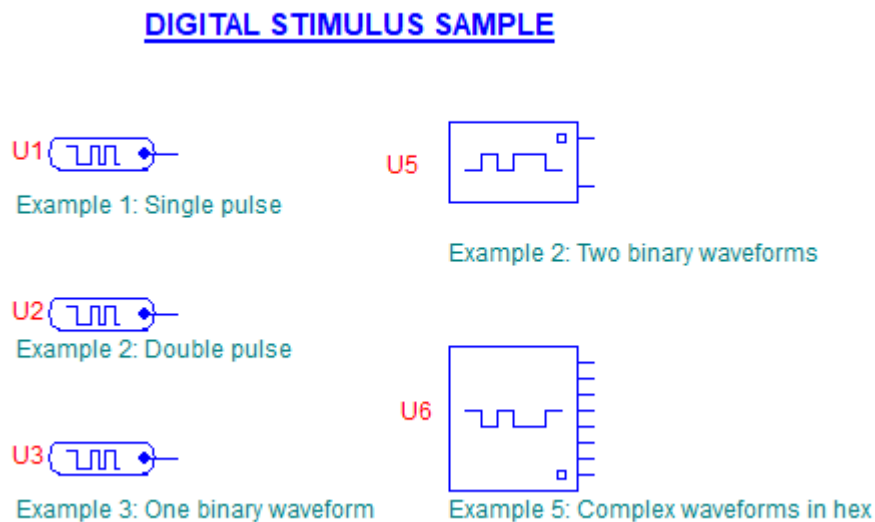


Рис. 4.8. Блоки Stim

Для більш детального ознайомлення із принципами роботи Stim у середовищі Micro-CAP пропонується набір демонстраційних схем. Так у меню [Help – Sample Circuits – Digital](#) пропонується ознайомитись із демо файлами [Digital Stimulus Generators](#).

Для прикладу згенеруємо послідовність двійкового коду від 0 до 111, що відповідає значенням від 0 до 7 у десятковій системі числення і є достатнім для формування таблиці істинності для де/мультиплексорів із  $N=3$ . Оскільки, трирозрядного Stim немає, то обираємо модель Stim4. У відкритому вікні налаштувань меню [Format](#) вписуємо 1111, що означатиме, що Stim буде видавати чотири двійкові числа. Можливі варіанти формату:

- 1 – двійкове число;
- 3 – число у вісімковій системі числення;
- 4 – число у шістнадцятковій системі числення.

Тепер обравши меню **COMMAND** запишемо наступний код (рис. 4.9.)

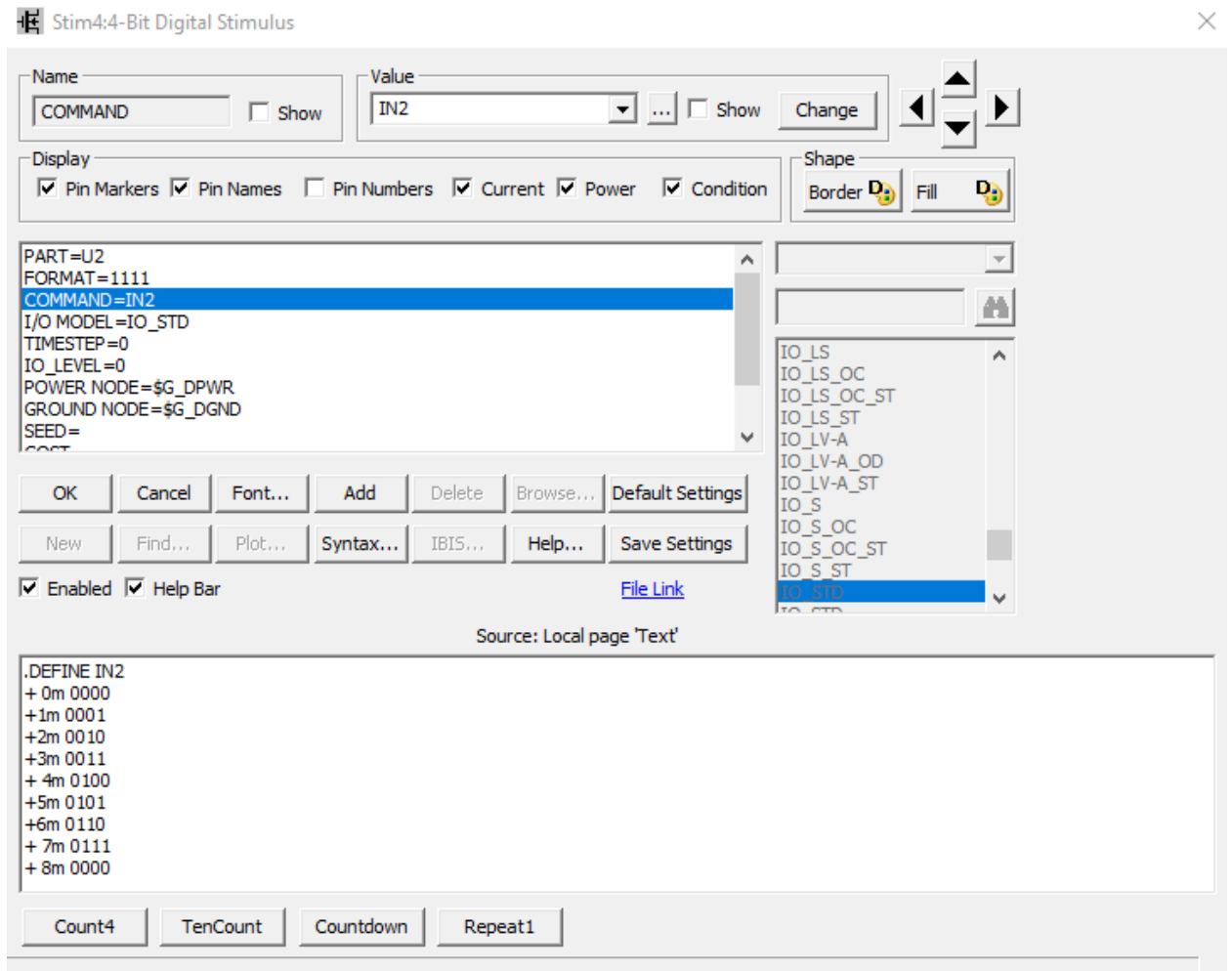


Рис. 4.9. Меню Stim із прикладом коду генерування значень від 0 до7

Значення **+ Xm** вказують на час формування заданого логічного рівня в мілісекундах. Так **+3m 0011** означає, що на 3-ій мілісекунді буде сформовано значення 0011. Результат моделювання показано на рис. 4.10.

Зазначимо, що при формуванні часових інтервалів можна використовувати стандартні десяткові приставки нано – n, мікро – u, мілі – m і т. д.

