

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Р.Є.Рикалюк

**ЛАБОРАТОРНИЙ (СИМУЛЯЦІЙНИЙ)
ПРАКТИКУМ
з курсу
«АРХІТЕКТУРА ЕОМ»**

Львів ЛНУ 2007-2015

Рикалюк Р.Є. Лабораторний (симуляційний) практикум з курсу
“Архітектура ЕОМ”. Видавн. центр Львів. ун-ту, 2007. - 57 с.

Розглянуто функціональні можливості пакета синтезу та дослідження електричних та електронних кіл Electronics Workbench, викладено методику роботи з ним, сформульовано завдання до проведення лабораторних робіт з курсу “Архітектура ЕОМ” (розділи «Основи алгебри логіки» та «Схемотехніка»).

Для студентів факультету прикладної математики та інформатики а також факультету електроніки.

Рецензент: доц. Рабик В.Г.

Редактор М.М.Мартиняк

©Р.Є.Рикалюк, 2007

Програма курсу “Архітектура ЕОМ” передбачає виконання лабораторних робіт з основ алгебри логіки та схемотехніки. Фізичний та технологічний аспекти процесу вивчення цифрової техніки не розглядають. Завдяки наявності потужного прикладного пакета синтезу й аналізу електричних та електронних кіл Electronics Workbench можна організувати віртуальну електронну лабораторію на екрані персонального комп’ютера. Цей практикум містить методичні рекомендації та завдання до виконання циклу лабораторних робіт з основ алгебри логіки і схемотехніки та головні відомості про роботу пакета Electronics Workbench, які подані у Додатку. У процесі виконання цього циклу робіт студенти вивчають принципи роботи та методи дослідження елементарних логічних схем та схем середнього ступеня інтеграції, на основі яких будують сучасні пристрої ЕОМ. Другий цикл лабораторних робіт передбачає вивчення і дослідження структури персонального комп’ютера (ПЕОМ), периферійних пристроїв, програмного керування ПЕОМ і не охоплений цим посібником.

ЗАГАЛЬНІ МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ

Виконання лабораторних робіт спрямоване на практичне засвоєння студентами теоретичного матеріалу лекційного курсу “Архітектура ЕОМ”.

Перш ніж допустити студента до виконання лабораторної роботи, викладач перевіряє його теоретичну підготовку з теми роботи, знання методики проведення лабораторної роботи та наявність оформленого звіту про попередню роботу.

Лабораторну роботу виконують згідно з планом, що наведений в інструкції. Для виконання окремих робіт студенти отримують індивідуальні завдання від викладача.

Звіти про виконання лабораторних робіт потрібно оформлювати у текстовому редакторі Microsoft Word або OpenOffice.org Writer окремими файлами з іменем:

“Прізвище студента”_”ас№лабораторної роботи”.doc.

У звіті необхідно подати:

- 1) назву та мету роботи;
- 2) хід виконання роботи з детальним описом методики її проведення;
- 3) логічні перетворення, що виконують для побудови комбінаційних схем;
- 4) зображення досліджуваних схем та результати дослідження;
- 5) висновки.

За результатами виконаної лабораторної роботи, які наведені у звіті, відбувається захист лабораторної роботи.

ЛАБОРАТОРНА РОБОТА №1

Тема: Побудова і дослідження простих логічних схем

Мета роботи. Освоїти роботу з пакетом Electronics Workbench (EWB). Ознайомитися з можливостями пакета щодо синтезу логічних схем. Застосувати набуті знання для побудови простих логічних схем за відомими записами з алгебри логіки. Перевірити роботу схеми і скласти таблицю істинності.

Теоретичні відомості. Усі пристрої ЕОМ складаються з елементарних логічних схем. Їхнє функціонування ґрунтується на законах і правилах алгебри логіки, яка оперує двома поняттями: істина і фальш. Основу цієї алгебри становлять дві бінарні операції (*кон'юнкція* та *диз'юнкція*) і одна унарна (*заперечення*). Крім цих трьох, вводять і інші, однак доведено, що будь-яку з них можна виразити за допомогою формули, у якій використано тільки три базові. (Наприклад, для функції від двох змінних таких логічних функцій є 16).

Мовою алгебри логіки будь-яку функцію зображають у вигляді таблиці відповідності всіх можливих логічних змінних та вихідних логічних функцій. Це так звана *таблиця істинності*.

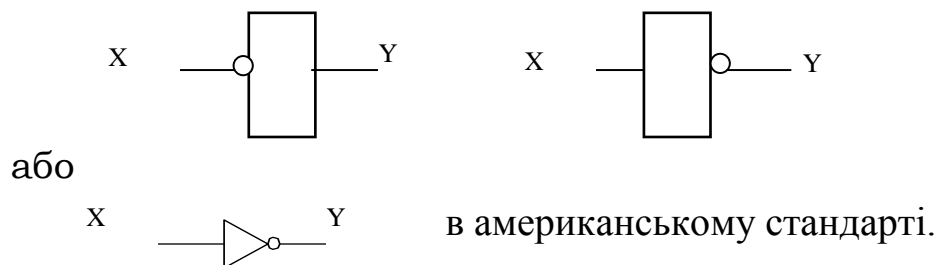
Розглянемо головні логічні функції.

Логічне заперечення. Позначають НЕ.

Означення. Логічним запереченням НЕ змінної X є логічна функція Y , яка істинна тільки тоді, коли X хибна, і навпаки. Інша назва: *інверсія*.

У символах алгебри логіки записують: $Y = \bar{X}$. Іноді використовують також позначення X' .

Графічно на схемах позначають кружечком на вході, чи виході логічного символу (за стандартом Міжнародної електротехнічної комісії МЕК 117-15А):



Таблиця істинності:

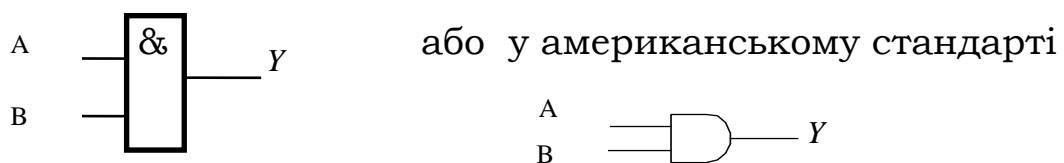
| | | |
|---|---|---|
| X | 0 | 1 |
| Y | 1 | 0 |

Логічне множення. Позначають І.

Означення. Логічним множенням двох змінних A і B є логічна функція Y , яка істинна тільки тоді, коли одночасно істинні входні змінні. Інша назва: *кон'юнкція*.

Запис: $Y = A * B$, або $Y = A \wedge B$.

Графічне зображення:



Таблиця істинності:

| | | | | |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| Y | 0 | 0 | 0 | 1 |

Логічне додавання. Позначають АБО.

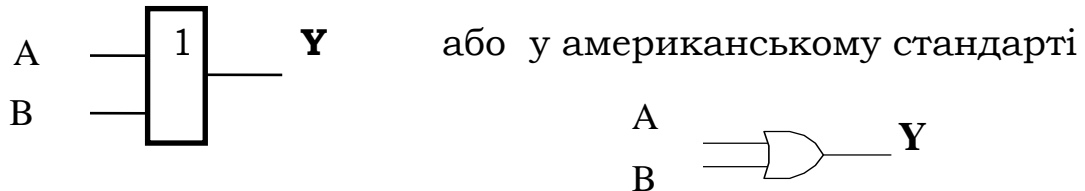
Означення. Логічною сумою змінних A і B є логічна функція Y , яка істинна, якщо хоча б одна із вхідних величин істинна.

Інша назва: *диз'юнкція*.

Запис:

$$Y = A + B, \text{ або } Y = A \vee B.$$

Графічне зображення:



Таблиця істинності:

| | | | | |
|---|---|---|---|---|
| A | 0 | 0 | 1 | 1 |
| B | 0 | 1 | 0 | 1 |
| Y | 0 | 1 | 1 | 1 |

У реальних схемах символічні ключі замінюють діодами, транзисторами чи просто використовують напилені p - n переходи у схемах більшої інтеграції. Схеми також можуть об'єднуватись відповідно до потреби реалізації певної логічної функції. Наприклад, дуже часто застосовують елементи

АБО-НЕ (стрілка Пірса)

чи І-НЕ (штрих Шеффера):



Функції, утворені із логічних змінних, можна перетворювати відповідно до правил або законів алгебри логіки.

Приклад.

1. Логічний добуток будь-якого аргументу на нуль завжди дорівнює нулю.
2. Логічний добуток будь-якого аргументу на 1 дорівнює значенню самого аргументу.
3. Логічний добуток аргументу з його інверсією дорівнює нулю.

