

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЦЕНТРАЛЬНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ
УНІВЕРСИТЕТ
ФАКУЛЬТЕТ АВТОМАТИКИ ТА ЕНЕРГЕТИКИ
КАФЕДРА АВТОМАТИЗАЦІЇ ВИРОБНИЧИХ ПРОЦЕСІВ

ЕЛЕКТРОННІ КОМПОНЕНТИ

Частина II

Методичні вказівки

до виконання лабораторних робіт

з навчальних дисциплін:

«Електронні компоненти телекомунікаційних систем»

за спеціальністю

172 «Телекомунікації та радіотехніка»

та

«Основи комп'ютерної схемотехніки»

за спеціальностями

141 «Електроенергетика, електротехніка та електромеханіка»;

151 «Автоматизація та комп'ютерно-інтегровані технології»

для студентів денної та заочної форм навчання

**Затверджено на засіданні кафедри
автоматизації виробничих процесів,
протокол № 7 від 16.12.2020 р.**

Електронні компоненти. Частина II. Методичні вказівки до виконання лабораторних робіт з навчальних дисциплін: *«Електронні компоненти телекомунікаційних систем»* за спеціальністю 172 «Телекомунікації та радіотехніка» та *«Основи комп'ютерної схемотехніки»* за спеціальностями 141 «Електроенергетика, електротехніка та електромеханіка»; 151 «Автоматизація та комп'ютерно-інтегровані технології» для студентів денної та заочної форм навчання. / Р. В. Жесан, О. П. Голик. – Кропивницький: ЦНТУ – 2021. – 54 с.

Укладачі: Жесан Р. В., кандидат технічних наук, доцент;

Голик О. П., кандидат технічних наук, доцент;

Рецензент: Каліч В. М., кандидат технічних наук, професор.

ЗМІСТ

ЛАБОРАТОРНА РОБОТА №4	
ДОСЛІДЖЕННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ.....	4
ЛАБОРАТОРНА РОБОТА №5	
ДОСЛІДЖЕННЯ ТРИГЕРІВ.....	19
ЛАБОРАТОРНА РОБОТА №6	
ПОБУДОВА ЛІЧИЛЬНИКІВ І ВИВЧЕННЯ ЇХ	
РОБОТИ.....	24
ЛАБОРАТОРНА РОБОТА №7	
ПОБУДОВА РЕГІСТРІВ, ДЕШИФРАТОРІВ,	
ШИФРАТОРІВ І ВИВЧЕННЯ ЇХ РОБОТИ.....	29
ЛАБОРАТОРНА РОБОТА №8	
ВИВЧЕННЯ РОБОТИ АРИФМЕТИЧНИХ	
СУМАТОРІВ.....	39
ЛАБОРАТОРНА РОБОТА №9	
КОНСТРУЮВАННЯ КОМБІНАЦІЙНИХ СХЕМ ТА ЇХ	
ОПТИМІЗАЦІЯ.....	44
ЛІТЕРАТУРА ДО ЛАБОРАТОРНОГО КУРСУ.....	51

ЛАБОРАТОРНА РОБОТА №4

ДОСЛІДЖЕННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ

Мета роботи: вивчити принципи роботи логічних елементів (на прикладі елементів «І», «АБО», «НЕ») за допомогою навчально-прикладної програми *Electronics Workbench (EWB)*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Логічні елементи (ЛЕ) є основними компонентами цифрових схем керування. Найпростішою формою ЛЕ є ключ – пристрій, що пропускає сигнал у замкненому стані і блокує його проходження електричним колом при розмиканні. Всі електронні ЛЕ базуються на дублюванні функції вимикача, тому для аналізу їхньої роботи зручно використовувати відповідні електричні схеми.

Логічна функція – це функція логічних змінних, яка може приймати тільки два значення: «0» та «1». Вона може бути задана у вигляді таблиці, яку називають *таблицею істинності (ТІ)* (див. **рис. 4.1**).

$X_1, X_2, X_3, \dots, X_n$	$Y=f(X_1, X_2, X_3, \dots, X_n)$
<i>Набір значень аргументів X_1, \dots, X_n</i>	<i>Значення функції Y (0 або 1)</i>

Рисунок 4.1 – Принцип побудови таблиці істинності для ЛЕ

Кількість рядків N в ТІ – це кількість можливих наборів значень аргументів. Вона дорівнює:

$$N = 2^n, \quad (4.1)$$

де n – кількість змінних.

Дуже побіжно ЛЕ вже були розглянуті у теоретичних відомостях до Лабораторної роботи №1. Не зупиняючись на математичній і елементній базі, розглянемо призначення ЛЕ, використовуваних у конструкторі, їхнє умовне графічне позначення, функціональний електричний аналог кожного з них і відповідну йому таблицю істинності.

Зупинимо свою увагу на двовходових ЛЕ ($n = 2$), використовуваних у конструкторі. При цьому позначатимемо через X_1 та X_2 вхідні логічні змінні, які приймають значення «0» або «1», а вихідний логічний вираз, що є їхньою функцією, через Y .

Елемент «І».

На виході ЛЕ «І» логічна «1» з'явиться тільки в тому випадку, якщо усі входи одночасно знаходяться в стані логічної 1. Інші (з 2^n) комбінації входів призводять до утворення на виході логічного «0».

Умовне графічне позначення ЛЕ «І» наведене на **рис. 4.2**.

ЛЕ «І» функціонує подібно послідовній електричній схемі, наведеній на **рис. 4.3**, у якій лампа засвітиться (логічний вихід дорівнюватиме «1») тільки в тому

випадку, якщо вимикачі X1 та X2 одночасно замкнуті (логічні входи дорівнюють «1»). Очевидно, що при будь-якому порушенні цієї умови (X1, X2 або обидва ключі розімкнені, чому відповідає їхній логічний стан «0») лампа буде погашеною (стан «0»).

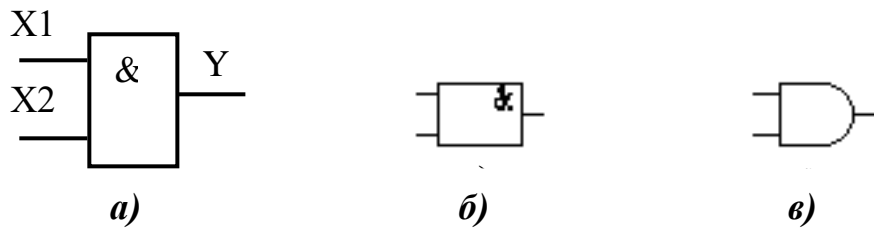


Рисунок 4.2 – Умовне графічне позначення ЛЕ «І»:
а) вітчизняний стандарт; б) європейський стандарт
(використовується в EWB); в) північноамериканський стандарт
(використовується в EWB)

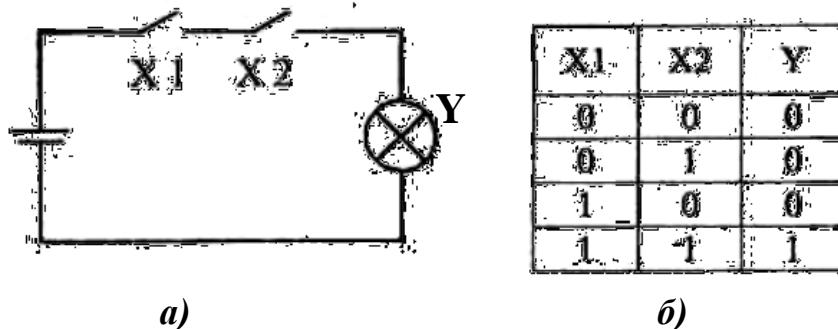


Рисунок 4.3 – Фізична інтерпретація ЛЕ «І»:
а) еквівалентна електрична схема; б) таблиця істинності

Як впливає з таблиці істинності (див. **рис. 4.3 б**)), такий елемент реалізує операцію «логічного множення» (кон'юнкцію) входніх змінних, котра може бути виражена логічним рівнянням вигляду:

$$Y = X1 \times X2 \quad \text{або} \quad Y = X1 \cap X2 \quad \text{або} \quad Y = X1 \wedge X2. \quad (4.2)$$

Елемент «АБО».

На виході ЛЕ «АБО» з'являється логічна «1», якщо хоча б один із входів знаходиться в стані логічної «1». Очевидно, що логічний «0» на виході елемента «АБО» з'явиться тільки в тому випадку, коли стани усіх входів одночасно дорівнюють логічному «0».

Умовне графічне позначення ЛЕ «АБО» наведене на **рис. 4.4**.

ЛЕ «АБО» функціонує подібно схемі, показаній нижче на **рис. 4.5**, у якій лампа горітиме (логічний вихід «1») при будь-якому сполученні замкнених (логічні входи «1») вимикачів: або X1, або X2, чи обох.

Очевидно, що тільки при розмиканні усіх вимикачів (логічні стани «0») лампа буде погашеною (логічний стан «0»).

Відповідно до таблиці істинності, елемент «АБО» реалізує операцію «логічного

додавання» (диз'юнкцію) входних змінних, котра може бути виражена логічним рівнянням вигляду:

$$Y = X1 + X2 \quad \text{або} \quad Y = X1 \cup X2 \quad \text{або} \quad Y = X1 \vee X2. \quad (4.3)$$

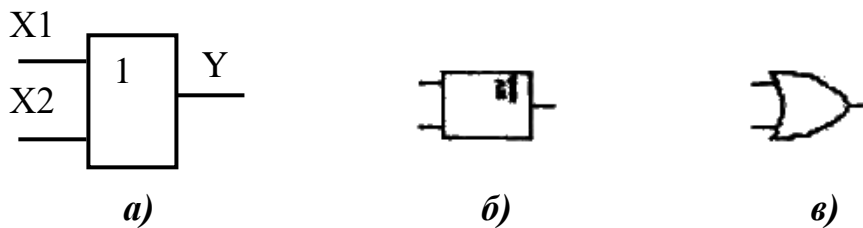


Рисунок 4.4 – Умовне графічне позначення ЛЕ «АБО»:
а) вітчизняний стандарт; б) європейський стандарт
(використовується в EWB); в) північноамериканський стандарт
(використовується в EWB)

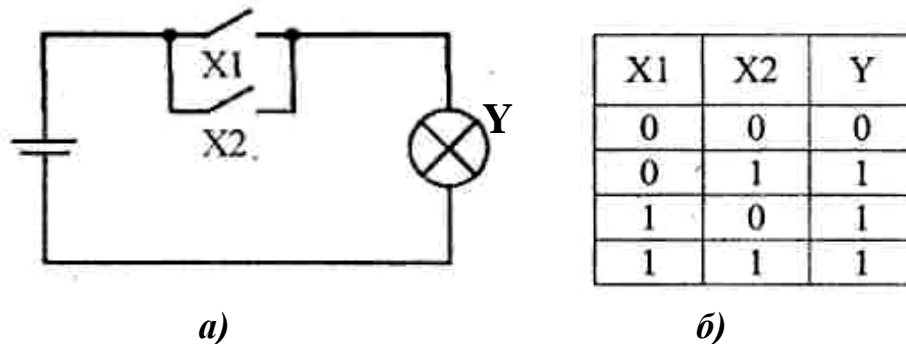


Рисунок 4.5 – Фізична інтерпретація ЛЕ «АБО»:
а) еквівалентна електрична схема; б) таблиця істинності

Елемент «НЕ» (інвертор).

На виході інвертора логічний рівень завжди протилежний входному.

Умовне графічне позначення ЛЕ «НЕ» наведене на **рис. 4.6**.

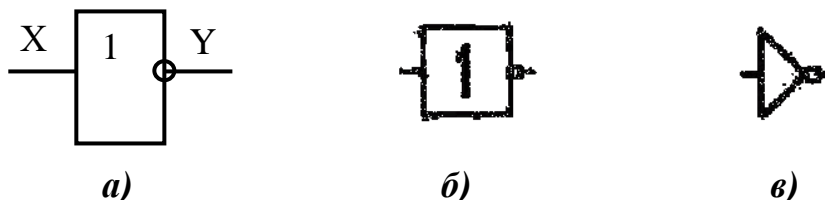


Рисунок 4.6 – Умовне графічне позначення ЛЕ «НЕ»:
а) вітчизняний стандарт; б) європейський стандарт
(використовується в EWB); в) північноамериканський стандарт
(використовується в EWB)

Інвертор реалізує операцію «логічного заперечення» входного сигналу, що, відповідно до таблиці істинності, може бути записана у вигляді:

$$Y = \overline{X1}. \quad (4.4)$$

Функціональний аналог елемента «НЕ» може бути представлений схемою, приведеною на **рис. 4.7**, у якій лампа світитиметься (логічний вихід «1») при розмиканні вимикача X1 (логічний вхід «0») і згасне (логічний «0») при його замиканні (логічна «1»).

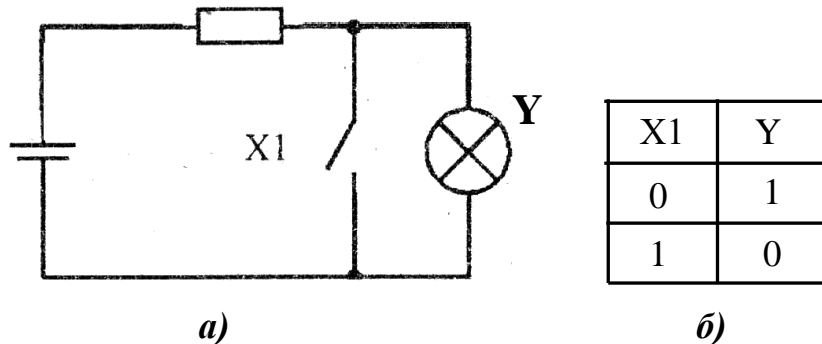


Рисунок 4.7 – Фізична інтерпретація ЛЕ «НЕ»:
а) еквівалентна електрична схема; б) таблиця істинності

Розглянуті вище елементи «І», «АБО» і «НЕ» є основою для створення більш складних логічних конструкцій, застосовуваних у цифровій техніці.

Нижче наведемо стисло решту ще не розглянутих логічних функцій (всього їх існує 16) та варіанти їх практичної реалізації – за допомогою відповідних ЛЕ чи без них.

Функція константа 0.

Математично виглядає елементарно:

$$Y = 0. \quad (4.5)$$

Технічна реалізація такої функції – з'єднання виводу Y із загальною шиною нульового потенціалу (логічним «0»).

X	Y
0	0
1	0

Рисунок 4.8 – Таблиця істинності функції константа 0

Функція константа 1.

$$Y = 1. \quad (4.6)$$

Технічна реалізація – з'єднання виводу Y із джерелом живлення (логічною «1»).

Функція повторення.

Існує в двох різновидах:

– повторення змінної X1

$$Y = f(X) = X1; \quad (4.7)$$

– повторення змінної $X2$

$$Y = f(X) = X2. \quad (4.8)$$

Технічна реалізація – з'єднання між собою виводів X та Y .

X	Y
0	1
1	1

Рисунок 4.9 – Таблиця істинності функції константа 1

X1	X2	Y
0	0	0
0	1	0
1	0	1
1	1	1

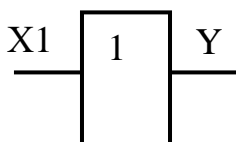
a)

X1	X2	Y
0	0	0
0	1	1
1	0	0
1	1	1

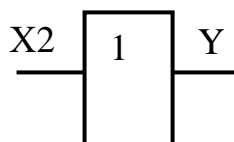
б)

Рисунок 4.10 – Таблиця істинності функції повторення:

a) змінної X1; б) змінної X2



a)



б)

Рисунок 4.11 – Умовне графічне позначення функції повторення:

a) змінної X1; б) змінної X2

Функція «АБО-НЕ» (стрілка Пірса).

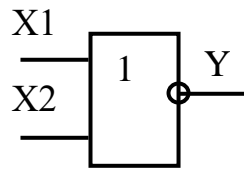
$$Y = \overline{(X1 + X2)} \quad \text{або} \quad Y = \overline{(X1 \vee X2)} \quad \text{або} \quad Y = X1 \downarrow X2. \quad (4.9)$$

Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.12**.

Функція «І-НЕ» (штрих Шеффера).

$$Y = \overline{X1 \cdot X2} \quad \text{або} \quad Y = \overline{X1 \wedge X2} \quad \text{або} \quad Y = X1 \mid X2. \quad (4.10)$$

Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.13**.

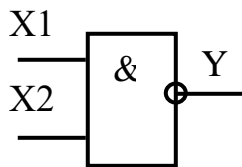


а)

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	1

б)

Рисунок 4.12 – ЛЕ «АБО-НЕ» (стрілка Пірса):
а) умовне графічне позначення; б) таблиця істинності



а)

X1	X2	Y
0	0	1
0	1	1
1	0	1
1	1	0

б)

Рисунок 4.13 – ЛЕ «І-НЕ» (штрих Шеффера):
а) умовне графічне позначення; б) таблиця істинності

Функція еквівалентності (рівнозначність, виключне АБО-НЕ).

$$Y = X1 \sim X2 \quad \text{або} \quad Y = X1 \equiv X2 \quad \text{або}$$

$$Y = \overline{X_1} \cdot X_2 \vee X_1 \cdot \overline{X_2} \vee \overline{X_1} \cdot \overline{X_2}. \quad (4.11)$$

Правило еквівалентності для будь-якої кількості вхідних змінних звучить так: значення функції Y дорівнюватиме «1» тоді і тільки тоді, коли входи X_i мають рівний стан, а значення Y дорівнюватиме «0» тоді, коли на входах X_i діє не рівний стан.

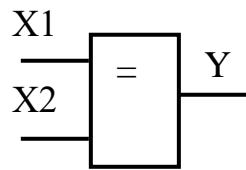
Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.14**.

Функція «Виключне АБО» (нееквівалентність, нерівнозначність, сума за модулем 2).

$$Y = X_1 \oplus X_2 \quad \text{або} \quad Y = X_1 \cdot \overline{X_2} + \overline{X_1} \cdot X_2 \quad \text{або}$$

$$Y = (X1 + X2) \cdot (\overline{X1} + \overline{X2}) \quad \text{або} \quad Y = (X1 \vee X2) \cdot (\overline{X1} \vee \overline{X2}). \quad (4.11)$$

Значення функції Y набуває значення «1», якщо вхідні змінні X_i різняться між собою, та набуває значення «0», якщо вхідні змінні X_i рівні між собою.



