

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

«ОСНОВИ КОМП'ЮТЕРНОЇ СХЕМОТЕХНІКИ»

М е т о д и ч н і в к а з і в к и

до виконання лабораторних робіт
для студентів денної та заочної форми навчання
за спеціальностями:

- 151 «Автоматизація та комп'ютерно-інтегровані технології»**
- 141 «Електроенергетика, електротехніка та електромеханіка»**
- 172 «Телекомунікації та радіотехніка»**

Затверджено на засіданні кафедри
автоматизації виробничих процесів,
протокол № 1 від 30.08.2019 р.

Кропивницький 2019

Основи комп'ютерної схемотехніки. Методичні вказівки до виконання лабораторних робіт для студентів денної та заочної форми навчання за спеціальностями: 151 «Автоматизація та комп'ютерно-інтегровані технології», 141 «Електроенергетика, електротехніка та електромеханіка», 172 «Телекомунікації та радіотехніка» / Р. В. Жесан, О. П. Голик, В. О. Зубенко. – Кропивницький: ЦНТУ. -2019. - 60 с.

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

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

Зубенко В. О., кандидат технічних наук, доцент.

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

ЗМІСТ

ВСТУП.....	4
ЛАБОРАТОРНА РОБОТА №1	
<i>ВИВЧЕННЯ СИСТЕМ ЧИСЛЕННЯ.....</i>	9
ЛАБОРАТОРНА РОБОТА №2	
<i>АРИФМЕТИЧНІ ОСНОВИ ПОБУДОВИ ЕЛЕМЕНТІВ І ВУЗЛІВ ОБЧИСЛЮВАЛЬНИХ МАШИН І СИСТЕМ.....</i>	19
ЛАБОРАТОРНА РОБОТА №3	
<i>ВИВЧЕННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ</i>	25
ЛАБОРАТОРНА РОБОТА №4	
<i>ДОСЛІДЖЕННЯ ТРИГЕРІВ.....</i>	36
ЛАБОРАТОРНА РОБОТА №5	
<i>ПОБУДОВА ЛІЧИЛЬНИКІВ І ВИВЧЕННЯ ЇХ РОБОТИ.....</i>	40
ЛАБОРАТОРНА РОБОТА №6	
<i>ПОБУДОВА РЕГІСТРІВ, ДЕШИФРАТОРІВ І ВИВЧЕННЯ ЇХ РОБОТИ</i>	44
ЛАБОРАТОРНА РОБОТА №7	
<i>ВИВЧЕННЯ РОБОТИ АРИФМЕТИЧНИХ СУМАТОРІВ.....</i>	48
ЛАБОРАТОРНА РОБОТА №8	
<i>КОНСТРУЮВАННЯ КОМБІНАЦІЙНИХ СХЕМ ТА ЇХ ОПТИМІЗАЦІЯ.....</i>	53
ЛІТЕРАТУРА ДО ЛАБОРАТОРНОГО КУРСУ.....	59

ВСТУП

Метою лабораторного практикуму є закріплення та поглиблення базових знань з основ комп'ютерної схемотехніки й отримання навиків з практичного їх застосування.

Вчасно захищеною вважається робота захист якої студентом відбувся в межах часу передбаченого для цієї роботи в робочій навчальній програмі та згідно з розкладом занять.

За результатами семестру студент отримує підсумкову оцінку за 100-бальною системою.

Загальна кількість балів отримана за семестр відповідає оцінці:

- 1-34 – «незадовільно» з обов'язковим повторним вивченням дисципліни;
- 35-59 - «незадовільно» з можливістю повторного складання;
- 60-63 - «задовільно» («достатньо»);
- 64-73 - «задовільно»;
- 74-81 - «добре»;
- 82-89 - «добре» («дуже добре»);
- 90-100 - «відмінно».

В час комп'ютерно-інтегрованих технологій у будь-якій галузі людської діяльності – в науці, техніці, виробництві - методи і засоби обчислювальної техніки спрямовані на підвищення продуктивності праці. В зв'язку з цим рівень спеціалістів значною мірою визначається їх підготовкою за такими напрямками:

- автоматизоване керування технологічними процесами, включно із автоматизованим контролем та діагностикою технічних засобів;
- використання ЕОМ для автоматизованого проектування, наукових досліджень, адміністративно-організаційного управління.

Оскільки спеціальності 151 «Автоматизація та комп'ютерно-інтегровані технології», 141 «Електроенергетика, електротехніка та електромеханіка», 172 «Телекомунікації та радіотехніка» мають за мету підготовку фахівців високої кваліфікації, курс «Основи комп'ютерної схемотехніки» повинен ознайомити студента з цифровими методами, моделями та пристроями, які дозволили б у інших базових курсах розглядати різноманітні застосування засобів комп'ютерної схемотехніки.

Лабораторні установки з фіксованим набором схемних елементів і певним набором вимірювальних приладів суттєво обмежують можливості навчального процесу. При використанні комп'ютерної програми конструювання електронних схем реалізується принципово новий підхід в розв'язанні навчальних задач. На екрані монітора може бути побудована електрична принципова схема пристрою будь-якої складності, параметри елементів якої

можуть задаватися і змінюватися в широкому діапазоні значень. При цьому вимірювання основних електричних параметрів можливе одночасно в необмеженій кількості точок.

В даних методичних вказівках для виконання лабораторних робіт використовується програма *Electronics Workbench* (електронний конструктор), яка дозволяє реалізувати технологію інтерактивної, тобто безпосередньо на екрані модельованої, побудови електричних схем.

Для роботи з *Electronics Workbench (EWB)* необхідно:

- мати елементарні навички роботи на персональному комп'ютері з основними програмами користувача, а головне, з операційною системою *MS Windows 9X*;

- навички роботи в середовищі *Windows* з ручним маніпулятором «миша»;
- основні знання з електроніки та мікросхемотехніки, метрології та імпульсній техніці;

- розуміння англійської мови.

Вікно програми *Electronics Workbench 5.12* наведено на рис 1.

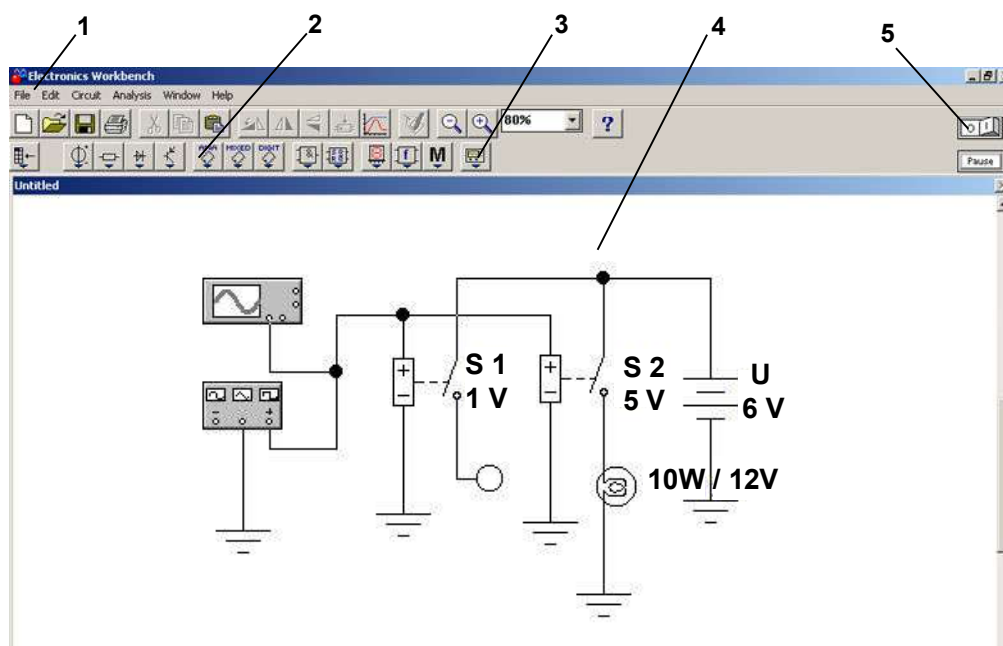


Рис. 1. Вікно програми Electronics Workbench 5.12

Вікно програми *EWB 5.12* містить:

- 1 – поле меню;
- 2 – лінійку бібліотек компонентів;
- 3 – бібліотека контрольно-вимірювальних приладів;
- 4 – робоче поле програми, в якому розташовується модельована схема з підключеними до неї іконками контрольно-вимірювальних приладів і короткий опис схеми (англійською мовою, на рисунку не показаний);
- 5 – вимикач, що вмикає та вимикає роботу модельованої схеми.

При потребі кожен з приладів може бути розгорнутий для встановлення параметрів його роботи і спостереження результатів. Лінійки прокручування використовуються лише для переміщення схеми.

Поле меню містить такі складові:

File- меню файлів. Містить в собі всі команди на тематику пов'язану з файлами (команди, які обслуговують роботу з файлами):

New - спорожнює вікно редагування схеми, зберігаючи її вміст у файлі (назва файлу: <ім'я>.СА4), при умові, що в цьому вікні вже були зроблені зміни;

Save - зберігає поточне вікно редагування схеми в файл на жорсткий диск, або в будь-якій інший, вказаний користувачем;

Save as - зберігає вікно редагування схеми в файл, додатково вказаний користувачем;

Revert to Saved - виконує процедуру, зворотну запису (зчитування того файлу, який був збережений останнім);

Print - дозволяє надрукувати на папері електричну схему що редагується (при наявності принтера);

Print Setup - друк параметрів - установок даної програми;

Exit - вихід з EWB.

(Все інші команди в даному розділі неідеальні оскільки доповнення SPICE та PCB відсутні.).

Edit - меню редагування. Містить в собі набір команд редагування електронної схеми, розташованої в вікні набору електронної схеми.

Cut - дозволяє вирізати ділянку електронної схеми, вибраної за допомогою блока;

Copy - копіює вибрану ділянку схеми в буфер обміну;

Paste - вставляє раніше скопійований блок із буфера обміну на місце розташування указника миші (якщо він знаходиться в вікні редагування);

Delete - вилучає вибраний елемент або блок схеми що редагується;

Select All - вибирає всі елементи схеми (відмічає їх червоним кольором);

Copy Bits - копіює бітовий образ екрану в буфер обміну;

Примітка. Буфер обміну (в даному випадку) - це спеціальна область пам'яті призначена для обміну інформацією між різними Windows - доповненнями.

Show Clipboard - завантажує додання Clipboard, якщо таке мається. За допомогою нього можна проглянути вміст буфера обміну, а також виконати деякі зміни в ньому.

Circuit - меню побудови.

Містить всі команди, пов'язані з побудовою електричних ланцюгів. **Activate** - запускає процес моделювання роботи електричного ланцюга; **Stop** - зупиняє процес моделювання;

Pause - при зупиняє процес моделювання;

Label... - дозволяє встановити мітку вибраного елемента схеми (приклад: для конденсатора $C1..Cn$, для резистора $R1..Rn,...$, або будь-яку іншу);

Model... - за допомогою цієї команди можна вибрати модель, виділеного електронного елемента, із бібліотеки елементів, що знаходиться в складі цієї прикладної програми;

Value... - дозволяє встановити розмірність вибраного електронного елемента (для резистора - опір в Ом, КОм та МОм; для конденсатора -ємність в мкФ, пФ, нФ, і т.д.);

Zoom - розгортає вибраний електронний прилад із схематичного зображення на схемі в повне вікно;

Rotate - обертає відмічений електронний елемент або групу елементів навколо центральної точки (на кут кратний 90°);

Fault - за допомогою цієї команди можна встановити похибку розрахунку кожного елемента, опір кожного з його виводів.

Subcircuit... - організовує функціональний блок схеми;

Wire Color... - встановлює колір з'єднувальних проводів;

Preferences... - підменю установки конфігурації робочого вікна. Це меню містить в собі такі команди:

Show grid - показує сітку набірного поля;

Use grid - використовує в роботі сітку набірного поля;

Show labels - показує мітки елементів схеми;

Show models - показує моделі елементів схеми;

Show values - показує кількісні параметри елементів схеми (тобто для резисторів - опір, для конденсаторів - ємність тощо);

Analysis Options (аналітичні опції) - за допомогою цих опцій встановлюються параметри, пов'язані з точністю розрахунку (моделювання). Точність моделювання можна встановити в межах 110%... 10%.

Window - меню вікон. Містить всі команди, які обслуговують роботу з вікнами.

Arrange (Ctrl+W) - впорядковує вікна на екрані монітора, встановлює їх в призначений порядок;

Circuit - перехід в вікно редагування;

Description (Ctrl+D) - Перехід в вікно опису (для кожної побудованої схеми, в цьому вікні можна зробити опис її принципу дії, або зробити будь-які примітки, але тільки на англійській мові);

Custom - перехід в вікно функціональних блоків;

Passive - перехід в вікно, яке містить «пасивні» електронні елементи;

Active - перехід в вікно з активними елементами;

Field Effect Transistors - перехід вікно, яке містить польові транзистори;

Control - вікно з електронними ключами та реле;

Hybrid - перехід в вікно з гібридними мікросхемами;

Indicators - перехід в вікно з індикаторами;

Gates - вікно логічних елементів;

Combinational - комбіновані логічні елементи;

Sequential - набір тригерів;

Integrated Circuits - інтегральні мікросхеми;

Help - меню допомоги. Містить в собі команди допомоги користувачеві.

Help (F1) - відкриває вікно допомоги (на англійській мові). Це вікно містить всю необхідну інформацію (підказки) для користування програмою, в короткому змісті.

Help index - видає зміст «книги» допомоги.

Панель контрольно-вимірювальних приладів знаходиться під полем меню робочого вікна програми. В середовищі *Electronics Workbench* задіяні такі прилади:

- 1) цифровий мультиметр;
- 2) функціональний генератор;
- 3) 2-канальний осцилограф;
- 4) вимірювач амплітудно-частотних та фазо-частотних характеристик;
- 5) генератор слів (кодовий генератор);
- 6) 8-канальний логічний аналізатор;
- 7) логічний перетворювач (конвертер).

Кожний електронний прилад має своє умовне графічне позначення.

Білі точки на приладах означають функціональні виводи. За допомогою цих виводів прилади підключаються до елементів схеми. Окрім графічного позначення кожний прилад може бути наданий в розгорнутому вигляді, як діюча модель на екрані комп'ютера.

Для того, щоб розгорнути прилад в діючу модель необхідно: навести указник маніпулятора на його умовне позначення і зробити подвійне натискання його лівої кнопки.

Елементна база *EWB* складається з: джерел струму; індикаторних приладів; комутаційних пристроїв; конденсаторів, елементів індуктивності, напівпровідникових діодів, біполярних транзисторів, польових транзисторів; операційних підсилювачів, цифрових мікросхем.

При побудові електронної схеми більшість дій виконується за допомогою маніпулятора «миша» (далі миша). Указник маніпулятора, на екрані монітора, має вигляд стрілки білого кольору. Для побудови схеми необхідно: вибираючи потрібні групи елементів (за допомогою кнопок, або за допомогою меню *Windows*), перемістити їх із вікна груп в вікно редагування та поєднати між собою, згідно з правилами і принципами електроніки.

Для з'єднання елементів необхідно: встановити указник маніпулятора на вивід одного з елементів (при цьому на виводі з'явиться точка чорного кольору «•»), потім треба натиснувши на ліву кнопку маніпулятора та утримуючи її, перемістити указник до потрібного виводу іншого елемента до появи аналогічної точки на ньому, тільки після цього можна відпустити кнопку маніпулятора (таким чином між виводами з'явиться з'єднувальна лінія - що означає з'єднання виводів провідником).

Кожний елемент можна переміщувати по вікну редагування (для найкращого сприйняття картини), навіть якщо він з'єднаний з іншими елементами.

Колір з'єднувальних ліній для зручності сприйняття можна змінювати. Для

цього необхідно навести указник маніпулятора на потрібну з'єднувальну лінію і зробити подвійне натискання з коротким проміжком часу між натисканнями (двічі «клікнути») лівої кнопки. У вікні, що з'явиться після цього, потрібно вибрати квадрат необхідного кольору (колір квадрата означає колір з'єднувальних проводів). Далі, при моделюванні роботи схеми, сигнал, який проходить через кольоровий провідник, буде відображатися на осцилографі або на аналізаторі логіки кольором, аналогічним кольору провідника.

Після закінчення набору або редагування електронної схеми - можна приступити до моделювання її роботи.

Активізація процесу моделювання відбувається за допомогою натискання кнопки старту моделювання (поз. 5, *рис. 1*) або за допомогою вибору в меню команд **Circuit** команди **Activate**. Після запуску процесу, *Electronics Workbench* починає моделювати роботу схеми в перерахунку на реальний час. Значення рахунку часу моделювання видається в вікні, яке розташовується, після активізації моделювання, ліворуч від кнопки **Activate**. В процесі моделювання можуть з'являтися повідомлення про помилки.

Analysis меню моделювання та аналізу.

Analysis Options... - набір команд для встановлення параметрів моделювання.