4. Правила де Моргана (закони інверсії) :

$$\overline{A * B} = \bar{A} + \bar{B}$$

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

Ці правила і закони застосовують для спрощення логічних функцій і зведення їх до вигляду, що полегшує схемну реалізацію.

План виконання лабораторної роботи.

1. Вивчити головні відомості про роботу прикладного пакета EWB, які подано у Додатку 1. Активізувати пакет і ознайомитися з загальним виглядом робочої панелі пакету EWB (рис. 1, 2).
2. За допомогою відповідних елементів І, НЕ, АБО синтезувати у робочому полі комбінаційну схему, яка відповідає варіанту алгебричного запису. Варіанти завдань подані у Додатку 2.
3. Вхід синтезованої схеми приєднати до вихідних клем молодших розрядів генератора слів, а вихід – до аналізатора логічних рівнів або індикатора (рис. 2).

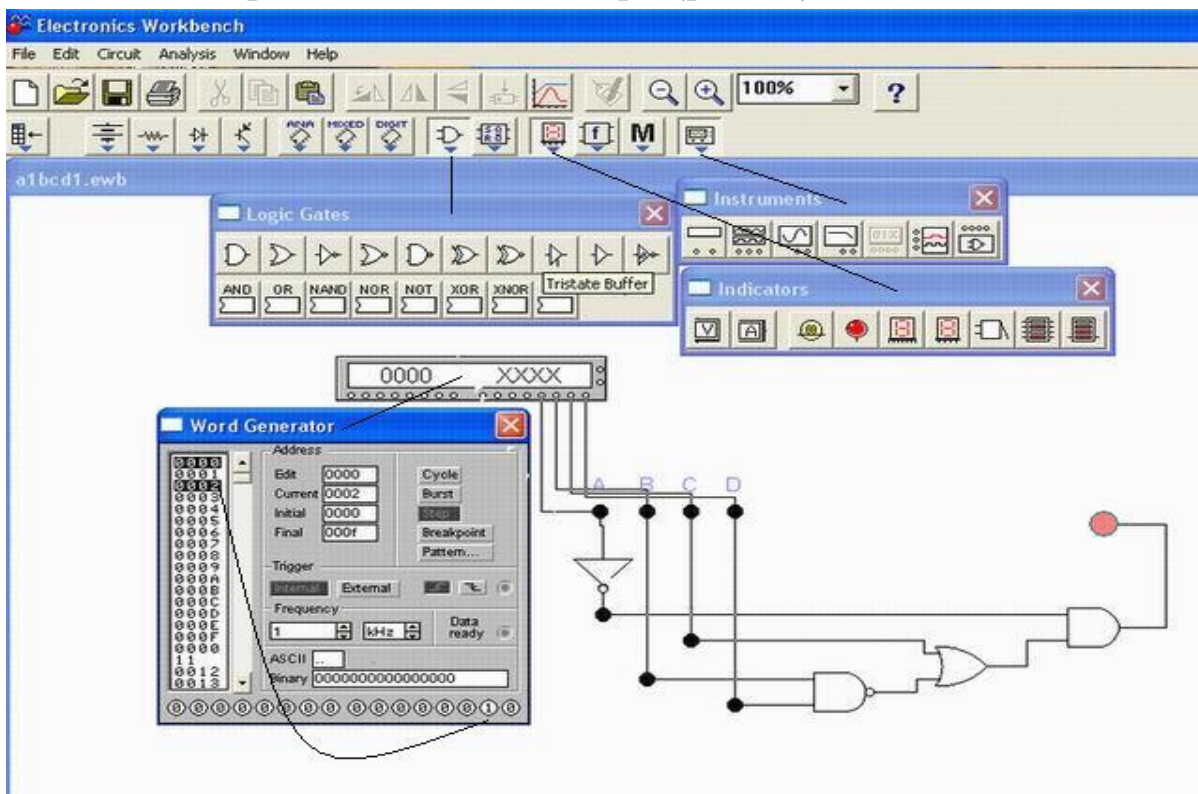


Рис. 1. Загальний вигляд робочої панелі пакету EWB.

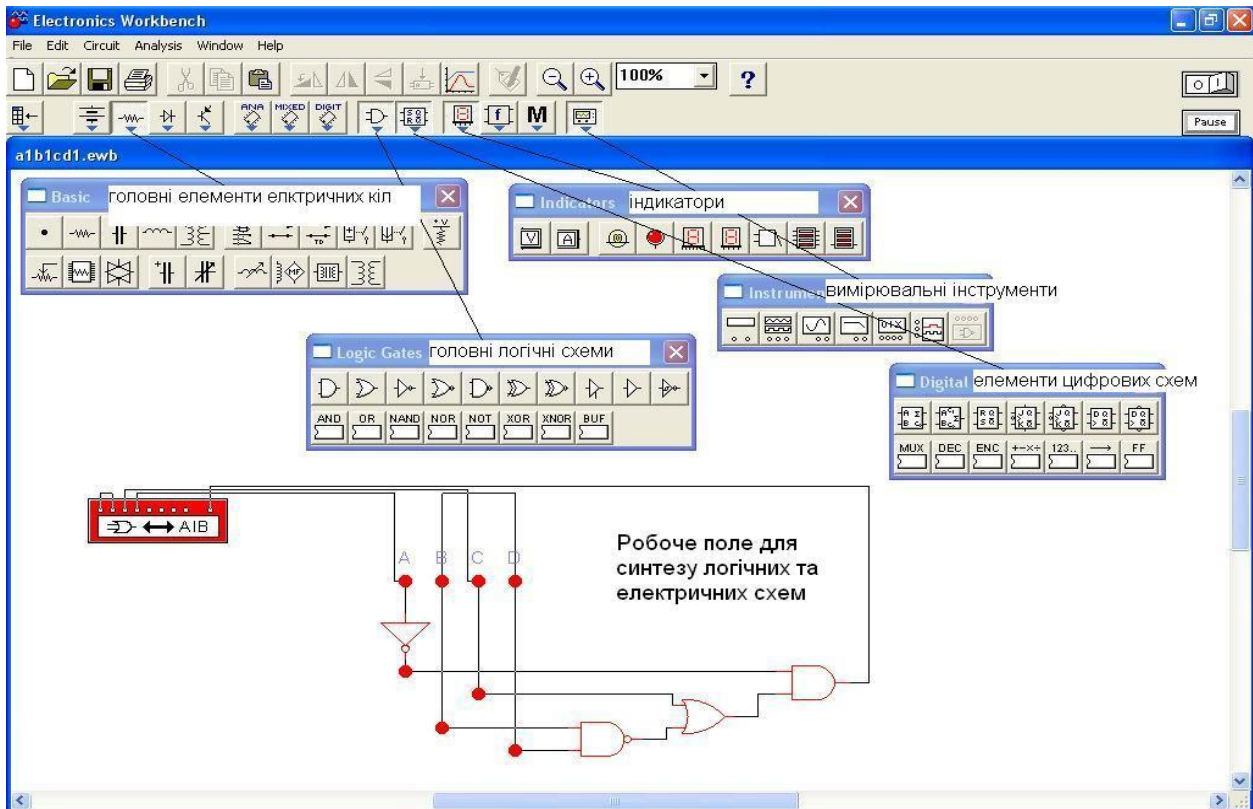


Рис. 2. Приклад приєднання генератора слів та індикатора рівнів до синтезованої логічної схеми.

4. Деталізувати зображення генератора слів (натиснути на зображенні на праву кнопку мишки і виконати пункт меню Open). На відповідних місцях комбінаційної матриці генератора слів записати усі можливі комбінації вхідних сигналів.
5. У випадку приєднання до вихідних клем аналізатора логічних рівнів деталізувати зображення аналізатора (натиснути на зображенні на праву кнопку мишки і виконати пункт меню Open).
6. За допомогою клавіші STEP генератора слів покроково перебрати усі комбінації вхідних сигналів. Результати роботи схеми спостерігати за допомогою аналізатора логічних рівнів або індикатора і записати їх у таблицю істинності.
7. Дослідити роботу генератора слів з використанням клавіш CYCLE, BURST та адресних полів INITIAL (початкова адреса) та FINAL (кінцева адреса вхідного коду).

8. Оформити звіт про виконання лабораторної роботи у якому записати:
- тему, мету та варіант завдання лабораторної роботи;
 - прізвище, ініціали та назву групи студента, що виконав роботу;
 - зображення досліджуваної схеми;
 - короткий опис методики синтезу комбінаційної схеми та вибору необхідних приладів;
 - логічну таблицю істинності, отриману в результаті дослідження схеми;
 - висновки.
9. Звіт оформити у вигляді файла з іменем: Прізвище_ac1.doc.