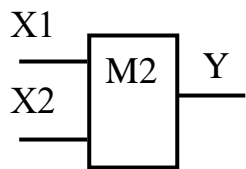
а)

X1	X2	Y
0	0	1
0	1	0
1	0	0
1	1	1

б)

Рисунок 4.14 – ЛЕ «еквівалентність» (рівнозначність, виключне АБО-НЕ):
а) умовне графічне позначення; б) таблиця істинності

Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.15**.



а)

X1	X2	Y
0	0	0
0	1	1
1	0	1
1	1	0

б)

Рисунок 4.15 – ЛЕ «Виключне АБО» (нееквівалентність, нерівнозначність, сума за модулем 2):
а) умовне графічне позначення; б) таблиця істинності

Ця функція і відповідний ЛЕ часто застосовується для контролю парності.

Функція «імплікація від X1 до X2» (інверсія декременту).

Імплікація – логічне співвідношення, яке полягає в тому, що одна річ «імплікує» іншу, тобто включає її в себе.

$$Y = X1 \rightarrow X2 \quad \text{або} \quad Y = \overline{X1} \vee X2 \quad \text{або} \quad Y = \overline{X1} + X2 \quad (4.12)$$

Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.16**.

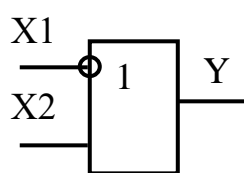
Правило для інверсії декременту можна сформулювати наступним чином.

На виході ЛЕ «імплікація від X1 до X2» буде мати значення «0» тоді і тільки тоді, коли $X2 < X1$. І на виході ЛЕ «імплікація від X1 до X2» буде встановлюватись значення «1» тоді і тільки тоді, коли $X2 \geq X1$.

Функція «імплікація від X2 до X1» (інверсія інкремента).

$$Y = X2 \rightarrow X1 \quad \text{або} \quad Y = X1 \vee \overline{X2} \quad \text{або} \quad Y = X1 + \overline{X2} \quad (4.13)$$

Відповідний ЛЕ і таблиця істинності наведені на **рис. 4.17**.

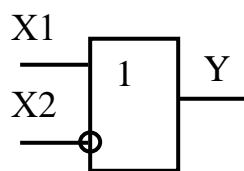


а)

X1	X2	Y
0	0	1
0	1	1
1	0	0
1	1	1

б)

Рисунок 4.16 – ЛЕ «імплікація від X1 до X2» (інверсія декремент):
а) умовне графічне позначення; б) таблиця істинності



а)

X1	X2	Y
0	0	1
0	1	0
1	0	1
1	1	1

б)

Рисунок 4.17 – ЛЕ «імплікація від X2 до X1» (інверсія інкремента):
а) умовне графічне позначення; б) таблиця істинності

Правило для інверсії інкремента можна сформулювати наступним чином.

На виході ЛЕ «імплікація від X2 до X1» буде мати значення «0» тоді і тільки тоді, коли $X2 > X1$. І на виході ЛЕ «імплікація від X1 до X2» буде встановлюватись значення «1» тоді і тільки тоді, коли $X2 \leq X1$.

Функція «заборона за X2» (декремент, інверсія імплікації від X1 до X2).

$$Y = X1 \Delta X2 \quad \text{або} \quad Y = X1 \wedge \overline{X2} \quad \text{або} \quad Y = X1 \cdot \overline{X2} \quad (4.14)$$

ЛЕ і таблиця істинності наведені на **рис. 4.18**.

Правило для функції «заборона за X2» звучить наступним чином.

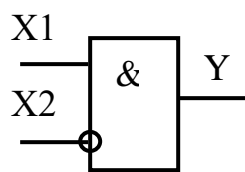
На виході ЛЕ «заборона за X2» встановлюється «1» тоді і тільки тоді, коли на вході ЛЕ: $X1 = 1$, а $X2 = 0$.

Функція «заборона за X1» (інкремент, інверсія імплікації від X2 до X1).

$$Y = X2 \Delta X1 \quad \text{або} \quad Y = \overline{X1} \wedge X2 \quad \text{або} \quad Y = \overline{X1} \cdot X2 \quad (4.15)$$

Відповідні ЛЕ та таблиця істинності наведені на **рис. 4.19**.

Правило для функції «заборона за X1» (інкремент, інверсія імплікації від X2 до X1) звучить наступним чином.



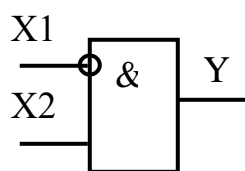
а)

X1	X2	Y
0	0	0
0	1	0
1	0	1
1	1	0

б)

Рисунок 4.18 – ЛЕ «заборона за X2» (декремент, інверсія імплікації від X1 до X2):

а) умовне графічне позначення; б) таблиця істинності



а)

X1	X2	Y
0	0	0
0	1	1
1	0	0
1	1	0

б)

Рисунок 4.19 – ЛЕ «заборона за X1» (інкремент, інверсія імплікації від X2 до X1):

а) умовне графічне позначення; б) таблиця істинності

На виході ЛЕ «заборона за X1» встановлюється «1» тоді і тільки тоді, коли на вході ЛЕ: $X1 = 0$, а $X2 = 1$.

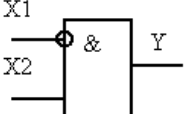
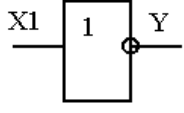
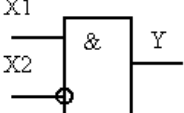
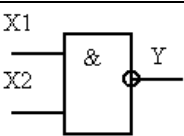
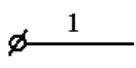
Для узагальнення і зручності сприйняття зведемо до таблиці (*табл. 4.1*) упорядковану послідовність булевих функцій та відповідних їм ЛЕ.

У *табл. 4.2* наведено зображення різних логічних елементів, представлених в *EWB 5.12*. Ті ЛЕ, яких немає у готовому вигляді, можуть бути легко одержані комбінаціями наявних, на основі відповідних таблиць істинності та виразів (4.2)-(4.15).

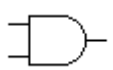
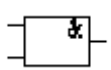

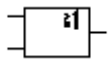


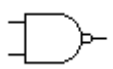
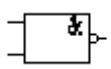

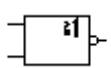

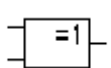

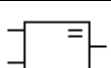

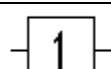
Таблиця 4.1 – Упорядкована послідовність булевих функцій та відповідних ЛЕ

Значення булевих функцій Y в залежності від аргументів $X1$ та $X2$					Назва	Умовне графічне позначення ЛЕ
$X1$	0	0	1	1		
$X2$	0	1	0	1		
1	2	3	4	5	6	7
$Y0$	0	0	0	0	Константа 0	
$Y1$	0	0	0	1	Кон'юнкція, логічне множення, логічне «І»	
$Y2$	0	0	1	0	Заборона за $X2$, декремент, інверсія імплікації від $X1$ до $X2$	
$Y3$	0	0	1	1	Повторення змінної $X1$	
$Y4$	0	1	0	0	Заборона за $X1$, інкремент, інверсія імплікації від $X2$ до $X1$	
$Y5$	0	1	0	1	Повторення змінної $X2$	
$Y6$	0	1	1	0	Нееквівалентність, нерівнозначність, сума за модулем 2, «Виключне АБО»,	
$Y7$	0	1	1	1	Диз'юнкція, логічне додавання, логічне «АБО»	
$Y8$	1	0	0	0	Заперечення диз'юнкції, стрілка Пірса, логічне «АБО-НЕ»	
$Y9$	1	0	0	1	Еквівалентність, рівнозначність, «Виключне АБО-НЕ»	
$Y10$	1	0	1	0	Інверсія $X2$, заперечення $X2$, логічне «НЕ» $X2$	

Продовження таблиці 4.1

1	2	3	4	5	6	7
Y11	1	0	1	1	Імплікація від X2 до X1, інверсія інкремента	
Y12	1	1	0	0	Інверсія X1, заперечення X1, логічне «НЕ» X1	
Y13	1	1	0	1	Імплікація від X1 до X2, інверсія декремента	
Y14	1	1	1	0	Заперечення кон'юнкції, штрих Шеффера, логічне «І-НЕ»	
Y15	1	1	1	1	Константа 1	

Таблиця 4.2 – Зовнішній вигляд логічних елементів в EWB 5.12

Зображення		Назва ЛЕ (функція булевої алгебри)
північноамериканський стандарт	європейський стандарт	
		«І»
		«АБО»
		«НЕ»
		«І-НЕ»
		«АБО-НЕ»
		«Виключне АБО»
		«Виключне АБО-НЕ»
		«Буфер»

В *EWB* при дослідженні логічних елементів використовуються такі інструменти: *генератор слів (Word Generator)* та *логічний аналізатор (Logic Analyzer)*.

Генератор слів (див. *рис. 4.20*) використовується для подачі послідовності бітів у схему. Його ліва частина вміщує 16 рядків (слів) по 8 біт кожний.

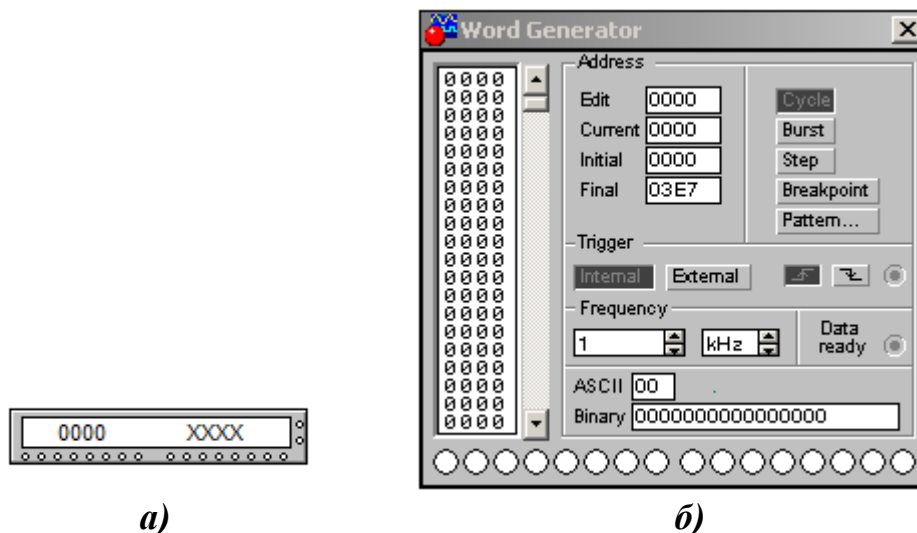


Рисунок 4.20 – Генератор слів (Word Generator) з EWB 5.12:
а) у компактному вигляді; б) у «розгорнутому» вигляді

Коли генератор активовано, слова одне за одним надходять до схеми. Крім того, пристрій має вихід внутрішнього генератора часових імпульсів, який використовується для синхронізації. Генератор призначений для генерації 16-розрядних двійкових слів. Кодові комбінації необхідно задавати в шістнадцятковому коді. Кожна комбінація вводиться за допомогою клавіатури, номер комірки (яку редагують) фіксується у вікні **Edit** блоку **Address**. Всього таких комірок, а, отже, і комбінацій – 2048. В процесі роботи генератора в блоці **Address** існують такі комірки:

- Current** – номер поточної комірки;
- Initial** – комірки ініціалізації або початку роботи;
- Final** – кінцева комірка.

На 16 виходів (у нижній частині генератора) надходять кодові комбінації у текстовому (*ASCII*) та двійковому (*Binary*) кодах.

Сформовані слова надходять на 16 розташованих у нижній частині приладу клем-індикаторів:

- в покроковому (при натисканні кнопки **Step**), циклічному (**Cycle**) або з обраного слова до кінця (**Burst**), при заданій частоті посилянь (**Frequency**);
- при внутрішньому (при натисканні кнопки **Internal**) або зовнішньому запуску (**External**), згідно з готовністю даних (клема **Data ready**), поряд розташована клема для підключення каналу синхронізації.

До органів керування належить також кнопка **Break point** – припинити роботу генератора у відповідній комірці.

Логічний аналізатор, зовнішній вигляд якого наведено на *рис. 4.21*, призначено для відображення на екрані монітору 16-розрядних кодових послідовностей

одночасно в 16 точках схеми, а також у вигляді двійкових чисел на вхідних клемах-індикаторах і, в основному полі, – часових діаграм сигналів, поданих на 16 входів.

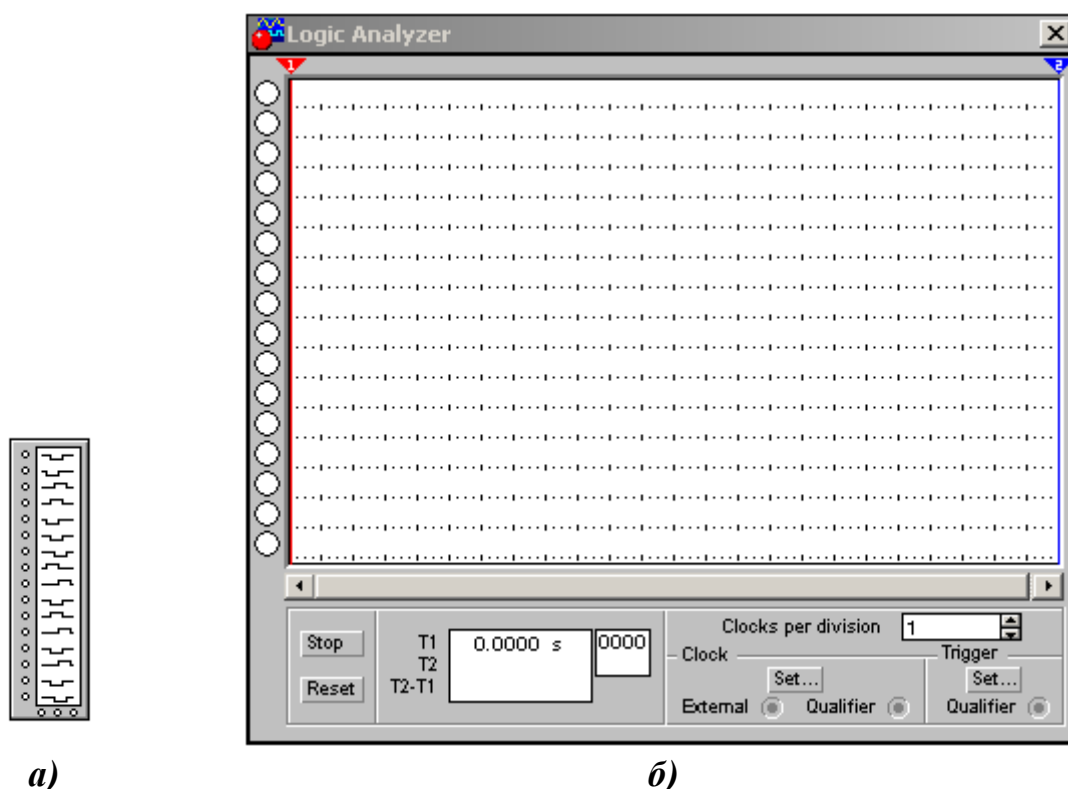


Рисунок 4.21 – Логічний аналізатор (Logic Analyzer) з EWB 5.12:
а) у компактному вигляді; б) у «розгорнутому» вигляді

У блоці **Clock** є клеми як для звичайного зовнішнього (**External**), так і вибіркового (**Qualifier**) джерела сигналів, параметри яких можуть бути встановлені за допомогою меню **Set**. У вікні **Clock qualifier** можна встановити значення логічного сигналу (0, 1 або X), при якому виконується запуск аналізатора.

На **рис. 4.22** наводиться приклад дослідження ЛЕ «НЕ» в програмному пакеті **EWB 5.12**.

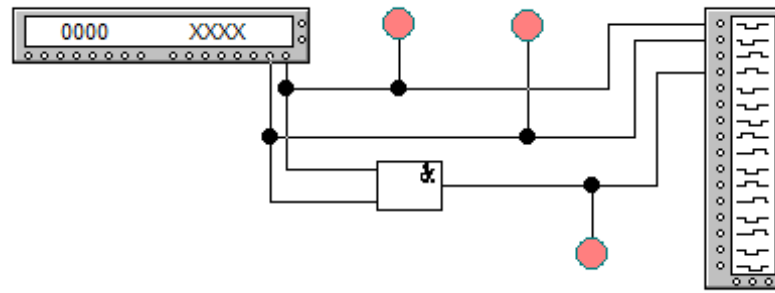
Аналогічним чином можна одночасно дослідити роботу декількох ЛЕ, зібравши схему, подібну до зображеної на **рис. 4.23**. Для цього необхідно спочатку розмістити на робочому полі програми необхідні ЛЕ, а потім з'єднати їх між собою паралельно, таким чином, щоб входи кожного логічного елемента були під'єднані до клем генератора слів. У той же час вихід кожного ЛЕ слід підключити до окремої клеми логічного аналізатора (див. **рис. 4.23**).

При налаштуванні параметрів генератора слів та логічного аналізатора слід звернути увагу на те, що вони повинні працювати синхронно. Тому частоту в генераторі слів та логічному аналізатора необхідно встановити однакову, наприклад, 10 МГц.

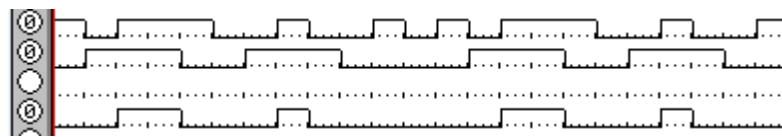
ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. Дослідити самостійно обрані з бази ЛЕ (не менше 5 шт.). Для цього

необхідно, аналогічно схемі **рис. 4.23**, паралельно підключити входи ЛЕ до генератора слів, а до виходу елемента приєднати по чергово окремі клери логічного аналізатора. Додатково, для наочності, на кожну з ліній встановлюється світлодіодний індикатор («червона лампочка»).



