

М. О. Притула, І. А. Дудатьєв, Я. О. Осадчук

ЦИФРОВІ ПРИСТРОЇ

Міністерство освіти і науки України
Вінницький національний технічний університет

М. О. Притула, І. А. Дудатьєв, Я. О. Осадчук

ЦИФРОВІ ПРИСТРОЇ

Електронний лабораторний практикум

Вінниця
ВНТУ
2026

УДК 621.391.8

П77

Рекомендовано до видання Вченою Радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 14 від 26.06.2025 р.)

Рецензенти:

С. К. Підченко, доктор технічних наук, професор

Д. В. Михалевський, доктор технічних наук, професор

Д. Х. Штофель, кандидат технічних наук, доцент

Притула, М. О.

П77 Цифрові пристрої : лабораторний практикум [Електронний ресурс] / Притула М. О., Дудатьєв І. А., Осадчук Я. О. – Вінниця : ВНТУ, 2026. – (PDF, 124 с.)

Лабораторний практикум присвячений матеріалам курсу з дисципліни «Цифрові пристрої в радіотехнічних системах» для здобувачів, що навчаються за спеціальністю «Електроніка, електронні комунікації, приладобудування та радіотехніка» денної та заочної форм навчання.

Мета лабораторного практикуму – надати здобувачам можливість більш детально вивчити аудиторний матеріал і підготуватися до іспиту, а також застосувати отримані знання для подальшої фахової роботи.

Перелік та зміст тем відповідає програмі вказаної вище дисципліни.

УДК 621.391.8

© ВНТУ, 2026

Зміст

Лабораторна робота 1	
Логічні елементи	4
Лабораторна робота 2	
Тригери.....	16
Лабораторна робота 3	
Лічильники.....	32
Лабораторна робота 4	
Шифратори та дешифратори	47
Лабораторна робота 5	
Мультиплексори та демюльтиплексори	60
Лабораторна робота 6	
Регістри.....	76
Лабораторна робота 7	
Аналого-цифрові перетворювачі.....	91
Лабораторна робота 8	
Двійкові суматори	108
ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ.....	123

Лабораторна робота 1

Логічні елементи

Основні теоретичні відомості

Логічні елементи є основними будівельними блоками цифрових пристроїв і виконують операції над двійковими даними. Вони реалізують логічні функції, які визначаються булевою алгеброю, і використовуються для обробки, зберігання та передачі інформації [1].

Логічні елементи є основою для побудови складніших цифрових пристроїв, таких як:

- комбінаційні схеми (мультиплексори, демультиплексори, шифратори, дешифратори);
- послідовнісні схеми (тригери, регістри, лічильники);
- арифметико-логічні пристрої (суматори, віднімачі);
- запам'ятовувальні пристрої;
- мікропроцесори та мікроконтролери.

Сигнали на входах цифрового пристрою в будь-який момент часу формують певне двійкове число, а вихідні сигнали залежать від цього набору згідно з певними правилами. Таким чином, пристрій функціонує як перетворювач цифрової інформації: вхідні сигнали являють собою логічні (булеві, двійкові) змінні x_i , а вихідні – логічні (булеві, перемикальні) функції:

$$y_j = f_j(x_1, x_2, \dots, x_m).$$

Логічні змінні та функції, внаслідок виконання певних операцій, можуть набувати одного з двох інформаційних значень: логічний 0 або логічна 1. З часом значення змінних, а відповідно і функцій, змінюються. Якщо якийсь біт інформації, наприклад, на вході x_i , залишається сталим, його називають константою нуля або одиниці, позначаючи знаками тотожності: $x_i \in 0$ або $x_i \in 1$. Якщо два значення змінюються майже одночасно і завжди є протилежними, тобто вони парафазно відображають один і той самий сигнал, то одне з них називають прямим x_i , а друге – інверсним $\overline{x_i}$.

Абстрагуючись від фізичних процесів у схемах та їх елементної бази, зручніше розглядати функціонування цифрових пристроїв та моделювання інформаційних перетворень за допомогою булевої алгебри. Для цього важливо визначити залежність кожного вихідного сигналу від вхідних значень, тобто задати його логічну функцію. Основним способом є табличний метод, з якого можна отримати логічний (булевий) вираз.

Найчастіше використовується таблиця відповідності (істинності), що зручно побудована: кожна змінна та функція розташовані в колонках, а

змінні формують рядки – вхідні кортежі (набори вхідних змінних), які мають двійкові значення. Якщо змінні впорядкувати у природному порядку, як двійкові числа – від старшого до молодшого розряду, вони займатимуть сталу позицію незалежно від їх кількості на графічній моделі. Для зручності посилань десятковий код вхідного кортежу часто позначається літерою. [2]

З усіх $k=2^m$ кортежів – можливих наборів змінних від 0 до $2^m - 1$ (де m – кількість змінних) – можна утворити 2^k різних функцій. Деякі з функцій є виродженими – це константи нуля чи одиниці, а також такі, що повторюють одну зі змінних або функцію з меншою кількістю аргументів. Інші функції не є самостійними, якщо вони відрізняються лише порядком змінних. У подальшому розглядаються лише дві функції з однією змінною (одномісні операції) та чотири пари оригінальних функцій з двома змінними (двомісні або бінарні операції). Функції з більшою кількістю аргументів можуть бути зведені до трьох із цих пар.

У таблиці 1.1 всі функції подано парами: у лівих вихідних колонках таблиці відповідності та умовних графічних позначеннях наведено пряму функцію (скорочені назви англійською мовою в дужках, які використовуються під час програмування) y_0 , а в правих – інверсну y_1 .


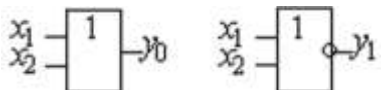
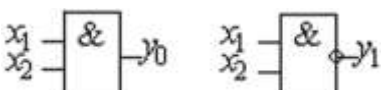
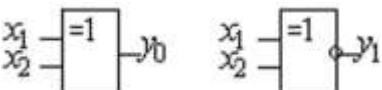
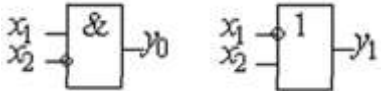
Лише одна з наведених операцій – функція повторення (також відома як твердження) $y_0=x_1$ (читається: як x_1) – є тривіальною. Проте повторювач або буферний підсилювач, що реалізує цю операцію, часто використовується для збільшення навантажувальної здатності цифрових пристроїв і їх з'єднання з шинами. Іншою одномісною операцією є функція НЕ (логічне заперечення, інверсія, доповнення), яка означає обернення аргументу x_1 , тобто зміну його логічного значення на протилежне. Інверсія позначається у формулах рискою над змінними: $y_1=\overline{x_1}$ (читається: не x_1), а на умовних графічних позначеннях інвертора (елемента НЕ) та інших елементів – маленьким колом.

Функція АБО (логічне додавання, диз'юнкція, об'єднання) є процесом об'єднання інформації з кількох джерел в один канал: вихід y_0 набуває значення логічної 1, якщо хоча б один з входів активний. Логічне додавання має відмінність від арифметичного в останньому рядку таблиці, де $1+1=1$. Воно позначається як $y_0=x_1+x_2=x_1 \vee x_2$ (читається: x_1 або x_2) і реалізується за допомогою елемента АБО (диз'юнктора). Якщо додати інверсію на виході, отримуємо функцію АБО–НЕ (логічне додавання з запереченням, стрілка Пірса, заперечення диз'юнкції):

$$y_1=\overline{x_1 + x_2} = \overline{x_1 \vee x_2}$$

(читається: ні x_1 , ані x_2), що виконується елементом АБО–НЕ (елементом Пірса).

Таблиця 1.1 – Основні логічні функції

Логічна функція		Таблиця відповідності		Умовне графічне позначення		
Назва	Позначення	x_2	x_1	y_0 y_1		
Повторення	$y_0 = x_1$	0	0	1		
НЕ (NOT)	$y_1 = \bar{x}_1$	1	1	0		
АБО (OR)	$y_0 = x_1 + x_2$	0	0	0	1	
АБО-НЕ (NOR)	$y_1 = \overline{x_1 + x_2}$	0	1	1	0	
		1	0	1	0	
І (AND)	$y_0 = x_1 x_2$	0	0	0	1	
І-НЕ (NAND)	$y_1 = \overline{x_1 x_2}$	0	1	0	1	
		1	0	0	1	
Виключне АБО (XOR)	$y_0 = x_1 \oplus x_2$	0	0	0	1	
Виключне АБО-НЕ (XNOR)	$y_1 = \overline{x_1 \oplus x_2}$	0	1	1	0	
		1	0	1	0	
Заборона	$y_0 = x_1 \setminus x_2$	0	0	0	1	
Імплікація	$y_1 = x_1 \rightarrow x_2$	0	1	1	0	
		1	0	0	1	

Функція І (логічне множення, добуток, кон'юнкція, перетин) виконує операцію визначення збігу інформації на всіх входах. Вона є аналогом двійкового множення одноцифрових чисел: набуває значення логічної 1, якщо високий рівень (логічна 1) присутній на всіх входах. У виразах ця функція позначається $y_0 = x_1 x_2 = x_1 \cap x_2$ (читається: x_1 і x_2), а на умовних позначеннях – знаком & (амперсанд, and). Симетричною до неї є функція І-НЕ (логічне множення із запереченням), яка також відома як риска Шеффера. Ця функція виконує заперечення кон'юнкції, несумісність): $y_1 = \overline{x_1 x_2} = x_1 \cap x_2$ (читається: не x_1 або не x_2), реалізується елементом І-НЕ (елементом Шеффера). Логічні функції АБО, І та їх заперечення можуть бути реалізовані для будь-якої кількості змінних [3].

Функція виключне АБО (додавання за модулем 2, нерівнозначність, антиеквівалентність) відрізняється від функції АБО тим, що у разі додавання кожної пари одиниць результатом є нуль. Зазвичай операцію над багатьма змінними називають додаванням за модулем 2, тоді як для двох аргументів використовується термін виключне АБО. Цю функцію позначають символом псевдо-плюс: $y_0 = x_1 \oplus x_2 = \overline{x_1} x_2 + x_1 \overline{x_2}$ (читається: або x_1 або x_2 ; x_1 або x_2 виключно), а на графічних зображеннях для двовходових

елементів позначкою \oplus . Елемент, що реалізує цю функцію з будь-якою кількістю входів, зазвичай називають суматором за модулем 2, а двовходовий елемент – виключним АБО (елементом нерівнозначності). Суміжною функцією є виключне АБО–НЕ, що відповідає рівнозначності або еквівалентності: $y_1 = \overline{y_0} = \overline{x_1 \oplus x_2} = x_1 \sim x_2$ (читається: x_1 як x_2). Цю операцію виконує цифровий пристрій, відомий також як елемент виключне АБО–НЕ (елемент рівнозначності або еквівалентор).

Функція заборони (НЕ, заборона, заперечення імплікації, різниця) означає, що вхідна змінна x_1 передається на вихід лише тоді, коли заборона x_2 відсутня (тобто $x_2=0$); в іншому випадку вихід має значення логічного 0. Зазвичай цю функцію розглядають як логічне множення з інверсією одного з двох входів: $y_0 = x_1 \cdot \overline{x_2} = x_1 \cdot \overline{x_2}$ (читається: x_1 , але не x_2). Однак її також можна інтерпретувати як своєрідне логічне віднімання, де з x_1 вилучаються одиниці, що є в x_2 . Реалізація цієї функції здійснюється за допомогою елемента заборони (елемента НЕ або елемента І з інверсійним входом) [4].

Функція імплікація (розділення із заборonoю, селекція) отримала свою назву з логіки висловлень через зв'язок «якщо $x_1=1$, то $y_1=x_2$ » (в іншому випадку $y_1=1$). Вона часто розглядається як логічне додавання з інверсією одного з входів (або функція заборони з запереченням на виході): $(y_1 = x_1 \rightarrow x_2 = \overline{x_1} + x_2)$ (якщо x_1 , то x_2 ; не x_1 або x_2). Елемент, що реалізує цю функцію, називається імплікатором (селектором).

Набір логічних елементів називається функціонально повним, якщо за допомогою елементів цього набору можна реалізувати будь-яку логічну функцію. *Приклади функціонально повних наборів:*

- {I, АБО, НЕ}
- {I–НЕ}
- {АБО–НЕ}

Параметри логічних елементів

1. Швидкодія – визначається часом затримки поширення сигналу від входу до виходу.
2. Завадостійкість – здатність правильно функціонувати за наявності шумів.
3. Навантажувальна здатність – кількість аналогічних елементів, які можна підключити до виходу цього елемента.
4. Споживана потужність – потужність, яку споживає елемент у робочому режимі.

Реалізація логічних елементів

Логічні елементи можуть бути реалізовані різними технологіями:

- транзисторно-транзисторна логіка (ТТЛ);
- емітерно-зв'язана логіка (ЕЗЛ);
- комплементарна метал-оксид-напівпровідникова логіка (КМОП);

- інтегрально-інжекційна логіка (І²Л).

Використання логічних елементів

Моделювання логічних елементів

Для перевірки коректності роботи цифрових схем використовуються спеціалізовані програмні середовища для моделювання, такі як:

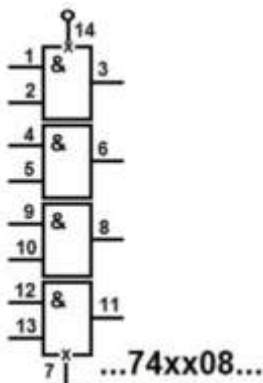
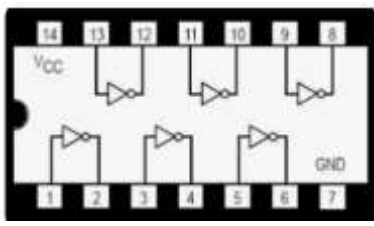
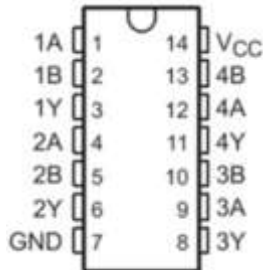
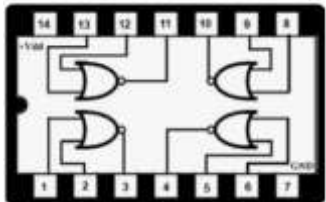
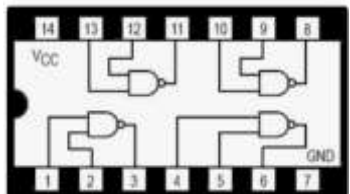
- Multisim;
- Proteus;
- LogiSim;
- ModelSim.

Ці інструменти дозволяють перевірити правильність функціонування схеми перед її фізичною реалізацією.

Порядок виконання роботи (off-line частина)

1. Вибрати мікросхеми (вітчизняні або зарубіжні) для реалізації свого варіанта завдання. Знайти в офіційній документації функціональне призначення виводів цих мікросхем. В цьому прикладі буде наведено роботу кількох різних логічних елементів, дані про які наведено в табл. 1.2.

Таблиця 1.2 – Дані про досліджувані логічні елементи

<p>КР1533ЛИ1: 4 елемента 2І</p> 	<p>К561ЛН2: 6 елементів НЕ</p> 
<p>КР1533ЛП5: 4 елемента ВИКЛ-АБО</p> 	<p>К561ЛЕ5А: 4 елемента 2АБО-НЕ</p> 
<p>КР1533ЛА3: 4 елемента 2І-НЕ</p> 	

2. Скласти на макетній платі схему на вибраних мікросхемах. Для зручності та простоти відображення інформації на входах та виходах підключити зелені та червоні світлодіоди послідовно з опором, відповідно. Результат показано на рис. 1.1.

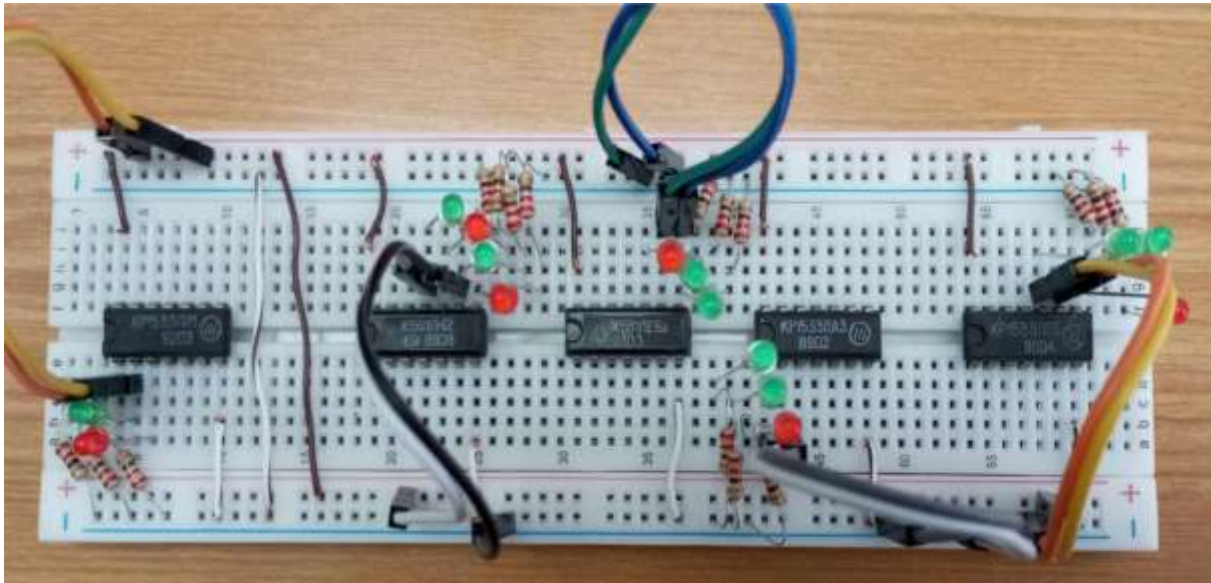


Рисунок 1.1 – Складена логіка на макетній платі

3. Подаючи на входи схем (перемикаючи провідники) логічний «0» (земля) або «1» (живлення +5 В) проводимо дослідження та заповнюємо таблицю відповідності досліджуваної функції. Результати дослідження для першого макета наведено на рис. 1.2., для другого – на рис. 1.3, для третього – на рис. 1.4, четвертого – рис. 1.5, для п'ятого – рис. 1.6.

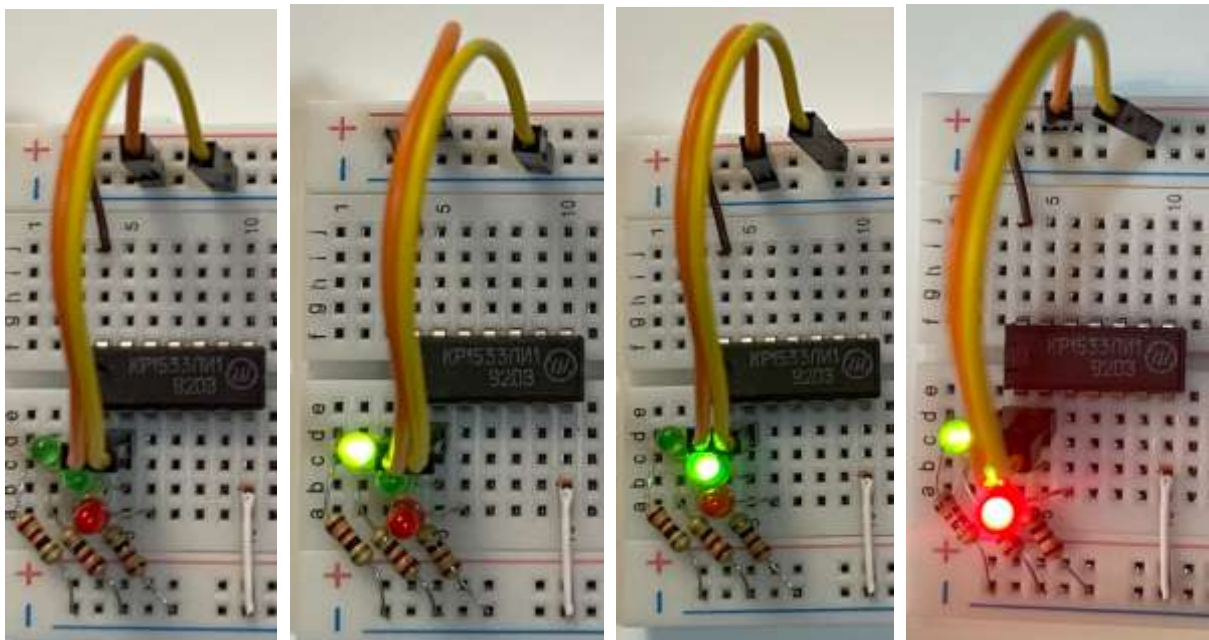


Рисунок 1.2 – Результати досліджень для макета 1 (КР1533ЛН1)

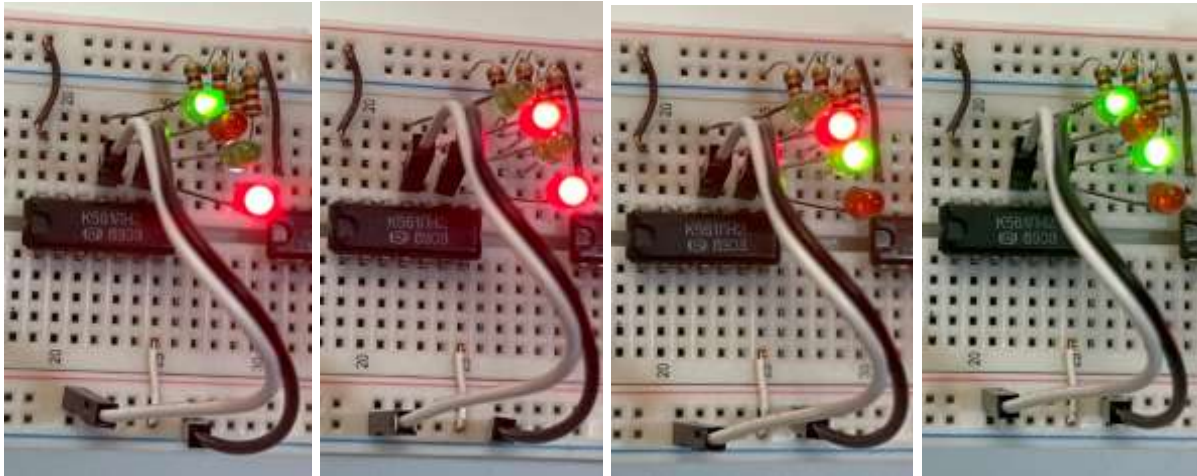


Рисунок 1.3 – Результати досліджень для макета 2 (K561LN2)

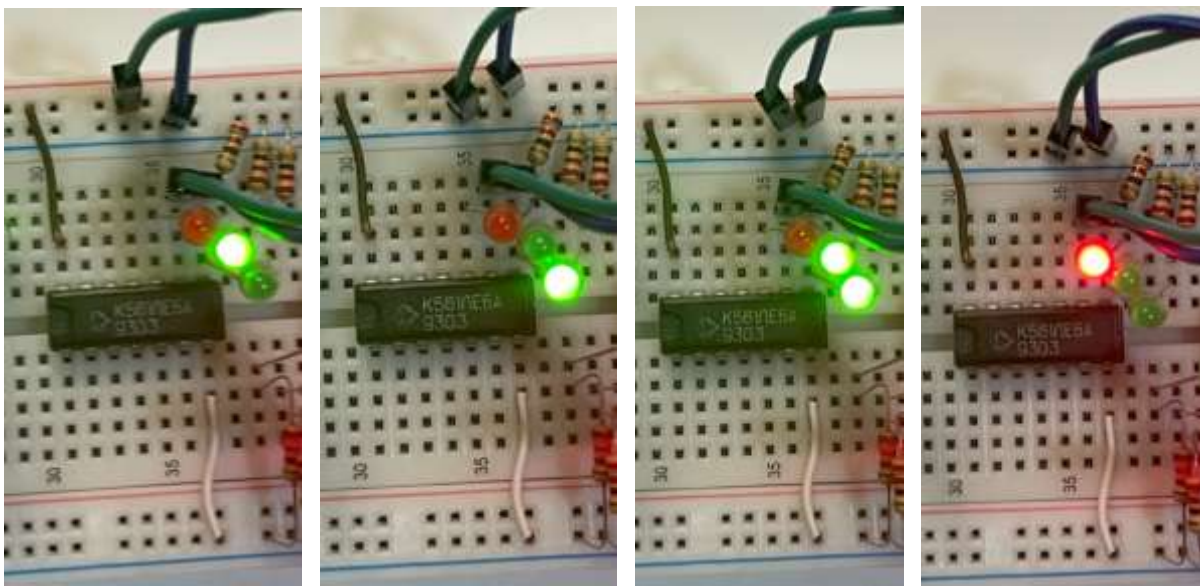


Рисунок 1.4 – Результати досліджень для макета 3 (K561LE5A)

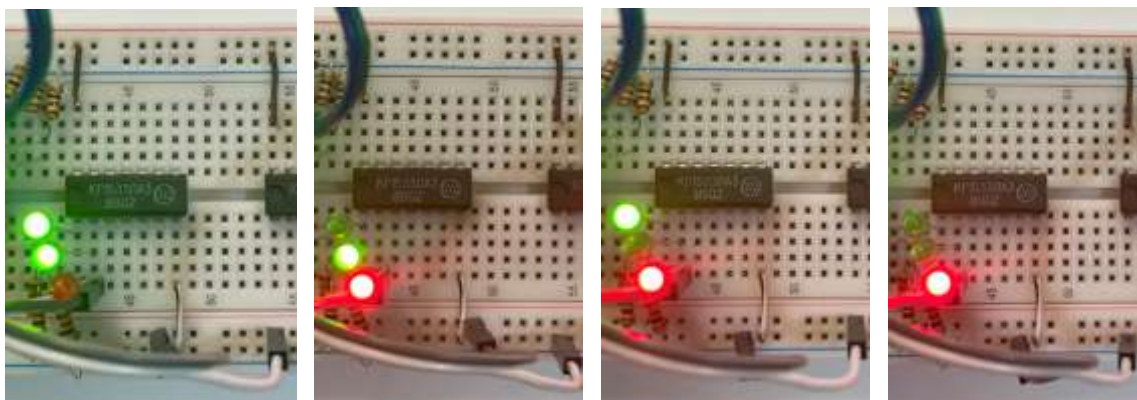


Рисунок 1.5 – Результати досліджень для макета 4 (KP1533LA3)

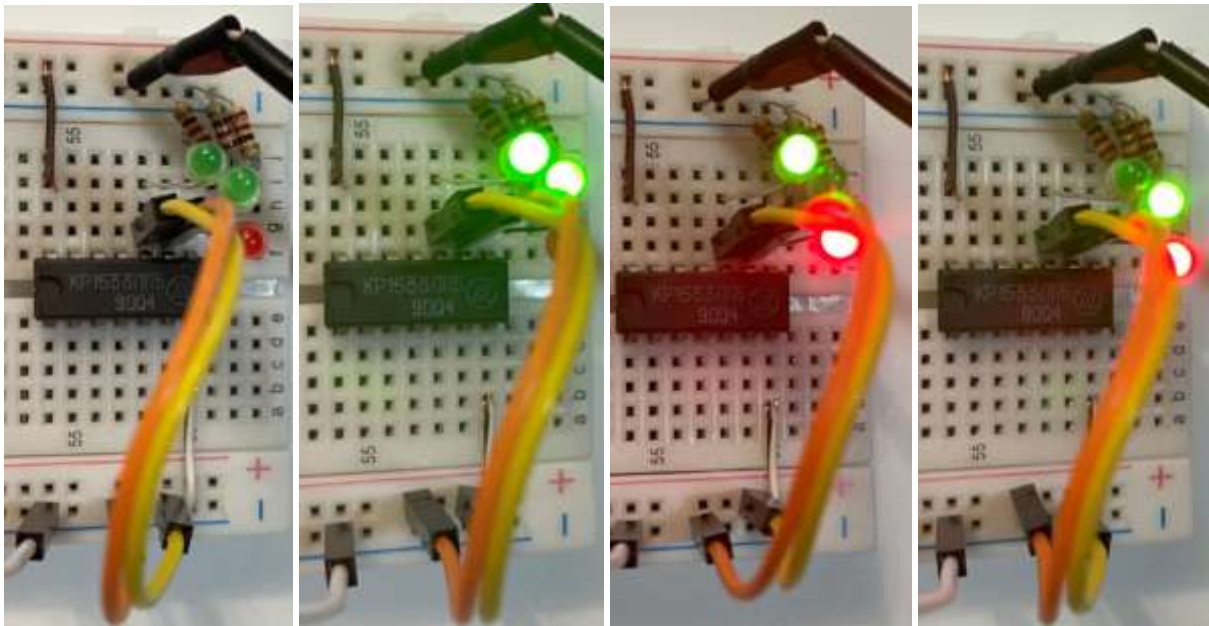


Рисунок 1.6 – Результати досліджень для макету 5 (KP1533ЛП5)

4. На основі результатів досліджень рис. 1.2 – рис. 1.6 складаємо таблицю відповідності для кожного макета. Результати наведено в табл. 1.3.

Таблиця 1.3 – Таблиці відповідностей для 5 макетів

KP1533ЛИ1			K561ЛН2			K561ЛЕ5А			KP1533ЛА3			KP1533ЛП5		
x_1	x_2	y	x_1	x_2	y	x_1	x_2	y	x_1	x_2	y	x_1	x_2	y
0	0	0	0	-	1	0	0	1	0	0	1	0	0	0
0	1	0	1	-	0	0	1	0	0	1	1	0	1	1
1	0	0	-	0	1	1	0	0	1	0	1	1	0	1
1	1	1	-	1	0	1	1	0	1	1	0	1	1	0

5. Проводячи аналіз результатів табл. 1.3, ми підтвердили та дослідили задані функції логічних елементів: KP1533ЛИ1 – елемент 2І, K561ЛН2 – елемент НЕ, K561ЛЕ5А – елемент 2АБО-НЕ, KP1533ЛА3 – елемент 2І-НЕ, KP1533ЛП5 – елемент ВИКЛ-АБО.

Порядок виконання роботи (on-line частина)

1. Для виконання лабораторної роботи в онлайн форматі можна використовувати різні моделювальні пакети. Обмежень щодо вибору немає, але перевага надається безкоштовним програмним пакетам. Ми будемо проводити лабораторну роботу в програмі Multisim.

2. Вибираємо схеми для дослідження.

Перша схема в нас буде елемент 3І-НЕ. Набираємо схему в моделювальному пакеті. Елементи можна брати ідеальними або конкретні мікросхеми. Обмежень щодо такого вибору немає. Результат наведено на рис. 1.7.

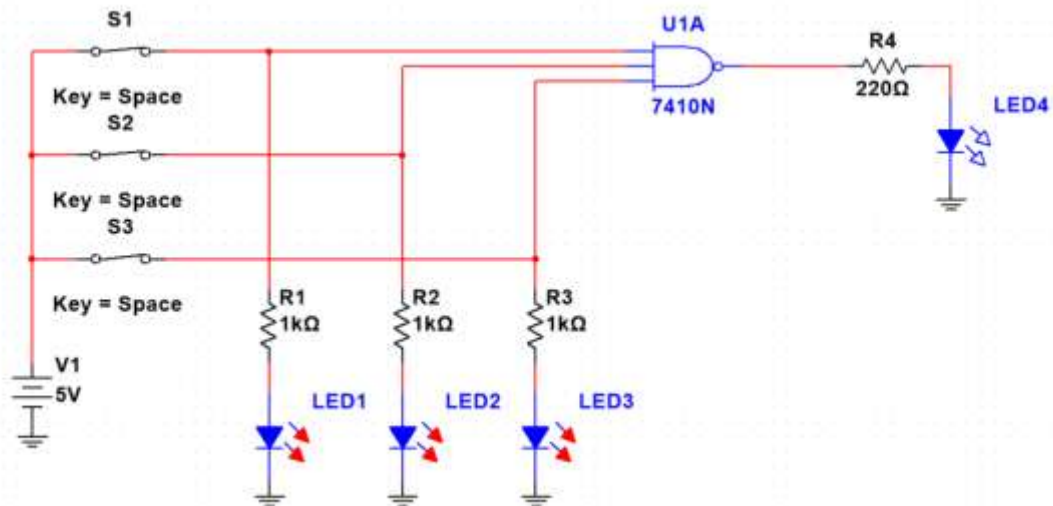


Рисунок 1.7 – Досліджувана схема 3І-НЕ в multisim

Результати дослідження цієї схеми наведено на рис. 1.8.

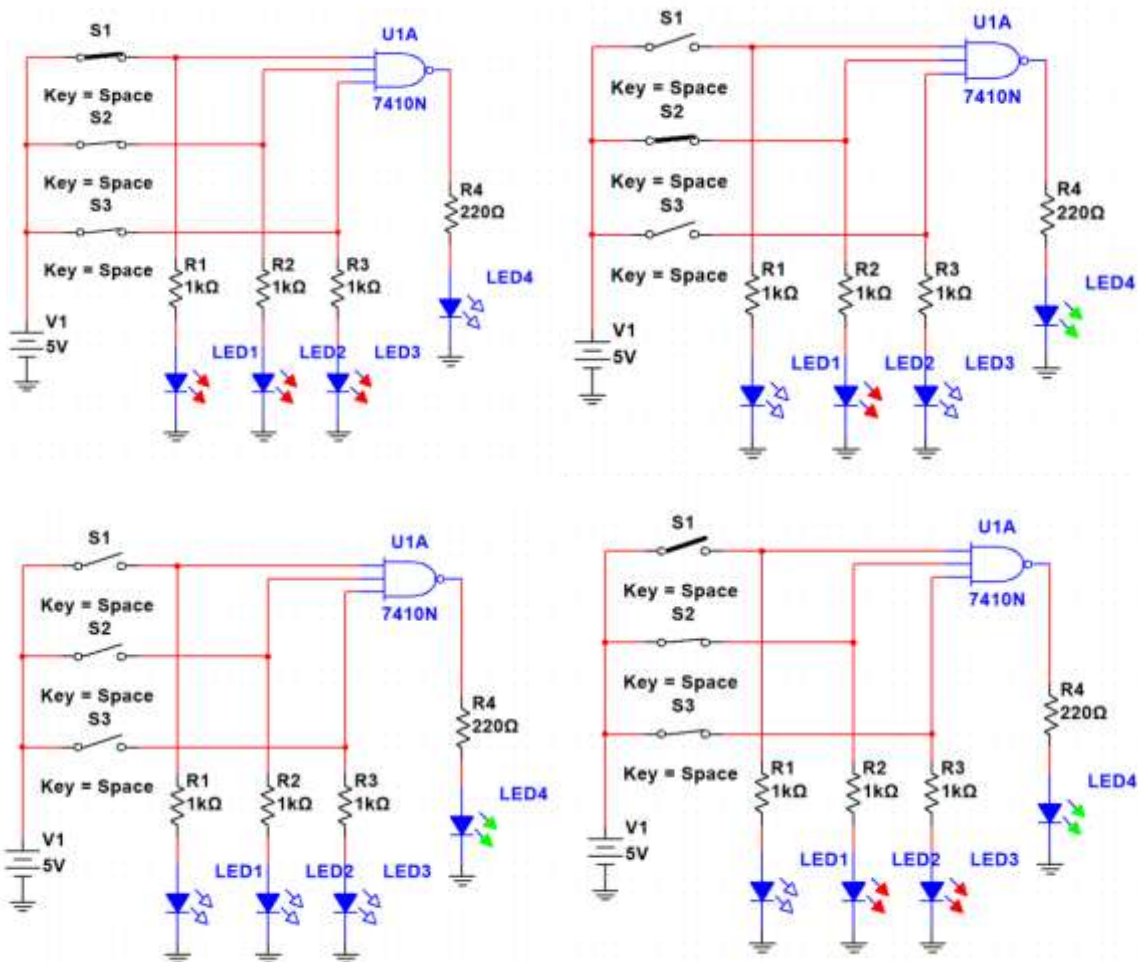


Рисунок 1.8 – Результати дослідження схеми 3І-НЕ в multisim

Друга схема в нас буде елемент ЗАБО-НЕ (рис. 1.9). Повторюємо всі кроки як для першої схеми. Результат дослідження наведено на рис. 1.10.

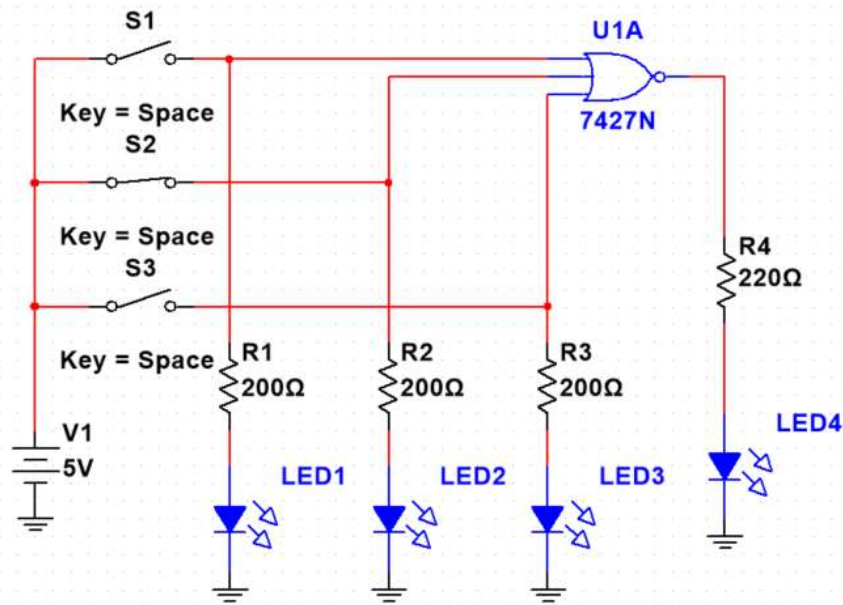


Рисунок 1.9 – Досліджувана схема 3І-НЕ в multisim

Результати дослідження цієї схеми наведено на рис. 1.10.

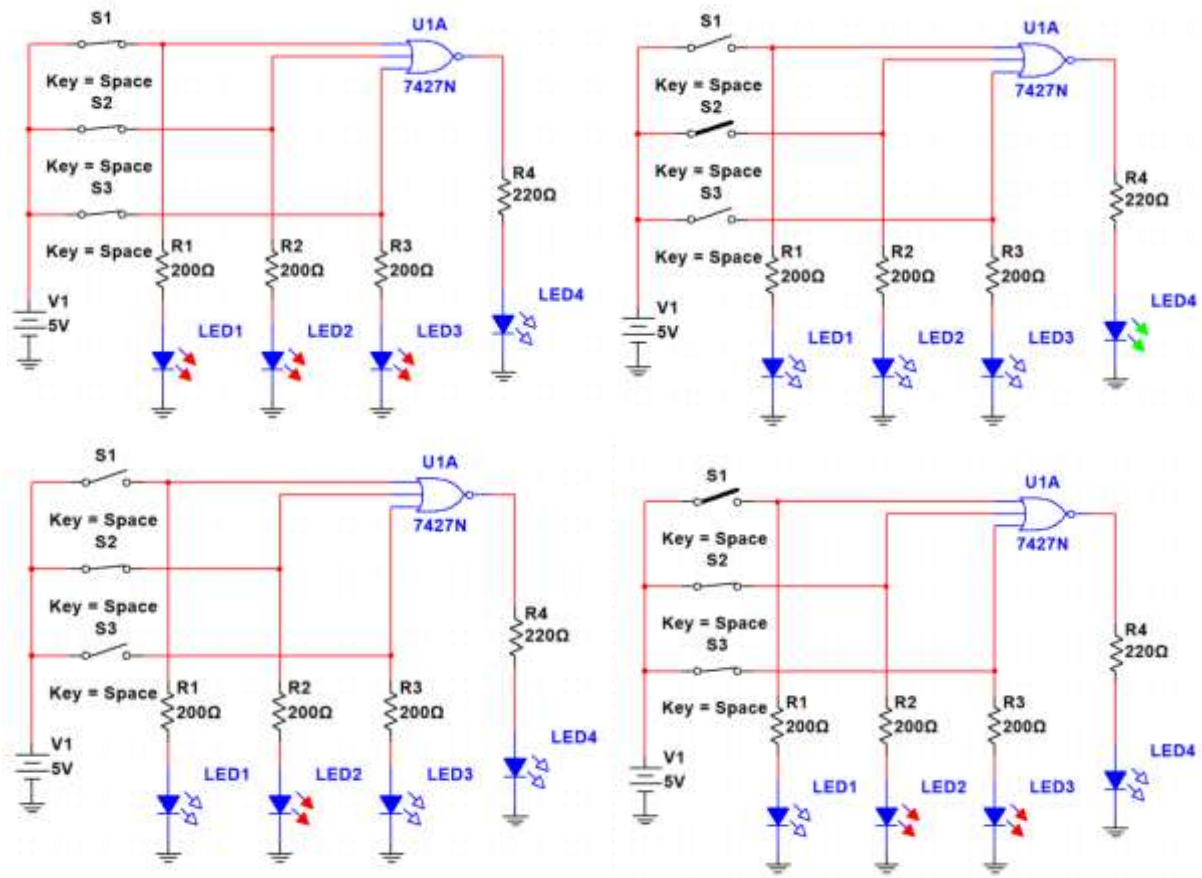


Рисунок 1.10 – Результати дослідження схеми ЗАБО-НЕ в multisim

3. Відповідно до результатів досліджень складаємо таблицю відповідностей (табл. 1.4.) для першої та другої схеми.

Таблиця 1.4 – Таблиці відповідностей для 5 макетів

3І–НЕ				3АБО–НЕ			
x_1	x_2	x_3	y	x_1	x_2	x_3	y
0	0	0	1	0	0	0	1
0	0	1	1	0	0	1	0
0	1	0	1	0	1	0	0
0	1	1	1	0	1	1	0
1	0	0	1	1	0	0	0
1	0	1	1	1	0	1	0
1	1	0	1	1	1	0	0
1	1	1	0	1	1	1	0

4. Проводячи аналіз результатів таблиці 1.4, ми підтвердили та дослідили задані функції логічних елементів 3І-НЕ та 3АБО-НЕ.

Варіанти завдань

Студентам пропонується розробити схему, що реалізує логічну функцію відповідно до вибраного варіанта з табл. 1.5, та перевірити її роботу.

Таблиця 1.5 – Таблиці відповідностей для 5 макетів

Функція	Елементи для побудови	Функція	Елементи для побудови
4І	2І	2АБО-НЕ	2АБО, НЕ
4І-НЕ	4І, НЕ	2І-НЕ	2І, НЕ
4І-НЕ	2І, НЕ	2Викл. АБО-НЕ	2Викл. АБО, НЕ
3АБО	2АБО	3І	2І, 2І
4АБО	2АБО	3І-2АБО	3І, 2АБО
3АБО-НЕ	2АБО, НЕ	4Викл. АБО	4Викл. АБО
2Викл. АБО	2Викл. АБО	4І-2Викл. АБО	4І, 2Викл. АБО

Зміст звіту

Звіт має містити:

1. Завдання.
2. Таблиці істинності спроектованих елементів.
3. Функціональні схеми реалізованих функцій.
4. Часові діаграми сигналів на входах і виходах схем.
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке логічний елемент і яка його роль у цифрових схемах?
2. Які основні типи логічних елементів використовуються в цифрових системах?
3. Як працює AND-елемент і яку логічну функцію він виконує?
4. У чому полягає різниця між OR-елементом та XOR-елементом?
5. Які істинні таблиці для логічних елементів AND, OR і NOT?
6. Як реалізується функція NAND за допомогою базових логічних елементів?
7. Які переваги використання NOR-елементів у цифрових схемах?
8. Які функції виконують логічні елементи NOT, NAND і NOR у цифрових системах?
9. Як працює логічний елемент XOR і в яких випадках його використовують?
10. Які основні характеристики логічних елементів, що впливають на їх продуктивність?
11. Як можна комбінувати логічні елементи для створення складніших логічних функцій?
12. Які типи вхідних і вихідних сигналів використовуються в логічних елементах?
13. Які проблеми можуть виникати під час проектування цифрових схем на основі логічних елементів?

Лабораторна робота 2

Тригери

Теоретична частина

Тригери є базовими елементами послідовнісних цифрових схем. Вони характеризуються здатністю перебувати в одному з двох стабільних станів та змінювати свій стан під дією керівних сигналів. Завдяки цій властивості тригери часто називають бістабільними пристроями.

Особливістю роботи тригера є те, що зміна його стану визначається не лише поточними вхідними сигналами, але і його попереднім станом. Інформація про поточний стан тригера передається через внутрішні зворотні зв'язки, що є невід'ємною частиною конструкції будь-якого тригера. [5]

Сучасні тригерні схеми реалізуються на базі логічних елементів (ЛЕ). Вони слугують фундаментальними блоками для створення більш складних цифрових пристроїв, таких як: лічильники імпульсів, подільники частоти, регістри різних типів, пристрої зберігання інформації.

Важливість тригерів для цифрової електроніки неможливо переоцінити, оскільки вони забезпечують функцію «пам'яті», необхідну для роботи послідовнісних схем.

Інтегральні тригери можна класифікувати за кількома ключовими ознаками: методом прийому інформації, архітектурою та функціональними характеристиками.

Відповідно до методу прийому інформації тригери поділяються на асинхронні та синхронні. Асинхронні тригери обробляють та реагують на інформаційні сигнали негайно після їх появи на відповідних входах. Синхронні тригери, натомість, обробляють вхідні дані лише за наявності спеціального дозвільного сигналу на керівному вході С (вході синхронізації).

Синхронні тригери додатково поділяються на два підтипи за принципом реакції на синхросигнал:

а) *тригери з керуванням за рівнем* сприймають інформаційні сигнали тільки за певного логічного рівня на синхровході – логічної одиниці (для прямого С-входу) або логічного нуля (для інверсного С-входу) [6];

б) *тригери з керуванням за фронтом* реагують на інформаційні сигнали виключно в момент зміни стану синхросигналу: або під час переходу з 0 в 1 (для прямого динамічного С-входу), або під час переходу з 1 в 0 (для інверсного динамічного С-входу).

Така класифікація дозволяє вибрати оптимальний тип тригера для конкретного застосування в цифрових системах.

За архітектурою синхронні тригери поділяються на *одноступеневі* та *двоступеневі* структури.

Одноступеневі тригери містять єдиний елемент запам'ятовування, який безпосередньо обробляє та зберігає інформацію. Ця архітектура є простішою, але має певні обмеження щодо стабільності роботи.

Двоступеневі тригери характеризуються наявністю двох послідовних запам'ятовувальних блоків. Процес обробки даних у таких тригерах відбувається у два етапи: спочатку інформація записується у перший блок, після чого передається до другого блока та з'являється на виході пристрою.

У технічній літературі двоступеневі тригери часто позначають як *MS*-тригери (від англійських термінів Master-Slave). Це позначення влучно відображає функціональний принцип їхньої роботи: вхідний (головний) блок формує нове значення вихідного сигналу Q , а вихідний (підпорядкований) блок виконує копіювання та виведення цього значення.

Така двоступенева структура забезпечує підвищену надійність роботи та зменшує ймовірність виникнення небажаних перехідних процесів.

Згідно з функціональними особливостями та організацією логічних зв'язків, тригери поділяються на такі основні типи:

1. Тригер з роздільними входами встановлення (*RS*-тригер). Містить два керівних входи: *R*-вхід для переведення пристрою в нульовий стан та *S*-вхід для переведення в одиничний стан. Назва утворена від англійських термінів: *R* (*Reset*) – скидання, *S* (*Set*) – встановлення.

2. Універсальний *JK*-тригер. Характеризується наявністю двох інформаційних входів: *J*-вхід для переходу в стан логічної одиниці та *K*-вхід для переходу в стан логічного нуля. Вважається найбільш функціонально повним типом тригера.

3. Тригер затримки (*D*-тригер). Має єдиний інформаційний вхід *D* (від англійського *Delay* – затримка). Особливість: стан тригера після спрацювання точно відповідає логічному рівню на *D*-вході в момент подачі синхросигналу.

4. Лічильний тригер (*T*-тригер). Оснащений спеціальним лічильним входом *T*. За подачі активного сигналу на *T*-вхід відбувається зміна стану тригера на протилежний.

5. Комбіновані тригери. Поеднують функціональні можливості різних типів тригерів. Приклад: *RST*-тригер, який має функції як лічильного режиму роботи, так і режиму прямого встановлення/скидання.

Кожен тип тригера має свої переваги та призначений для вирішення специфічних задач у цифрових системах [7].