Більш детально про синтаксис і команди для формування сигналів блоком Stim див. в меню **Syntax...**, як видно на рис. 4.9.

При побудові часової діаграми зручно виводити не рівні напруги, а безпосередньо логічні рівні. Для цього у меню аналізу перехідних процесів **Analysis/Transient** в полі Y Expression треба вказати D і номер вузла/значення або елемент із частини схеми де необхідно провести аналіз.

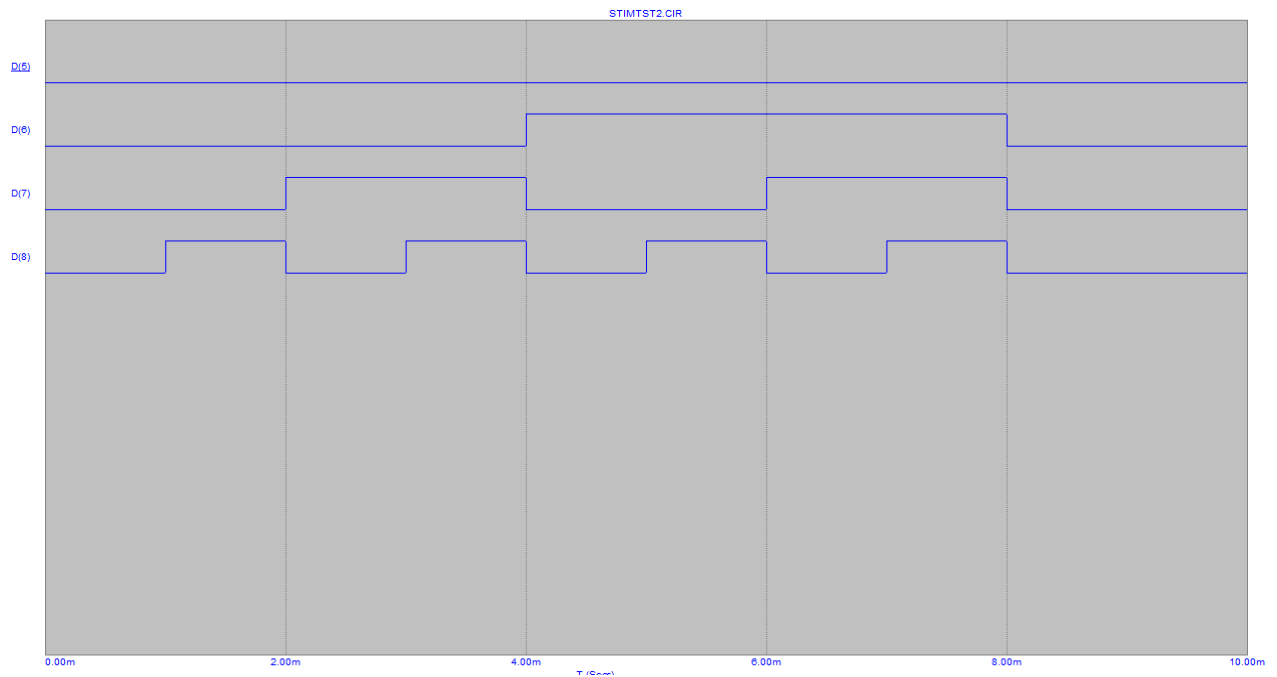


Рис. 4.10. Результат моделювання сигналу Stim.

#### 4.4. Завдання для виконання

*Для непарних варіантів.* У середовищі MicroCAP побудувати мультиплексор 8 в 1, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

*Для парних варіантів.* У середовищі MicroCAP побудувати мультиплексор 1 в 8, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

Для формування сигналів на адресних та інформаційних входах використовувати джерело програмованого логічного сигналу Stim.

Використовуючи макетні плати та набір мікросхем логіки, побудувати пристрій по своєму варіанту при  $N=1$ . Впевнитись у його працездатності.

При відсутності необхідного елемента в наборі мікросхем логіки, замінити на наявний, використовуючи закони бульової алгебри для заміни.

Додаткове завдання. Зібрати таку ж схему з використанням мікросхеми мультиплексора/демультиплексора CD74HC4067 (або іншого, з наявного набору, узгодивши із викладачем), впевнитись в її працездатності.

#### **4.4. Зміст звіту**

1. Схема мультиплексора 8 в 1 чи демультиплексора 1 в 8 за своїм варіантом.
2. Діаграми, що доводять працездатність схеми.
3. Таблиця істинності для схеми свого варіанту.
4. Фото зібраної схеми на макетній платі за своїм варіантом.
5. Висновки по роботі.

#### **4.4. Контрольні запитання**

1. Що таке мультиплексор, які функції він виконує?
2. Що таке демультиплексор, які функції він виконує?
3. Як побудувати мультиплексор/демультиплексор використовуючи дешифратор відповідної розрядності?
4. Які спільні риси мають дешифратор та демультиплексор?
5. Чи можна демультиплексор використовувати у якості дешифратора?
6. Які сигнали можуть подаватись на інформаційні входи де/мультиплексора?
7. Як сформувати сигнал використовуючи джерело програмованого логічного сигналу Stim?
8. Як при моделюванні часових діаграм вивести логічний рівень?

# КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 5

## КОМПАРАТОРИ ТА СУМАТОРИ

**Мета роботи:** ознайомитись із принципами роботи компараторів та суматорів, побудувати схеми на базових логічних елементах та підтвердити їх працездатність.

### 5.1. Теоретичні відомості

#### 5.1.1. Компаратори

Цифрові компаратори виконують порівняння двох чисел, заданих в двійковому коді. Вони можуть визначати рівність двох двійкових чисел  $A$  і  $B$  з однаковою кількістю розрядів або вид нерівності  $A > B$  чи  $A < B$ .

Найпростіший однобітний компаратор реалізований на логічних елементах виглядає як показано на рис. 5.1.а, а умовне графічне зображення мікросхеми – на рис. 5.1.б відповідно.