а)



б)

Рисунок 4.22 – Дослідження ЛЕ «НЕ» в програмному пакеті EWB 5.12:
а) схема дослідження; б) часова діаграма (фрагмент основного поля логічного аналізатора)

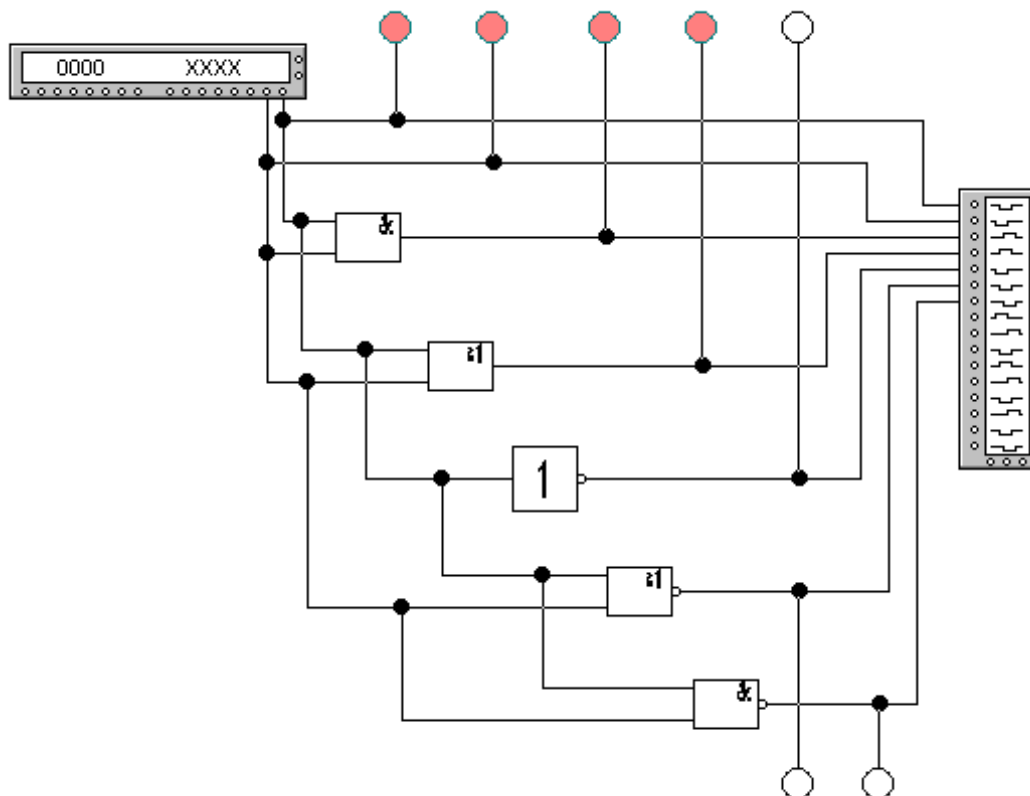


Рисунок 4.23 – Одночасне дослідження п'ятих ЛЕ в програмному пакеті EWB 5.12

Для побудови й запуску досліджуваної схеми:

- а) перетягнути мишею необхідні елементи схеми з бібліотек компонентів;
- б) оптимально розмістити їх у робочому просторі;
- в) з'єднати компоненти провідниками;
- г) задати параметри або моделі компонентів;
- д) під'єднати генератор слів та світлодіодні індикатори, а також логічний аналізатор;
- е) увімкнути схему.

3. Використовуючи засоби і можливості програмного пакету *EWB*, відлагодити роботу схеми.

4. За результатами дослідження ЛЕ побудувати для них часові діаграми і таблиці істинності. При цьому, починати дослідження кожного наступного ЛЕ слід тільки після перевірки викладачем правильності одержаних результатів для попереднього.

СКЛАД ЗВІТУ

1. Назва роботи.
2. Мета.
3. Умовні графічні зображення ЛЕ, досліджуваних в ході роботи (у відповідності до ГОСТ 2.708-81 ЄСКД).
4. Результати досліджень ЛЕ у вигляді часових діаграм та таблиць істинності.
5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛАБОРАТОРНА РОБОТА №5

ДОСЛІДЖЕННЯ ТРИГЕРІВ

Мета роботи: вивчити принципи роботи та різновиди тригерів (на прикладі *RS*-тригерів) на логічних елементах за допомогою навчально-прикладної програми *EWB*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Тригером (від англійського *trigger* – засувка, спусковий гачок) – називають пристрій із зворотними зв'язками, який має 2 стійких стани («0» та «1») і здатний стрибком переходити з одного стану до іншого під дією зовнішнього керуючого сигналу. В загальному випадку тригер є запам'ятовуючим елементом. Для переходу тригера з одного стійкого стану в інший треба, щоб вихідний сигнал переважував деяке порогове значення.

Основні галузі застосування тригера:

- запам'ятовувальна комірка в пристроях електронної пам'яті цифрової техніки;
- елемент ділення на 2 в імпульсних лічильниках і дільниках частоти;
- пристрій для розширення (збільшення тривалості) імпульсів;
- пристрій, який поновлює форму прямокутного імпульсу.

Усі різновиди тригерів являють собою елементарний автомат – автомат Мура (рис. 5.1), який складається з елемента пам'яті (ЕП) та схеми керування (СхК), яка утворює вхідну логіку.

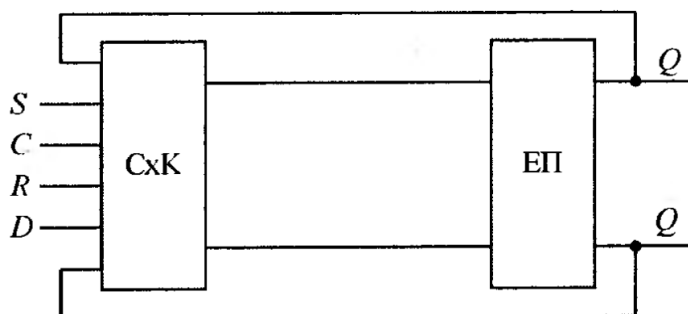


Рисунок 5.1 – Структура тригера у вигляді автомата Мура:
ЕП – елемент пам'яті; СхК – схема керування

Зміна стану тригера (його перемикання) забезпечується зовнішніми сигналами й сигналами зворотного зв'язку з виходу тригера на вхід СхК. Зазвичай зовнішні сигнали і відповідні входи тригера позначають латинськими літерами *R*, *S*, *T*, *C*, *V* та ін. Оскільки функціональні властивості тригерів визначаються їхньою СхК, то назви основних входів переносяться на всю схему. У найпростіших схемах тригерів окрема СхК може бути відсутньою.

При цьому один вихідний сигнал *Q* (прямий вихід) має значення, тотожне стану тригера в момент часу, що розглядається, а другий вихідний сигнал \bar{Q} (інверсний вихід) має значення, протилежне значенню першого вихідного сигналу. Тому

принцип функціонування тригерного пристрою можна описати всіма способами, прийнятими для опису таких автоматів. На практиці для опису принципу функціонування тригерів широко використовують таблиці переходів і часові діаграми. При цьому таблиця переходів одночасно є і таблицею виходів для прямого виходу Q , оскільки новий стан тригера та його вихідний сигнал на прямому виході за значенням тотожні.

Прийнято позначати старий стан тригера через Q_n або $Q(t)$, а новий стан – через Q_{n+1} або $Q(t+1)$ відповідно.

Тригери класифікуються за такими основними ознаками:

- 1) літерними позначенням входів або логікою функціонування (RS -тригери, JK -тригери, T -тригери, D -тригери та ін.);
- 2) способом синхронізації запису інформації (асинхронні та синхронізовані);
- 3) кількістю тактів синхронізації (однотактні, двотактні і тритактні);
- 4) моментом реакції на тактовий сигнал (статичні та динамічні);
- 5) кількістю ступенів (одноступеневі або двоступеневі);
- 6) складом логічних елементів, на яких побудовані.

Асинхронний RS-тригер.

Асинхронний (несинхронізований) RS -тригер на інтегральних елементах «АБО-НЕ» показаний на **рис. 5.2**.

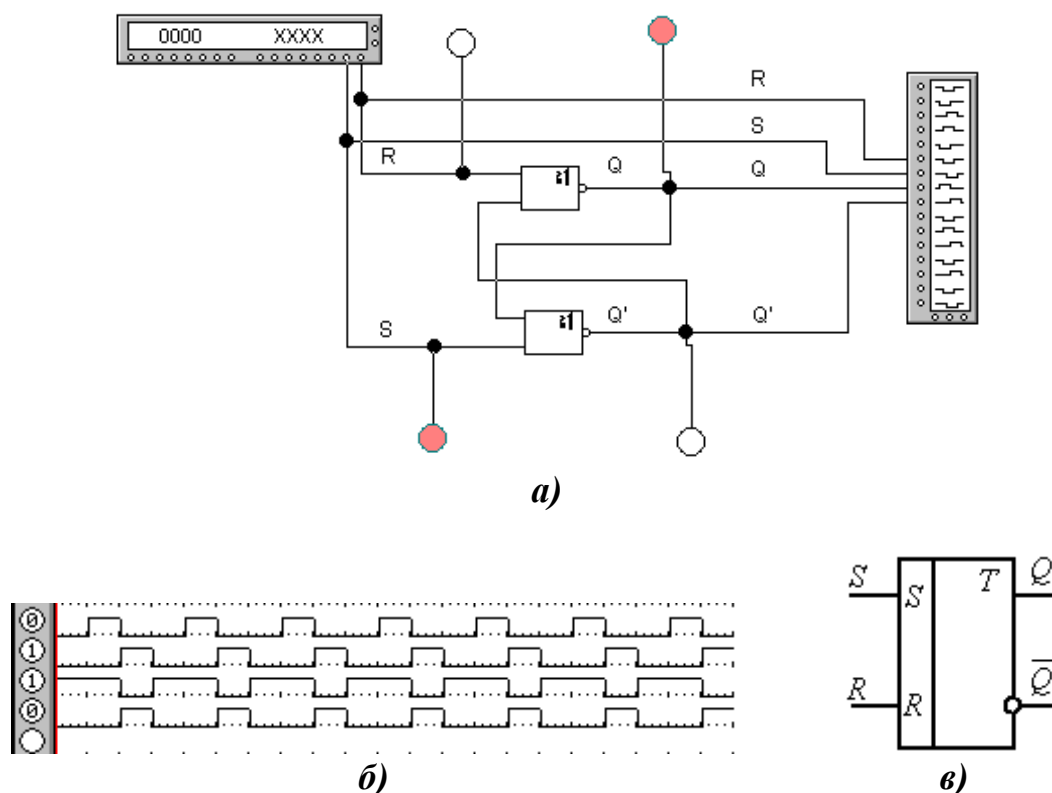


Рисунок 5.2 – Асинхронний RS -тригер:

а) схема дослідження в програмному пакеті EWB 5.12; б) часова діаграма; в) умовне графічне позначення

Тригер складений із двох комбінаційних схем «АБО-НЕ», з'єднаних таким чином, що виникають зворотні зв'язки, завдяки яким у стійкому стані вихідний

транзистор однієї схеми «АБО-НЕ» закритий, а іншої – відкритий.

При $R = 1$ і $S = 0$ тригер встановлюється в нульовий стан ($Q = 0$); при $R = 0$ і $S = 1$ він встановлюється в одиничний стан ($Q = 1$); при $R = S = 0$ тригер зберігає стан, в якому він знаходився до моменту надходження на його входи нульових сигналів.

При $R = S = 1$ на прямому і інверсному виходах встановлюється нульовий сигнал. Тригерне коло перетворюється в два незалежних інвертори, і, при переході до збереження ($R = S = 0$), тригер може встановлюватися в непередбачуваний стан. Тому така комбінація входних сигналів вважається забороненою.

Однотактний синхронізований RS-тригер.

На **рис. 5.3** наведена схема однотактного синхронізованого RS-тригера на елементах «І-НЕ».

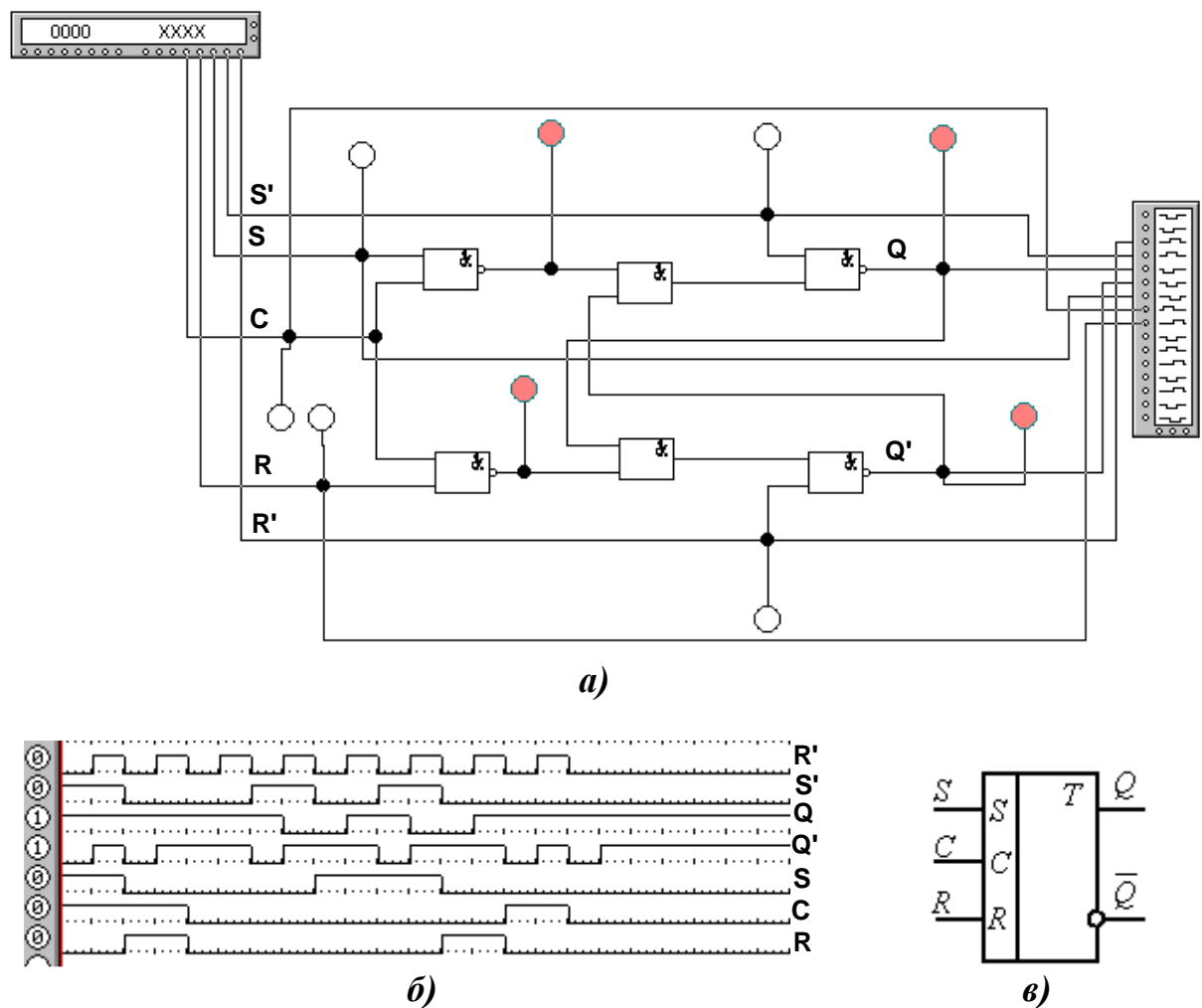


Рисунок 5.3 – Однотактний синхронізований RS-тригер:
а) схема дослідження в програмному пакеті EWB 5.12; б) часова діаграма; в) умовне графічне позначення

Тут перші два елементи утворюють схему входної логіки асинхронного RS-тригера, побудованого на другій парі елементів. Такі RS-тригери мають два інформаційних входи R та S і вхід синхронізації C . Крім того, тригер може мати несинхронізовані входи R' та S' . Функціонування тригера здійснюється або під впливом несинхронізованих входів, при $C = 0$, або під впливом синхронізованих

входів, при $R' = T$ і $S' = 1$.

Двотактний RS-тригер.

Найпростіша схема двовходового двотактного RS-тригера показана на *рис. 5.4*.

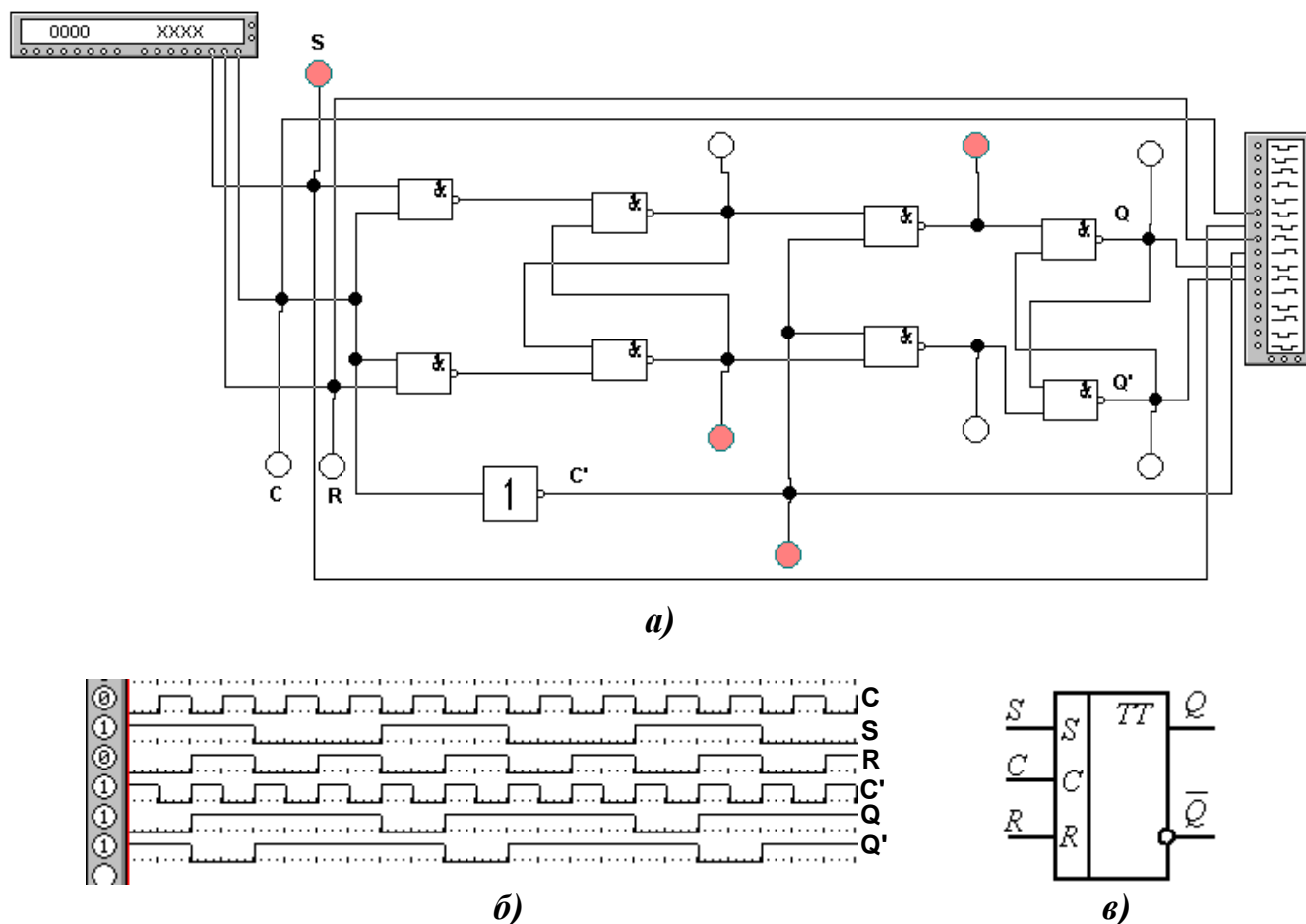


Рисунок 5.4 – Двотактний RS-тригер:
а) схема дослідження в програмному пакеті EWB 5.12; б) часова
діаграма; в) умовне графічне позначення

Вона складається з двох однотактних RS-тригерів і інвертора в колі синхронізації. При надходженні на вхід C RS-тригера сигналу $C = 1$ вхідна інформація заноситься в перший однотактний синхронізований RS-тригер, а другий при цьому буде зберігати інформацію, яка відноситься до попереднього періоду представлення.