ЛАБОРАТОРНА РОБОТА №1

ВИВЧЕННЯ СИСТЕМ ЧИСЛЕННЯ

Мета роботи: вивчити системи числення (СЧ), позиційні і непозиційні системи числення, переклад чисел з однієї системи числення в іншу, вибір системи числення для ЕОМ.

ТЕОРЕТИЧНІ ВІДОМОСТІ

По величині основи одержує назву система числення: десяткова (основа 10), двійкова (основа 2), вісімкова (основа 8) і т.д. Якщо основа числа r перевищує 10, то цифри починаючи з 10, при записі позначають прописними літерами латинського алфавіту: A, B, ..., Z. При цьому цифрі 10 відповідає знак «A», цифрі 11 – знак «B» і т.д.

В табл. 1.1 наведено відповідності чисел між різними СЧ.

При вивченні різних позиційних СЧ слід мати на увазі, що всі ці системи побудовані за одним принципом, так само, як і десяткова СЧ. Тому побудова чисел і виконання арифметичних операцій проводиться за тими ж правилами, що і побудова для десяткової СЧ.

Таблиця 1.1. Відповідності чисел між різними системами числення

<i>r=10</i>	<i>r=2</i>	<i>r=16</i>
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

У практиці рахунку застосовується *десятькова система числення*. У ній будь-яке число представляється в сумі певного числа одиниць, десятків, сотень і т.д. У десятичній системі числення основою є 10; будь-яке ціле число записується як сума значень 10^0 , 10^1 , 10^2 і т. д., кожна з яких може бути взята 1-9 разів.

Наприклад, в записі 343,31 число 3, що стоїть зліва на першому місці, означає число сотень, число 3 перед комою — число одиниць, а число 3 після коми — число десятих долей одиниць. Послідовність цифр 343,31 є скорочено записом виразу

$$343,31 = 3 * 10^2 + 4 * 10^1 + 3 * 10^0 + 3 * 10^{-1} + 1 * 10^{-2}.$$

В обчислювальній техніці найчастіше використовується *двійкова система числення*. Це обумовлено тим, що двійкова система числення найлегше реалізується за допомогою двопозиційних елементів (наприклад, тригерів). Крім того, вона дає можливість спростити виконання арифметичних дій. При двійковій системі числення потрібні всього лише дві цифри 0 і 1 (нуль і одиниця). Основою цієї системи числення є число 2; воно записується двома цифрами: 1 і 0. У двійковій системі числення будь-яке число записується як сума значень 2^0 , 2^1 , 2^2 , 2^3 і т. д., кожна з яких може бути узята 0 або 1 разів.

В загальному вигляді всяке число в двійковій системі числення може бути представлено як

$$A_r = \sum_{k=-m}^n a_k r^k = a_n 2^n + a_{n-1} 2^{n-1} + \dots + a_0 2^0 + a_{-1} 2^{-1} + \dots + a_{-m} 2^{-m}, a_k = 0,1$$

і записано у вигляді послідовності одиниць і нулів.

Для переведення десяткового числа в двійкове необхідно послідовно ділити десяткове число на 2 до тих пір, поки частинне від ділення дасть число 1 або 0. При цьому сукупність залишків (0 або 1) від ділення на 2 утворює шукане.

Наприклад, представимо десяткове число 26 в двійковій системі числення. Переведення числа в двійкову систему числення проводиться за вказаним вище правилом;

Ділення	Залишок
$26:2=13$	0 молодший розряд
$13:2=6$	1
$6:2=3$	0
$3:2=1$	1
$1:2=0$	1 старший розряд

Залишки, одержані при діленні, починаючи з останнього, утворюють двійковий запис десяткового числа 26:

$$26_{(10)} = 11010_{(2)},$$

де 26—десятковий код числа; 11010 – двійковий код числа.

Це відповідає запису:

$$26 = 1 * 2^4 + 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0 = 16 + 8 + 0 + 2 + 0.$$

Переклад чисел з однієї СЧ в іншу може бути здійснений різними способами. Найбільш зручні для ручного рахунку способи розподілу на основу і підсумовування ступенів основи.

В загальному випадку при основі r число представляється у вигляді:

$$A_r = \sum_{k=-m}^n a_k r^k, \quad a_k = 0, 1, 2, \dots, (r-1)$$

і записується як послідовність коефіцієнтів a_k .

Переклад чисел за способом розподілу на основу базується на наступному.

Правило. Для перекладу цілого числа в систему числення з основою r ділять задане число на основу цієї системи. Залишок від розподілу є молодшим розрядом цього числа. Потім приватне знову ділять на основу, і залишок дає другий розряд. Так повторюють до тих пір, поки не отримають приватне, рівне нулю. Останній залишок дає старший розряд числа в системі числення з основою r . Розподіл потрібно проводити в тій системі числення, в якій задано число.

$$A_{10} = 37.$$

Приклад. Вимагається записати в двійковій системі числення число.

Вказані в правилі дії запишемо у вигляді таблиці:

Ділення	Частне	Залишок
37:2	18	1 молодший розряд
18:2	9	0
9:2	4	1
4:2	2	0
2:2	1	0
1:2	0	1 старший розряд

$$A_2 = 100101.$$

Як було вказане, залишок при першому розподілі дає молодший розряд, а при останньому – старший.

Переклад дробових чисел здійснюється шляхом множення на основу.

Правило. Для перекладу правильного дробу в систему числення з основою r , необхідно помножити дріб на основу і виділити цілу частину; отримана цифра в цілій частині буде першою цифрою (після коми) шуканого числа. Помноживши дробову частину отриманого числа, що залишилася, знову на основу і узявши цілу частину нового числа, знайдемо другу цифру. Множення слід повторювати до тих пір, поки не буде отримана кількість розрядів шуканого числа, що задовольняє необхідній точності представлення цього числа. Множення проводити в тій системі числення, в якій задано число.

Приклад. Вимагається записати в двійковій системі числення дріб $A_{10} = 0,732$ з точністю $1/32$.

При заданій точності вимагається визначити чотири розряди двійкового дробу. Запис дій проведемо у вигляді:

$0,732 * 2 = 1,464$
$0,464 * 2 = 0,928$
$0,928 * 2 = 1,856$
$0,856 * 2 = 1,712$

Для запису двійкового дробу необхідно записати цифри, що стоять в числі зліва від запису. А саме:

$$A_2 = 0.1011$$

При перекладі чисел з системи числення з якою-небудь основою в десяткову систему зручно використовувати спосіб підсумовування ступенів основи.

Правило. Для перекладу чисел з однієї системи числення з основою r в десяткову систему, потрібно перемножити розряди заданого числа на відповідні їм ступені r і скласти.

Приклад. Перевести число $A=100101$ з двійкової системи числення в десяткову.

$$A_{10} = 2^5 + 2^2 + 2^0 = 37$$

Старшому розряду відповідає п'ятий ступінь двійки.

Десяткові числа від 0 до 10 в двійковому коді записуються таким чином:

0→0000;	4→0100;	8→1000;
1→0001;	5→0101;	9→1001;
2→0010;	6→0110;	10→1010.
3→0011;	7→0111;	

У цифрових обчислювальних машинах окрім двійкової застосовуються і інші системи числення. Так, наприклад, при програмуванні використовується вісімкова система числення, а для введення і виведення даних — двійково-десяткова.

У вісімковій системі числення основою є число 8. Будь-яке число у вісімковій системі числення може бути представлено як сума чисел $8^3, 8^2, 8^1, 8^0, 8^{-1}, 8^{-2}, 8^{-3}$..., кожне з яких може бути узятé 1—7 разів.

Аналогічно запису цілого числа в десятковій позиційній системі числення у вісімковій системі числення представляється у вигляді

$$\overline{abcd}_8 = a \cdot 8^3 + b \cdot 8^2 + c \cdot 8^1 + d \cdot 8^0$$

де a, b, c, d менше 8.

Наприклад: $26_8 = 2 \cdot 8^1 + 6 \cdot 8^0 = 22_{10}$.

Отже, число 22_{10} є десятковий вид вісімкового числа 26.

Десяткове число 22 в двійковому коді має наступний запис:

$$22_{(10)} = 10110.$$

Якщо розглянути дане двійкове число, то виявляється, що перші дві цифри означають 2, а інші три — 6, тобто це не що інше, як вісімковий код, десяткового числа 22.

Отже, для переведення вісімкового числа в двійкове необхідно кожен цифру вісімкового числа окремо представити в двійковому коді.

Наприклад:

$$15_{(8)} = (001)(101)_{(2)},$$

$$\text{де } (001)_2 = (1)_8; (101)_2 = (5)_8;$$

$$23_{(8)} = (010)(011)_{(2)},$$

$$\text{де } (010)_2 = (2)_8; (011)_2 = (3)_8.$$

Таким чином, одному розряду вісімкової системи відповідають три розряди двійкової системи. Вісімкова система зручніша тим, що вона вимагає менше число розрядів, ніж двійкова. Тому вісімкова система застосовується при складанні програм для кодування адрес і команд. Після того, як програма складена, вона переводиться в двійкову систему числення і потім вводиться в машину.

ПРИКЛАДИ ПЕРЕВЕДЕННЯ ЧИСЕЛ В РІЗНІ СИСТЕМИ ЧИСЛЕННЯ

Приклад 1. Перевести число 19_{10} в шістнадцяткову СЧ.

$$\begin{array}{r|l} 19 & 16 \\ \hline 16 & 1 \\ \hline 3 & \end{array}$$

Маємо: $19_{10} = 13_{16}$.

Приклад 2. Перевести число 123_{10} в шістнадцяткову СЧ.

$$\begin{array}{r|l} 123 & 16 \\ \hline 112 & 7 \\ \hline 11 & \end{array}$$

..... Маємо: $123_{10} = 7B_{16}$.

Приклад 3. Перевести число 13_{16} в десяткову СЧ.

Маємо: $13_{16} = 1 \cdot 16^1 + 3 \cdot 16^0 = 16 + 3 = 19$.

Правила перекладання чисел з двійкової СЧ в шістнадцяткову.

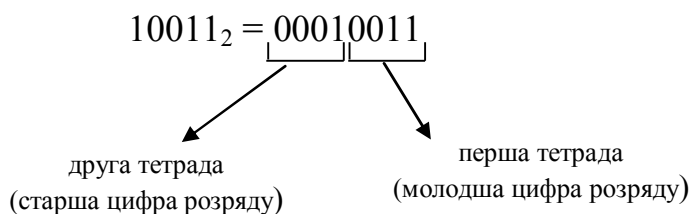
1. Початкове число розбивається на тетради (тобто на 4 цифри), починаючи з молодших розрядів. Якщо кількість цифр початкового двійкового числа не кратне 4, воно доповнюється зліва незначущими нулями до досягнення кратності 4;

2. Кожна тетрада замінюється відповідною шістнадцятковою цифрою у відповідності з табл. 1.1.

Приклад 4. Перевести число 10011_2 в шістнадцяткову СЧ.

Оскільки в початковому двійковому числі кількість цифр не кратна 4, доповнюємо його зліва незначущими нулями до досягнення кратності 4 числа цифр.

Маємо:



У відповідності з табл. 1.1: $0011_2 = 11_2 = 3_{16}$ та $0001_2 = 1_2 = 1_{16}$.
Тоді $10011_2 = 13_{16}$.

Правила перекладання з шістнадцяткової СЧ в двійкову.

1. Кожна цифра початкового числа замінюється тетрадою двійкових цифр у відповідності з табл. 1.1. Якщо в таблиці двійкове число має менше 4 цифр, то воно доповнюється *зліва* незначущими нулями до тетради;
2. Незначущі нулі в результуючому числі відкидаються.

Приклад 5. *Перевести число 13_{16} в двійкову СЧ.*

Згідно табл. 1.1 маємо: $1_{16} = 1_2$

після доповнення незначущими нулями, маємо: $1_2 = 0001_2$;

$$3_{16} = 11_2$$

після доповнення незначущими нулями, маємо: $11_2 = 0011_2$.

Тоді $13_{16} = 00010011_2$, після відкидання незначущих нулів маємо: $13_{16} = 10011_2$.

Правила перекладання правильних дробів

З десятикової СЧ в двійкову та шістнадцяткову.

1. Початковий дріб спочатку помножують на основу СЧ, в яку переводиться число (2 або 16);
2. В отриманому добутку цілу частину перетворюють у цифру (у відповідності з табл. 1.1) та відкидають – вона є старшою цифрою отриманого дробу;
3. Частина дробу, яка залишилась, знову множать на основу СЧ, в яку переводять (у відповідності з кроками 1 та 2);
4. Процедура множення триває до тих пір, поки не буде отримано нульовий результат в дробовій частині добутку або не буде досягнуто необхідну кількість цифр в результаті.
5. Формують результат: послідовно відкинуті на кроці 2 цифри складають дробову частину результату, причому в порядку зменшення старшинства.

Приклад 6. *Перевести число 0,847 в двійкову СЧ. Перевод виконувати до 4 значущих цифр після коми.*

Маємо:

$$\begin{array}{rcl} \begin{array}{r} 0,847 \\ \cdot 2 \\ \hline 1,694 \end{array} & \longrightarrow & \begin{array}{r} 0,694 \\ \cdot 2 \\ \hline 1,388 \end{array} \\ & & \longrightarrow & \begin{array}{r} 0,388 \\ \cdot 2 \\ \hline 0,776 \end{array} \\ & & & \longrightarrow & \begin{array}{r} 0,776 \\ \cdot 2 \\ \hline 1,552 \end{array} \text{ і т.д.} \end{array}$$

0,1101

В даному прикладі процедуру переводу було перервано на 4 кроці, оскільки отримано необхідну кількість розрядів результату. Очевидно, це призвело до втрати ряду цифр.

Приклад 7. Перевести число 0,847 в шістнадцяткову СЧ. Перевод виконувати до 3 значущих цифр.

Маємо:

$$\begin{array}{rcl}
 \begin{array}{r} 0,847 \\ \cdot \quad 16 \\ \hline 13,532 \end{array} & \longrightarrow & \begin{array}{r} 0,532 \\ \cdot \quad 16 \\ \hline 8,832 \end{array} \\
 0,D8D & & \begin{array}{r} 0,832 \\ \cdot \quad 16 \\ \hline 13,312 \end{array} \text{ і т.д.}
 \end{array}$$

З двійкової та шістнадцяткової СЧ в десяткову.

В цьому випадку розраховується повне значення числа, причому коефіцієнт a_i приймають десяткові значення у відповідності з СЧ.

Приклад 8. Перевести з двійкової СЧ в десяткову число $0,1101_2$.

Маємо:
$$\begin{aligned}
 0,1101_2 &= 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 0 \cdot 2^{-3} + 1 \cdot 2^{-4} = 1 \cdot \frac{1}{2^1} + 1 \cdot \frac{1}{2^2} + 0 \cdot \frac{1}{2^3} + 1 \cdot \frac{1}{2^4} = \\
 &= 0.5 + 0.25 + 0 + 0.0625 = 0.8125
 \end{aligned}$$

Розходження отриманого результату з початковим (0,847) викликано тим, що процедуру переводу в двійковий дріб було перервано.

Приклад 9. Перевести з шістнадцяткової СЧ в десяткову число $0,D8D_{16}$.

Маємо:
$$\begin{aligned}
 0,D8D_{16} &= 13 \cdot 16^{-1} + 8 \cdot 16^{-2} + 13 \cdot 16^{-3} = 13 \cdot \frac{1}{16^1} + 8 \cdot \frac{1}{16^2} + 13 \cdot \frac{1}{16^3} = \\
 &= 13 \cdot 0.0625 + 8 \cdot 0.003906 + 13 \cdot 0.000244 = 0.84692
 \end{aligned}$$

Розходження отриманого результату з початковим (0,847) викликано тим, що процедуру переводу в шістнадцятковий дріб було перервано.

З двійкової СЧ в шістнадцяткову.

1. Початковий дріб необхідно поділити на тетради, починаючи з десяткової крапки праворуч. Якщо кількість цифр дробової частини не кратне 4, то вона доповнюється праворуч незначущими нулями до досягнення кратності 4;

2. Кожна тетрада замінюється шістнадцятковою цифрою, у відповідності з табл. 1.1.

Приклад 10. *Перевести з двійкової СЧ в шістнадцяткову число $0,0010101_2$.*

Оскільки кількість цифр дрібної частини не кратне 4, то додамо праворуч незначущий нуль: $0,0010101_2 = 0,00101010_2$.

У відповідності з табл. 1.1: $0010_2 = 10_2 = 2_{16}$ та $1010_2 = A_{16}$.

Тоді маємо: $0,0010101_2 = 0,2A_{16}$.

З шістнадцяткової СЧ в двійкову.

1. Кожна цифра початкового дробу замінюється тетрадою двійкових цифр у відповідності з табл. 1.1;

2. Незначущі нулі відкидаються.

Приклад 11. *Перевести з шістнадцяткової СЧ в двійкову число $0,2A_{16}$.*

Згідно таблиці маємо: $2_{16} = 0010_2$ та $A_{16} = 1010_2$.