Функціональна класифікація тригерів безпосередньо пов'язана з типами їхніх входів. Незалежно від типу, кожен тригер обладнаний двома виходами – прямим Q та інверсним \bar{Q} . Значення сигналу на прямому виході Q визначає поточний стан тригера.

Під час проектування цифрових систем враховують такі основні характеристики тригерів.

- Швидкодія – характеризується максимальною частотою зміни станів, яка для сучасних пристроїв може сягати сотень мегагерц.
- Порогова чутливість – визначається мінімальним рівнем вхідної напруги, необхідним для надійного перемикавання тригера.
- Енергоспоживання – кількість потужності, що витрачається під час роботи пристрою.
- Завадостійкість – здатність тригера коректно працювати за наявності зовнішніх електричних перешкод.
- Функціональність – визначається різноманітністю вхідних сигналів, які тригер може обробляти, та типами цих входів.

Для повної характеристики тригера необхідно подати його структурну схему, побудовану на базових логічних елементах (найчастіше використовуються елементи І-НЕ або АБО-НЕ), та описати принцип його функціонування. Опис функціонування може бути виконаний у вигляді логічного рівняння або таблиці переходів, яка відображає зміни станів залежно від вхідних сигналів.

Варто зазначити, що фундаментом для більшості типів тригерів слугує базовий RS-тригер, який може мати як прямі, так і інверсні входи. На основі цієї базової структури створюються більш складні тригерні схеми з розширеними функціональними можливостями [8].

Асинхронний RS-тригер з прямими входами має два інформаційні входи, позначені як R та S, які використовуються для встановлення його у стани 0 та 1, відповідно. Він також має два виходи— прямий Q та інверсний \bar{Q} . Цей тригер побудований на основі двох логічних елементів АБО-НЕ, з'єднаних зворотними зв'язками. Схематичне відображення та умовне графічне позначення (УГП) цього тригера можна знайти на рис. 2.1, у частинах а) та б), відповідно.

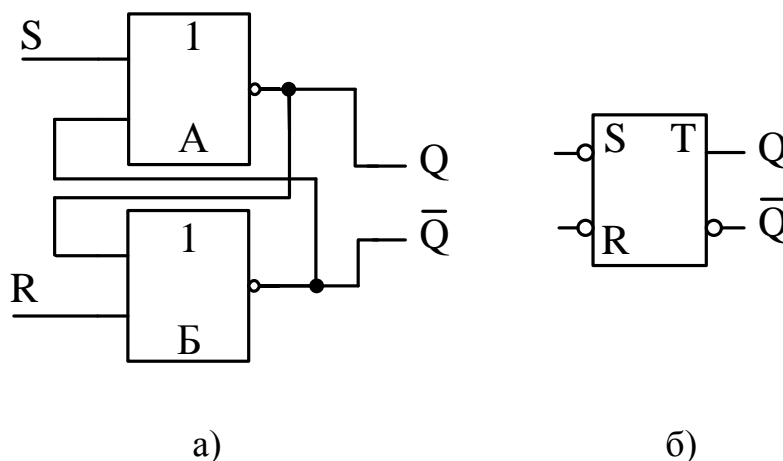


Рисунок 2.1 – Схема та УГП асинхронного RS-тригера з прямими входами

У цій схемі вихід кожного логічного елемента АБО-НЕ під'єднано до одного з входів іншого елемента, що забезпечує наявність двох стійких станів тригера. В RS-тригерах з прямими входами керівними сигналами є лише високі логічні рівні. Вхідні сигнали, які викликають перемикання елементів, називаються активними, а ті, що не викликають перемикання – пасивними. Для елементів АБО-НЕ активним сигналом є логічна одиниця.

Розглянемо випадок, коли на входи тригера подано $R=0$ та $S=0$. Якщо початковий стан тригера $Q=0$, то логічний нуль з виходу Q надходить на один з входів елемента Б, де на обох входах діють логічні нулі, і на виході елемента Б з'являється логічна одиниця ($\bar{Q}=1$). Логічна одиниця з виходу елемента Б надходить на вхід елемента А, що забезпечує на його виході логічний нуль. Це один зі стійких станів тригера. У стані 1 тригера $Q=1$, і відповідно $\bar{Q}=0$, водночас на обох входах елемента А діють логічні нулі, що забезпечує $Q=1$.

Таким чином, у кожному з двох стійких станів тригера елементи А і Б перебувають у протилежних станах. Перехід тригера з одного стійкого стану в інший відбувається за надходження активних сигналів на входи. Якщо $R=1$, і тригер перебував у стані 0 ($Q=0$), то цей стан не зміниться. Якщо ж тригер перебував у стані 1, то у разі надходження сигналу $R=1$ він перейде в стан 0. Аналогічно, якщо $S=1$, то тригер перейде в стан $Q=1$.

Одночасна подача активного сигналу (логічної 1) на обидва входи ($S=R=1$) призводить до невизначеного стану тригера після зняття сигналу. Це відбувається тому, що обидва виходи встановлюються в нульовий стан, а після припинення дії активних сигналів стан тригера стає випадковим, залежно від зовнішніх факторів. [9]

Функціонування тригера можна наочно подати за допомогою таблиці переходів (табл. 2.1), де Q^t – поточний стан тригера, а Q^{t+1} – наступний стан, в який він перейде після подачі комбінації сигналів R та S .

Таблиця 2.1 – Таблиця переходів асинхронного RS-тригера з прямими входами

R	S	Q^t	Q^{t+1}		Режим роботи тригера
0	0	0	0	Q^t	Режим зберігання інформації
0	0	1	1		
0	1	0	1	1	Встановлення 1
0	1	1	1		
1	0	0	0	0	Встановлення 0 (скидання)
1	0	1	0		
1	1	0	-	-	Комбінація заборонена
1	1	1	-		

Висновок. За $S=R=0$ тригер зберігає попередній стан (режим зберігання). За $R=1, S=0$ тригер переходить у стан 0, незалежно від

попереднього стану. Аналогічно, за $R=0, S=1$ тригер переходить у стан 1, незалежно від попереднього стану. Комбінація вхідних сигналів $S=R=1$ є недопустимою для RS-тригера з прямими входами.

Асинхронний RS-тригер з інверсними входами реалізовано за допомогою елементів І-НЕ. У цій конфігурації логічний нуль (0) на входах ініціює активну дію, тоді як логічна одиниця (1) не викликає змін. Схематичне відображення та умовне графічне позначення цього тригера можна побачити на рисунку 2.2, а) та 2.2, б), відповідно [10].

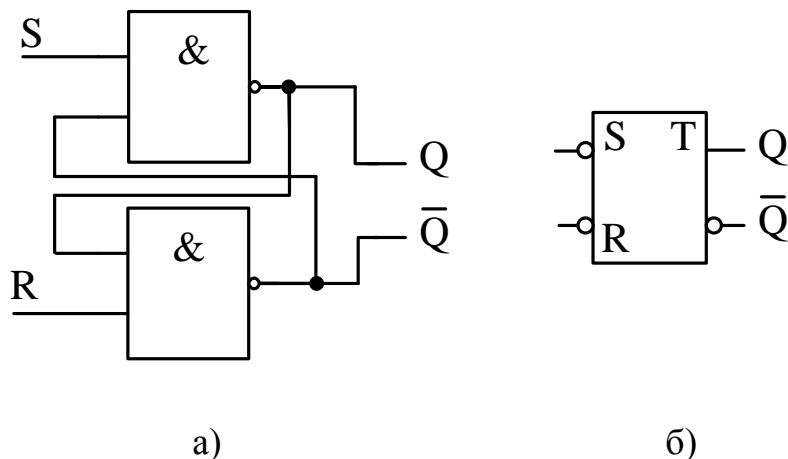


Рисунок 2.2 – Схема та УГП тригера з інверсними входами

Можливі стани тригера показано у таблиці переходів (табл. 2.2).

Висновки

1. За $S = R = 1$ тригер залишається у попередньому стані.
2. За $R = 1; S = 0$ $Q^{t+1} = 1$ і аналогічно за $R = 0; S = 1$ $Q^{t+1} = 0$.
3. Комбінація $S=R=0$ є забороненою.

Таблиця 2.2 – Таблиця переходів асинхронного RS-тригера з інверсними входами

S	R	Q^{t+1}
0	0	-
0	1	1
1	0	0
1	1	Q^t

Синхронний RS-тригер має додатковий вхід C , призначений для синхронізації тактовими сигналами. Структура синхронного тригера складається з асинхронного RS-тригера та комбінаційної цифрової схеми, як це показано на рисунку 2.3, а). Умовне графічне позначення такого тригера представлено на рисунку 2.3, б).

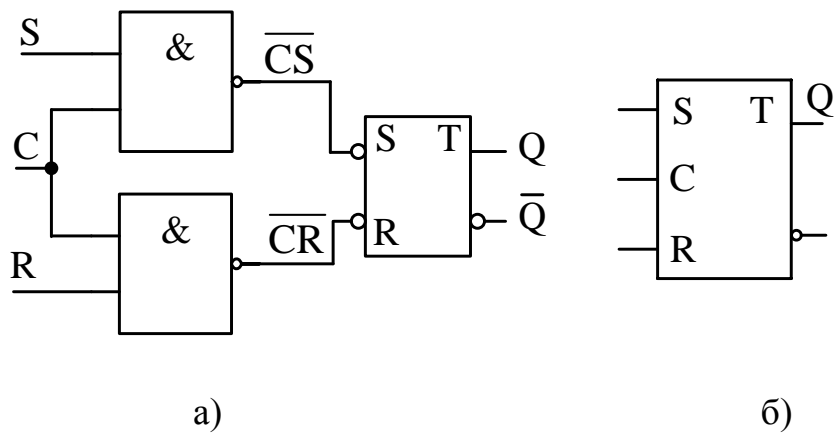


Рисунок 2.3 – Схема та УГП синхронного RS-тригера

Використовуючи елементи І-НЕ для створення комбінаційної схеми на вході, забезпечується передача активних сигналів на входи S та R синхронного тригера до інверсних входів внутрішнього асинхронного тригера лише тоді, коли на синхровході C присутня логічна одиниця. Якщо $C = 1$, стан тригера визначається сигналами на входах, подібно до асинхронного тригера. За $C = 0$ тригер ігнорує сигнали на S та R . Функціонування тригера описано в таблиці переходів (табл. 2.3).

Рядки, де $C = 0$, виключені з таблиці, оскільки вони не надають додаткової інформації.

Таблиця переходів цього синхронного тригера практично ідентична таблиці переходів асинхронного RS-тригера з прямими входами (без урахування стовпця C у таблиці 2.3), хоча використовується асинхронний RS-тригер з інверсними входами. Це пояснюється тим, що в схемі на рисунку 2.3 сигнали інвертуються двічі: на виходах елементів І-НЕ комбінаційної схеми та на інверсних входах асинхронного тригера. Згідно з законом подвійної інверсії ($\overline{\overline{X}} = X$) це еквівалентно відсутності інверсії. Тому в схемі можна прибрати всі позначення внутрішніх інверсій, і таблиця переходів залишиться незмінною.

Таблиця 2.3 – Таблиця переходів синхронного RS-тригера

S	R	C	Q^{t+1}
0	0	1	Q^t
0	1	1	0
1	0	1	1
1	1	1	-

Отже, наявність синхронізувального входу C розширює функціональні можливості тригерів, що робить його стандартною характеристикою в промислово виготовлених пристроях. Водночас зберігається можливість асинхронної роботи, яка активується подачею

відповідного сигналу на вхід С. Обмеженням RS-тригерів є їхня нездатність обробляти певні комбінації вхідних сигналів. Статичні RS-тригери використовуються як будівельні блоки для регістрів пам'яті.

JK-тригер, як універсальний елемент, працює подібно до RS-тригера, де вхід J відповідає входу S, а вхід K – входу R. Це означає, що активний сигнал (логічна одиниця) на вході J встановлює тригер у стан 1, а на вході K – у стан 0. Відмінність полягає в тому, що за одночасного активного сигналу на J і K тригер змінює свій стан на протилежний. JK-тригер усуває обмеження RS-тригера, дозволяючи обробляти всі комбінації вхідних сигналів [11].

Найпростіший JK-тригер можна сконструювати з тих самих компонентів, що й синхронний RS-тригер. Схематичне зображення та таблицю переходів для цього тригера подано на рис. 2.4 та в табл. 2.4, відповідно.

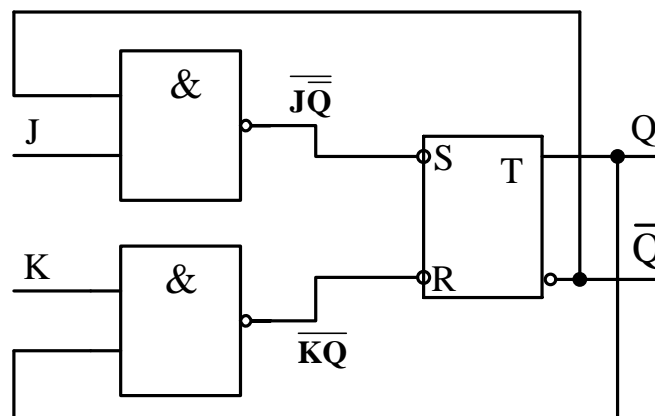


Рисунок 2.4 – Схема найпростішого JK-тригера

Таблиця 2.4 – Таблиця переходів JK-тригера

J	K	Q^{t+1}
0	0	Q^t
0	1	0
1	0	1
1	1	\bar{Q}^t

Розглянемо випадок, коли тригер перебуває в стані 0 ($Q = 0, \bar{Q} = 1$), а на його входи подаються сигнали $J = K = 0$. Згідно з таблицею 2.4, рядок 4, така комбінація вхідних сигналів ($R = S = 1$) в асинхронному RS-тригері з інверсними входами підтримує режим збереження інформації, тобто стан тригера не змінюється. Якщо ж на входи тригера подати сигнали $J = K = 1$, то вихідний сигнал другого елемента І-НЕ залишиться незмінним, а на виході першого елемента з'явиться сигнал логічного 0. За такої комбінації вхідних сигналів ($R = 1, S = 0$) асинхронний RS-тригер з інверсними

входами переходить у стан 1 (табл. 2.4, рядок 2). Аналогічно можна проаналізувати випадок, коли тригер спочатку перебував у стані 1. Отже, JK-тригер у разі подачі сигналів $J = K = 1$ на його входи дійсно змінює свій стан на протилежний.

Варто зазначити, що на практиці використовуються більш складні схеми JK-тригерів, ніж показано на рисунку 2.4.

По-перше, аналіз роботи найпростішої схеми показує, що у випадку тривалої подачі сигналів $J=K=1$ на входи тригер буде безперервно змінювати свій стан. Тому реальні JK-тригери завжди мають вхід синхронізації.

По-друге, синхронні JK-тригери зазвичай будуються за двоступеневим принципом, тобто мають два RS-тригери: керований та керівний. Така структура синхронних тригерів (не лише JK, а й інших типів) запобігає встановленню тригера в невизначений стан через перехідні процеси або короткочасні зміни вхідних сигналів. Важливо, щоб сигнали синхронізації на керований та керівний RS-тригери надходили в протифазі.

Схему одноступеневого синхронного JK-тригера показано на рисунку 2.5, а), а умовне графічне позначення двоступеневого JK-тригера – на рисунку 2.5, б).

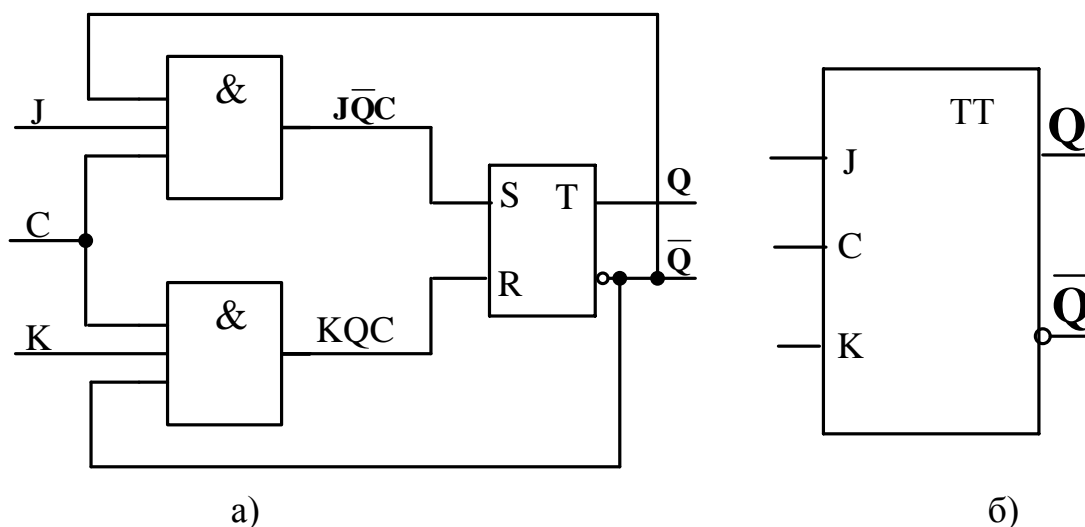


Рисунок 2.5 – Схема синхронного JK-тригера

D- і T-тригери

D-тригер обладнаний одним інформаційним входом (D-вхід) і входом синхронізації C. Його основна функція полягає в затримці сигналу на один тактовий імпульс. Конструктивно D-тригер може бути реалізований на основі JK-тригера шляхом з'єднання входу K з входом J через інвертор, як це показано на рисунку 2.6, а).

Роботу D-тригера можна описати за допомогою таблиці переходів (табл. 2.5), де рядки для $C = 0$ опущено. За $C = 0$ тригер зберігає попередній стан протягом необмеженого часу. За $C = 1$, дані, що надходять

на вхід D, передаються на вихід тригера з затримкою на один тактовий імпульс. Графічне позначення D-тригера подано на рисунку 2.6, б).

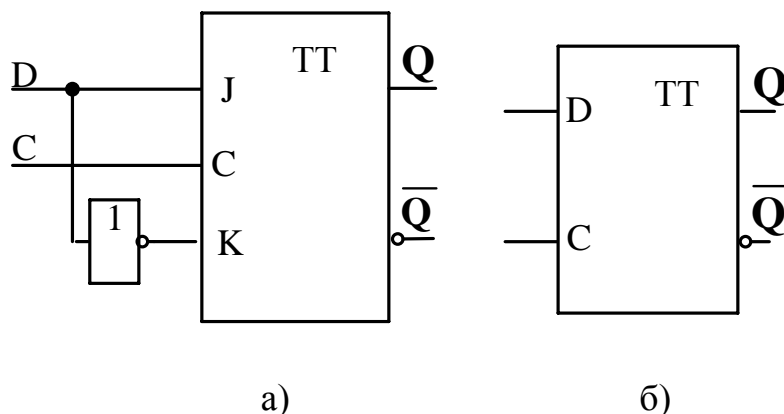


Рисунок 2.6 – Схема та УГП D-тригера

Таблиця 2.5 – Таблиця переходів D-тригера

D	C	Q^{t+1}
1	1	1
0	1	0

D-тригер також може бути отриманий із схеми найпростішого RS-тригера, як це зображено на рисунку 2.7.

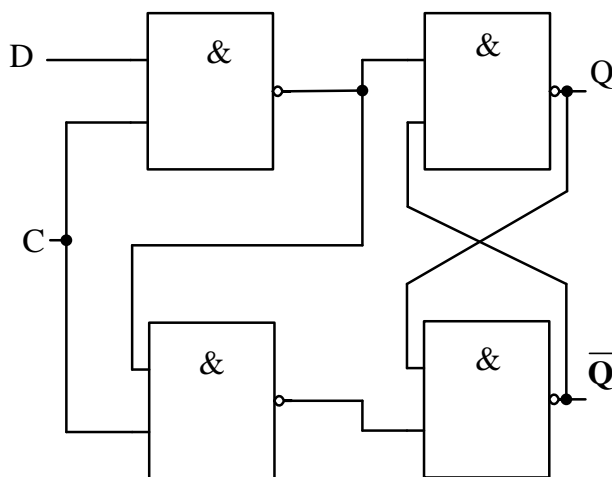


Рисунок 2.7 – Схема D-тригера

Як відомо, у разі подачі логічної одиниці на обидва входи JK-тригера ($J = K = 1$), він змінює свій стан на протилежний. З цього погляду, найпростішим способом перетворення JK-тригера в асинхронний T-тригер є з'єднання його J і K-входів для створення лічильного T-входу, що показано на рисунку 2.8, а) [12].

Однак на практиці частіше використовується синхронна схема T-тригера, де вхід C синхронного JK-тригера використовується як T-вхід, а

на J- і K-входи подається логічна одиниця, як показано на рисунку 2.8, б). Алгоритм роботи T-тригера, який є однаковим для схем 2.8, а) та 2.8, б) (без урахування сигналу синхронізації), подано у таблиці переходів (табл. 2.6). Умовне графічне позначення тригера показано на рисунку 2.8, в).

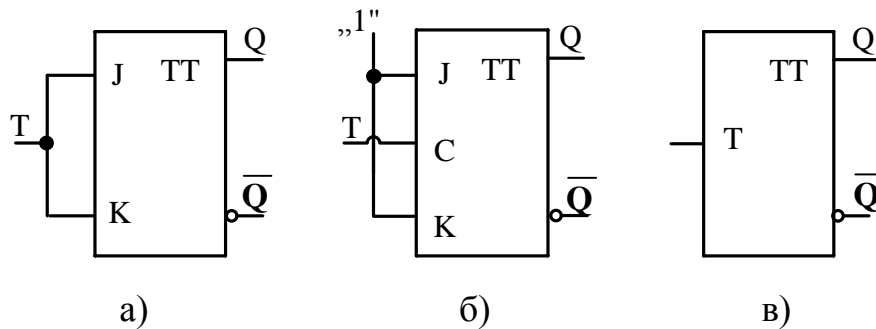


Рисунок 2.8 – Варіанти побудови та УГП T-тригера

Таблиця 2.6 – Таблиця переходів T-тригера

T	Q^{t+1}
0	Q^t
1	$\overline{Q^t}$

T-тригер також можна реалізувати, використовуючи тригери інших типів. На рисунку 2.9 подано приклади схем асинхронного T-тригера, побудованих на D-тригері та синхронному RS-тригері [13].

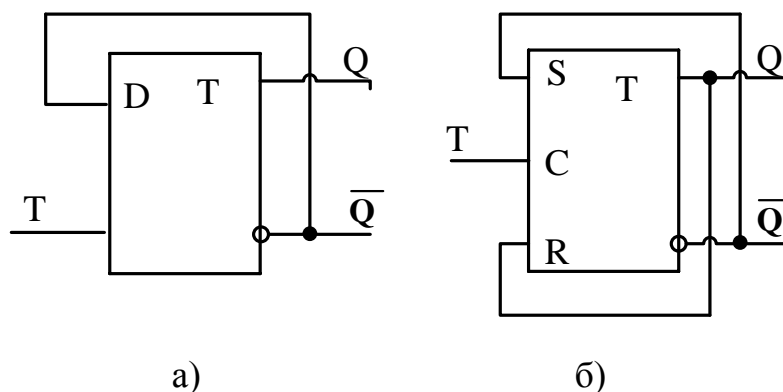


Рисунок 2.9 – Варіанти побудови T-тригера

Таким чином можна підсумувати, що перетворення одного типу тригера в інший є поширеною практикою в цифровій схемотехніці. Це дозволяє використовувати наявні елементи для реалізації необхідної функціональності, оптимізуючи схему та зменшуючи кількість необхідних компонентів. Перетворення тригерів широко використовується в практичних застосуваннях, таких як лічильники, регістри, схеми керування

та інші цифрові пристрої. Під час перетворення тригерів необхідно враховувати особливості їх роботи, такі як час затримки, навантажувальна здатність та інші параметри. Вибір оптимального способу перетворення залежить від конкретного застосування та вимог до схеми. Таким чином, взаємне перетворення тригерів є важливим інструментом в арсеналі інженера-електронщика, що дозволяє створювати ефективні та функціональні цифрові схеми.

Порядок виконання роботи (off-line частина)

1. Вибираємо мікросхему тригера.

Виходячи з критеріїв наявності, ми вибрали мікросхему K561TM2. Ця мікросхема являє собою два D-тригери зі встановленням 0 і 1 (її іноземний аналог – CD4013A). Технічні характеристики K561TM2 наведено в табл. 2.7. На рис. 2.10 наведено призначення виводів мікросхеми (а) та її умовне позначення (б).

Таблиця 2.7 – Технічні характеристики мікросхеми K561TM2

Кількість виводів або контактів	14
Маса виробу, г	0,97
Інтервал робочих температур	від -45 до +85°C
Напруга живлення	10 V
Вихідна напруга	низького рівня не більше ніж 1,0 V; високого рівня не менше ніж 9,0 V
Вихідний струм	низького рівня не менше ніж 0,9 mA; високого рівня не менше ніж 0,6 mA
Вхідний струм	0,3 μA
Споживаний струм	20 мкА
Час затримки поширення	не більше ніж 150 ns

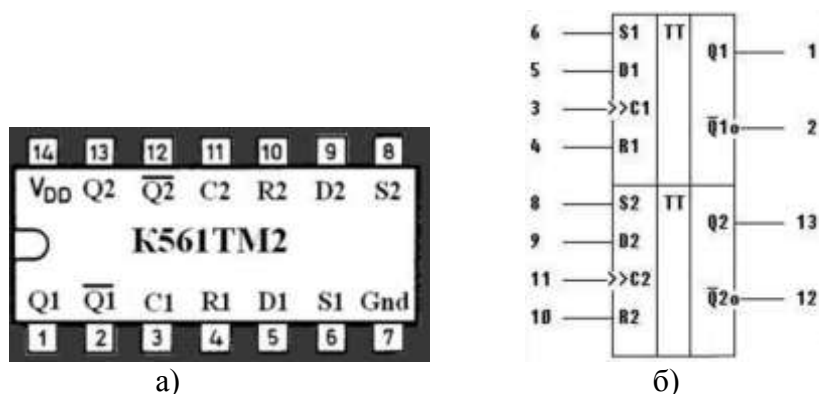


Рисунок 2.10 – Призначення виводів мікросхеми K561TM2 (а) та умовне позначення тригерів K561TM2 (б)

2. За схемою рис. 2.11 складаємо макет (рис. 2.12). Потрібно зазначити, що на вихід ми підключаємо ще послідовно діод та резистор, щоб візуально бачити результат на виході.

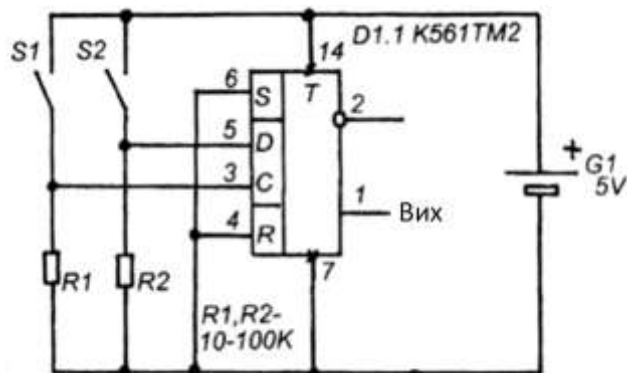


Рисунок 2.11 – Схема макета D-тригера на К561ТМ2.

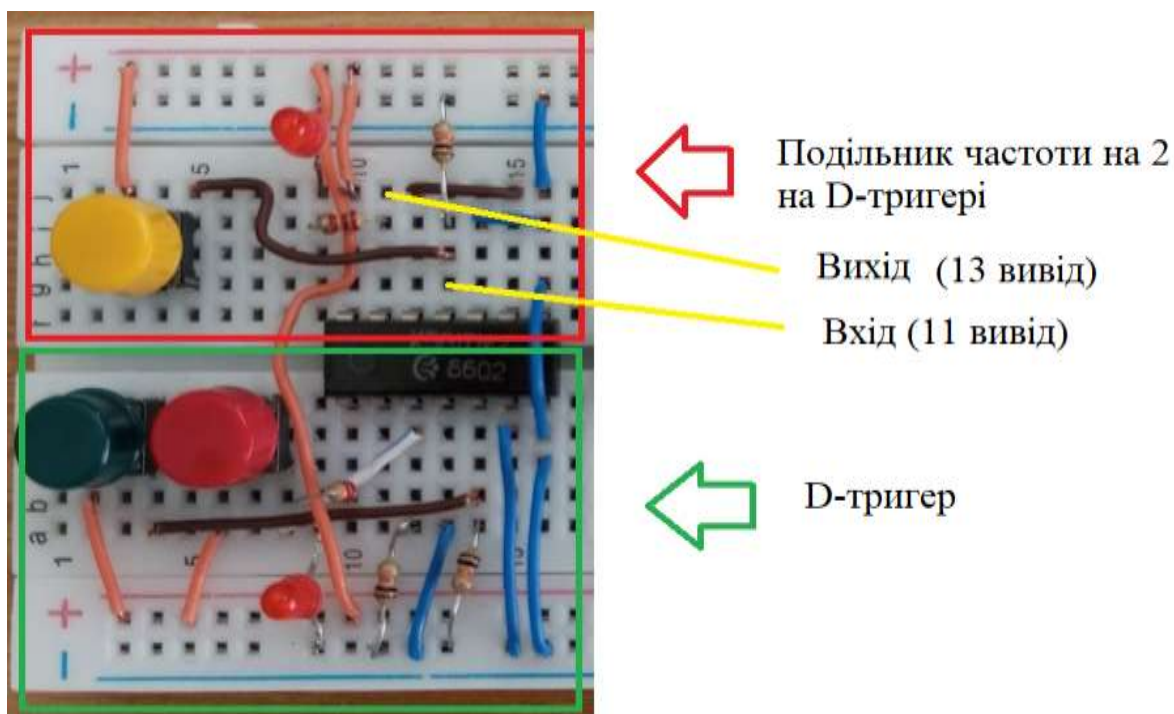


Рисунок 2.12 – Схема макета D-тригера на К561ТМ2

Наведемо опис роботи схем рис. 2.11 та 2.12. S1 (червона), S2 (зелена) – кнопки, входи S і R тригера з'єднаємо зі спільним мінусом живлення, щоб вони нам не заважали. У момент увімкнення живлення тригер опиниться в одному з двох положень: або нуль на виводі 1, або на ньому одиниця. Припустимо, на виході нуль (червоний діод не горить). Якщо ми будемо натискати на S1, нічого не зміниться. Але якщо спочатку замкнути S2 (на вхід D подати одиницю), а потім, утримуючи S2 в замкнутому стані, натиснути на S1, то тригер перекинеться в одиничний стан, і на його виводі 1 буде одиниця (напруга близько напруги живлення).

Тепер, утримуючи S2 все ще в натиснутому стані, спробуємо знову натиснути на S1 – нічого не змінюється. Тригер жорстко тримається в одиничному стані. Спробуємо розімкнути S2 (тепер на вхід D надходить нуль через R2). Знову натиснемо на S1 – тригер повернеться в нульовий стан (нуль на виводі 1). Таким чином, за натискання на S1 тригер встановлюється в таке положення, за якого логічний рівень на його прямому виході буде таким самим, як на вході D. Після відпускання S1 тригер залишиться у встановленому положенні, йому буде «все одно», що на вході D, якщо на вході C (кнопка S1) нуль. Тобто, якщо на вході D буде одиниця, то в момент натискання на S1 (подача одиниці на вхід C) стан тригера стане одиничним (одиниця на виводі 1) і залишиться таким і після відпускання S1 та зміни рівня на D. Але якщо на D подати нуль і, утримуючи цей нуль, натиснути на S1 (подати одиницю на C), то тригер перейде в нульове положення.

Зауважимо, що рівні на виводах 1 і 2 протилежні, оскільки вивід 2 – інверсний вихід (начебто сигнал з виводу 2 подали на інвертор і знімають з його виходу). Таким чином, коли тригер в одиничному стані, на виводі 2 буде нуль, а коли в нульовому, на цьому виводі буде одиниця.

3. Проводимо дослідження D-тригера.

Результати дослідження роботи D-тригера макета наведено на рис. 2.13.

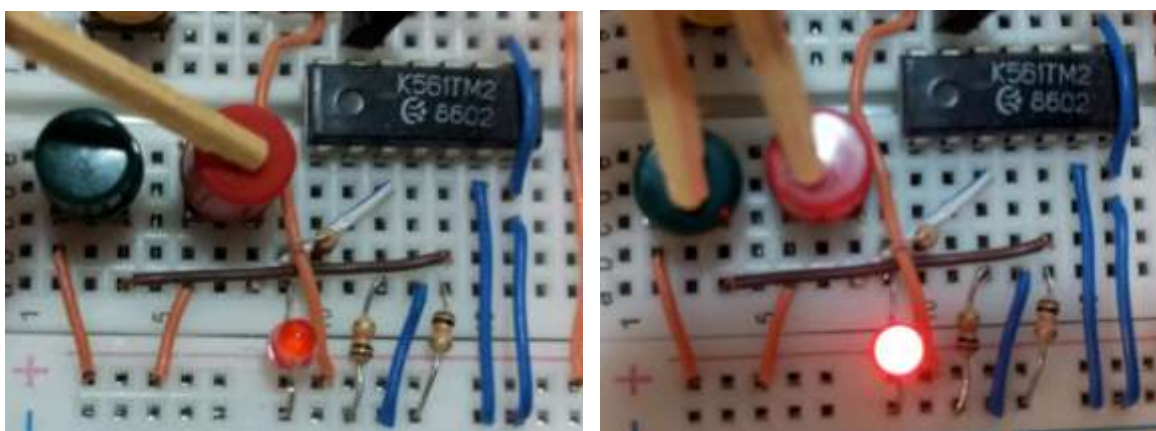


Рисунок 2.13 – Результати дослідження D-тригера на K561TM2

Результати досліджень рис. 2.13 підтвердили функціонування D-тригера, який ми склали. Без натискання зеленої кнопки ми на виході червоною кнопкою одиницю не встановимо. Лише натиснувши зелену і не відпускаючи її натисканням червоною ми встановимо логічну одиницю на виході. Скинути одиницю в нуль на виході ми можемо просто натиснувши червону кнопку.

Порядок виконання роботи (on-line частина)

1. Складаємо в multisim схему D-тригера як показано на рис. 2.14.

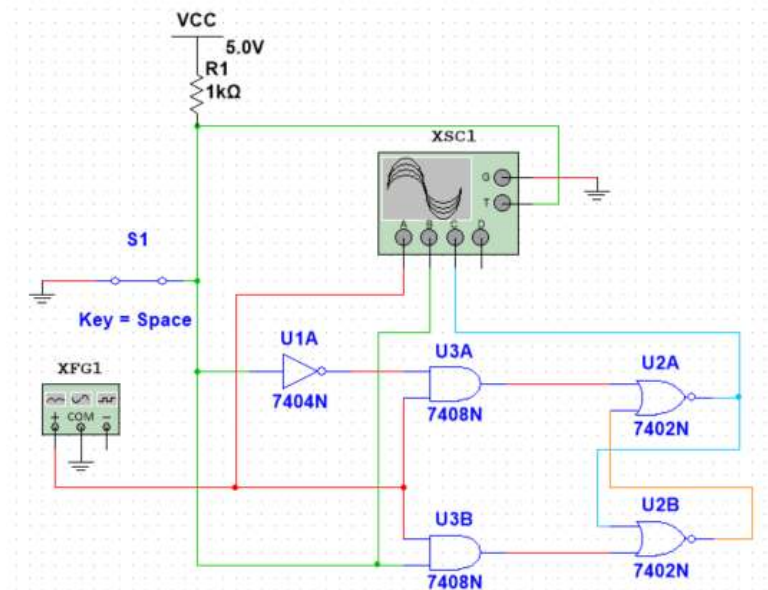


Рисунок 2.14 – Схема D-тригера на логічних елементах в multisim

2. Проводимо дослідження. Результат зображено на рис. 2.15.

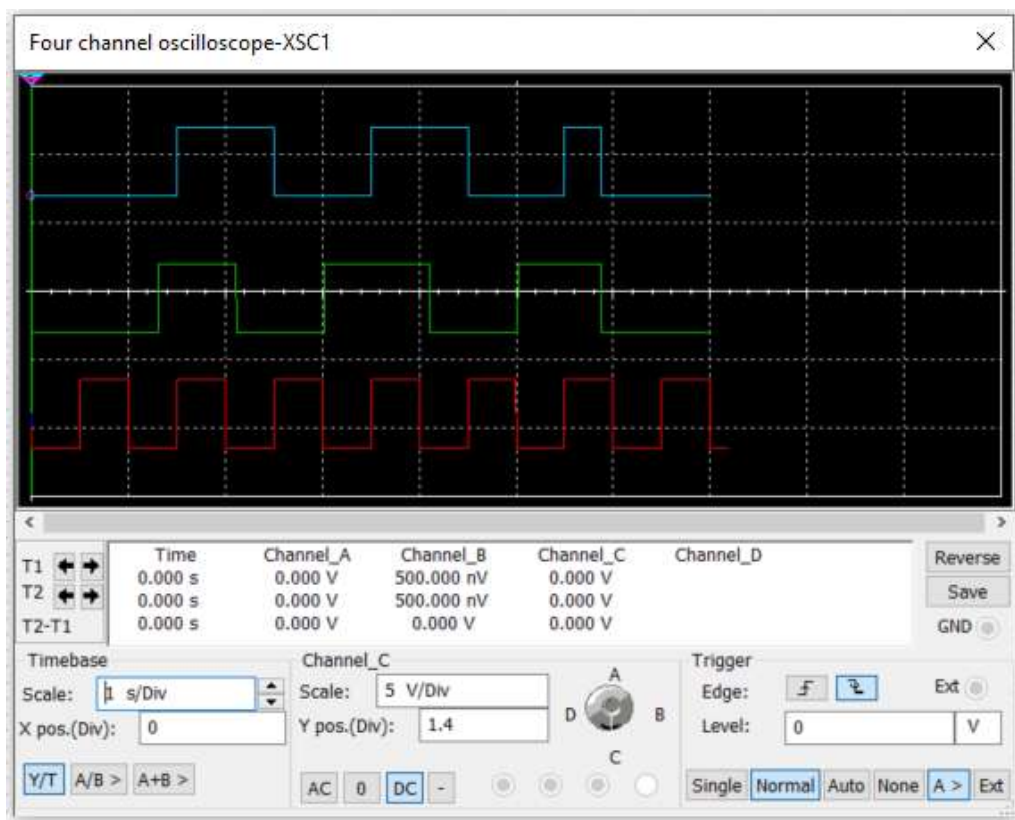


Рисунок 2.15 – Результат досліджень D-тригера на логічних елементах в multisim

Результати досліджень рис. 2.15 підтвердили таблицю станів (табл. 2.8) D-тригера.

Таблиця 2.8 – Стани D-тригера

Q^n	D	C	Q^{n+1}
0	X	0	0
1	X	1	1
0	0	┘	0
1	0	┘	1
X	0	┘	0
X	1	┘	1

Таким чином, результати моделювання відповідають нашим очікуванням.

Варіанти завдань

1. Побудуйте схему S-R тригера з використанням базових логічних елементів і протестуйте його на різних комбінаціях входів.
2. Створіть D-тригер і проаналізуйте його роботу. Визначте, як він зберігає дані.
3. Розробіть T-тригер і продемонструйте його функції, зокрема можливість підрахунку імпульсів.
4. Досліджуйте синхронний S-R тригер. Як він відрізняється від асинхронного? Створіть таблицю істинності.
5. Використайте D-тригери для реалізації двійкового лічильника. Проаналізуйте його роботу.
6. Проаналізуйте, як синхронізація тригерів впливає на їхні характеристики. Створіть схему для демонстрації.
7. Побудуйте D-тригер, використовуючи тільки NAND або NOR елементи. Поясніть, чому це можливо.
8. Досліджуйте затримки в різних типах тригерів. Як це вплине на їх продуктивність?
9. Розробіть проєкт, у якому тригери використовуються для реалізації певної логічної функції.
10. Проаналізуйте, як тригери використовуються в пам'яті (наприклад, в оперативній пам'яті).

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного тригера.
4. Результати досліджень (рисунок, фото, скрін екрана, таблиці відповідності і т. д.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке тригер і яка його основна функція в цифрових схемах?
2. Які основні типи тригерів існують?
3. Як працює D-тригер і в яких випадках його використовують?
4. Яка роль S-R тригера в цифрових системах?
5. Як працює T-тригер і які його основні характеристики?
6. Яка різниця між асинхронними та синхронними тригерами?
7. Як реалізується функція пам'яті в тригерах?
8. Які істинні таблиці для S-R, D та T тригерів?
9. Як тригери використовуються для побудови лічильників?
10. Яка роль тригерів у системах автоматичного управління?
11. Як впливають затримки на роботу тригерів?
12. Які проблеми можуть виникати у разі використання тригерів у схемах?
13. Як можна скласти тригер за допомогою базових логічних елементів?
14. Як змінюється вихідна інформація в тригері за змін вхідних сигналів?
15. Які електричні характеристики впливають на роботу тригерів?

Лабораторна робота 3

Лічильники

Теоретична частина

Лічильник – це цифровий пристрій, який відображає кількість вхідних імпульсів у певному коді. Основу лічильника становлять тригери, з'єднані у кола для передачі інформації з одного розряду в інший.

Основні функції лічильників:

- підрахунок імпульсів, що надходять на вхід;
- тимчасове зберігання кожного стану лічильника;
- перетворення послідовності вхідних імпульсів у паралельний код на виходах (у двійковій формі);
- поділ частоти вхідного імпульсного сигналу.

Лічильники є одними з найбільш поширених цифрових пристроїв і використовуються для перетворення кількості сигналів у певний код, частотного ділення, а також для підсумовування чи віднімання сигналів. Вони застосовуються для створення розподілювачів сигналів, цифрових фазоперетворювачів тощо. Лічильник є одним з основних функціональних елементів різних цифрових та інформаційно-вимірювальних систем. Основне застосування лічильників:

- утворення послідовності адрес команд програми (лічильник команд або програмний лічильник);
- підрахунок числа циклів під час виконання операцій ділення, множення, зсуву (лічильник циклів);
- одержання сигналів мікрооперацій і синхронізації; аналого-цифрові перетворення і побудова електронних таймерів (годинників реального часу).

Двійковий лічильник складається з послідовно з'єднаних Т-тригерів, які формують пам'ять з певною кількістю стабільних станів (рис. 3.1). Вхідні імпульси можуть надходити на лічильник як періодично, так і в довільний спосіб. Амплітуда та тривалість рахункових імпульсів мають відповідати технічним вимогам для використовуваних мікросхем [14].

Розрядність лічильника n визначається кількістю Т-тригерів. Кожен вхідний імпульс змінює стан лічильника, який зберігається до отримання наступного сигналу. Значення виходів тригерів лічильника Q_n, Q_{n-1}, Q_1 відображають результат підрахунку у вибраній системі числення.

Логічна функція лічильника позначається буквами СТ (counter). До мікрооперацій лічильника входять попереднє встановлення в початковий стан, інкремент або декремент збереженого слова, а також видача слів у паралельному коді.

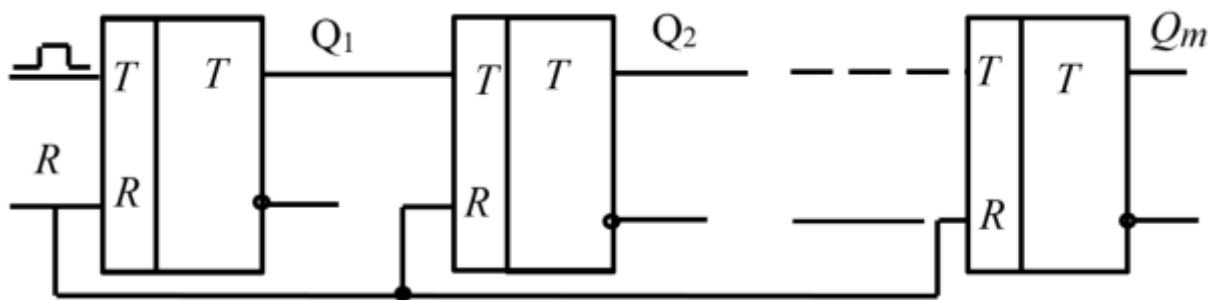


Рисунок 3.1 – Логічна структура лічильника

Лічильник має характеристики, такі як модуль і ємність рахунку. **Модуль рахунку** $K_{лч}$ визначає кількість станів лічильника. Для двійкового n -розрядного лічильника модуль виражається як цілий ступінь двійки $M = 2^n$. Після підрахунку імпульсів $N_{вх} = K_{лч}$ лічильник повертається до початкового стану. Таким чином, модуль рахунку, який часто називають коефіцієнтом перерахування, визначає цикл роботи лічильника, після якого його стан повторюється. Тому число вхідних імпульсів і стан лічильника однозначно визначаються лише для першого циклу.

Ємність рахування N_{max} визначає максимальну кількість вхідних імпульсів, яку лічильник може зафіксувати за один цикл роботи. Вона обчислюється за формулою $N_{max} = K_{лч} - 1$, якщо лічильник починає свою роботу з нульового початкового стану.

Лічильники функціонують в трьох режимах: управління, накопичення і ділення. У режимі управління інформацію зчитують після кожного вхідного рахункового імпульсу, як, наприклад, у лічильнику адрес команд. Режим накопичення зосереджений на підрахунку певної кількості імпульсів або нарахуванні протягом певного часу. У режимі ділення (перерахування) основна мета полягає в зменшенні частоти імпульсів, що надходять, у $K_{лч}$ разів [15].

Більшість лічильників можуть працювати в усіх цих режимах, проте спеціальні лічильники-дільники дозволяють зміну станів у процесі рахунку в довільному порядку, що спрощує конструкцію вузла.

Лічильники класифікують за кількома ознаками.

- Спосіб кодування: позиційні та непозиційні.
- Модуль рахування: двійкові, десяткові, з фіксованим або змінним (програмованим) модулем.
- Напрямок рахування: прості (які підсумовують або віднімають) і реверсивні.
- Спосіб організації міжрозрядних зв'язків: з послідовними, наскрізними, паралельними та комбінованими перенесеннями (позиною).
- Тип використовуваних тригерів: Т, JK, D у рахунковому режимі.
- Елементний базис: потенційні, імпульсні та потенційно-імпульсні.

Двійкові підсумовувальні та віднімальні лічильники

Двійкові лічильники здійснюють підрахунок вхідних імпульсів у двійковій системі числення. Кількість розрядів n двійкового підсумовувального лічильника для заданого модуля M обчислюють за формулою $n = \log_2 M$.

Значення поточного числа N вхідних імпульсів n -розрядного підсумовувального лічильника, починаючи з нульового початкового стану, визначається за формулою:

$$N = \sum_{i=1}^n r_i Q_i = r_n Q_n + r_{n-1} Q_{n-1} + \dots + r_1 Q_1,$$

де 2^{i-1} – вага i -го розряду;

$Q_i \{0,1\}$ – логічне значення прямого виходу тригера i -го розряду

Розрядність лічильника « n » дорівнює числу JK-тригерів або інших типів тригерів (для побудови лічильників використовуються Т-тригери, JK- і D-тригери, які працюють в лічильному режимі).

У двійковому лічильнику, що підсумовує, перенесення P_i у старший сусідній розряд Q_{i+1} відбувається, якщо на момент надходження чергового рахункового імпульсу U всі молодші розряди знаходяться в одиничному стані, тобто $P_i = U + Q_i Q_{i-1} \dots Q_1 = 1$. Після цього перенесення старший розряд перемикається в стан «1», а всі молодші розряди переходять у стан «0».

Підсумовувальні асинхронні лічильники на двоступеневих Т-тригерах організовані таким чином, що вхідні імпульси U надходять лише на рахунковий вхід першого (молодшого) розряду. Сигнали перенесення передаються асинхронно (попередньо в часі) з прямих виходів молодших розрядів на Т-входи сусідніх старших розрядів. Зміна станів тригерів відбувається за спадом рахункового імпульсу для першого розряду, а для останніх – за спадом сигналу перенесення (рис. 3.2, б).