Після закінчення дії сигналу синхронізації, коли $C = 0$, а $C' = 1$, перший RS-тригер перейде в режим збереження, а другий прийме той же стан, що і перший. В результаті до наступного такту на виході двотактного RS-тригера з'явиться сигнал нового стану. Двотактний тригер змінює свій стан тільки після закінчення дії сигналу синхронізації $C = 1$ (перехід у режим збереження інформації).

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. Побудувати схеми асинхронного RS-тригера, синхронізованого RS-тригера,

двотактного *RS*-тригера. Після складання кожну схему підключити до приладів генератора слів і логічного аналізатора, причому на входи логічного аналізатора необхідно подавати не тільки вихідні сигнали досліджуваної схеми, але і використовувані при цьому вхідні сигнали генератора слів.

3. Використовуючи засоби і можливості програми *EWB*, відлагодити роботу схеми.

Для побудови й запуску схеми:

- а) перетягнути мишею необхідні елементи схеми з бібліотек компонентів;
- б) оптимально розмістити їх у робочому просторі;
- в) з'єднати компоненти провідниками;
- г) задати параметри або моделі компонентів;
- д) під'єднати генератор слів та світлодіодні індикатори, а також логічний аналізатор;
- е) увімкнути схему.

4. За результатами досліджень побудувати часові діаграми і таблиці істинності. При цьому, починати дослідження кожної наступної схеми слід тільки після перевірки викладачем правильності одержаних результатів для попередньої.

СКЛАД ЗВІТУ

1. Назва роботи.
2. Мета.
3. Схеми досліджуваних тригерів (асинхронного *RS*-тригера, синхронізованого *RS*-тригера і двотактного *RS*-тригера) та відповідні їм умовні графічні зображення тригерів (у відповідності до ГОСТ 2.708-81 ЄСКД).
4. Результати досліджень у вигляді часових діаграм та таблиць істинності.
5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛАБОРАТОРНА РОБОТА №6

ПОБУДОВА ЛІЧИЛЬНИКІВ І ВИВЧЕННЯ ЇХ РОБОТИ

Мета роботи: вивчити принципи роботи різноманітних лічильників за допомогою навчально-прикладної програми EWB.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Лічильником називається типовий функціональний вузол цифрової техніки, призначений для лічби входних сигналів (імпульсів). Лічильник являє собою зв'язаний ланцюг T -тригерів, які утворюють пам'ять із заданим числом сталих станів (рис. 6.1).

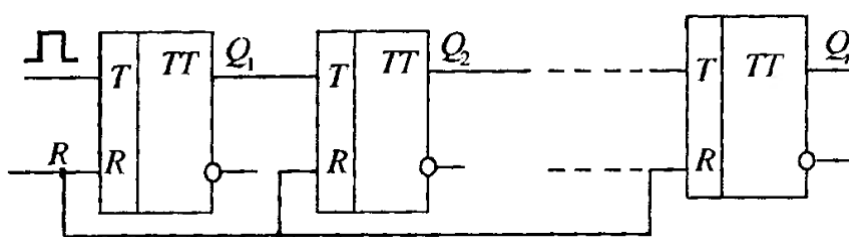


Рисунок 6.1 – Логічна структура лічильника

Розрядність лічильника n дорівнює кількості T -тригерів. Кожний вхідний імпульс змінює стан лічильника, який зберігається до надходження наступного сигналу. Значення виходів тригерів лічильника Q_n, Q_{n-1}, \dots, Q_1 відображають результат лічби в прийнятій системі числення.

Список мікрооперацій лічильника вміщує попереднє встановлення в початковий стан, інкремент або декремент слова, яке зберігається, видачу слів паралельним кодом та ін.

Вхідні імпульси можуть надходити на лічильник як періодично, так і довільно розподіленими у часі. Амплітуда і тривалість лічильних імпульсів мають задовольняти технічні вимоги для серій мікросхем, які використовуються.

Лічильник є одним з основних функціональних вузлів комп'ютера, а також різних цифрових керуючих та інформаційно-вимірювальних систем. Основне застосування лічильників:

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

Лічильники можуть бути побудовані на основі лічильних тригерів зі спеціальними міжрозрядними зв'язками, на основі регістрів зсуву (кільцеві лічильники) і на основі багатостійких елементів.

Кількість дозволених станів лічильників називають його *періодом*, *модулем* або *коефіцієнтом перерахунку* K .

Основними часовими характеристиками лічильників є:

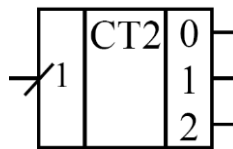
f – максимальна частота надходження рахункових сигналів;

t – час переходу лічильника з одного стану в інший.

Прийнято класифікувати лічильники за наступними основними ознаками:

- 1) способом кодування (позиційні та непозиційні);
- 2) модулем лічби (двійкові, десяткові, з довільним постійним або змінним [програмованим] модулем);
- 3) за характером виконуваних операцій, або напрямком лічби (підсумовуючі, віднімаючі та реверсивні);
- 4) способом організації міжрозрядних зв'язків (з послідовним, наскрізним, паралельним або груповим [комбінованим] переносом);
- 5) типом використовуваних тригерів (T –, JK –, D – у лічильному режимі);
- 6) розрядністю n (дорівнює кількості тригерів, як відзначалося вище).

Логічна функція лічильника позначається літерами СТ (від англійського *counter* – лічильник). За необхідності, праворуч від них вказують у вигляді числа коефіцієнт перерахунку K . На **рис. 6.2** наведено приклад умовного графічного позначення 3-розрядного двійкового лічильника.



**Рисунок 6.2 – 3-розрядний двійковий лічильник.
Умовне графічне позначення**

У лічильниках з *послідовним переносом* перенос (позиція) у сусідній старший розряд формується тільки після перемикаання тригера в попередньому розряді, тобто тригери перемикаються не одночасно. При проектуванні таких лічильників виникають труднощі, пов'язані з необхідністю аналізу не тільки логічного рівня сигналів, що формуються в схемі, але і моментів зміни рівнів сигналів.

Кола *наскрізного переносу* організовуються таким чином, щоб функція переносу i -го розряду лічильника була аргументом функції переносу $(i + 1)$ -го розряду. В цьому випадку сигнали переносів для кожного розряду формуються по черзі, починаючи з молодших розрядів лічильника. Такі лічильники вимагають меншої кількості входів логічних елементів для організації кіл переносу, але поступаються лічильникам з паралельним переносом у швидкодії.

В лічильниках із *паралельним переносом* аргументами функцій переносів для кожного розряду є тільки сигнали на виходах тригерів відповідних розрядів. Переноси для всіх розрядів лічильника формуються одночасно (за умови, що всі логічні елементи в схемі мають однаковий час перемикаання).

В зв'язку з обмеженнями на побудову лічильників із паралельним переносом великої розрядності широке розповсюдження отримали лічильники з *груповим переносом* (послідовно-паралельним переносом). Розряди таких лічильників розбиваються на групи, всередині яких організується принцип паралельного переносу. Групи з'єднують послідовно з використанням кон'юнкторів, які формують перенос в наступну групу при одиничному стані всіх попередніх

тригерів. При одиничному стані всіх тригерів групи, надходження наступного вхідного сигналу створить перенос з цієї групи. Ця ситуація підготовляє кон'юнктор до прямого пропускання вхідного сигналу на наступну групу.

Несинхронізований 4-розрядний підсумовуючий лічильник з послідовним переносом.

На **рис. 6.3** наведено схему несинхронізованого 4-розрядного підсумовуючого лічильника з послідовним переносом.

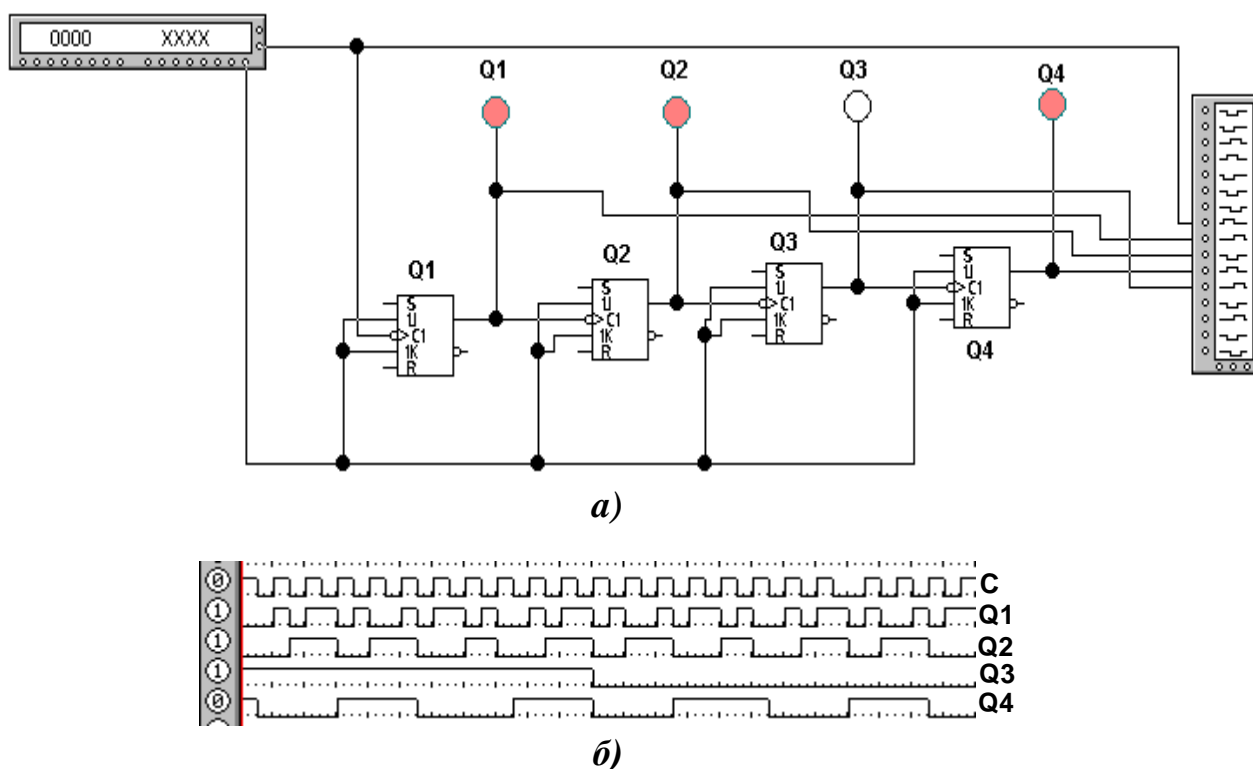


Рисунок 6.3 – Несинхронізований 4-розрядний підсумовуючий лічильник з послідовним переносом:

а) схема дослідження в програмному пакеті EWB 5.12;
б) часова діаграма

На входи J і K JK-тригерів подаються сигнали «1». Вихід кожного попереднього тригера під'єднаний до входу синхронізації наступного. Із спадом одиничного вхідного сигналу змінюється стан тригера першого розряду лічильника на протилежний. В подальших розрядах стан змінюється із спадом вихідного сигналу попереднього тригера.

Несинхронізований 4-розрядний підсумовуючий лічильник з паралельним переносом.

На **рис. 6.4** наведено схему несинхронізованого 4-розрядного підсумовуючого лічильника з паралельним переносом.

Особливістю цієї схеми є те, що з виходів Q всіх попередніх тригерів сигнали подаються на входи J і K наступного тригера. Зі збільшенням порядкового номера тригера збільшується кількість входів в ЛЕ «І» JK-тригера, але тривалість

перехідного процесу в такому лічильнику дорівнює тривалості перемикання одного розряду.

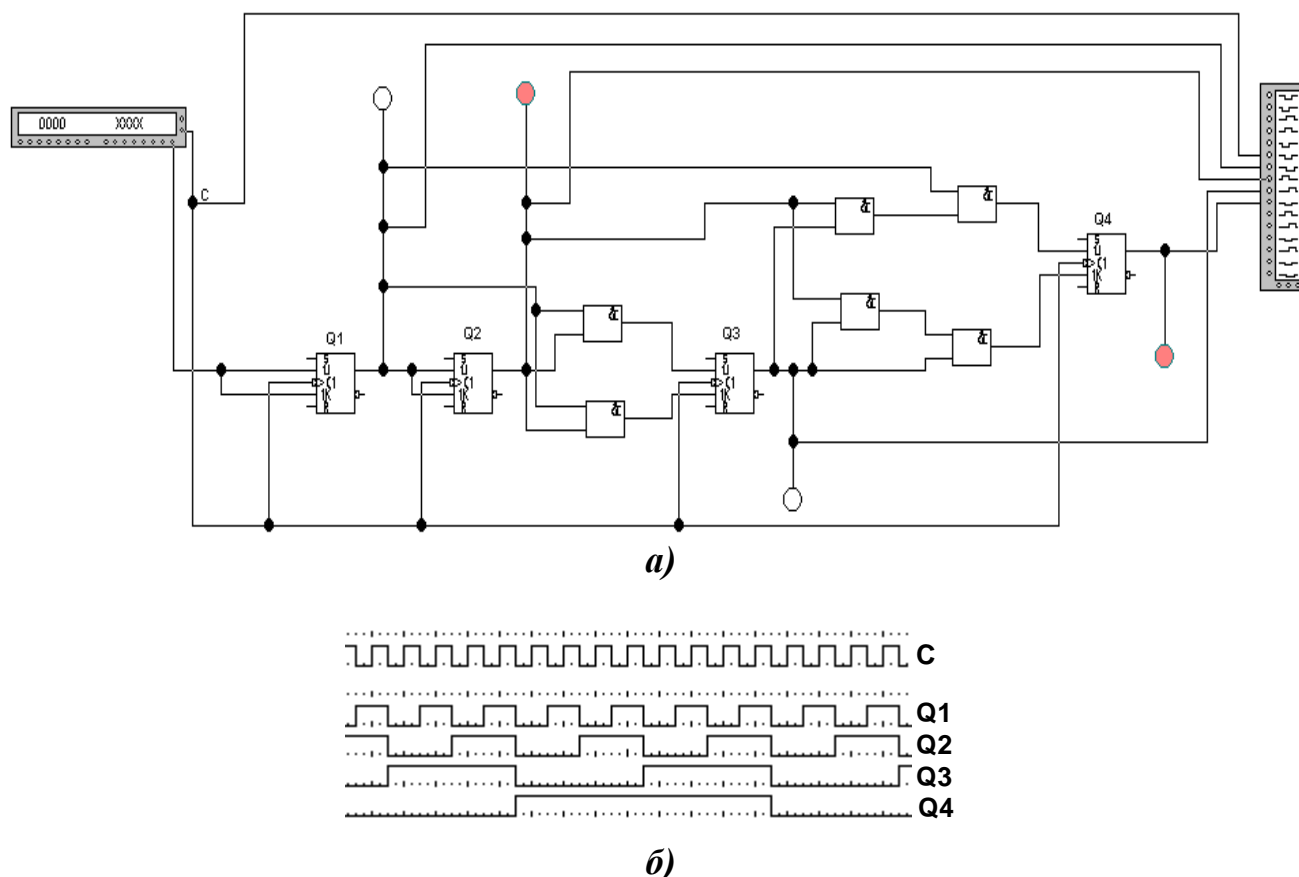


Рисунок 6.4 – Несинхронізований 4-розрядний підсумовуючий лічильник з паралельним переносом:

а) схема дослідження в програмному пакеті EWB 5.12;

б) часова діаграма

Для зменшення розмірів схеми на екрані вся вона або її частина можуть бути згорнуті у підсистему («мікросхему») і позначатись в подальшому одним елементом. Для виконання цієї дії необхідно скористатись підпунктом *Create Subcircuit* пункту меню **Circuit**. У віконці, що відкриється необхідно задати назву новостворюваного елемента і обрати кнопку *Replace in Circuit*.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. Побудувати схеми лічильників з послідовним і паралельним переносом (див. **рис. 6.3** та **рис. 6.4**). Після складання кожену схему підключити до приладів генератора слів і логічного аналізатора. На синхронізуючі входи лічильників рекомендується подавати синхроімпульси з виходу генератора слів, розташованого в його правому нижньому куті. На входи логічного аналізатора необхідно подавати не тільки вихідні сигнали досліджуваної схеми, але і використовувані при цьому вхідні сигнали генератора слів.
3. Використовуючи засоби і можливості програми *EWB*, відлагодити роботу

схеми.