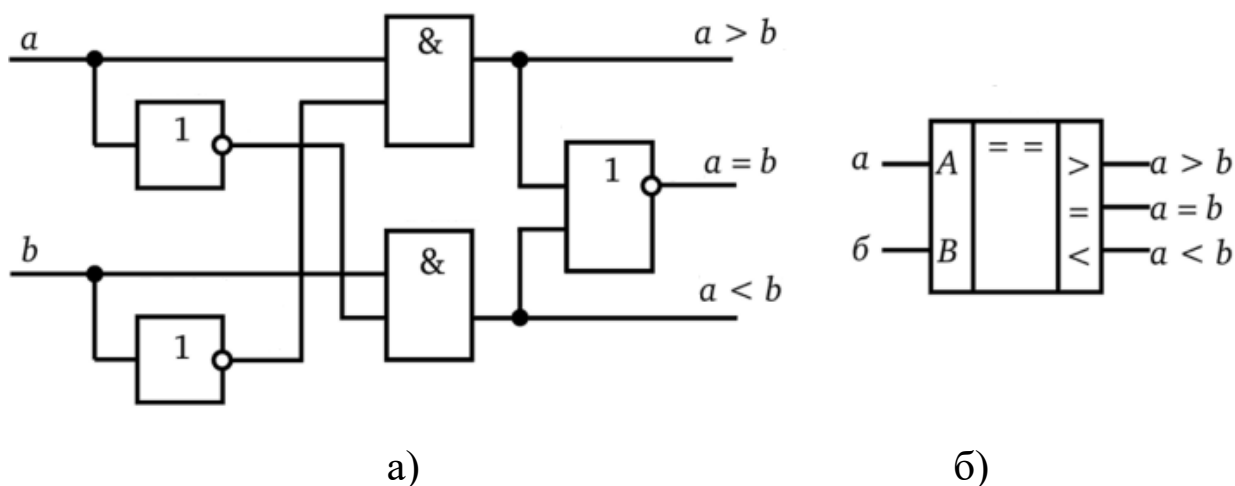


Рис. 5.1. Однобітний компаратор: а) Принципова схема на логічних елементах; б) Графічне позначення мікросхеми компаратора

У залежності від відношення вхідних сигналів  $A$  та  $B$  на відповідному виході формується сигнал у вигляді логічної одиниці.

Таблиця істинності однобітного компаратора тривіальна (табл.5.1.).

Компаратори більшої розрядності можуть бути побудовані на основі однобітних компараторів, приклад реалізації такої схеми показано на рис. 5.2.

Таблиця 5.1. Таблиця істинності однорозрядного компаратора

| Входи |   | Виходи |     |     |
|-------|---|--------|-----|-----|
| A     | B | A>B    | A=B | A<B |
| 0     | 0 | 0      | 1   | 0   |
| 0     | 1 | 0      | 0   | 1   |
| 1     | 0 | 1      | 0   | 0   |
| 1     | 1 | 0      | 1   | 0   |

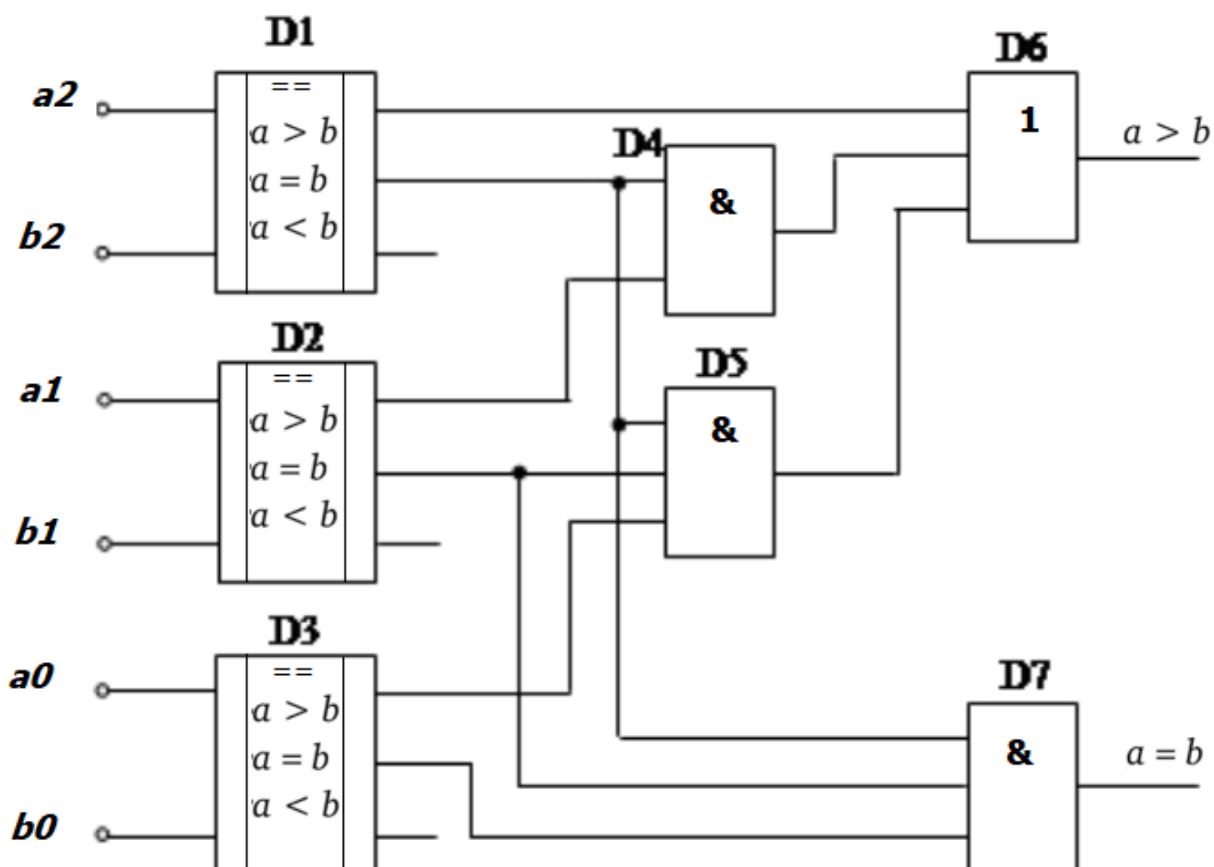
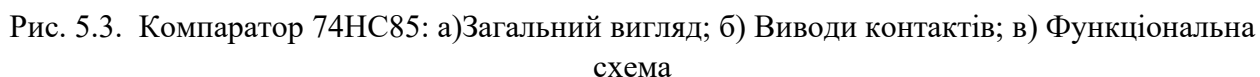


Рис. 5.2. Схема трирозрядного компаратора

Аналізуючи схему на рис. 5.2., можна побачити, що якщо всі розряди однакові, то однобітні компаратори формують на відповідних виходах логічні одиниці. Далі вони подаються на логічний елемент І, який і формує остаточний сигнал.

Перевірка на нерівність формується поетапно, починаючи із старшого розряду. Для розуміння схеми треба керуватись наступною логікою: якщо

Прикладом мікросхеми багаторозрядного компаратора є мікросхема 74НС85. Загальний її вигляд, виводи контактів та функціональна схема показані на рис. 5.3., а таблиця істинності відображена в табл. 5.2.