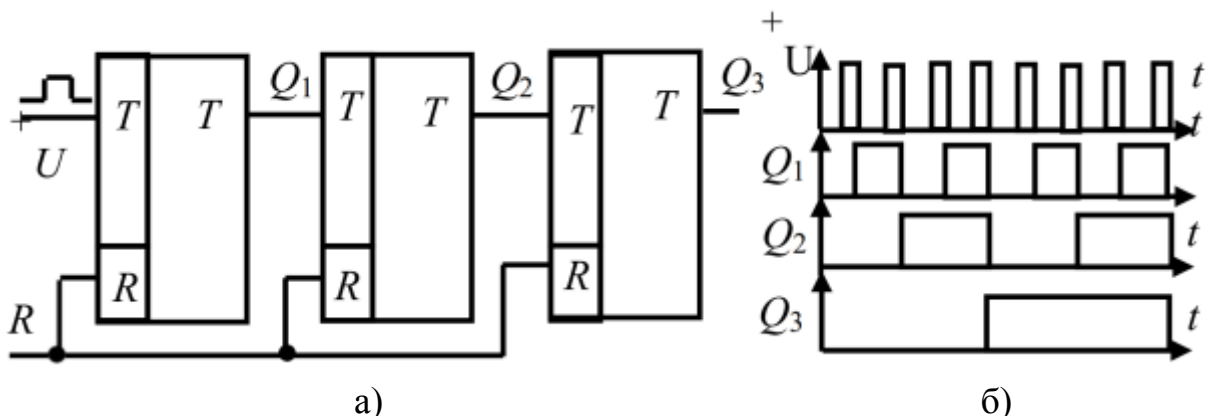


Рисунок 3.2 – Асинхронний підсумовувальний лічильник на двоступеневих Т-тригерах: а) – схема; б) – часова діаграма роботи

Після підрахунку семи імпульсів трирозрядний лічильник показує двійковий код $Q_3 Q_2 Q_1 = 111$, що відповідає максимальному значенню його ємності. За надходження восьмого вхідного імпульсу U підсумовувальний лічильник переходить у початковий нульовий стан послідовно:

спочатку зменшується напруга на виході Q_1 , потім на Q_2 і так далі.

Лічильник можна також скинути в нульовий стан у будь-який час, використовуючи імпульс на вході скидання R .

Часові параметри n -розрядного асинхронного лічильника на двоступеневих тригерах визначаються за такими співвідношеннями:

- *Роздільна здатність:* $t_{pz} = t_{T1} + t_{T2} = 2t_T$, де t_{T1} і t_{T2} – час перемикавання для першого і другого ступеня тригера, причому $t_1 = -t_{T2} = t_T$.
- *Час встановлення:* $t_{уст} = t_{T1} + nt_{T2} \cup nt_T$.
- *Максимальна частота рахування* в режимах ділення і управління: $F_m = 1/2t_T$; (для ділення) та $F_m = kU_1/nt_T$ (для управління).

У режимі управління швидкість роботи асинхронного лічильника приблизно в n разів менша, ніж у режимі ділення [3].

Схему трирозрядного асинхронного двійкового підсумовувального лічильника на T -тригерах з динамічним управлінням по фронту подано на рис. 3.3. Рахункові імпульси U^+ подаються лише на T -вхід першого (молодшого) розряду, а інші тригери перемикаються асинхронно, отримуючи сигнали перенесення з інверсних виходів сусідніх молодших розрядів.

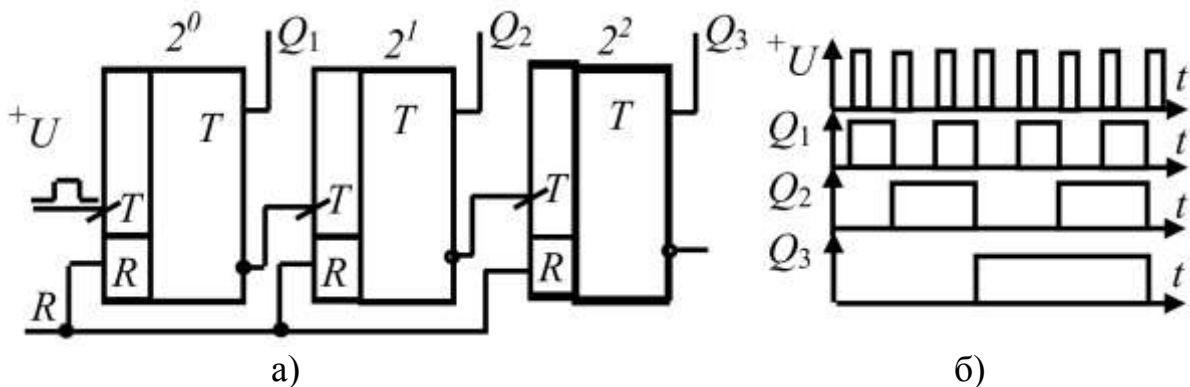


Рисунок 3.3 – Асинхронний підсумовувальний лічильник на тригерах з динамічним управлінням по фронту:
а) – схема; б) – часова діаграма роботи

Часові параметри асинхронного n -розрядного двійкового лічильника на тригерах з динамічним управлінням мають такі значення: $t_{pz} = t_T$, $t_{уст} = nt_T$, $F_m = 1/t_T$, $F_m = kU_1/nt_T$.

Основною перевагою асинхронних лічильників є простота їхньої конструкції: для збільшення розрядності достатньо підключити необхідну кількість тригерів. Проте, до недоліків таких лічильників відноситься відносно низька швидкість роботи в режимі управління, яка залежить від кількості розрядів, а також виникнення проміжних двійкових кодів під час послідовного перемикавання тригерів у новий стан.

Для зменшення часу перемикавання лічильника використовують паралельні перенесення (рис. 3.4). У кожному розряді синхронного лічильника є схема, яка аналізує стани всіх попередніх молодших тригерів і генерує функції перенесення згідно з певними логічними

співвідношеннями.

$$P_1 = UQ_1; P_2 = UQ_2Q_1; P_3 = Q_3Q_2Q_1; P_4 = Q_4Q_3Q_2Q_1$$

Коли надходить наступний лічильний імпульс $U+$, перемикаються лише ті тригери, для яких всі попередні (молодші) розряди знаходяться в стані «1». Час встановлення синхронного лічильника не залежить від кількості розрядів і має постійне значення:

$$t_{\text{в}} = t_{\text{ТТ}} + t_{\text{р}},$$

де $t_{\text{ТТ}}$ – час перемикання двоступінчатого тригера;

$t_{\text{р}}$ – час затримки поширення сигналу вентилям у колі перенесення.

Під час створення багаторозрядних синхронних лічильників виникають проблеми, пов'язані зі збільшенням кількості входів вентилів у колі перенесення та зростанням навантаження на виходи тригерів.

У двійковому віднімальному лічильнику кожен імпульс $U-$, що віднімається, зменшує значення на одиницю. Поточна кількість входних імпульсів n -розрядного двійкового віднімального лічильника визначається за такою формулою:

$$\sum U = N_n - N,$$

де N – значення коду на прямих виходах тригерів лічильника;

N_n – заздалегідь записане початкове число.

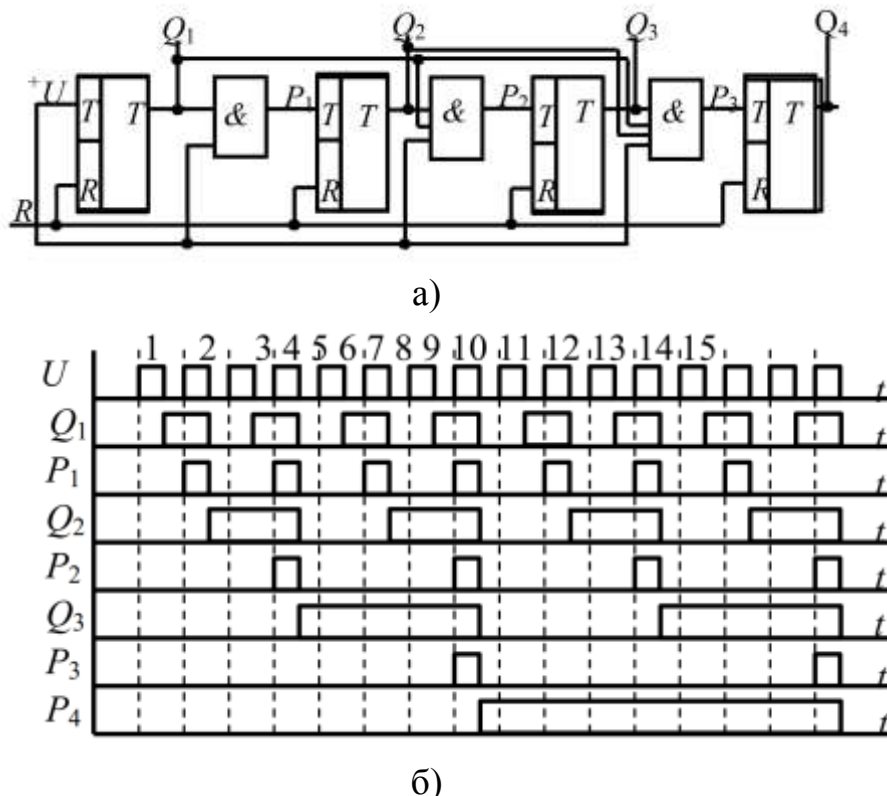


Рисунок 3.4 – Підсумовувальний лічильник з паралельними перенесеннями: а) – схема, б) – часова діаграма роботи

У віднімальних лічильниках сигнали віднімального зв'язку називаються позиками. Згідно з правилом двійкового віднімання, під час

надходження рахункового імпульсу U - позики зі старшого розряду з одиничним значенням виникає лише тоді, коли всі молодші тригери перебувають у нульовому стані. Після цього всі молодші тригери перемикаються в стан «1», а старші – в «0». Сигнали позики формуються на інверсних виходах двоступеневих тригерів або на прямих виходах тригерів з динамічним керуванням [16].

Параметри віднімального лічильника (модуль, ємність лічби, швидкість) збігаються з такими самими характеристиками підсумовувальних лічильників. Перед початком роботи всі тригери лічильника встановлюються в стан «1» за допомогою сигналу на загальному вході S , що формує вихідний код 1. Імпульс U - надходить на рахунковий вхід тільки першого молодшого розряду, а сигнали позики знімаються асинхронно з інверсних виходів тригера.

Схему двійкового асинхронного віднімального лічильника на двоступеневих тригерах можна побачити на рис. 3.5, а). Після надходження семи імпульсів, які віднімаються, всі тригери лічильника переходять у стан "0", формуючи вихідний код 0. Восьмий імпульс, що віднімається, перемикає лічильник у стан 1 (за умови використання піднімальної схеми). У віднімальному лічильнику на тригерах з динамічним керуванням сигнали позики знімаються з прямих виходів тригерів.

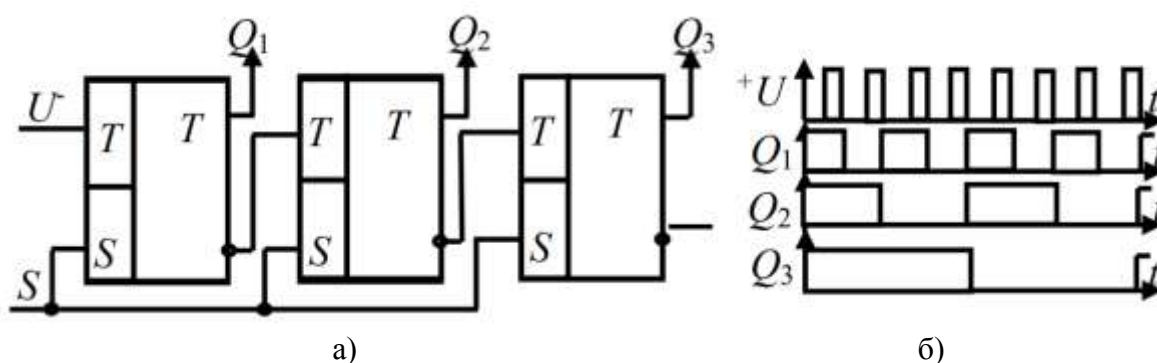


Рисунок 3.5 – Асинхронний віднімаючий лічильник на двоступеневих тригерах: а) – схема; б) – часова діаграма роботи

Двійкові **реверсивні лічильники** здатні рахувати в обох напрямках: вперед (завдяки підсумовувальним сигналам U^+) і назад (за допомогою віднімальних сигналів U^-). Поточне значення різниці підрахованих імпульсів визначається за таким співвідношенням.

$$\sum U^+ - \sum U^- = N - N_n,$$

де N – значення коду на прямих виходах тригерів лічильника;

N_n – заздалегідь записуване до лічильника початкове число.

Під час рахування має виконуватися умова

$$\sum U^- \leq N_n + \sum U^+ \leq 2^n - 1$$

Існують одноканальні та двоканальні реверсивні лічильники. В одноканальних реверсивних лічильниках підсумовувальні сигнали U^+ і віднімальні U^- по черзі надходять на загальний рахунковий вхід, а напрямок рахунку визначається конфігурацією кіл для віднімальних перенесень або позик. Для перемикання віднімальних зв'язків у реверсивному віднімальному лічильнику необхідні додаткові керівні сигнали.

Двоканальні реверсивні лічильники мають два рахункових входи: один призначений для підсумовування імпульсів U^+ , а інший – для віднімальних U^- . Перемикання кіл віднімальних зв'язків відбувається автоматично за рахунок рахункових сигналів: імпульси U^+ використовуються для перенесень, а імпульси U^- – для позик. Схему віднімального двійкового реверсивного лічильника можна побачити на рис. 3.6.

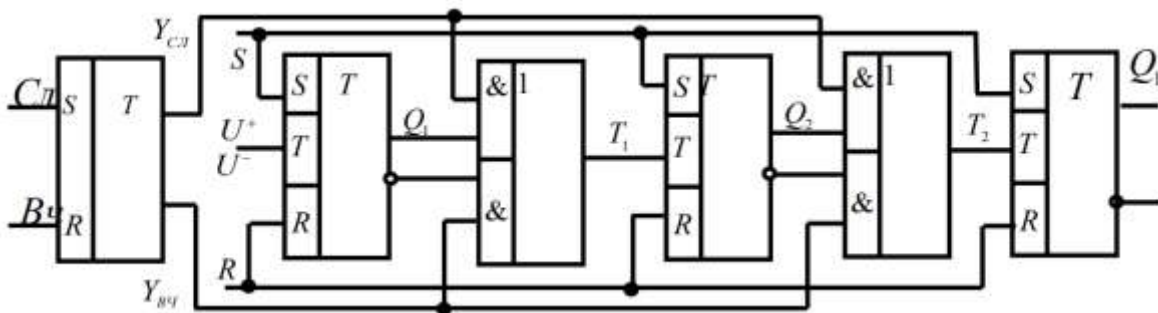


Рисунок 3.6 – Схема віднімального реверсивного лічильника

Міжрозрядні зв'язки реалізуються за допомогою логічних елементів типу І-АБО. Для визначення напрямку рахунку використовується віднімальний RS-тригер: з його прямого виходу отримується сигнал управління додаванням $Y_{ск}$ (який активує кола перенесення), а з інверсного виходу – сигнал керування відніманням $Y_{від}$ (який активує кола позики). На виходах елементів І-АБО, відомих як «схеми реверсу», формується сигнал T для рахункових входів старших розрядів:

$$T_i = Y_{д} Q_i \vee Y_{в} \overline{Q_i}, i = 1, 2, 3, \dots, n.$$

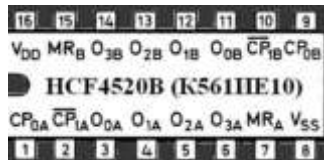
Таким чином, якщо керівний RS-тригер знаходиться в стані «1», лічильник переключиться в режим прямого рахунку вхідних імпульсів (підсумовування). В іншому випадку активується режим зворотного рахунку (віднімання). У обох режимах тригери перемикаються асинхронно. На практиці, з урахуванням можливостей мікросхем середнього рівня інтеграції, віднімальні реверсивні лічильники реалізуються у формі групової структури. Кожна група може бути подана мікросхемою віднімального реверсивного лічильника з паралельними перенесеннями й позиками. Між цими групами можуть встановлюватись послідовні або паралельні зв'язки [17].

Порядок виконання роботи (off-line частина)

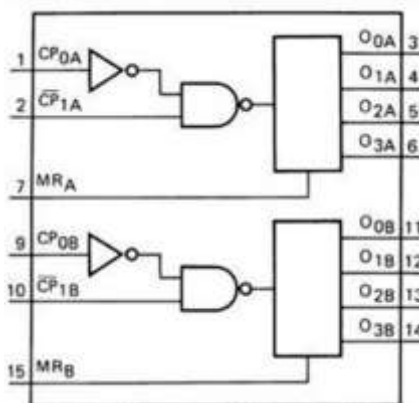
1. Вибрати мікросхему двійкового лічильника.

Як лічильник ми вибираємо мікросхему K561IE10 (рис. 3.7, а), яка містить два синхронних лічильника-дільника (рис. 3.7, б) на основі D-тригерів. Входи CP_0 і CP_1 взаємозамінні, але відрізняються протилежними активними рівнями, тому можна організувати лічення як по фронту, так і по спаду тактових імпульсів. Лічильник працює за низького рівня на вході MR. Технічні характеристики K561IE10 наведено в табл. 3.1, а в табл. 3.2 наведено стани лічильників K561IE10.

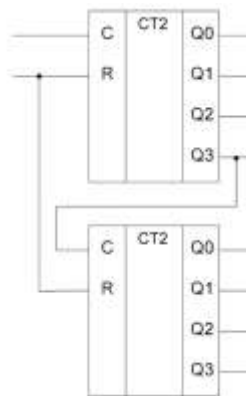
Можливе послідовне з'єднання двох лічильників (рис. 3.7, в) для збільшення розрядності. З'єднання двох синхронних лічильників буде асинхронним. Вихід O_{3A} потрібно з'єднати з входом CP_{1B} , подавши на CP_{0B} напругу низького рівня.



а)



б)



в)

Рисунок 3.7 – Розпіновка (а), функціональна схема (б) та схема підключення для збільшення розрядності (в) лічильника на K561IE10

Таблиця 3.1 – Технічні дані K561IE10

Кількість розрядів	2x4
Входи керування	C,R, \bar{C}
Керування по входу C	\downarrow
Напруга живлення	3...15 В
Струм споживання за максимальної напруги живлення	0,7 мА
Час затримки	2000 нс
Вихідний струм низького рівня	0,16 мА
Температура навколишнього середовища	-45...+85°C

Таблиця 3.2 – Стан лічильників К561ІЕ10

Вхід			Режим
С	$\overline{E\overline{C}}$	R	
1	1	0	Лічильник працює
0	1	0	Лічильник працює
1	X	0	Код без змін
X	1	0	Код без змін
1	0	0	Код без змін
1	1	1	Код без змін
X	X	1	Асинхронне скидання

2. Складаємо на макетній платі двійковий лічильник прямої лічби на К561ІЕ10 (рис. 3.8).

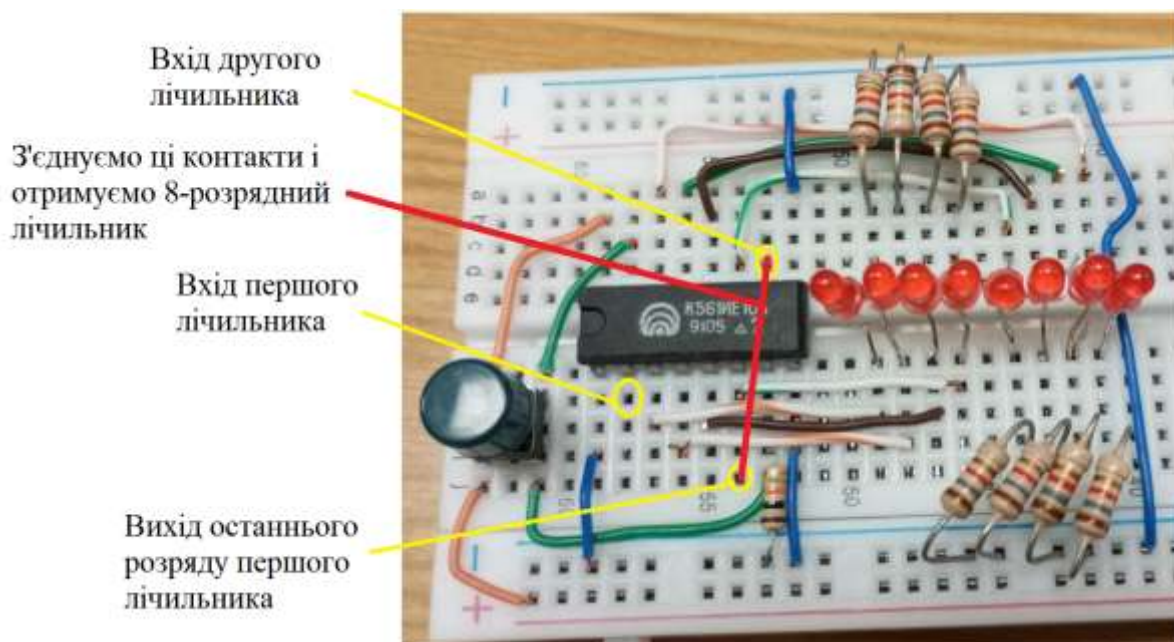


Рисунок 3.8 – Складена схема лічильника К561ІЕ10 на макетній платі

На вхід лічильника подаватимемо імпульсний сигнал з генератора. Зелена кнопка на рис. 3.8 – кнопка скидання лічильника.

3. Проводимо дослідження першого та другого 4-розрядного лічильника.

Результати наведено на рис. 3.9 та 3.10, відповідно.

Збільшуємо розрядність лічильника до 8 як показано на рис. 3.7, в) та досліджуємо 8-розрядний лічильник.

Результат наведено на рис. 3.11.

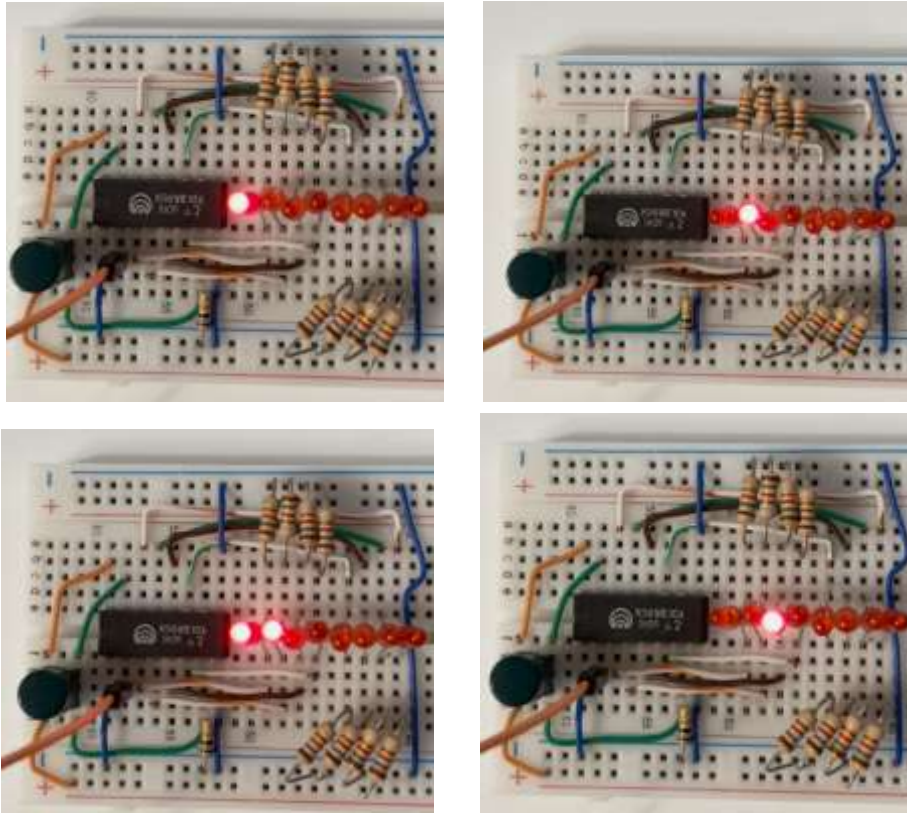


Рисунок 3.9 – Результат дослідження першого 4-розрядного лічильника.

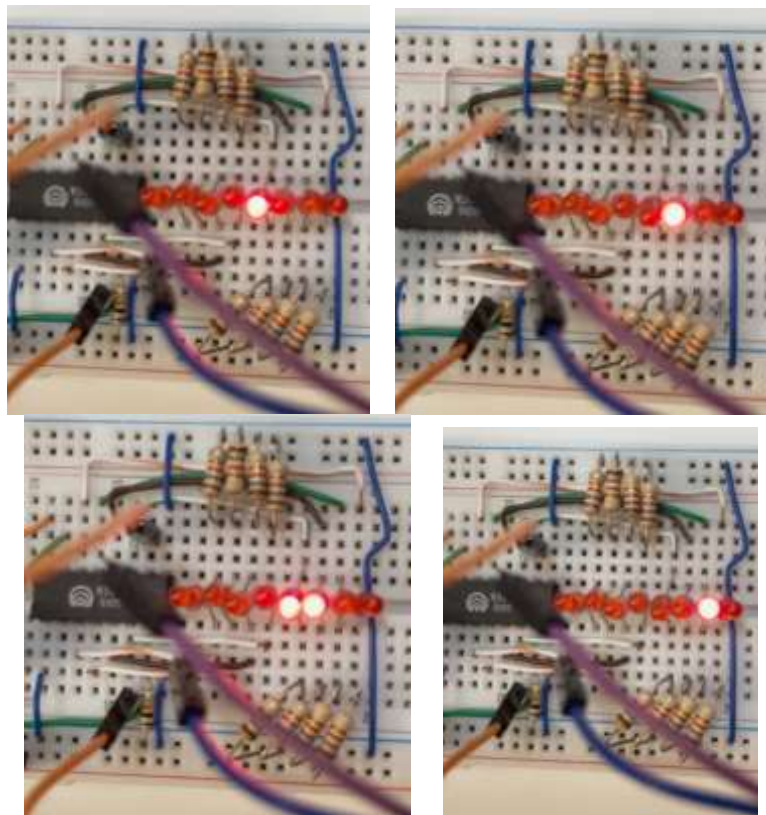


Рисунок 3.10 – Результат дослідження другого 4-розрядного лічильника.

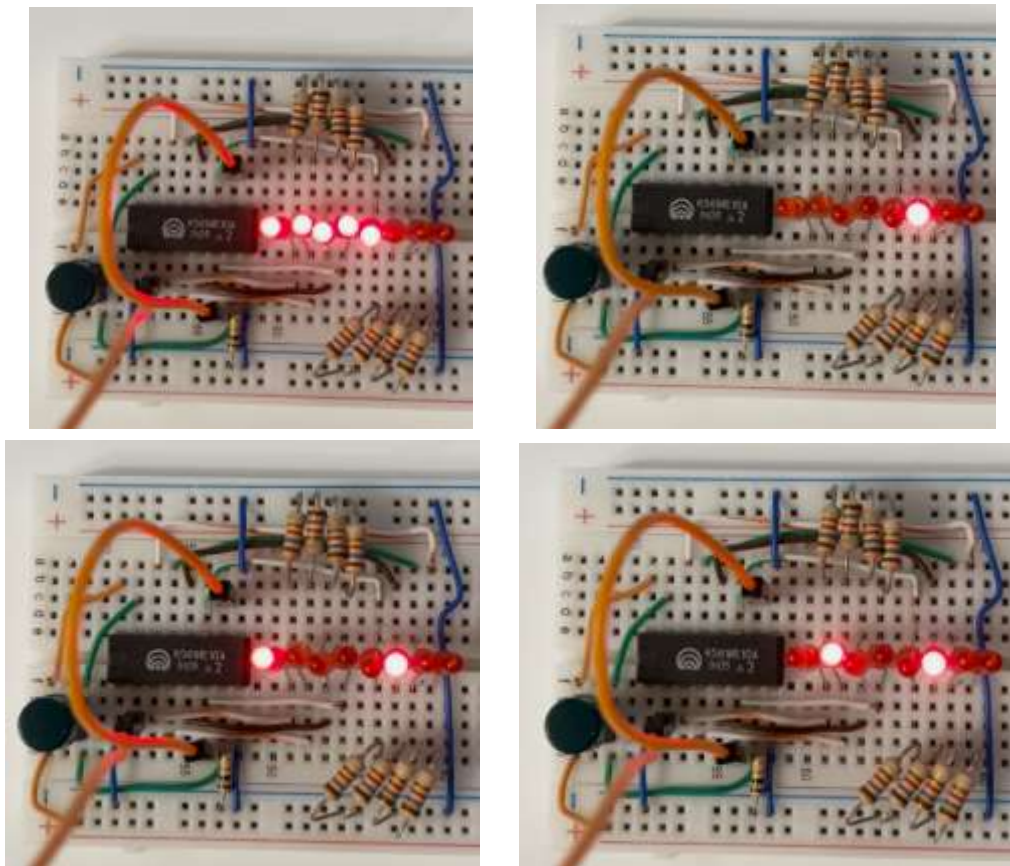


Рисунок 3.11 – Результат дослідження 8-розрядного лічильника.

4. Вибираємо мікросхему для дослідження лічильника в прямому та зворотному напрямку.

Для цих досліджень вибираємо мікросхему К155ІЕ7, яка являє собою двійковий чотирирозрядний лічильник в прямому та зворотному напрямку (рис. 3.12).

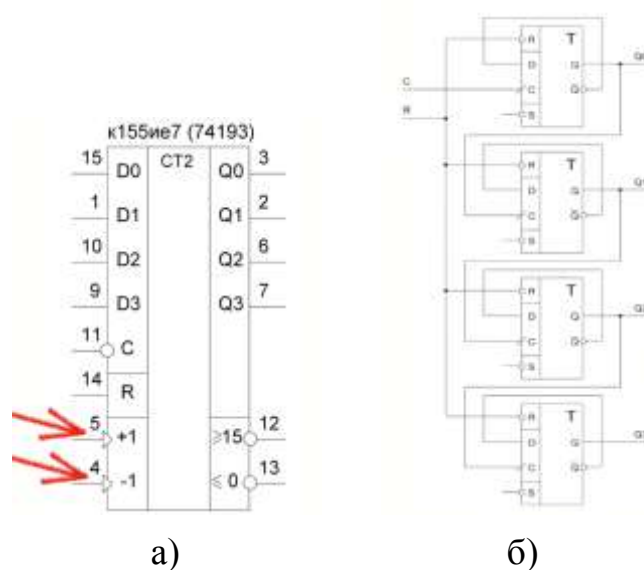


Рисунок 3.12 – Призначення виводів (а) та внутрішня структура (б) мікросхеми К155ІЕ7

5. Складаємо на макетній платі наш лічильник на K155IE7. Результат зображено на рис. 3.13. Жовта кнопка – кнопка скидання лічильника. Результат дослідження прямої лічби наведено на рис. 3.14 та зворотної лічби – на рис. 3.15.

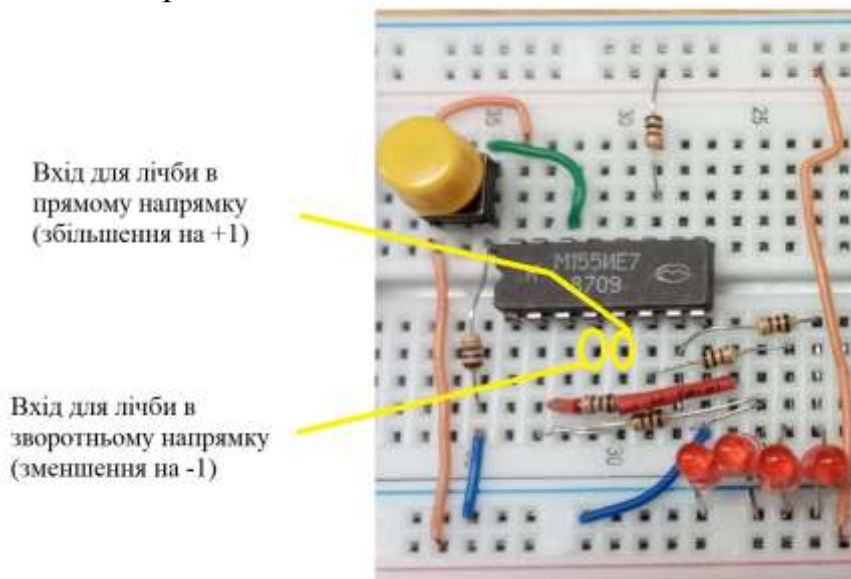


Рисунок 3.13 – Двійковий лічильник на K155IE7.

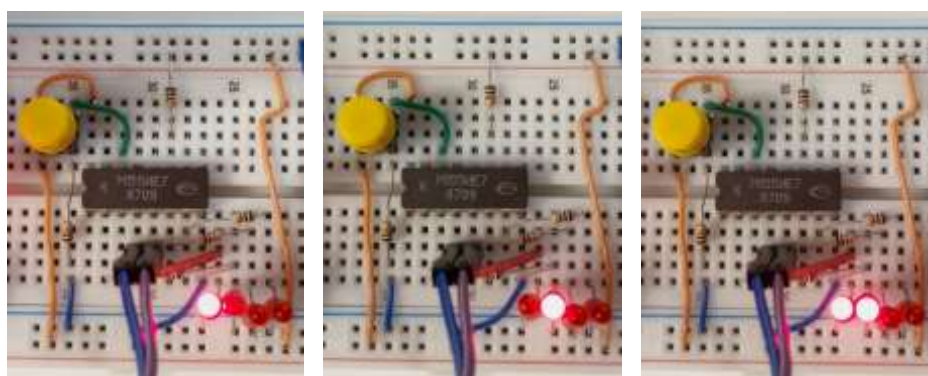


Рисунок 3.14 – Результат дослідження 4-розрядного лічильника в режимі прямої лічби.

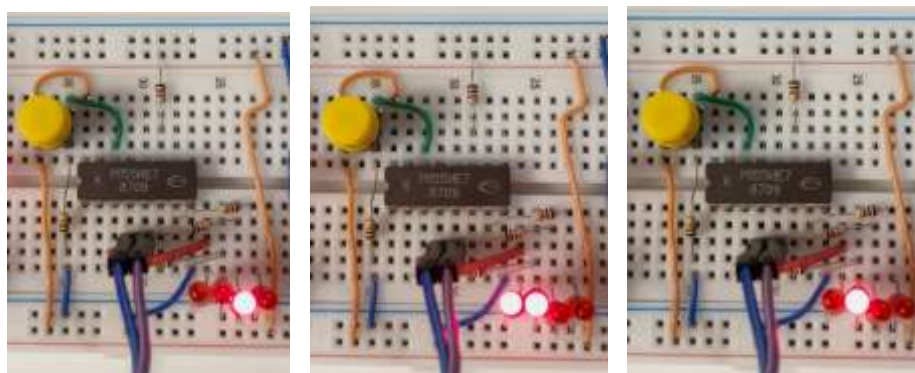


Рисунок 3.15 – Результат дослідження 4-розрядного лічильника в режимі зворотної лічби.

Порядок виконання роботи (on-line частина)

1. Складаємо досліджувану схему лічильника в програмі Multisim. Результат наведено на рис. 3.16.

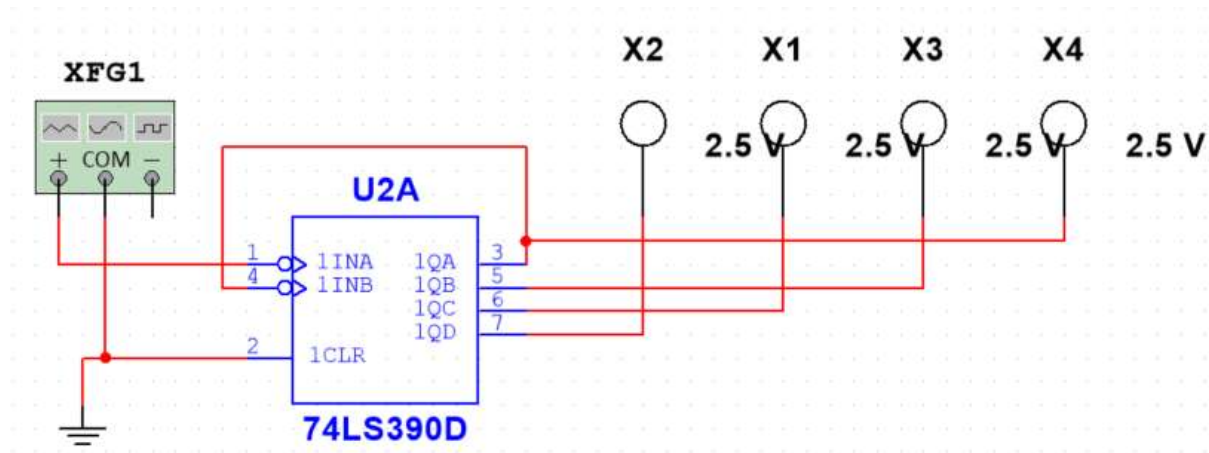


Рисунок 3.16 – Досліджувана схема лічильника в multisim

2. Проводимо дослідження схеми лічильника в програмному пакеті. Результати наведено на рис. 3.17.

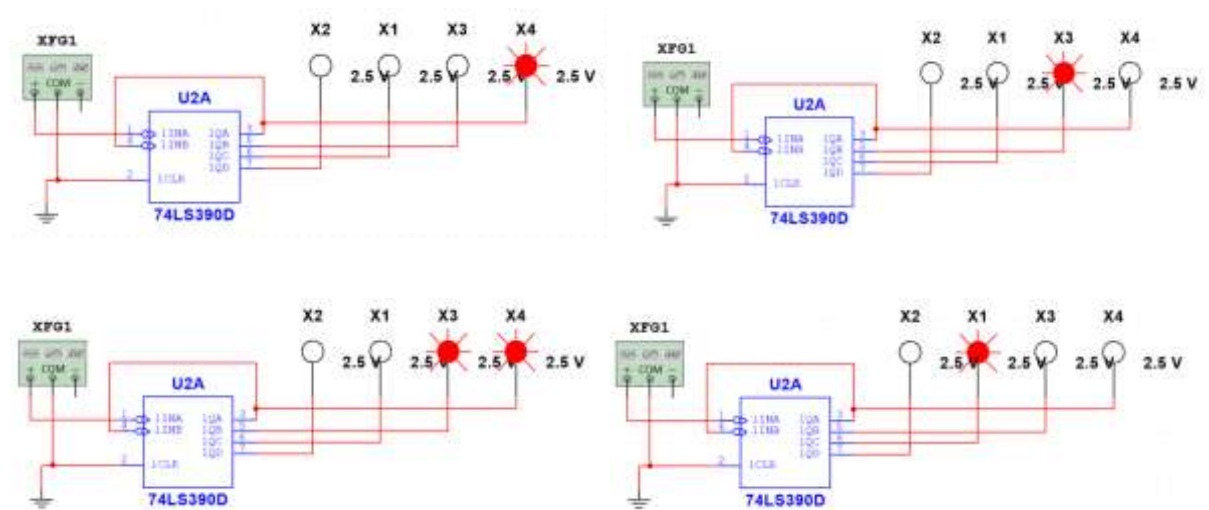


Рисунок 3.17 – Результати досліджень лічильника в multisim

3. Проводячи аналіз даних рис. 3.17, можна підтвердити правильне функціонування двійкового 4-розрядного лічильника імпульсів.

Варіанти завдань

1. Використання інтегральних мікросхем лічильників: дослідити роботу мікросхеми двійкового лічильника (наприклад, 74LS93).
2. Використання інтегральних мікросхем лічильників: дослідити роботу мікросхеми лічильника Джонсона (наприклад, 74LS194).
3. Дослідження реверсивного лічильника: побудувати схему реверсивного лічильника та дослідити його роботу.
4. Використання лічильника для поділу частоти: дослідити можливість використання двійкового лічильника для поділу частоти вхідного сигналу.
5. Дослідження лічильника Джонсона: побудувати схему лічильника Джонсона на D-тригерах та дослідити його роботу.
6. Дослідження кільцевого лічильника: побудувати схему кільцевого лічильника на D-тригерах та дослідити його роботу.
7. Каскадне з'єднання лічильників: побудувати схему каскадного з'єднання двох лічильників для збільшення розрядності.
8. Дослідження роботи лічильника з обмеженим діапазоном: скласти схему лічильника, який рахує в заданому діапазоні (наприклад, від 3 до 7).
9. Створення лічильника з ручним керуванням: додати кнопки для покрокового збільшення та зменшення значення лічильника.
10. Дослідження роботи лічильника з різними частотами: спостерігати роботу лічильника за різних частот вхідних імпульсів.

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного лічильника.
4. Результати досліджень (рисунок, фото, скрін екрана і т. д.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке лічильник?
2. Які галузі, сфери застосування лічильників?
3. Сформулюйте ознаки класифікації лічильників.
4. Наведіть основні параметри й ознаки класифікації лічильників.
5. Яким чином досягається підвищення швидкодії лічильника?
6. Поясніть принцип роботи реверсивного лічильника.
7. Чим відрізняється двійковий лічильник від десяткового?

8. Яким чином за допомогою трьох лічильників, що мають коефіцієнт перерахування 4, 5 та 6, можна одержати лічильник з коефіцієнтом перерахування: а) 20? б) 60?

Чому дорівнює коефіцієнт перерахування лічильника $K_{лч}$?

9. Припустимо, що п'ятирозрядний двійковий лічильник починає відраховувати зі стану 00000. Яке значення буде встановлено після 144-го вхідного імпульсу?

10. Як лічильник може бути використаний у цифрових пристроях, таких як годинники або таймери?

Лабораторна робота 4

Шифратори та дешифратори

Теоретична частина

Усі цифрові пристрої можна розділити на два основні класи: комбінаційні та послідовнісні.

Комбінаційні пристрої виконують функції, які залежать лише від поточної комбінації вхідних змінних і не враховують стан пристрою в попередньому моменті. Прикладом таких пристроїв є логічні елементи.

Навпаки, послідовнісні пристрої реалізують функції, які залежать не тільки від вхідних змінних у цей момент, але й від стану пристрою раніше: вони мають пам'ять. Найпростіші приклади таких пристроїв – тригери.

Серед найбільш поширених мікроелектронних цифрових пристроїв можна виділити:

- 1) дешифратори;
- 2) мультиплексори;
- 3) лічильники імпульсів;
- 4) регістри;
- 5) цифро-аналогові та аналого-цифрові перетворювачі.

Шифратор призначений для перетворення цифрових даних, поданих унітарним n -розрядним кодом, в еквівалентний двійковий m -розрядний код.

Унітарний код передбачає лише один активний стан змінної X_i серед вхідних сигналів $\{X_{n-1} \dots X_1, X_0\}$. Таким чином, шифратор виступає як перетворювач унітарного коду «1 з n » у двійковий паралельний код, де кількість виходів m прямо пов'язана з кількістю входів n за формулою 2^m .

Якщо $n=2^m$, що вказує на використання повного набору вихідних двійкових комбінацій Y_i , такий шифратор називається повним. Наприклад, шифратор 8-3 є повним, оскільки він реалізує всі можливі комбінації змінних X_i ($n=8$) у повний вихідний набір Y_i ($m=3$), що відповідає $2^3=8$.

У випадку неповного шифратора кількість входів n менша за всі можливі вихідні комбінації 2^m (причому завжди $n < 2^m$), що призводить до наявності деякої кількості невикористаних виходів. Прикладом неповного шифратора, який часто використовується, є шифратор 10-4, призначений для кодування десяткових чисел у двійково-десятковий код (ДДК) /8-4-2-1/. Цей шифратор може кодувати десяткові символи (0...9), наприклад, з клавіатури пульта керування. Схематичне позначення шифратора подано на рис. 4.1, а). Схему його внутрішньої будови зображено на рис. 4.1, б). Синтез як повного, так і неповного шифратора можна здійснити на основі логічних елементів.

Як приклад, розглянемо синтез повного шифратора 8-3 [18].

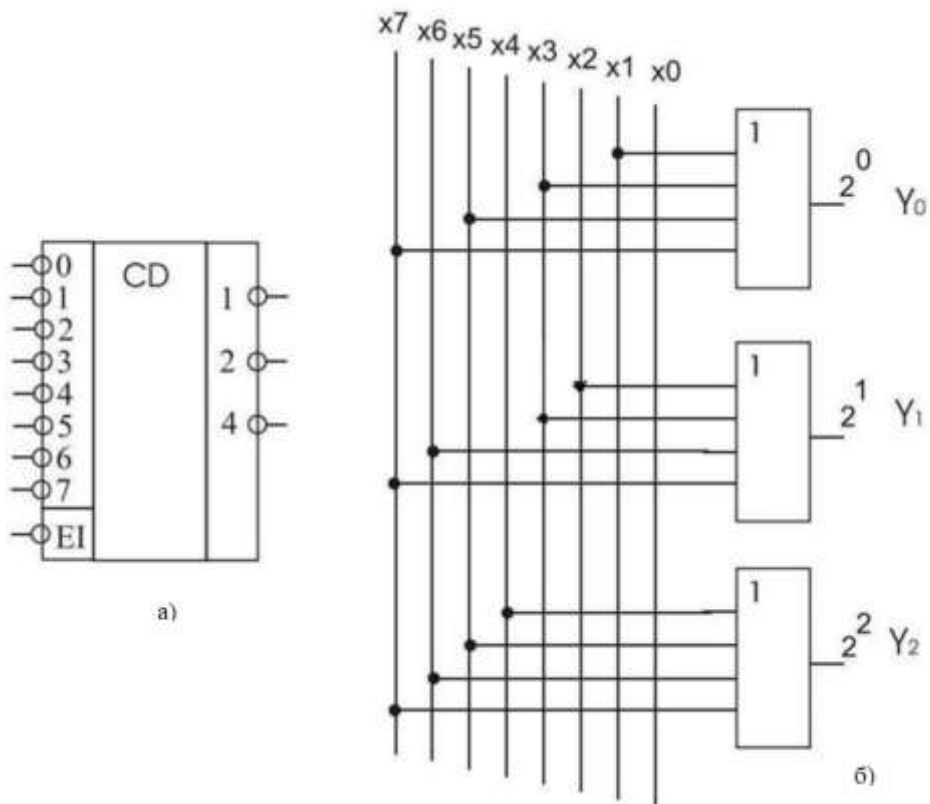


Рисунок 4.1 – Умовне позначення (а) та внутрішня будова (б) шифратора 8-3

Шифратор 8-3 може бути описаний таблицею істинності (табл. 4.1).

Таблиця 4.1 – Таблиця істинності шифратора 8-3

N	Входи								Виходи		
	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7	y_2	y_1	y_0
0	1	0	0	0	0	0	0	0	0	0	0
1	0	1	0	0	0	0	0	0	0	0	1
2	0	0	1	0	0	0	0	0	0	1	0
3	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	1	0	0	0	1	0	0
5	0	0	0	0	0	1	0	0	1	0	1
6	0	0	0	0	0	0	1	0	1	1	0
7	0	0	0	0	0	0	0	1	1	1	1

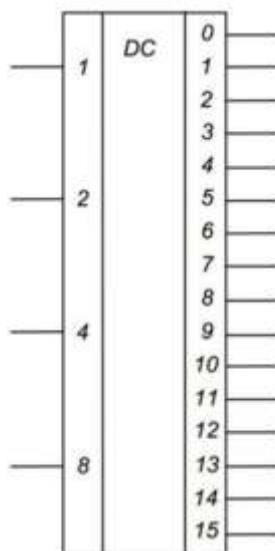
Окремі інтегральні мікросхеми рідко зустрічаються, адже переважно існують мікросхеми, які виконують комбіновані функції, такі як «шифратор/дешифратор». На практиці використовуються пріоритетні шифратори, в яких вихідний код завжди відповідає найбільшому номеру активного вхідного сигналу $\{x_0...x_n\}$. Наприклад, якщо на вході маємо $\{x_0=1, x_1=1, x_2=0, x_3=0, x_4=1\}$, то вихід буде $Y=4$.

Дешифратор (декодер) – це комбінаційний пристрій, який призначений для перетворення закодованих двійкових вхідних сигналів у сигнали управління для виконавчих пристроїв або систем відображення інформації.

Основна мета дешифратора полягає в розпізнаванні числа, поданого у вигляді позиційного n -розрядного двійкового коду. Найчастіше дешифратор з n входами і m виходами виконує функцію перетворення двійкового коду в унітарний код «1 з m », тобто виконує дію, протилежну шифратору. Для повного дешифратора справедливе співвідношення $m=2^n$, де m – порядковий номер виходу Y_i дешифратора.

Загалом, дешифратор має кілька входів, що відповідають кількості розрядів двійкових чисел, які потрібно декодувати, та кілька виходів. Кожній комбінації вхідних сигналів відповідає певна комбінація вихідних сигналів.

Умовне позначення **двійкового дешифратора** наведено на рисунку 4.2.