Для побудови й запуску схеми:

- а) перетягнути мишею необхідні елементи схеми з бібліотек компонентів;
- б) оптимально розмістити їх у робочому просторі;
- в) з'єднати компоненти провідниками;
- г) задати параметри або моделі компонентів;
- д) під'єднати генератор слів та світлодіодні індикатори, а також логічний аналізатор;
- е) задати таблицю вхідних сигналів на генераторі слів;
- ж) увімкнути подачу вхідного сигналу за допомогою органів керування генератора слів або вимикача розташованого в правому верхньому кутку екрана.

4. За результатами досліджень побудувати часові діаграми. При цьому, починати дослідження кожної наступної схеми слід тільки після перевірки викладачем правильності одержаних результатів для попередньої.

СКЛАД ЗВІТУ

1. Назва роботи.
2. Мета.
3. Схеми досліджуваних лічильників (несинхронізованого 4-розрядного підсумовуючого лічильника з послідовним переносом та несинхронізованого 4-розрядного підсумовуючого лічильника з паралельним переносом).
4. Результати досліджень у вигляді часових діаграм.
5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛАБОРАТОРНА РОБОТА №7

ПОБУДОВА РЕГІСТРІВ, ДЕШИФРАТОРІВ, ШИФРАТОРІВ І ВИВЧЕННЯ ЇХ РОБОТИ

Мета роботи: вивчити принципи роботи регістру зсуву, дешифратора та шифратора за допомогою навчальної прикладної програми EWB.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Регістри.

Регістром називається типовий функціональний вузол цифрової техніки, призначений для запам'ятовування n -розрядного двійкового слова, а також для виконання над словом певних логічних перетворень.

Регістр містить регулярний набір однотипних тригерів, у кожному з яких зберігається значення одного двійкового розряду машинного слова (*рис. 7.1*).

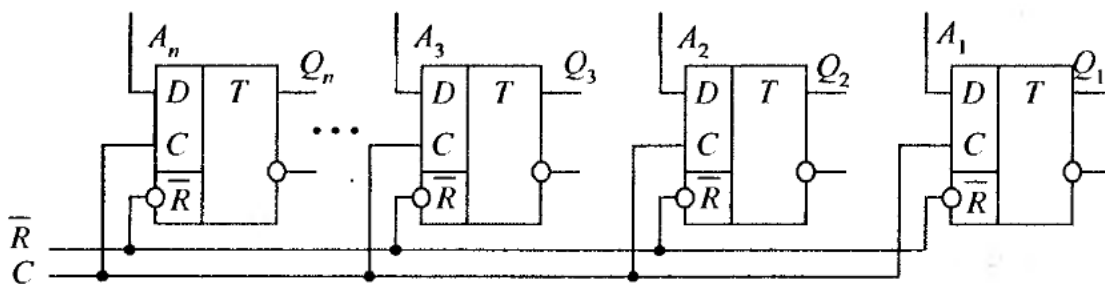


Рисунок 7.1 – Схема регістра на D-тригерах

Мікрооперація – це елементарна машинна дія, в результаті якої змінюються значення слова чи здійснюється його пересилання.

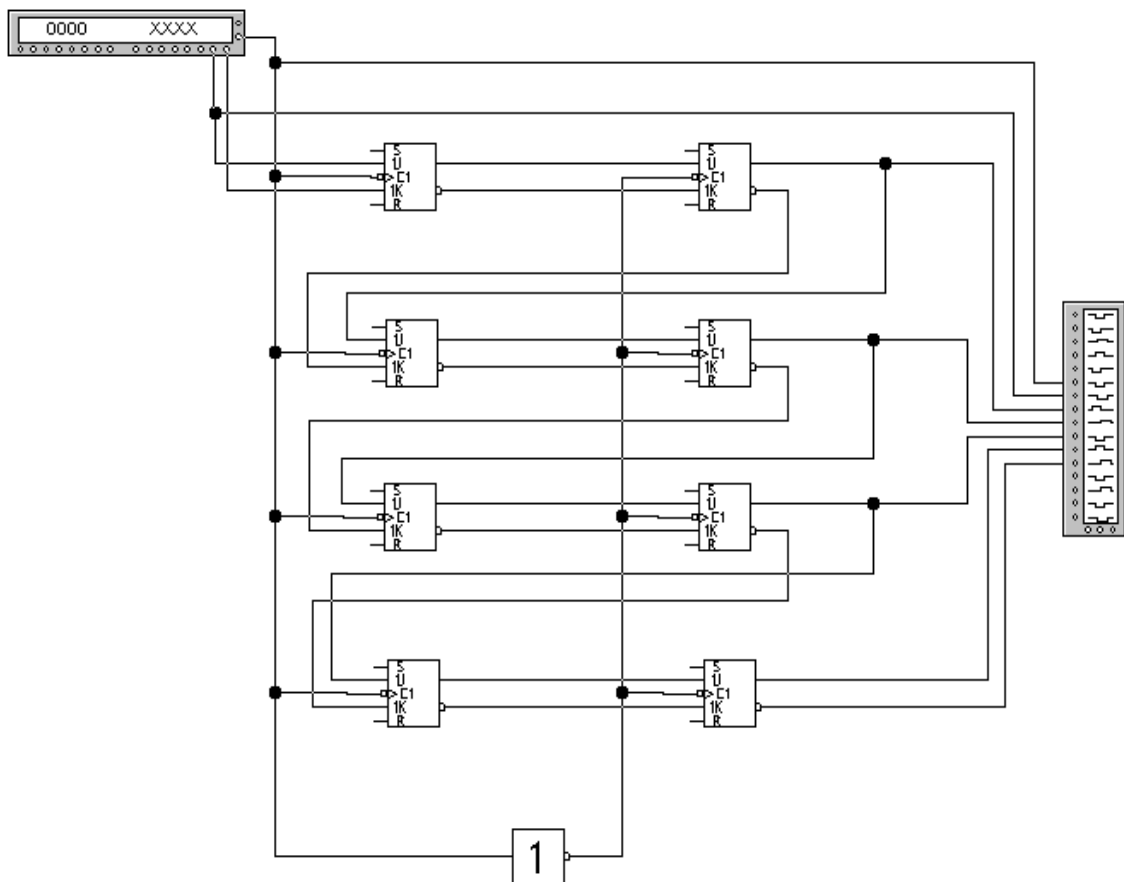
Регістри, призначені тільки для приймання (записування), зберігання і передачі інформації, називаються *елементарними* або *фіксаторами*. Регістри, в яких зберігання даних поєднується з мікроопераціями зсуву, називаються *зсувовими* (*регістрами зсуву*). Елементарні регістри будують на одноступеневих тригерах, а регістри зсуву – на двоступеневих або D-тригерах з динамічним керуванням. Логічна функція регістра позначається буквами RG (від англійського *register* – регістр). Регістри забезпечують зберігання команд, адреси пам'яті, результатів операцій, індексів та ін.

Регістри класифікують за такими ознаками:

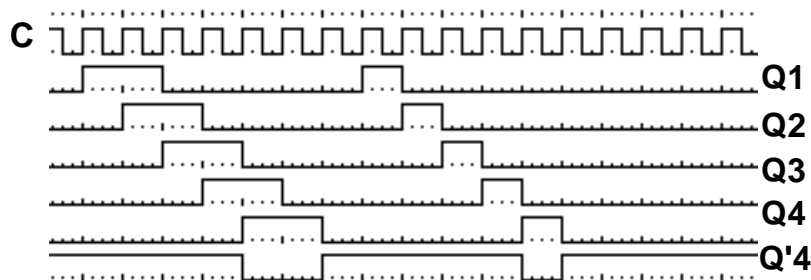
- 1) способом керування записуванням (асинхронні та синхронні);
- 2) способом записування і видачі двійкових слів (паралельні, послідовні й універсальні); у паралельних регістрах записування і видача слів виконується одночасно всіма розрядами, а в послідовних – розряд за розрядом в напрямку від молодших розрядів до старших або навпаки; універсальні регістри забезпечують як паралельний, так і послідовний обмін інформацією;
- 3) кількістю ліній для представлення значення одного розряду слова [біта інформації] (однофазні й пара фазні); при однофазному поданні значення

кожного розряду слова передається по одній пінії зв'язку, а при парафазному – по двох лініях [одночасно відображається пряме та інверсне значення розряду];

- 4) числом тактів для записування слова (одно-, дво- і багатотактні);
- 5) складом мікрооперацій, які виконуються (установлювальні, записування, читання, порозрядні логічні й зсуву, а також перетворення послідовного коду в паралельний і навпаки);
- 6) напрямом зсуву (односторонні [лівий або правий зсув] і двосторонні [реверсивні]);
- 7) типом тригерів, що використовуються (найчастіше *RS*-, *JK*-, *D*-тригери);
- 8) елементною структурою (потенціальні, імпульсні й потенціально-імпульсні).

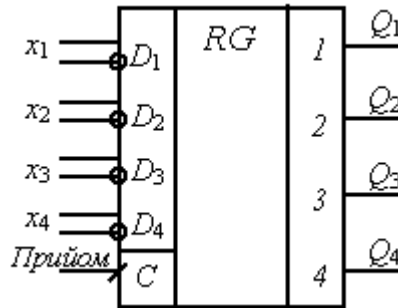


а)



б)

Рисунок 7.2 – 4-розрядний регістр зсуву на однокліткових *JK*-тригерах:
а) схема дослідження в програмному пакеті *EWB 5.12*;
б) часова діаграма



**Рисунок 7.3 – 4-розрядний регістр зсуву.
Умовне графічне позначення**

У синхронних регістрах операції виконують по тактуючому сигналу T . Настроювання регістра на відповідну мікрооперацію здійснюється попередньою установкою на керуючих входах коду мікрооперації. Число розрядів у коді визначається зі співвідношення

$$T \geq \lceil \log_2 k \rceil, \quad (7.1)$$

де k – кількість мікрооперацій.

В асинхронних регістрах тактовий вхід T відсутній. Кожна i -та мікрооперація виконується під дією власного керуючого сигналу. Кількість таких сигналів дорівнює числу мікрооперацій k , тобто, в загальному випадку, більше, ніж розрядів у коді мікрооперації синхронного регістра.

При виконанні мікрооперацій у кожному розряді регістра здійснюється однакове перетворення інформації.

Характер мікрооперації можна описати через стани сигналів у момент часу, що передуює виконанню мікрооперації, і в момент завершення мікрооперації.

Зазвичай в регістрах всі розряди виконуються однаково. Розряд синхронного регістра (без врахування кіл видачі інформації) може бути побудований як на синхронному, так і на асинхронному тригері.

Асинхронні регістри також можуть бути реалізовані з використанням і синхронних, і асинхронних тригерів. Синхронні тригери найбільш зручні, коли на регістрі виконується одна мікрооперація (наприклад, мікрооперація зсуву).

На **рис. 7.2** зображено схему 4-розрядного регістра зсуву на одноктактних RS -тригерах, побудовану в програмному пакеті *EWB*., а на **рис. 7.3** – відповідне їй умовне графічне позначення.

Операція зсуву коду – це переміщення в регістрі всіх розрядів коду слова на однакову кількість позицій. На вхід подається парафазний послідовний код. Розряди слова, що вийшли з розрядної сітки губляться.

Дешифратори.

Дешифратором називається комбінаційна схема, з кількома входами і виходами, яка перетворює вхідний двійковий код у сигнал лише на одному з виходів. Тобто, на одному з виходів дешифратора з'являється «1», а на інших виходах – «0» (або навпаки, якщо використовуються інверсні виходи). В загальному

випадку дешифратор має n входів та до $m = 2^n$ виходів (n – розрядність [довжина] коду). Дешифратор із максимально можливою кількістю виходів $m = 2^n$ називається *повним* (відповідно, всі інші – *неповними*).

Функціонування повного дешифратора описується системою логічних виразів вигляду:

[illegible]

де X_1, \dots, X_n – вхідні двійкові змінні; F_0, F_1, \dots, F_{m-1} – вихідні логічні функції, що являють собою мінтерми (конституенти «1») n змінних.

Індекс функції F_i визначає номер обраного виходу і відповідає десятковому еквіваленту вхідного коду. Вихід, на якому з'являється керуючий сигнал, називається *активним*. Якщо значення сигналу на активному виході відображається логічною «1», то на решті пасивних виходів встановлюється логічний «0». Двійковий код, який вміщує завжди тільки одну одиницю, а інші – нулі, називається *унітарним*. Тому дешифратор є перетворювачем вхідного позиційного коду в унітарний вихідний код.

У дешифраторах в інтегральному виконанні стан активного виходу часто відображається значенням логічного «0», а на інших пасивних виходах встановлюється логічна «1». Функціонування повного дешифратора з інверсними виходами представляється системою вигляду:

[illegible]

де L_0, L_1, \dots, L_{m-1} – вихідні логічні функції, що є макстермами (конституенти «0») n змінних.

Індекс функції L_i визначає номер вибраного виходу і відповідає десятковому еквіваленту вхідного коду. Між двома видами вихідних функцій існує простий зв'язок:

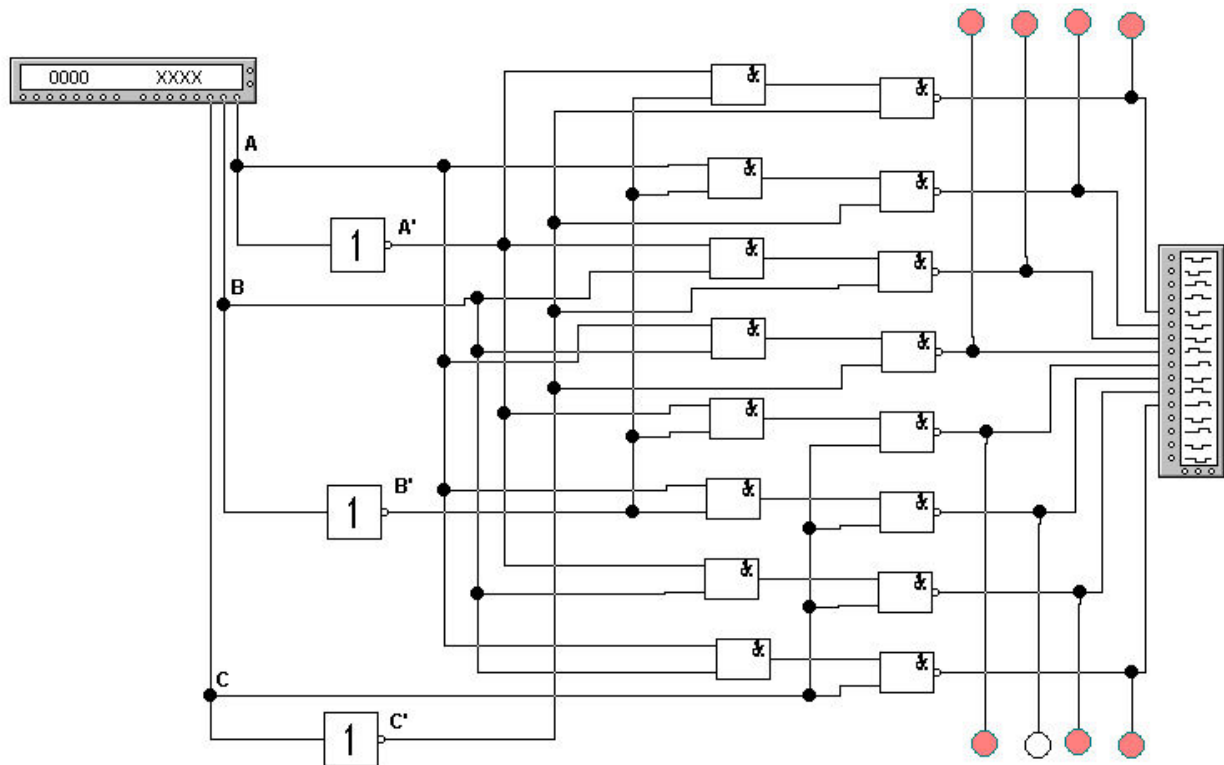
$$F_i = \overline{L_i}. \quad (7.4)$$

Крім інформаційних входів дешифратори зазвичай мають додаткові входи керування. Сигнали на цих входах, наприклад, дозволяють функціонування дешифратора або переводять його в пасивний стан, при якому, незалежно від сигналів на інформаційних входах, на всіх виходах встановиться рівень логічної «1».

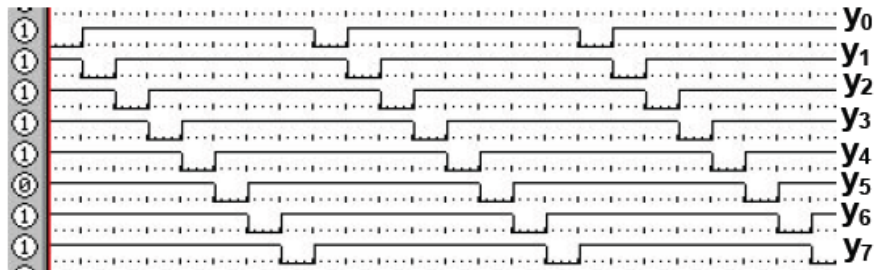
Тобто, можна сказати, що існує деяка функція дозволу, значення якої визначається станами керуючих входів. У дешифраторів з декількома входами керування функція дозволу, як правило, представляє собою логічний добуток всіх

дозволяючих [стробуючих] сигналів керування.

Зазвичай входи керування використовуються для збільшення розрядності дешифраторів або при паралельній роботі декількох схем на загальні вихідні лінії.



а)



б)

Рисунок 7.4 – Дешифратор для 3-розрядного вхідного слова [3×8]:

а) схема дослідження в програмному пакеті EWB 5.12;

б) часова діаграма

Дешифратори класифікують за такими ознаками:

- 1) способом структурної організації (одноступеневі [лінійні] і багатоступеневі, в т. ч. пірамідальні та прямокутні [матричні]);
- 2) форматом вхідного коду (двійкові, двійково-десяткові);
- 3) розрядністю коду, який дешифрується (2, 3, ..., n);
- 4) формою подачі вхідного коду (з однофазними і парафазними входами);
- 5) кількістю виходів (повні й неповні дешифратори);
- 6) видом вхідних стробуючих сигналів (в прямому або інверсному значеннях);
- 7) типом використовуваних логічних елементів («І», «НЕ», «АБО», «І-НЕ», «АБО-НЕ» і т. д.).

До основних характеристик дешифратора відносять: кількість ступенів [каскадів] дешифрації, кількість використаних логічних елементів або мікросхем, загальне число входів логічних елементів, час дешифрації і споживану потужність.