Таблиця 5.2. Таблиця істинності компаратора 74НС85

| COMPARING INPUTS                  |         |         |         | CASCADING INPUTS |       |       | OUTPUTS |       |       |
|-----------------------------------|---------|---------|---------|------------------|-------|-------|---------|-------|-------|
| A3, B3                            | A2, B2  | A1, B1  | A0, B0  | A > B            | A < B | A = B | A > B   | A < B | A = B |
| SINGLE DEVICE OR SERIES CASCADING |         |         |         |                  |       |       |         |       |       |
| A3 > B3                           | X       | X       | X       | X                | X     | X     | H       | L     | L     |
| A3 < B3                           | X       | X       | X       | X                | X     | X     | L       | H     | L     |
| A3 = B3                           | A2 > B2 | X       | X       | X                | X     | X     | H       | L     | L     |
| A3 = B3                           | A2 < B2 | X       | X       | X                | X     | X     | L       | H     | L     |
| A3 = B3                           | A2 = B2 | A1 > B1 | X       | X                | X     | X     | H       | L     | L     |
| A3 = B3                           | A2 = B2 | A1 < B1 | X       | X                | X     | X     | L       | H     | L     |
| A3 = B3                           | A2 = B2 | A1 = B1 | A0 > B0 | X                | X     | X     | H       | L     | L     |
| A3 = B3                           | A2 = B2 | A1 = B1 | A0 < B0 | X                | X     | X     | L       | H     | L     |
| A3 = B3                           | A2 = B2 | A1 = B1 | A0 = B0 | H                | L     | L     | H       | L     | L     |
| A3 = B3                           | A2 = B2 | A1 = B1 | A0 = B0 | L                | H     | L     | L       | H     | L     |
| A3 = B3                           | A2 = B2 | A1 = B1 | A0 = B0 | L                | L     | H     | L       | L     | H     |

Входи  $a > b$ ,  $a = b$  та  $a < b$  призначені для каскадування мікросхем, що дозволяє створювати компаратори збільшеної розрядності.

### 5.1.2. Суматори

Суматор – логічний комбінаційний пристрій, що виконує арифметичне додавання двійкових, трійкових або  $n$ -розрядних кодів.

Суматор (або повний суматор) складається із так званих неповних суматорів або напів суматорів. Неповний суматор – логічна схема, що має два входи і два виходи (бінарний суматор). Дозволяє обчислювати суму  $A + B$ , де  $A$  і  $B$  – це розряди двійкового числа. При цьому результатом будуть два біта  $S$  і  $C$  (або  $P$ ), де  $S$  – це біт суми по модулю 2, а  $C$  ( $P$ ) – біт переносу.

На електричних принципових схемах напівсуматори позначаються HS або знаком суми  $\Sigma$ . Позначення напівсуматора та схема на логічних елементах показані на рис. 5.4.

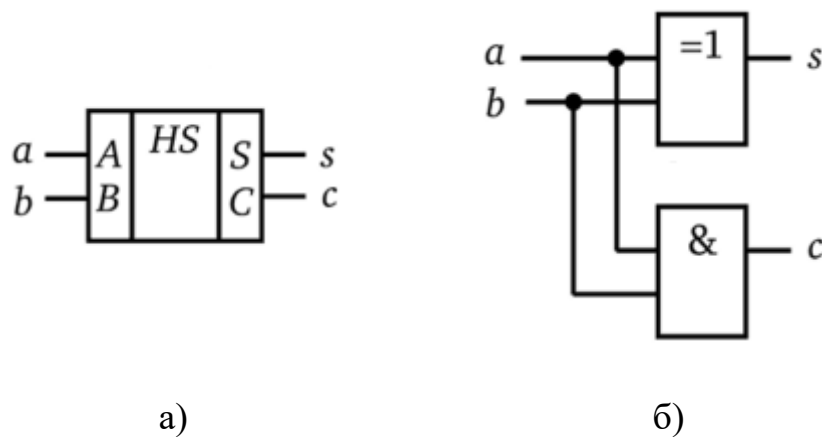


Рис. 5.4. Напівсуматор: а)Позначення; б) Схема на логічних елементах

Таблиця істинності буде мати наступний вигляд:

Таблиця 5.3. Таблиця істинності напівсуматора

| Входи |   | Виходи |   |
|-------|---|--------|---|
| A     | B | c      | s |
| 0     | 0 | 0      | 0 |
| 0     | 1 | 0      | 1 |
| 1     | 0 | 0      | 1 |
| 1     | 1 | 1      | 0 |



Як було сказано вище, повний суматор складається із двох напів суматорів і має вхід переносу із молодшого розряду. Схема такого пристрою показана на рис. 5.5.

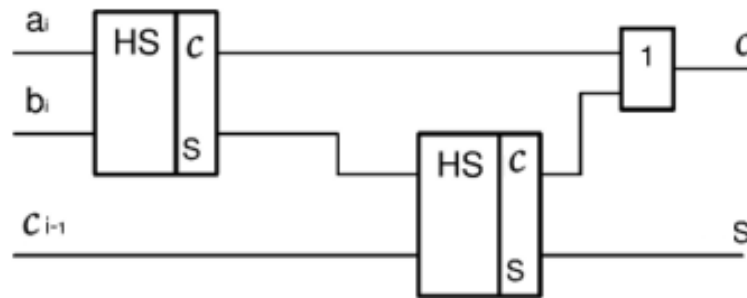


Рис. 5.5. Схема повного суматора

Таблиця істинності для такого пристрою буде виглядати наступним чином:

Таблиця 5.4. Таблиця істинності повного суматора

| Входи |   |          | Виходи |   |
|-------|---|----------|--------|---|
| A     | B | $C_{-1}$ | C      | S |
| 0     | 0 | 0        | 0      | 0 |
| 0     | 1 | 0        | 0      | 1 |
| 1     | 0 | 0        | 0      | 1 |
| 1     | 1 | 0        | 1      | 0 |
| 0     | 0 | 1        | 0      | 1 |
| 0     | 1 | 1        | 1      | 0 |
| 1     | 0 | 1        | 1      | 0 |
| 1     | 1 | 1        | 1      | 1 |

Для збільшення розрядності суматора послідовно об'єднують декілька однорозрядних повних суматорів, приклад такого об'єднання показано на рис. 5.6.

Прикладом мікросхеми такого суматора є sn74s283 - це двійковий чотирирозрядний повний суматор із швидким переносом. Схема повторює структуру, що наведена на рис. 5.6. Більш детальний опис sn74s283 наведено у технічній документації.