Тоді $0,2A_{16} = 0,00101010_2$. Відкидаємо незначущий нуль та отримуємо кінцевий результат: $0,2A_{16} = 0,0010101_2$.

Правило перекладання дробових чисел

Окремо переводиться ціла частина числа, окремо – дрібна. Результати складають.

Приклад 12. *Перевести з десяткової СЧ в шістнадцяткову числа 19,847. Перевод виконувати до 3 значущих цифр після коми.*

Представимо початкове число як суму цілого числа та правильного дробу: $19,847 = 19 + 0,847$.

З попереднього прикладу (1) $19_{10} = 13_{16}$, а у відповідності до прикладу (7) $0,847_{10} = 0,D8D_{16}$. Тоді маємо: $19_{10} + 0,847_{10} = 13_{16} + 0,D8D_{16} = 13,D8D_{16}$.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Згідно варіанту виконати наступні операції над числами:

1. Перевести число з десяткової СЧ в шістнадцяткову, згідно варіанту (табл. 1.3).

2. Перевести число з шістнадцяткової СЧ в десяткову, згідно варіанту (табл. 1.4).

3. Перевести число з двійкової СЧ в шістнадцяткову, згідно варіанту (табл. 1.5).

4. Перевести число з шістнадцяткової СЧ в двійкову, згідно варіанту (табл. 1.4).

5. Перевести правильний дріб з десяткової СЧ в двійкову та шістнадцяткову, згідно варіанту (табл. 1.6). Перевод виконувати до 4 значущих цифр після коми.

6. Перевести правильний дріб з двійкової СЧ в десяткову, згідно варіанту (табл. 1.7).

7. Перевести правильний дріб з шістнадцяткової СЧ в десяткову, згідно варіанту (табл. 1.8).

8. Перевести правильний дріб з двійкової СЧ в шістнадцяткову, згідно варіанту (табл. 1.7).

9. Перевести правильний дріб з шістнадцяткової СЧ в двійкову, згідно варіанту (табл. 1.8).

10. Перевести дрібне число з десяткової СЧ в шістнадцяткову, згідно варіанту (табл. 1.9). Перевод виконувати до 3 значущих цифр після коми.

Таблиця 1.3

Варіант	1	2	3	4	5	6	7	8	9	10
Число	23	54	41	100	33	75	58	68	112	202

Таблиця 1.4

Варіант	1	2	3	4	5	6	7	8	9	10
Число	15	22	33	61	42	77	82	40	25	34

Таблиця 1.5

Варіант	1	2	3	4	5	6	7	8	9	10
Число	01110	110010	001100	01010	11001	0101101	110110	001011	11100	000111

Таблиця 1.6

Варіант	1	2	3	4	5	6	7	8	9	10
Число	0,501	0,633	0,805	0,303	0,444	0,225	0,106	0,777	0,911	0,666

Таблиця 1.7

Варіант	1	2	3	4	5	6	7	8	9	10
Число	0,110	0,0011	0,1010	0,1100	0,0101	0,1111	0,0111	0,1110	0,1000	0,1101

Таблиця 1.8

Варіант	1	2	3	4	5	6	7	8	9	10
Число	0.A7E	0.3DC	0.C8F	0.B3D	0.2EA	0.5DF	0.DF3	0.B8A	0.EC6	0.D7B

Таблиця 1.9

Варіант	1	2	3	4	5	6	7	8	9	10
Число	20,404	33,02	150,843	100,115	44,201	22,444	18,041	51,002	22,761	31,052

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
2. Результати виконання завдання, згідно варіанту.

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

ЛАБОРАТОРНА РОБОТА №2

АРИФМЕТИЧНІ ОСНОВИ ПОБУДОВИ ЕЛЕМЕНТІВ І ВУЗЛІВ ОБЧИСЛЮВАЛЬНИХ МАШИН І СИСТЕМ.

Мета роботи: вивчити способи представлення чисел з фіксованою і плаваючою комою, форми представлення чисел в ЕОМ з фіксованою і плаваючою комою, основні арифметичні операції над числами в двійковій системі числення.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Дана тема знайомить з арифметичними основами ЕОМ, тобто з тим, як представляються числа в машині і як виконуються арифметичні операції з кодами в двійковій системі числення. Подальший матеріал, а саме: побудова вузлів ЕОМ; арифметичних, запам'ятовуючих і управляючих пристроїв ЕОМ – заснований на матеріалі даної теми і не може бути зрозумілим при його недостатньому відпрацюванні. Всі питання теми повинні бути відпрацьовані не тільки теоретично. Особливо це торкається виконання арифметичних операцій в двійковій системі числення і з числами в нормальній формі.

Арифметичні дії над двійковими числами проводяться за тими ж правилами, що і над десятковими:

Складання	Віднімання	Множення
$0 + 0 = 0;$	$0 - 0 = 0;$	$0 * 0 = 0;$
$0 + 1 = 1;$	$0 - 1 = 0;$	$0 * 1 = 0;$
$1 + 0 = 1;$	$1 - 1 = 0;$	$1 * 0 = 0;$
$1 + 1 = 10.$	$10 - 1 = 1.$	$1 * 1 = 1.$

Приклади арифметичних дій над числами в десятковій і двійковій системах числення приведені табл. 2.1.

У двійковій системі числення, так само як і в десятковій і двійковій, можуть бути записані дробові і змішані числа. Ціла і дробова частини в двійковій і в десятковій системах числення відділяються комою.

Таблиця 2.1. – Арифметичні дії над числами в двійковій і десятковій системах числення

Система	Складання	Віднімання	Множення	Ділення	
Десяткова	8	8	8	8	4
	+ 4	– 4	*4	- 8	2
	12	4	32	0	
Двійкова	1000	1000	1000	1000	100
	+ 100	- 100	* 100	-1000	10
	1100		0000		
			0000		
			+ 1000		
			100 000		
	1100	100	0000		
			0000		
			+1000		
			100 000		

У десятковій системі числення в дробовій частині підсумовуються десятки, соті, тисячні і т.д. долі числа, а в двійковій – половинні, четвертні, восьмі, шістнадцяті і т.д.

Наприклад: $101,011 = 1*2^2 + 0*2^1 + 1*2^0 + 0*2^{-1} + 1*2^{-2} + 1*2^{-3}$.

Отже при використуванні двійкової системи числення для представлення в ЕОМ одного розряду числа необхідно мати фізичний елемент з двома стійкими станами. Один стан відповідає нулю, а інший – одиниці. Для того, щоб представити в ЕОМ число, необхідно мати ряд двійкових елементів, для кожного розряду свій елемент.

Сукупність двійкових елементів, необхідних для представлення одного числа, утворює розрядну сітку машини. Залежно від того, як використовується розрядна сітка, розрізняють два способи представлення чисел: в природній і в нормальній формі.

При природній формі представлення чисел в розрядній сітці машини відводиться певна кількість розрядів для цілої частини, для дробової частини і один розряд (іноді два) для знака числа. Звичайно кому розташовують після знакового розряду, і отже, всі числа представляються правильними дробами.

Схематичне зображення розрядної сітки з комою, фіксованою після знакового розряду, і представлення чисел:

+0.10010111 (а) і

-0.01011001 (б) приведено на рисунку 2.1:

а) зн. Числова частина (дріб)

0	1	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---

б) зн. Числова частина (дріб)

1	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---

Рис. 2.1. Представлення чисел з фіксованою комою

Діапазон чисел, які можна представити в машині при природній формі, визначається виразом:

$$2^{-n} \leq N \leq 1 - 2^{-n},$$

де n – кількість розрядів числової частини.

Обмежене число розрядів розрядної сітки приводить до погрішності в представленні чисел.

Для найбільшого можливого числа N_{max} відносна похибка:

$$\delta_{\max_1} = \frac{\Delta_{\max}}{N_{\max}} = \frac{0.5 * 2^{-n}}{1 - 2^{-n}} \cong 0.5 * 2^{-n}.$$

Для якнайменшого уявного числа відносна похибка:

$$\delta_{\max_2} = \frac{0.5 * 2^{-n}}{2^{-n}} = 0.5$$

В нормальній формі число представляється у вигляді:

$$N=M*10^p,$$

де M - мантиса.

10 - основа системи числення, записане в цій с-мі

p - порядок.

$$0,110101 \cdot 10^{+100} (a) \qquad 0,101100 \cdot 10^{-10} (b)$$

В розрядній сітці машини відводять розряди для мантиси, порядку, знака числа і знака порядку, схематичне зображення розрядної сітки і представлення двійкових чисел приведено на рисунку 2.2.

 $A_7 :$

3H. 3H.

Пір.	числа.
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15
16	16
17	17
18	18
19	19
20	20
21	21
22	22
23	23
24	24
25	25
26	26
27	27
28	28
29	29
30	30
31	31
32	32
33	33
34	34
35	35
36	36
37	37
38	38
39	39
40	40
41	41
42	42
43	43
44	44
45	45
46	46
47	47
48	48
49	49
50	50
51	51
52	52
53	53
54	54
55	55
56	56
57	57
58	58
59	59
60	60
61	61
62	62
63	63
64	64
65	65
66	66
67	67
68	68
69	69
70	70
71	71
72	72
73	73
74	74
75	75
76	76
77	77
78	78
79	79
80	80
81	81
82	82
83	83
84	84
85	85
86	86
87	87
88	88
89	89
90	90
91	91
92	92
93	93
94	94
95	95
96	96
97	97
98	98
99	99
100	100

0	1	0	0	0	1	1	0	1	0	1
---	---	---	---	---	---	---	---	---	---	---

порядок мантиса

 $B_2 :$

3H. 3H.

Пір.		числа.								
1	0	1	0	0	1	0	1	1	0	0
порядок		мантиса								

Рис. 2.2. Представлення чисел з плаваючою комою.

Діапазон чисел, які можна представити в машині при нормальній формі, визначається виразом:

$$2^{-2^p} \leq N \leq (1 - 2^{-m}) * 2^{(2^p - 1)}$$

де P – кількість розрядів порядку;

M – кількість розрядів мантиси.

Виконання арифметичних операцій з цілими числами в нормальній формі має певні особливості.

При складанні і відніманні необхідно провести вирівнювання, яке полягає в тому, що меншому числу привласнюється порядок більшого числа. Після цього проводиться дія з мантисою, а потім нормалізація результатів. В результаті повинна бути отриманий мантиса у вигляді правильного дробу із значущим старшим розрядом. При множенні і розподілі проводиться окремо дія з мантисами і окремо з порядками, а потім нормалізується результат.

Для представлення двійкових чисел в ЕОМ застосовують пряму, зворотну і додаткову коди. У всіх цих кодах передбачається додатковий розряд для представлення знака числа, причому знак (+) кодується цифрою 0, а знак (-) - цифрою 1. При допомозі цих кодів операція віднімання або складання алгебри зводиться до арифметичного складання. В результаті спрощуються арифметичні пристрої ЕОМ.

Позитивні числа в прямому, зворотному і додаткових кодах мають один і той же вигляд, негативні - різний.

Щоб представити двійкове негативне число в зворотному коді, треба поставити в знаковий розряд одиницю, а в усіх інших розрядах замінити одиниці нулями, а нулі – одиницями.

Наприклад, число:

$$(-3.5)_{10} - (-11.1)_2,$$

Представлене в зворотному коді, має вигляд: $(100.0)_{\text{обр}}$, де перша одиниця в знаковому розряді указує на те, що число негативне.

При записі числа в додатковому коді ставлять одиницю в розряд знака, а цифрову частину числа замінюють доповненням модуля числа до цілої одиниці. Наприклад, число $(-0.01)_2$ в додатковому коді матиме вид $(1.11)_{\text{доп}}$. Тут перша одиниця указує на те, що число негативне.

При складанні алгебри двох цілих чисел з використанням зворотного коду позитивні доданки представляються в прямому коді, а негативні в зворотному, проводиться арифметичне підсумовування цих кодів, включаючи розряди

знаків. При виникненні перенесення з розряду знака одиниця перенесення додається до молодшого розряду суми кодів (таке перенесення називають круговим або циклічним). В результаті виходить сума алгебри в прямому коді, якщо ця сума позитивна, і в зворотному коді, якщо вона негативна.

Додавання двійкових чисел

Операцію додавання починають з обробки найменших значень цифр в останній позиції праворуч. Якщо результат додавання найменших значущих цифр двох доданків не вміщається у відповідному розряді результату, то відбувається перенесення. Цифра, яку переносять в сусідній зліва розряд, додається до вмісту останнього. Перенесення виникає, якщо результат додавання цифр однойменних розрядів більше 1.

Приклад 1. Додати наступні числа $01100011 + 01011111$.

	1 1 1 1 1 1	- перенесення
	0 1 1 0 0 0 1 1	- доданок 1
+		
	0 1 0 1 1 1 1 1	- доданок 2
	<u>1 1 0 0 0 0 1 0</u>	- сума

Віднімання двійкових чисел

Якщо вміст розряду зменшуваного менше вмісту однойменного розряду від'ємника, то має місце позика. Це відбувається в тому випадку, коли від 0 віднімають 1.

Приклад 2. Від числа 01101101 відняти число 00110001 .

	1 1	- позика
	0 1 1 0 1 1 0 1	- зменшуване
-		
	0 0 1 1 0 0 0 1	- від'ємник
	<u>0 0 1 1 1 1 0 0</u>	- різниця

Множення двійкових чисел

При множенні одного числа на інше одне число називається множене, а інше – множитком. Множення виконують порозрядно. Часто виникає необхідність перенесення в наступний розряд. По завершенню множення множеного на значення молодшого розряду множника отримують перший частковий добуток. В результаті множення множеного на значення наступного розряду множника формується другий частковий добуток. Для отримання остаточного добутку, зміщені відносно один одного часткові добутки додають з

урахуванням перенесення.

Приклад 3. Помножити наступні числа $11101 * 011$.

$$\begin{array}{rcl}
 & 11101 & - \text{множене} \\
 * & \underline{011} & - \text{множник} \\
 & 11101 & - \text{перший частковий добуток} \\
 + & & \\
 & 11101 & - \text{другий частковий добуток} \\
 + & & \\
 & 00000 & - \text{третій частковий добуток} \\
 \hline
 & 1010111 & - \text{остаточний добуток}
 \end{array}$$

Даний спосіб множення двійкових чисел має назву *множення шляхом зсуву та додавання*. Основні правила цього способу:

1. Формування першого часткового добутку. Якщо значення молодшого розряду множника дорівнює 0, то і результат дорівнює 0, якщо значення цього розряду дорівнює 1, то результат є копією множеного.

2. Правило зсуву. При використанні чергового розряду множника для формування часткового добутку виконують зсув множеного на один розряд ліворуч.

3. Правило додавання. Кожний раз, коли значення розряду множника дорівнює 1, до результату необхідно додати множене, яке розташоване в позиції, що визначена правилом зсуву.

4. Визначення остаточного добутку. Він є результатом виконання всіх операцій зсуву та додавання.

ЗАВДАННЯ ДО ЛАБОРАТОРНОЇ РОБОТИ

Виконати основні арифметичні операції (додавання, віднімання та множення) згідно варіанту.

Варіанти

Варіант	Додавання		Віднімання		Множення	
	доданок 1	доданок 2	зменшуване	від'ємник	множник	множене
1	001101	101111	101111	100111	000101	001101
2	110011	110111	110111	011001	001111	010110
3	010110	100111	100111	101111	110011	101110
4	000101	011001	011001	110111	010110	100111
5	101110	001101	001111	010110	110111	101111
6	001111	110011	101110	000101	001101	110111

Варіант	Додавання		Віднімання		Множення	
	доданок 1	доданок 2	зменшуване	від'ємник	множник	множене
7	011001	010110	000101	001101	101111	011001
8	100111	000101	010110	110011	011001	001111
9	110111	101110	110011	001111	100111	000101
10	101111	001111	001101	101110	101110	110011

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
2. Результати виконання завдання, згідно варіанту.

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

ЛАБОРАТОРНА РОБОТА №3

ВИВЧЕННЯ ЛОГІЧНИХ ЕЛЕМЕНТІВ

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

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).

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

Кількість рядків в таблиці – це кількість можливих наборів значень аргументів. Воно дорівнює 2^n , де n – кількість змінних.

Не зупиняючись на математичній і елементній базі ЛЕ, досить докладно представлених у роботах [2-5], розглянемо коротко призначення логічних елементів, використовуваних у конструкторі, їхнє умовне графічне позначення,

функціональний електричний аналог кожного з них і відповідну йому таблицю істинності.

Попередньо нагадаємо, що для ДЕ з n входами можна отримати 2^n вхідних комбінацій. Ми ж зупинимо свою увагу на двовходових ЛЕ ($n=2$), використовуваних у конструкторі. При цьому позначимо через X_1 та X_2 вхідні логічні змінні, які приймають значення 0 або 1, а вихідне логічне вираження, що є їхньою функцією, через Y .

Елемент «І»

На виході елемента «І» логічна 1 з'явиться тільки в тому випадку, якщо усі входи одночасно знаходяться в стані логічної 1. Інші (з 2^n) комбінації входів приводять до утворення на виході логічного 0. Умовне графічне позначення логічного елемента «І» наведене на *рис. 3.1*.