ЛАБОРАТОРНА РОБОТА №2

Тема: Побудова і дослідження шифраторів та дешифраторів

Мета роботи. З використанням можливості пакета EWB побудувати логічні схеми шифраторів та дешифраторів на базі логічних елементів. Перевірити роботу схем та створити макроеlementи кожної схеми.

Теоретичні відомості.

Шифратор – це вузол ЕОМ, що виконує операцію формування відповідного двійкового коду в разі появи сигналу на одному з входів шифратора. У обчислювальній техніці застосовують переважно багатоімпульсні шифратори, які дозволяють кодувати дані при записуванні програм на носії інформації чи дані, виражені якоюсь фізичною величиною (напр., напругою).

Дешифратор – пристрій який виконує перетворення n -розрядного двійкового коду в однорозрядний з основою $p=2^n$, тобто функцію обернену до шифрування. Використовують у багатьох пристроях, зокрема у пристроях керування, для розшифровування коду операції та видавання сигналів

керування у ті кола машин, які повинні працювати під час виконання цієї операції; у запам'ятовувальних пристроях для розшифрування адреси чи команди, записування або читання коду з певної комірки пам'яті.

Розглянемо роботу простої схеми дешифрування на два розряди. Як і в попередній лабораторній роботі, побудуємо її на основі комбінаційних схем (рис. 3). Пояснення роботи такого пристрою не потребує особливих зусиль. Вихідні логічні рівні отримуємо скориставшись законами алгебри логіки і врахуванням схемного сполучення елементів між собою.

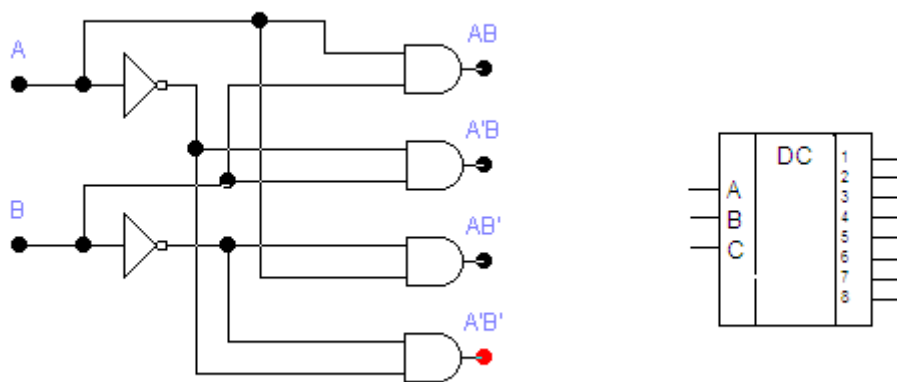


Рис. 3. Схема дворозрядного дешифратора (з правого боку показано схемне позначення трирозрядного дешифратора)

План виконання лабораторної роботи.

1. За допомогою елементарних логічних схем синтезувати у робочому полі логічну схему шифратора на вісім входів.
2. Входи синтезованих схем приєднати до вихідних клем молодших розрядів генератора слів, а виходи – до аналізатора логічних рівнів або індикатора (рис. 4).

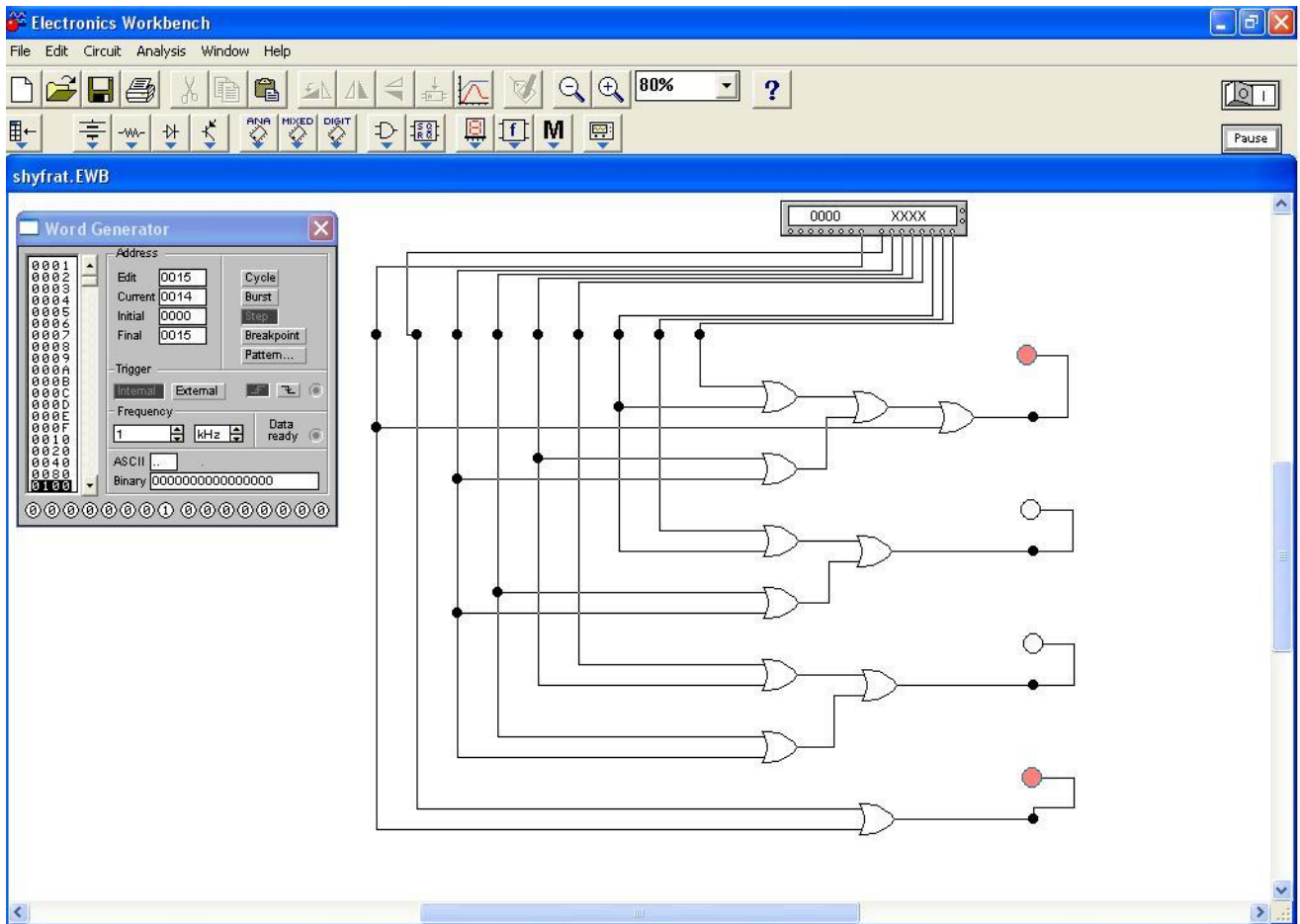


Рис.4. Приклад синтезованої схеми шифратора на дев'ять входів у робочому полі пакета EWB.

3. Деталізувати зображення генератора слів і на відповідних місцях комбінаційної матриці генератора слів записати усі можливі комбінації вхідних сигналів.
4. За допомогою клавіші STEP генератора слів покроково перебрати всі комбінації вхідних сигналів. Результати роботи схеми спостерігати за допомогою аналізатора логічних рівнів або індикатора і записати їх у таблицю істинності.
5. Створити макроелемент 3AND (процедуру створення мікроелементів див. Додаток 1). Записати у пам'ять файл зі створеним макроелементами (функція меню Create Subcircuit, рис. 5).
6. За допомогою логічних елементів синтезувати у робочому полі схему трирозрядного двійкового дешифратора (див. рис. 5).