На **рис. 7.4 а)** показано один із способів побудови дешифратора для 3-розрядного вхідного слова [дешифратор 3×8], а на **рис. 7.4 б)** – часова діаграма його роботи, при подачі на вхід послідовності двійкових чисел у порядку зростання.

Варіанти умовних графічних позначень дешифраторів на схемах електричних принципів відображені на **рис. 7.5**.

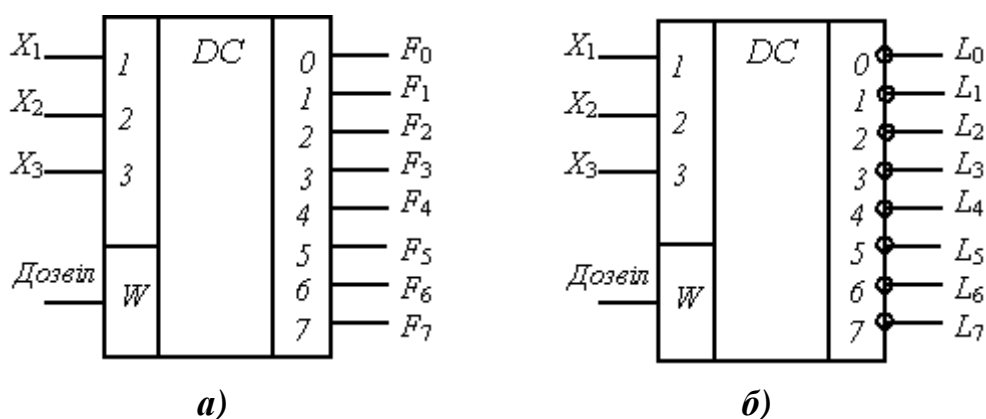


Рисунок 7.5 – Дешифратор 3×8 . Умовне графічне позначення:
а) при прямих виходах; б) при інверсних виходах

Логічна функція дешифратора позначається буквами DC (від англійського *decoder* – дешифратор). Мітки лівого додаткового поля в умовному позначенні відображають десяткові ваги вхідних змінних, а мітки правого додаткового поля відповідають десятковим еквівалентам вхідних комбінацій двійкових змінних. У схему дешифраторів вбудовуються один або два стробуючих [дозволяючих] входи, наприклад, W (див. **рис. 7.5**). За допомогою сигналу на вході W визначається момент спрацювання дешифратора; крім того, вхід W використовується для нарощування розрядності вхідного коду. На практиці повний дешифратор на n входів і m виходів для стислості називають дешифратором «з n у m » [або « $n \rightarrow m$ », або $n \times m$]. Наприклад, у дешифраторі «з 3 у 8» активізується одна з 8 вихідних ліній.

В комп'ютерах та іншій цифровій техніці дешифратори використовують для виконання таких операцій:

- дешифрації коду операції, записаного в регістр команд процесора, що забезпечує вибір потрібної мікропрограми;
- перетворення коду адреси операнда в команді у керуючі сигнали вибору заданої комірки пам'яті в процесі запису або читання інформації;
- забезпечення візуалізації на зовнішніх пристроях;
- реалізації логічних операцій та побудови мультиплексорів і демультиплексорів.

Дешифратори випускаються, як правило, у вигляді мікросхем із кількістю виходів 4, 8, 10, 16, 32. Ряд мікросхем мають інверсні виходи.

Шифратори.

Шифратором називається функціональний вузол цифрової техніки,

призначений для перетворення m -розрядного унітарного коду у вихідний n -розрядний двійковий позиційний код. Двійкові шифратори виконують функцію, обернену функції дешифратора. При активізації однієї з вхідних ліній шифратора, на його виходах формується код, який відображає номер активного входу (саме тому інколи кажуть, що шифратори перетворюють десятковий код у двійковий [хоча вхідний унітарний код може бути і вісімковим, і шіснадцятковим]).

Як і дешифратори, шифратори можуть бути *повними* і *неповними*. Повний двійковий шифратор має $m = 2^n$ входів і n виходів. А в неповному завжди $m < 2^n$. Наприклад, шифратор «з 8 у 3» $[8 \times 3]$ є повним, бо він реалізує повний набір можливих комбінацій змінних X_i ($m = 2^n = 2^3 = 8$). Прикладом неповного шифратора, який найчастіше зустрічається на практиці, є шифратор «з 10 у 4» [або « $10 \rightarrow 4$ », або 10×4], що використовується для кодування десятичних чисел у двійково-десятковий код [код 8421] (див. **табл. 7.1**).

Таблиця 7.1 – Таблиця істинності шифратора 10×4

№ набору	Входи										Виходи			
	X_9	X_8	X_7	X_6	X_5	X_4	X_3	X_2	X_1	X_0	F_3	F_2	F_1	F_0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
2	0	0	0	0	0	0	0	1	0	0	0	0	1	0
3	0	0	0	0	0	0	1	0	0	0	0	0	1	1
4	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	0	0	0	0	1	0	0	0	0	0	0	1	0	1
6	0	0	0	1	0	0	0	0	0	0	0	1	1	0
7	0	0	1	0	0	0	0	0	0	0	0	1	1	1
8	0	1	0	0	0	0	0	0	0	0	1	0	0	0
9	1	0	0	0	0	0	0	0	0	0	1	0	0	1

З наведеної у **табл. 7.1** відповідності десятикового та двійково-десятикового кодів, випливає, що, наприклад, змінна F_0 на виході дорівнює логічній «1», якщо це значення має одна з вхідних змінних X_1, X_3, X_5, X_7, X_9 . Отже, $F_0 = X_1 \vee X_3 \vee X_5 \vee X_7 \vee X_9$.

Аналогічним чином, для всіх можливих вхідних комбінацій [наборів] можемо записати:

$$\left. \begin{aligned} F_0 &= X_1 \vee X_3 \vee X_5 \vee X_7 \vee X_9 \\ F_1 &= X_2 \vee X_3 \vee X_6 \vee X_7 \\ F_2 &= X_4 \vee X_5 \vee X_6 \vee X_7 \\ F_3 &= X_8 \vee X_9 \end{aligned} \right\}. \quad (7.5)$$

Одержані вирази (7.5) характеризують роботу шифратора. Неважко побачити, що у шифраторі даного типу сигнал, що подається на вхід X_0 , не використовується. Тому відсутність сигналу на будь-якому з входів X_i трактується схемою, як наявність на вході нульового сигналу [логічного «0»].

Варіант побудови найпростішого шифратора $[4 \times 2]$ наведено на **рис. 7.6**.

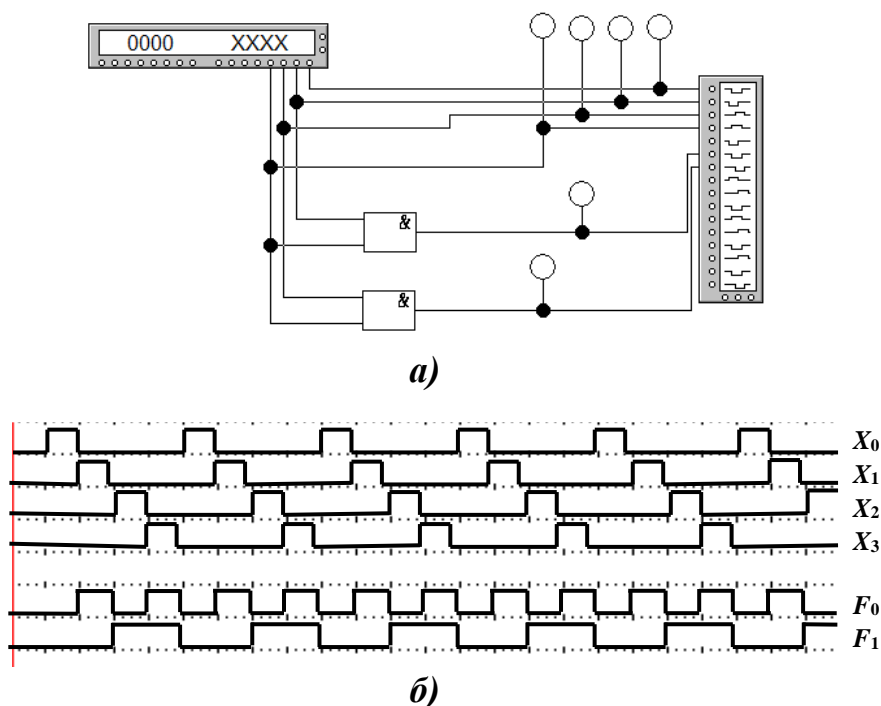


Рисунок 7.6 – Шифратор 4×2:
а) схема дослідження в програмному пакеті EWB 5.12;
б) часова діаграма

Вхідні величини надходять за 4 лініями: X_3 , X_2 , X_1 , X_0 . А вихідні величини F_1 та F_0 формуються за такими співвідношеннями:

$$\left. \begin{aligned} F_0 &= X_1 \vee X_3 \\ F_1 &= X_2 \vee X_3 \end{aligned} \right\} \quad (7.6)$$

Вхідний сигнал X_0 для шифрування не використовується.

Умовне графічне позначення шифратора на схемах електричних принципів зображено на **рис. 7.7**.

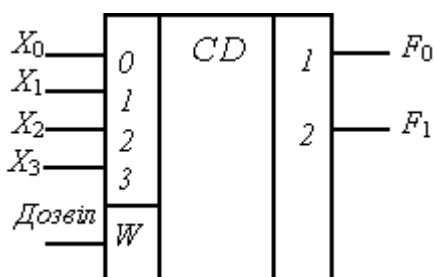


Рисунок 7.7 – Шифратор 4×2.
Умовне графічне позначення

Функція шифратора позначається у широкому полі буквами CD (від англійського *coder* – шифратор). Входи шифратора нумеруються послідовними десятковими цифрами $0, 1, 2, \dots, m-1$. А позначення виходів відображають десяткові ваги вихідних двійкових змінних $1, \dots, 2^{n-1}$. Як і у схемах дешифраторів, у шифраторах можуть додатково вводиться стробуючі [дозволяючі] входи W . Ці ж

входи можуть використовуватись для каскадування шифраторів (з метою збільшення розрядності вхідного слова).

У цифровій техніці шифратори використовуються для таких операцій:

- перетворення унітарного вхідного коду у вихідний двійковий позиційний код; введення десяткових даних з клавіатури;
- визначення старшої одиниці в слові;
- передачі інформації між різними пристроями при обмеженій кількості ліній зв'язку.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.

2. Побудувати схеми регістру зсуву, дешифратора та шифратора (див. *рис. 7.2, рис. 7.4 та рис. 7.6*).

Примітка: Зверніть увагу на те, що при дослідженні регістру зсуву на однотактних *RS*-тригерах на входи першого тригера з генератора слів мають подаватись взаємно протилежні сигнали.

3. Після складання кожену схему підключити до приладів генератора слів і логічного аналізатора. На входи логічного аналізатора необхідно подавати не тільки вихідні сигнали досліджуваної схеми, але і використовувані при цьому вхідні сигнали генератора слів.

4. Використовуючи засоби і можливості програми *EWB*, відлагодити роботу схеми.

Для побудови й запуску схеми:

- а) перетягнути мишею необхідні елементи схеми з бібліотек компонентів;
- б) оптимально розмістити їх у робочому просторі;
- в) з'єднати компоненти провідниками;

Примітка: Для зменшення розмірів схеми на екрані вся вона або її частина можуть бути згорнуті у підсистему («мікросхему») і позначатись в подальшому одним елементом. Для виконання цієї дії необхідно скористатись підпунктом *Create Subcircuit* пункту меню **Circuit**. У віконці, що відкриється необхідно задати назву новостворюваного елемента і обрати кнопку *Replace in Circuit*.

г) задати параметри або моделі компонентів;

д) під'єднати генератор слів та світлодіодні індикатори, а також логічний аналізатор;

е) задати таблицю вхідних сигналів на генераторі слів;

ж) увімкнути подачу вхідного сигналу за допомогою органів керування генератора слів або вимикача розташованого в правому верхньому кутку екрана.

4. За результатами досліджень побудувати часові діаграми. При цьому, починати дослідження кожної наступної схеми слід тільки після перевірки викладачем правильності одержаних результатів для попередньої.

СКЛАД ЗВІТУ

- 1. Назва роботи.
- 2. Мета.

3. Схеми досліджуваних 4-розрядного регістру зсуву, дешифратора 3×8 та шифратора 4×2 та відповідні їм умовні графічні зображення (у відповідності до ГОСТ 2.708-81 ЄСКД).
4. Результати досліджень у вигляді часових діаграм.
5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛАБОРАТОРНА РОБОТА №8

ВИВЧЕННЯ РОБОТИ АРИФМЕТИЧНИХ СУМАТОРІВ

Мета роботи: ознайомитись з принципами побудови та роботи арифметичних суматорів за допомогою можливостей навчальної прикладної програми *EWB*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Арифметичним суматором називається функціональний вузол комп'ютерної та іншої цифрової техніки, призначений для додавання двох n -розрядних слів [чисел] (операція віднімання замінюється додаванням слів в оберненому або доповняльному коді; операції множення та ділення зводяться до реалізації багаторазового додавання та зсуву). Суматори є важливою складовою частиною арифметико-логічних пристроїв (АЛП).

Як правило, суматор складається з окремих схем, які називаються *1-розрядними суматорами*. Вони виконують усі дії з додавання значень однойменних розрядів двох чисел [операндів].

Умове графічне позначення суматора наведено на на схемах електричних принципів зображено на **рис. 8.1**.

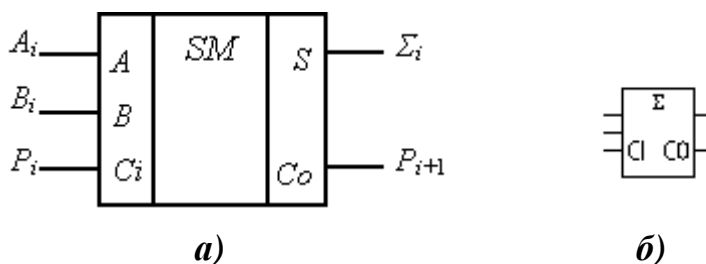


Рисунок 8.1 – 1-розрядний арифметичний суматор:
а) умове графічне позначення на схемах електричних принципів;
б) зображення у програмному пакеті EWB 5.12

Функція суматора позначається (див. **рис. 8.1**) буквами *SM* (від англійського *summarization* – підсумовування) або символом Σ .

При додаванні 2 чисел A і B у кожному i -му розряді відбувається додавання 3 цифр: цифри i -го розряду першого доданку A_i , цифри i -го розряду другого доданку B_i та цифри переносу з молодшого розряду P_i . У результаті додавання на виходах одержується сума S_i та сигнал переносу до старшого розряду P_{i+1} .

Вхід сигналу переносу з попереднього [молодшого] розряду часто позначають (зокрема й у *EWB 5.12*) C_i (від англійського *carry in...* – перенесення до...). А вихід сигналу переносу до наступного [старшого] розряду можуть позначати C_o (від англійського *carry out...* – перенесення з...).

Суматори класифікуються за такими ознаками:

- 1) способом додавання (паралельні, послідовні та паралельно-послідовні);
- 2) кількістю входів (напівсуматори, 1-розрядні та багаторозрядні суматори);
- 3) організацією зберігання результату додавання (комбінаційні, накопичувальні, комбіновані);

- 4) організацією переносів між розрядами (з послідовним, паралельним або груповим переносом);
- 5) системою числення (позиційні [двійкові, двійково-десяткові, трійкові та ін.] і непозиційні);
- 6) розрядністю (довжиною) операндів (8-, 16-, 32-, 64-розрядні);
- 7) способом представлення від'ємних чисел (в оберненому або доповняльному кодах, а також в їхніх модифікаціях);
- 8) часом додавання (синхронні, асинхронні).

1-розрядний арифметичний суматор на ЛЕ.

На **рис. 8.2** наведено схему побудови 1-розрядного арифметичного суматора на ЛЕ.

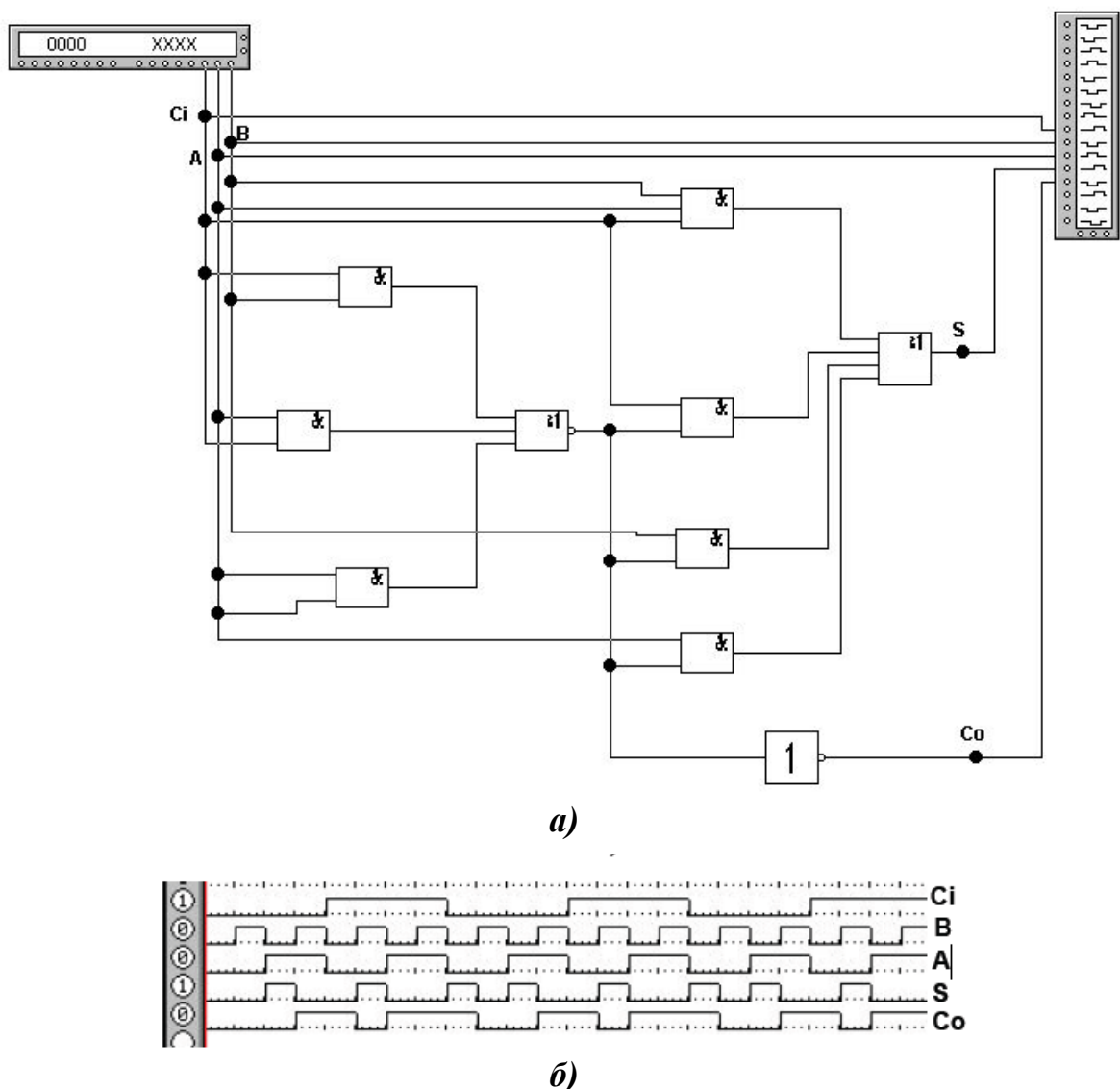


Рисунок 8.2 – 1-розрядний арифметичний суматор на ЛЕ:

а) схема дослідження в програмному пакеті EWB 5.12;

б) часова діаграма

В цій схемі безпосередньо з аргументів A , B , Ci будується функція Co , яка потім використовується в якості четвертого аргументу для побудови функції Σ .

$$\overline{Co} = \overline{Ci \cdot A \vee Ci \cdot B \vee A \cdot B} \quad (8.1)$$

$$\Sigma = Ci \cdot \overline{Co} \vee A \cdot \overline{Co} \vee B \cdot \overline{Co} \vee Ci \cdot A \cdot B \quad (8.2)$$

Саме ці співвідношення і реалізує схема, зображена на **рис. 8.2**.