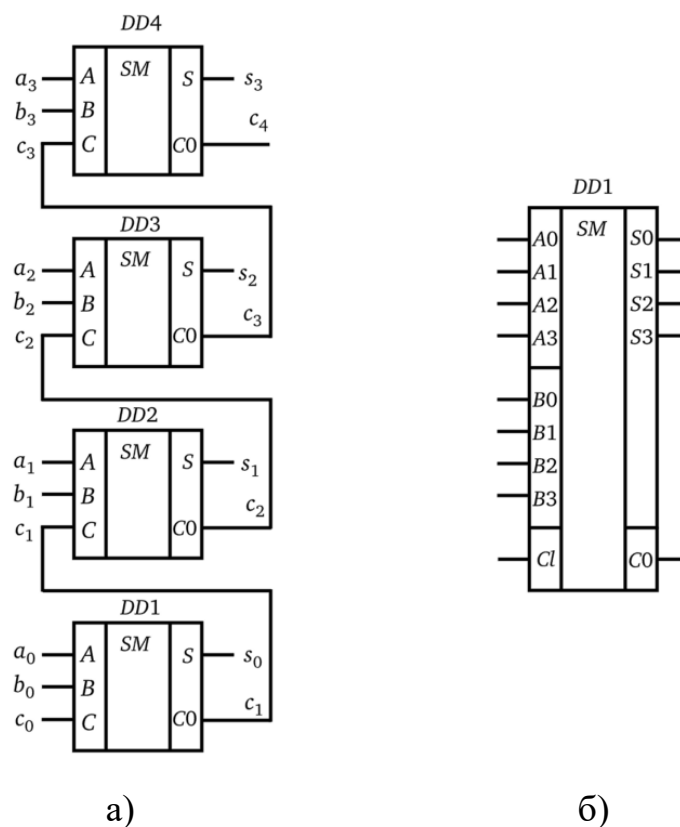


Рис. 5.6. Багаторозрядний суматор: а) Схема формування суматора із однобітних повних суматорів; б) Позначення

## 5.2. Завдання для виконання

*Для непарних варіантів.* У середовищі MicroCAP побудувати однорозрядний та трирозрядний компаратори, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

У бібліотеці моделей MicroCAP знайти модель двійкового компаратора, провести моделювання схеми, впевнитись правильності підключення, сформувати таблицю істинності.

*Для парних варіантів.* У середовищі MicroCAP побудувати напівсуматор та повний суматор, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

У бібліотеці моделей MicroCAP знайти модель повного суматора, провести моделювання схеми, впевнитись правильності підключення, сформувати таблицю істинності.

Використовуючи макетні плати та набір мікросхем логіки, побудувати пристрій по своєму варіанту – однобітний компаратор або напівсуматор. Впевнитись у його працездатності.

При відсутності необхідного елемента в наборі мікросхем логіки, замінити на наявний, використовуючи закони бульової алгебри для заміни.

Додаткове завдання. Зібрати таку ж схему з використанням мікросхем sn74s283, 74HC85 (або іншого, з наявного набору, узгодивши із викладачем), впевнитись в її працездатності.

### **5.3. Зміст звіту**

1. Схеми одно та багаторозрядного компаратора, або напів та повного суматора згідно свого варіанту.
2. Діаграми, що підтверджують працездатність схеми.
3. Таблиця істинності для схеми свого варіанту.
4. Фото зібраної схеми на макетній платі за своїм варіантом.
5. Висновки по роботі.

### **5.4. Контрольні запитання**

1. Які функції виконує двійковий компаратор?
2. Як збільшити розрядність компаратора?
3. На яких елементах можна побудувати однорозрядний компаратор?
4. У чому відмінність аналогового і цифрового компаратора?
5. Які функції виконує суматора та напів суматор, в чому їхня відмінність?
6. Навіщо потрібен вхід  $C_{-1}$ ?
7. Як збільшити розрядність суматора?

## КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 6

### ТРИГЕРИ ТА РЕГІСТРИ

**Мета роботи:** ознайомитись із принципами роботи тригерів та регістрів на їх базі, дослідити та підтвердити працездатність таких пристроїв.

#### 6.1. Теоретичні відомості

##### 6.1.1. Тригери

Тригер (англ. trigger) - електронний пристрій, що має два стійкі стани і здатний стрибком переходити з одного стану в інший під впливом зовнішнього сигналу. Тригерами або точніше тригерними системами називають великий клас електронних пристроїв, що мають здатність довго перебувати в одному з двох стійких станів і чергувати їх під впливом зовнішніх імпульсів.

Кожному стану тригера відповідає певний (високий або низький) рівень вихідної напруги:

- тригер встановлено в одиничний стан – рівень «1»;
- тригер скинутий у нуль – рівень «0» на виході.

Стан, що встановився, зберігається як завгодно довго і може бути змінено зовнішнім імпульсом або відключенням напруги живлення. Таким чином, на відміну від комбінаційних мікросхем, вихідні данні яких однозначно визначені при відомих входних впливах, у тригера з'являється властивість зберігати елементарну одиницю інформації (1 біт). Тобто, тригер – є елементарним елементом пам'яті.

Тригери можуть бути побудовані на дискретних елементах, логічних елементах, інтегральних мікросхемах.

До основних типів тригерів відносять: RS-, D-, T- та JK-тригери. Найпростішим із тригерів вважається RS-тригер. Його позначення на схемі та реалізація на базових логічних елементах показана на рис.6.1.

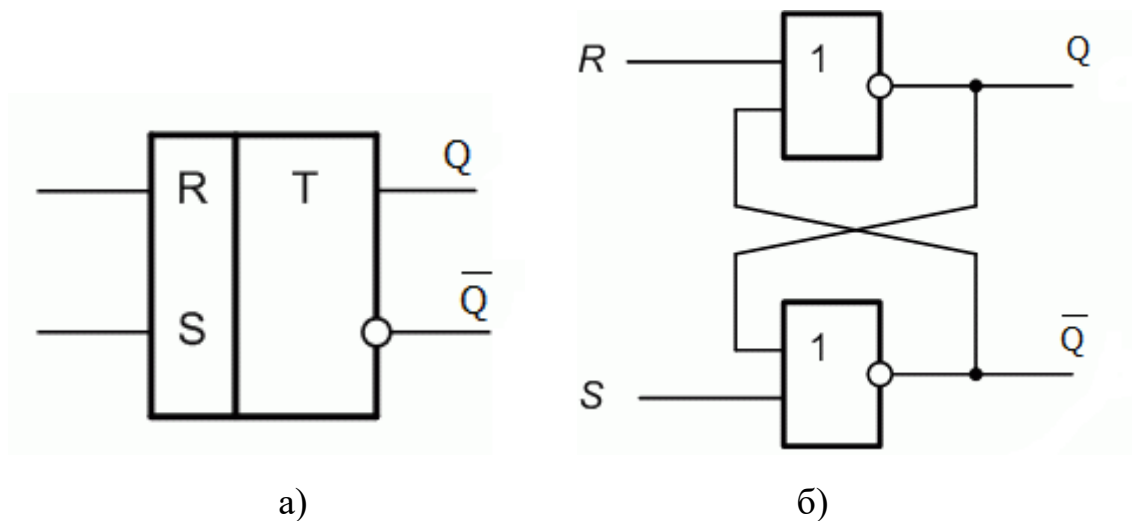


Рис. 6.1. RS-тригер: а) Умовне графічне позначення на схемах; б) Схема на базових логічних елементах