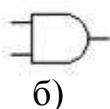
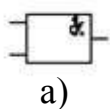
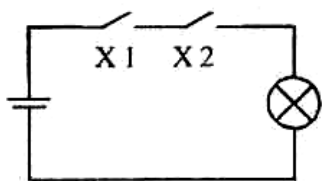


Рис. 3.1. Умовне графічне позначення логічного елемента «І»:
а) вітчизняний та європейський стандарт (використовується в EWB);
б) північноамериканський стандарт (використовується в EWB)

Логічний елемент «І» функціонує подібно послідовній електричній схемі, наведеній на *рис. 3.2*, у якій лампа засвітиться (логічний вихід дорівнюватиме 1) тільки в тому випадку, якщо вимикачі X_1 та X_2 одночасно замкнуті (логічні входи дорівнюють 1). Очевидно, що при будь-якій порушенні цієї умови (X_1 , X_2 або обидва ключі розімкнені, чому відповідає їхній логічний стан 0) лампа буде погашеною (стан 0).



X_1	X_2	Y
0	0	0
0	1	0
1	0	0
1	1	1

Рис. 3.2. Фізична інтерпретація і таблиця істинності логічного елемента «І»

Як впливає з таблиці істинності, такий елемент реалізує операцію «логічного множення» (кон'юнкцію) вхідних перемінних, котра може бути виражена логічним рівнянням вигляду:

$$Y = X_1 \times X_2 \quad \text{або} \quad Y = X_1 \cap X_2$$

Елемент «АБО»

На виході елемента «АБО» з'являється логічна 1, якщо хоча б один із входів (чи усі входи) знаходиться в стані логічної 1. Очевидно, що логічний 0 на виході елемента «АБО» з'явиться тільки в тому випадку, коли стану усіх входів одночасно рівні логічному 0. Умовне графічне позначення логічного елемента «АБО» наведено на рис. 3.3.

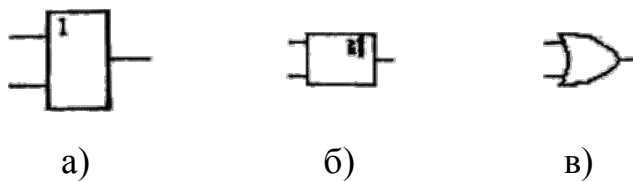


Рис. 3.3. Умовне графічне позначення логічного елемента «АБО»:

- а) вітчизняний стандарт;
- б) європейський стандарт (використовується в EWB);
- в) північноамериканський стандарт (використовується в EWB)

Логічний елемент «АБО» функціонує подібно схемі, показаній нижче на рис. 3.4, у якій лампа горітиме (логічний вихід 1) при будь-якій сполученні замкнених (логічні входи 1) вимикачів: або X1, або X2, чи обох.

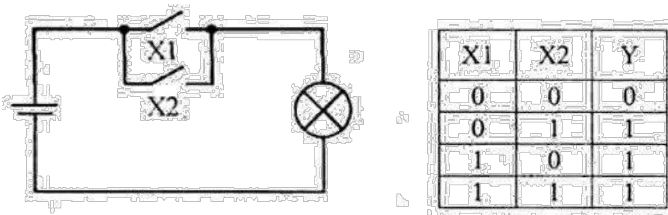


Рис. 3.4. Фізична інтерпретація і таблиця істинності логічного елемента «АБО»

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* вхідних змінних, котра може бути виражена логічним рівнянням вигляду:

$$Y = X1 + X2 \quad \text{або} \quad Y = X1 \cup X2$$

Елемент «НЕ» (інвертор)

Умовне графічне позначення логічного елемента «НЕ» наведено на рис. 3.5. На виході інвертора логічний рівень завжди протилежний вхідному.

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

$$Y = \bar{X}1$$

Функціональний аналог елемента «НЕ» може бути представлений схемою, приведеної на рис. 3.6, у якій лампа займеться (логічний вихід 1) при розмиканні вимикача X1 (логічний вхід 0) і згасне (логічний 0) при його замиканні (логічна 1).

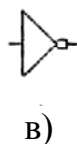
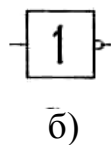
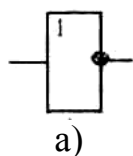
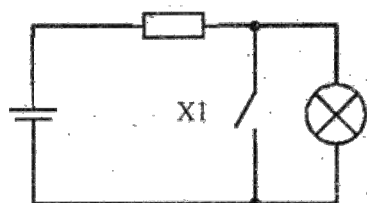


Рис. 3.5. Умовне графічне позначення логічного елемента «НЕ»:
 а) вітчизняний стандарт;
 б) європейський стандарт (використовується в EWB);
 в) північноамериканський стандарт (використовується в EWB)



X1	Y
0	1
1	0

Рис. 3.6. Фізична інтерпретація і таблиця істинності логічного елемента «НЕ»

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

Нижче наведено основні логічні функції.

1. Функція константа 0, $Y=0$. Технічна реалізація такої функції – з'єднання виводу Y з загальною шиною з нульовим потенціалом.

Ті функції константа 0 має вигляд:

X	Y
0	0
1	0

2. Функція повторення – $Y=f(X)=X$. Технічна реалізація – з'єднання між собою виводів X та Y.

Ті функції повторення має вигляд:

X	Y
0	0
1	1

3. Функція константа 1, $Y=1$. Технічна реалізація – з'єднання виводу Y з джерелом живлення.

Ті функції константа 1 має вигляд:

X	Y
0	1
1	1

4. Функція стрілка Пірса (АБО-НЕ): $Y = \text{NOT}(X1 + X2)$.

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

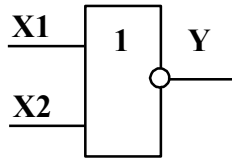


Рис. 3.7. Умовне графічне позначення і таблиця істинності логічного елемента АБО-НЕ

5. Функція штрих Шиффера (І-НЕ): $Y = \text{NOT}(X1 \cdot X2)$.

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

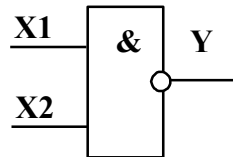


Рис. 3.8. Умовне графічне позначення і таблиця істинності логічного елемента І-НЕ

6. Еквівалентність (рівнозначність): $Y = X1 \sim X2$.

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

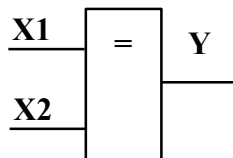


Рис. 3.9. Умовне графічне позначення і таблиця істинності логічного елемента рівнозначність

Правило еквівалентності з будь-якою кількістю входів звучить так:

На виході буде:

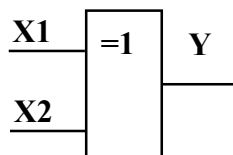
«1» тоді і тільки тоді, коли на вході діє парна кількість «1»,

«0» тоді і тільки тоді, коли на вході діє непарна кількість «1».

7. ВИКЛЮЧАЮЧЕ АБО (XNOR): $Y = \overline{X1} \cdot X2 \vee X1 \cdot \overline{X2} \vee \overline{X1} \cdot \overline{X2}$.

Ті має такий же вигляд, як і у попередньої функції.

На схемах функція ВИКЛЮЧАЮЧЕ АБО має наступний вигляд:



8. Сума по модулю (ВИКЛЮЧАЮЧЕ АБО, нерівнозначність): $Y = X1 \oplus X2$.

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

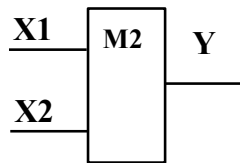


Рис. 3.10. Умовне графічне позначення і таблиця істинності логічного елемента **ВИКЛЮЧАЮЧЕ АБО**

Правило для суми по модулю з будь-якою кількістю входів звучить так:
На виході буде:

«1» тоді і тільки тоді, коли на входах діє непарна кількість «1»,
«0» тоді і тільки тоді, коли на входах діє парна кількість «1».

Ці три логічних елемента (ЕКВІВАЛЕНТНІСТЬ, ВИКЛЮЧАЮЧИ АБО та СУМА ПО МОДУЛЮ) повністю взаємо заміняються один одним у випадку двох вхідних змінних та мають ідентичні таблиці істинності.

9. Нееквівалентність, нерівнозначність.

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

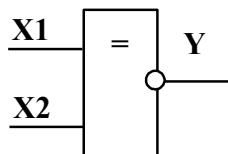


Рис. 3.11. Умовне графічне позначення і таблиця істинності логічного елемента **нерівнозначність**

10. Імплікація від X1 до X2 (інверсія декременту).

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

Ті цієї функції має вигляд:

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

Правило для інверсії декременту звучить так:

На виході буде:

«0» тоді і тільки тоді, коли на «X2» менше «X1»,
«1» тоді і тільки тоді, коли на «X2» більше або дорівнює «X1».

11. Імплікація від X2 до X1 (інверсія інкременту).

Інкремент – (збільшення) операція збільшення на одиницю або ж, в деяких випадках, на вимірювану величину. Зворотною операцією називають декремент (зменшення).

Ті цієї функції має вигляд:

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

Правило для інверсії звучить так:

На виході буде:

«0» тоді і тільки тоді, коли на «X2» більше «X1»,

«1» тоді і тільки тоді, коли на «X2» менше або дорівнює «X1».

12. Декремент. Заборона імплікації по X2.

Інверсія імплікації від X1 до X2. Ті цієї функції має вигляд:

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

Правило для інверсії звучить так:

На виході буде:

«0» тоді і тільки тоді, коли на «X1» менше або дорівнює «X2»,

«1» тоді і тільки тоді, коли на «X1» більше «X2».

13. Інкремент. Заборона імплікації по X1.

Інверсія імплікації від X2 до X1.

Ті цієї функції має вигляд:

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

Правило для інверсії звучить так:

На виході буде:

«0» тоді і тільки тоді, коли на «X2» більше «X1»,

«1» тоді і тільки тоді, коли на «X2» більше або дорівнює «X1».

В табл. 3.1 наведено зображення різних логічних елементів в *EWB 5.12*.

Таблиця 3.1. – Зовнішній вигляд цифрових елементів в EWB 5.12.

Зображення	Функція булевої алгебри
	I
	АБО
	НЕ
	I-НЕ
	АБО-НЕ
	ВИКЛЮЧАЮЧЕ АБО
	Буфер
	Буфер з трьома станами
	ВИКЛЮЧАЮЧЕ АБО-НЕ

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

Генератор слів використовується для посилення послідовності бітів у схемі. Його ліва частина (див. рис. 3.12) вміщує 16 рядків (слів) по 8 біт кожний.

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.

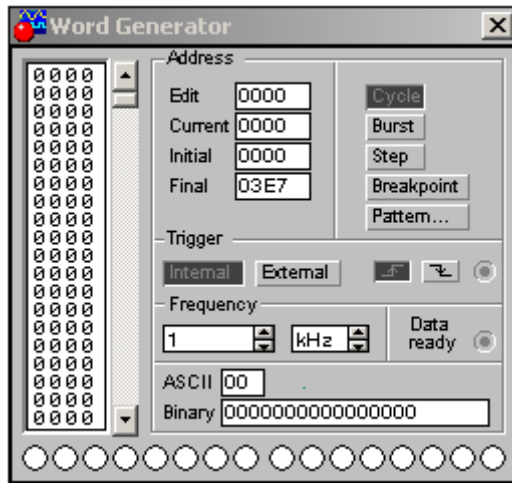


Рис. 3.12. Зовнішній вигляд генератора слів в EWB 5.12

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. Його зовнішній вигляд наведено на рис. 3.13. Аналізатор 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.

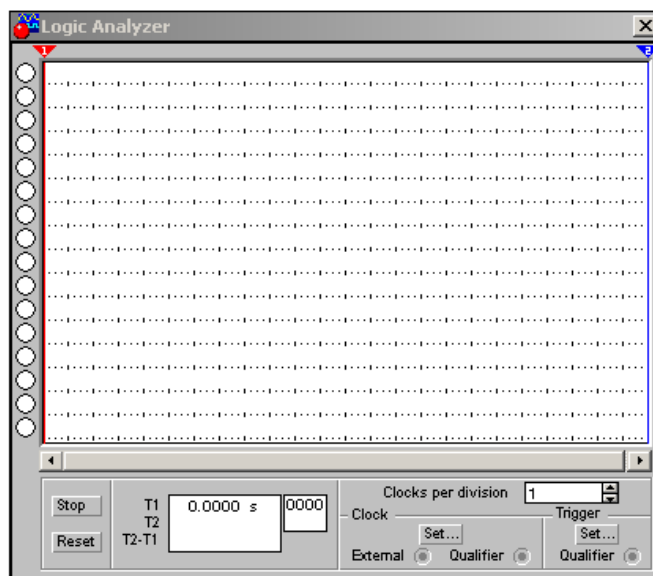
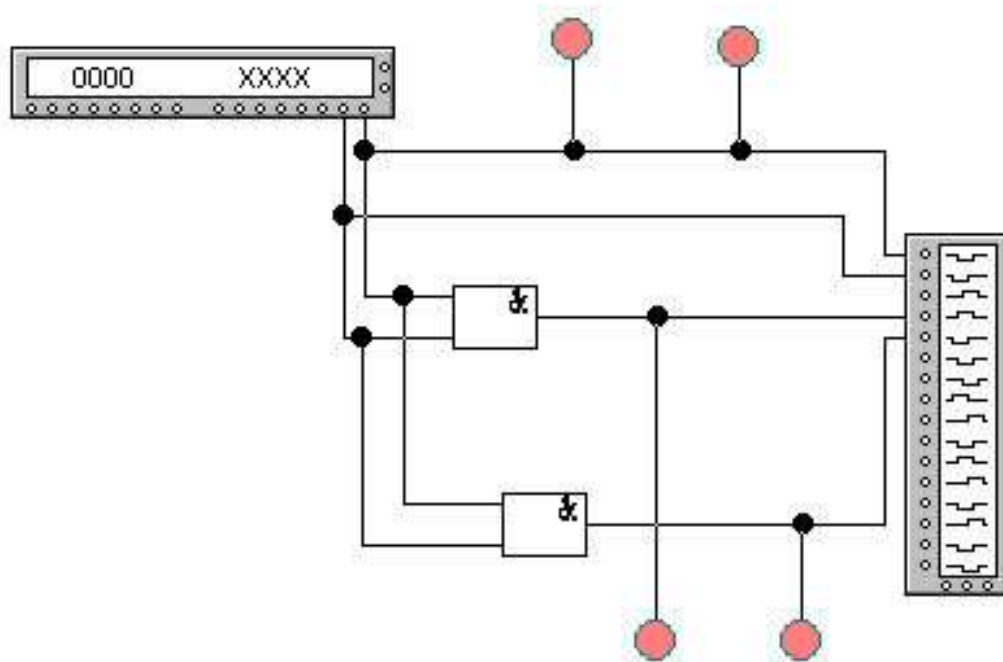
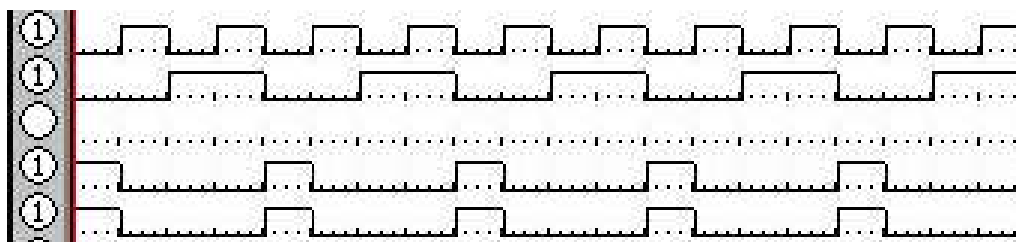


Рис. 3.13. Зовнішній вигляд логічного аналізатора в EWB 5.12

У блоці **Clock** є клеми як для звичайного (*Extend*), 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.