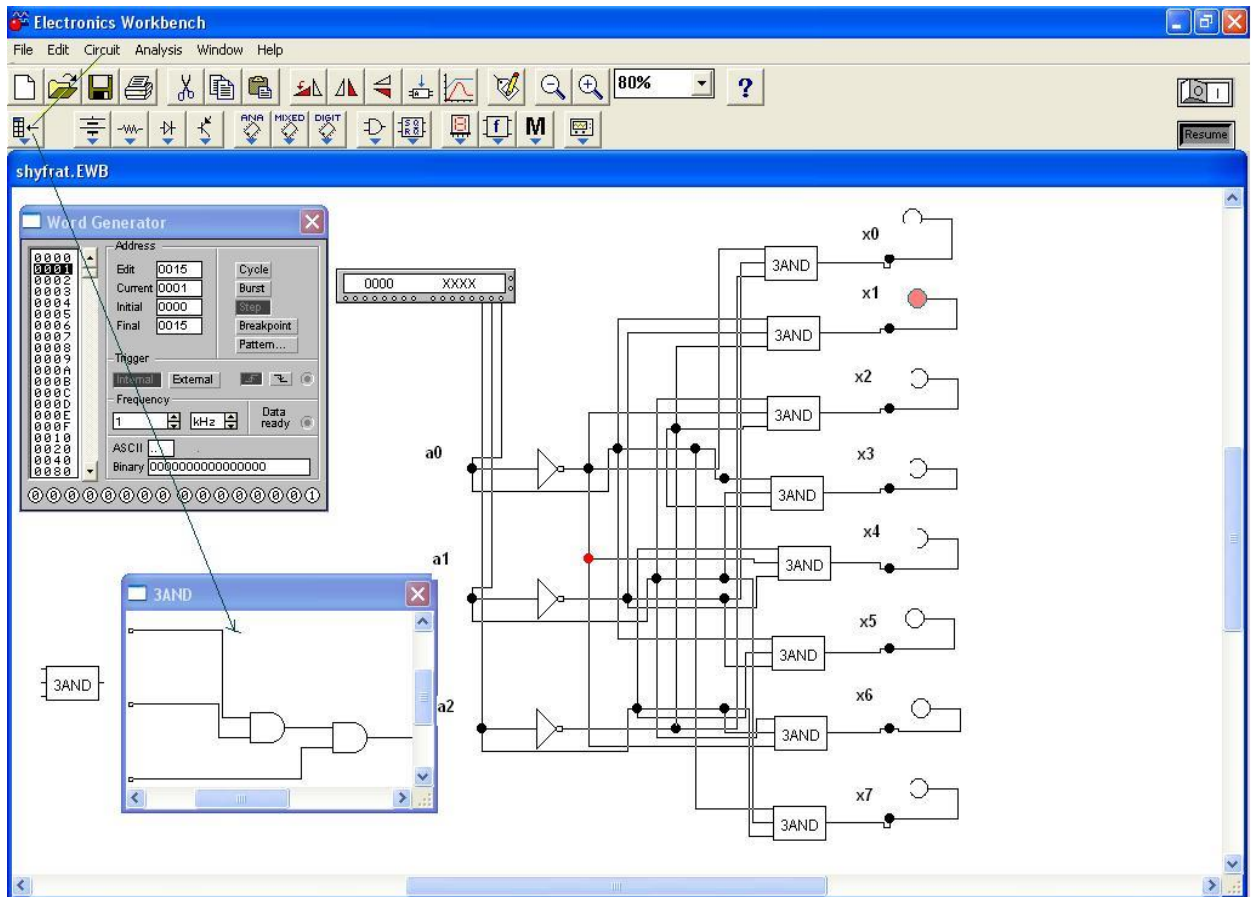


Рис. 5. Приклад синтезованої схеми дешифратора на три входи в робочому полі пакета EWB.

7. Виконати пункт 3.
8. Виконати пункт 4.
9. Оформити звіт про виконання лабораторної роботи у якому подати:
 - а. тему, мету та завдання лабораторної роботи;
 - б. прізвище, ініціали та назву групи студента, що виконав роботу;
 - с. зображення досліджуваних схем;
 - д. логічні таблиці істинності та зняті часові діаграми, отримані в результаті дослідження схеми;
 - е. висновки.
10. Звіт оформити у вигляді файла з іменем: Прізвище_ac2.doc.

ЛАБОРАТОРНА РОБОТА №3

Тема: Побудова і дослідження суматорів

Мета роботи. З використанням можливості пакета EWB побудувати логічні схеми напівсуматорів та суматорів на базі логічних елементів. Перевірити роботу схем та створити макроеlementи кожної схеми.

Теоретичні відомості.

Суматор – пристрій, що виконує сумування чисел на підставі правил порозрядного додавання з урахуванням переносів, які спрямовують у старші розряди.

Суматори класифікують:

1. за виглядом елементів, які використовують – комбінаційного та накопичувального типів;
2. за способом введення-виведення чисел – паралельної та послідовної дії;
3. за способом зображення чисел – двійкові та десяткові;
4. за способом організації перенесення – з послідовним та наскрізним перенесенням.

Розглянемо суматори комбінаційного типу (комбінація логічних елементів І, АБО, НЕ)

Сигнал на виході (сума) утворюється тільки в разі визначеної комбінації вхідних сигналів (доданків), які подаються одночасно. Після зникнення вхідних сигналів вихідний сигнал зникає (запам'ятовувальних властивостей немає). Тому такі суматори працюють з регістром, у який записується результат.

Процес сумування і перенесення з сусіднього розряду в однорозрядній сумувальній схемі розбито на дві аналогічні операції: сумування дворозрядних доданків і сумування з отриманим результатом одиниці перенесення. Кожну з функцій виконує схема, яку називають *напівсуматором*.

Розглянемо роботу напівсуматора, яку описують логічними виразами для суми:

$$\begin{aligned} S_i &= X_i Y_i' + X_i' Y_i = X_i Y_i' + X_i' Y_i + X_i X_i' + Y_i Y_i' = \\ &= X_i(X_i' + Y_i') + Y_i(X_i' + Y_i') = (X_i' + Y_i')(X_i + Y_i) = \overline{X_i Y_i}(X_i + Y_i) \end{aligned}$$

та перенесення $P_{i+1} = X_i Y_i$.

Результат для суми отримано внаслідок певних алгебричних перетворень. Схема, яка реалізує кінцевий етап перетворень показана на рис. 23.

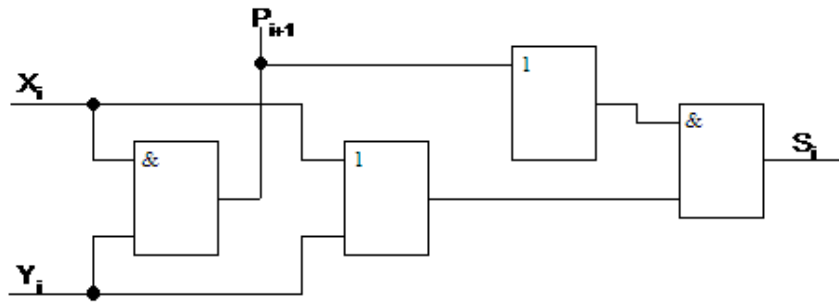
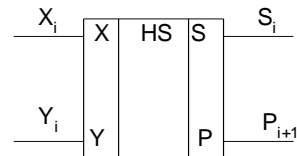
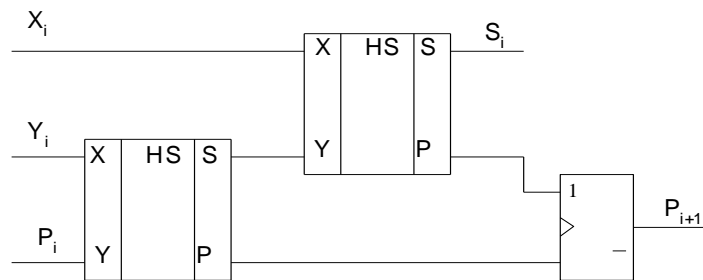


Рис. 23. Схема напівсуматора комбінаційного типу:

На схемах позначають



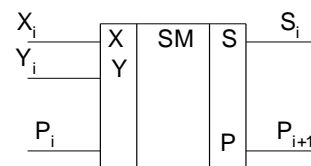
Якщо додати ще один напівсуматор, то отримаємо схему



повного суматора на три входи (рис. 24).

Рис.24. Схема повного суматора на три входи

Повний суматор на схемах позначають так:



Багаторозрядний суматор можна отримати простим послідовним з'єднанням однорозрядних суматорів (рис. 25).

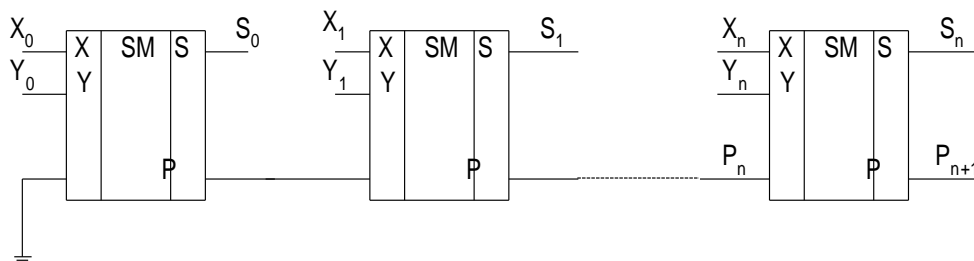


Рис. 25. Схема багаторозрядного суматора

Для зменшення часу поширення сигналу перенесення використовують схеми прискореного перенесення.