RS-тригер отримав назву за назвою своїх входів. Вхід S ( від Set – встановити (англ.)) дозволяє встановлювати вихід тригера Q в одиничний стан. Вхід R (від Reset – скинути) – дозволяє скидати вихід тригера Q (Quit) у нульовий стан. При одночасній подачі на входи значень логічного 0 тригер переходить у режим збереження і утримує вихідне значення, що визначалось попереднім станом входів. Вихід  $\bar{Q}$  є інверсним до Q. При подачі на входи R та S логічних одиниць — на виходах Q та  $\bar{Q}$  з'являться нулі, що протирічить логіці роботи виходів (прямий та інверсний). Тому такий стан умовно вважається забороненим. Таблиця істинності RS-тригера показана в таблиці 6.1.

Таблиця 6.1. Таблиця істинності RS-тригера

| Входи |   | Виходи                      |                      |
|-------|---|-----------------------------|----------------------|
| S     | R | Q                           | $\bar{Q}$            |
| 0     | 0 | Зберігання інформації       |                      |
| 0     | 1 | 0                           | 1 (скидання тригера) |
| 1     | 0 | 1 (установка тригера)       | 0                    |
| 1     | 1 | Заборонений(нестійкий) стан |                      |

Необхідно відмітити, що роботу всіх елементів з властивостями пам'яті зручно досліджувати за допомогою часових діаграм. Приклад такої діаграми наведено на рис. 6.2.

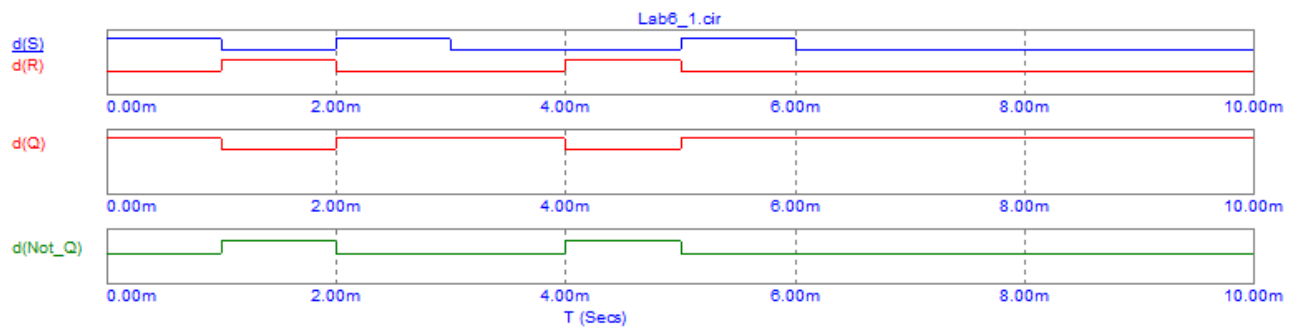


Рис. 6.2. Часова діаграма для дослідження RS-тригера у MicroCAP

Проаналізувавши діаграму, видно, що при встановленні логічних нулів на входи R та S тригер утримує попередні вихідні значення (інтервали [3м;4м] та [6м;10м]).

Наступною модифікацією є синхронний RS-тригер, який дозволяє спрацювання лише при заданих значеннях тактового сигналу. Умовне графічне позначення і схема на логічних елементах показані на рис. 6.3.

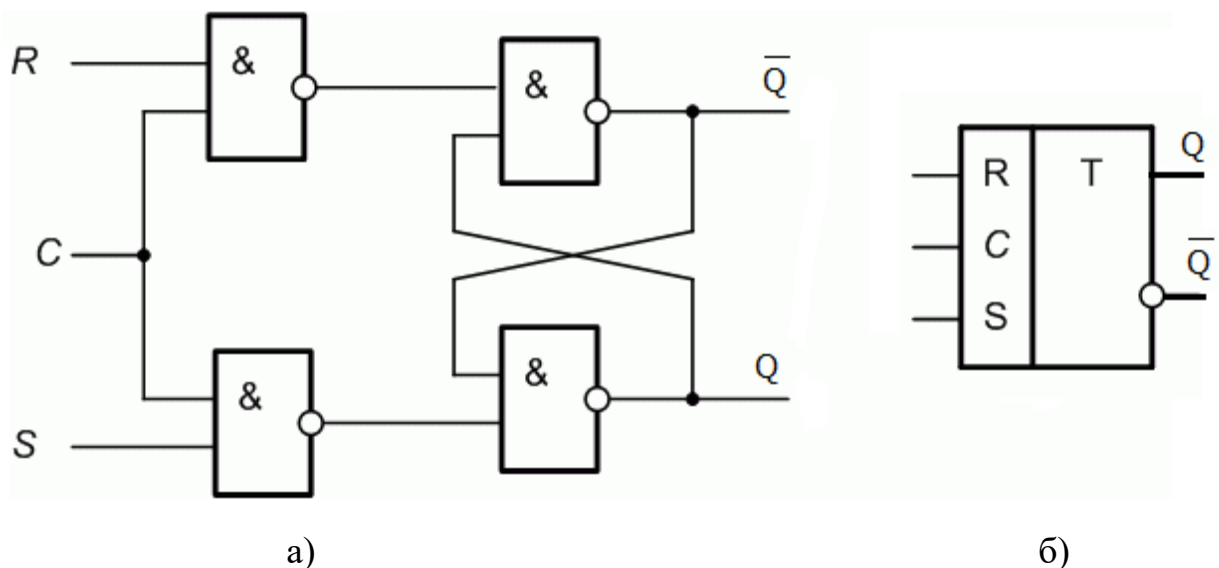


Рис. 6.3. Синхронний RS-тригер: а) Схема на базових логічних елементах.; б) Умовне графічне позначення на схемах

Перевагою використання синхронного RS-тригера є можливість керування моментом спрацювання тригера. Момент спрацювання визначається рівнем тактового сигналу C, а не моментом приходу сигналів на входи R або S, крім того, така схема є більш стійкою до зовнішніх збурень, наводок і випадкових спрацювань на лініях RS. Таблиця істинності синхронного RS-тригера має наступний вигляд:

Таблиця 6.2. Таблиця істинності синхронного RS-тригера

| Входи |   |   | Виходи                                    |                      |
|-------|---|---|-------------------------------------------|----------------------|
| S     | R | C | Q                                         | $\bar{Q}$            |
| 0     | 0 | 1 | З б е р і г а н н я   і н ф о р м а ц і ї |                      |
| 0     | 1 | 1 | 0                                         | 1 (скидання тригера) |
| 1     | 0 | 1 | 1 (установка тригера)                     | 0                    |
| 1     | 1 | 1 | Заборонений(нестійкий) стан               |                      |
| x     | x | 0 | З б е р і г а н н я   і н ф о р м а ц і ї |                      |

Відповідно, з таблиці істинності видно, що при значенні тактових сигналів C, що дорівнюють 1, синхронний RS-тригер працює як звичайний (статичний): при приході логічного нуля на вхід C, тригер переходить у режим збереження інформації. Зазвичай, частота тактових сигналів у декілька разів вища частоти зміни сигналів на входах R та S.