а)



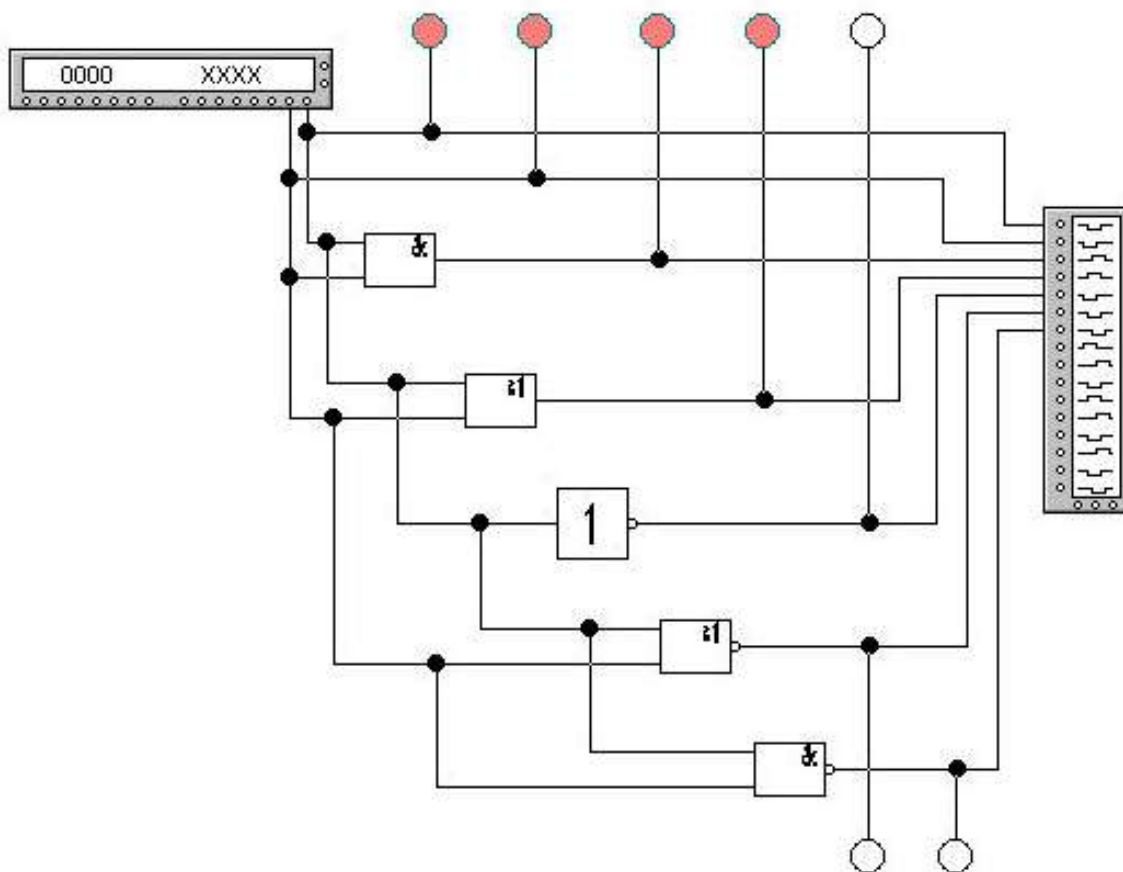
б)

Рис. 3.14. Логічний елемент «I»:
а) схема побудови в середовищі EWB; б) діаграма його роботи

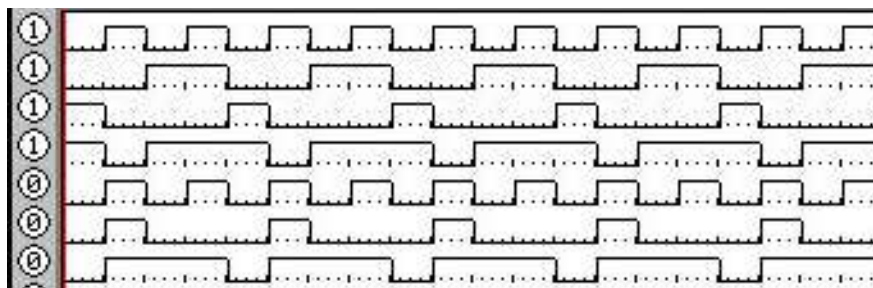
Аналогічним чином можна дослідити роботу декількох логічних елементів з'єднавши їх між собою, як це показано на рис. 3.15.

Для цього необхідно спочатку розмістити на робочому полі програми необхідні логічні елементи, а потім з'єднати їх між собою паралельно, щоб вхід кожного логічного елемента був підключений до генератора слів.

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.



а)



б)

Рис. 3.15. Логічні елементи «І», «АБО», «НЕ», «І-НЕ», «АБО-НЕ»:
а) схема побудови в середовищі EWB; б) діаграми їх роботи

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. В процесі виконання роботи необхідно дослідити окремі логічні елементи (**мінімум 7 різних логічних елементів**) та одержати для них таблиці істинності. Для цього необхідно почергово підключити входи логічних елементів до генератора слів, а до виходу елемента приєднати світлодіодний індикатор (червона лампочка).
3. Використовуючи засоби і можливості програми *Electronics Workbench* відлагодити роботу схеми.

4. Починати дослідження чергового елемента тільки після перевірки правильності одержаних результатів викладачем.

Для побудови й запуску схеми (див. *рис. 3.14 та 3.15*):

- 1) Перетягніть мишею компоненти з бібліотек компонентів.
- 2) Розмістіть їх у робочому просторі.
- 3) З'єднайте провідниками компоненти.
- 4) Задайте розміри або моделі компонентів.
- 5) Підключить генератор слів та світлодіодні індикатори.
- 6) Ввімкніть схему.
- 7) Запишіть таблиці істинності логічних елементів.

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
2. Отримані таблиці істинності кожного логічного елемента.

1. Умовно-графічні зображення логічних елементів, досліджених в ході роботи.

2. Результати спрацювання побудованих схем у вигляді таблиць істинності.

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

ЛАБОРАТОРНА РОБОТА №4

ДОСЛІДЖЕННЯ ТРИГЕРІВ

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

ТЕОРЕТИЧНІ ВІДОМОСТІ

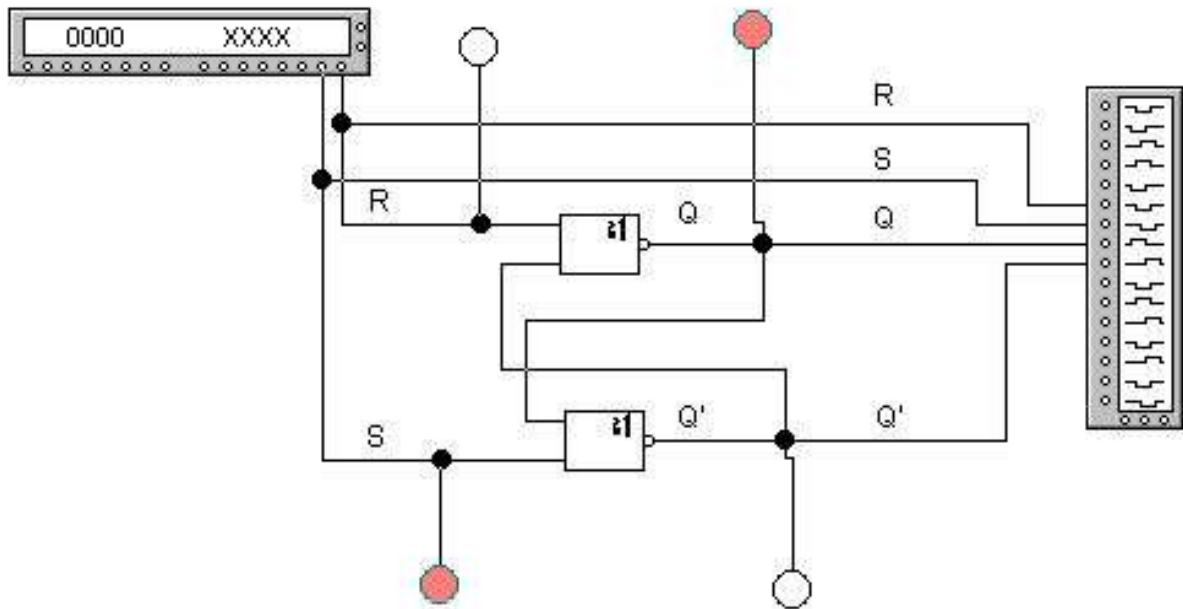
Тригер (від англійського trigger - засувка, курок) - логічний пристрій із зворотними зв'язками. Якщо сигнали на входах найпростішого тригера позначити R та S , а на виходах - Q та Q' , то зв'язок між цими змінними можна зобразити у вигляді:

$$Q = \overline{R + Q'}, \quad Q' = \overline{S + Q}$$

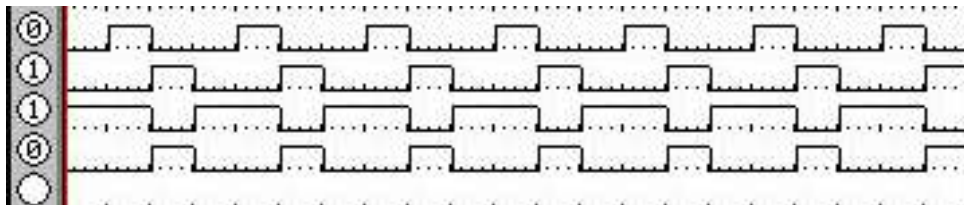
Ці формули реалізує асинхронний (несинхронізований) RS-тригер.

Асинхронний RS-тригер

Асинхронний (несинхронізований) RS-тригер на інтегральних елементах «АБО-НЕ» показаний на *рис. 4.1*. 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 інший — відкритий.



а)



б)

Рис. 4.1. Асинхронний RS-тригер:

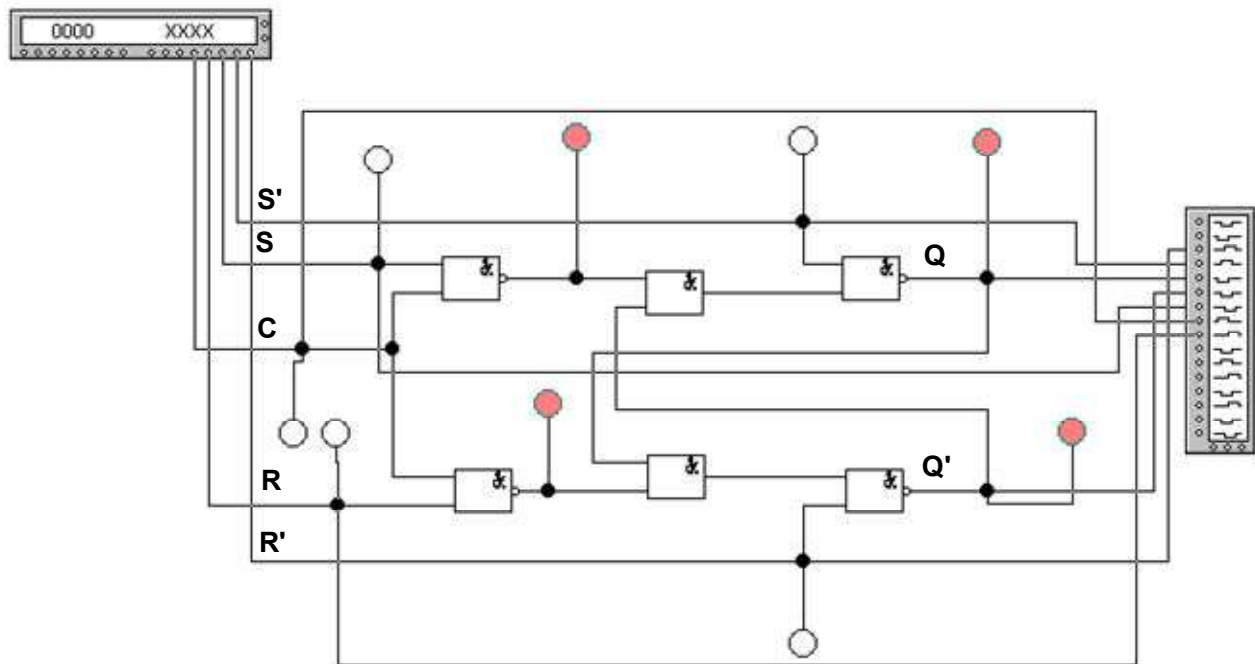
а) схема побудови в середовищі EWB; б) часові діаграми його роботи.

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.

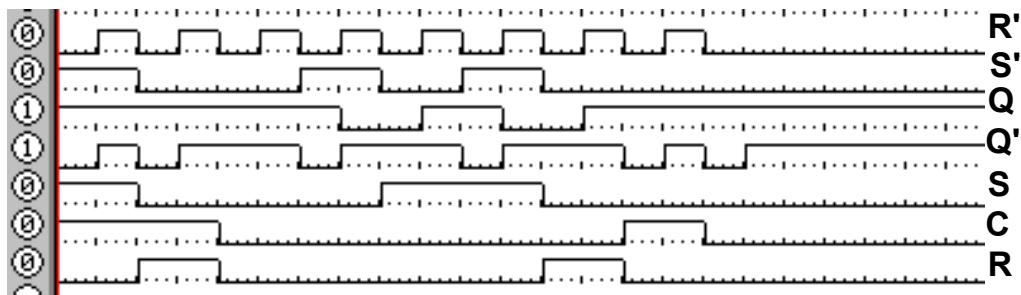
При $R=S=1$ на прямому і інверсному виходах встановлюється нульовий сигнал. Тригерне коло перетворюється в два незалежних інвертори, і при переході до збереження ($R=S=0$) тригер може встановлюватися в непередбачуваний стан. Тому така комбінація входних сигналів вважається забороненою.

Однотактний синхронізований RS-тригер

На рис. 4.2 наведена схема однотактного синхронізованого RS-тригера на елементах «І-НЕ». Тут перші два елементи утворюють схему входної логіки асинхронного RS-тригера, побудованого на другій парі елементів. Такі RS-тригери мають два інформаційних входи R і 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.



а)



б)

Рис. 4.2. Однотактний синхронізований RS-тригер:
а) на логічних елементах; б) часові діаграми його роботи.

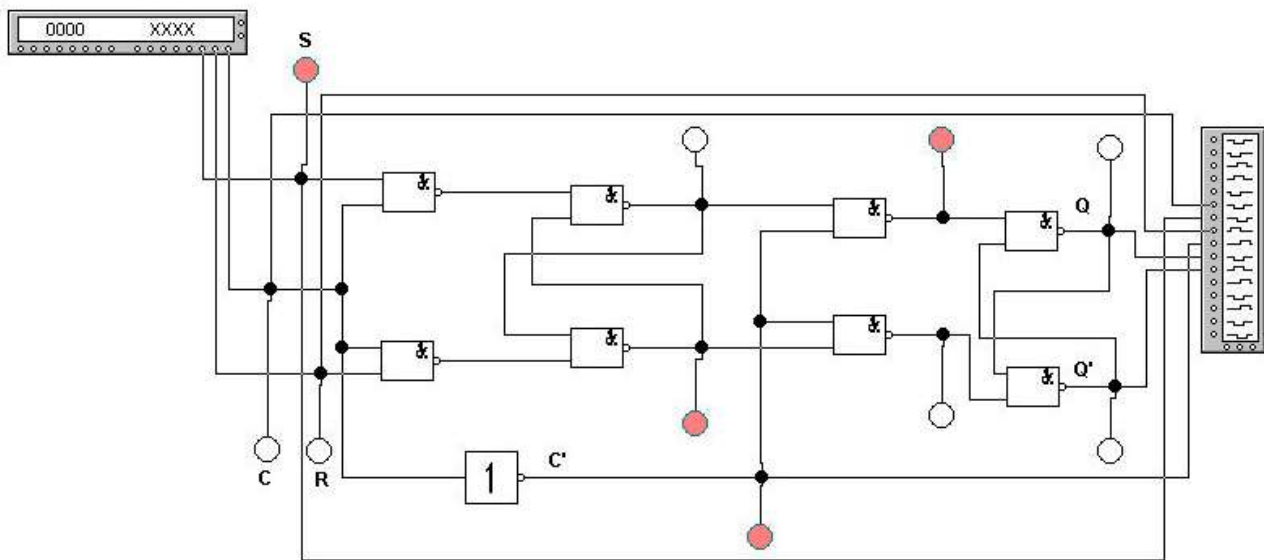
Двотактний RS- тригер

Найпростіша схема двовходового двотактного RS-тригера показана на рис. 4.3.

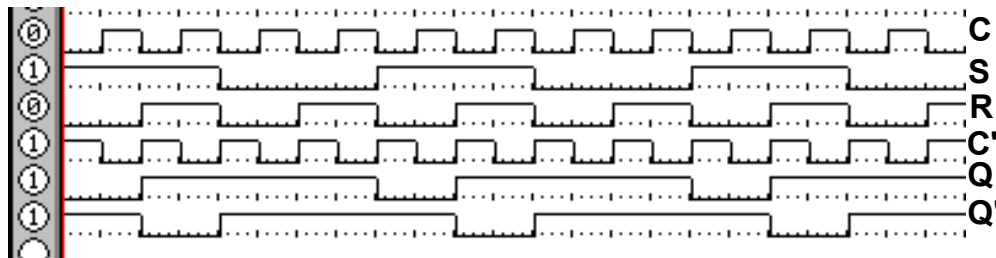
Вона складається з двох однотактних RS-тригерів і інвертора в колі синхронізації. При надходженні на вхід C RS-тригера сигналу C=1 вхідна інформація заноситься в перший однотактний синхронізований RS-тригер, а

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).



а)



б)

Рис 4.3. Двотактний RS – тригер:
а) на елементах «І-НЕ»; б) часові діаграми його роботи

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.

2. В процесі виконання роботи необхідно побудувати схеми з'єднання окремих логічних елементів, одержати їхні таблиці істинності, зібрати схеми асинхронного RS-тригера, синхронізованого RS-тригера, двотактного RS-тригера. Після складання кожену 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 вхідні сигнали генератора слова.