У паралельних n -розрядних суматорах значення всіх розрядів операндів надходять одночасно на відповідні входи 1-розрядних підсумовуючих схем. У послідовних суматорах значення розрядів операндів та переносів, що запам'ятовувалися в минулому такті, надходять послідовно в напрямку від молодших розрядів до старших на входи одного 1-розрядного суматора. В паралельно-послідовних суматорах числа розбиваються на частини, наприклад, байти, розряди байтів надходять на входи 8-розрядного суматора паралельно [одночасно], а самі байти – послідовно, в напрямку від молодших до старших байтів, з урахуванням запам'ятованого перенесу.

У комбінаційних суматорах результат операції додавання запам'ятовується в регістрі результату. В накопичувальних суматорах процес додавання поєднується із зберіганням результату. Це пояснюється використанням T -тригерів у якості 1-розрядних схем додавання.

Організація переносів практично визначає час виконання операції додавання. Послідовні переноси схемно створюються досить просто, але володіють низькою швидкістю. Паралельні переноси схемно організуються значно складніше, але дають високу швидкість.

Розрядність суматорів знаходиться в широких межах: 4-64 і, навіть, більше. Суматори з постійним інтервалом часу для додавання називаються *синхронними*.

Суматори, в яких інтервал часу для додавання визначається моментом фактичного закінчення операції, називаються *асинхронними*. В асинхронних суматорах є спеціальні схеми, які визначають фактичний момент закінчення додавання і повідомляють про це у пристрій керування. На практиці переважно використовуються синхронні суматори.

Суматори характеризуються такими параметрами:

- швидкістю – часом виконання операції додавання t_{Σ} , який відрховується від початку подачі операндів до одержання результату;
- апаратними витратами (вартість 1-розрядної схеми додавання визначається загальною кількістю логічних входів використаних елементів, а вартість багаторозрядного суматора – загальною кількістю використаних мікросхем);
- споживаною суматором потужністю.

Суматор із послідовним переносом.

Найпростіший спосіб побудови n -розрядного суматора – це послідовне вмикання n 1-розрядних суматорів, які підсумовують відповідні [одноіменні] розряди доданків, як показано, наприклад, на **рис. 8.3**.

Суматор із паралельним переносом.

В суматорі з паралельним переносом тракти всіх 1-розрядних суматорів

ввімкнені послідовно. Тому, навіть при мінімальній затримці тракту переносу 1-розрядного суматора, затримка n -розрядного суматора не може бути менше n .

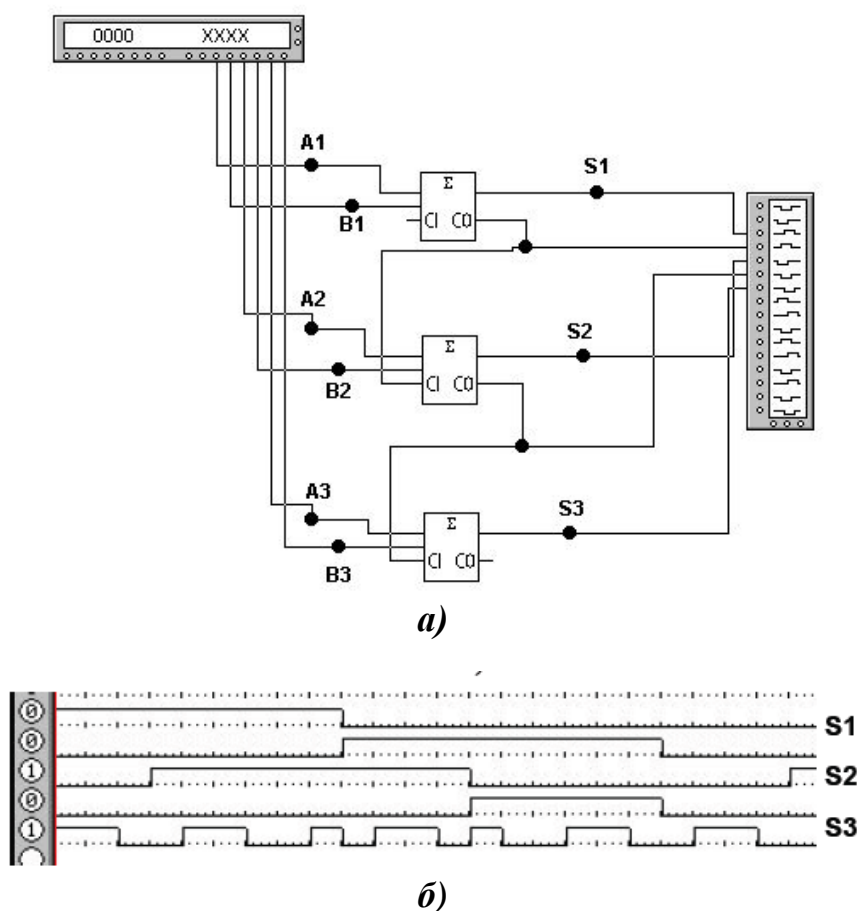


Рисунок 8.3 – 3-розрядний арифметичний суматор із послідовним переносом:

а) схема дослідження в програмному пакеті EWB 5.12;
б) часова діаграма

Для зменшення затримки використовується принцип паралельного переносу, коли вхідний перенос кожного розряду виробляється незалежно від переносу сусіднього меншого розряду.

Витрати обладнання на побудову суматора такого типу, особливо при великій кількості розрядів, настільки великі, що в чистому вигляді він практично не знаходить застосування. Принцип паралельного формування переносу використовується в суматорах із груповим переносом.

Суматор із груповим переносом.

Арифметичний n -розрядний суматор із груповим переносом є компромісом між послідовним і паралельним суматорами. Оскільки він є більш швидкодіючим, ніж суматор із послідовним переносом, а за конструкцією – більш простим і менш матеріаломістким, ніж суматор із паралельним переносом.

Принцип групового переносу застосовується при великій кількості розрядів. Суть його полягає в тому, що кількість n вхідних розрядів розбивають на *групи* приблизно однакового розміру; всередині кожної з груп організовують паралельні

переноси, а між групами – послідовні.

Щоправда, така організація переносів у багаторозрядних арифметичних суматорах потребує введення у схему додаткових елементів для врахування всіх наявних переносів (зокрема, між групами) та можливої корекції кінцевого результату. А це, в свою чергу, може значно підвищити складність конструкції, її матеріаломісткість і, як наслідок, вартість.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. Побудувати схеми 1-розрядного арифметичного суматора на ЛЕ (*рис. 8.2*) та 3-розрядного арифметичного суматора з послідовним переносом (*рис. 8.3*).
3. Після складання кожену схему підключити до приладів генератора слів і логічного аналізатора.
4. Використовуючи засоби і можливості програми *EWB*, відлагодити роботу схем.

Для побудови й запуску схеми:

- а) перетягнути мишею необхідні елементи схеми з бібліотек компонентів;
- б) оптимально розмістити їх у робочому просторі;
- в) з'єднати компоненти провідниками;
- г) задати параметри або моделі компонентів;
- д) під'єднати генератор слів та світлодіодні індикатори (на схемах *рис. 8.2* та *рис. 8.3* не вказані), а також логічний аналізатор;
- е) задати таблицю вхідних сигналів на генераторі слів;
- ж) увімкнути подачу вхідного сигналу за допомогою органів керування генератора слів або вимикача розташованого в правому верхньому кутку екрана.

4. За результатами досліджень побудувати часові діаграми. При цьому, починати дослідження наступної схеми слід тільки після перевірки викладачем правильності одержаних результатів для попередньої.

СКЛАД ЗВІТУ

1. Назва роботи.
2. Мета.
3. Схеми досліджуваних 1-розрядного арифметичного суматора на ЛЕ й 3-розрядного арифметичного суматора з послідовним переносом та відповідні їм умовні графічні зображення (у відповідності до ГОСТ 2.708-81 ЄСКД).
4. Результати досліджень у вигляді часових діаграм.
5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛАБОРАТОРНА РОБОТА №9

КОНСТРУЮВАННЯ КОМБІНАЦІЙНИХ СХЕМ ТА ЇХ ОПТИМІЗАЦІЯ

Мета роботи: вивчити методи конструювання та оптимізації комбінаційних схем за допомогою можливостей навчально-прикладної програми EWB.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Існують три «основних знаряддя праці» для розв'язання задач логіки: умовні позначення ЛЕ, таблиці істинності і булеві вирази.

Логічну функцію для зручності запису і наступного синтезу виражають у вигляді суми добутків змінних – *диз'юнктивної нормальної форми* (ДНФ), або у вигляді добутку їх сум – *кон'юнктивної нормальної форми* (КНФ).

Для кожної логічної функції може існувати декілька рівнозначних диз'юнктивних і кон'юнктивних форм. Водночас є тільки один вид ДНФ або КНФ, у якому функція може бути записана єдиним способом (*досконалі нормальні форми*).

Логічна функція найбільш наочно і повно представляється *таблицею істинності*, у якій для кожної комбінації значень змінних вказується значення функції.

Наприклад, нехай деяка функція F задана у вигляді таблиці (див. **табл. 9.1**).

**Таблиця 9.1 – Таблиця істинності деякої функції F ,
взятої для прикладу**

№ комбінації	Вхідні змінні			Функція F
	A	B	C	
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Для комбінацій змінних 3, 5, 6, 7 функція $F = 1$, що означає для зазначених комбінацій рівність одиниці наступних добутків:

$$\bar{A} \cdot B \cdot C = 1; A \cdot \bar{B} \cdot C = 1; A \cdot B \cdot \bar{C} = 1; A \cdot B \cdot C = 1. \quad (9.1)$$

Комбінації змінних, при яких $F = 1$, називають конститuentами «1» або *мінтермами*. Представлення логічної функції у вигляді суми мінтермів визначає її досконалу ДНФ, тобто в даному випадку:

$$F = \bar{A} \cdot B \cdot C + A \cdot \bar{B} \cdot C + A \cdot B \cdot \bar{C} + A \cdot B \cdot C. \quad (9.2)$$

Функція, обумовлена таблицею істинності, може бути подана не тільки її одиничними, але і нульовими значеннями.

Так, на підставі **табл. 9.1**, можна записати:

$$F = (A + B + C) \cdot (A + B + \bar{C}) \cdot (A + \bar{B} + C) \cdot (\bar{A} + B + C). \quad (9.3)$$

Кожний співмножник у виразі (9.3) знаходиться із суми змінних, для яких функція перетворюється у нуль відповідно до таблиці істинності (див. **табл. 9.1**). Такі суми називають конститuentами «0» або *макстермами*. Таким чином, зміст макстермів визначає досконалу КНФ функції.

Для зменшення кількості ЛЕ, що реалізують функцію, застосовується спрощення її форми (*мінімізація*). Для мінімізації нескладних функцій використовуються алгебраїчні перетворення, а для функцій, що мають велику кількість змінних (понад 3) й велику кількість складових, – спеціальні методи. Одним із таких методів є *метод карт Карно-Вейча*.

Система елементів, що дозволяє будувати на їх базі логічну схему будь-якої складності, називається функціонально повною системою елементів або *базисом*. Базис утворюють ЛЕ «АБО», «І», «НЕ». Крім того, на практиці широко «застосовуються ЛЕ, що виконують не елементарні, щойно перераховані, операції, а реалізують найпростіші функції двох аргументів «АБО-НЕ» (стрілка Пірса), «І-НЕ» (штрих Шеффера) тощо.

Приклади побудови схем на основі булевих виразів наведені на **рис. 9.1** та **рис. 9.2**.

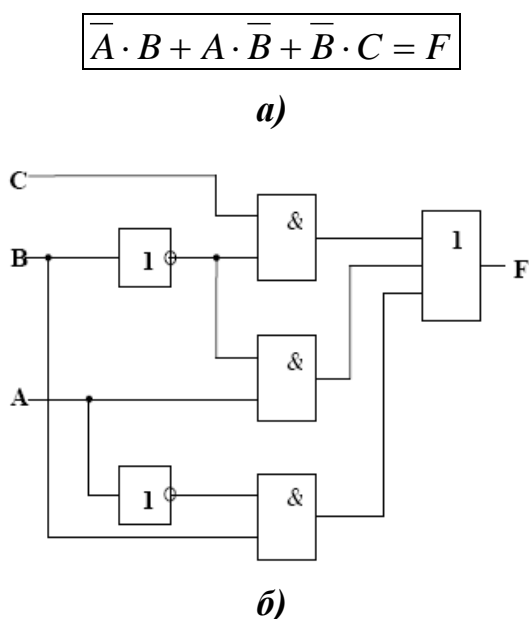


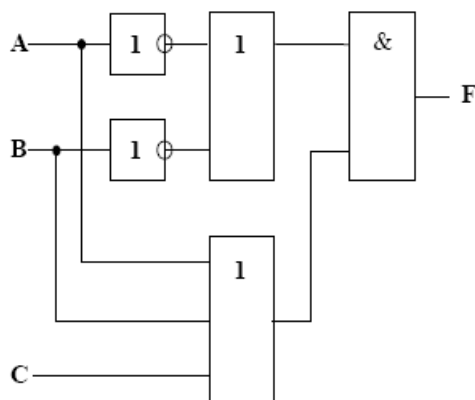
Рисунок 9.1 – Приклад булевого виразу та схеми до спрощення:
а) буле вий вираз; б) схема

Карта Карно-Вейча повинна бути складена так, щоб, по мірі зміщення уздовж сторони карти, на кожному кроці змінювалася лише одна змінна.

Сусідні одиниці об'єднуються в один контур групами по 2^n (тобто 2, 4, або 8) одиниць. Побудова контурів продовжуються доти, поки всі одиниці не виявляться всередині контурів. Кожний контур являє собою новий член спрощеного булевого виразу. Всередині контуру одна й та ж змінна з інверсією і без неї опускається.

$$(A + B + C) \cdot (\bar{A} + \bar{B}) = F$$

а)



б)

Рисунок 9.2 – Приклад булевого виразу та схеми після спрощення за допомогою карт Карно-Вейча:
а) буле вий вираз; б) схема

Приклади мінімізації функцій алгебри логіки методом карт Карно-Вейча наведено на **рис. 9.3**.

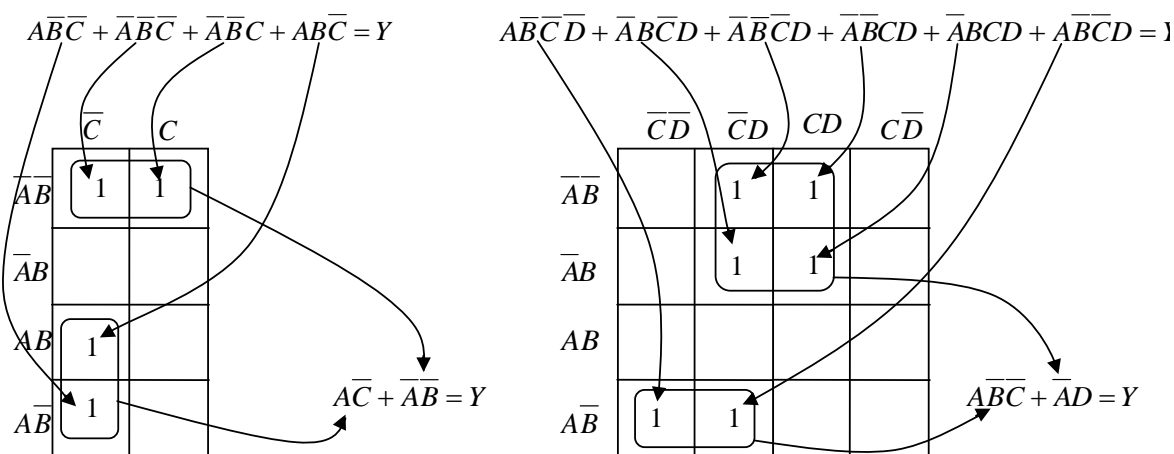


Рисунок 9.3 – Приклади мінімізації функцій алгебри логіки методом карт Карно-Вейча

Для конструювання комбінаційних схем за допомогою програми *EWB* використовується прилад *логічний перетворювач (Logic Converter)*,

Логічний перетворювач – потужний комп'ютерний прилад, що виконує окремі перетворення представлень схеми (**рис. 9.4**). Він може використовуватись для перетворень:

- схеми в таблицю істинності або діаграму кола;
- таблиці істинності в булевий вираз;
- булевого виразу в схему або таблицю істинності.