Рисунк 4.2 – Умовне позначення двійкового дешифратора

Він має чотири входи ($n=4$) з ваговими коефіцієнтами 1, 2, 4, 8, що відповідає чотирьом розрядам послідовного двійкового коду ($2^0, 2^1, 2^2, 2^3$), і шістнадцять виходів: від 0 до 15 ($N=2^n=2^4=16$).

Кожній комбінації нулів і одиниць на входах відповідає одиниця на відповідному виході. У двійково-десятковому дешифраторі з інверсними виходами, який зображено на рисунку 4.3, кожна з перших десяти двійкових комбінацій (двійково-десятковий код) приводить до нуля на відповідному виході. Такі дешифратори у формі інтегральних мікросхем використовуються для управління десятковими неоновими індикаторами, де десяткові знаки відображаються фігурними катодами неонові лампи або на семисегментному індикаторі [4].

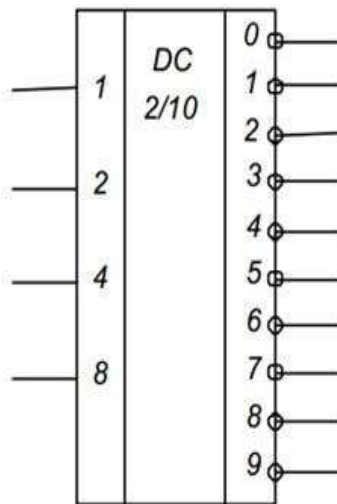


Рисунок 4.3 – Умовне позначення двійково-десятькового дешифратора

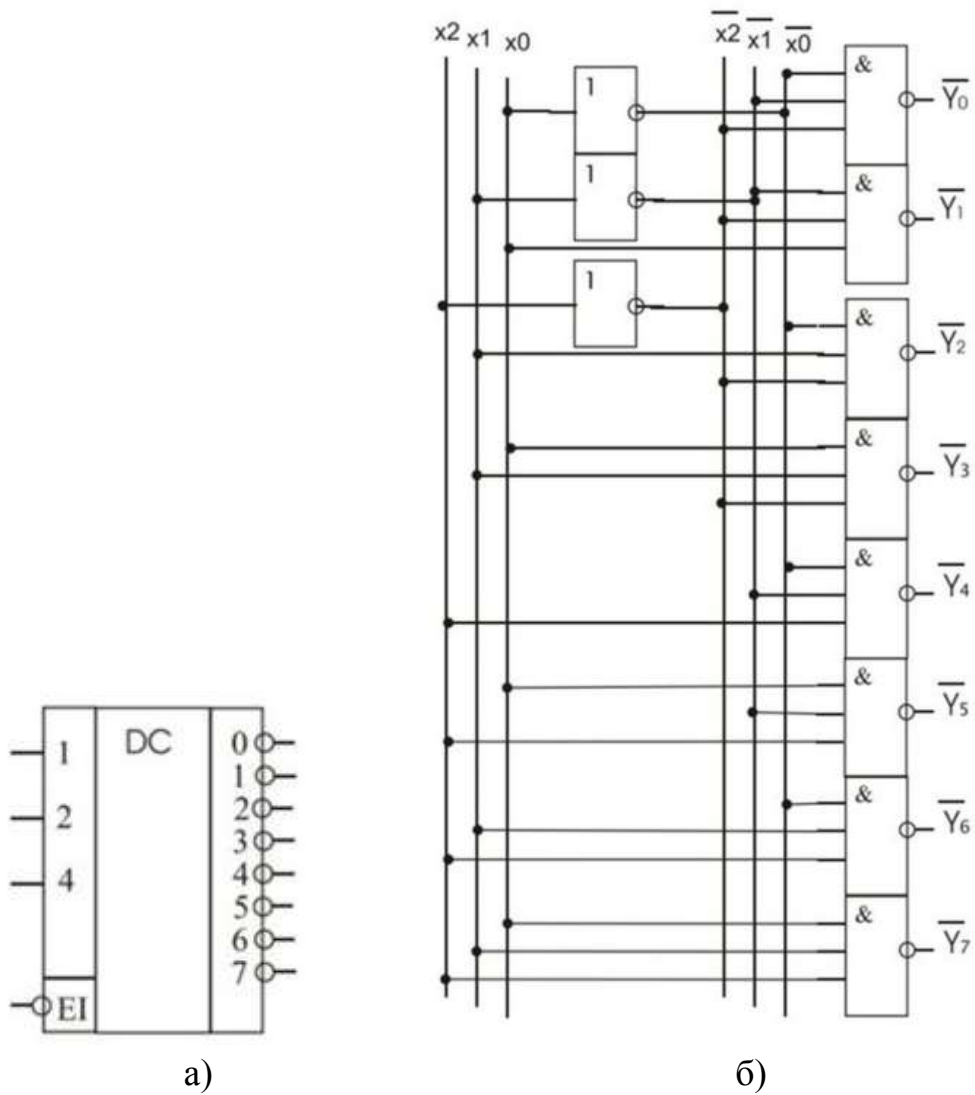


Рисунок 4.4 – Схематичне позначення (а) та внутрішня структура (б) дешифратора 3-8

Дешифратор 3-8 (рис. 4.4) може бути описаний таблицею істинності табл. 4.2.

Таблиця 4.2 – Таблиця істинності дешифратора із трьома входами

N	Входи			Виходи							
	x_2	x_1	x_0	y_0	y_1	y_2	y_3	y_4	y_5	y_6	y_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

У неповному дешифраторі кількість виходів m менша за 2^n ($m=2^n$). В інтегральному виконанні існують як повні (наприклад, К155ИД3, К155ИД7), так і неповні (К555ИД5, К555ИД10, К561ИД1), а також здвоєні (К155ИД4) дешифратори.

Мікросхеми функціональних комутаційних пристроїв для обробки цифрових сигналів, окрім інформаційних входів, можуть мати додаткові входи керування – ЕІ (дозвіл по входу) та ЕО (дозвіл по виходу). Завдяки цим входам дозволу можна також реалізувати нарощення розрядності, що дозволяє збільшувати кількість входів або виходів шляхом каскадування відповідних інтегральних схем.

У ситуаціях, коли необхідно створити дешифратор з великою кількістю виходів на основі дешевих дешифраторів з меншою кількістю виходів, використовують принцип каскадування. Цей підхід передбачає розподіл входів дешифраторів на групи, кожна з яких реалізує певну логічну функцію. Потрібно, щоб всі дешифратори мали керівні входи, тобто входи дозволу ЕІ. На рис. 4.5 зображено двокаскадне з'єднання двох дешифраторів 3-8 для створення повного дешифратора 4-16, який також має входи дозволу ЕІ. Каскад 1 активується у разі активних входів X_0, X_1, X_2 за умови, що $X_3=0$. Коли на входах дешифратора з'являється код, $\{x_3x_2x_1x_0\}=\{1000\}$, верхній каскад закриється, оскільки $X_3=EI=1$, а відкриється нижній каскад, який через інвертор отримує рівень дозволу $X_3=EI=0$.

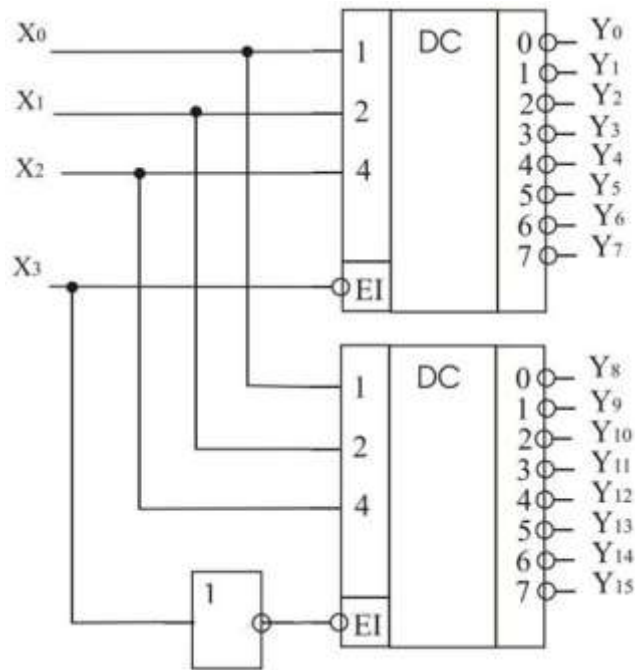


Рисунок 4.5 – Каскадний синтез дешифратора 4-16

Двійково-семисегментний дешифратор показано на рисунку 4.6, а.

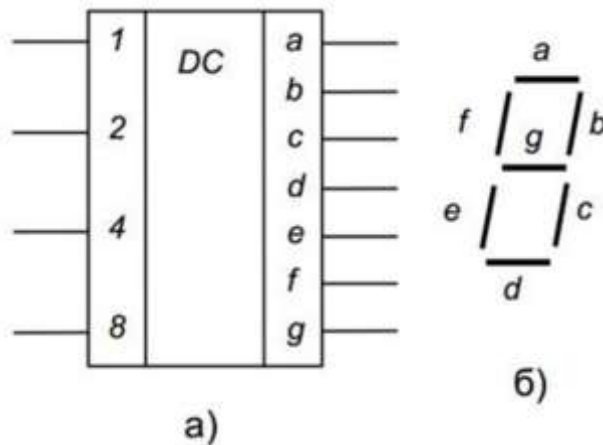


Рисунок 4.6 – Двійково-семисегментний дешифратор (а) і схема розміщення сегментів індикатора (б)

Він перетворює двійкову комбінацію вхідних сигналів у відповідну комбінацію вихідних сигналів, необхідну для активації певних сегментів семисегментного десяткового індикатора, як видно на рисунку 4.6,б.

Порядок виконання роботи (off-line частина)

1. Вибираємо мікросхему дешифратора для дослідження.

В цьому випадку ми вибрали мікросхему дешифратора двійкового коду в код для відображення десяткових чисел на семисегментному індикаторі. Це мікросхема К176ИД2. Необхідно зазначити, що К176ИД2

К176ИД1 мають лише одну різницю, яка пов'язана з типом семисегментного індикатора (спільний анод чи спільний катод).

На рис. 4.7 показано різницю під'єднання до мікросхем індикаторів різних типів.

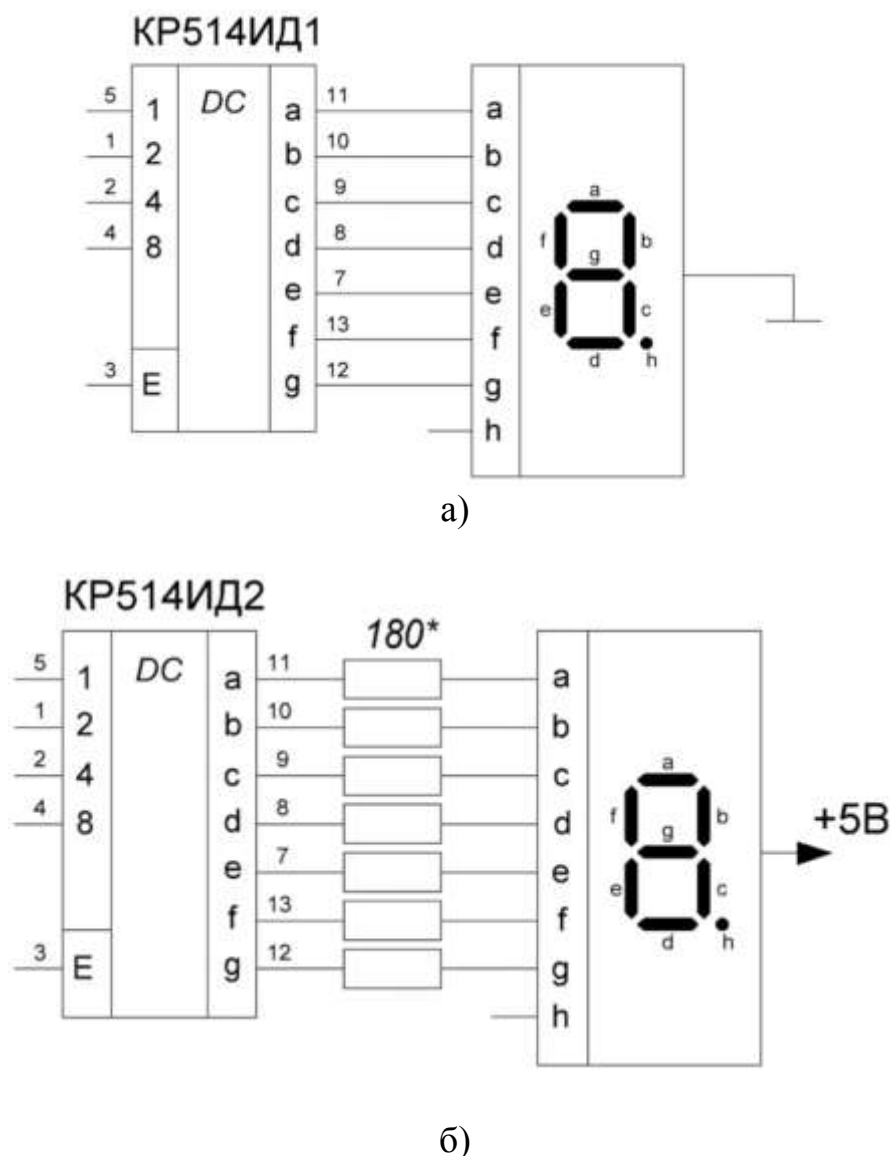


Рисунок 4.7 – Підключення дешифраторів до індикаторів різних типів:
 а) – індикатор зі спільним катодом, б) – індикатор зі спільним анодом

При червоному індикаторі зі спільним катодом струм сегменту становить близько 5 мА. За червоного індикатора зі спільним анодом та опору резистора 180 Ом, струм сегменту становить близько 15 мА.

Якщо вхід E=1, то індикатор показує цифру, якщо 0 – індикатор не світиться.

2. Складаємо на макетній платі схему (рис. 4.8) для дослідження дешифратора.

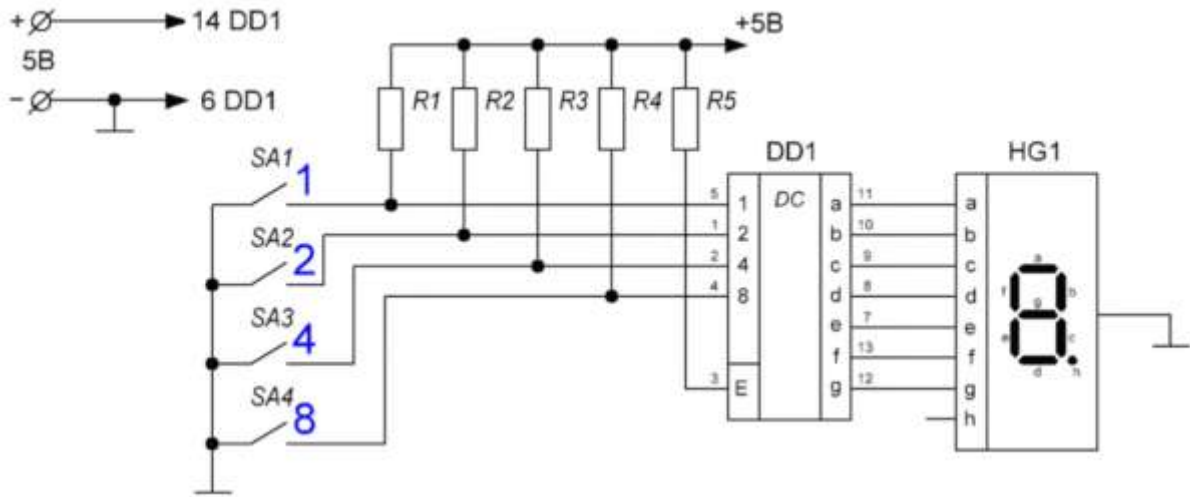


Рисунок 4.8 – Схема підключення дешифратора до семисегментного індикатора

- На схемі рис. 4.8 маємо такі елементи:
- HG1 – Індикатор зі пільним катодом;
 - DD1 – мікросхема дешифратора;
 - R1-R5 – резистори 1кОм;
 - SA1-SA4 – перемикачі.

Відповідно до схеми рис. 4.8, маємо складену на макетній платі схему, наведену на рис. 4.9.

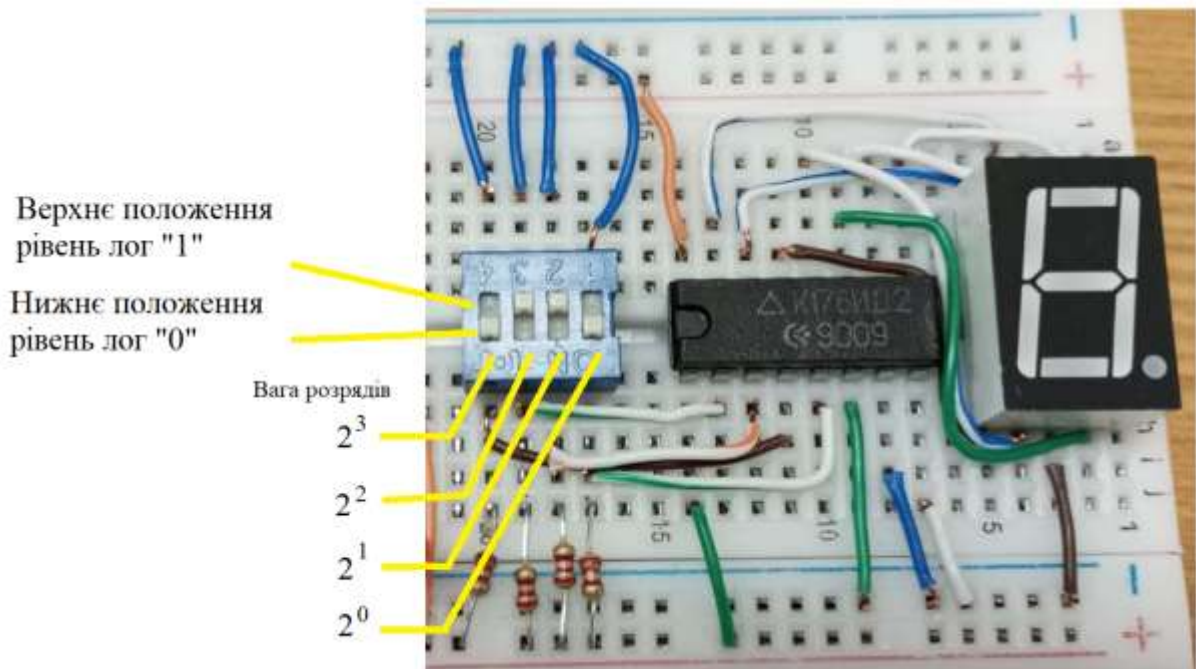


Рисунок 4.9 – Складена досліджувана схема дешифратора на макетній платі

3. Проводимо дослідження дешифратора, шляхом перемикання ключів. Вагу розрядів наведено на рис. 4.9. Результати досліджень наведено на рис. 4.10.

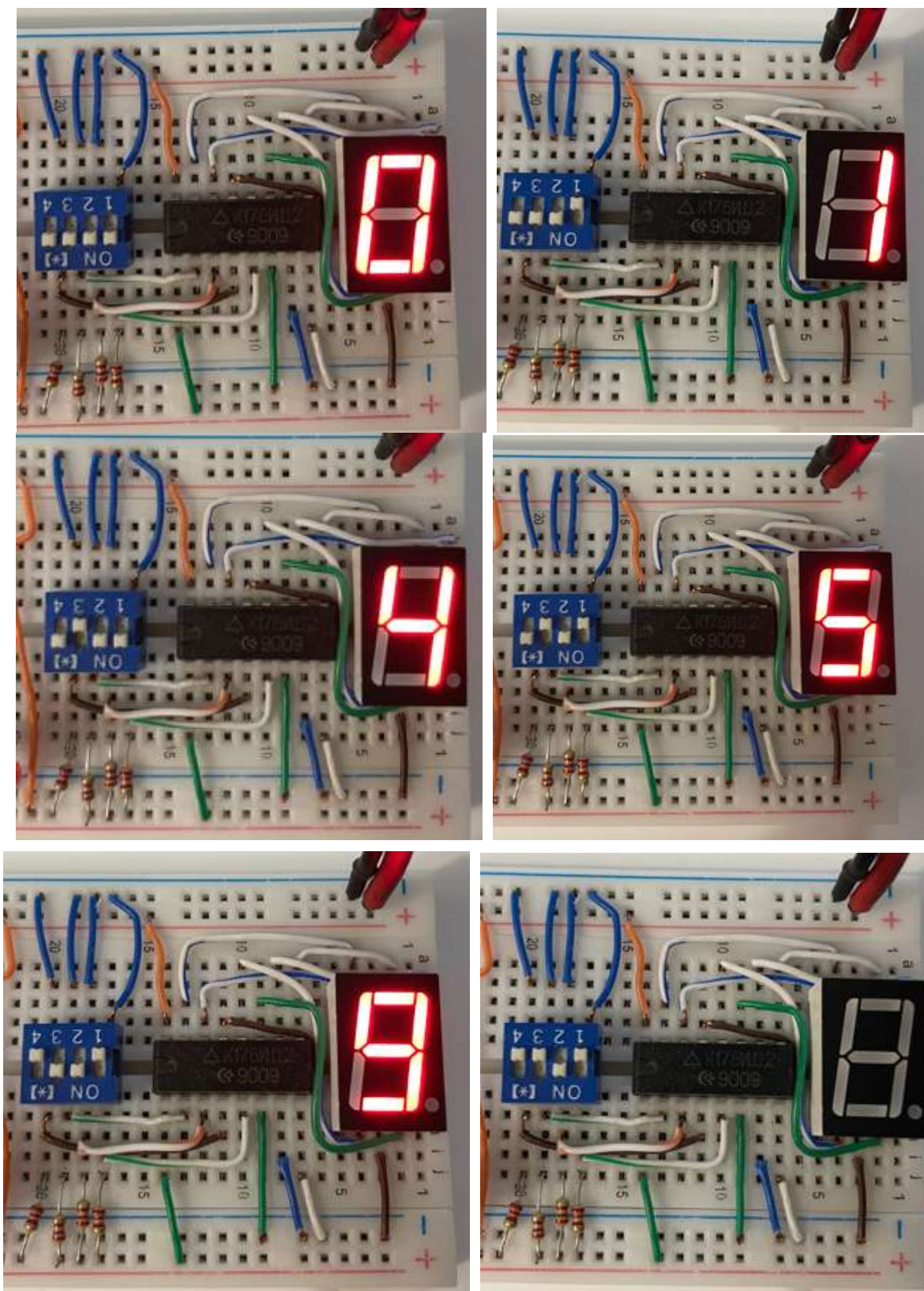


Рисунок 4.10 – Результати досліджень дешифратора.

Результати досліджень (рис. 4.10) показують, що кодова комбінація двійкового коду (в нашому випадку синій перемикач: положення вгору – логічна одиниця, положення вниз – логічний нуль) відповідає десятковому значенню на семисегментному індикаторі.

Порядок виконання роботи (on-line частина)

1. Складаємо досліджувану схему, в нашому випадку це двійковий дешифратор, який перетворює двійковий код в код для семисегментного індикатора, щоб користувачі могли бачити відображення десяткових цифр. Дослідження дешифратора в Multisim починається з вибору відповідної мікросхеми з бібліотеки компонентів. Моделювання дешифратора дозволяє візуалізувати його роботу та перевірити таблицю істинності. Під час моделювання дешифратора важливо правильно підключити вхідні та вихідні сигнали. Вхідні сигнали дешифратора можуть бути подані двійковим кодом, а вихідні – активним рівнем на одному з виходів. Використання логічного аналізатора в Multisim допомагає відстежувати зміни сигналів на входах і виходах дешифратора. В процесі моделювання дешифратора потрібно звернути увагу на час затримки сигналу, особливо за високих частот. Дослідження дешифратора в Multisim містить перевірку його роботи за різних комбінацій вхідних сигналів. Моделювання дешифратора дозволяє виявити можливі помилки в схемі та виправити їх до фізичної реалізації. Моделювання дешифратора дозволяє оптимізувати його роботу для конкретного застосування.

Досліджувану схему наведено на рис. 4.11.

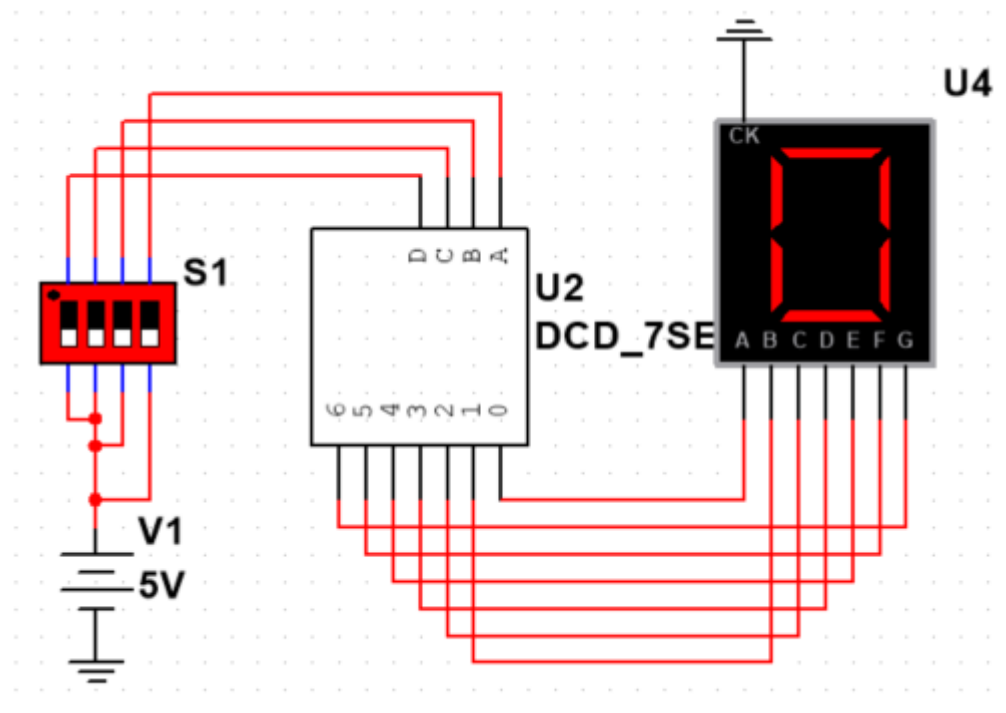


Рисунок 4.11 – Досліджувана схема в дешифратора в multisim

2. Проводимо дослідження схеми дешифратора в програмному пакеті. Результати наведено на рис. 4.12.

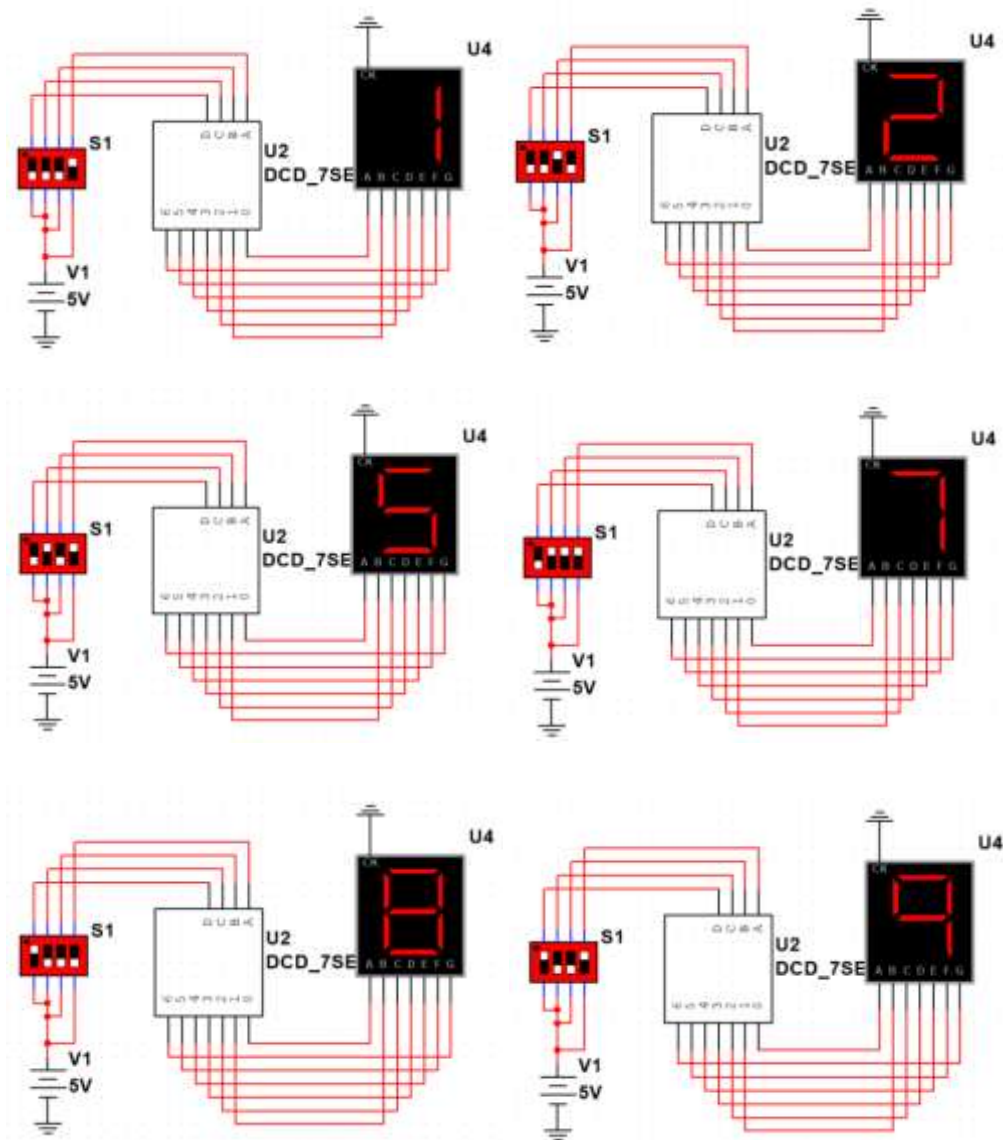


Рисунок 4.12 – Результати досліджень дешифратора в multisim

3. Проводячи аналіз даних (рис. 4.12), можна підтвердити правильне функціонування двійкового дешифратора для перетворення двійкового коду для відображення десяткових чисел на семисегментному індикаторі.

Варіанти завдань

1. Дослідження дешифратора 2-в-4: Побудувати схему дешифратора 2-в-4 на логічних елементах та дослідити його таблицю істинності.
2. Дослідження шифратора 4-в-2: Побудувати схему шифратора 4-в-2 на логічних елементах та дослідити його таблицю істинності.

3. Використання інтегральних мікросхем: дослідити роботу інтегральної мікросхеми дешифратора (наприклад, 74LS139) та шифратора (наприклад, 74LS147).

4. Використання інтегральних мікросхем: дослідити роботу інтегральної мікросхеми шифратора (наприклад, 74LS147).

5. Каскадне з'єднання дешифраторів: Побудувати схему каскадного з'єднання двох дешифраторів 2-в-4 для отримання дешифратора 3-в-8.

6. Використання дешифратора для керування світлодіодами: Побудувати схему, де дешифратор використовується для керування вісьмома світлодіодами.

7. Дешифратор як генератор функцій: Дослідити можливість використання дешифратора як генератора логічних функцій.

8. Шифратор пріоритету: Побудувати схему шифратора пріоритету та дослідити його роботу.

9. Використання дешифратора для керування семисегментним індикатором: побудувати схему, де дешифратор використовується для відображення цифр на семисегментному індикаторі.

10. Використання шифратора для кодування сигналів датчиків: Побудувати схему, де шифратор використовується для кодування сигналів від кількох датчиків.

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного шифратора або дешифратора.
4. Результати досліджень (рисунок, фото, скрін екрана, таблиці відповідності і т. д.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке шифратор і яка його основна функція?
2. Яка різниця між шифратором і дешифратором?
3. Які основні типи шифраторів існують (наприклад, двійкові, десяткові)?
4. Як працює двійковий шифратор?
5. Які входи та виходи має шифратор n -до- m ?
6. Яким чином реалізується дешифратор на базі логічних елементів?
7. Які основні застосування шифраторів у цифрових системах?
8. Як шифратори можуть використовуватися у системах адресації пам'яті?

9. Які особливості роботи шифраторів з активними високими та низькими рівнями?
10. Як здійснюється реалізація шифраторів та дешифраторів на ПЛІС (програмованих логічних інтегральних схемах)?
11. Яка різниця між пріоритетним і звичайним шифратором?
12. Як шифратор поводить себе у випадку одночасної активації кількох входів?
13. Що таке сигнали дозволу (Enable) у шифраторах і навіщо вони потрібні?
14. Як забезпечити коректну роботу дешифратора за некоректного або забороненого коду?
15. Як виглядає таблиця істинності дешифратора 3-в-8?
16. Які логічні елементи найчастіше використовуються для побудови шифраторів і дешифраторів?
17. У чому полягає роль дешифратора в управлінні пристроями введення/виведення?
18. Які особливості використання дешифраторів у багаторівневій пам'яті або багатосегментній адресації?
19. Як реалізуються шифратори/дешифратори мовами опису апаратури, наприклад, VHDL або Verilog?
20. Які існують стандартизовані мікросхеми шифраторів і дешифраторів (наприклад, 74147, 7442)?

Лабораторна робота 5

Мультиплексори та демультиплексори

Теоретична частина

Мультиплексори та демультиплексори – це комбінаційні пристрої, призначені для комутації цифрових каналів під дією двійкового коду керівних сигналів.

Мультиплексор (MUX) – це схема, яка спрямовує сигнали з одного чи з декількох інформаційних входів (шини даних) на єдиний вихід. Іноді мультиплексори називають селекторами входів.

На умовному графічному позначенні замість аббревіатури MUX може використовуватися MS, що означає «мультиплексор-селектор».

Кожному інформаційному входу мультиплексора присвоюється двійковий порядковий номер, який називається адресою (у цьому випадку 00...11). За подачі синхронізувального імпульсу на вхід С, мультиплексор вибирає один з інформаційних входів, чия адреса визначається двійковим кодом на адресних входах A_0, A_1 , і підключає його до виходу Y.

На рисунку 5.1 зображено умовне графічне позначення (УГП) мультиплексора з чотирма інформаційними входами.

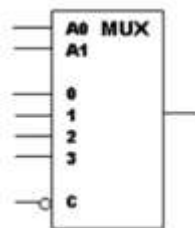


Рисунок 5.1 – УГП мультиплексора на чотири входи

Таблиця 5.1 – Таблиця істинності мультиплексора на чотири входи

A_1	A_0	C	Y
0	0	0	X_0
0	1	0	X_1
1	0	0	X_2
1	1	0	X_3

Коли синхронізувальний сигнал відсутній (вхід С інверсний, тобто активний нуль), передача даних між входами та виходом не відбувається. Це відображено в таблиці 5.1, де рядки для $C = 1$ ($Y = 0$) відсутні.

Згідно з таблицею 5.1, логічна функція (ЛФ) у диз'юнктивній нормальній формі (ДДНФ) виглядає таким чином:

$$Y = \bar{A}_1 \bar{A}_0 \bar{C} X_0 + \bar{A}_1 A_0 \bar{C} X_1 + A_1 \bar{A}_0 \bar{C} X_2 + A_1 A_0 \bar{C} X_3$$

Аналізуючи цей вираз, можна помітити, що комбінації адресних сигналів A_1, A_0 для кожного входу X у мультиплексорі ідентичні комбінаціям у дешифраторі з двома входами та активним рівнем вихідного сигналу. Отже, мультиплексор містить дешифратор адресних сигналів, який формує адресні комбінації $\bar{A}_1 \bar{A}_0, \bar{A}_1 A_0, A_1 \bar{A}_0, A_1 A_0$. З урахуванням цього, схему, що реалізує необхідну ЛФ з використанням дешифратора, подано на рисунку 5.2.

Для обробки даних більшої розрядності застосовують паралельне з'єднання мультиплексорів. Мультиплексори, подібно до дешифраторів, використовуються для синтезу складних логічних схем, що дозволяє зменшити кількість необхідних інтегральних мікросхем. На мультиплексорах з чотирма входами можна реалізувати будь-яку ЛФ трьох змінних, а на мультиплексорах з вісьмома входами – будь-яку ЛФ чотирьох змінних.

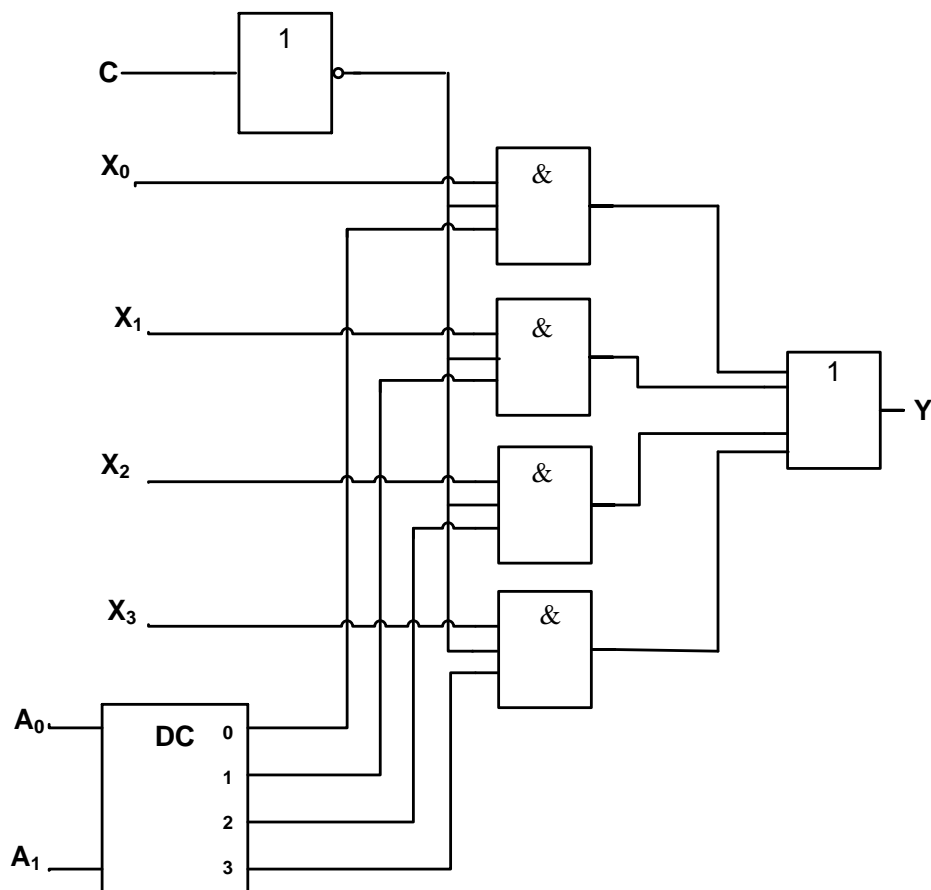


Рисунок 5.2 – Мультиплексор на основі дешифратора

Демультимплексор – це комбінаційна цифрова схема, яка розподіляє вхідний сигнал на один з декількох виходів, керуючись заданим кодом на керівних входах. Тобто, демультимплексор виконує функцію, обернену мультиплексору.

Умовне графічне позначення (УГП) демультимплексора з чотирма виходами подано на рис. 5.3. У таблиці функціонування (табл. 5.2) також наведено ЛФ для кожного вихідного сигналу.

Аналізуючи дані табл. 5.2 і ЛФ для виходів $Y_0 \dots Y_3$ видно, що цей демультимплексор також можна розглядати як дешифратор з додатковим входом синхронізації D. Тому синхронні дешифратори часто застосовують як демультимплексори, де інформаційні входи виконують роль адресних, а вхід синхронізації – роль інформаційного входу.

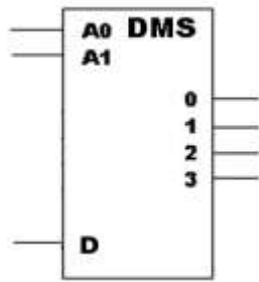


Рисунок 5.3 – УГП демультиплексора на чотири виходи

Таблиця 5.2 – Таблиця істинності мультиплексора на чотири входи

A ₁	A ₀	D	Вихід
0	0	1	$Y_0 = \bar{A}_1 \bar{A}_0 D$
0	1	1	$Y_1 = \bar{A}_1 A_0 D$
1	0	1	$Y_2 = A_1 \bar{A}_0 D$
1	1	1	$Y_3 = A_1 A_0 D$

Як згадувалося раніше, мультиплексор, окрім інформаційних входів, має адресні входи a_j , двійковий код на яких визначає, який з інформаційних входів буде підключений до виходу схеми. Таким чином, мультиплексор має 2^n+n входів і один вихід (де 2^n – кількість інформаційних входів, а n – кількість адресних входів). Керований мультиплексор додатково має вхід дозволу мультиплексування E.

На рисунку 5.4, а) подано модифіковану схему мультиплексора 4-1. Для створення мультиплексора 2^n-1 необхідний багатовходовий логічний елемент типу І-АБО (рис. 5.4, б), який забезпечує передачу одного з 2^n сигналів з інформаційної шини даних, а також дешифратор для керування комутацією.

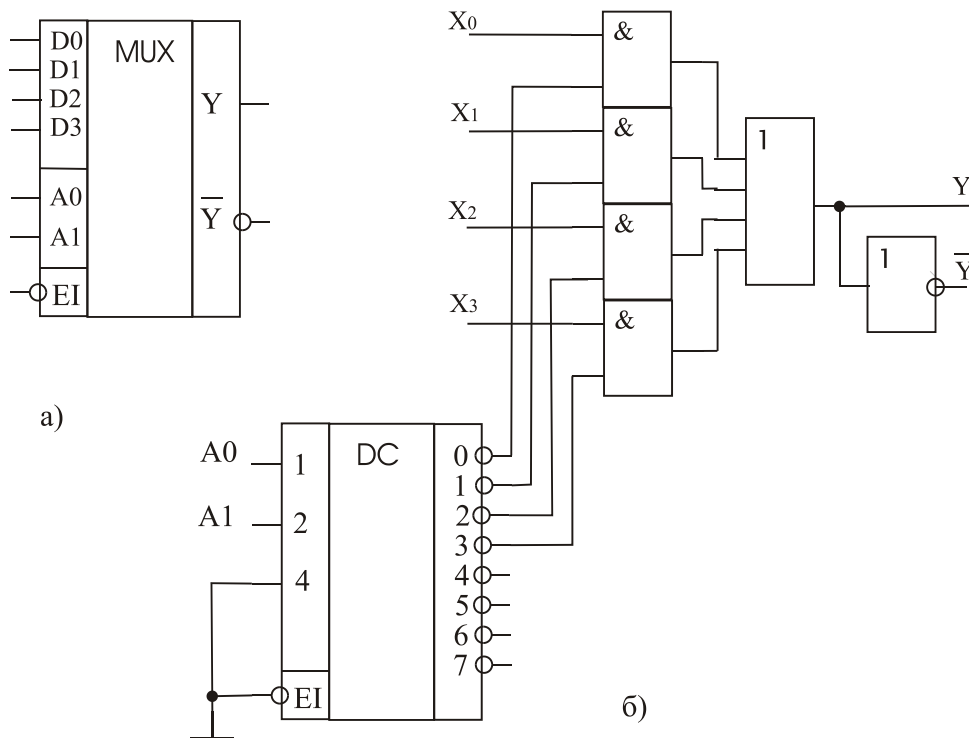


Рисунок 5.4 – Схематичне позначення (а) та структура (б) мультиплексора 4-1

Він використовуючи $n=2$ адресних сигнали a_1 і a_2 вибирає один з $2^2 = 4$ вхідних сигналів X_i . Логічна функція мультиплектора 4-1 описується таким чином:

$$Y = \sum_{i=0}^{2^A-1} x_i m_i(a)$$

Цю логічну залежність можна узагальнити для довільної кількості адресних входів:

$$Y = \sum_{i=0}^{2^A-1} x_i m_i(a)$$

де x_i – вхідні інформаційні сигнали,

$m_i(a)$ – мінтерми n адресних змінних .

У промисловості доступні інтегральні схеми мультиплексорів з різною кількістю адресних входів, найчастіше n набуває значення 2, 3, 4 тощо. Під час створення пристроїв з використанням мультиплексорів потрібно враховувати, що більшість з них призначені для комутації цифрових сигналів, хоча існують деякі мікросхеми (особливо КМОН), які здатні комутувати й аналогові сигнали. Для отримання детальної інформації про конкретну мікросхему необхідно ознайомитися з її технічною документацією.

Для комутації (мультиплексування) великої кількості сигналів використовується принцип каскадування. Наприклад, на рис. 5.5 показано каскадне з'єднання двох мультиплексорів 4-1 для мультиплексування 8-розрядної шини даних.

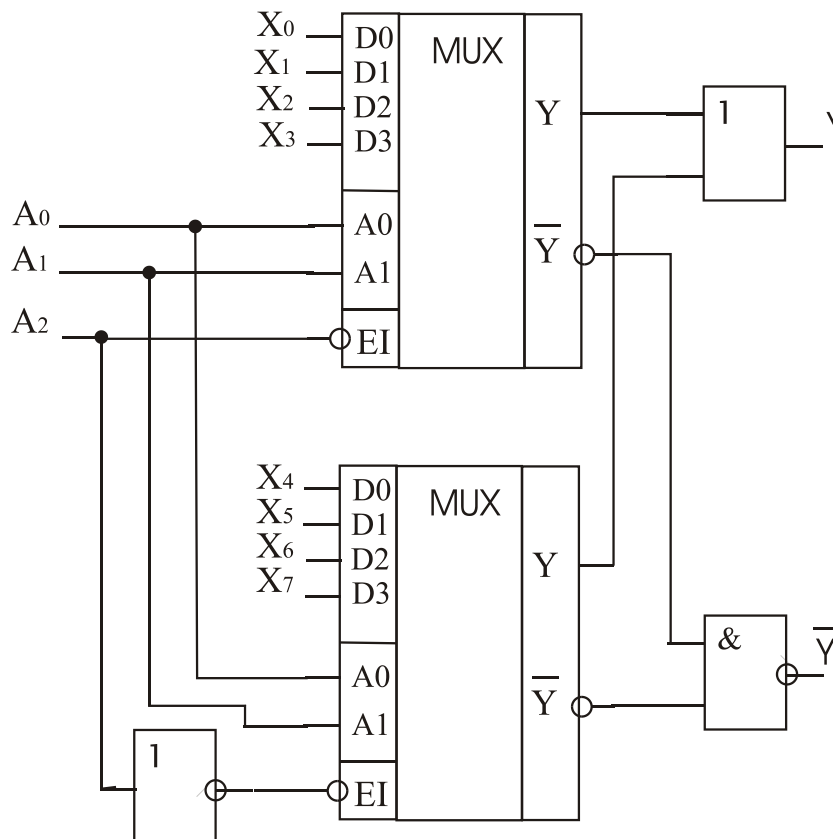


Рисунок 5.5 – Синтез мультиплектора 8-1 за допомогою мультиплексорів 4-1

Розподіл шини адрес здійснюється за принципом, аналогічним каскадному з'єднанню дешифраторів. Мультиплексори використовуються для реалізації різноманітних цифрових, а іноді й цифро-аналогових схем. Зокрема, вони застосовуються для створення: комутаційних пристроїв з багатьма змінними, багатоканальних комутаторів цифрових та аналогових сигналів, запам'ятовувальних пристроїв, генераторів послідовностей двійкових чисел тощо. Крім того, мультиплексор можна використовувати для перетворення паралельного коду, що подається на інформаційні входи, в послідовний, шляхом потактної зміни стану адресних входів за допомогою лічильника імпульсів. Цей метод також дозволяє послідовно опитувати сигнали на 2^n інформаційних входах мультиплексора.

Демультиплексор (DMX) виконує функцію, обернену мультиплексору, тобто передає (комутує) сигнал з єдиного інформаційного входу на один з 2^n виходів, залежно від коду на n -адресних входах. Демультиплексор можна реалізувати на основі дешифратора з n -входами, де вхід дозволу E використовується як інформаційний. Таким чином, демультиплексор є розподільником цифрових сигналів, керованим двійковим входом.

Приклад схеми демультиплексора 1-8 зображено на рисунку 5.6. Паралельний код, що керує роботою демультиплексора, подається на входи дешифратора, які стають адресними, а виходи дешифратора під'єднуються до входів кон'юнкторів. У загальному випадку демультиплексор реалізує логічну функцію виду

$$Y_i = X m_i(a).$$

Інформаційний вхід X демультиплексора формується шляхом об'єднання других входів логічних елементів I . Таким чином, демультиплексор передає вхідний сигнал X на відповідний вихід Y_i лише через той елемент I , на другому вході якого присутня логічна одиниця, що надходить з виходу дешифратора.