3. Використовуючи засоби і можливості програми *Electronics Workbench* відлагодити роботу схеми.

4. Починати складання чергової схеми тільки після перевірки правильності роботи попередньої зібраної схеми викладачем.

Для побудови й запуску схеми:

- 1) Перетягніть мишею компоненти з бібліотек компонентів.
- 2) Розмістіть їх у робочому просторі.
- 3) З'єднайте провідниками компоненти.
- 4) Задайте розміри або моделі компонентів.
- 5) Підключить контрольно-вимірювальні прилади.
- 6) Ввімкніть схему.
- 7) Запишіть таблиці істинності логічних елементів і графіки вхідних і вихідних сигналів досліджуваних схем.

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
2. Отримані таблиці істинності кожного логічного елемента.
3. Схеми, тригерів, побудовані на екрані комп'ютера з поясненнями.
4. Результати спрацьовування побудованих схем у вигляді таблиці переходів й осцилограм вхідних і вихідних сигналів.

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

ЛАБОРАТОРНА РОБОТА № 5

ПОБУДОВА ЛІЧИЛЬНИКІВ І ВИВЧЕННЯ ЇХ РОБОТИ

Мета роботи: вивчити принципи роботи різномісних лічильників за допомогою навчальної прикладної програми *Electronics Workbench*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Лічильником називається вузол ЕОМ, призначений для підрахування вхідних сигналів. Лічильники можуть бути побудовані на основі рахункових тригерів зі спеціальними міжрозрядними зв'язками, на основі регістрів, що зсувають (кільцеві лічильники) і на основі багато стійких елементів.

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

Основними часовими характеристиками лічильників є:

f – максимальна частота надходження рахункових сигналів;

t – час переходу лічильника з одного стану в інший.

По характеру операцій лічильники можна розділити на *підсумовуючі*, що *віднімають* та *реверсивні*.

За способом організації ланцюгів переносу між розрядами лічильники підрозділяють на такі типи з: *послідовним, паралельним, наскрізним, груповим переносом*.

У лічильниках з ***послідовним переносом*** перенос (позиція) у сусідній старший розряд формується тільки після переключення тригера в попередньому розряді, тобто тригери переключаються 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.

Ланцюги *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 меншого числа входів логічних елементів для організації ланцюгів переносу, але уступають лічильникам з паралельним переносом у швидкодії.

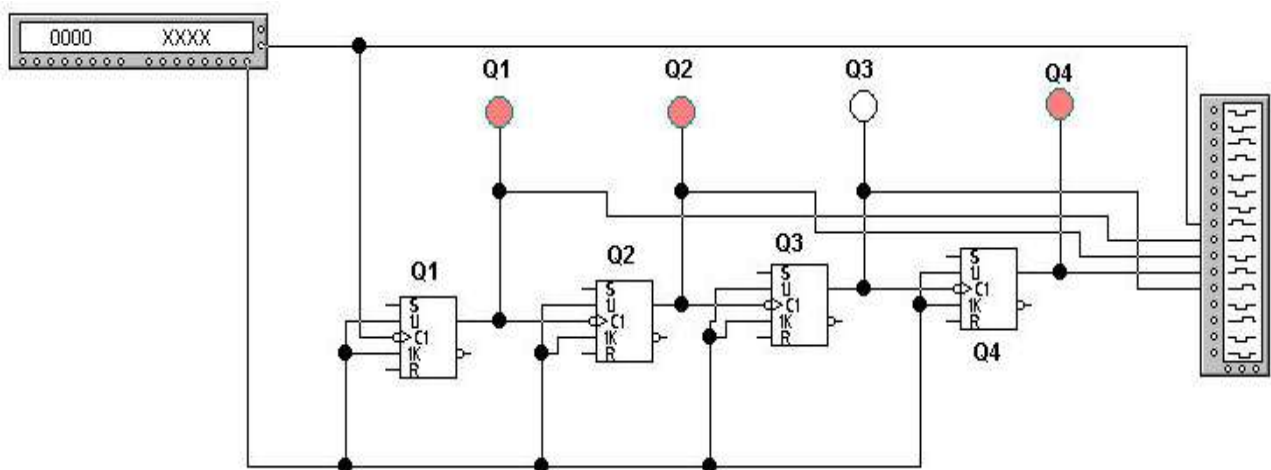
В лічильниках з *паралельним переносом* аргументами функцій переносів для кожного розряду є тільки сигнали на виходах тригерів відповідних розрядів. Переноси для всіх розрядів лічильника формуються одночасно (за умови, що всі логічні елементи в схемі мають однаковий час переключення).

В зв'язку з обмеженнями на побудову лічильників з паралельним переносом великої розрядності широке розповсюдження отримали лічильники з *груповим переносом (послідовно-паралельним переносом)*. Розряди таких лічильників розбиваються на групи, усередині яких організується принцип паралельного переносу. Групи з'єднують послідовно з використанням кон'юкторів, 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'юктор до прямого пропускання вхідного сигналу на наступну групу.

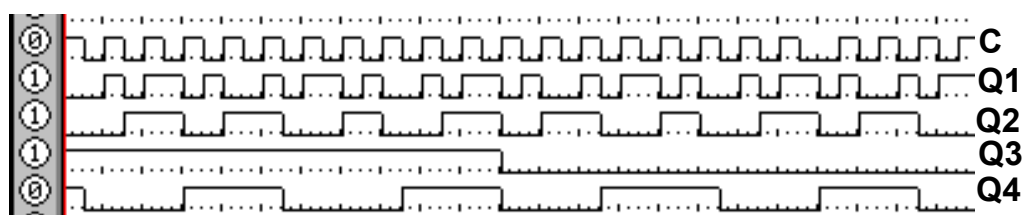
Несинхронізований чотирьохрозрядний підсумовуючий лічильник з послідовним переносом

На *рис. 5.1* наведено схему несинхронізованого чотирьохрозрядного підсумовуючого лічильника з послідовним переносом.

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.



a)

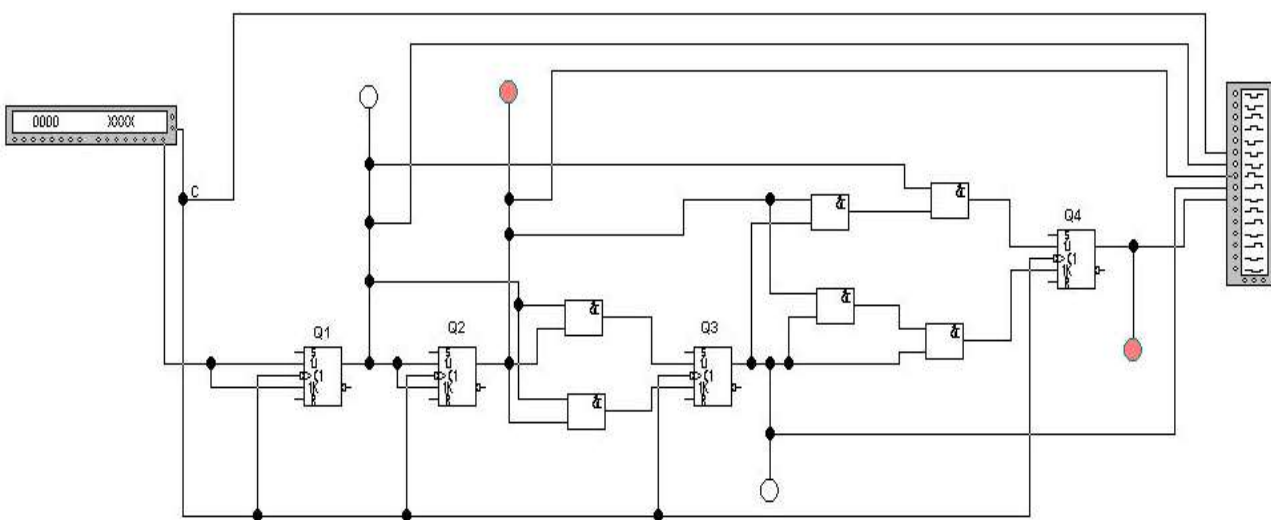


б)

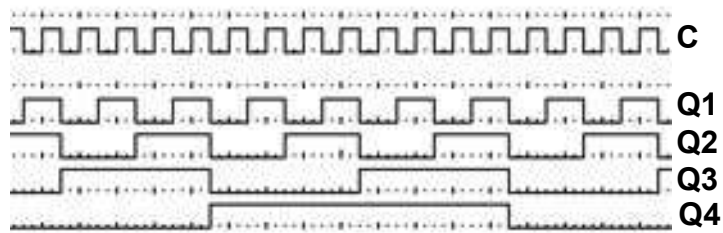
Рис 5.1. Несинхронізований чотирьохрозрядний підсумовуючий лічильник з послідовним переносом (а) та часові діаграми його роботи (б)

Несинхронізований чотирьохрозрядний підсумовуючий лічильник з паралельним переносом

На *рис. 5.2* наведено схему несинхронізованого чотирьох розрядного підсумовуючого лічильника з паралельним переносом.



a)



б)

Рис 5.2. Несинхронізований чотирьохрозрядний підсумовуючий лічильник з паралельним переносом (а) та часові діаграми його роботи (б)

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 елементах «І» JK-тригера, але тривалість перехідного процесу в такому лічильнику дорівнює тривалості перемикання одного розряду.

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*.

Для побудови і дослідження схем:

- 1) Перетягніть мишкою елементи схеми з бібліотек компонентів.
- 2) Розмістіть їх в робочому просторі екрану.
- 3) З'єднайте елементи схеми провідниками.
- 4) Перетягніть мишею контрольно-вимірювальні прилади: генератор слів та логічний аналізатор з бункера приладів та приєднайте їх до схеми.
- 5) Задайте таблицю вхідних сигналів на генераторі слів. Вмикайте подачу вхідного сигналу за допомогою органів керування генератора слів або вимикача розташованого в правому верхньому кутку екрана.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. В процесі виконання роботи необхідно побудувати схеми лічильників з послідовним і паралельним переносом.
3. Після складання кожену схему підключити до приладів: генератора слів і логічного аналізатора.
4. 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 й вихідні сигнали генератора слів.
5. Використовуючи наявні засоби налагодити, роботу схеми одержати

осцилограми роботи досліджуваних схем.

6. Приступати до складання чергової схеми тільки після перевірки викладачем правильності роботи попередньої схеми.

7. Скласти звіт з лабораторної роботи.

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
2. Рисунки схем, побудованих на екрані комп'ютера з поясненнями.
3. Результати спрацювання побудованих схем у вигляді осцилограм вхідних і вихідних сигналів.

4. Висновки.

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

ЛАБОРАТОРНА РОБОТА № 6

ПОБУДОВА РЕГІСТРІВ, ДЕШИФРАТОРІВ І ВИВЧЕННЯ ЇХ РОБОТИ

Мета роботи: вивчити принципи роботи регістру зсуву та дешифратора за допомогою навчальної прикладної програми *Electronics Workbench*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Регістром називається пристрій, призначений для запам'ятовування слова, а також для виконання над словом певних логічних перетворень.

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

Розрізняють *синхронні* й *асинхронні* регістри. У синхронних регістрах операції виконують по тактуючому сигналу T . Налаштування регістра на відповідну мікрооперацію здійснюється попередньою установкою на керуючих входах коду мікрооперації. Число розрядів у коді визначається зі співвідношення $T \geq |\log_2 k|$, де k – кількість мікрооперацій.

В асинхронних регістрах тактуючий 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.

Найбільш часто на регістрах виконують операції занесення (прийому,

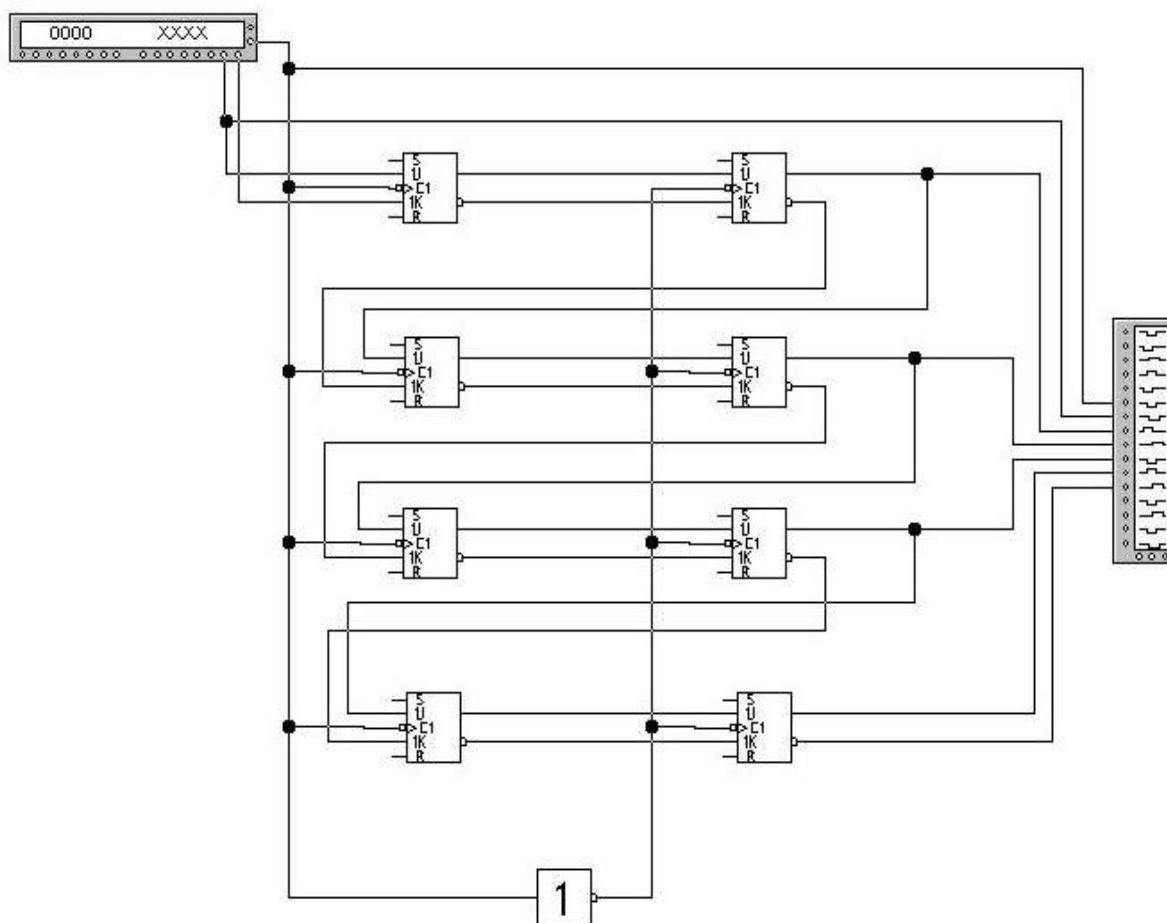
запису) слова паралельним кодом, зсуву слова, а також установки вхідного (як правило, нульового) стану.

Для побудови регістрів в залежності від виконуваних мікрооперацій можуть бути використані тригери, різні по функціональній ознаці (наприклад, RS-, JK-, T-, D-типу) і по організації (синхронні й асинхронні, із внутрішньою затримкою і без внутрішньої затримки).

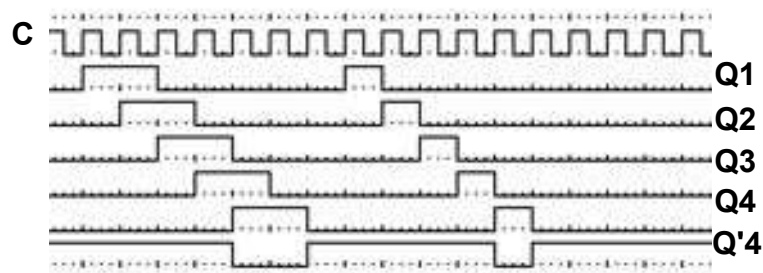
Зазвичай в регістрах всі розряди виконуються однаково. Розряд синхронного регістра (без обліку ланцюгів видачі інформації) може бути побудований як на синхронному, так і на асинхронному тригері.

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. 6.1 зображено схему регістра зсуву на одноканальних RS-тригерах. 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.



а)



б)

**Рис. 6.1. Регістр зсуву на одноканальних RS-тригерах (а)
та часові діаграми його роботи (б)**

Дешифратором називається комбінаційна схема, з кількома входами і виходами, яка перетворює код, що подається на входи в сигнал на одному з виходів. Тобто, на одному з виходів дешифратора з'являється «0», а на інших виходах - «1». В загальному випадку дешифратор з n входами має « 2^n » виходів.

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, представляють собою логічний добуток всіх дозволяючих сигналів керування.