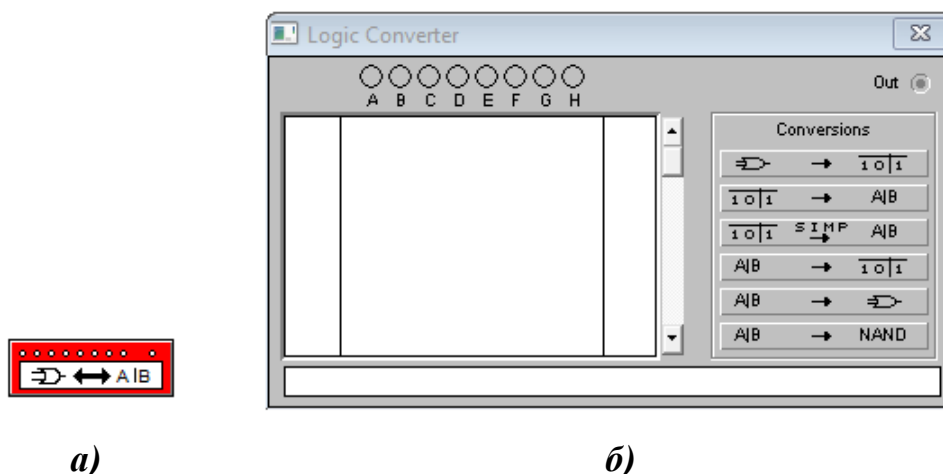


Рисунок 9.4 – Логічний перетворювач (Logic Converter) з EWB 5.12:
а) у компактному вигляді; б) у «розгорнутому» вигляді

Уздовж правого боку відкритого вікна логічного перетворювача (див. **рис. 9.4, б**)) є набір кнопок перетворення (Conversions), що використовуються для виконання операцій. Нижче наведені короткі інструкції щодо виконання різних перетворень.

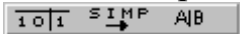
Перетворення схеми в таблицю істинності. Логічний перетворювач може створювати таблицю істинності для схем, що вміщують до восьми входів і одного виходу. Приєднайте входи схеми до клем логічного перетворювача (група з 8 однотипних клем зверху ліворуч [A, B, ..., H]). Тепер з'єднайте вихід схеми з клемою молодшого розряду (окремо від групи праворуч [Out]) на «іконці» компактного вигляду логічного перетворювача. Після цього, відкрийте логічний перетворювач у «розгорнутий» вигляд і натисніть кнопку

Перетворення таблиці істинності в булевий вираз. Щоб перетворювати таблицю істинності в Булевий вираз, натисніть кнопку


Введення таблиці істинності. Щоб створювати таблицю істинності, «натисніть» на номер входних каналів, необхідних Вам [A, B, ..., H], розташованих горизонтально у верхній частині «розгорнутого» логічного перетворювача. Відредагуйте значення у вихідній колонці, щоб задати бажаний вихід для кожного набору входних станів. Щоб змінювати значення, «натисніть» на нього і введіть 1, 0 або X [X означає що припустимі як «1», так і «0»]. В результаті матимете картину, зображену на **рис. 9.5**:

	A	B	C	D	E	F	G	H	OUT
1 4 1	1	0	0	0	1	1	0	1	0
1 4 2	1	0	0	0	1	1	1	0	X
1 4 3	1	0	0	0	1	1	1	1	1
1 4 4	1	0	0	1	0	0	0	0	0
1 4 5	1	0	0	1	0	0	0	1	0
1 4 6	1	0	0	1	0	0	1	0	0

Рисунок 9.5 – Приклад введеної таблиці істинності в логічному перетворювачі

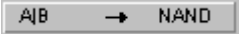
Спрощення булевого виразу. Щоб зменшити булевий вираз, спочатку перетворіть його в таблицю істинності (див. вище), а вже потім спрощуйте. Щоб зменшити булевий вираз натисніть на третю згори кнопку .

Примітка: спрощення булевого виразу вимагає великої кількості пам'яті. Якщо Ваш комп'ютер не має достатньої пам'яті, можливе його «зависання», а операція перетворення залишиться незавершеною.

Перетворення булевого виразу в таблицю істинності. Ви можете ввести булевий вираз в текстовому полі в нижній частині логічного перетворювача (див. *рис. 9.4, б*). Для перетворення введенного булевого виразу в таблицю істинності служить кнопка .

Перетворення булевого виразу в схему. Щоб одержати схему з булевого виразу, натисніть п'яту зверху кнопку .

Логічні вентиля, що виконують вираження, з'являться в робочому просторі. Компоненти виділені так, що Ви можете переміщувати (*move*) їх у різні місця або копіювати (*copy*) і вставляти (*paste*) їх у підсхему.

Створення схеми з ЛЕ «І-НЕ». Щоб створити схему, яка повторює стани булевого виразу, використовуючи тільки лише ЛЕ «І-НЕ», натисніть на останню кнопку .

Приклад спрощення схеми за допомогою логічного перетворювача.

Візьмемо, в якості прикладу, схему, наведену на *рис. 9.5*.

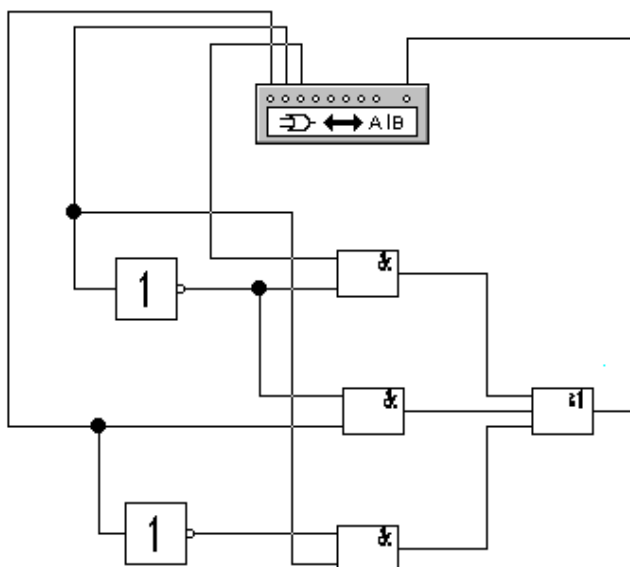


Рисунок 9.5 – Схема на ЛЕ, обрана в якості прикладу для спрощення

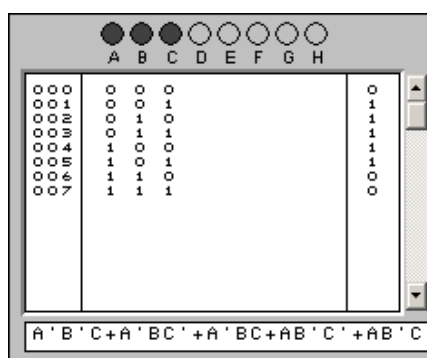
Неважко побачити, що використовуються 3 вхідних змінних (A, B, C).

Булевий вираз і таблиця істинності схеми до спрощення представлені на *рис. 9.6* (як фрагмент «розгорнутого» зображення логічного перетворювача).

А булевий вираз і таблиця істинності схеми після спрощення – на **рис. 9.7**.

Одержана, у результаті використання логічного перетворювача, спрощена схема має вигляд, наведений на **рис. 9.8**.

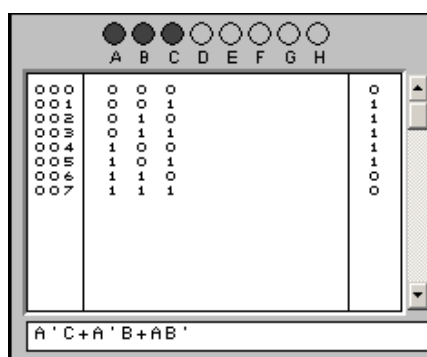
Примітка: спрощена схема з'являється (у «виділеному» червоним кольором вигляді) посередині екрану і може накладатись на інші зображення. Тому одразу після появи її на екрані, не знімаючи виділення, її необхідно перетягнути на вільне місце. Якщо цього не зробити, спрощена схема може переплестися із початковою і їх буде дуже важко роз'єднати.



	A	B	C	D	E	F	G	H
000	0	0	0					0
001	0	0	1					1
002	0	1	0					1
003	0	1	1					1
004	1	0	0					1
005	1	0	1					1
006	1	1	0					0
007	1	1	1					0

$$A'B'C + A'BC' + A'BC + AB'C' + AB'C$$

Рисунок 9.6 – Булевий вираз і таблиця істинності схеми до спрощення



	A	B	C	D	E	F	G	H
000	0	0	0					0
001	0	0	1					1
002	0	1	0					1
003	0	1	1					1
004	1	0	0					1
005	1	0	1					1
006	1	1	0					0
007	1	1	1					0

$$A'C + A'B + AB'$$

Рисунок 9.7 – Булевий вираз і таблиця істинності схеми після спрощення

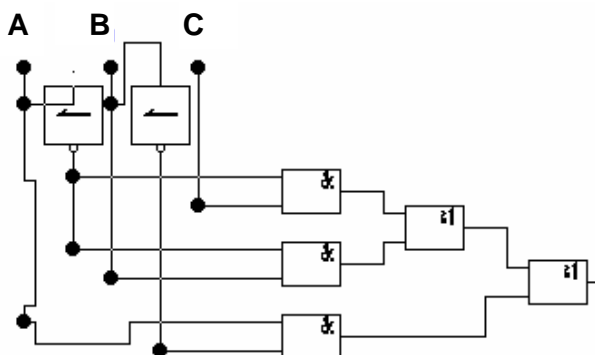


Рисунок 9.8 – Спрощена схема на ЛЕ

Для побудови схеми, еквівалентної спрощеній (**рис. 9.8**), тільки на одних ЛЕ «І-НЕ» [у базисі «І-НЕ»], задаємо булевий вираз і натискаємо останню кнопку логічного перетворювача $A \cdot B \rightarrow \text{NAND}$. В результаті одержуємо схему, зображену

на **рис. 9.9**.

Примітка: як і в попередньому випадку, спрощена схема у базисі «І-НЕ» з'являється (у «виділеному» червоним кольором вигляді) посередині екрану і може накладатись на інші зображення і, навіть, переплітатися з ними. Тому одразу після появи схеми на екрані, не знімаючи виділення, її необхідно перетягнути на вільне місце.

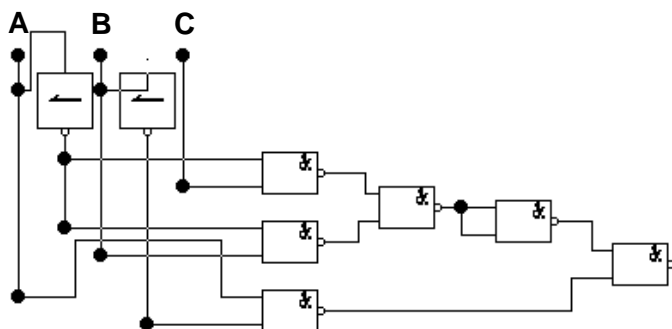


Рисунок 9.9 – Спрощена схема у базисі «І-НЕ»

Інколи спрощена схема, побудована на одних лише ЛЕ «І-НЕ», на перший погляд може видатись більш складною, ніж попередня спрощена схема, складена із будь-яких існуючих ЛЕ. Проте на практиці, через те, що мікросхеми містять в собі по декілька однотипних ЛЕ, більш громіздка, з вигляду, схема у базисі «І-НЕ» може звестися до використання меншої кількості радіокомпонентів [корпусів]. І вся схема може звестися, навіть, до використання єдиної мікросхеми (з декількома ЛЕ «І-НЕ» всередині), виводи якої будуть з'єднані у відповідності до спрощеної схеми у базисі «І-НЕ».

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. Одержати таблицю істинності для булевого виразу, відповідно до варіанта; мінімізувати булевий вираз і, за необхідності, одержати для нього нову таблицю істинності; побудувати електричну схему, використовуючи будь-які ЛЕ; побудувати електричну схему в базисі «І-НЕ».
3. Результати всіх виконаних дій відобразити у звіті.

СКЛАД ЗВІТУ

1. Назва роботи.
2. Мета.
3. Булевий вираз, одержаний згідно варіанту.
4. Результати роботи із логічним перетворювачем:
 - таблиця істинності;
 - мінімізований булевий вираз;
 - нова таблиця істинності, якщо це викликано мінімізованим булевим виразом;
 - електрична схема, на основі будь-яких ЛЕ;

– електрична схема в базисі «I-HE».

5. Висновки.

Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297×210 мм).

ЛІТЕРАТУРА ДО ЛАБОРАТОРНОГО КУРСУ

1. Бабич М. П. Комп'ютерна схемотехніка: Навч. посібник. / М. П. Бабич, І. А. Жуков. – К.: «МК-Прес», 2004. – 412 с.
2. Бубнов А. В. Аналоговая и цифровая схемотехника: учеб. пособие. / А. В. Бубнов, К. Н. Гвозденко, М. В. Гокова. – Омск: Изд-во ОмГТУ, 2010. – 80 с.
3. Волгов Д. В. Детали и узлы радиоэлектронной аппаратуры. Изд. 2-е, перераб. и доп. / Д. В. Волгов. – М.: Энергия, 1977. – 656 с.
4. Гершунский Б. С. Основы электроники и микроэлектроники. 4-е изд. перераб. и дополн. / Б. С. Гершунский. – К.: Выща школа, 1989. – 442 с.
5. Електроніка і мікросхемотехніка: Підручник. У 4-х т. Том 1. Елементна база електронних пристроїв. / [Сенько В. І., Панасенко М. В., Сенько Є. В., Юрченко М. М., Сенько Л. І., Ясінський В. В.]. За ред. В. І. Сенька. – К.: Обереги, 2000. – 300 с.
6. Жеребцов И. П. Основы электроники. – 5-е изд. перераб. и доп. / И. П. Жеребцов. – Л.: Энергоатомиздат. Ленингр. отд-ние, 1989. – 352 с.
7. Каган Б. М. Электронные вычислительные машины и системы. / Б. М. Каган. – М.: Энергоатомиздат, 1991. – 592 с.
8. Калабеков Б. А. Цифровые устройства и микропроцессорные системы. / Б. А. Калабеков. – М.: Телеком, 2000. – 336 с.
9. Каплан Д. Практические основы аналоговых и цифровых схем. / Д. Каплан, К. Уайт; пер. с англ. А. А. Кузьмичевой; под ред. А. А. Лапина. – М.: Техносфера. 2006. – 176 с.
10. Кардашев Г. А. Виртуальная электроника. Компьютерное моделирование аналоговых устройств. / Г. А. Кардашев. – М.: Горячая линия – Телеком, 2002. – 260 с.
11. Каяцкас А. А. Основы радиоэлектроники. / А. А. Каяцкас. – М.: Высшая школа, 1988. – 464 с.
12. Кочубей О. О. Прикладна теорія цифрових автоматів. Логічні основи: навч. посіб. / О. О. Кочубей, О. В. Сопільник. – Дніпропетровськ: РВВ ДНУ; Вид-во ДНУ, 2009. – 264 с.
13. Кравчук А. Ф. Основы дискретной математики. / А. Ф. Кравчук. – К.: УМК ВО, 1992. – 196 с.
14. Марченко А. Л. Основы электроники. Учеб. пособие для вузов. / А. Л. Марченко. – М.: ДМК Пресс, 2008. – 296 с.
15. Нефедов В. И. Основы радиоэлектроники и связи: учеб. пособие. / В. И. Нефедов, А. С. Сигов; Под ред. В.И. Нефедова. – М.: Высш. шк., 2009. – 735 с.

16. Партала О. Н. Цифровая электроника. / О. Н. Партала. – СПб.: Наука и техника, 2000. – 208 с.
17. Першин В. Т. Основы современной радиоэлектроники: учеб. пособие. / В. Т. Першин. – Ростов-на-Дону: Феникс, 2009. – 541 с.
18. Потёмкин И. С. Функциональные узлы цифровой автоматики. / И. С. Потёмкин. – М.: Энергоатомиздат, 1988. – 320 с.
19. Рычина Т. А. Электрорадиоэлементы. Учебник. / Т. А. Рычина. – М.: Сов. Радио, 1976. – 336 с.
20. Степаненко И. П. Основы микроэлектроники. 2-е изд. перераб. и доп. / И. П. Степаненко. – М.: Лаборатория Базовых Знаний, 2001. – 488 с.
21. Трушкин Н. С. Основы аналоговой электроники: учеб. пособие. / Н. С. Трушкин. – М.: МИФИ, 2008. – 76 с.
22. Тули М. Справочное пособие по цифровой электронике. / М. Тули. – М.: Энергоатомиздат, 1990. – 176 с.
23. Уилкинсон Б. Основы проектирования цифровых схем. / Б. Уилкинсон. Пер. с англ. – М.: Издательский дом «Вильямс», 2004. – 320 с.
24. Фрике К. Вводный курс цифровой электроники. / К. Фрике. – М.: Техносфера, 2003. – 432 с.
25. Чернышова Т. И. Моделирование электронных схем: Учебное пособие. / Т. И. Чернышова, Н. Г. Чернышов. – Тамбов : Изд-во ГОУ ВПО ТГТУ, 2010. – 80 с.
26. Шкурко А. И. Компьютерная схемотехника в примерах и задачах. / Шкурко А. И., Процюк Р. О., Корнейчук В. И. – К.: «Корнійчук», 2003. – 144 с.
27. Электрические и электронные компоненты устройств и систем: учеб.-метод. пособие. / В. В. Баранов [и др.]. – Минск: БГУИР, 2019. – 136 с.
28. Электроника. Энциклопедический словарь. / [под ред. В. Г. Колесникова]. – М.: Сов. энциклопедия, 1991. – 695 с.

Навчально-методичне видання

ЕЛЕКТРОННІ КОМПОНЕНТИ

Частина II

Методичні вказівки

до виконання лабораторних робіт з навчальних дисциплін:

«Електронні компоненти телекомунікаційних систем»

за спеціальністю

172 «Телекомунікації та радіотехніка»

та

«Основи комп'ютерної схемотехніки»

за спеціальностями

141 «Електроенергетика, електротехніка та електромеханіка»;

151 «Автоматизація та комп'ютерно-інтегровані технології»

для студентів денної та заочної форм навчання

Укладачі: *Р. В. Жесан*

О. П. Голик