Однорозрядний суматор можна використати для виконання операції віднімання. Для цього на вхід Y подають або прямий, або обернений код доданка, який відповідає його від'ємному значенню (рис. 26).

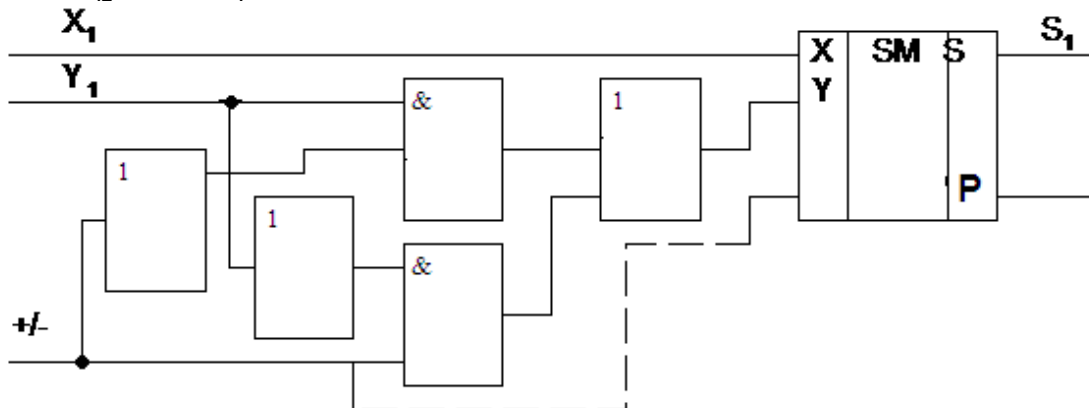


Рис. 26. Використання однорозрядного суматора для операції віднімання

Залежно від сигналу керування $+/-$ на вхід суматора подається прямий код Y (при $+/- = 0$), або обернений код Y (при $+/- = 1$). Для формування додаткового коду доданка до його інверсного коду треба додати 1 молодшого розряду. Цього досягають з'єднанням входу керування $+/-$ з входом суматора молодшого розряду.

План виконання лабораторної роботи.

1. За допомогою елементарних логічних схем синтезувати у робочому полі логічну схему напівсуматора.
2. Входи синтезованих схем приєднати до вихідних клем молодших розрядів генератора слів, а виходи – до аналізатора логічних рівнів або індикатора (рис. 27).
3. Деталізувати зображення генератора слів і на відповідних місцях комбінаційної матриці генератора слів записати усі можливі комбінації вхідних сигналів.

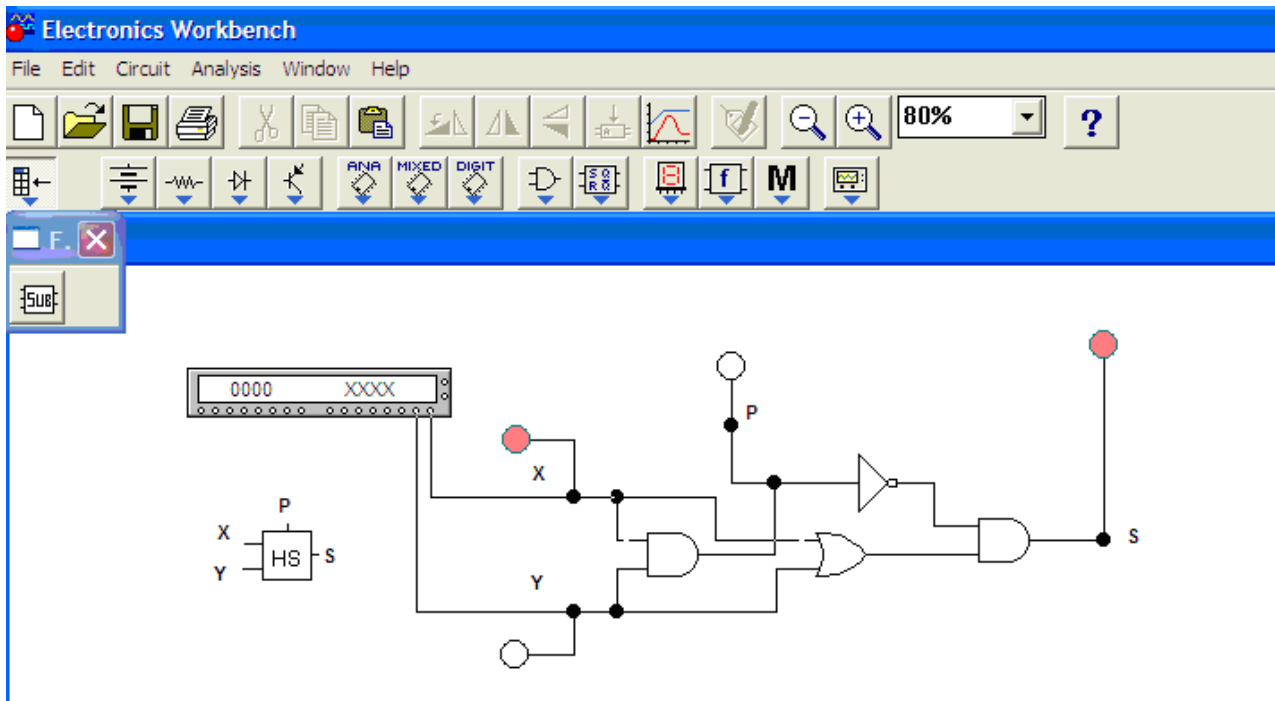


Рис.27. Приклад синтезованої схеми напівсуматора у робочому полі пакета EWB.

4. За допомогою клавіші STEP генератора слів покроково перебрати всі комбінації вхідних сигналів. Результати роботи схеми спостерігати за допомогою аналізатора логічних рівнів або індикатора і записати їх у таблицю істинності.
5. Створити макроелемент HS (процедуру створення макроелементів див. Додаток 1). Записати у пам'ять файл зі створеним макроелементами (функція меню Create Subcircuit, рис. 2д).
6. За допомогою логічних елементів та макроелементів HS синтезувати у робочому полі схему повного суматора SM (див. рис. 28).

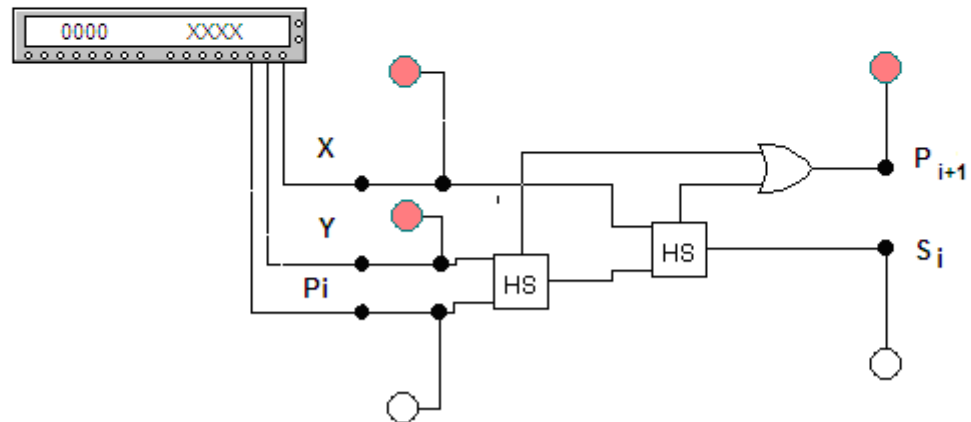
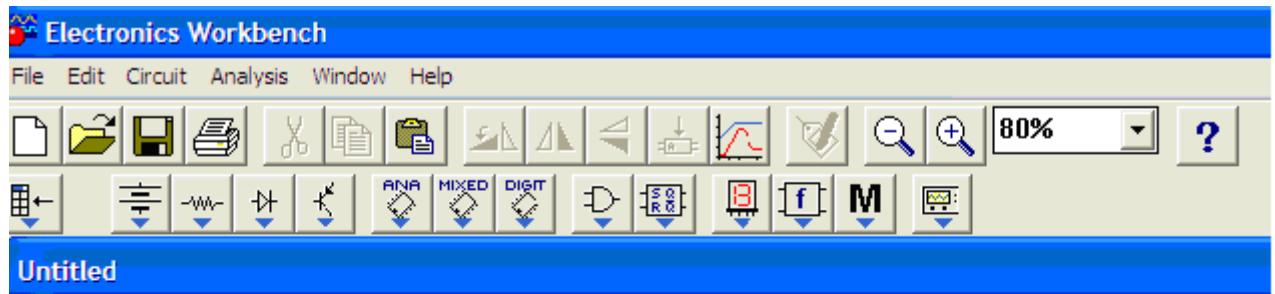


Рис. 28. Приклад синтезованої схеми повного суматора в робочому полі пакета EWB.