Деякі інтегральні схеми ТТЛ, зокрема дешифратори К155ИД3, К155ИД4, К155ИД7, можуть виконувати функції демультиплексора, залежно від способу їх підключення. Ця функціональна подвійність мультиплексора та демультиплексора зручна для передачі двійкових сигналів на відстань, наприклад, по телефонних лініях або кабелях. У такій системі передачі даних мультиплексор виконує роль передавача-перетворювача паралельного коду в послідовний, а демультиплексор – роль приймача-перетворювача послідовного коду в паралельний. За наявності лінії керування обома перетворювачами забезпечується синхронна робота системи. Перевагою такої системи передачі даних на відстань, незважаючи на її відносно низьку швидкість, є економія коштів, що неминуче у разі паралельної передачі інформації.

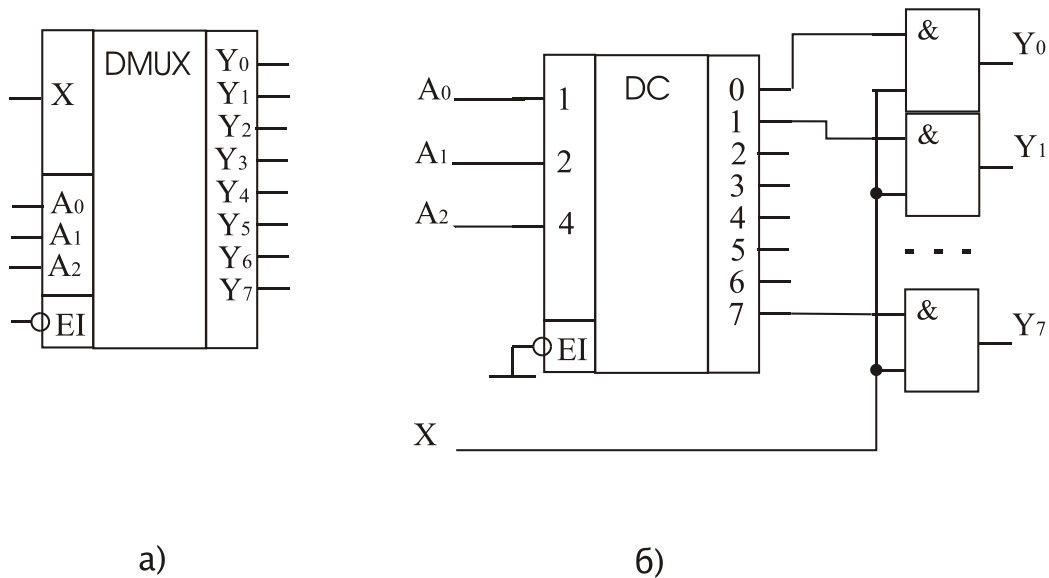


Рисунок 5.6 – Схематичне позначення (а) та структура (б) демультимплексора 1-8

Мультиплектори та демультимплектори широко використовуються в автоматичному багатоканальному електрозв'язку для передачі цифрових сигналів. Багатоканальний електрозв'язок дозволяє організувати велику кількість незалежних каналів на одній лінії зв'язку. Формування та розділення цифрових сигналів у багатоканальних системах з часовим розподілом здійснюється за допомогою мультиплексорів та демультимплексорів. Спрощена схема на рисунку 5.7 ілюструє принцип формування та розподілу багатоканального сигналу.

Вхідні сигнали, кожен з яких є послідовністю одиниць та нулів, передаються на вихід мультиплексора по черзі. Для кожного сигналу виділяється власний часовий інтервал, і ці інтервали не перетинаються.

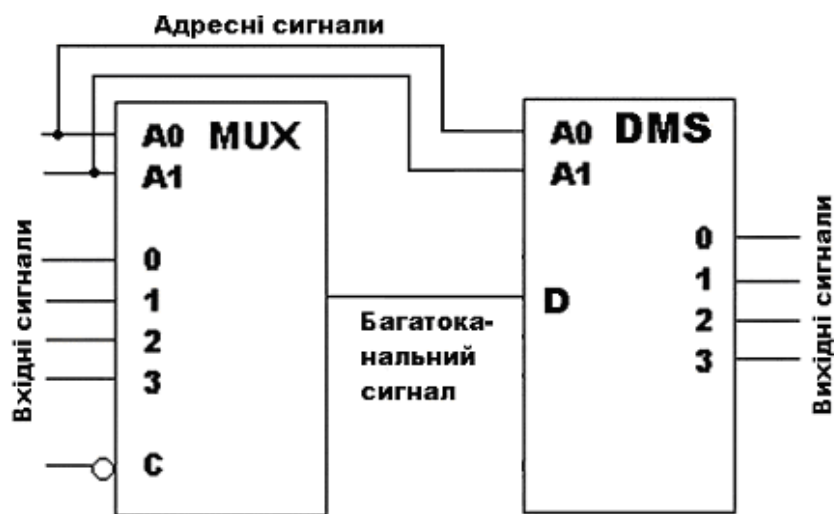


Рисунок 5.7 – Спрощена схема цифрового мультиплексорного каналу

На приймальній стороні демультимплексор використовується для розподілу сигналів. Для коректної роботи системи необхідно, щоб сигнали керування (адресації) та синхронізації були однаковими як для мультимплексора-передавача, так і для демультимплексора-приймача.

Мультимплексори та демультимплексори, разом з іншими комбінаційними пристроями (дешифраторами, шифраторами), застосовуються для комутації каналів та опитування окремих пристроїв з метою виявлення змін у їхньому стані (цей процес називається скануванням).

Порядок виконання роботи (off-line частина)

1. Вибираємо мікросхему мультимплексора для дослідження.

В цьому випадку ми вибрали мікросхему КМ155КП7. Іноземним аналогом мікросхеми К155КП7 КМ155КП7 є мультимплексори SN74151N, SN74151J).

На рис. 5.8 показано умовне графічне позначення КМ155КП7 та призначення виводів.

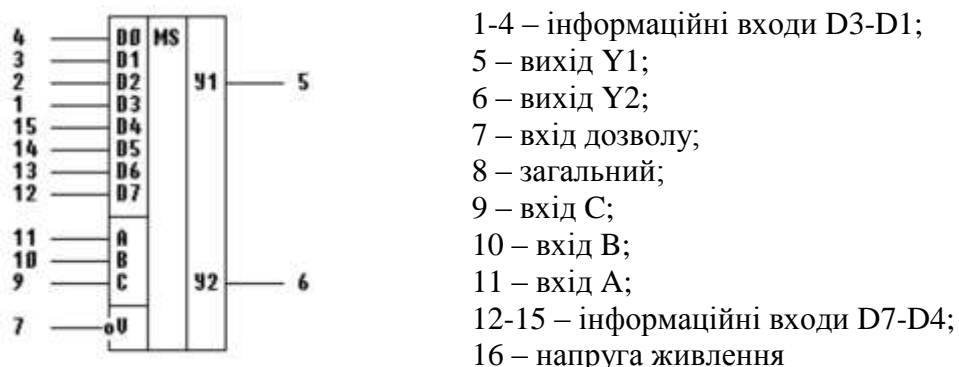


Рисунок 5.8 – Умовне графічне позначення та призначення виводів

Електричні параметри КМ155КП7 наведено в табл. 5.3. В табл. 5.4 наведено таблицю істинності для КМ155КП7.

Таблиця 5.3 – Електричні параметри КМ155КП7

Номінальна напруга живлення	5 В
Вихідна напруга низького рівня	не більше 0,4 В
Вихідна напруга високого рівня	не менше 2,4 В
Вхідний струм низького рівня	не більше -1,6 мА
Вхідний струм високого рівня	не більше 0,04 мА
Струм споживання	не більше 48 мА

Таблиця 5.4 – Таблиця істинності КМ155КП7

Вхід				Вихід
Вибір		Дозвіл		
С	В	А	U	Y
х	х	х	В	Н
Н	Н	Н	Н	D0
Н	Н	В	Н	D1
Н	В	Н	Н	D2
Н	В	В	Н	D3
В	Н	Н	Н	D4
В	Н	В	Н	D5
В	В	Н	Н	D6
В	В	В	Н	D7

2. Складаємо на макетній платі схему для дослідження цього мультиплексора (рис. 5.9). Необхідно зазначити, що для дослідження мультиплексора будуть використані генератори сигналів на мікросхемі NE555.

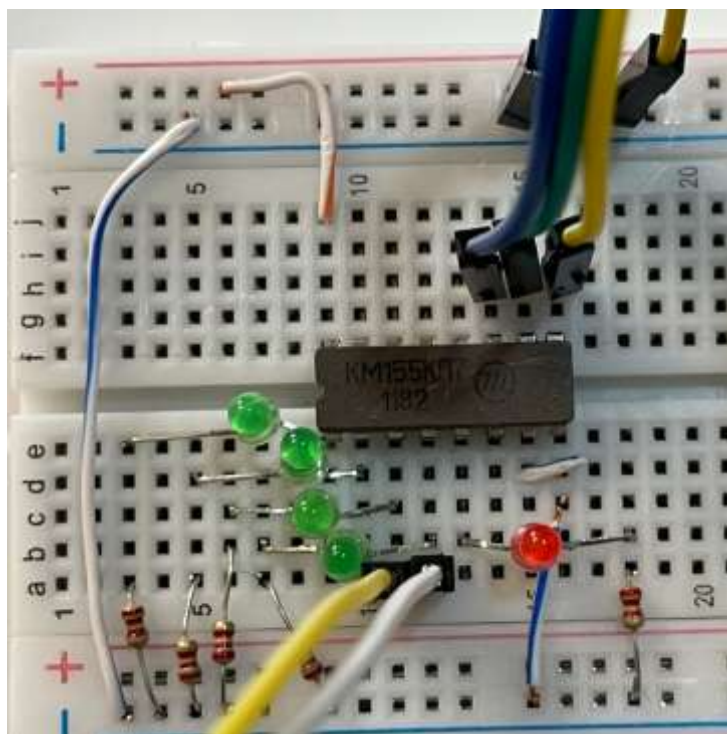
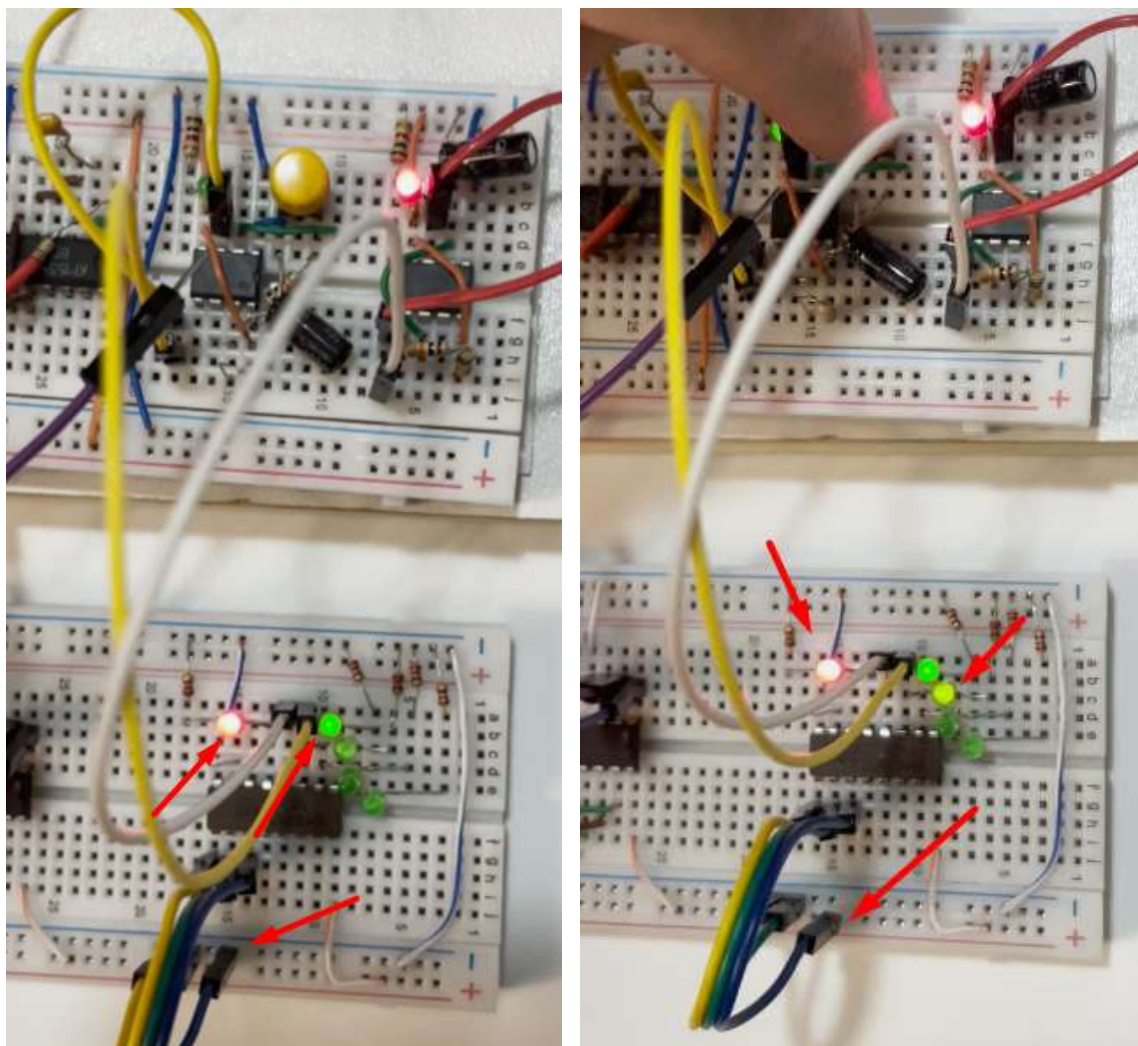


Рисунок 5.9 – Складена досліджувана схема мультиплексора на макетній платі

3. Проводимо дослідження мультиплексора, шляхом перемикання адресних входів А, В, С на живлення або на землю. Додатково зазначимо, що в нашому варіанті буде продемонстровано перемикання адресним входом А між «0» та «1», В і С будуть дорівнювати «0». Тому ми бачитимемо перемикання між виходом D0 і D1. Результати досліджень наведено на рис. 5.10.



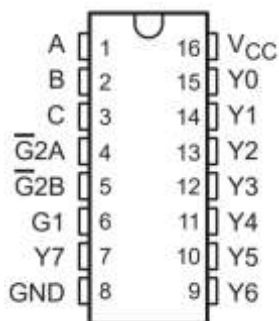
а) б)
Рисунок 5.10 – Результати досліджень мультиплексора

Результати досліджень (рис. 5.10) показують, що кодова комбінація адресного входу відповідає перемиканню відповідного входу на вихід. Стрілочка на проводи з адресними входами показує, що адреса становить 000, і світяться червоний та зелений світлодіоди, на які вказують стрілки на рисунку 5.10, а. На рисунку 5.10, б показано, що адреса змінилась на 001 і другий діод світиться одночасно з вихідним червоним діодом.

4. Вибираємо мікросхему демультиплексора для дослідження.

В цьому випадку ми вибрали мікросхему SN74HC138. Ми вибрали цю мікросхему, оскільки вона доступна для користувачів на цей момент і її легко придбати.

На рис. 5.11 показано умовне графічне позначення SN74HC138 та призначення виводів.



- 1 – адресний вхід A;
- 2 – адресний вхід B;
- 3 – адресний вхід C;
- 4 – інформаційні вхід G2A;
- 5 – інформаційні вхід G2B;
- 6 – інформаційні вхід G1;
- 8 – загальний;
- 7, 9-15 – інформаційні виходи Y7-Y0;
- 16 – напруга живлення

Рисунок 5.11 – Умовне графічне позначення SN74HC138 та призначення виводів

Електричні параметри SN74HC138 наведено в табл. 5.3.1. В табл. 5.4.1 наведено таблицю істинності для SN74HC138.

Таблиця 5.3.1 – Електричні параметри SN74HC138

Напруга живлення	5 В
Вихідна напруга низького рівня	не більше 0,5 В
Вихідна напруга високого рівня	не менше 3,15 В
Вхідний струм низького рівня	не більше -2,5 мА
Вхідний струм високого рівня	не більше 0,08 мА
Струм споживання	не більше 62 мА

Таблиця 5.4.1 – Таблиця істинності SN74HC138

G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	L	H	H	H	H	L	H	H	H	H	H
H	L	L	H	L	L	H	H	H	H	L	H	H	H
H	L	L	H	L	H	H	H	H	H	H	L	H	H
H	L	L	H	H	L	H	H	H	H	H	H	L	H
H	L	L	H	H	H	H	H	H	H	H	H	H	L

2. Складаємо на макетній платі схему для дослідження цього демультимплектора (рис. 5.12). Необхідно зазначити, що для дослідження демультимплектора будуть використані генератори сигналів на мікросхемі NE555.

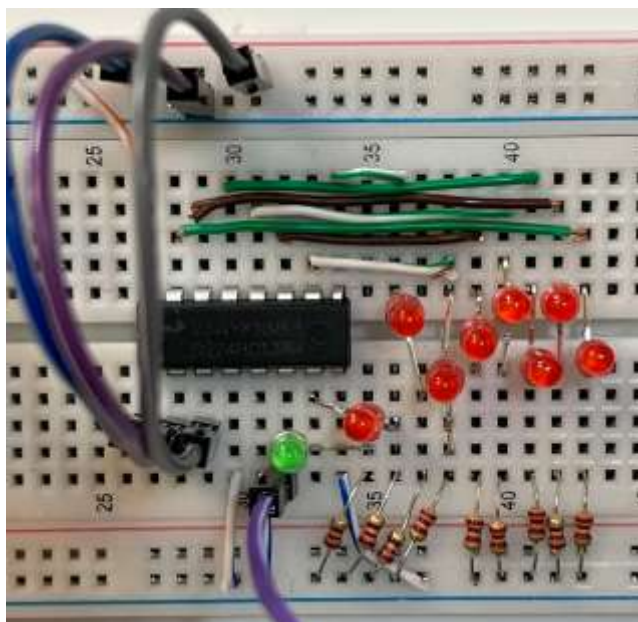


Рисунок 5.12 – Складена досліджувана схема демультимплектора на макетній платі

3. Проводимо дослідження демультимплектора шляхом перемикання адресних входів А, В, С на живлення або на землю. Результати досліджень наведено на рис. 5.13.

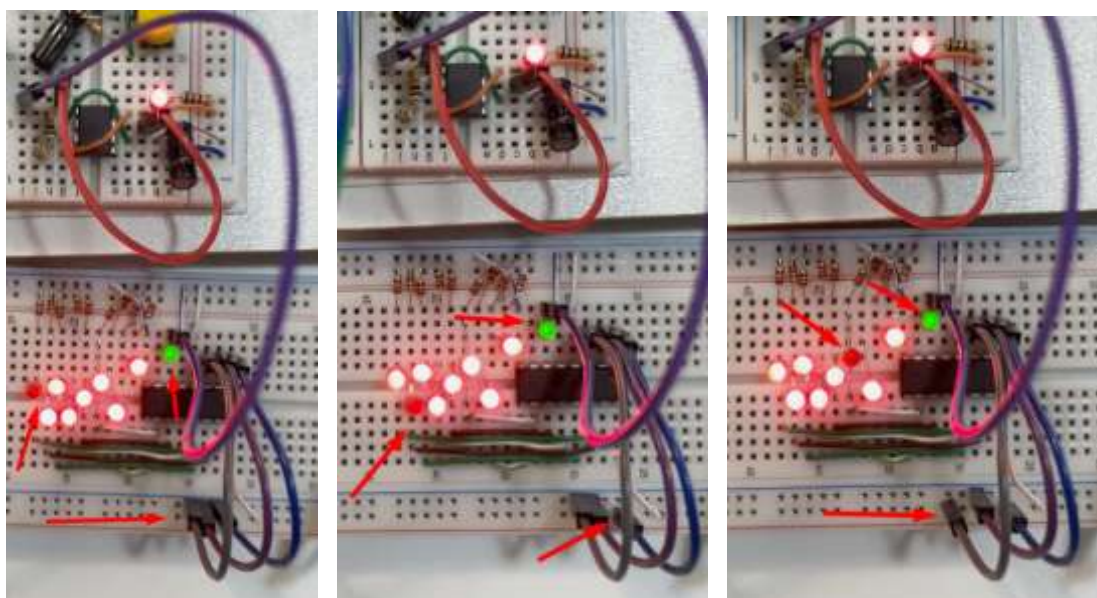


Рисунок 5.13 – Результати досліджень демультимплектора

Результати досліджень (рис. 5.13) показують, що кодова комбінація адресного входу відповідає перемиканню відповідного виходу на вхід. На рис. 5.13 стрілками показано як змінились контакти на керуючих входах і де відповідний вихідний червоний діод не горить.

Порядок виконання роботи (on-line частина)

1. Складаємо досліджувану схему, в нашому випадку це мультиплексор, який комутує відповідний вхід з виходом. Досліджувану схему наведено на рис. 5.14.

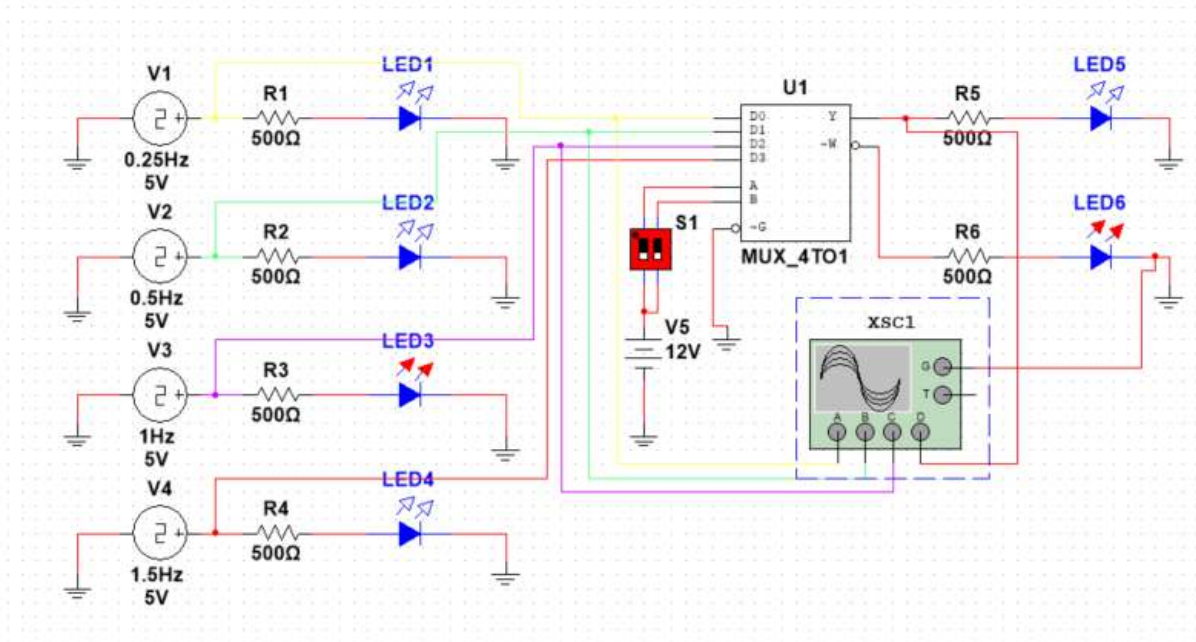


Рисунок 5.14 – Досліджувана схема мультиплексора в multisim

2. Проводимо дослідження схеми мультиплексора в програмному пакеті. Результати наведено на рис. 5.15 та рис. 5.16.

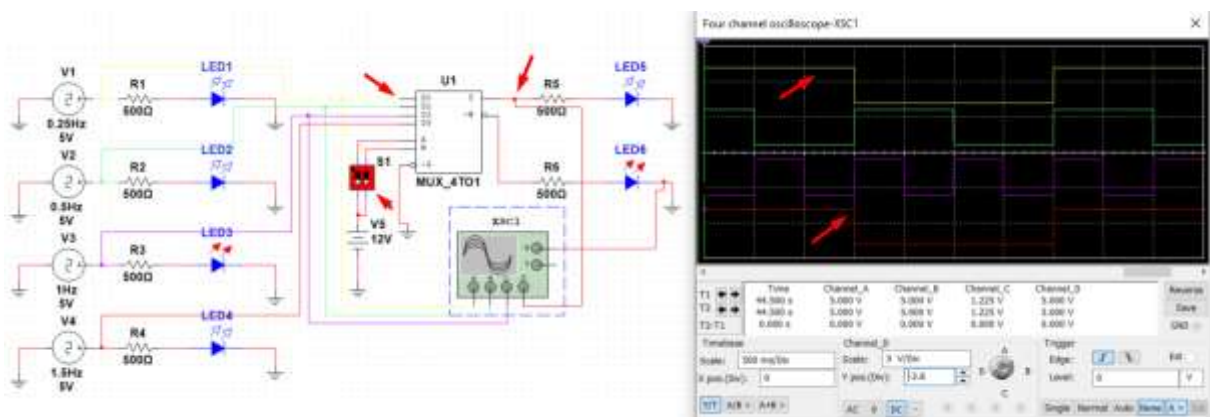


Рисунок 5.15 – Результати досліджень мультиплексора в multisim за кодової комбінації 00

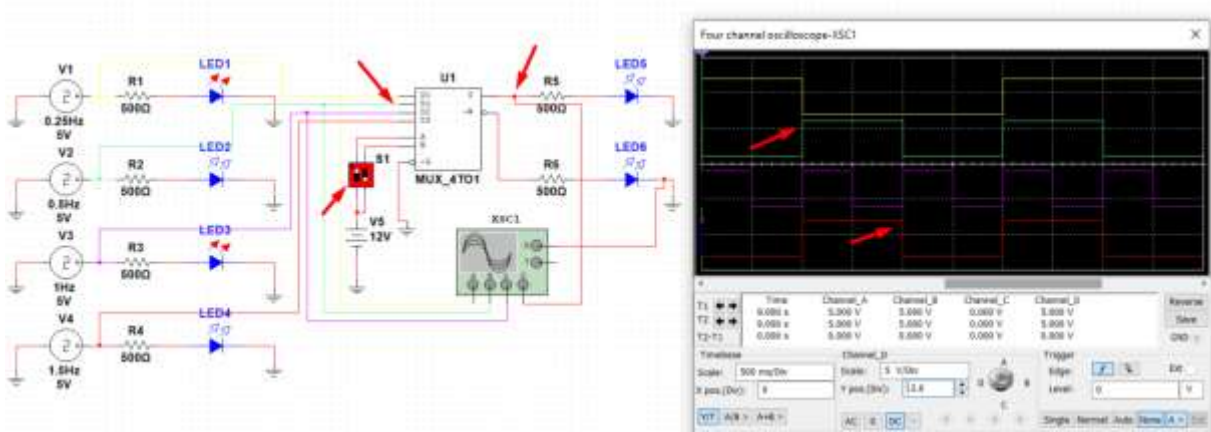


Рисунок 5.16 – Результати досліджень мультиплектора в multisim за кодової комбінації 01

3. Проводячи аналіз даних рис. 5.15 та рис. 5.16, можна підтвердити правильне функціонування мультиплектора для комутації сигналів з багатьох входів на один вихід.

4. Складаємо другу досліджувану схему, в нашому випадку це демультиплектор, який комутує один вхід з багатьма виходами. Досліджувану схему наведено на рис. 5.17.

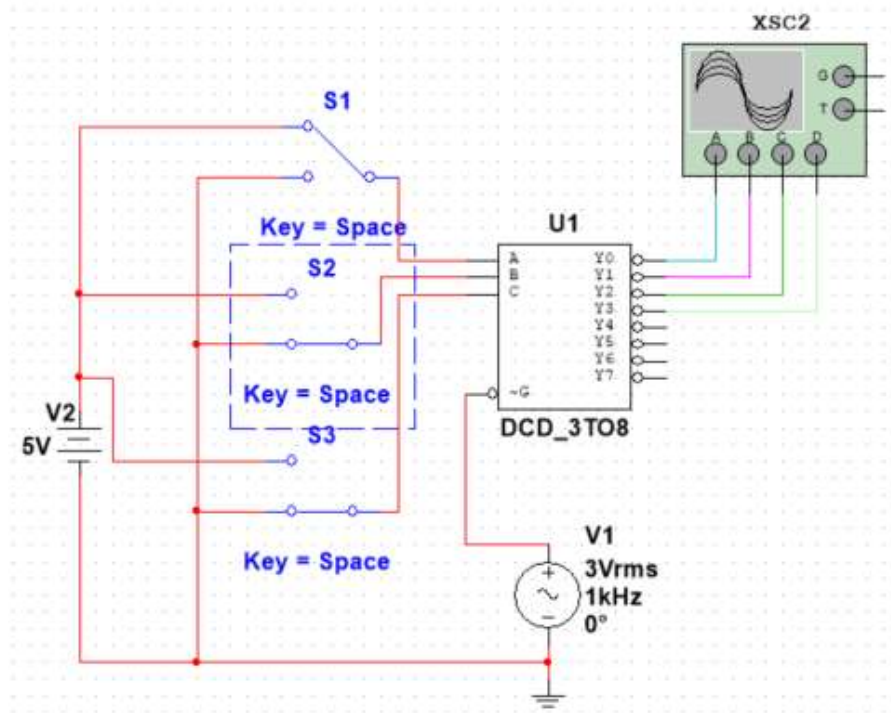


Рисунок 5.17 – Досліджувана схема демультиплектора в multisim

2. Проводимо дослідження схеми демультиплектора в програмному пакеті. Результати наведено на рис. 5.18.

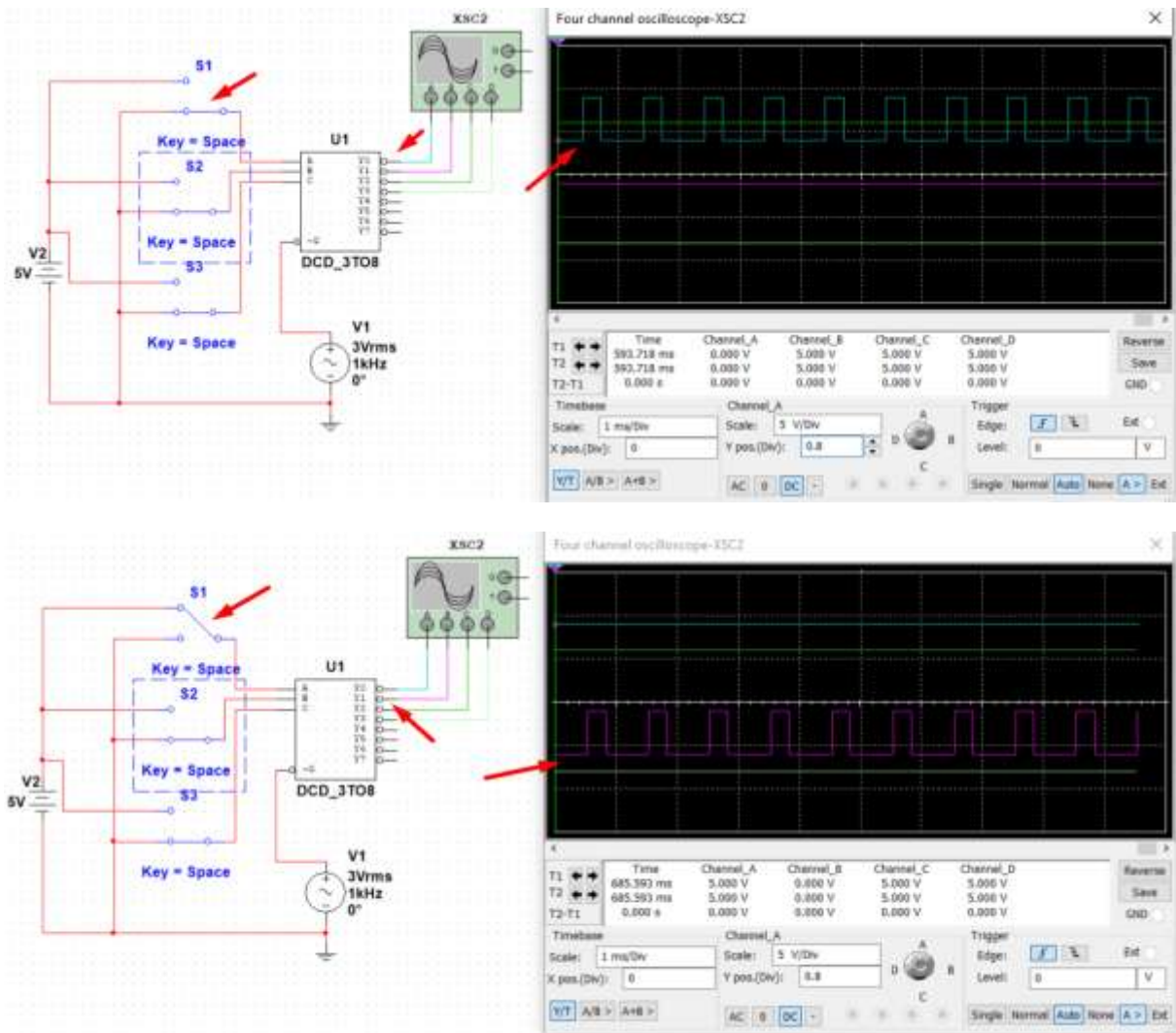


Рисунок 5.18 – Результати досліджень демультимплектора в multisim

3. Проводячи аналіз даних рис. 5.17, можна підтвердити правильне функціонування демультимплектора для комутації сигналів з багатьох виходів з одним входом.

Варіанти завдань

1. Побудуйте схему 2:1 мультимплектора з використанням базових логічних елементів і протестуйте на різних входних комбінаціях.
2. Розробіть схеми 4:1 та 8:1 мультимплекторів та порівняйте їх функціонування.
3. Створіть схему демультимплектора 1:2. Визначте, як виходи реагують на зміни вибіркового сигналу.
4. Використайте два 2:1 мультимплектори для створення 4:1 мультимплектора. Поясніть принцип роботи.

5. Мультиплексор з логічними функціями. Реалізуйте логічну функцію $(A + B) \cdot C$ за допомогою 4:1 мультиплектора. Створіть таблицю істинності.

6. Розробіть і протестуйте 8:1 мультиплексор. Опишіть його застосування в цифрових схемах.

7. Створіть схему демультимплектора 1:4. Проаналізуйте його характеристики та можливості використання.

8. Розробіть сигнальну схему, яка використовує мультиплексор для вибору між двома аналоговими сигналами.

9. Створіть схему, яка демонструє використання мультиплектора в системі зворотного зв'язку.

10. Розробіть схему, яка демонструє як мультиплексор може бути використаний для кодування даних.

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного мультиплектора або демультимплектора.
4. Результати досліджень (рисунок, фото, скрін екрана і т. д.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке мультиплексор і яка його основна функція в цифрових схемах?
2. Які основні типи мультиплексорів існують?
3. Як працює мультиплексор і які його основні характеристики?
4. Яка роль вибіркових входів у мультиплексорі?
5. Як реалізується функція демультимплексування?
6. Які основні параметри демультимплектора?
7. Яка різниця між мультиплексором і демультимплексором?
8. Які переваги використання мультиплексорів у цифрових системах?
9. Як впливають затримки на роботу мультиплексорів?
10. Яка роль мультиплексорів у системах передачі даних?
11. Як можна реалізувати мультиплексор за допомогою базових логічних елементів?
12. Які проблеми можуть виникати під час використання демультимплексорів у схемах?

13. Як змінюється вихідна інформація в демультиплексорі у випадку зміни вибіркового входу?
14. Які приклади застосування мультиплексорів у побутових пристроях?
15. Як демультиплексори використовуються в системах управління?
16. Як впливають електричні характеристики на роботу мультиплексорів і демультиплексорів?
17. Як мультиплексори використовуються в системах управління?

Лабораторна робота 6

Регістри

Теоретична частина

Регістр – це спеціальна комірка пам'яті всередині центрального процесора (CPU) або іншого цифрового пристрою, призначена для швидкого зберігання та обробки даних. Ці структури призначені для оперативного зберігання та маніпулювання даними. Регістри використовуються для тимчасового розміщення операндів, що беруть участь в обчисленнях, адрес пам'яті для доступу до даних, результатів виконаних операцій, а також іншої критично важливої для функціонування процесора інформації. Завдяки їхній значно вищій швидкодії порівняно з основною оперативною пам'яттю (RAM), процесор має можливість швидко отримувати необхідні дані та ефективно виконувати інструкції. Класифікація регістрів містить різні типи, такі як регістри загального призначення, регістри команд, регістри адрес, регістри стану, кожен з яких виконує чітко визначену функцію. Розрядність регістра визначається кількістю тригерів, що входять до його складу, а отже, і кількістю розрядів двійкового числа, яке може зберігати регістр. Швидкодія регістрів визначається швидкодією тригерів, що входять до складу регістрів. За способом прийому і передачі інформації у двійковій формі регістри поділяють на паралельні, послідовні та універсальні. [1]

Регістр являє собою функціональний блок, призначений для збереження цифрових слів. Зазвичай він забезпечує виконання ряду базових мікрооперацій, серед яких: встановлення нульового значення (очищення), отримання даних від іншого блока (регістра, лічильника тощо), передача збережених даних до іншого блока, інвертування бітів слова, зсув бітів вліво або вправо на задану кількість позицій, а також перетворення між послідовним і паралельним поданням коду.

Конкретні реалізації регістрів можуть підтримувати лише частину з цих операцій. Як правило, основою для створення регістрів слугують різні типи тригерів (RS, JK, D, DV, T), які самі по собі забезпечують функцію запам'ятовування двійкових слів.

Додаткові мікрооперації реалізуються шляхом підключення логічних елементів до входів і виходів тригерів, а також шляхом відповідної організації зв'язків між ними.

Паралельний регістр визначається як такий, що підтримує всі згадані мікрооперації, за винятком операцій зсуву та перетворення послідовного коду в паралельний і навпаки [2].

Парафазний паралельний регістр характеризується подачею вхідної інформації на кожен розряд по двох каналах у протифазі. У випадку використання лише одного каналу (прямого або інверсного) для передачі інформації в кожному розряді, регістр називається однофазним. Приклад схеми паралельного однофазного регістра, що реалізує операції

встановлення в нуль та прийому даних з іншого регістра, наведено на рис. 6.1.

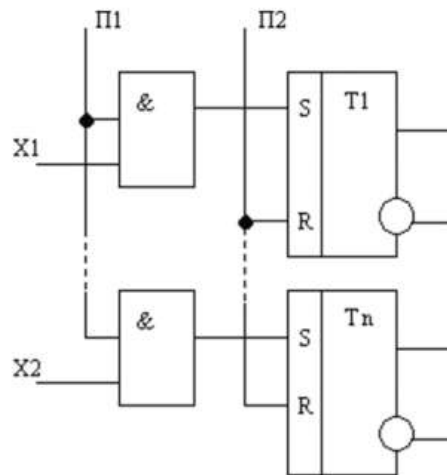


Рисунок 6.1 – Схема паралельного однофазного регістра

На нульові входи всіх тригерів подається сигнал установлення нульового стану $\Pi 0$. Після подачі цього сигналу всі тригери регістра будуть знаходитись в нульовому стані до появи на входних шинах записуючого слова і сигналу прийому $\Pi 1$. В тих розрядах, де $X_i = 1$, здійсниться установлення тригерів в одиничний стан. Там, де $X_i = 0$, стани тригерів не зміняться. Видача інформації з регістра може проводитись в прямому, інверсному та парафазному кодах [3].

Схему видачі інформації в прямому та інверсному кодах показано на рис. 6.2, де $B1$ – сигнал видачі прямого коду; $B2$ – сигнал видачі інверсного коду.

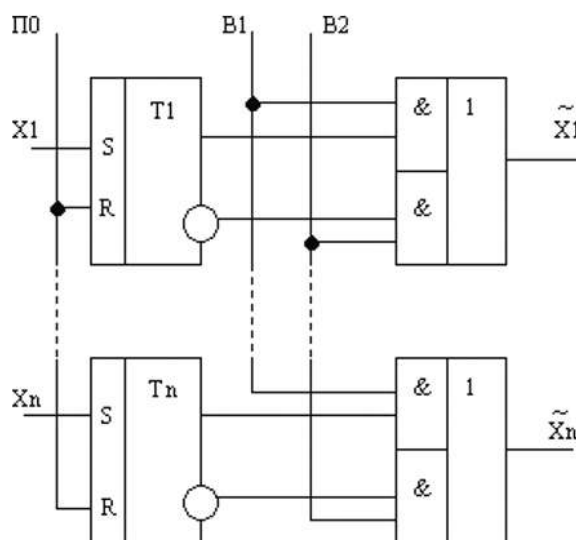


Рисунок 6.2 – Схема видачі інформації в прямому та інверсному кодах

Одночасна поява сигналів $B1$ і $B2$ заборонена.

Схему регістра з видачею парафазного коду наведено на рис. 6.3.

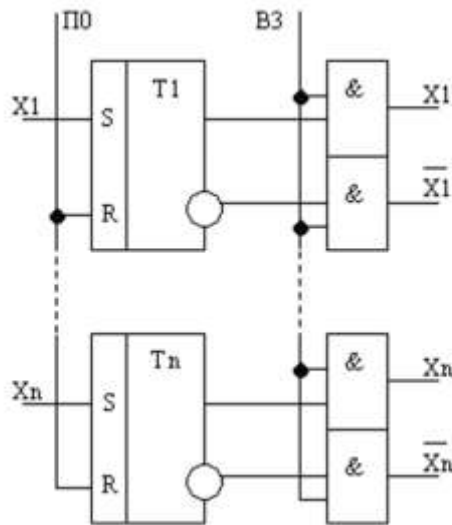


Рисунок 6.3 – Схема видачі інформації в парафазному коді

Схема прийому інформації в регістр без попереднього установлення його в «0» подано на рис. 6.4.

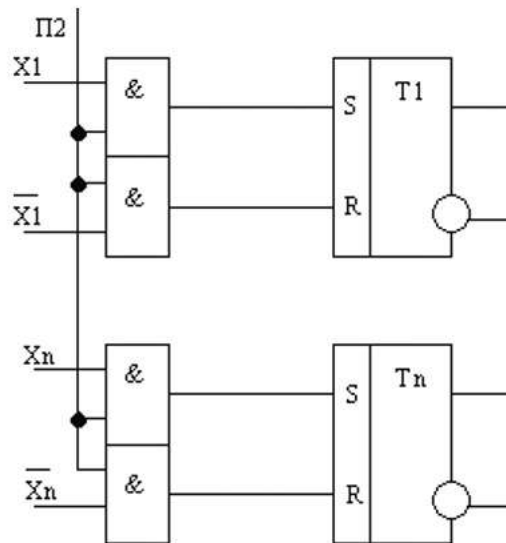


Рисунок 6.4 – Схема прийому інформації в регістр без попереднього установлення його в «0»

Використання парафазного коду забезпечує установлення тригерів в необхідний стан незалежно від тієї інформації, яка була записана на них раніше.

Регістром зсуву називається такий регістр, який виконує мікрооперацію зсуву, а також мікрооперації прийому та видачі слів.

Якщо дві останні операції реалізуються тільки в крайніх розрядах регістра (у першому або n -ому), то регістр зсуву називають також послідовним регістром.

У випадку зсуву слова в регістрі розряди, що вийшли за межі розрядної сітки регістра, втрачаються, а розряди, що звільнилися, заповнюються нулями [12].

Регістри, які можуть здійснювати зсув слова як вліво, так і вправо, називають реверсивними.

Схему реверсивного регістра зсуву з парафазною передачею слів зображено на рис. 6.5, де $C1$ – сигнал зсуву вмісту регістра вправо на один розряд; $C2$ – сигнал зсуву вмісту регістра на один розряд вліво.

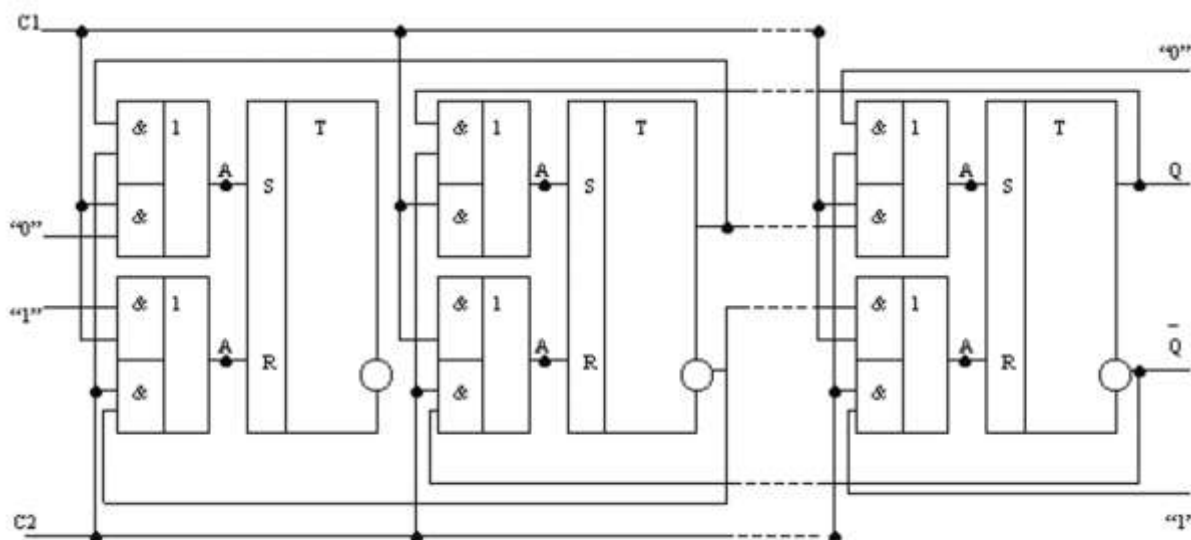


Рисунок 6.5 – Схема реверсивного регістра зсуву з парафазною передачею слів

Схема реверсивного регістра зсуву з парафазною передачею слів не потребує скидання регістра в нуль.

Регістр зсуву реалізує перетворення послідовного коду в паралельний і навпаки. Під час перетворення послідовного коду в паралельний, запис слова в регістр здійснюється синхронно зі зсувом його вмісту вліво, якщо послідовний код надходить зі старших розрядів, або вправо, якщо код надходить з молодших розрядів.

Після заповнення всіх розрядів йде паралельна видача слова за допомогою елементів, підключених до виходів тригерів регістра. Під час перетворення паралельного коду в послідовний інформація заноситься в регістр паралельним кодом, а потім йде серія з n зсовувальних сигналів.

Послідовний код зчитується з тригера молодшого розряду під час зсуву вправо або з тригера старшого розряду у разі зсуву вліво [16].

До складу 155 серії входить мікросхема 155ИР1 – чотирирозрядний регістр зсуву, умовне графічне зображення якого наведено на рис. 6.6.

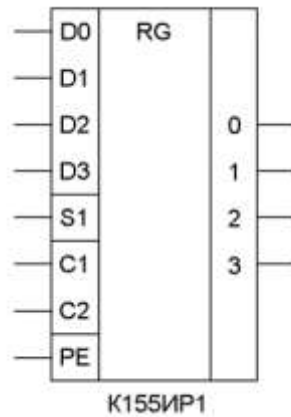


Рисунок 6.6 – Умовне графічне зображення мікросхеми 155IP1

Регістр зсуву має послідовний вхід даних S1, чотири паралельні входи D0-D3, а також чотири виходи Q0-Q3 від кожних з тригерів. Регістр має два тактових входи C1 та C2. Від будь-якого з п'яти входів даних код надходить на виходи синхронно з негативним перепадом, поданим на один з тактових входів. Вхід дозволу паралельного завантаження PE слугує для вибору режиму роботи регістра. Якщо на вхід PE подається напруга високого рівня, то дозволяється робота тактовому входу C2. У момент приходу на цей вхід негативного перепаду тактового імпульсу в регістр завантажуються дані від паралельних входів D0-D3. Якщо на вхід PE подано напругу низького рівня, то дозволяється робота по тактовому входу C1. Негативні фронти тактових імпульсів зсовують дані, які надходять від послідовного входу S1, на вихід Q0, потім на Q1, Q2, Q3, тобто відбувається зсув вправо. Зсув даних вліво здійснюється, якщо з'єднати вихід Q3 та вхід D2, вихід Q2 та вхід D1 і вихід Q1 та вхід D0.