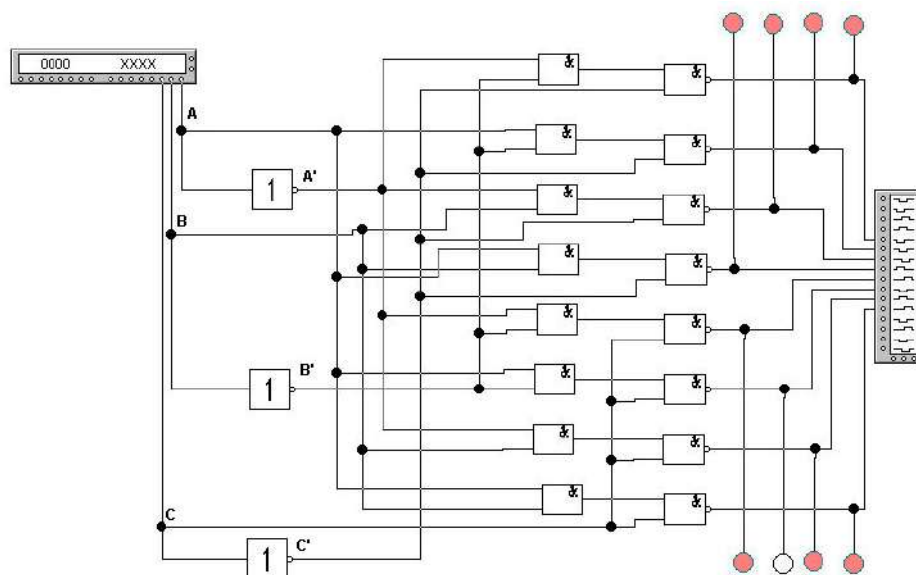
Зазвичай входи керування використовуються для збільшення розрядності дешифраторів або при паралельній роботі декількох схем на загальні вихідні лінії.

На *рис. 6.2 та 6.3* наведено спосіб побудови дешифратора для трьохрозрядного вхідного слова та показано дешифратор адреси.

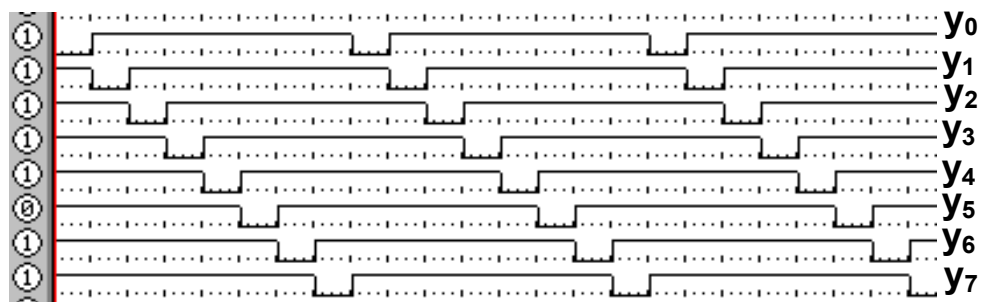
Для зменшення розмірів схеми на екрані вся вона або її частина можуть бути згорнуті у підсистему («мікросхему») і позначатись в подальшому одним елементом. Для виконання цієї дії необхідно скористатись підпунктом *Create Subcircuit* пункту меню **Circuit**. У віконці, що відкриється необхідно задати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

Для побудови і дослідження схем:

- 1) Перетягніть мишкою елементи схеми з бібліотек компонентів.
- 2) Розмістіть їх в робочому просторі екрану.
- 3) З'єднайте елементи схеми провідниками.
- 4) Перетягніть мишею контрольно-вимірювальні прилади: генератор слів та логічний аналізатор з бункера приладів та приєднайте їх до схеми.
- 5) Задайте таблицю вхідних сигналів на генераторі слів. Вмикайте подачу вхідного сигналу за допомогою органів керування генератора слів або вимикача розташованого в правому верхньому кутку екрана.

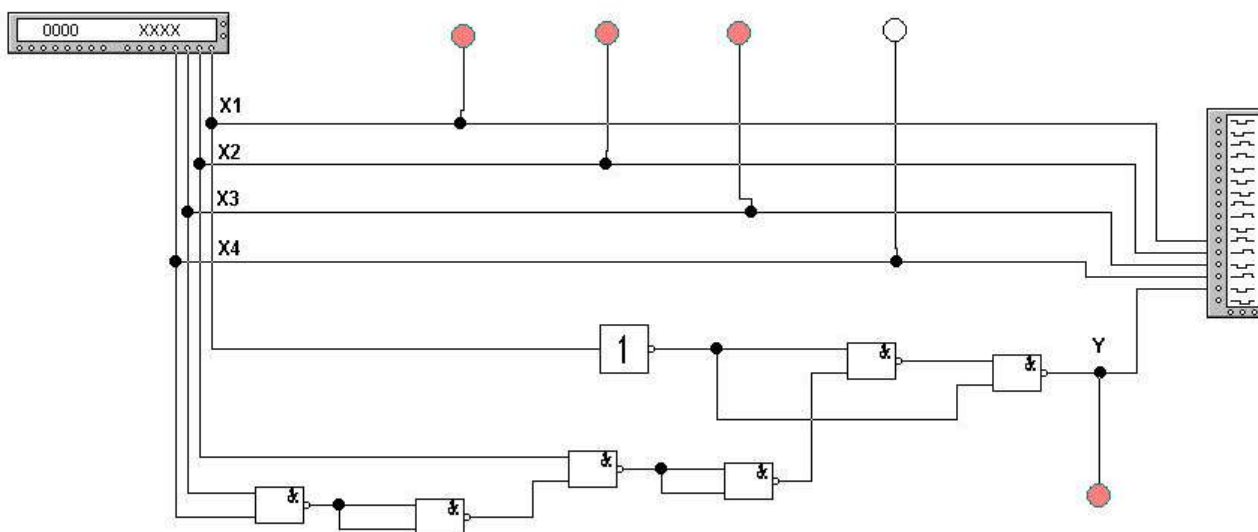


a)

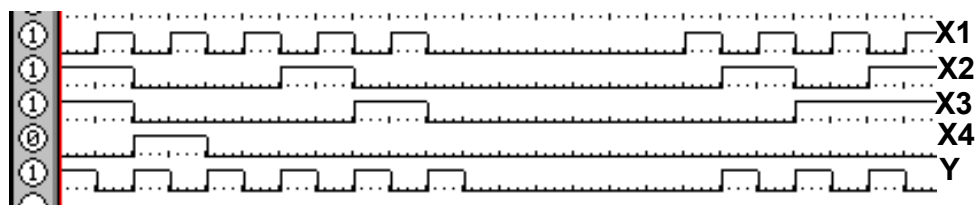


б)

Рис 6.2. Дешифратор для трьохрозрядного вхідного слова (а) та часові діаграми його роботи при подачі на вхід послідовності двійкових чисел у порядку зростання (б)



a)



б)

Рис 6.3. Дешифратор адреси (а) та часові діаграми його роботи (б)

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. В процесі виконання роботи необхідно побудувати схеми регістру зсуву на RS-тригерах та дешифратори.
3. Після складання кожну схему підключити до приладів: генератора слів і логічного аналізатора.
4. При дослідженні регістру зсуву на одноканальних RS-тригерах на входи першого тригера з генератора слів мають подаватись взаємно протилежні сигнали.
5. Використовуючи наявні засоби налагодити, роботу схеми одержати осцилограми роботи досліджуваних схем.
6. Приступати до складання чергової схеми тільки після перевірки викладачем правильності роботи попередньої схеми.
7. Скласти звіт з лабораторної роботи.

СКЛАД ЗВІТУ:

1. Назва і мета роботи.
 2. Рисунки схем, побудованих на екрані комп'ютера з поясненнями.
 3. Результати спрацювання побудованих схем у вигляді осцилограм вхідних і вихідних сигналів.
 4. Висновки.
- Звіт з лабораторної роботи повинен бути виконаний на аркушах формату А4 (297*210 мм).

ЛАБОРАТОРНА РОБОТА №7

ВИВЧЕННЯ РОБОТИ АРИФМЕТИЧНИХ СУМАТОРІВ

Мета роботи: ознайомитись з принципами побудови та роботи арифметичних суматорів за допомогою можливостей навчально-прикладної програми *Electronics Workbench*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Арифметичні суматори є складовою частиною арифметико-логічних пристроїв (АЛП). Суматор виконує арифметичне складання чисел. Для складання чисел, наведених у позиційному паралельному двоїстому коді, застосовуються паралельні двоїсті суматори.

Суматор (рис 7.1) має n входів розрядів доданку A , n входів доданку B і вхід переносу C_i (від англійського *carry in*.. - перенесення до...). Виходами суматора є n виходів розрядів суми Σ і вихід переносу (переповнення) C_o (від англійського *carry out*... - перенесення з...).

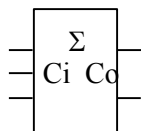


Рис. 7.1. Зображення суматора в EWB 5.12

Класифікація суматорів.

В залежності від системи числення розрізняють: двійкові, десяткові, двійково-десяткові та ін.

За кількістю розрядів додавання чисел, які одночасно оброблюють: однорозрядні та багаторозрядні.

По способу представлення та обробки додатків чисел багаторозрядні суматори бувають:

- послідовні, в яких обробка чисел відбувається по чергові, розряд за розрядом на одному й тому є обладнанні;
- паралельні, в яких доданки додаються одночасно по всіх розрядах, і для кожного розряду є своє обладнання.

Швидкодія суматора визначається часом розповсюдження сигналу по ланцюгу переносу. Зменшення цього часу – основна задача при побудові паралельних суматорів.

Для зменшення часу розповсюдження сигналу переносу використовують: конструктивні рішення, коли використовують у ланцюгу переносу найбільш швидкодіючі елементи; старанно виконувати монтаж без довгих провідників; структурні методи прискорення проходження сигналу переносу.

По способу організації процесу додавання однорозрядних доданків розрізняють: комбінаційні, накопичуючі та комбінаційно-накопичуючі суматори.

За організацією ланцюгів переносу між розрядами суматори бувають з: послідовним переносом, наскрізним, паралельним та груповим переносом.

Суматор із паралельним переносом

В суматорі з паралельним переносом тракти всіх однорозрядних суматорів ввімкнені послідовно. Тому, навіть при мінімальній затримці тракту переносу однорозрядного суматора затримка n -розрядного суматора не може бути менше n .

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 послідовним переносом

Найпростіший спосіб побудови n -розрядного суматора - це послідовне вмикання n однорозрядних суматорів, які підсумовують одноіменні розряди доданків, як показано на рис. 7.2.

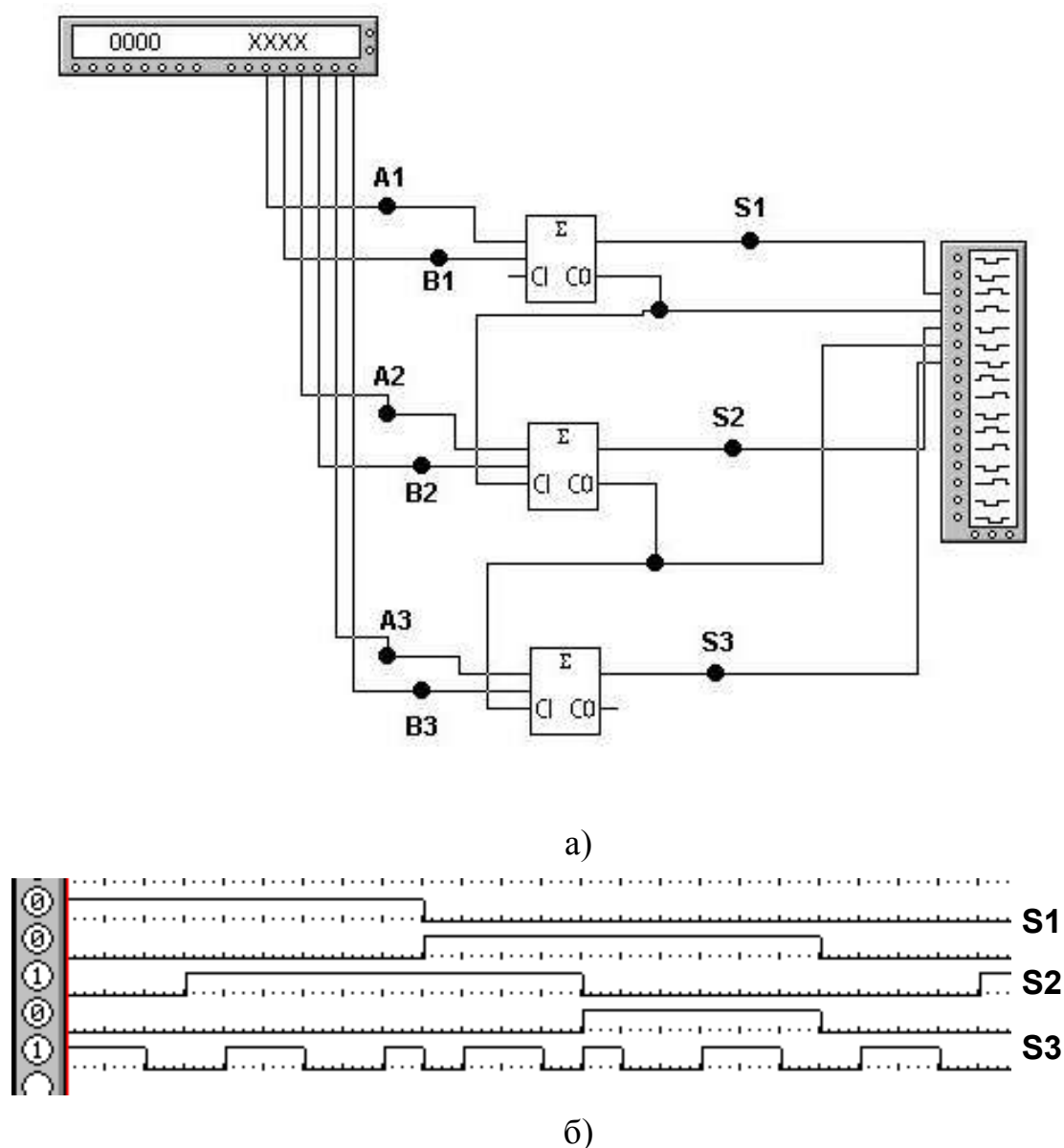


Рис. 7.2. Трьохрозрядний суматор з послідовним переносом (а)
та часові діаграми його роботи (б)

Суматор із груповим переносом

Суматор розбивається на кілька груп приблизно рівної довжини. Сигнал переносу, що надходить на вхід молодшого розряду групи, при наявності умов розповсюдження переносу у всіх розрядах даної групи передається на вхід молодшого розряду сусідньої, більш старшої групи в обхід даної групи.

Схема формування сигналу переносу в молодшому розряді кожної групи доповнюється для цього схемою І, що реалізує булеву функцію:

$$P_{\text{приск}} = P_i \cdot c_i \cdot c_{i+1} \cdot \dots \cdot c_{i+k-1},$$

де $P_{\text{приск}}$ – сигнал прискорення переносу;

P_i – сигнал переносу в молодший розряд групи, що складається з розрядів;

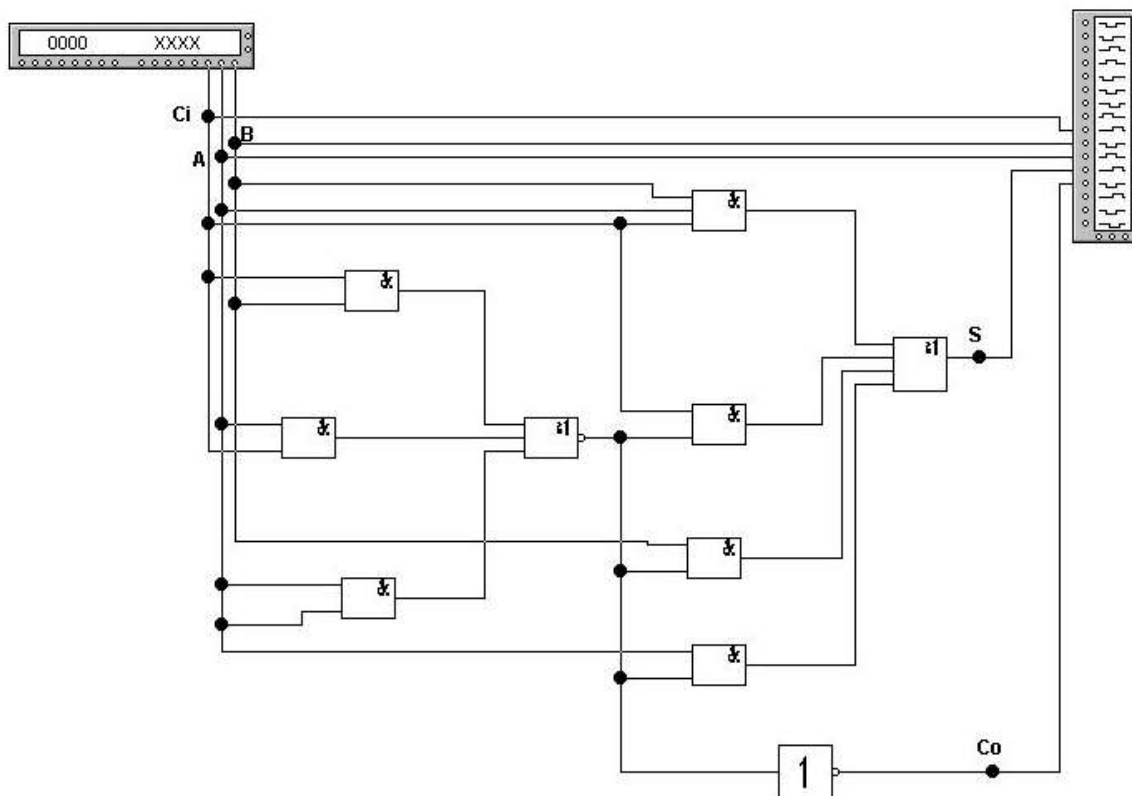
$c_i, \dots, c_{i+k-1}$ – умови поширення переносу в розрядах групи ($c_i = a_i \vee b_i$).