Для усунення нестійкого стану, коли на два входи приходять логічні одиниці, використовується D-тригер. Умовне графічне позначення і схема на логічних елементах показані на рис. 6.4.

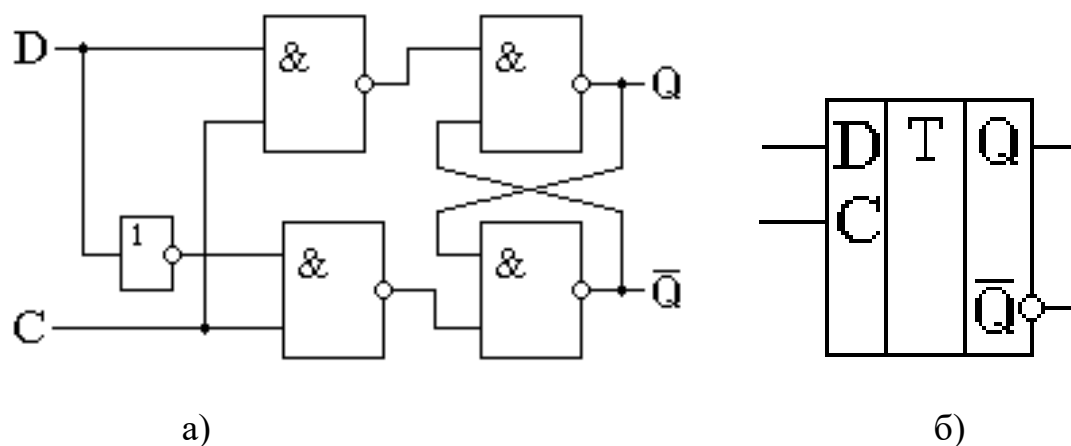


Рис. 6.4. D-тригер: а) Схема на базових логічних елементах.; б) Умовне графічне позначення на схемах

По суті, такий тригер є модифікованим синхронним RS-тригером, у якого ходи R та S узгоджені через інвертор. Таким чином на цих входах фізично відсутня можливість приходу однакових значень.

Таблиця істинності D-тригера представлена в таблиці 6.3.

Таблиця 6.3. Таблиця істинності D-тригера

| Входи |   | Виходи                |           |
|-------|---|-----------------------|-----------|
| С     | D | Q                     | $\bar{Q}$ |
| 0     | 0 | Зберігання інформації |           |
| 0     | 1 | Зберігання інформації |           |
| 1     | 0 | 0                     | 1         |
| 1     | 1 | 1                     | 0         |

### 6.1.2. Регістри

У сучасній цифровій схемотехніці та електроніці часто необхідно зберігати більше ніж один біт інформації, тому тригери об'єднуються за певним принципом, формуючи пристрої, що отримали назву регістрів. Для безпосереднього збереження інформації тригери (зазвичай D-тригери) об'єднуються паралельно. Такий пристрій має N виходів та N інформаційних входів і один тактовий. Принципова схема і графічне позначення паралельного регістра показано на рис. 6.5.

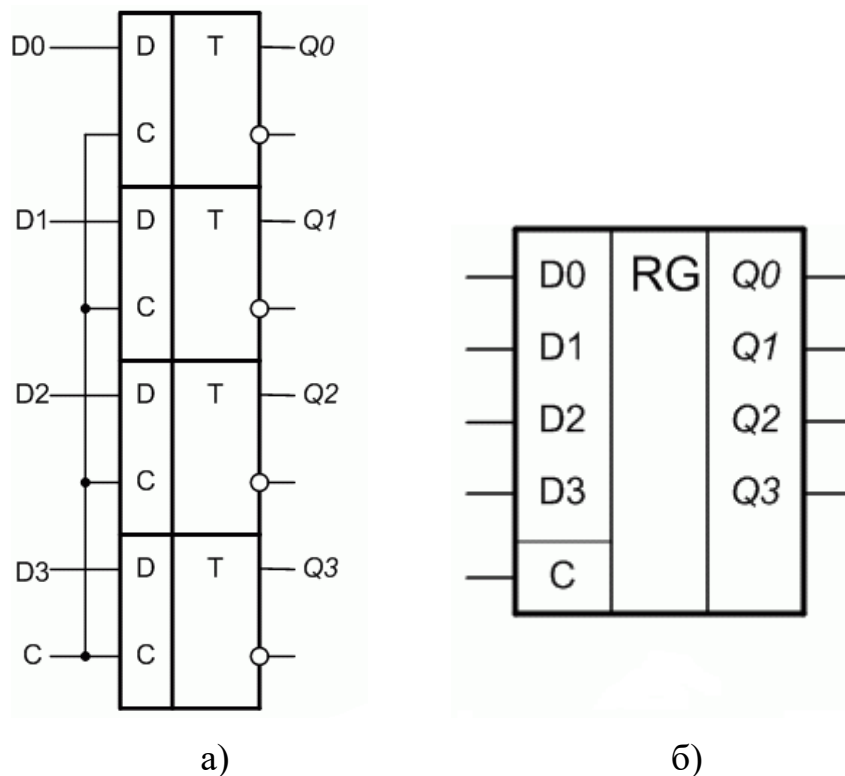


Рис. 6.5. Паралельний регістр: а) Принципова схема на D-тригерах.; б) Умовне графічне позначення мікросхеми



Принцип збереження інформації у такому регістрі повністю співпадає із принципом функціонування D-тригера.

При послідовному з'єднанні тригерів такі конструкції регістрів можуть мати різноманітні функції. Оскільки інформація буде передаватися послідовно із тригера в тригер, то такі регістри отримали назву зсувні, або регістри зсуву. Вони використовуються для перетворення паралельного коду в послідовний, послідовного в паралельний і т. ін., що часто використовується для побудови інтерфейсів зв'язку.

Приклад реалізації зсувного регістра для перетворення послідовного коду в паралельний показано на рис. 6.6.