Щоб перевести регістр в паралельний режим, необхідно подати на вхід PE напругу високого рівня. Напругу на вході PE можна змінювати тільки у випадку, якщо на обох тактових входах C1 та C2 рівні низькі. Проте, якщо на вході C1 напруга низького рівня, зміна сигналу на вході PE від низького рівня до високого не змінює стану виходів.

Порядок виконання роботи (off-line частина)

1. Вибираємо мікросхему регістра зсуву.

Виходячи з доступності та вартості на ринку вибираємо мікросхему 74НС595. Цей регістр зсуву має вісім вихідних ліній. Це дозволяє під'єднувати до нього до восьми виконавчих пристроїв, таких як світлодіоди чи звукові сигналізатори. Вісім вихідних контактів позначені як QA-QH, де QA (пін 15) є першим. За ним послідовно розташовуються виходи QB-QH (піни 1-7). Важливо, що ці вихідні піни напругу не контактують з мікроконтролером Arduino. Замість цього вони з'єднуються

з керованими елементами, у нашому випадку – світлодіодами. В табл. 6.1 наведено функціональне призначення виводів мікросхеми 74HC595.

Для забезпечення живлення мікросхеми 74HC595 на пін VCC подається напруга +5 В, а пін GND з'єднується із землею (GND).

Вхідний пін послідовних даних (Serial Data Input) підключається через підтягувальний резистор до землі. Таке підключення забезпечує за замовчуванням низький рівень на лінії даних, коли на неї не впливають. Під час подачі сигналу (наприклад, натисканні кнопки) рівень стає високим. Таким чином, на вхід послідовних даних подається логічна одиниця (HIGH). Передача цього значення до внутрішнього регістра зберігання відбувається за позитивного фронту тактового імпульсу на пині CLOCK. Якщо на вхід послідовних даних подано логічний нуль (кнопка не натиснута), то до регістра зберігання записується нуль. За логічної одиниці на вході даних (кнопка натиснута) – записується одиниця.

Тактовий вхід (CLOCK, пін 11) використовується для синхронізації зсуву бітів всередині регістра. Регістр 74HC595 реагує на зростаючий фронт тактового сигналу. Це означає, що для зсуву даних до регістра зберігання тактовий сигнал має змінюватися з низького рівня на високий. Передача бітів до регістра зберігання відбувається саме в момент цього переходу. Отже, для завантаження нових даних необхідно, щоб тактовий сигнал мав зростаючий фронт.

Пін фіксації (LATCN, пін 12) відіграє ключову роль у виведенні збережених даних. Коли на цей пін подається високий рівень, дані з внутрішнього регістра зберігання копіюються на вихідні пини QA-QH. Це означає, що спочатку передані дані потрапляють до регістра зберігання, але не відображаються на виходах. Вони стають видимими (наприклад, вмикаються світлодіоди) лише після встановлення високого рівня на пині фіксації. Таким чином, пін фіксації є завершальним етапом у процесі відображення результату АЦП-перетворення на вихідних пристроях.

Пін очищення (CLEAR, активний низький рівень) використовується для скидання всіх бітів у регістрі зберігання до нуля. Щоб запобігти випадковому очищенню, цей пін утримується у високому стані шляхом підключення до VCC. Оскільки цей вивід є активним за низького рівня, для забезпечення його високого стану використовується підтягувальний резистор, підключений до VCC. Підтягувальний резистор за замовчуванням забезпечує високий рівень, і лише за подачі низького рівня (наприклад, у разі натискання кнопки, підключеної до землі) пін CLEAR активується, скидаючи всі одиниці в регістрі зберігання на нулі.

Пін увімкнення виходу (Output Enable, активний низький рівень) контролює стан усіх вихідних пинів. Зазвичай необхідно, щоб виходи були ввімкнені для відображення змін. Якщо цей пін перебуває у високому стані, жоден з вихідних пинів не може бути активним. Оскільки це активний низький рівень, для постійного увімкнення виходів цей пін

завичай підключається до землі або через резистор до низького рівня. У цій схемі він підключений через підтягувальний резистор до землі.

Пін послідовного виходу (QH') використовується для каскадного з'єднання кількох регістрів зсуву. Це дозволяє збільшити кількість керованих виходів (наприклад, з 8 до 16). Принцип каскадування полягає у підключенні піна послідовного виходу першого регістра зсуву до піна послідовного входу другого регістра зсуву. Таким чином, дані, що «виштовхуються» з першого регістра, потрапляють безпосередньо до другого. Оскільки в цій схемі не передбачено використання додаткових регістрів зсуву, вказаний пін залишається непідключеним.

Таблиця 6.1 – Функціональне призначення виводів 74НС595

Вивід	Опис	Функція
QA-QH	Вихідні контакти	Виходи регістра зсуву
VCC (Pin 16)	Потужність	Позитивна напруга для регістра зсуву
GND (Pin 8)	Потужність	Заземлення для регістра зсуву
QH' (Pin 9)	Серійний вихід	Послідовний вихід використовується для перенесення даних на інший регістр зсуву 74НС595.
CLEAR (Pin 10)	Майстер Reclear, активний низький	Це встановлює всі біти в регістрі зсуву в 0 або вимикає їх, якщо вони переведені в низький рівень.
Shift Clock (Pin 11)	Вивід годинника регістра зсуву	Якщо підняти рівень ВИСОКИЙ, це зсуне всі значення в регістрі зсуву на одне значення вперед.
Latch Clock (Pin 12)	Штифт блокування регістра зберігання	Коли виведено високий рівень, він виводить нові значення регістра зсуву.
Output Enable (Pin 13)	Вихід увімкнено, активний низький	Це вмикає вихід, коли він заземлений, і вимикає його, коли він має високий рівень.
Serial Data Input (Pin 14)	Вхідні дані для нових послідовних даних	Це вхідний контакт для нових послідовних даних.

2. Для дослідження складаємо електричну схему, показану на рис. 6.7, на макетній платі. Результат наведено на рис. 6.8.

Ця схема зсувного регістра має досить прості з'єднання.

Спочатку під'єднується живлення. Пін VCC зсувного регістра з'єднується з джерелом напруги +5 В, а пін GND – з загальною шиною (землею) живлення. Це забезпечує достатнє живлення мікросхеми.

Далі тактовий пін (пін 11) підключається до підтягувального резистора, який з'єднує його з землею.

Аналогічно пін фіксації (пін 12) також під'єднується до землі через підтягувальний резистор. Вхід послідовних даних (Serial Data Input) також з'єднується з землею за допомогою підтягувального резистора.

Оскільки пін увімкнення виходу (Output Enable) є активним за низького рівня, для його активації він підключається до підтягувального резистора, з'єднаного з землею, забезпечуючи таким чином низький рівень.

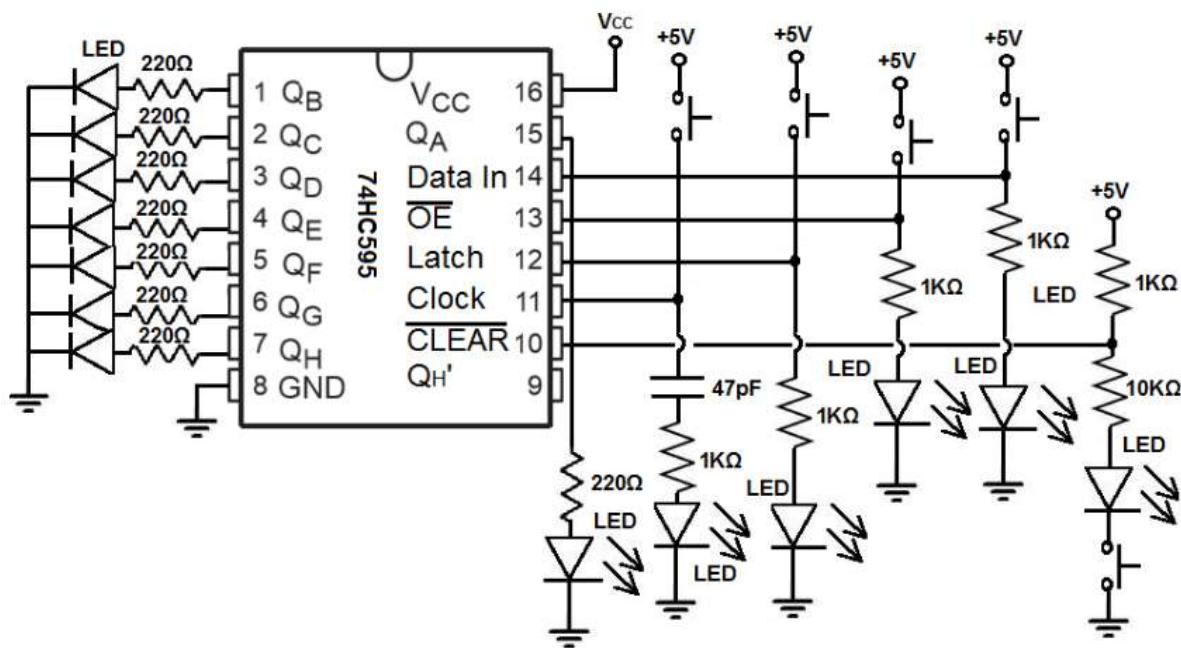


Рисунок 6.7 – Схема для дослідження регістра зсуву 74HC595

Пін очищення (CLEAR) є активним за низького рівня, тому для його утримання у високому стані він підключається до джерела живлення +5 В через подільник напруги, що складається з резисторів 1 кОм (верхній) та 10 кОм (нижній). Більша частина напруги спадатиме на резисторі 10 кОм, гарантуючи високий рівень на пині CLEAR. Через високий опір резистора 10 кОм, світлодіод може ледь помітно світитися, що є прийнятним для демонстраційних цілей.

Наостанок, світлодіоди підключаються до вихідних пинів QA – QH. До кожного світлодіода послідовно під'єднується струмообмежувальний резистор опором 220 Ом.

Принцип роботи схеми полягає у послідовній передачі бітів до регістра зберігання. Для цього необхідно, щоб на тактовому пині був високий рівень під час подачі імпульсу на тактову кнопку. Якщо вхід послідовних даних у цей момент має низький рівень, до регістра зберігання записується 0. Якщо ж вхід послідовних даних має високий рівень – до регістра зберігання передається 1. Наприклад, якщо натиснути тактову кнопку 8 разів за високого рівня на тактовому пині, до регістра зберігання буде записано вісім одиниць. Комбінація чотирьох натискань на кнопку вхідних даних за високого рівня на тактовому пині та чотирьох натискань на тактову кнопку за відпущеної кнопки вхідних даних приведе до запису послідовності 11110000 до регістра зберігання.

Після передачі необхідних даних до регістра зберігання, наступним кроком є їх фіксація на виходах. Передані дані не відображаються на світлодіодах до моменту їх фіксації. Для цього необхідно подати імпульс на кнопку фіксації, що приведе до передачі даних з регістра зберігання на вихідні піни. Після цього світлодіоди засвітяться відповідно до записаних у регістрі зберігання бітів. Наприклад, за значення 11111111 у регістрі зберігання увімкнуться всі світлодіоди, за 10101010 – кожен другий, а за 11110000 – перші чотири світлодіоди будуть увімкнені, а наступні чотири – вимкнені.

Для вимкнення всіх світлодіодів можна скористатися кнопкою увімкнення виходу або лінією очищення. Натискання кнопки увімкнення виходу приводить до вимкнення всіх вихідних пінів. Натискання кнопки, підключеної до лінії очищення, скидає всі біти в регістрі зберігання на нулі. Після цього необхідно також натиснути кнопку фіксації, щоб побачити, як усі світлодіоди вимкнуться. Важливо розрізняти функції пінів очищення та увімкнення виходу. Пін очищення скидає значення в регістрі зберігання на нулі, після чого можна записувати нові значення. Пін увімкнення виходу ж повністю блокує будь-яку активність на вихідних пінах.

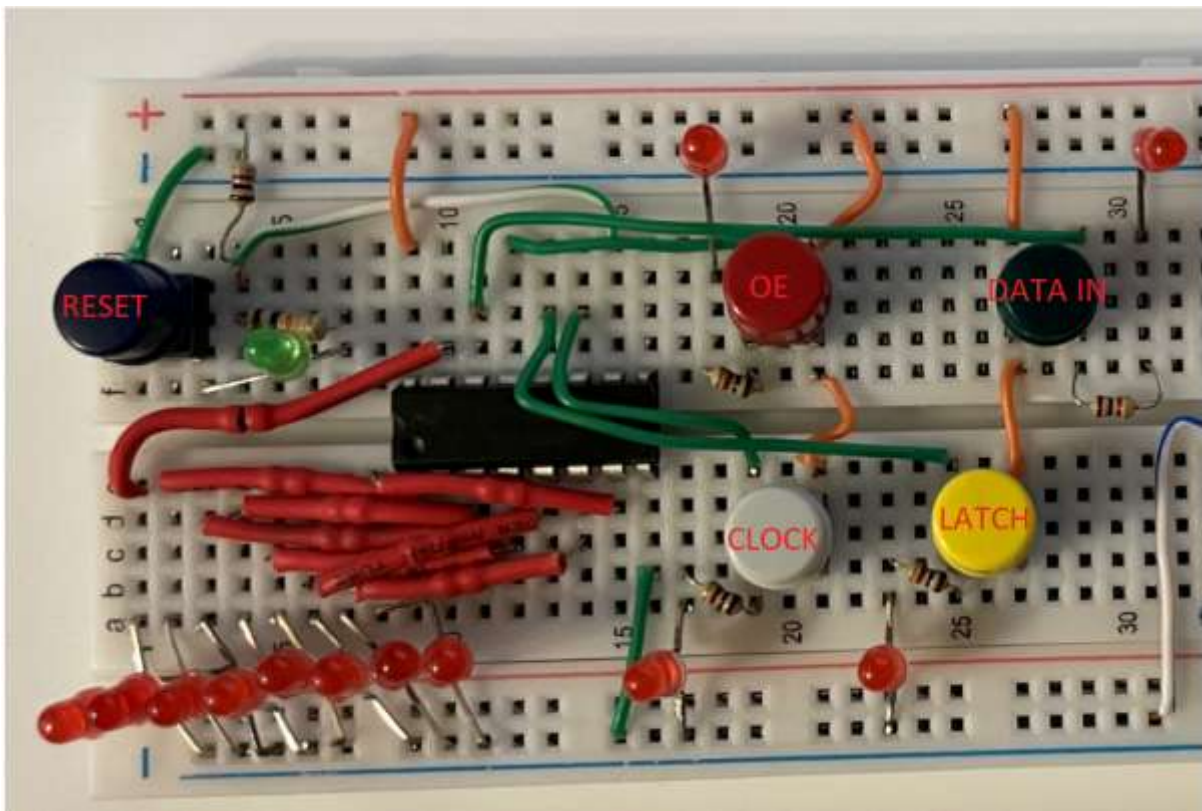


Рисунок 6.8 – Складена макетна плата для дослідження регістра зсуву 74HC595

3. Проводимо дослідження регістра зсуву.

Натиснули і зажали зелену кнопку, потім двічі білу та один раз жовту. Зелену відпустили, і натиснули один раз білу і потім один раз жовту. Таким чином зсуваємо два діоди, що світяться, на одну позицію, як показано на рис. 6.9.

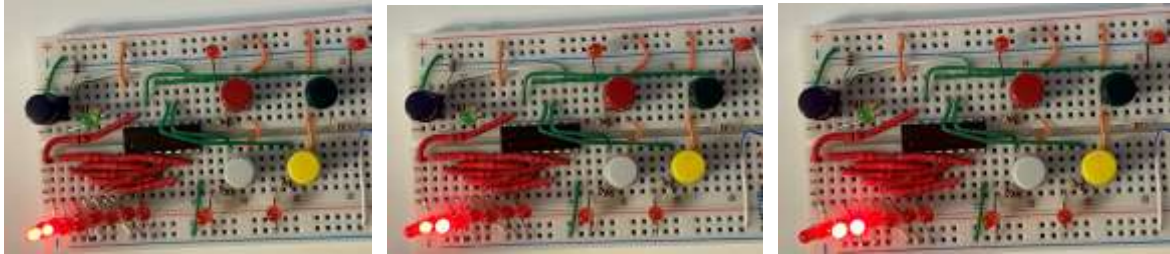


Рисунок 6.9 – Результати досліджень, коли одночасно горять два діоди

Натиснули і зажали зелену кнопку, потім один раз білу та один раз жовту. Зелену відпустили, і нажали один раз білу і потім один раз жовту. Таким чином зсуваємо один діод, який світиться, на одну позицію, як показано на рис. 6.10.

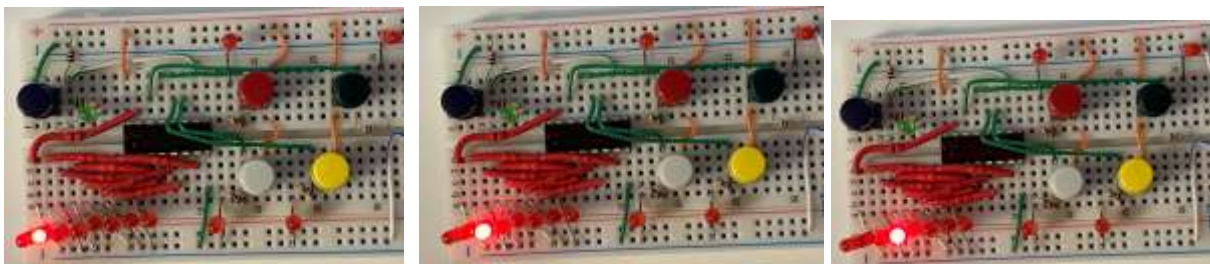


Рисунок 6.10 – Результати досліджень, коли одночасно горить лише один діод

4. Висновки

Аналізуючи результати експериментальних досліджень рис. 6.9 та рис. 6.10, можна чітко відслідковувати роботу регістра зсуву. Ми можемо задавати скільки одночасно діодів світлитиметься, з яким кроком їх переміщуватимемо.

Порядок виконання роботи (on-line частина)

1. Для виконання лабораторної роботи в онлайн форматі можна використовувати різні моделювальні пакети. Обмежень щодо вибору немає, але перевага надається безкоштовним програмним пакетам. Ми будемо виконувати цю лабораторну роботу в програмі Proteus. Аргументацією є те, що в цьому програмному пакеті є модель мікросхеми регістра зсуву 74НС595.

2. Складаємо схему для дослідження.

Схему в моделювальному пакеті наведено на рис. 6.11.

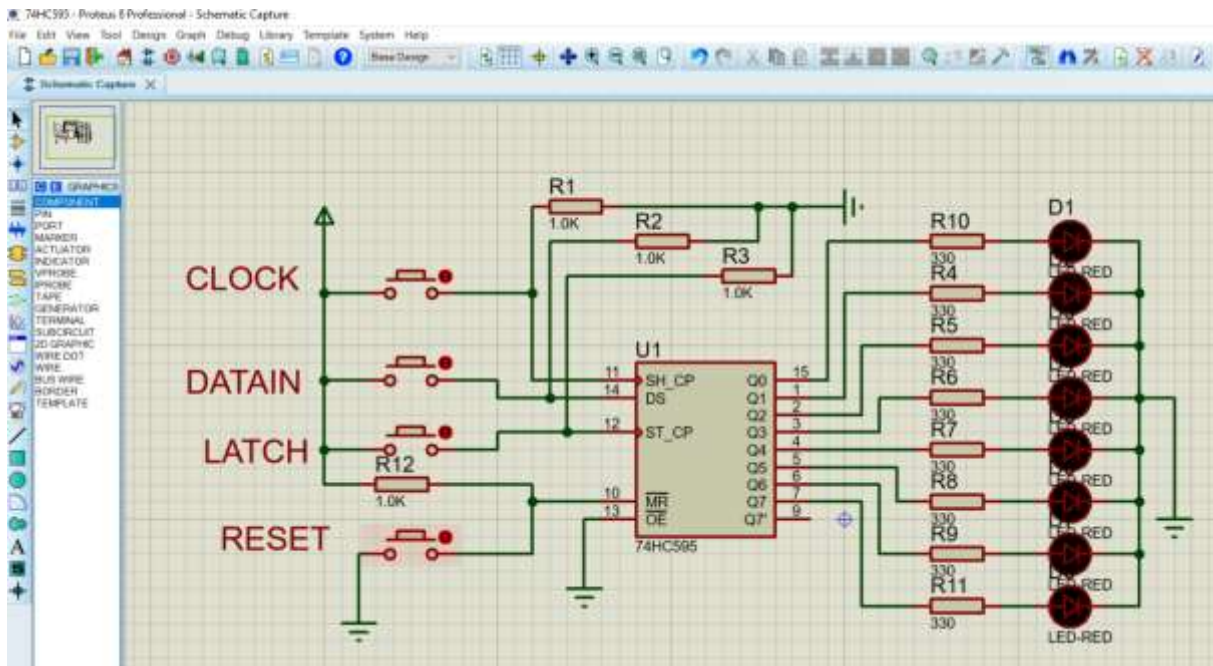


Рисунок 6.11 – Досліджувана схема регістра зсуву в АЦП в Proteus

Керуючи кнопками ми можемо задавати параметри роботи регістра зсуву.

3. Проводимо дослідження роботи регістра зсуву в АЦП в Proteus.

Спочатку налаштуємо, щоб на виході регустра зсуву в нас світився лише один світлодіод, зробимо біжучий рядок з одного світлодіода. Для цього спочатку натискаємо кнопку DATAIN (фіксуємо її), потім натискаємо один раз CLOCK і один раз LATCH. Загоряється верхній світлодіод як показано на рис. 6.12.

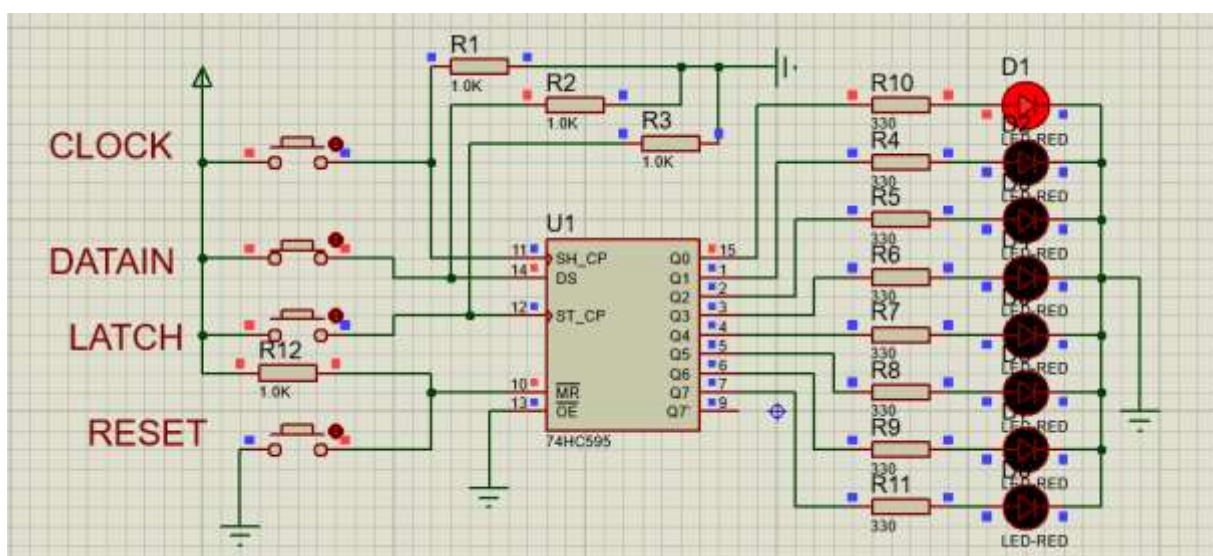


Рисунок 6.12 – Запалювання першого світлодіода

Далі розжимаємо кнопку DATAIN і натискаємо один раз CLOCK та LATCH; діоди послідовно загорятимуться один за одним як показано на рис. 6.13.

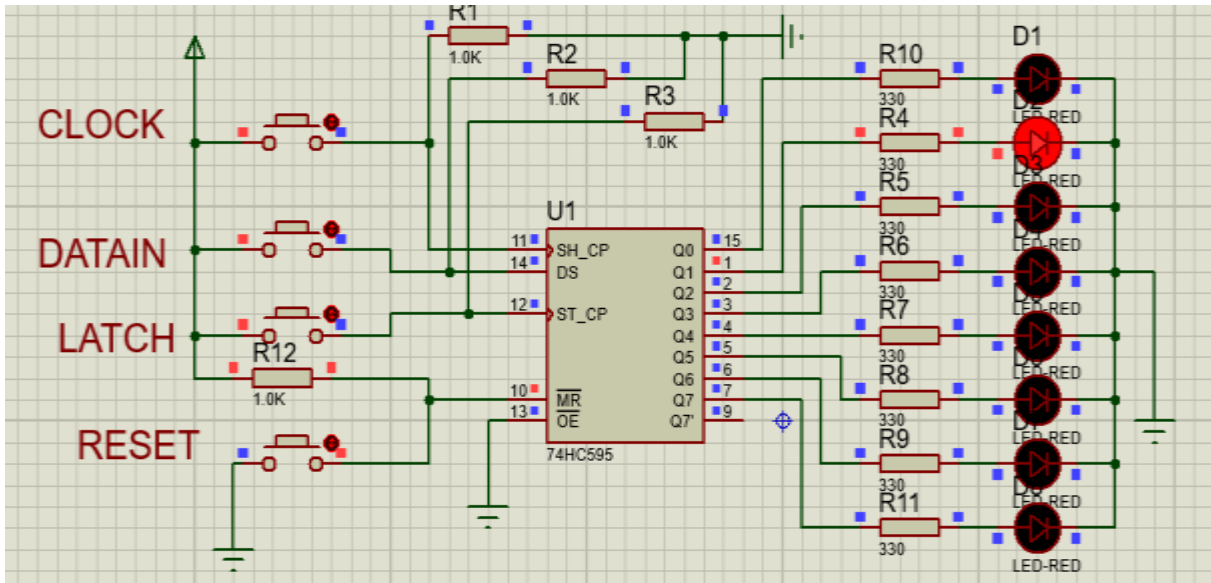


Рисунок 6.13 – Загорання другого світлодіода

Якщо на CLOCK натиснемо двічі і LATCH один раз, діоди будуть зарорятися через один як показано на рис. 6.14.

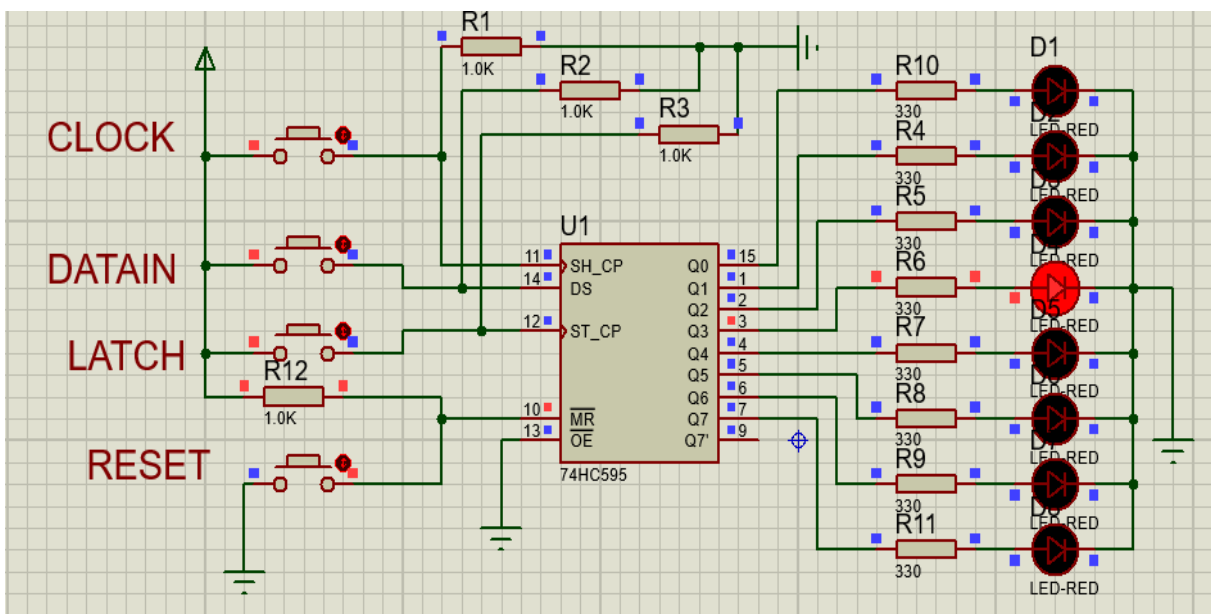


Рисунок 6.14 – Загорання світлодіода через один

Якщо захочемо щоб по три діоди горіло, то фіксуємо DATAIN, тричі натискаємо CLOCK і один раз LATCH. Результат показано на рис. 6.15.

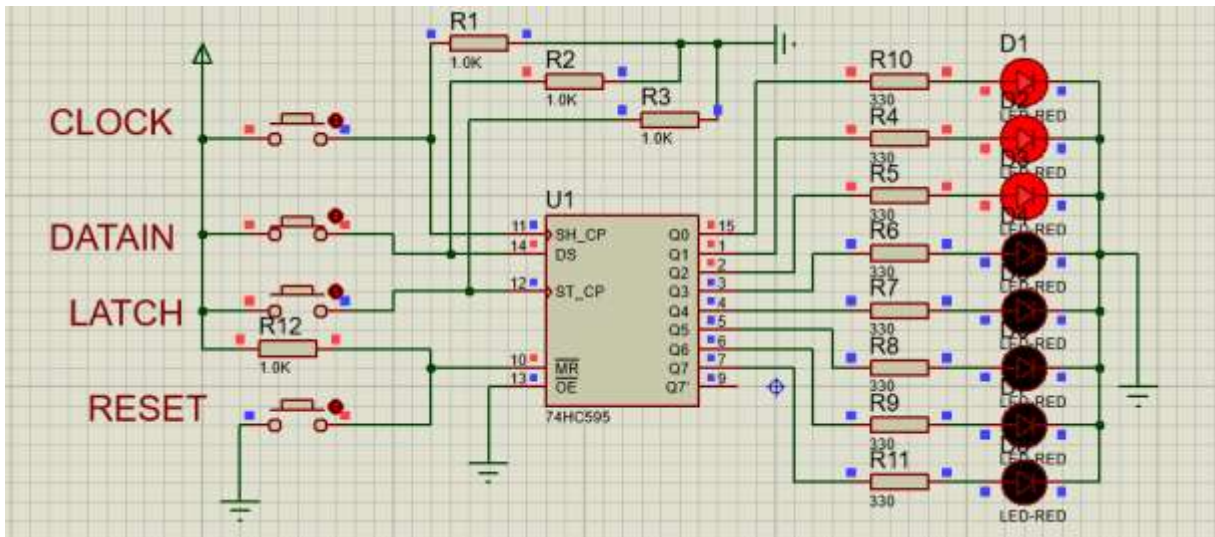


Рисунок 6.15 – Загоряння одночасно трьох світлодіодів

Наступне натискання один раз CLOCK і один LATCH зсуне три діоди, які світяться, на одну позицію як показано на рис. 6.16.

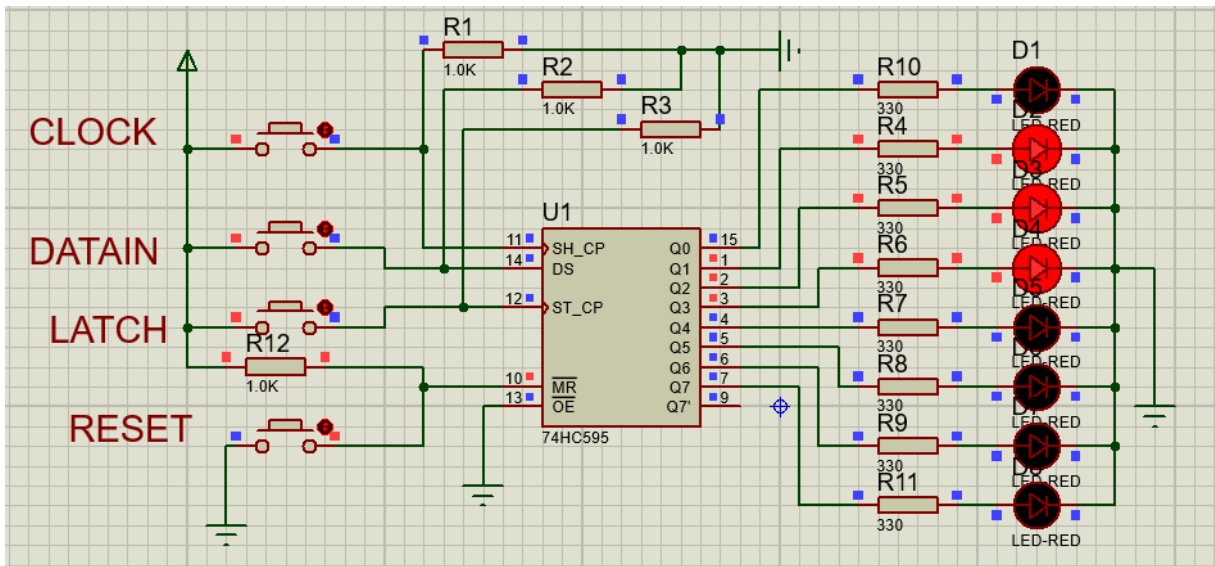


Рисунок 6.16 – Зміщення на один трьох світлодіодів

Щоб всі діоди скинути, необхідно натиснути RESET та потім LATCH. Тоді всі світлодіоди погасять.

4. Висновки

Дослідження регістра зсуву показало, що цей елемент є ключовим у цифрових системах для виконання операцій з бітами, таких як зсув, вставка та видалення бітів. Регістр зсуву може реалізовувати як логічний, так і арифметичний зсув, що дозволяє ефективно виконувати множення або ділення на двійкові числа.

Варіанти завдань

1. Дослідження послідовного регістра зсуву на D-тригерах: скласти схему 4-бітного послідовного зсувного регістра на D-тригерах. Перевірити його роботу, подаючи різні послідовності вхідних даних та спостерігаючи за зміною стану виходів після кожного тактового імпульсу.

2. Дослідження паралельного регістра: скласти схему 4-бітного паралельного регістра на D-тригерах з паралельним завантаженням.

3. Дослідження універсального зсувного регістра: використовуючи інтегральну мікросхему універсального зсувного регістра (наприклад, 74НС194), дослідити режими зсуву вліво, зсуву вправо, паралельного завантаження та зберігання даних.

4. Використання регістра зсуву для керування світлодіодним біжучим рядком: підключити кілька світлодіодів до виходів зсувного регістра (наприклад, 74НС595), щоб створити ефект біжучого рядка.

5. Реалізація послідовно-паралельного перетворювача коду: використовуючи зсувний регістр, реалізувати схему, яка приймає послідовний бітовий потік даних та перетворює його на паралельний 8-бітний код.

6. Реалізація паралельно-послідовного перетворювача коду: використовуючи паралельний регістр з можливістю зсуву, реалізувати схему, яка приймає паралельний 4-бітний код та видає його послідовно.

7. Дослідження каскадного з'єднання зсувних регістрів: з'єднати послідовно дві мікросхеми зсувних регістрів (наприклад, 74НС595) для керування більшою кількістю вихідних пристроїв (наприклад, 16 світлодіодами).

8. Використання регістра зсуву для реалізації простого лічильника: скласти схему лічильника на основі зсувного регістра зі зворотним зв'язком (наприклад, кільцевий лічильник або лічильник Джонсона).

9. Реалізація схеми керування семисегментним індикатором за допомогою зсувного регістра: використовуючи зсувний регістр (наприклад, 74НС595) та дешифратор (за потреби), розробити схему для відображення цифр на семисегментному індикаторі.

10. Дослідження роботи регістра з асинхронним та синхронним скиданням: скласти схему регістра на D-тригерах з асинхронним та синхронним входами скидання.

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного регістру.
4. Результати досліджень (рисунок, фото, скрін екрана і т. д.).

5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке регістр у цифровій схемотехніці?
2. Яке основне призначення регістрів?
3. Які типи регістрів вам відомі?
4. Опишіть принцип роботи кожного з різних типів регістрів.
5. Що таке послідовний регістр зсуву? Як відбувається процес зсуву даних у такому регістрі?
6. Що таке паралельний регістр? Які переваги та недоліки він має порівняно з послідовним регістром?
7. Поясніть принцип роботи універсального зсувного регістра. Які операції він може виконувати?
8. Що таке тактовий сигнал у контексті роботи регістрів? Яку роль він відіграє?
9. Що таке фронт тактового сигналу (зростаючий та спадаючий)? Як регістри реагують на різні фронти?
10. Для чого використовуються входи керування регістром (наприклад, Load, Shift, Clear, Enable)? Опишіть їхнє призначення.
11. Що таке регістр зберігання? Чим він відрізняється від зсувного регістра?
12. Поясніть принцип роботи інтегральної мікросхеми зсувного регістра, наприклад 74НС595. Які її основні виводи та їх призначення?
13. Як можна використовувати регістр зсуву для розширення кількості вихідних портів мікроконтролера? Наведіть приклад.
14. Як можна використовувати регістр зсуву для перетворення послідовного коду в паралельний і навпаки?
15. Що таке каскадне з'єднання регістрів зсуву? Для чого воно використовується?

Лабораторна робота 7

Аналого-цифрові перетворювачі

Теоретична частина

Аналого-цифровим перетворювачем (АЦП) називають електронний пристрій, який перетворює аналогові сигнали в цифрову форму. Процес аналого-цифрового перетворення полягає в переведенні безперервної функції напруги $u(t)$ у послідовність двійкових чисел, що містить три основні операції: дискретизацію, квантування та кодування.

Під час дискретизації безперервна функція перетворюється на послідовність її миттєвих значень, званих відліками. Сигнал подається у вигляді коротких імпульсів, амплітуда яких відповідає значенню сигналу в моменти вибірки. Таким чином, дискретизація замінює безперервний сигнал (рис. 7.1, а) набором значень, які він набував у різні моменти часу $t_1, t_2, t_3, \dots, t_k$. (рис. 7.1, б) [7].

Другою операцією є квантування, яке полягає в заміні миттєвих значень функції $u(t)$ найближчими дозволеними рівнями. Це означає, що сигнал дискретизується за рівнем. Для цього весь діапазон значень $u(t)$ ділиться на певні рівні квантування. Відстань Δ між сусідніми дозволеними рівнями називається кроком квантування. Квантування можна розглядати як округлення миттєвих значень сигналу до певної точності. Внаслідок квантування аналоговий сигнал $u(t)$, який є безперервною функцією, перетворюється на ступінчасту функцію (рис. 7.1, в).

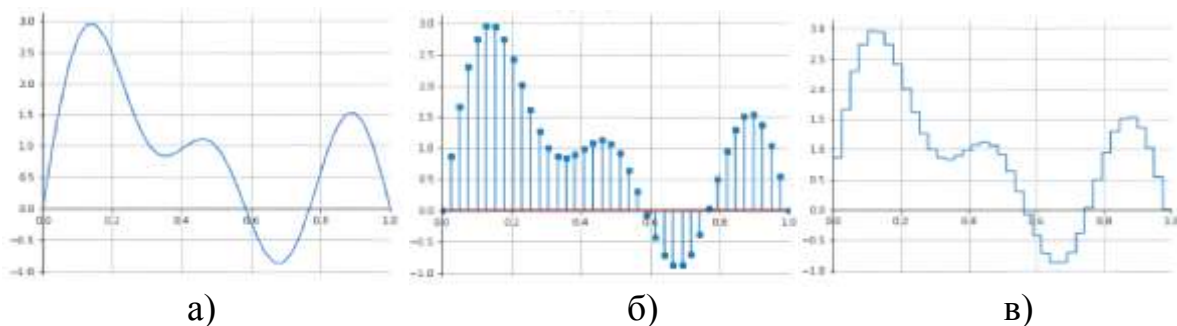


Рисунок 7.1 – Аналоговий (а) та відповідні йому дискретний (б) та квантований (в) сигнали

Під час третьої операції, що називається кодуванням, дискретні квантовані значення сигналу отримують відповідний цифровий код. Це кодування дозволяє подати значення сигналу у числовій формі. Процес кодування відбувається в рамках аналого-цифрового перетворення. Кількість бітів, використаних для кодування відліків аналогового сигналу,

визначається як розрядність квантування. Чим більше бітів у коді, тим більше рівнів квантування можна реалізувати, що забезпечує більш точне подання миттєвих значень початкового сигналу (зменшуючи похибку квантування).

Всі типи АЦП можна розділити на дві основні групи: АЦП миттєвих значень та АЦП середніх значень. Перші містять такі види, як послідовної лічби, послідовного наближення, паралельні, послідовно-паралельні та з проміжним перетворенням у певний інтервал часу. АЦП середніх значень подано інтегральними АЦП.

АЦП послідовної лічби містить компаратор, лічильник та ЦАП (рис. 7.2). На один з входів компаратора подається вхідний сигнал, а на інший – сигнал зворотного зв'язку з ЦАП.

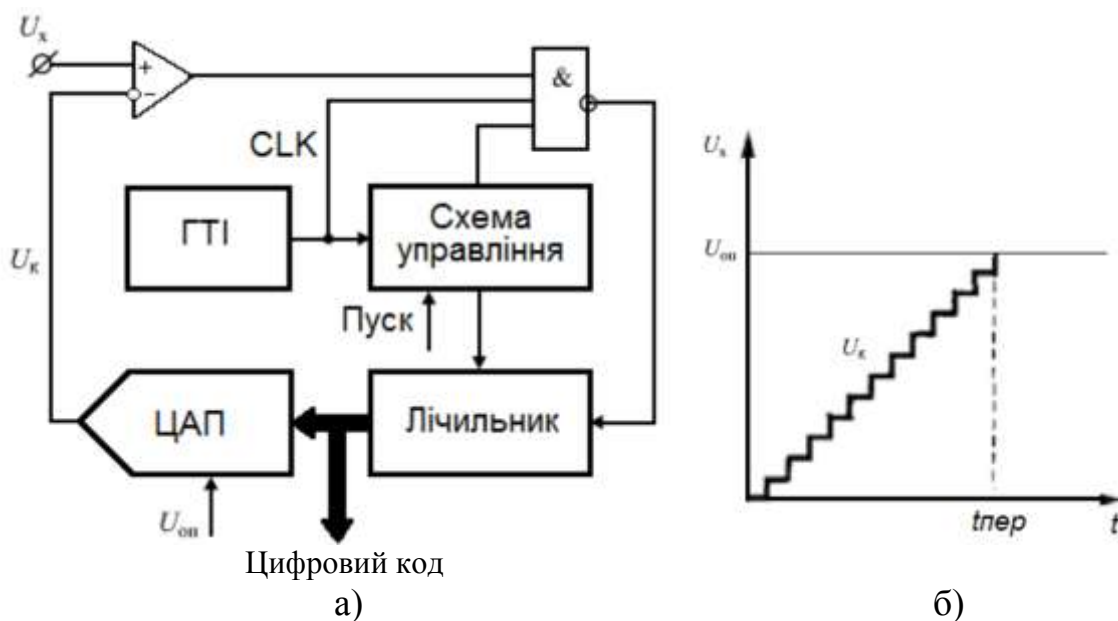


Рисунок 7.2 – АЦП послідовної лічби:

а) – структурна схема; б) – вихідна характеристика

Робота перетворювача починається з отримання імпульсу запуску, який активує лічильник, який підсумовує кількість імпульсів, що надходять від генератора тактових імпульсів (ГТВ). Вихідний код лічильника передається на ЦАП, де він перетворюється в напругу зворотного зв'язку. Цей процес триває до тих пір, поки напруга зворотного зв'язку не зрівняється з вхідною напругою. У цей момент вихідний сигнал компаратора стає нульовим, це блокує надходження тактових імпульсів до лічильника, що свідчить про завершення перетворення. Двійковий код, накопичений у лічильнику, пропорційний вхідній напрузі на момент завершення [9].

Час перетворення для АЦП цього типу є змінним і залежить від вхідної напруги. Максимальний час відповідає максимально можливій

вхідній напрузі, і розраховується за формулою, що містить розрядність двійкового лічильника N та частоту тактових імпульсів f :

$$t_{\text{пер.мах}} = (2^N - 1) \frac{1}{f}$$

АЦП послідовного наближення, відомий також як АЦП з порозрядним врівноваженням, є найпоширенішим типом послідовних АЦП. Основою роботи цього класу перетворювачів є принцип дихотомії (рис. 7.3), що передбачає послідовне порівняння вимірюваного значення з частками, такими як $1/2, 1/4, 1/8$ і т. д., від максимального можливого значення. Це дозволяє N -розрядному АЦП послідовного наближення виконати весь процес перетворення за N кроків (ітерацій), що є суттєвим покращенням у швидкодії порівняно з $2^N - 1$ кроками, необхідними для послідовного відліку. Наприклад, за $N=10$ цей вигош може досягати 100 разів, що дозволяє отримувати до $10^5 \dots 10^6$ перетворень на секунду. Водночас статична похибка таких перетворювачів, здебільшого пов'язана з використанням ЦАП, може бути дуже малою, що забезпечує роздільну здатність до 18 двійкових розрядів за частоти вибірки до 200 кГц (наприклад, DSP101 компанії Burr-Brown).

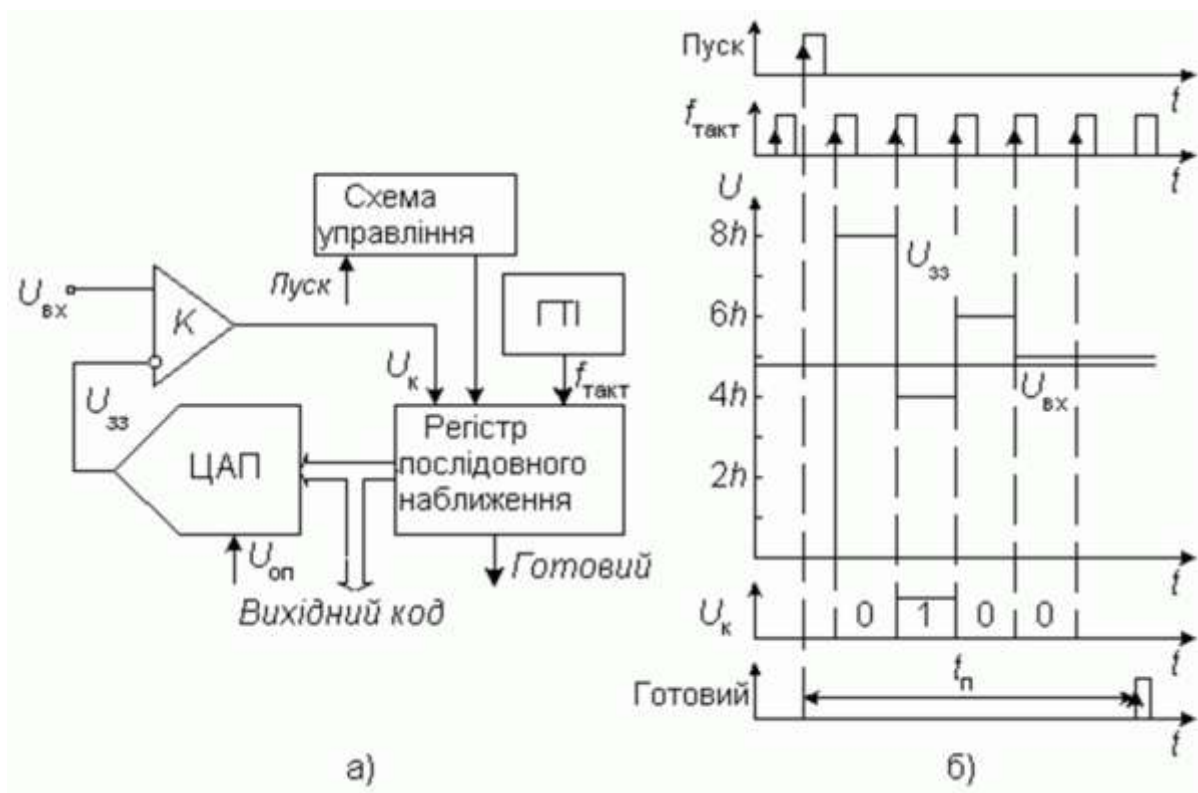


Рисунок 7.3 – АЦП послідовного наближення:
а) – структурна схема; б) – вихідна характеристика

Під час роботи без пристрою вибірки-зберігання апертурний час визначається проміжком між початком і фактичним завершенням перетворення, яке, як і в АЦП послідовного відліку, залежить від вхідного сигналу, тобто є змінним. Внаслідок цього виникають апертурні похибки, які також мають нелінійний характер. Тому для ефективного використання АЦП послідовного наближення доцільно підключати ПВЗ (підсилювач вибірки-зберігання) між його входом і джерелом перетвореного сигналу. Більшість сучасних ІМС АЦП послідовного наближення (наприклад, 12-розрядний MAX191, 16-розрядний AD7882 тощо) мають вбудовані пристрої вибірки-зберігання або, частіше, пристрої спостереження-збереження (track-hold), які керуються сигналом запуску АЦП. Пристрій спостереження-збереження постійно функціонує в режимі вибірки, переходячи в режим збереження лише під час перетворення сигналу.