7. За допомогою макроелементів SM синтезувати у робочому полі схему 4-розрядного двійкового суматора (див. рис. 29).

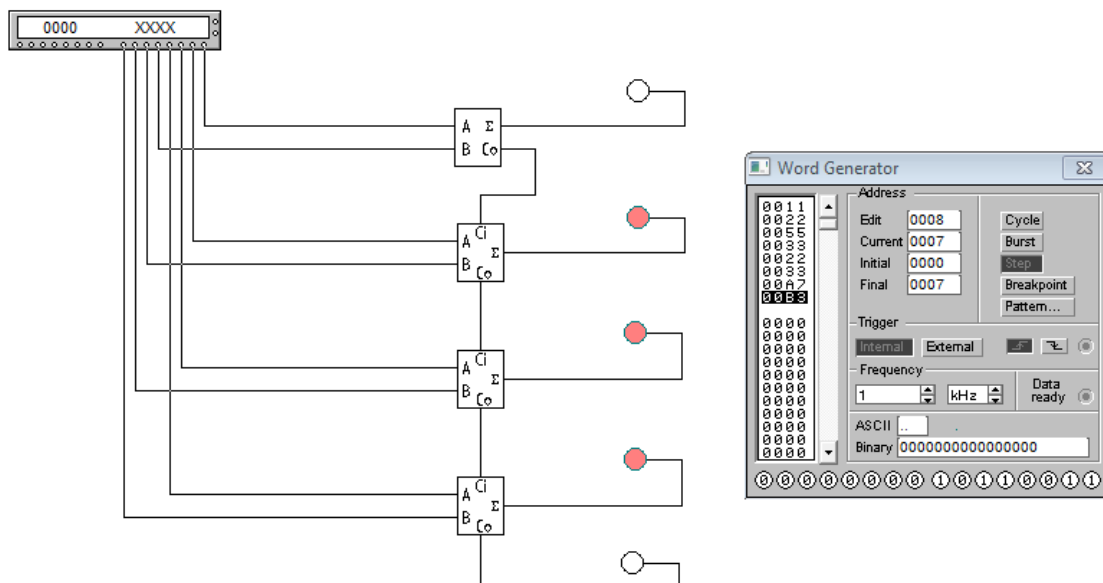


Рис. 29. Приклад синтезованої схеми 4-розрядного суматора в робочому полі пакета EWB.

8. Виконати операцію додавання двох 4-розрядних чисел. Варіант контрольного завдання у Додатку 2.
9. Виконати пункт 4.
10. Оформити звіт про виконання лабораторної роботи у якому подати:
 - a. тему, мету та завдання лабораторної роботи;
 - b. прізвище, ініціали та назву групи студента, що виконав роботу;
 - c. зображення досліджуваних схем;
 - d. отримані результати дослідження схеми суматора;
 - e. висновки.
11. Звіт оформити у вигляді файла з іменем: Прізвище_ас3.doc.

ЛАБОРАТОРНА РОБОТА №4

Тема: Побудова і дослідження елементів пам'яті на базі тригерних схем

Мета роботи. З використанням можливостей пакета EWB побудувати логічні схеми елементів пам'яті (RS-тригер, D-тригер, JK-тригер, T-тригер). Перевірити роботу схем, скласти таблиці істинності та створити макроеlementи кожної схеми.

Теоретичні відомості.

Головним елементом пам'яті ЕОМ є **тригери** – логічні пристрої, які мають два стійкі стани. Для перемикавання тригерів з одного стану в інший використовують входні логічні схеми. За способом перемикавання розрізняють такі тригери:

- RS-тригери з роздільним установленням 0 і 1;
- D-тригер із затримкою;
- JK-тригери (універсальні);
- T-тригери (лічильні);

Ці назви походять від перших букв входних сигналів: S (set – встановити); R (reset – вимкнути); T (toggle – релаксатор); J (jerk – різко увімкнути); K (kill – різко вимкнути); D (delay – затримка). Вихідний сигнал тригера прийнято позначати буквою Q.

RS-тригери з роздільним установленням 0 та 1

Найпростіший запам'ятовувальний елемент має два входи: S і R . У разі комбінації сигналів $S=1$, $R=0$ тригер буде встановлений в одиничний стан, тобто $Q=1$.

Якщо ж комбінація вхідних сигналів $S=0$, $R=1$, то тригер встановиться в нуль, $Q=0$. Комбінація $S=0$, $R=0$ залишить тригер у попередньому стані, а от комбінація $S=1$, $R=1$ буде невизначеною, отже, забороненою для такого елемента ($Q = \overline{Q} = 0$).

Функціонування RS-тригера описує вираз

$$Q(t+1) = S(t) \vee Q(t) \overline{R}(t),$$

де $S(t)R(t)=0$, t – момент часу, який передуює зміні стану.

Такі тригери реалізують на логічних елементах І-НЕ, АБО-НЕ, з'єднуючи їхні входи і виходи навхрест. На схемах їх зображають прямокутником, у якому з лівого боку позначають вхідні сигнали S , R , а з правого – вихідні. Вхід S називають установлювальним, а вхід R – скидальним. Приклади таких RS-тригерів показано на рис.6.

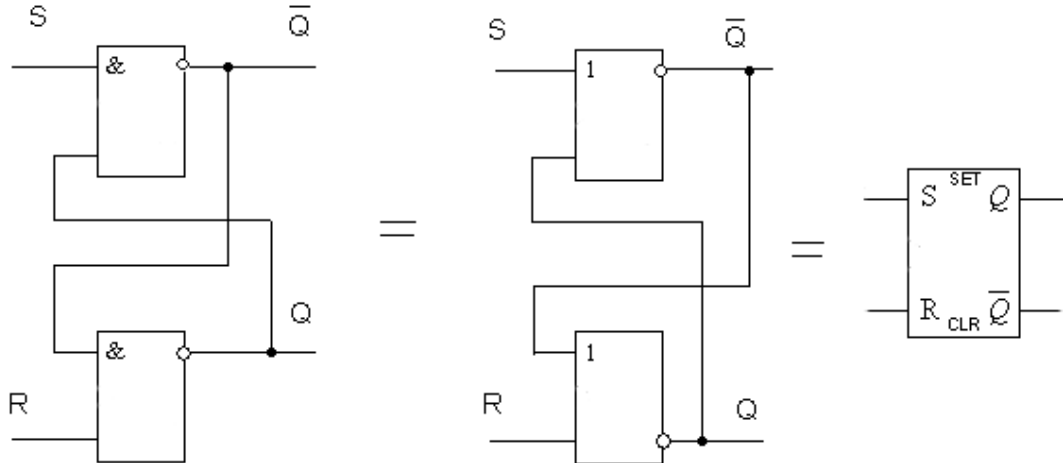


Рис. 6. RS - тригери на базі елементів І та АБО

Іноді вводять синхронізацію зміни стану тригера (позначають вхід C (clocking – синхронізація)). При $C=1$ тригер перемикається за законом RS-тригера, при $C=0$ він зберігає попередній стан.

D-тригер (затримки)

Такий тригер має один інформаційний вхід і вхід для синхроімпульсу. Головне призначення такого тригера – затримка і збереження сигналу, який подають на вхід за умови $C=1$. На рис.7 показана схема синхронізованого двотактного D-тригера, який затримує сигнал на один період; і його описує формула:

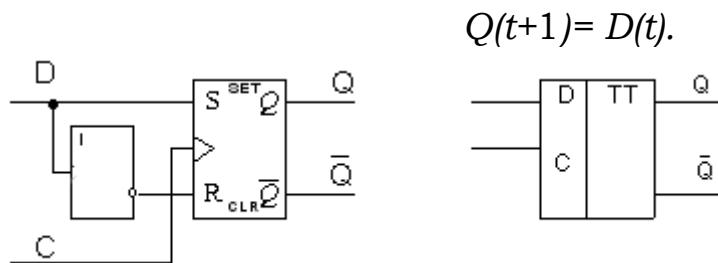


Рис. 7. Синхронізований *D*-тригер

Різновидом *D*-тригера є *DV*-тригер (*V*-value – вентиль), у якому через вхід керування *V* дозволено перемикання, або тригер не реагує на перемикання при *V*=0.