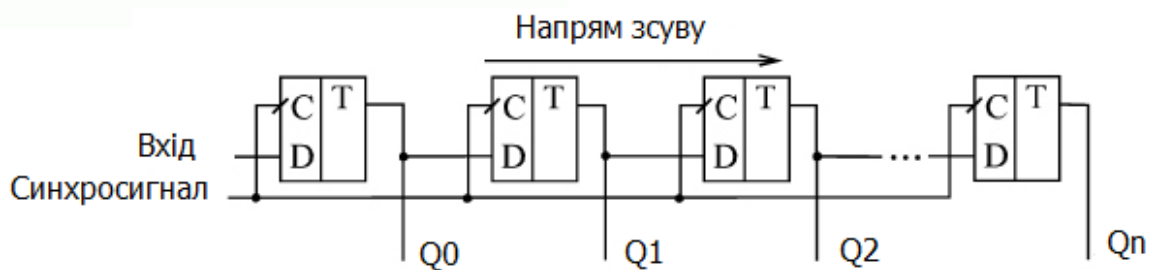


Рис. 6.6. Приклад реалізації зсувного регістра для перетворення послідовного коду в паралельний

Прикладом такого пристрою є мікросхема 8-розрядного зсувного регістра 74HC595N, загальний вигляд якої і позначення виводів показані на рис. 6.7.

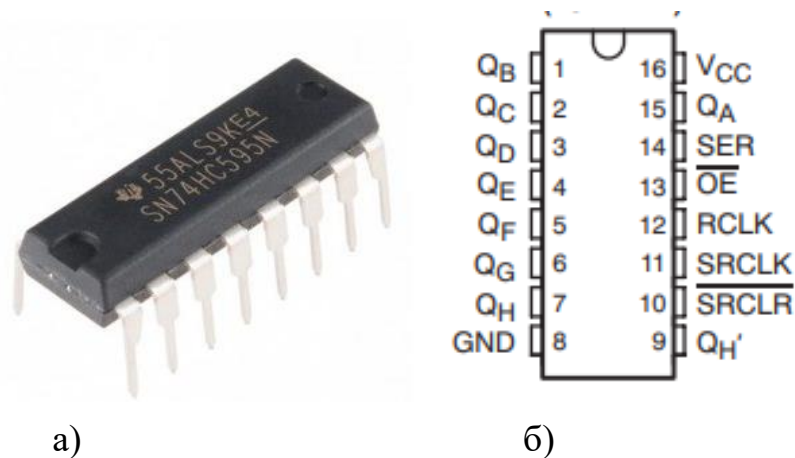


Рис. 6.7. Регістр зсуву 74HC595N: а) Загальний вигляд; б) Позначення виводів

Для більш детального освоєння пропонується ознайомитись із технічною документацією, що надається в додаткових матеріалах.

## **6.2. Завдання для виконання**

Завдання 1. Для непарних варіантів. Використовуючи моделі логічних елементів, у середовищі MicroCAP побудувати схему синхронного RS-тригера, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

У бібліотеці моделей MicroCAP знайти модель RS-тригера (наприклад Generic SR Latch), провести моделювання схеми, впевнитись правильності підключення, сформувані таблицю істинності. Порівняти із результатами отриманими при попередньому моделюванні.

Для парних варіантів. Використовуючи моделі логічних елементів, у середовищі MicroCAP побудувати схему D-тригера, провести моделювання роботи схеми та впевнитись в її працездатності, скласти таблицю істинності.

У бібліотеці моделей MicroCAP знайти модель D-тригера (наприклад Generic D Latch), провести моделювання схеми, впевнитись правильності підключення, сформувані таблицю істинності. Порівняти із результатами отриманими при попередньому моделюванні.

Завдання 2. Використовуючи макетні плати та набір мікросхем логіки, побудувати пристрій по своєму варіанту – RS або D-тригер. Впевнитись в його працездатності.

Завдання 3. Обравши із таблиці 6.4. модель регістра, провести його моделювання, переконавшись в правильності його функціонування та підключення.

Додаткове завдання. Зібрати схему з використанням мікросхеми 8-розрядного зсувного регістра 74HC595N (або іншого, з наявного набору, узгодивши із викладачем), попередньо ознайомившись із технічною документацією на мікросхему, впевнитись в коректності підключення та функціонування.

Таблиця 6.3. Таблиця варіантів завдань

| № варіанту | Модель регістру |
|------------|-----------------|
| 1          | 74AC378         |
| 2          | 74ALS165        |
| 3          | 74F166          |
| 4          | 74ACT164        |
| 5          | 74LS165A        |
| 6          | 74165           |
| 7          | 74166           |
| 8          | 74HCT165        |
| 9          | 74ACT164        |
| 10         | 74F166A         |

Якщо варіант більше ніж 10, то починаємо спочатку, тобто 11й варіант – знову 1 в таблиці, 12-2, 13-3 т.д.

### 6.3. Зміст звіту

1. Схеми тригерів за своїм варіантом.
2. Діаграми, що підтверджують працездатність схеми, таблиця істинності для схеми свого варіанту.
3. Схема із моделлю тригера. Діаграми, що підтверджують працездатність схеми, таблиця істинності.
4. Схема із моделлю регістра за номером варіанту. Діаграми, що підтверджують працездатність схеми.
5. Фото зібраної схеми на макетній платі.
6. Висновки по роботі.

### 6.4. Контрольні запитання

1. Що таке двійковий тригер, які його функції?
2. Дайте класифікацію тригерів.
3. Які недоліки та переваги є у RS-тригера?
4. У чому відмінність синхронного та асинхронного RS-тригерів?
5. Чим відрізняється D-тригер від RS-тригерів? Які його функції?
6. Які ви ще знаєте види тригерів, в чому їх відмінність?
7. Що таке регістр? Які функції він виконує?

## 8. Які є різновиди реєстрів?

9. Для чого використовуються реєстри зсуву? Як вони реалізовані?

Відповідь поясніть на прикладах.

10. Які особливості використання мікросхеми 74НС595N?

**ДЛЯ НОТАТОК**

This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Победаш, К. К. Комп'ютерна електроніка [Електронний ресурс] : навч. посіб. для студ. спеціальності 151 "Автоматизація та комп'ютерно-інтегровані технології", спеціалізації "Комп'ютерно-інтегровані оптико-електронні системи та технології"/ К. К. Победаш ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1файл: 21,4 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2019. – 364 с.
2. Клен, К. С. Схемотехніка: Частина 2. Цифрова схемотехніка: Практикум [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітніми програмами «Електронні компоненти і системи» та «Електронні прилади та пристрої» спеціальності 171 Електроніка / К. С. Клен, П. С. Сафронов ; КПІ ім. Ігоря Сікорського.– Київ : КПІ ім. Ігоря Сікорського, 2021. – 67 с
3. Micro-Cap 10.0 Electronic Circuit Analysis Program Reference Manual  
Copyright 1982-2012 by Spectrum Software 1021 South Wolfe Road Sunnyvale,  
CA 94086