Цей клас АЦП займає проміжне місце за швидкістю, вартістю та роздільною здатністю між послідовно-паралельними та інтегровальними АЦП і широко застосовується в системах управління, контролю та цифрової обробки сигналів.

Паралельні АЦП виконують квантування сигналу одночасно за допомогою групи компараторів, підключених паралельно до джерела вхідного сигналу. На рис. 7.4 подано реалізацію паралельного методу АЦ-перетворення для 3-розрядного числа. Три двійкові розряди можуть кодувати вісім різних значень, включно й нуль, що потребує сім компараторів. Вісім еквідистантних опорних напруг утворюються за допомогою резистивного дільника [11].

Якщо вхідна напруга знаходиться в межах від $5/2h$ до $7/2h$, де $h = U_{оп}/7$ – квант вхідної напруги, що відповідає одиниці молодшого розряду АЦП, то компаратори з 1-го по 3-й переходять у стан 1, а з 4-го по 7-й – у стан 0. Конвертація цієї групи кодів в тризначне двійкове число виконується логічним пристроєм, відомим як пріоритетний шифратор. Пряме підключення пріоритетного шифратора до виходу АЦП може призвести до помилок у зчитуванні вихідного коду. Наприклад, у випадку переходу з трьох на чотири, або в двійковому коді з 011 на 100, якщо старший розряд змінить свій стан раніше, виникає тимчасове число 111, що дорівнює сім. У цьому випадку величина помилки становитиме половину вимірювального діапазону.

Оскільки результати АЦ-перетворення зазвичай записуються у запам'ятовувальній пристрій, існує ймовірність отримання неправильних значень. Цю проблему можна вирішити, наприклад, за допомогою пристрою вибірки-зберігання (ПВЗ). Деякі інтегральні мікросхеми (ІМС) паралельних АЦП, такі як MAX100, обладнані надшвидкісними ПВЗ з часом вибірки близько 0,1 нс. Іншим рішенням є використання коду Грея, який змінює лише одну кодову позицію під час переходу між значеннями. Крім того, в деяких АЦП (наприклад, MAX1151) для зменшення ймовірності помилок у випадку паралельного АЦ-перетворення

застосовується двотактний цикл, коли спочатку фіксується стан виходів компараторів, а потім, після встановлення стану пріоритетного шифратора, активний фронт подається на синхровхід вихідного регістра для запису вихідного слова АЦП.

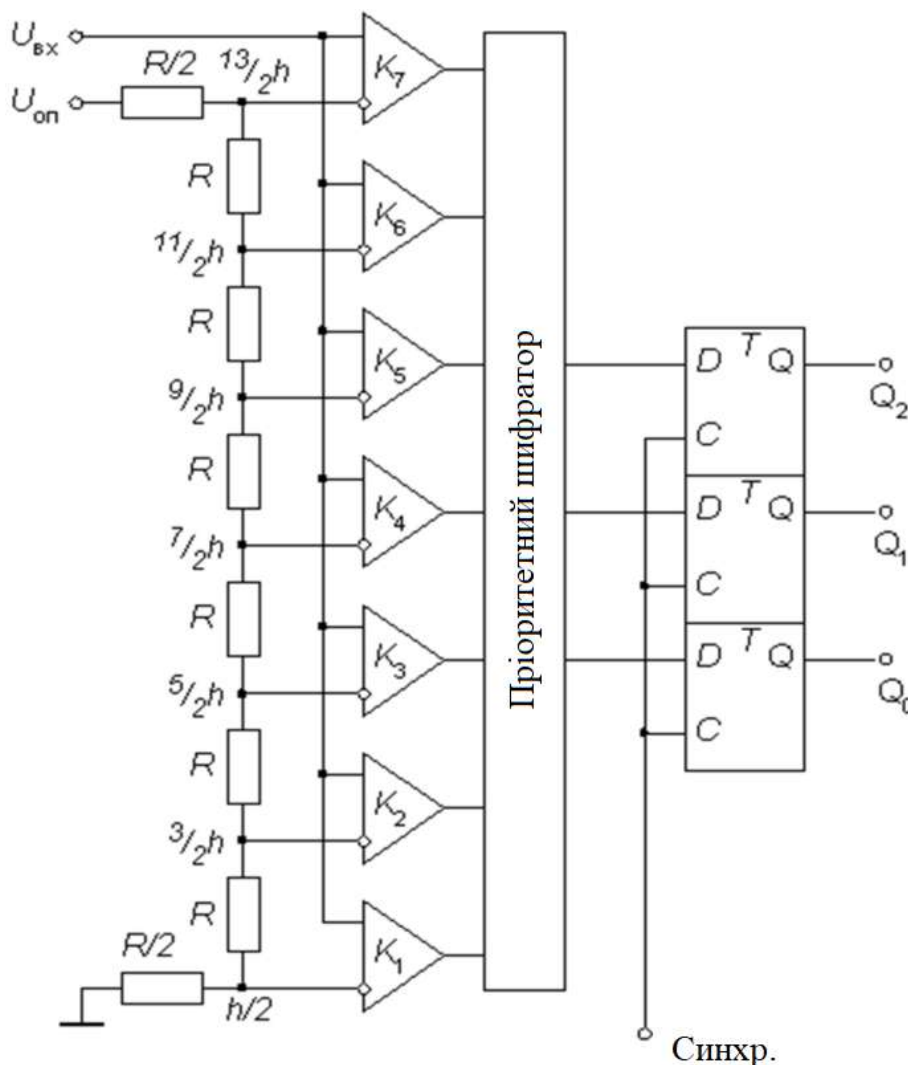


Рисунок 7.4 – Схема паралельного АЦП

Завдяки одночасній роботі компараторів, паралельний АЦП є найбільш швидким. Наприклад, восьмирозрядний перетворювач MAX104 забезпечує до 1 мільярда відліків на секунду з часом затримки сигналу не більше 1,2 нс. Проте, недоліком цієї схеми є її висока складність. Для N-розрядного паралельного АЦП потрібно 2^{N-1} компараторів і $2N$ узгоджених резисторів, що призводить до значних витрат (сотні доларів США) і високого споживання енергії. Наприклад, MAX104 споживає приблизно 4 Вт.

Послідовно-паралельні АЦП являють собою компроміс між прагненням до високої швидкодії та зменшенням витрат. Вони займають проміжне місце за роздільною здатністю та швидкістю між паралельними

АЦП і АЦП послідовного наближення. Цей тип перетворювачів поділяється на багатоступінчасті, багатотактні та конвеєрні.

У багатоступінчастому АЦП процес перетворення вхідного сигналу розділений у просторі. Наприклад, на рис. 7.5 показано схему двоступінчастого 8-розрядного АЦП.

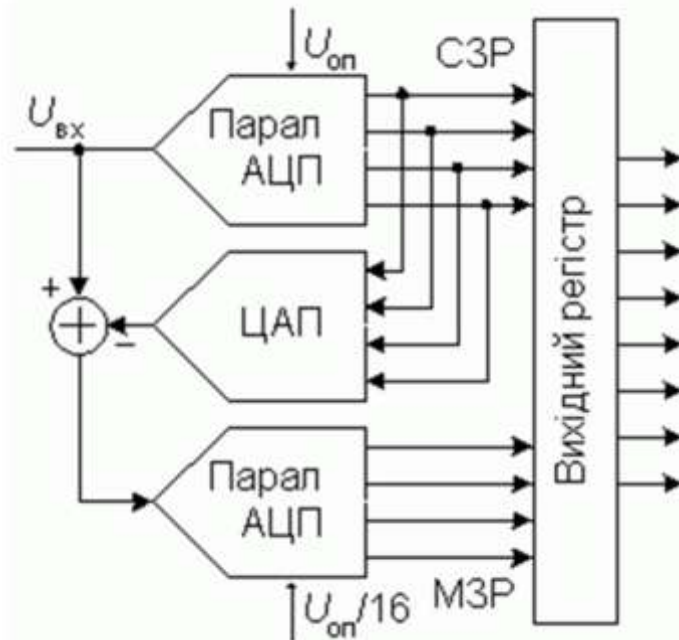


Рисунок 7.5 – Схема двоступінчастого 8-розрядного АЦП

Верхній АЦП виконує первинне перетворення сигналу, формуючи чотири старші розряди виходу. Цифрові сигнали з виходу цього АЦП надходять до вихідного регістра і одночасно на вхід 4-розрядного швидкодійного ЦАП. У багатьох інтегральних мікросхемах багатоступінчастих АЦП (таких як AD9042, AD9070 тощо) цей ЦАП реалізований за схемою підсумовування струмів з використанням диференціальних перемикачів, хоча деякі (наприклад, AD775, AD9040A) використовують ЦАП з підсумовуванням напруги. Різниця між вхідною та вихідною напругами ЦАП надходить на вхід другого АЦП, опорна напруга якого в 16 разів менша, ніж у першого АЦП. Як результат, квант другого АЦП в 16 разів менший, ніж у першого. Цей залишок, перетворений у цифрову форму другим АЦП, – це чотири молодші розряди виходу. Головна відмінність між першим і другим АЦП полягає в вимогах до точності: у першого АЦП точність має відповідати 8-розрядному перетворювачу, тоді як у другого допускається точність 4 розрядів.

Грубо наближена та точна величини мають відповідати одній і тій самій вхідній напрузі $U_{вх}(t_j)$. Проте, через затримку сигналу на першій стадії виникає тимчасове запізнення. Тому для цього методу важливо підтримувати вхідну напругу сталою за допомогою пристрою вибірки-зберігання до отримання всіх необхідних значень.

Розглянемо приклад іншого 8-розрядного багатотактного послідовно-паралельного АЦП (рис. 7.6). Тут процес перетворення розділений у часі.

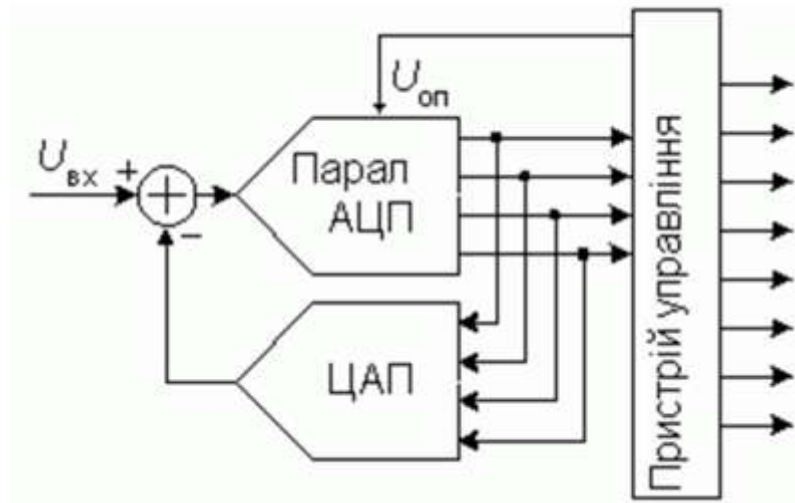


Рисунок 7.6 – Схема 8-розрядного послідовно-паралельного багатоконтактного АЦП

Перетворювач складається з 4-розрядного паралельного АЦП, де квант h визначається опорною напругою, 4-розрядного ЦАП і пристрою керування. Якщо максимальний вхідний сигнал становить 2,56 В, то в першому такті перетворювач працює з кроком квантування $h_1=0,16$ В, вхідний код ЦАП дорівнює нулю. Пристрій керування передає отримане від АЦП в першому такті слово до чотирьох старших розрядів вихідного регістра, подає це слово на вхід ЦАП і зменшує опорну напругу АЦП в 16 разів. Отже, в другому такті крок квантування $h_2=0,01$ В, а залишок, що утворюється під час віднімання вихідної напруги ЦАП від вхідної напруги, буде перетворений у молодший півбайт вихідного слова [16].

Варто зазначити, що 4-розрядні АЦП та ЦАП, які використовуються в цій схемі, потрібно, щоб мали точність 8 розрядів. Інакше можуть виникнути пропуски кодів, що призведе до того, що вихідний код АЦП не прийматиме деякі значення за монотонного збільшення вхідної напруги. Як і в попередньому перетворювачі, вхідна напруга багатотактного АЦП має залишатися сталою протягом перетворення, тому рекомендується використовувати пристрій вибірки-зберігання між його входом і джерелом вхідного сигналу.

Швидкодія розглянутого багатотактного АЦП визначається загальним часом перетворення 4-розрядного АЦП, часом спрацьовування цифрових схем управління та часом установлення ЦАП з похибкою не більше 0,2–0,3 кванти 8-розрядного АЦП. Час перетворення АЦП враховується в загальному часі перетворення двічі. Результат – за однакових умов цей тип перетворювача виявляється повільнішим за

двоступінчастий, хоча й простіший і дешевший. За швидкодією багатотактні АЦП займають проміжне місце між багатоступінчастими АЦП і АЦП послідовного наближення. Прикладами багатотактних АЦП є тритактний 12-розрядний AD7886 з часом перетворення 1 мкс та тритактний 16-розрядний AD1382 з часом перетворення 2 мкс.

Конвеєрні АЦП. Швидкодію багатоступінчастого АЦП можна покращити за допомогою конвеєрного принципу обробки вхідного сигналу. У традиційному багатоступінчастому АЦП спочатку формуються старші розряди виходу перетворювачем АЦП1, після чого відбувається період установлення вихідного сигналу ЦАП. В цей час АЦП2 не виконує роботи. На другому етапі, коли перетворюється залишок, АЦП1 простоє, поки працює АЦП2 [18].

Впроваджуючи елементи затримки для аналогових і цифрових сигналів між ступенями перетворення, ми отримуємо конвеєрний АЦП, схему 8-розрядного варіанта якого подано на рис. 7.7. Пристрій вибірки-зберігання ПВ32 виконує роль аналогового елемента затримки, а цифрового – чотири D-тригери, які затримують передачу старшого півбайта до вихідного регістра на один період тактового сигналу CLK.

Сигнали вибірки, створені з тактового сигналу, надходять до ПВ31 і ПВ32 у різний час. ПВ32 переходить у режим збереження пізніше, ніж ПВХ1, на період, що дорівнює загальній затримці поширення сигналу через АЦП1 і ЦАП. Задній фронт тактового сигналу контролює запис кодів у D-тригери та вихідний регістр. Повний цикл обробки вхідного сигналу триває приблизно два періоди CLK, але частота появи нових значень виходу дорівнює частоті тактового сигналу.

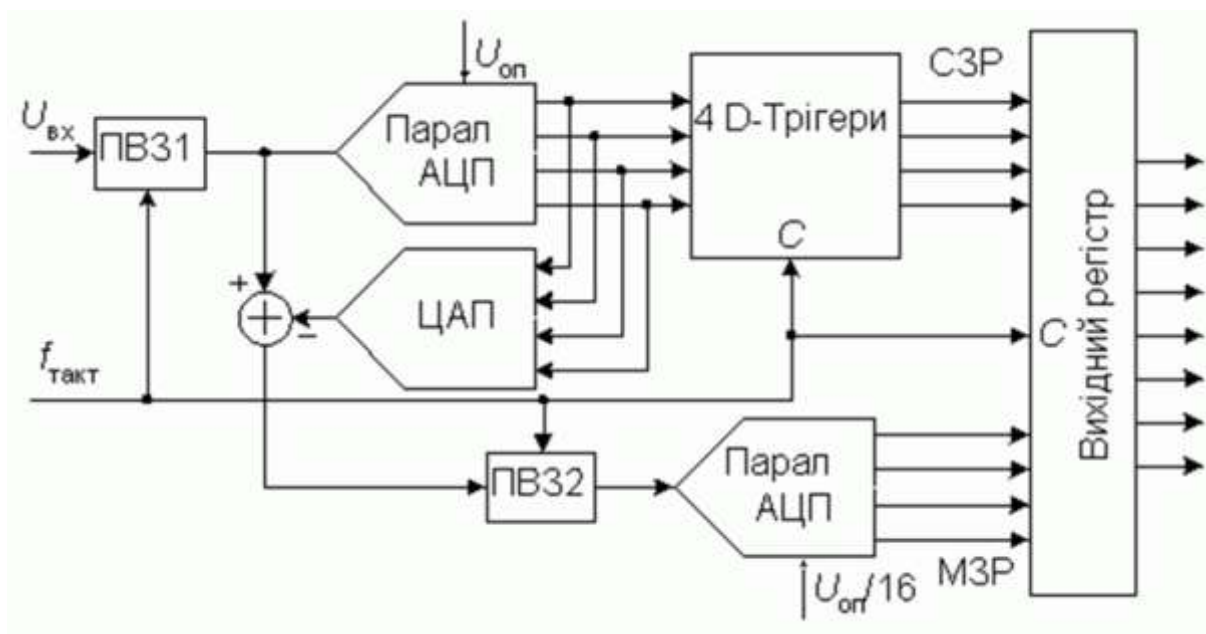


Рисунок 7.7 – Схема 8-розрядного конвеєрного АЦП

Таким чином, конвеєрна архітектура значно (у кілька разів) підвищує максимальну частоту вибірок багатоступінчастого АЦП. Хоча загальна затримка проходження сигналу залишається такою, як у звичайному багатоступінчастому АЦП з аналогічною кількістю ступенів, це не є критичним, оскільки час для подальшої цифрової обробки сигналів значно перевищує цю затримку. Завдяки цьому можна збільшити кількість ступенів АЦП без втрати швидкодії, водночас зменшуючи розрядність кожної з них. Більша кількість ступенів знижує складність АЦП. Наприклад, для створення 12-розрядного АЦП з чотирьох 3-розрядних потрібно 28 компараторів, тоді як реалізація з двох 6-розрядних потребуватиме вже 126 компараторів.

Багато сучасних багатоступінчастих АЦП мають конвеєрну архітектуру. Серед них, наприклад, 2-ступеневий 10-розрядний AD9040A, який виконує до 40 млн. перетворень на секунду (Мпс), та 4-ступеневий 12-розрядний AD9220 (10 Мпс), що споживає лише 250 мВт. Під час вибору конвеєрного АЦП потрібно враховувати, що багато з них не призначені для роботи з низькими частотами вибірок. Наприклад, виробник не рекомендує використовувати ІМС AD9040A з частотою перетворень менше 10 Мпс, а 3-ступеневий 12-розрядний AD9022 – з частотою нижче 2 Мпс. Це пов'язано з тим, що внутрішні ПВХ мають високу швидкість розряду конденсаторів збереження, і робота з великим тактовим періодом може призвести до значних змін у перетворюваному сигналі під час перетворення.

Порядок виконання роботи (off-line частина)

1. Вибрати мікросхеми (вітчизняні або зарубіжні) для реалізації свого варіанта завдання. Знайти в офіційній документації функціональне призначення виводів цих мікросхем. В цьому прикладі буде наведено роботу мікросхеми АЦП ADC0808.

Особливості ADC0808

- Легко підключається до всіх мікропроцесорів або може працювати автономно.
- Восьмиканальний 8-бітний АЦП модуль.
- Здатний безперервно вимірювати до 8 аналогових значень.
- Вбудований тактовий генератор відсутній, потрібен зовнішній осцилятор (синхронізація).
- Діапазон цифрового виходу від 0 до 255.
- Споживана потужність 15 мВт та час перетворення 100 мкс.
- За $V_{ref} = 5 \text{ В}$, кожен 19.53 мВ аналогового значення призводять до збільшення на один біт на цифровій стороні (розмір кроку).
- Доступний у 28-вивідних корпусах PDIP та SOIC.

Ця мікросхема ідеально підходить для використання з мікропроцесорами, такими як Raspberry Pi, BeagleBone тощо, або навіть як

окремий модуль АЦП. Кожен модуль АЦП для функціонування потребує тактового сигналу; ця інтегральна схема потребує зовнішнього тактового імпульсу для роботи. Отже, якщо ви шукаєте модуль АЦП з прийнятною роздільною здатністю 8 біт, який може вимірювати до 8 каналів, то ця інтегральна схема саме для вас.

Оскільки інтегральна схема ADC0808 може вимірювати до восьми аналогових напруг і не має внутрішнього тактового генератора, для її роботи потрібно трохи більше компонентів порівняно з її попередником ADC0804. Живлення мікросхеми може здійснюватися напругою +5 В. Напруга +5 В також має бути подана на V_{ref+} та вихідний строб (Output enable) для отримання вихідних даних. Живлення V_{ref} напругою +5 В забезпечить роботу мікросхеми з розміром кроку 19.53 мВ. Зовнішній тактовий сигнал потрібно підключити до тактового піна; це може бути як схема генератора, так і просто імпульс, що генерується мікроконтролером/мікропроцесором.

Відповідну вхідну аналогову напругу можна подавати на піни з IN1 до IN7, але мікросхема може зчитувати напругу лише з одного каналу одночасно. Вибір каналу здійснюється за допомогою пінів ADD A, ADD B та ADD C (табл. 7.1). Ці три біти мають бути встановлені відповідно до таблиці нижче для доступу до відповідного аналогового каналу.

Таблиця 7.1 – Вибір вхідного каналу

SELECTED ANALOG CHANNEL	ADDRESS LINE		
	C	B	A
IN0	L	L	L
IN1	L	L	H
IN2	L	H	L
IN3	L	H	H
IN4	H	L	L
IN5	H	L	H
IN6	H	H	L
IN7	H	H	H

Після вибору каналу його потрібно активувати, подавши на пін стробу адресної заціпки (Address latch enabled – ALE) короткий високий рівень.

Цифровий вихід буде отримано з пінів 2^{-1} (OUT 1) до 2^{-8} (OUT 8), а аналогову напругу потрібно підключити до піна $V_{in(+)}$ відповідно до схеми. Також зверніть увагу, що інший кінець джерела напруги (датчика/модуля) також має бути заземлений на схемі для коректної роботи аналого-цифрового перетворення.

Для початку процесу АЦП-перетворення необхідно подати високий рівень на пін START одразу після того, як на піні EOC з'явиться високий

рівень. Це можна зробити програмно або просто з'єднати піни EOC та START.

Призначення виводів мікросхеми АЦП наведено в табл. 7.2. Нумерацію виводів мікросхеми наведено на рис. 7.8.

Таблиця 7.2 – Призначення виводів мікросхеми ADC0808

Номер піна	Назва піна	Опис
1-5, 27, 28	Аналогові канали	Ці 7 пінів є вхідними пінами для аналогової напруги (від датчика).
6	START	Це вхідний піна, на який подається високий рівень для початку перетворення.
7	Кінець перетворення (EOC)	Це вихідний піна, який стає високим після завершення перетворення.
8, 14, 15, 18, 19, 20, 21	Вихід (2 ⁻¹ до 2 ⁻⁷)	Вихідні цифрові піни, які надають результат АЦП-перетворення.
9	Строб виходу (OUT EN)	Має бути встановлений у високий рівень для отримання вихідних даних на вихідних пінах.
10	CLOCK	На цей піна необхідно подавати тактові сигнали (0 В – 5 В) з частотою приблизно 20 МГц.
11	V _{cc}	Піна живлення мікросхеми, зазвичай 5 В.
12	V _{ref(+)}	Піна позитивної опорної напруги, зазвичай використовується +5 В.
13	Ground	Підключається до загальної шини (землі) схеми.
16	V _{ref(-)}	Піна негативної опорної напруги, зазвичай підключається до землі.
22	Строб адресної заціпки (ALE)	Цей піна потрібно тимчасово встановлювати у високий рівень для вибору аналогового каналу АЦП.
23,24,25	ADD A, ADD B, ADD C	Ці три піни використовуються для вибору вхідного аналогового каналу.

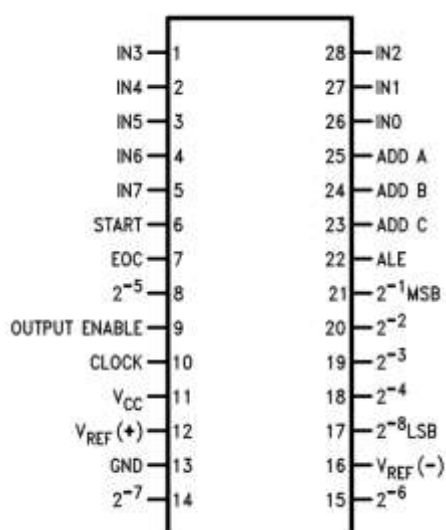


Рисунок 7.8 – Призначення та нумерація виводів ADC0808

2. Скласти на макетній платі робочу схему АЦП на мікросхемі ADC0808 (рис. 7.9).

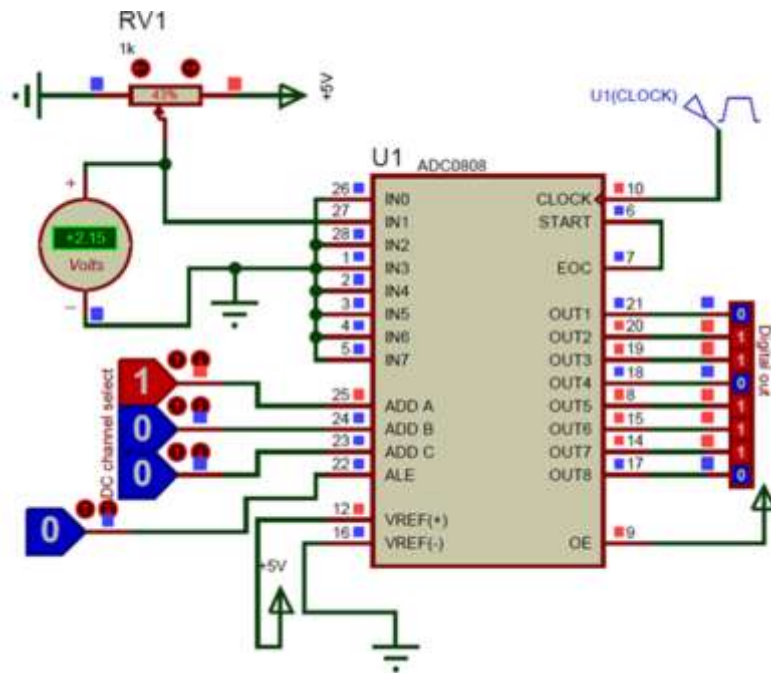


Рисунок 7.9 – Електрична схема дослідження АЦП ADC0808

Кінцевий вигляд макета зображено на рис. 7.10.

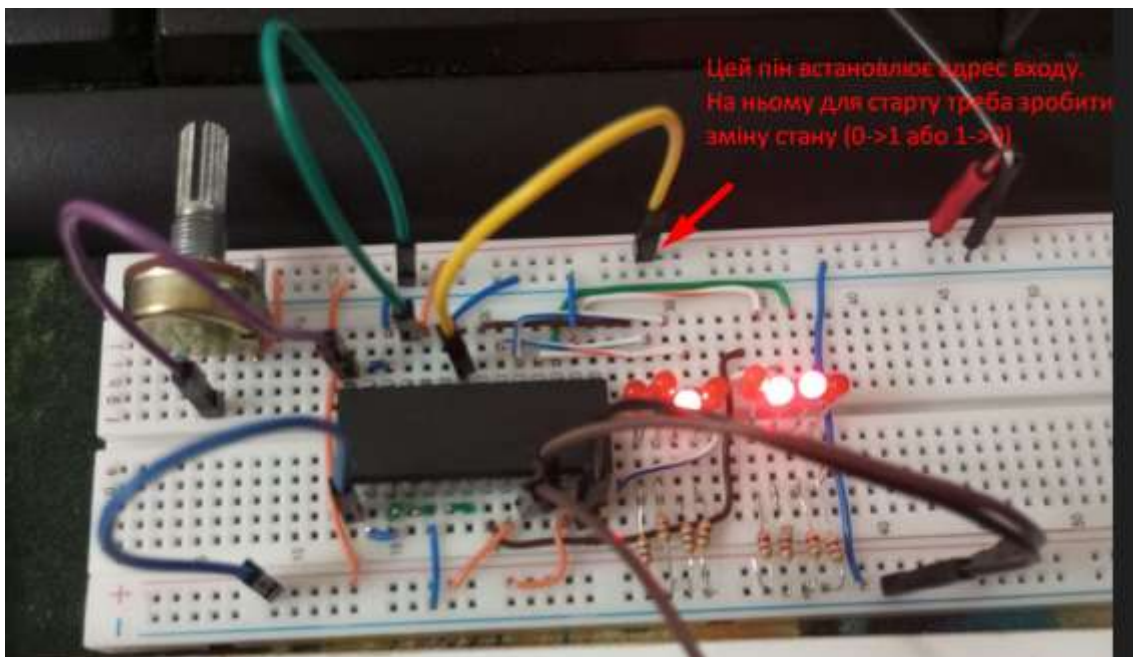


Рисунок 7.10 – Складений макет для дослідження АЦП ADC0808

3. Вибравши необхідний вхід, необхідно пам'ятати, що для зміни стану на OLE (це встановлення номера входу) достатньо просто перемкнути раз на протилежний рівень. Тільки тоді АЦП почне перетворення. Змінюючи вхідну напругу змінним резистором,

досліджуємо значення вихідного коду. Результати досліджень наведено на рис. 7.11.

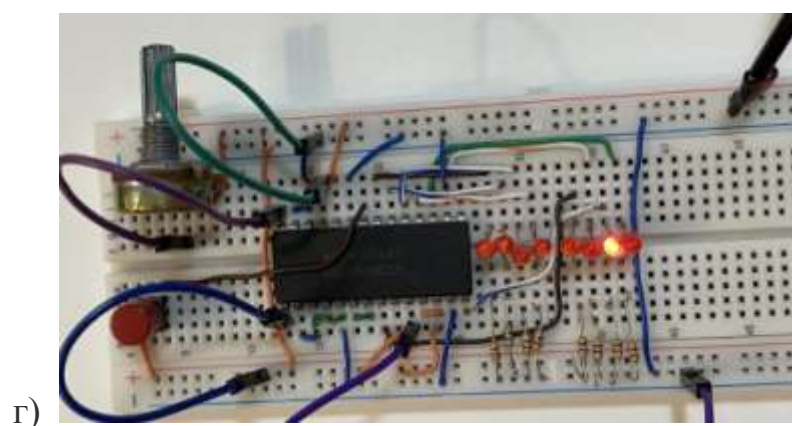
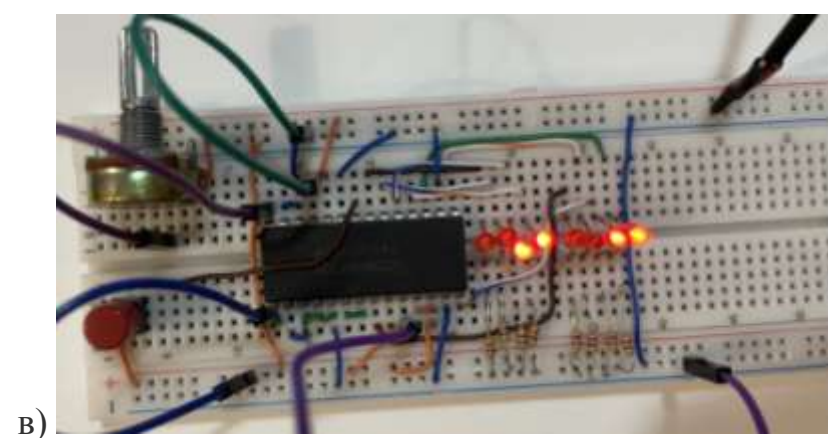
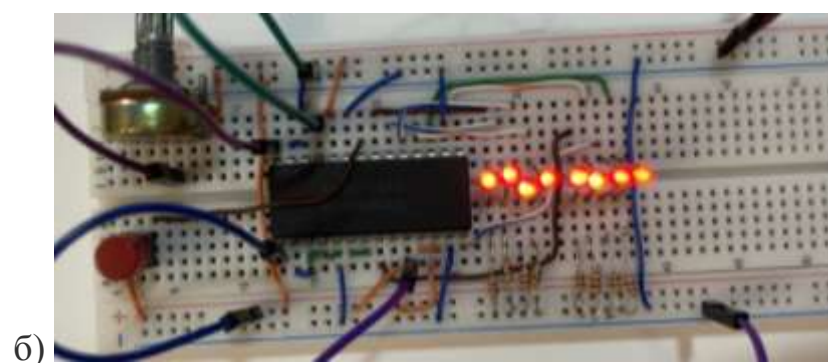
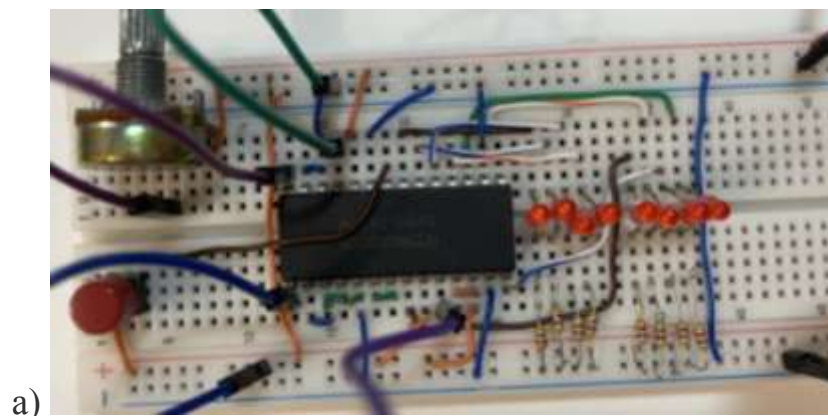


Рисунок 7.11 – Результати досліджень АЦП

Аналізуючи рис. 7.11 можна відмітити, що за відсутності сигналу на вході (положення резистора кінцеве 1) на виході (рис. 7.11, а), ми також маємо кодову комбінацію 00000000 (всі світлодіоди не горять). В крайньому положенні 2 змінного резистора (рис. 7.11, б), ми маємо 5 В на вході і відповідно кодову комбінацію на виході 11111111 (всі світлодіоди горять). В проміжних двох положеннях (рис. 7.11, в) і г) ми маємо відповідні вихідні комбінації 00110011 та 00000010, відповідно.

Можемо орієнтовно перерахувати рівень вхідної напруги. В нас 8-розрядний АЦП, тобто маємо $2^8=256$ рівнів. В нас опорна напруга 5 В. Відповідно, вартість одного рівня становить $5/256=19,5$ мВ.

Для рис. 7.11, в) маємо 00110011 = 204, а вхідна напруга дорівнює $204 \cdot 19,5=3,98$ В. Для рис. 7.11, г) маємо 00000010 = 64, а вхідна напруга дорівнює $64 \cdot 19,5=1,25$ В. Таким чином експериментальні дослідження підтвердили теоретичні відомості.

Порядок виконання роботи (on-line частина)

1. Для виконання лабораторної роботи в онлайн форматі можна використовувати різні моделювальні пакети. Обмежень щодо вибору немає, але перевага надається безкоштовним програмним пакетам. Ми будемо проводити лабораторну роботу в програмі Multisim.

2. Складемо схему для дослідження.

Схему в моделювальному пакеті наведено на рис. 7.12.

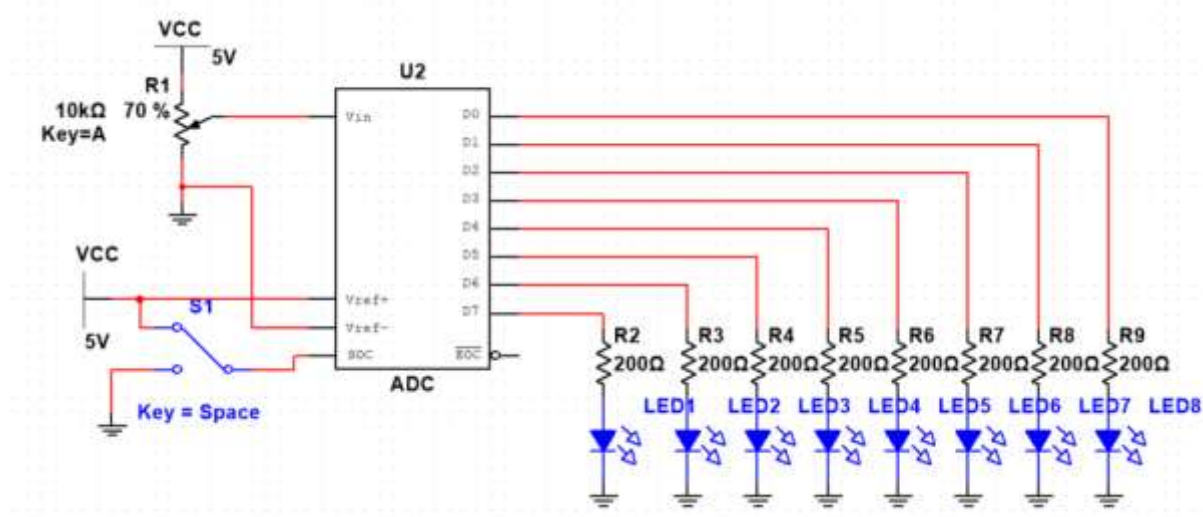
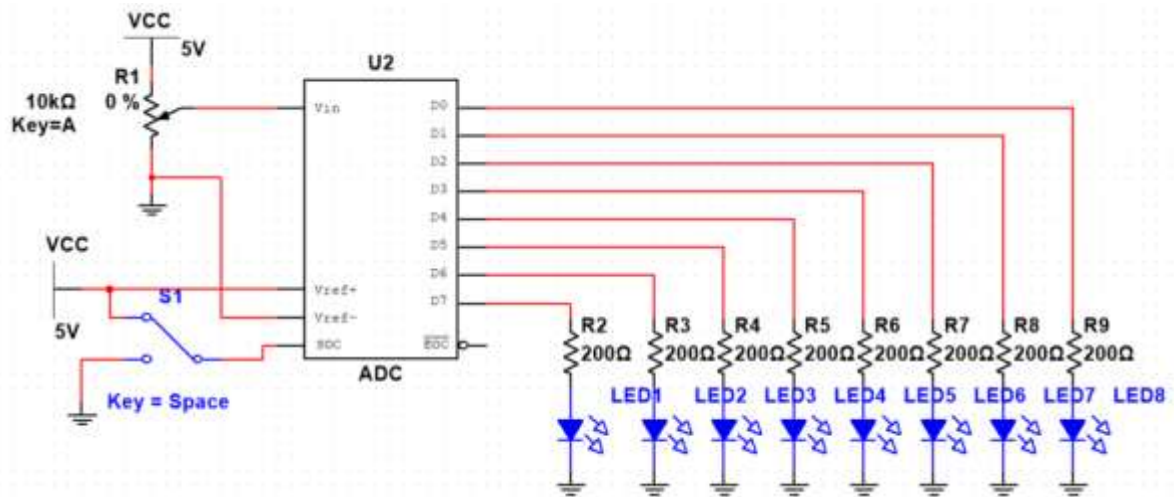


Рисунок 7.12 – Досліджувана схема АЦП в multisim

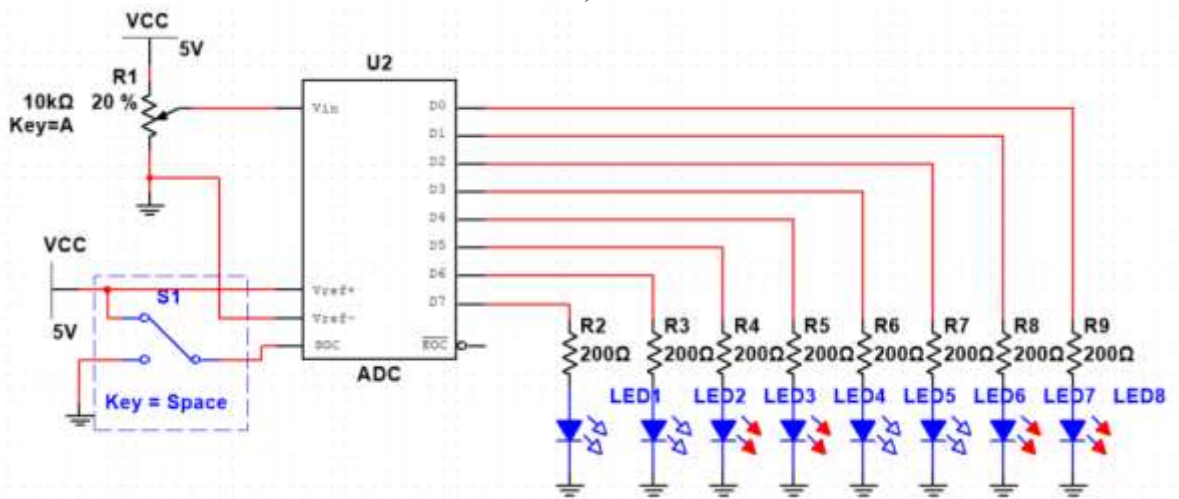
Змінним резистором R1 задаватимемо рівень вхідної напруги. Перемикачем S1 стартуватимемо перетворення. За кожної зміни опоры змінного резистора необхідно перемикачем заново стартувати процес перетворення.

3. Проводимо дослідження роботи АЦП в Multisim.

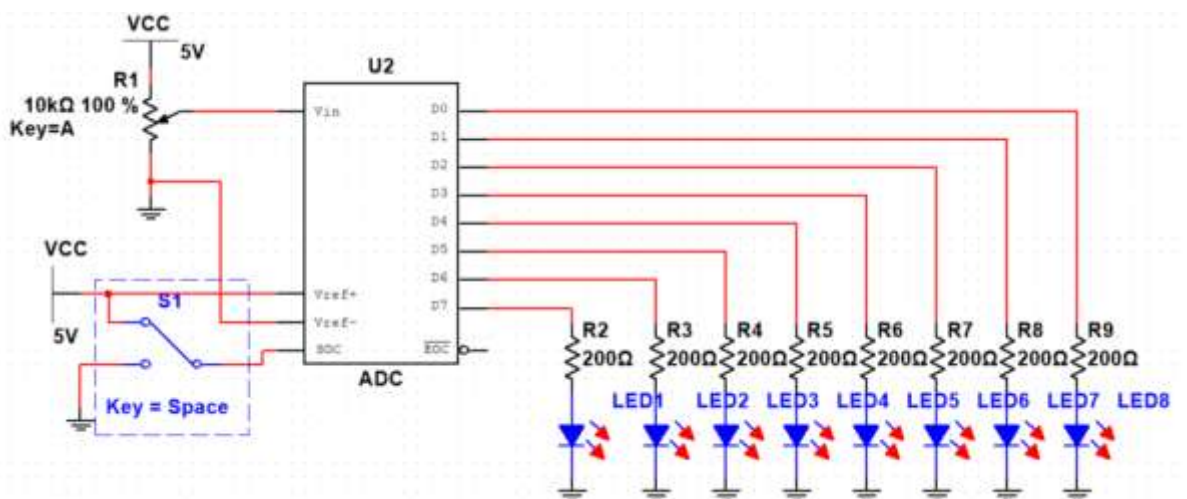
Результати дослідження цієї схеми наведено на рис. 7.13.



a)



b)



в)

Рисунок 7.13 – Результати досліджень АЦП в multisim

4. Висновки

Відповідно до аналізу схем рис. 7.13, а) – в), можна сказати, що вихідний код змінюється залежно від рівня вхідного сигналу. За максимального вхідного сигналу (рис. 7.13, в), вихідний код дорівнює 11111111. В нас опорна напруга також дорівнює 5 В. За мінімального (нульового) вхідного сигналу (рис. 7.13, а), вихідний код дорівнює 00000000.

Варіанти завдань

Студентам пропонується розробити схему, що реалізує процес аналого-цифрового перетворення відповідно до вибраного із табл. 7.3 варіанта та перевірити її роботу.

Таблиця 7.3 – Вихідні дані

Розрядність АЦП	Опорна напруга	Розрядність АЦП	Опорна напруга
4	1 В	4	1,5 В
6	2 В	6	2,5 В
8	3 В	8	3,5 В
4	2 В	4	2,5 В
6	3 В	6	3,5 В
8	4 В	8	5 В

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного АЦП.
4. Результати досліджень (рисунок, фото, скрін екрана і т. п.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке аналогово-цифровий перетворювач (АЦП) і яка його основна функція?
2. Які основні етапи процесу перетворення сигналу в АЦП?
3. Які типи АЦП існують і в чому їх основні відмінності?
4. Як працює АЦП з послідовною комутацією?
5. Що таке бітова глибина АЦП і як вона впливає на точність перетворення?
6. Які основні параметри АЦП, що впливають на його продуктивність?

7. Як працює АЦП на основі принципу «паралельного порівняння»?
8. Яка роль фільтрації в процесі підготовки сигналу до АЦП?
9. Як можна зменшити шум під час перетворення сигналу в АЦП?
10. Які методи калібрування АЦП використовуються для підвищення точності?
11. Яка роль АЦП в системах обробки сигналів і керування?
12. Як реалізується перетворення сигналу в АЦП з використанням мікроконтролера?
13. Які приклади застосування АЦП в промисловості та побуті?
14. Як здійснюється інтеграція АЦП в цифрові системи?
15. Які перспективи розвитку технологій АЦП у майбутньому?

Лабораторна робота 8

Двійкові суматори

Теоретична частина

Цифрові пристрої (ЦП) надзвичайно часто використовують арифметичні операції. Принципи виконання арифметичних дій з двійковими числами схожі на правила, що застосовуються в десятковій системі числення, і їх узагальнено в таблицях 8.1 – 8.4 [1].

Таблиця 8.1 – Правила та приклад двійкового додавання

Доданки <i>k</i> -го розряду	Сума <i>k</i> -го розряду	Перене- сення в <i>k</i> +1-й розряд		Приклад
0 + 0 = 0	0	0	+	1100 – перенесення
0 + 1 = 1	0	0		1101 – 1-й доданок
1 + 0 = 1	0	0		1100 – 2-й доданок
1 + 1 = 0	1	1		11001 – сума

Таблиця 8.2 – Правила та приклад двійкового віднімання

Зменшуване <i>k</i> -го розряду	Від’ємник <i>k</i> -го розряду	Різниця <i>k</i> -го розряду	Позика з <i>k</i> +1-го розряду		Приклад
0 – 0 = 0	0	0	0	–	010 – позика
0 – 1 = 1	1	1	1		1101 – зменшуване
1 – 0 = 1	0	1	0		1010 – від’ємник
1 – 1 = 0	1	0	0		0011 – різниця

Таблиця 8.3 – Правила та приклад двійкового множення

Множене <i>k</i> -го розряду	Множник <i>k</i> -го розряду	Добуток <i>k</i> -го розряду		Приклад
0 × 0 = 0	0	0	×	1010 – множене 101 – множник
0 × 1 = 0	1	0		+
1 × 0 = 0	0	0	0000	
1 × 1 = 1	1	1	1010	
				110010 – добуток

Під час виконання арифметичних дій з двійковими числами, що мають знак, використовується додатковий біт – знаковий розряд. Цей біт інформує про полярність числа: 0 вказує на додатне значення, а 1 – на від’ємне. Наприклад, десяткове число +5 у двійковому форматі із зазначенням знака (знаковий біт відокремлюється крапкою) матиме вигляд 0.101, а -3 буде подано як 1.011.

Таблиця 8.4 – Правила та приклад двійкового ділення

Ділене к-го розряду	:	Дільник к-го розряду	=	Частка к-го розряду	Приклад	
0	:	0	=	?	$\begin{array}{r} 1111 \\ - 10 \\ \hline 11 \\ - 10 \\ \hline 11 \\ - 10 \\ \hline 1 \end{array}$	10
0	:	1	=	0		111 – частка
1	:	0	=	?		
1	:	1	=	1		
						залишок

Під час додавання двох чисел з однаковим знаком їхні двійкові розряди підсумовуються, а отриманому результату присвоюється той самий знаковий біт, що й у вихідних доданків. Наприклад

$$\begin{array}{r} + 3 \\ + 2 \\ \hline 5_{10} \end{array} \Rightarrow \begin{array}{r} + 0.011 \\ + 0.010 \\ \hline 0.101_2 \end{array} \quad \begin{array}{r} + -3 \\ + -2 \\ \hline -5_{10} \end{array} \Rightarrow \begin{array}{r} + 1.011 \\ + 1.010 \\ \hline 1.101_2 \end{array}$$

Додавання двійкових чисел з протилежними знаками (що фактично є відніманням) є більш складним процесом. У такому випадку необхідно спочатку порівняти абсолютні значення чисел, потім виконати віднімання більшого за модулем числа від меншого, а кінцевому результату присвоїти знак того числа, яке мало більше абсолютне значення.