JK-тригер (універсальний)

Працює за принципом *RS*-тригера, проте в ньому комбінація $J=K=1$ не заборонена, у цьому разі він змінює свій стан на протилежний до того, у якому був. Переважно його реалізують за двоступеневою схемою (див. рис.8).

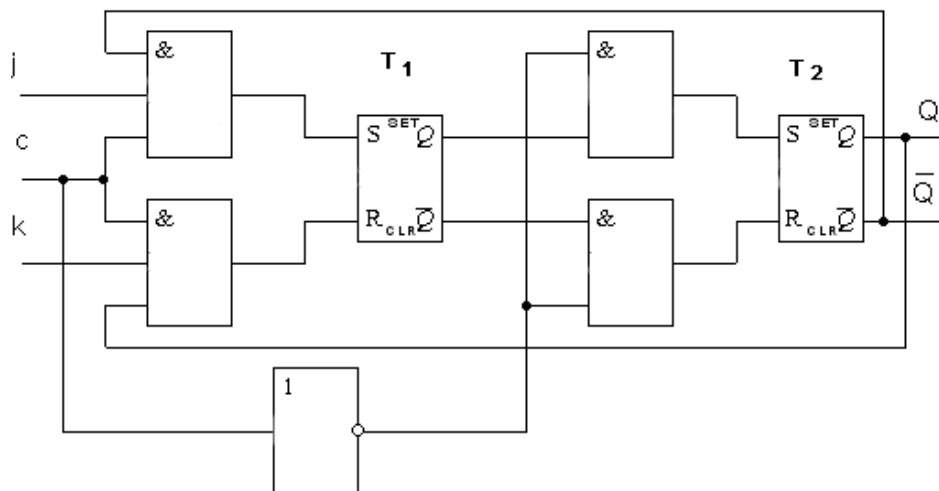
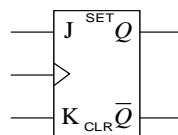


Рис. 8. Схемна реалізація *JK*-тригера

На схемах позначають так:



При $C=1$ відбувається записування інформації в *T1*, при $C=0$ – переписування інформації з *T1* в *T2*. Функція переходів *JK* за умови $J=K=1$ має вигляд

$$Q(t+1) = Q(t) \bar{K}(t) \vee \bar{Q}(t) J(t).$$

Тригер типу *JK* є універсальним тому, що може виконувати функції *RS*-тригера (при роздільному надходженні сигналів *J* і *K*), *T*-тригера (при одночасному поданні сигналів *J* і *K*), *D*-тригера (при поданні сигналу від входу *J* через інвертор на вхід *K*).

T-тригер (лічильний)

Цей тригер змінює свій стан у разі надходження кожного вхідного імпульсу. Він може бути реалізований на базі JK-тригера. При значеннях змінних $J=K=1$ сигналом C можна змінювати стан тригера. Позначають як на рис. 9.

T-тригер реалізує таку функцію:

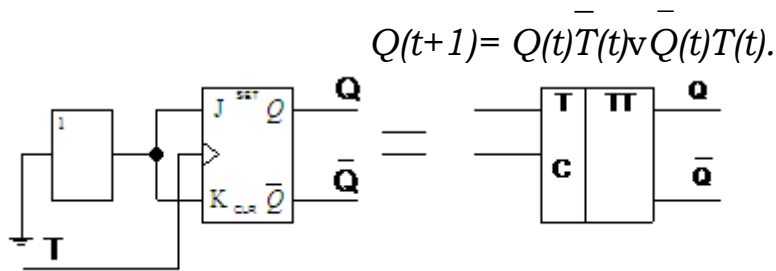


Рис. 9. Схемна реалізація лічильного тригера

За допомогою описаних логічних пристроїв можна будувати різні блоки чи пристрої ЕОМ. До них належать: регістри, лічильники, шифратори, дешифратори, мультиплексори, суматори та арифметико-логічні пристрої як цілочислової так і плаваючої арифметики.

План виконання лабораторної роботи.

За допомогою логічних елементів І, НЕ, АБО синтезувати у робочому полі комбінаційні схеми тригерів (рис. 10).

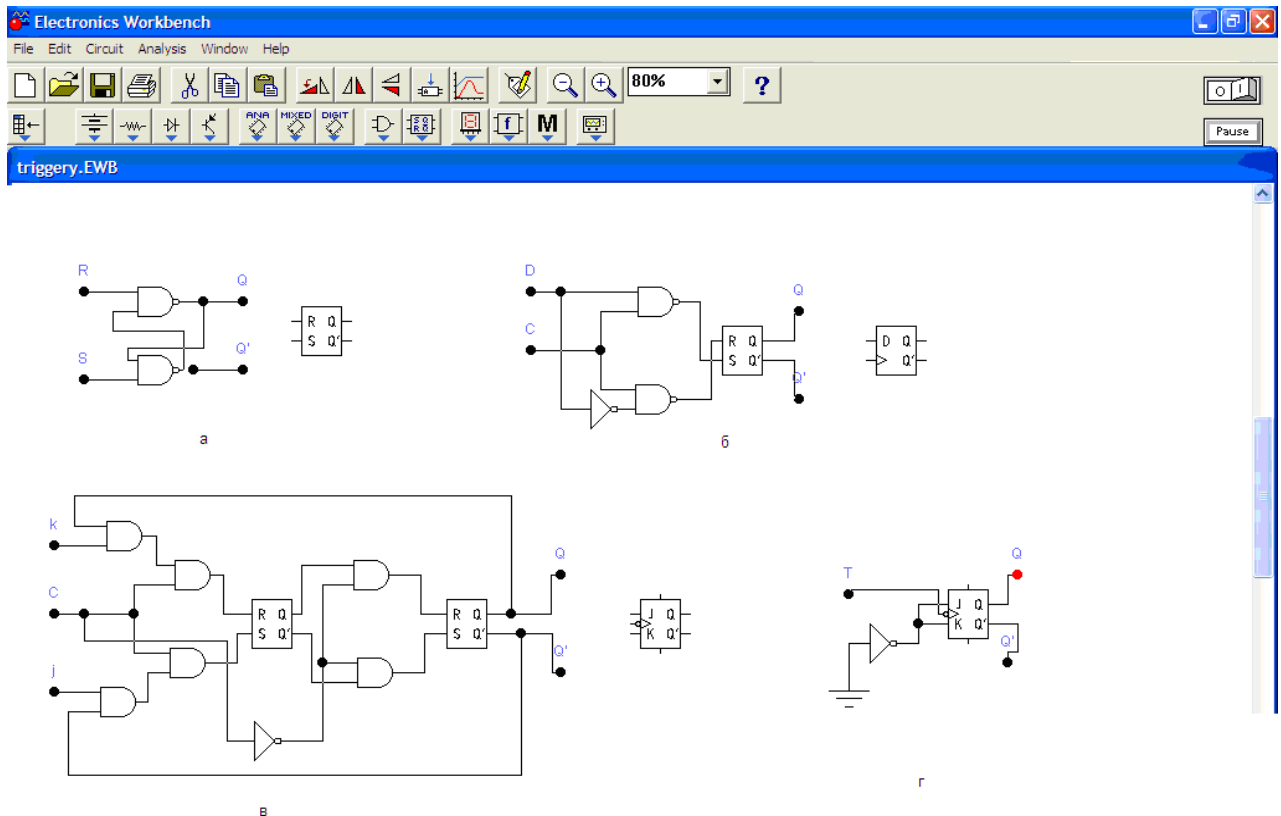


Рис. 10. Приклади синтезованих тригерних схем у робочому полі пакета EWB: а) – RS-тригер, б) – D-тригер, в) – JK-тригер, г) – T-тригер .

1. Входи синтезованих схем приєднати до вихідних клем молодших розрядів генератора слів, а виходи – до аналізатора логічних рівнів або індикатора (рис. 11).
2. Деталізувати зображення генератора слів (натиснути на зображенні на праву клавішу мишки і виконати пункт меню Open). На відповідних місцях комбінаційної матриці генератора слів записати всі можливі комбінації вхідних сигналів.
3. Синхронізувальний імпульс подавати від клем Data Ready генератора слів. Під час дослідження D-тригера на вхід D подавати серію з логічних 1 та 0. Під час дослідження T-тригера на вхід T подавати логічні 1.
4. За допомогою клавіші STEP генератора слів покроково перебрати усі комбінації вхідних сигналів. Результати роботи

схеми спостерігати за допомогою аналізатора логічних рівнів або індикатора і записати їх у таблицю істинності.