Максимальна затримка сигналу переносу в такому суматорі визначається затримкою в його молодшій, старшій групах і в колах обходу інших груп.

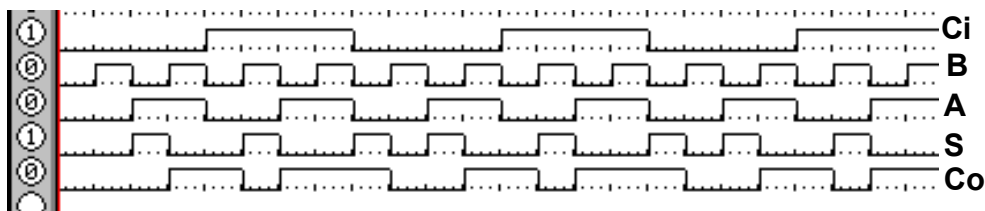
Максимальна затримка може бути зменшена, якщо при розбитті суматора на групи використати паралельне (одночасне) формування переносу в середині груп.

Суматор на логічних елементах

Схеманаведена на рис. 7.3.



a)



б)

**Рис 7.3. Однорозрядний суматор на логічних елементах (а)
та часові діаграми його роботи (б)**

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

$$\overline{Co} = \overline{Ci \cdot A \cup Ci \cdot B \cup A \cdot B} ;$$

$$\Sigma = Ci \cdot \overline{Co} \cup A \cdot \overline{Co} \cup B \cdot \overline{Co} \cup Ci \cdot A \cdot B.$$

Більш докладно про суматори можна дізнатись з джерел [1, 2, 5, 6].

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Вивчити теоретичні відомості.
2. В процесі виконання роботи необхідно побудувати схеми: трьохрозрядного суматора з послідовним переносом та одноразрядного суматора на елементах.
3. Після складання, кожену схему підключити до приладів необхідних для проведення потрібних вимірів.
4. Використовуючи наявні засоби налагодити роботу схеми одержати осцилограми роботи досліджуваних схем.
5. Приступати до складання чергової схеми тільки після перевірки викладачем правильності роботи складеної схеми.
6. Скласти звіт з лабораторної роботи.

СКЛАД ЗВІТУ

1. Назва і мета роботи.
 2. Рисунки схем, побудованих на екрані комп'ютера з короткими поясненнями.
 3. Результати спрацювання побудованих схем у вигляді осцилограм вхідних і вихідних сигналів.
 4. Висновки.
- Звіт по лабораторній роботі повинен бути виконаний на аркушах формату А4 (297*210 мм).

ЛАБОРАТОРНА РОБОТА №8

КОНСТРУЮВАННЯ КОМБІНАЦІЙНИХ СХЕМ ТА ЇХ ОПТИМІЗАЦІЯ

Мета роботи: Вивчити методи конструювання та оптимізації комбінаційних схем за допомогою можливостей навчально-прикладної програми *Electronics Workbench*.

ТЕОРЕТИЧНІ ВІДОМОСТІ

Існують три «основних знаряддя праці» для розв'язання задач логіки: умовні позначення логічних елементів, 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) [3-5].

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 найбільше наочно і повно представляється таблицею істинності, у якій для кожної комбінації значень змінних вказується значення функції.

Наприклад, нехай функція F задана у вигляді таблиці. Для комбінацій змінних 4,6,7,8 функція $Y = 1$, що означає для зазначених комбінацій рівність одиниці наступних добутків: $A'BC = 1$, $AB'C = 1$; $ABC' = 1$, $ABC = 1$. Комбінації змінних, при яких $F = 1$, називають конститuentами одиниці або *мінтермами*. Представлення логічної функції у вигляді суми мінтермів визначає її досконалу ДНФ, тобто в даному випадку

$$Y = A'BC + AB'C + ABC' + ABC$$

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

Так, на підставі таблиці

Номери комбінацій	A	B	C	X
1	0	0	0	0
2	0	0	1	0
3	0	1	0	0
4	0	1	1	1
5	1	0	0	0
6	1	0	1	1
7	1	1	0	1
8	1	1	1	1

можна записати:

$$Y = (A+B+C)(A+B+C')(A+B'+C)(A'+B+C)$$

Кожний співмножник у цьому виразі знаходиться із суми змінних, для яких функція перетворюється у нуль відповідно до таблиці істинності. Такі суми називають *конституентами нуля або макстермами*. Таким чином, зміст макстермів визначає досконалу КНФ функції.

Для зменшення числа логічних елементів, що реалізують функцію, застосовується спрощення її форми (*мінімізація*). Для мінімізації нескладних функцій використовуються алгебраїчні перетворення, а для функцій, що мають велику кількість змінних (понад три) й велику кількість складових, - спеціальні методи. Одним із таких методів є *метод карт Карно-Вейча* [3-5].

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

Приклади побудови схем на основі булевих виразів.

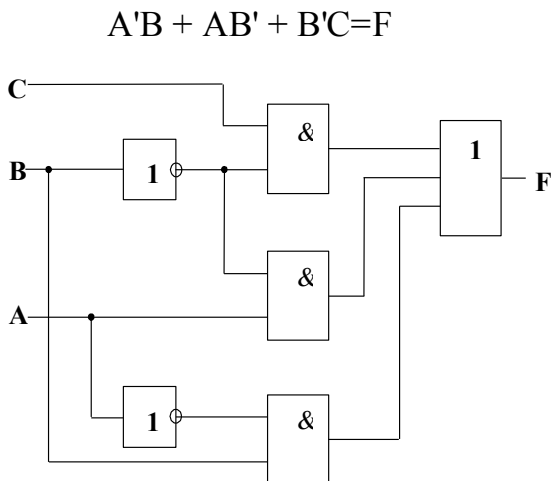


Рис. 8.1. Приклад булевого виразу та схеми до спрощення

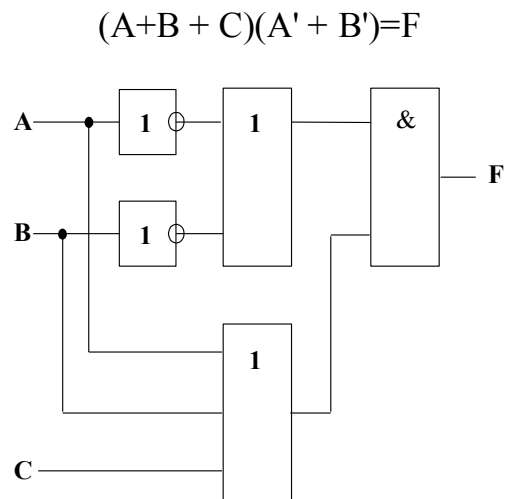


Рис. 8.2. Приклад булевого виразу та схеми після спрощення за допомогою карт Карно-Вейча

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 дві, чотири або вісім 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.

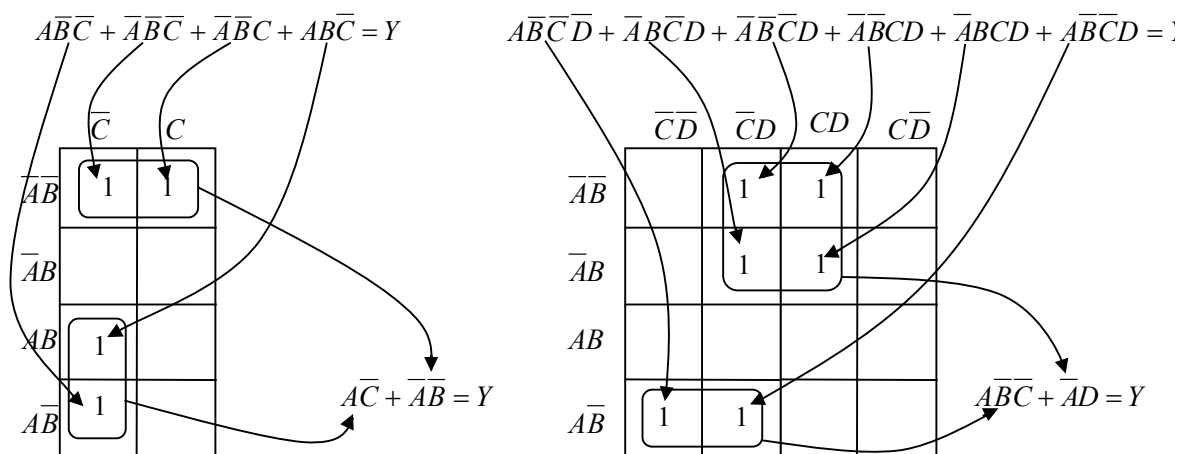


Рис. 8.3. Мінімізація функції алгебри логіки методом карт Карно-Вейча

Для конструювання комбінаційних схем за допомогою програми *Electronics Workbench* використовується прилад Логічний перетворювач.

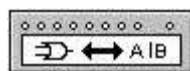


Рис. 8.4. Вигляд приладу «логічний перетворювач» в EWB

Логічний перетворювач – потужний комп'ютерний прилад, що виконує окремі перетворення представлень схеми. Ви можете використовувати його, щоб перетворювати:

- схему в таблицю істинності або діаграму кола;
- таблицю істинності в булевий вираз;
- булевий вираз в схему або таблицю істинності.

Уздовж правого боку відкритого вікна логічного перетворювача є набір кнопок перетворення, що використовуються, щоб виконувати певні операції.

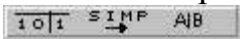
Перетворення схеми в таблицю істинності. Логічний перетворювач може створювати таблицю істинності для ланцюгів, що вміщують до восьми входів і одного виходу. Приєднайте входи схеми до клем логічного перетворювача. Тепер з'єднайте вихід схеми з клемою молодшого розряду на «іконці»



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, від А до Н, розташованих горизонтально у верхній частині логічного перетворювача. Відредагуйте значення у вихідній колонці, щоб задати бажаний вихід для кожного набору входних станів. Щоб змінювати значення, «натисніть» на неї і введіть 1, 0 або Х. (Х означає що припустима як 1, так і 0.)

	A	B	C	D	E	F	G	H	OUT
141	1	0	0	0	1	1	0	1	0
142	1	0	0	0	1	1	1	0	X
143	1	0	0	0	1	1	1	1	1
144	1	0	0	1	0	0	0	0	0
145	1	0	0	1	0	0	0	1	0
146	1	0	0	1	0	0	1	0	0

Спрощення Булевого виразу. Щоб зменшити Булевий вираз, спочатку перетворіть його в таблицю істинності, а вже потім спрощуйте. Щоб зменшити Булевий натисніть на третю згори кнопку .

Примітка: Булеве спрощення вимагає великої кількості пам'яті. Якщо ваш комп'ютер не має достатньої пам'яті, буде неможливо завершити цю операцію.

Перетворення Булевого виразу в таблицю істинності. Ви можете ввести Булевий вираз в текстовому полі в нижній частині логічного перетворювача.

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

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

Створення схеми з логічних елементів «І-НЕ».

Щоб створити схему, яка повторює стани Булевого виразу, використовуючи тільки лише логічні елементи «І-НЕ», натисніть на останню кнопку .

ПРИКЛАД СПРОЩЕННЯ СХЕМИ ЗА ДОПОМОГОЮ ЛОГІЧНОГО АНАЛІЗАТОРА

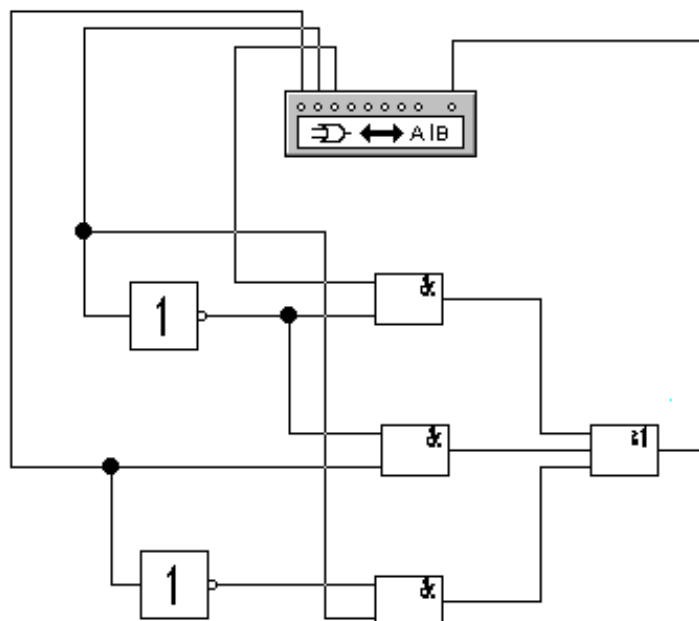


Рис. 8.5. Схема на логічних елементах до спрощення

	A	B	C	F
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	1
007	1	1	1	0

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

Рис. 8.6. Булевий вираз та таблиця істинності схеми до спрощення

	A	B	C	F
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	1
007	1	1	1	0

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

Рис. 8.7. Булевий вираз та таблиця істинності схеми після спрощення

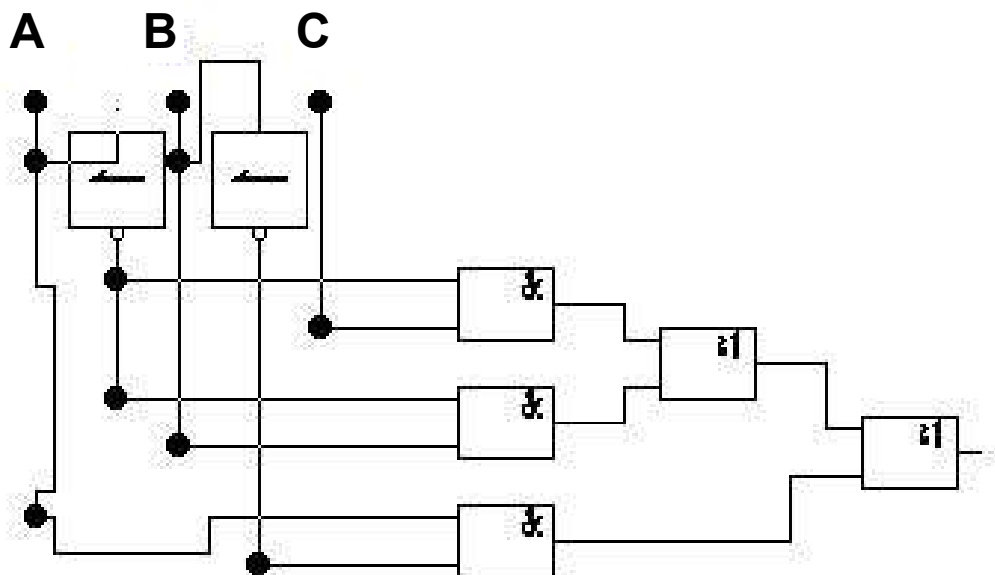


Рис. 8.8. Спрощена схема

ЛІТЕРАТУРА ДО ЛАБОРАТОРНОГО КУРСУ

1. Каган Б. М. Электронные вычислительные машины и системы / Каган Б. М. – М.: Энергоатомиздат, 1991. – 592 с.
2. Калабеков Б. А. Цифровые устройства и микропроцессорные системы / Калабеков Б. А. – М.: Телеком, 2000. – 336 с.
3. Кравчук А. Ф. Основы дискретной математики / Кравчук А. Ф. – К.: УМК ВО, 1992. – 196с.
4. Кравчук А. Ф. Проектування дискретних пристроїв автоматики / Кравчук А. Ф., Віхрова Л. Г. – К.: ІСДО, 1993. – 208 с.
5. Партала О. Н. Цифровая электроника / Партала О. Н. – Санкт-Петербург: Наука и техника, 2000. – 208 с.
6. Потемкин И. С. Функциональные узлы цифровой автоматики / Потемкин И. С. – М.: Энергоатомиздат, 1988. – 320 с.

Навчально-методичне видання

Основи комп'ютерної схемотехніки

Методичні вказівки до виконання лабораторних робіт
для студентів денної та заочної форми навчання за напрямками підготовки:
151 «Автоматизація та комп'ютерно-інтегровані технології», 141
«Електроенергетика, електротехніка та електромеханіка»
172 «Телекомунікації та радіотехніка»

Укладачі: *Р. В. Жесан*
О. П. Голик
В. О. Зубенко