Для полегшення цієї процедури числа, що беруть участь в операції, часто перетворюють на обернений або доповняльний коди. Це пов'язано з тим, що дія віднімання (або алгебраїчного додавання) може бути зведена до звичайного арифметичного додавання двійкових чисел, якщо вони подані в одному з цих кодів. Варто зазначити, що для додатних чисел прямий, обернений та доповняльний коди виглядають ідентично, тоді як для від'ємних чисел їх подання відрізняються [6].

Щоб отримати обернений код від'ємного двійкового числа, необхідно в найстаршому біті (знаковому розряді) записати 1, а всі інші біти вихідного двійкового числа інвертувати (замінити 0 на 1, а 1 на 0).

Для подання від'ємного двійкового числа в доповняльному коді, знаковий розряд також встановлюється в 1, а решта бітів отримуються шляхом додавання одиниці до молодшого розряду оберненого коду цього числа.

Розглянемо приклади подання знакових двійкових чисел у прямому, оберненому та доповняльному кодах.

Число	Прямий код	Обернений код	Доповняльний код
+6	0.110	0.110	0.110
-5	1.101	1.010	1.011
-11	1.1011	1.0100	1.0101

Розглянемо процедуру віднімання чисел 5 і 3, та 3 і 5. Послідовність та взаємозв'язок операцій подано в табл. 8.5.

Таблиця 8.5 – Послідовність та взаємозв'язок операцій під час віднімання

	Обернений код	Доповняльний код
$A-B=$ $= 5-3=$ $=5+(-3) =2$	$A=0.101$ $B=1.100$ $\begin{array}{r} 0.101 \\ + \\ 1.100 \\ \hline 10.001 \\ \boxed{} \uparrow \\ 0.010 \end{array}$ Перенесення в молодший розряд. Результат додатний.	$A=0.101$ $B=1.100$ $\begin{array}{r} 0.101 \\ + \\ 1.101 \\ \hline 10.010 \\ \boxed{} \uparrow \\ 0.010 \end{array}$ Одиниця перенесення в молодший розряд ігнорується. Результат додатний.
$A-B=$ $= 3-5=$ $=3+(-5) = -2$	$A=0.011$ $B=1.010$ $\begin{array}{r} + 0.011 \\ 1.010 \\ \hline 1.101 \end{array}$ Перенесення в молодний розряд відсутнє. Результат від'ємний і поданий в оберненому коді.	$A=0.011$ $B=1.010$ $\begin{array}{r} 0.011 \\ + \\ 1.011 \\ \hline 1.110 \end{array}$ Результат від'ємний і поданий в доповняльному коді.

Аналізуючи наведені приклади, стає очевидним, що застосування оберненого коду в двійковому суматорі, призначеному для додавання багаторозрядних чисел, потребує наявності кола циклічного переносу. Натомість, за використання доповняльного коду таке коло не потрібне.

З огляду на це, можна дійти висновку, що в цифрових пристроях, зокрема в комп'ютерах, немає необхідності мати два окремих обчислювальних блоки – один для додавання, а інший для віднімання двійкових чисел. Завдяки простому математичному прийому, який полягає у поданні двійкових чисел в оберненому або доповняльному кодах, звичайний двійковий суматор може успішно виконувати обидві ці арифметичні операції.

Більш того, використовуючи двійковий суматор, можна також реалізувати операції множення та ділення двійкових чисел, тобто всі базові арифметичні дії. Це можливо тому, що множення по суті є серією послідовних операцій додавання, а ділення – серією послідовних операцій віднімання.

Двійкові суматори

Для додавання багаторозрядних двійкових чисел $A=a_n a_{n-1} \dots a_0$ та $B=b_n b_{n-1} \dots b_0$ застосовується порозрядне додавання з урахуванням переносів між сусідніми розрядами. Ключовим елементом у структурі багаторозрядних суматорів є комбінаційний однорозрядний суматор. Цей елемент виконує додавання трьох однобітових вхідних сигналів: біта

першого числа (a_i), біта другого числа (b_i) для поточного розряду, а також біта переносу (p_i) з попереднього, молодшого розряду. Результатом цієї операції є два вихідні біти: біт суми (S_i) для цього розряду та біт переносу (p_{i+1}) до наступного, старшого розряду.

На рис. 8.1 подано умовне позначення однорозрядного суматора та таблиця істинності, що відображає всі можливі комбінації вхідних сигналів і відповідні вихідні результати.

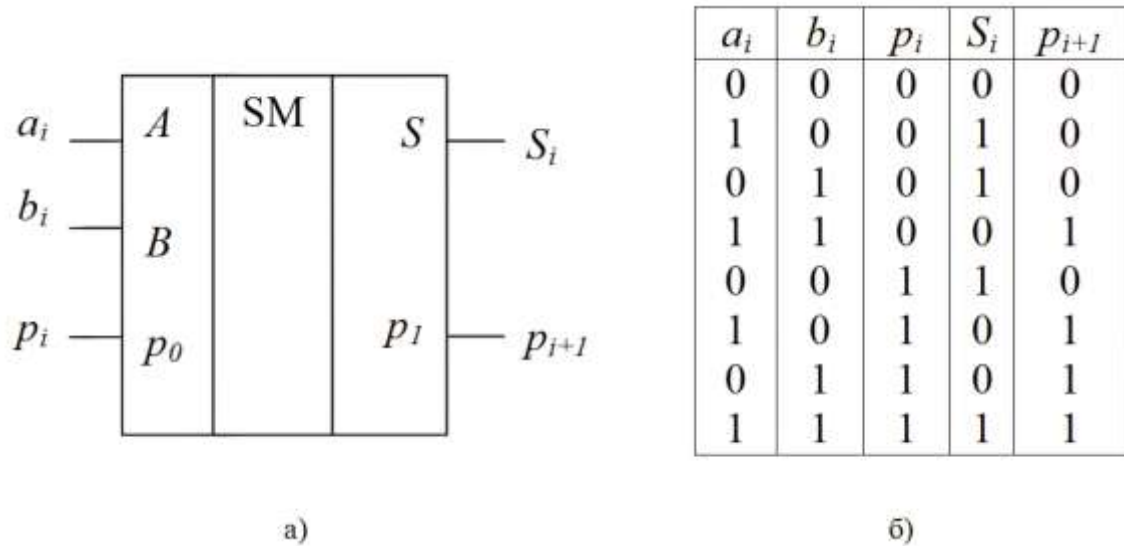


Рисунок 8.1 – Умовне позначення (а) і таблиця істинності (б) однорозрядного суматора

Для синтезу схеми однорозрядного суматора запишемо вирази для S_i і p_{i+1} (виходів суматора):

$$S_i = a_i \bar{b}_i \bar{p}_i + \bar{a}_i b_i \bar{p}_i + \bar{a}_i \bar{b}_i p_i + a_i b_i p_i = (a_i \bar{b}_i + \bar{a}_i b_i) \bar{p}_i + (\bar{a}_i \bar{b}_i + a_i b_i) p_i = (a_i \oplus b_i) \bar{p}_i + \overline{(a_i \oplus b_i)} p_i = (a_i \oplus b_i) \oplus p_i \quad (8.1)$$

$$p_{i+1} = a_i b_i \bar{p}_i + a_i \bar{b}_i p_i + \bar{a}_i b_i p_i + a_i b_i p_i = a_i b_i (\bar{p}_i + p_i) + a_i \bar{b}_i p_i + \bar{a}_i b_i p_i = a_i b_i + p_i (a_i \oplus b_i) \quad (8.2)$$

На рис. 8.2 зображено електричну схему однорозрядного суматора, розроблену згідно з математичними формулами (8.1) та (8.2).

Паралельний багаторозрядний суматор може бути сконструйований з кількох однорозрядних суматорів, кількість яких відповідає розрядності чисел, що додаються. З'єднання між цими суматорами здійснюється шляхом передачі сигналу переносу з виходу одного розряду на вхід переносу наступного, старшого розряду. Такий принцип організації перенесення називається послідовним.

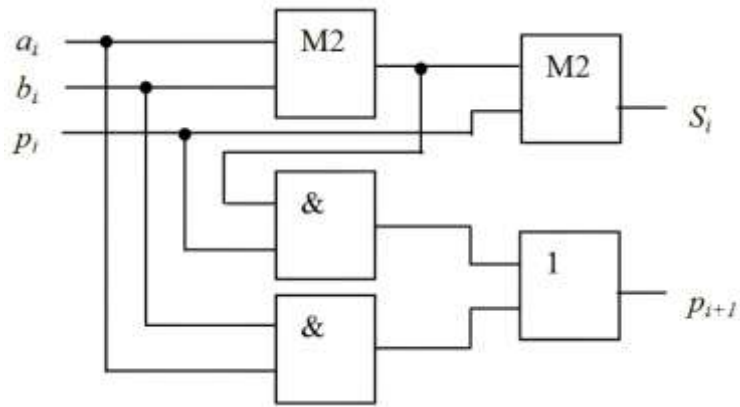


Рисунок 8.2 – Схема однорозрядного суматора

На рис. 8.3 наведено приклад побудови 3-розрядного суматора за таким принципом. У суматорах цього типу сигнал переносу передається від молодшого розряду до старшого послідовно, у міру обчислення суми в кожному окремому розряді. За найгірших умов, коли, наприклад, додаються числа, що складаються з одних одиниць (11...11) та число, де лише молодший розряд є одиницею (00...01), одиничний сигнал переносу буде поширюватися через усі розряди суматора, від найменшого до найбільшого. Отже, максимальний час затримки сигналу переносу

$$T_{зм.п.пер.} = n \cdot t_{зм.п.пер.},$$

де $t_{зм.п.пер.}$ – час затримки сигналу переносу в одному розряді,
 n – число розрядів суматора.

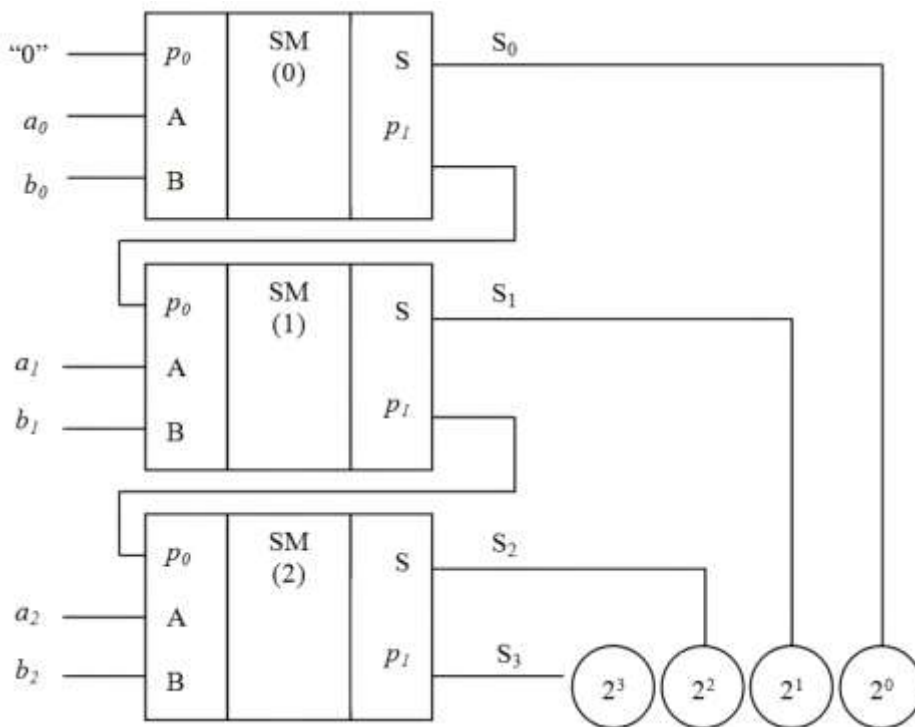


Рисунок 8.3 – Функціональна схему 3-розрядного паралельного суматора, де перенесення між розрядами відбувається послідовно

Така конструкція суматора відрізняється схемотехнічною простотою в частині формування сигналів переносу, однак його швидкодія є відносно невисокою.

Суматори з паралельним перенесенням мають значно кращі показники швидкості обробки даних. У таких суматорах сигнали переносу для всіх розрядів генеруються одночасно, що досягається завдяки використанню спеціалізованих схем прискореного перенесення [13].

Двійкові віднімачі

Вище було зазначено, що віднімання двійкових чисел можна замінити їхнім додаванням. Для цього число, від якого віднімають, і число, яке віднімають, подаються в оберненому або доповняльному коді.

Розглянемо як двійковий суматор може виконувати операцію віднімання. На рис. 8.4, а) показано 4-розрядний двійковий віднімач, де число, яке віднімається ($B=b_2b_1b_0$), подано в оберненому коді. Ця схема відрізняється від паралельного двійкового суматора наявністю трьох інвертувальних елементів, що перетворюють від'ємник на його обернений код (шляхом інверсії кожного біта), а також колом додаткового (циклічного) переносу, яке з'єднує вихід переносу найстаршого розряду з входом переносу наймолодшого розряду.

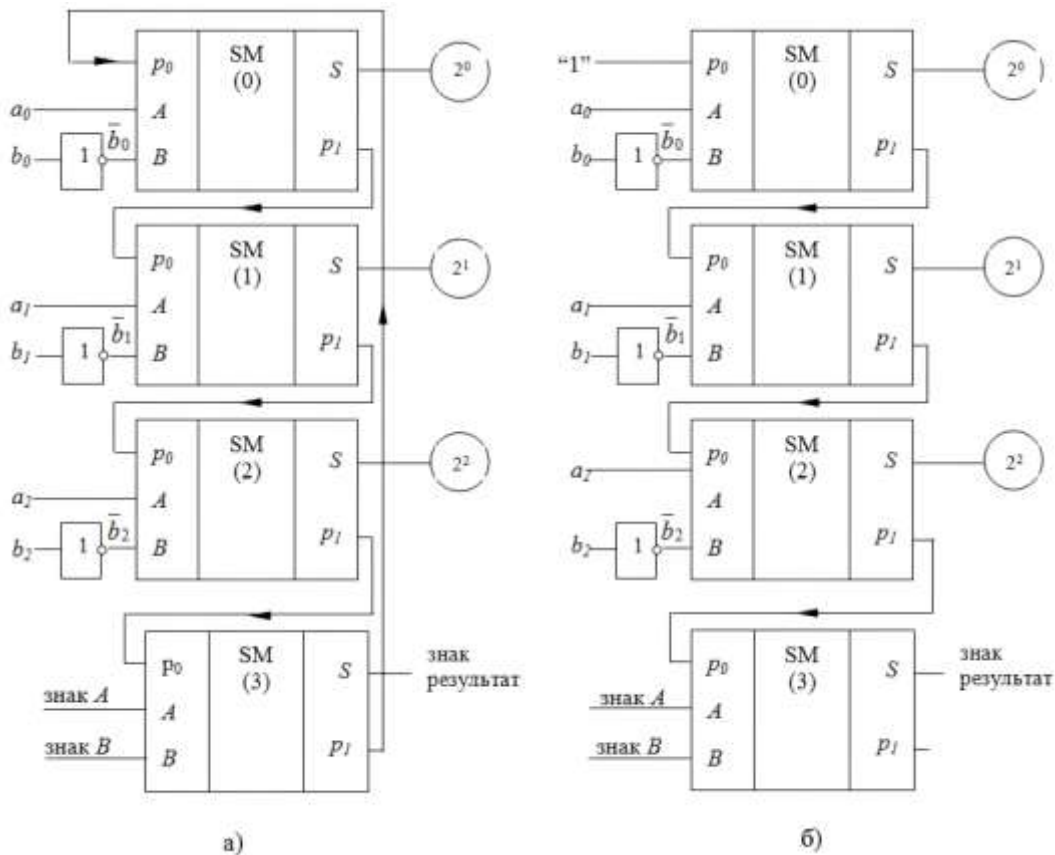


Рисунок 8.4 – Функціональна схема двійкових віднімачів, у яких від'ємник подано в оберненому (а) та доповняльному коді (б).

На рис. 8.4, б) зображено 4-розрядний віднімач, у якому від'ємник (B) подано в доповняльному коді. Таке подання досягається шляхом додавання одиниці до наймолодшого біта оберненого коду від'ємника (що реалізується подачею «1» на вхід переносу p_0 молодшого розряду). У цьому випадку потреба в колі циклічного переносу зникає.

Універсальні двійкові суматори-віднімачі

Оскільки двійкові суматори здатні виконувати як операцію додавання, так і віднімання, ми розробимо комбіновану схему – суматор-віднімач. За основу буде взято схему віднімача (рис. 8.4, б), де замість чотирьох однорозрядних суматорів застосуємо дві мікросхеми, що містять по два 2-розрядні суматори (К155ИМ2), а інвертори замінимо на логічні елементи типу «М2». Модифікований варіант цієї схеми подано на рис. 8.5. Для того, щоб ця схема функціонувала як двійковий суматор, необхідно подати низький логічний рівень (0) на керівний вхід V . В такому випадку числові розряди двійкового числа B ($b_2b_1b_0$) пройдуть через логічні елементи «М2» без змін (інверсії). До того ж, логічний нуль з керівного входу V буде передано на вхід переносу молодшого розряду двійкового суматора. Отже, за $V=0$ схема виконує додавання двійкових чисел A та B [2].

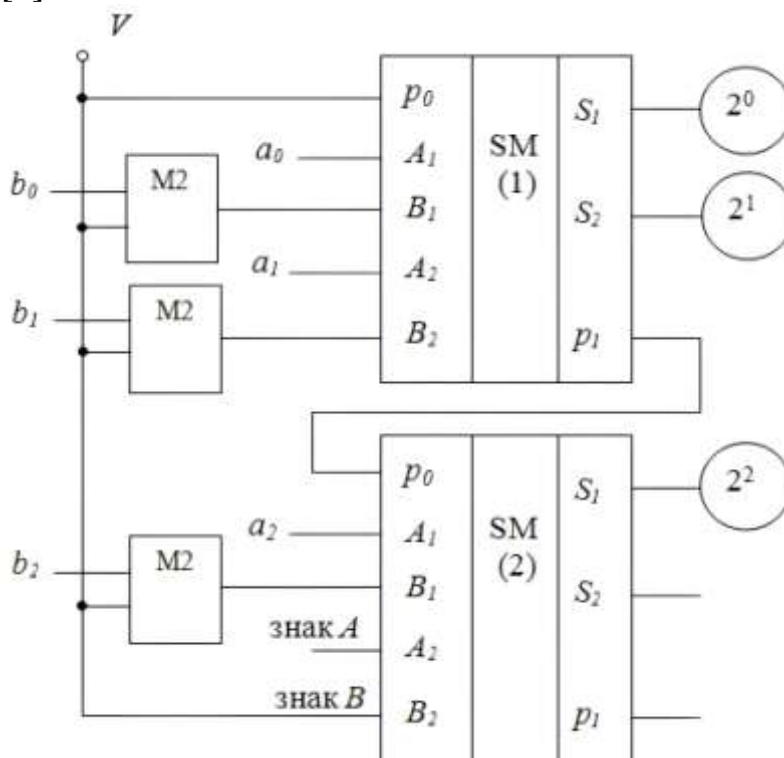


Рисунок 8.5 – Функціональна схема суматора-віднімача

Для того, щоб ця схема виконувала функцію віднімання, на її керівний контакт V необхідно подати високий логічний рівень 1. За такої умови логічні елементи, позначені як «М2», інвертують біти двійкового

числа B (b_2, b_1, b_0). Окрім цього, логічна одиниця з керівного входу V передається на вхід переносу наймолодшого біта двійкового суматора. Це приводить до того, що число B набуває вигляду у додатковому коді, завдяки чому схема здійснює віднімання числа B від числа A ($A-B$). На вихідному контакті S_2 суматора SM_2 , коли на V подано логічну 1 (режим віднімання), формується знак кінцевого результату (різниці), а коли на V подано логічний 0 (режим додавання) – біт суми з ваговим коефіцієнтом 2^3 .

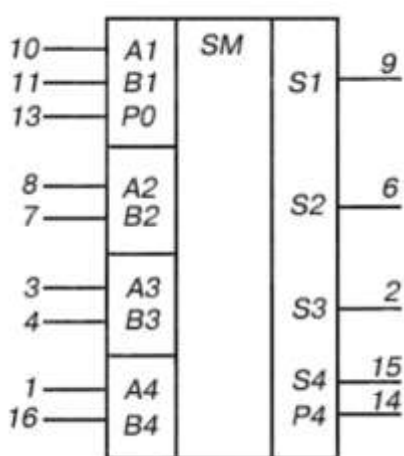
Порядок виконання роботи (off-line частина)

1. Вибираємо мікросхему двійкового суматора.

Виходячи з критеріїв наявності, ми вибрали мікросхему КМ155ИМ3. Мікросхема КМ155ИМ3 є чотирирозрядним двійковим суматором з паралельним перенесенням, що належить до популярної логічної серії К155 (аналог TTL-серії 74). Її основне призначення – додавання двох чотирирозрядних двійкових чисел. Вона також має вхід для сигналу переносу з попереднього молодшого розряду та вихід сигналу переносу до наступного старшого розряду, що дозволяє каскадувати декілька таких мікросхем для обробки чисел більшої розрядності.

Кожен з чотирьох однорозрядних суматорів, що входять до складу КМ155ИМ3, має три входи: два для бітів операндів (A та B) та один для вхідного сигналу переносу. На виході кожного однорозрядного суматора формуються два сигнали: біт суми та біт вихідного переносу. Завдяки паралельній організації перенесення, сигнали переносу генеруються одночасно для всіх розрядів, що значно підвищує швидкодію суматора порівняно з суматорами з послідовним перенесенням.

Іноземними аналогами є SN7483N, SN7483J. Технічні характеристики мікросхеми наведено в табл. 8.6. На рис. 8.6 наведено призначення виводів мікросхеми (б) та її умовне позначення (а).



а)

- 1 – вхід доданка A_4 ;
- 2 – вихід суми S_3 ;
- 3 – вхід доданка A_3 ;
- 4 – вхід доданка B_3 ;
- 5 – напруга живлення;
- 6 – вихід суми S_2 ;
- 7 – вхід доданка B_2 ;
- 8 – вхід доданка A_2 ;
- 9 – вихід суми S_1 ;
- 10 – вхід доданка A_1 ;
- 11 – вхід доданка B_1 ;
- 12 – загальний;
- 13 – вхід переносу P_0 ;
- 14 – вихід переносу четвертого розряду P_4 ;
- 15 – вихід суми S_4 ;
- 16 – вхід доданка B_4 ;

б)

Рисунок 8.6 – Умовне графічне позначення КМ155ИМ3 (а) та призначення виводів мікросхеми КМ155ИМ3 (б)

Таблиця 8.6 – Технічні характеристики мікросхеми КМ155ИМ3

Параметр	Значення
Номінальна напруга живлення	5 В ± 5 %
Вихідна напруга низького рівня	не більше 0,4 В
Вихідна напруга високого рівня	не менше 2,4 В
Завадостійкість	не менше 0,4 В
Вхідний пробивний струм	не більше 1 мА
Струм короткого замикання	-18...-55 мА
Струм споживання	не більше 128 мА
Споживана статична потужність	не більше 670 мВт
Час затримки поширення у разі ввімкнення від виводу 13 до 15	не більше 55 нс
Час затримки у разі вимкнення від виводу 13 до 15	не більше 55 нс
Час затримки поширення по колу суми	не більше 37,5 нс
Час затримки поширення по колу переносу	не більше 40 нс
Номінальна напруга живлення	5 В ± 5 %
Вихідна напруга низького рівня	не більше 0,4 В
Вихідна напруга високого рівня	не менше 2,4 В

2. Складаємо макет (рис. 8.7). Потрібно зазначити, що на кожен вихід ми підключаємо ще послідовно діод та резистор, щоб візуально бачити результат на виході.

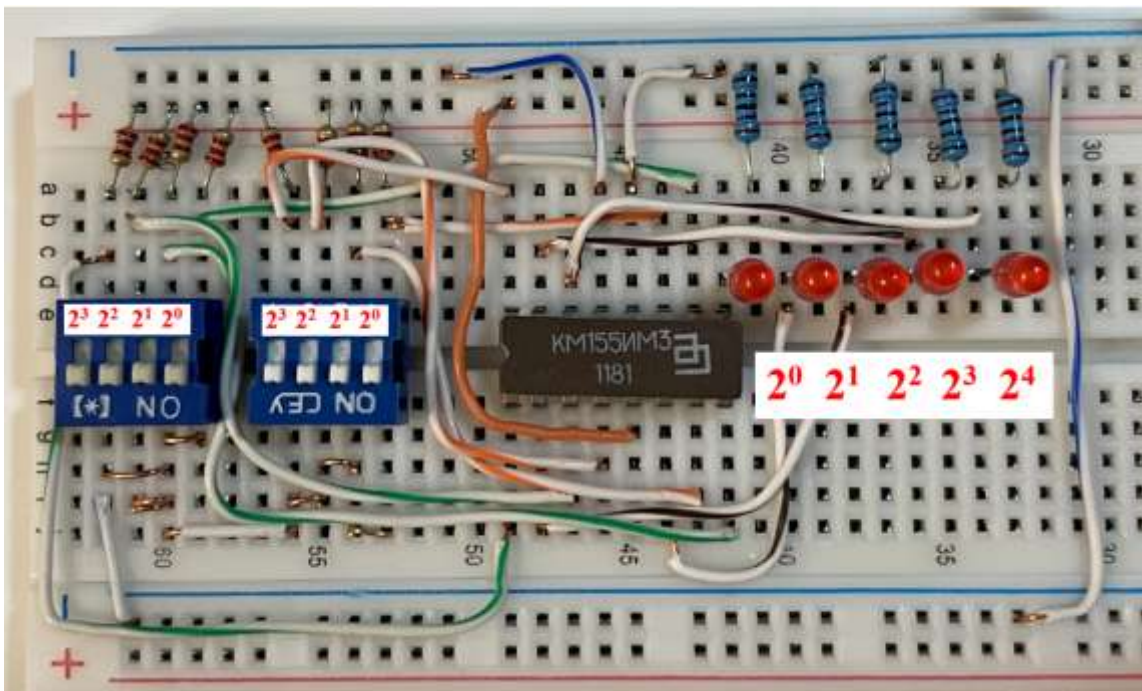


Рисунок 8.7 – Схема макета двійкового суматора на КМ155ИМ3

В перемикачах положення вверх дорівнює логічній «1», вниз – «0». Ваги розрядів на рис. 8.7 вказані як на перемикачах, так і на вихідних діодах

3. Проводимо дослідження двійкового суматора.

Результати дослідження роботи двійкового суматора макета наведено на рис. 8.8.

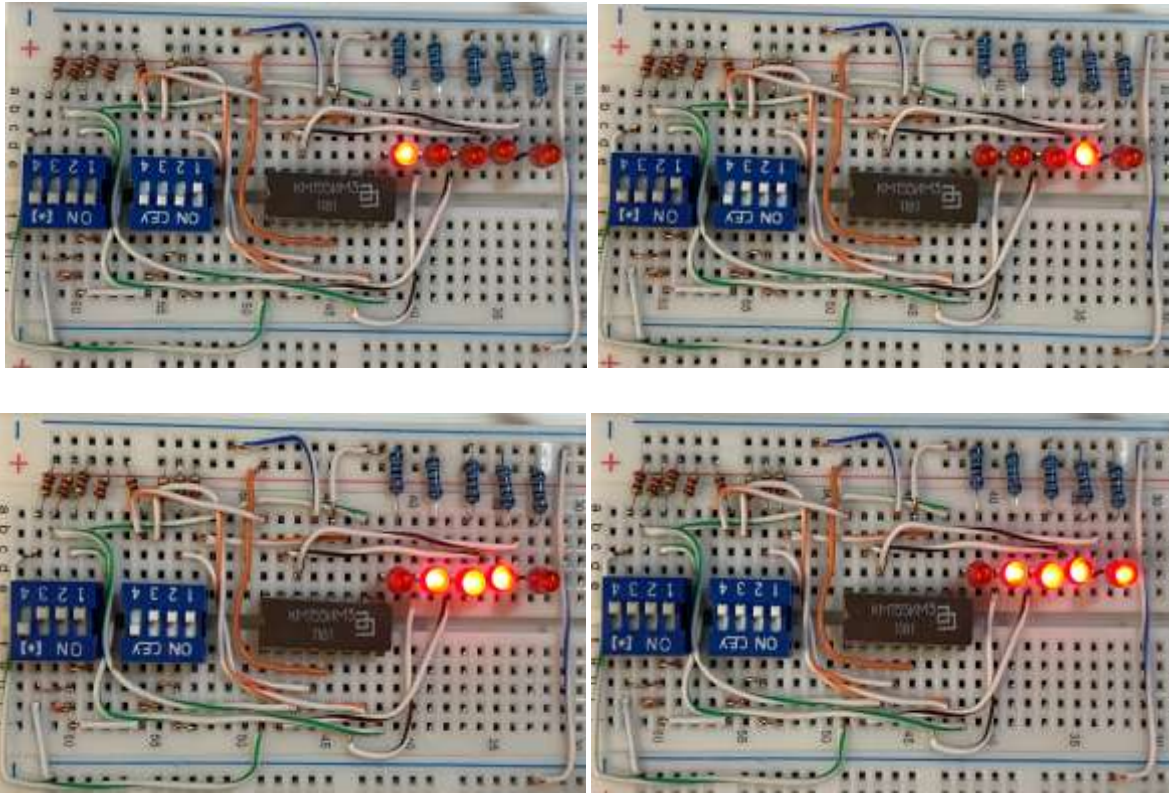


Рисунок 8.8 – Результати дослідження суматора на KM155IM3

4. Висновки

Відповідно до результатів (рис. 8.8) маємо (в дужках біля двійкового числа його еквівалент в десятковій системі числення):

$$\begin{aligned}0000 (0) + 0001 (1) &= 00001 (1); \\0001 (1) + 0111 (7) &= 01000 (8); \\0111 (7) + 0111 (7) &= 01110 (14); \\1111 (15) + 1111 (15) &= 11110 (30).\end{aligned}$$

Таким чином ми переконались у правильній роботі двійкового суматора на мікросхемі KM155IM3.

Порядок виконання роботи (on-line частина)

1. Складаємо в Multisim схему двійкового суматора на мікросхемі 74LS83D як показано на рис. 8.9.

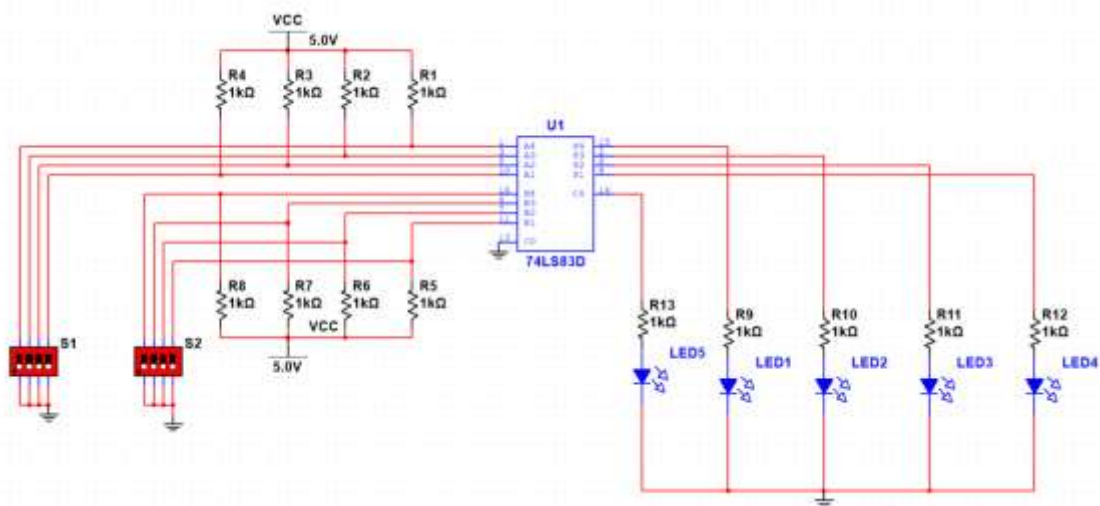


Рисунок 8.9 – Схема двійкового суматора в Multisim

2. Проводимо дослідження. Результати зображено на рис. 8.10 – 8.13.

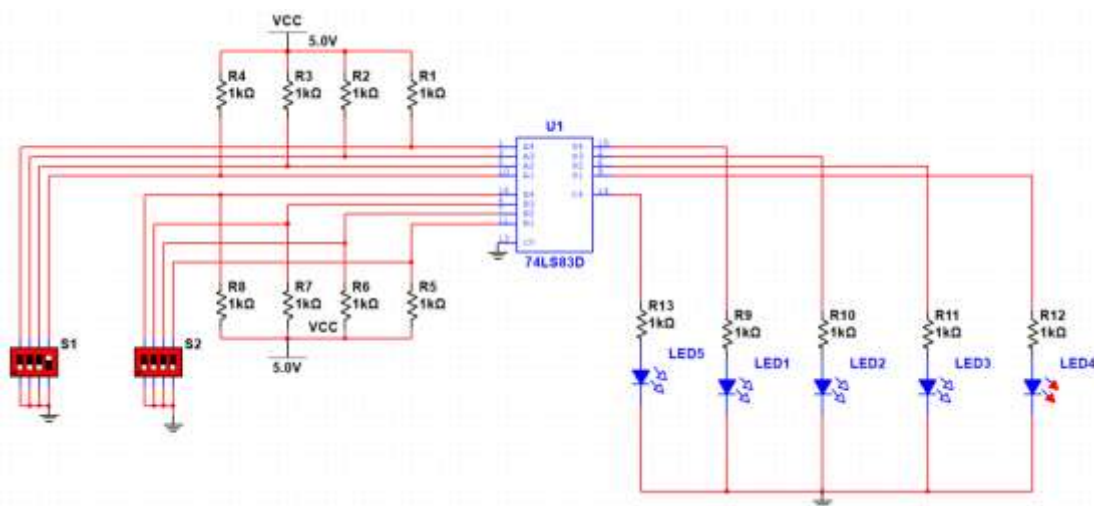


Рисунок 8.10 – Додавання 0001 та 0000

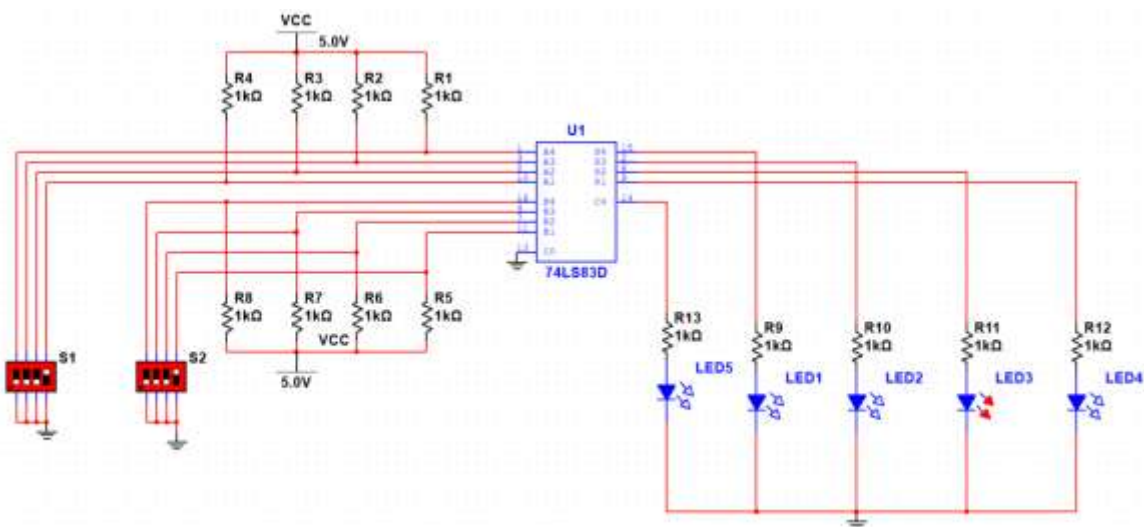


Рисунок 8.11 – Додавання 0001 та 0001

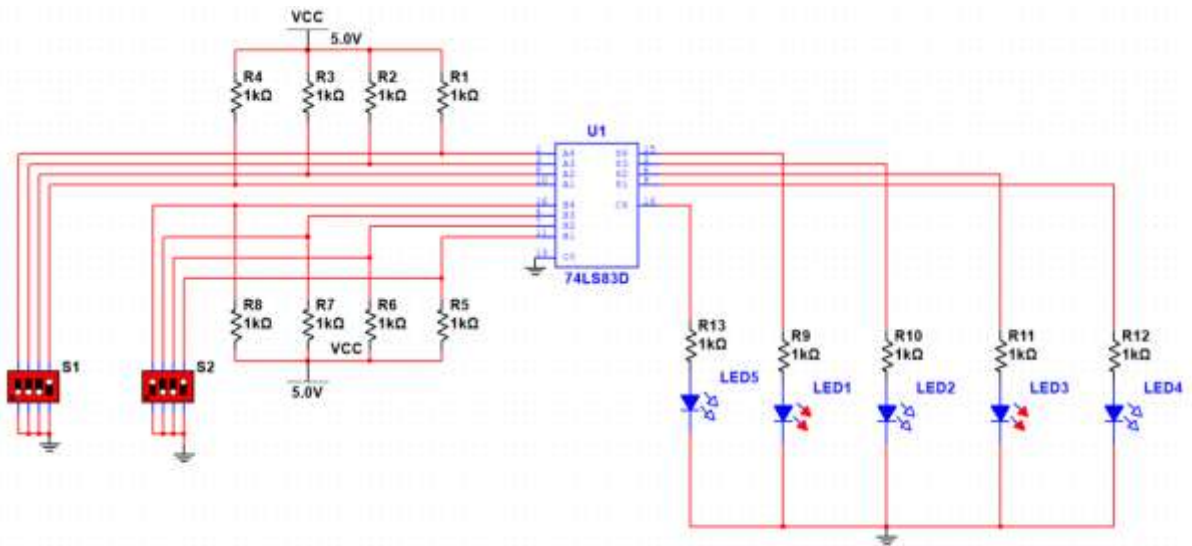


Рисунок 8.12 – Додавання 0001 та 1001

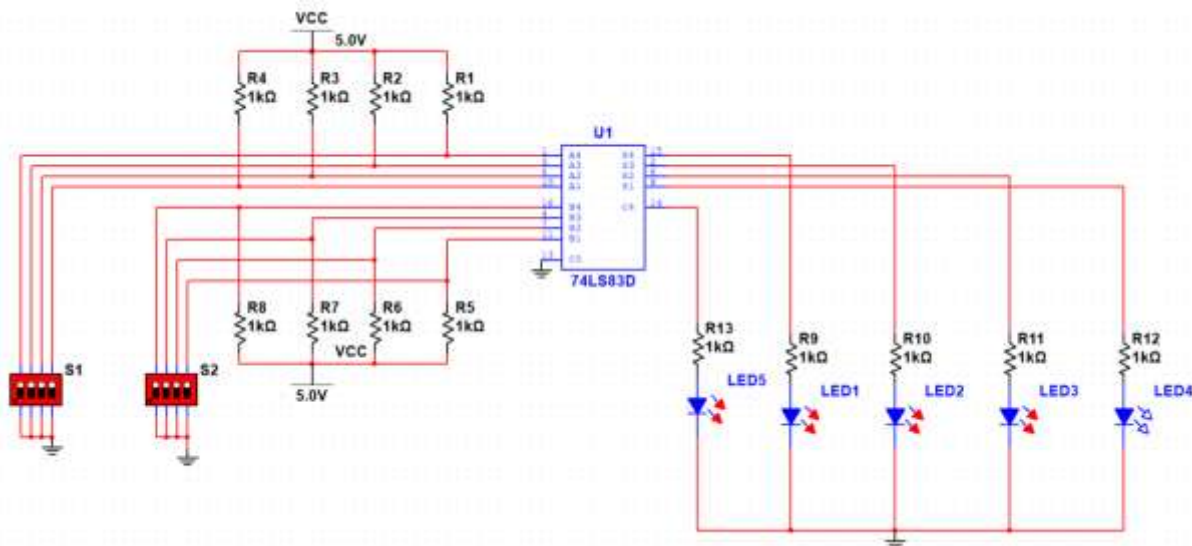


Рисунок 8.13 – Додавання 1111 та 1111

Результати досліджень рис. 8.10 – 8.13 запишемо в таблицю 8.7.

Таблиця 8.7 – Результати досліджень

Перший доданок	Другий доданок	Сума
0001	0000	00001
0001	0001	00010
0001	1001	01010
1111	1111	11110

Таким чином, результати моделювання відповідають теоретичним відомостям та нашим сподіванням.

Варіанти завдань

1. Дослідження таблиці істинності однорозрядного суматора. Підключіть один розряд мікросхеми КМ155ИМ3 як однорозрядний суматор (використовуючи вхід переносу як третій вхід) та експериментально визначте його таблицю істинності. Порівняйте отримані результати з теоретичною таблицею істинності повного суматора.

2. Вивчення роботи півсуматора. Сконфігуруйте один розряд КМ155ИМ3 як півсуматор (зафіксувавши вхід переносу на логічному нулі) та дослідіть його таблицю істинності. Порівняйте з таблицею істинності повного суматора та півсуматора.

3. Дослідження додавання двох 4-розрядних чисел без переносу. Підключіть два 4-розрядні двійкові числа до входів А та В мікросхеми КМ155ИМ3 (вхід переносу встановіть у логічний нуль). Подайте різні комбінації вхідних сигналів та зафіксуйте вихідні суми.

4. Дослідження додавання двох 4-розрядних чисел з переносом. Подайте на вхід переносу (P0) логічну одиницю та дослідіть, як це впливає на результат додавання 4-розрядних чисел. Продемонструйте випадки, коли виникає перенос у старші розряди.

5. Каскадування двох мікросхем КМ155ИМ3 для додавання 8-розрядних чисел. З'єднайте вихід переносу однієї мікросхеми з входом переносу іншої. Подайте 8-розрядні двійкові числа на входи та дослідіть процес додавання, включно з перенесенням між мікросхемами.

6. Реалізація 4-розрядного двійкового віднімача з використанням доповняльного коду та КМ155ИМ3. Додайте необхідні логічні елементи (інвертори) для формування доповняльного коду від'ємника та використайте КМ155ИМ3 для виконання операції віднімання. Протестуйте схему для різних пар чисел.

7. Дослідження часу затримки поширення сигналу суми. Використовуючи осцилограф, виміряйте час затримки між зміною вхідних сигналів (одного з доданків) та відповідною зміною сигналу суми на виході одного з розрядів КМ155ИМ3.

8. Дослідження часу затримки поширення сигналу переносу. За допомогою осцилографа виміряйте час затримки між зміною вхідних сигналів та появою сигналу переносу на виході мікросхеми КМ155ИМ3. Порівняйте з часом затримки сигналу суми.

9. Реалізація суматора ВСD-коду (двійково-десятькового коду). Додайте необхідні логічні елементи до КМ155ИМ3 для виконання додавання двох однорозрядних ВСD-чисел з корекцією результату в разі переповнення (сума більше 9).

10. Використання КМ155ИМ3 для порівняння двох 4-розрядних чисел. Розробіть логічну схему на базі суматора, яка визначає, чи є одне число більшим, меншим або дорівнює іншому.

11. Реалізація простого 4-розрядного АЛП (арифметико-логічного пристрою) з операціями додавання та віднімання. Використайте КМ155ИМ3 та додаткові логічні елементи для вибору між операціями додавання та віднімання залежно від керівного сигналу.

Зміст звіту

Звіт має містити:

1. Завдання.
2. Обґрунтування вибору мікросхеми.
3. Функціональну схему досліджуваного пристрою.
4. Результати досліджень (рисунок, фото, скрини екрана, таблиці відповідності і т. д.).
5. Висновки за результатами проведених досліджень.

Контрольні запитання

1. Що таке двійковий суматор і для чого він використовується?
2. Які основні вхідні та вихідні сигнали має однорозрядний двійковий суматор?
3. Поясніть принцип роботи однорозрядного двійкового суматора.
4. Що таке біт суми (Sum) та біт переносу (Carry-out) в однорозрядному суматорі?
5. Наведіть таблицю істинності для однорозрядного двійкового суматора.
6. Які логічні елементи використовуються для побудови однорозрядного суматора? Наведіть схему.
7. Чим відрізняється півсуматор від повного однорозрядного суматора? Для чого використовується півсуматор?
8. Як побудувати багаторозрядний паралельний двійковий суматор з однорозрядних суматорів?
9. Поясніть принцип послідовного перенесення в багаторозрядному суматорі. Який його недолік?
10. Що таке час поширення переносу в багаторозрядному суматорі з послідовним перенесенням? Від чого він залежить?
11. Які існують методи прискорення переносу в багаторозрядних суматорах? Назвіть хоча б два.
12. Поясніть принцип роботи суматора з паралельним перенесенням. Які його переваги порівняно з суматором із послідовним перенесенням?
13. Що таке генерація та поширення переносу в контексті суматорів з прискореним перенесенням?
14. Наведіть приклад використання інтегральної схеми багаторозрядного двійкового суматора. Які її основні характеристики?

15. Як двійковий суматор може бути використаний для виконання операції додавання знакових двійкових чисел?

16. Поясніть, як за допомогою двійкового суматора та доповняльного коду можна виконати операцію віднімання двійкових чисел.

17. Що таке суматор-віднімач? Як його можна реалізувати на базі двійкового суматора?

18. Які застосування двійкових суматорів в сучасній цифровій техніці та комп'ютерних системах вам відомі?

ПЕРЕЛІК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

1. Білинський Й. Й., Книш Б. П. Цифрова схемотехніка. Електронно-обчислювальні пристрої : навч. посібник. Вінниця : ВНТУ, 2021. 66 с.
2. Єсаулов С. М., Бабічева О. Ф. Аналіз, синтез і проектування цифрових систем керування : навчальний посібник. Харків : ХНУМГ ім. О. М. Бекетова, 2018. 150 с.
3. Гулий В. Д., Жуйков В. Я., Рябенський В. М. Цифрова схемотехніка : навчальний посібник для ВНЗ / за ред. В. М. Рябенського. Львів : Новий світ-2000, 2020. 736 с.
4. Матвієнко М. П. Пристрої цифрової електроніки : навчальний посібник. Суми : СУМДУ, 2021. 392 с.
5. Godse A. P., Godse Dr. D. A. Digital Electronics. Vendors: Technical Publications, 2021. 220 p.
6. Tertulien Ndjountche. Digital Electronics 1: Combinational Logic Circuits. Wiley-ISTE, 2021. 276 p.
7. Alexander Axelevitch. Digital Electronic Circuits- The Comprehensive View. World Scientific, 2020. 436 p.
8. Tokheim R. L., Hoppe P. Digital Electronics: Principles and Applications. 9th ed. New York : McGraw-Hill, 2022. 576 p.
9. Ghosh B. Basics of Digital Electronics. Boca Raton : CRC Press, 2023. 258 p.
10. Timberlake K. Digital Electronics: A Practical Approach. New York : States Academic Press, 2022. 244 p.
11. Natarajan D. Fundamentals of Digital Electronics. Cham : Springer, 2022. 248 p.
12. Ward H. H. Mastering Digital Electronics. Cham : Springer, 2023. 300 p.
13. Bolt B. Fundamentals of Digital Electronics. New York : NY Research Press, 2023. 200 p.
14. Crowe J., Hayes-Gill B. Introduction to Digital Electronics. New York : Electronic07, 2023. 320 p.
15. Robbins M. F. Ultimate Electronics: Practical Circuit Design and Analysis, 2021. 450 p.
16. Maini A. K. Digital Electronics: Principles and Applications. New Delhi : Wiley India, 2021. 600 p.
17. Jain R. P. Modern Digital Electronics. New Delhi : McGraw-Hill Education, 2021. 720 p.
18. Alam M., Alam B. Digital Logic Design. New Delhi : Cambridge University Press, 2023. 450 p.

Електронне навчальне видання

**Максим Олександрович Притула
Ігор Андрійович Дудатьєв
Ярослав Олександрович Осадчук**

Цифрові пристрої

Лабораторний практикум

Рукопис оформив *М. Притула*

Редактор *Т. Старічек*

Оригінал-макет виготовила *Т. Старічек*

Підписано до видання 20.02.2026 р.
Гарнітура Times New Roman.
Зам. № P2026-022.

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95,
м. Вінниця, 21021.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.