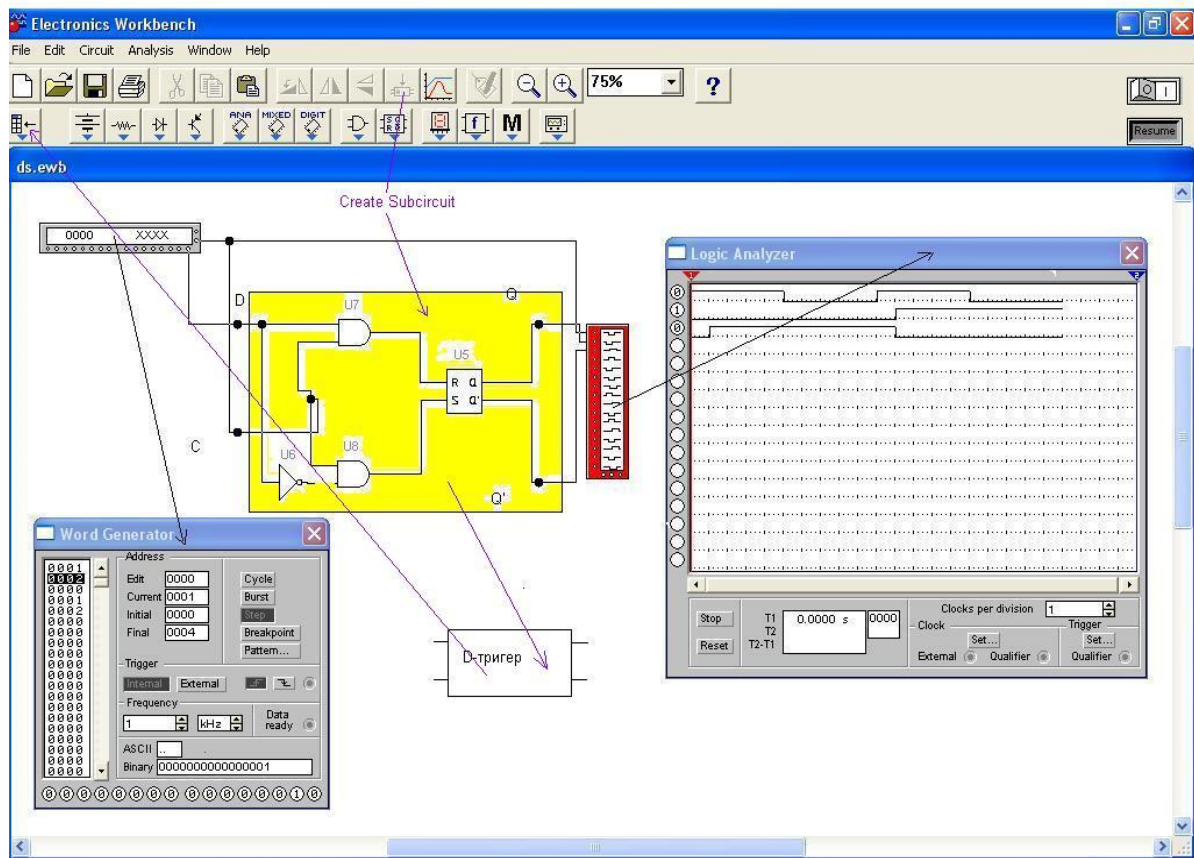


Рис. 11. Приклад приєднання генератора слів та індикатора рівнів до синтезованої логічної схеми.

5. Створити макроеlementи – JK-тригер, D-тригер, T-тригер. Записати у пам'ять файли зі створеними макроеlementами (функція меню Create Subcircuit, див. рис. 11).
6. Оформити звіт про виконання лабораторної роботи, у якому записати:
 - тему, мету та завдання лабораторної роботи;
 - прізвище, ініціали та назву групи студента, що виконав роботу;
 - зображення досліджуваних схем;
 - логічні таблиці істинності та зняті часові діаграми, отримані в результаті дослідження схеми;
 - висновки.
7. Звіт оформити у вигляді файла з іменем: Прізвище_ac4.doc.

ЛАБОРАТОРНА РОБОТА №5

Тема: Побудова і дослідження регістрових схем

Мета роботи. З використанням можливостей пакета EWB побудувати логічні схеми регістрових схем на базі тригерних елементів пам'яті. Перевірити роботу схем та створити макроеlementи кожної схеми.

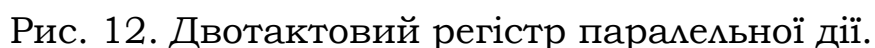
Теоретичні відомості.

Регістр – це вузол ЕОМ, який виконує тимчасове збереження та перетворення інформації. Регістри будують на основі тригерних схем. Кількість тригерів визначає розрядність слів, які записують чи зберігають у регістрі. Регістри є послідовні і паралельні, одно- і двотактні, зсувні і перетворювальні. Розглянемо кілька прикладів схем регістрів. Побудуємо регістр, наприклад, трирозрядний. Очевидно, що для цього треба мінімум три тригери, які будемо використовувати для запису кожного розряду.

Використаємо звичайні RS-тригери. Ліворуч організуємо синхронний запис інформації, а праворуч – формування прямого та оберненого коду. Перед записом інформації всі тригери сигналом керування R поставимо в нуль ($Q=0$). Запис у тригери виконується за тактовим імпульсом T_{i1} . Власне ці два сигнали (R і T_{i1}) визначають тип регістра, який називають двотактовим паралельної дії (рис. 12). Код регістра видають за допомогою другого і третього тактових імпульсів: T_{i2} – прямий код, T_{i3} – обернений код.

Якщо на вхід можна подавати парафазний код (тобто вхідне значення подають у прямому й оберненому коді), то відпадає потреба у такті установлення в "0", тобто отримаємо однотактний паралельний регістр.

Регістр є дуже зручним пристроєм для "зсування" інформації праворуч чи ліворуч або перетворення послідовного коду у паралельний (рис. 13).



25

Ще розглянемо роботу однотактового регістра зі зсуванням числа праворуч, який виконує функцію перетворення паралельного коду в послідовний (рис. 14).

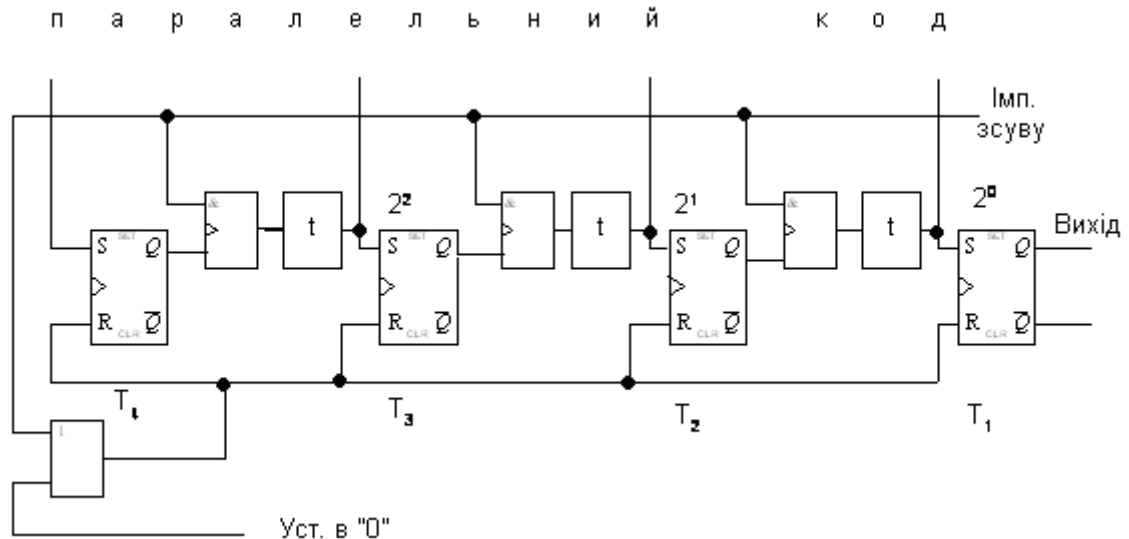


Рис. 14. Однотактовий регістр для зсування коду праворуч

Тут для спрощення схеми ланки зчитування числа пропущено. Елемент t виконує функцію затримки сигналу на час дії імпульсу зсуву. Для зсуву числа потрібно подати n імпульсів зсуву.

Нехай у регістрі є код 1011. Стан тригерів після кожного імпульсу зсуву t_i буде змінюватися відповідно до таблиці:

| Імпульс | t_1 | t_2 | t_3 | t_4 | |
|---------|-------|-------|-------|-------|--------------------------------|
| T_4 | 1 | 0 | 0 | 0 | $t_2 = t_1 + t_{\text{затр.}}$ |
| T_3 | 0 | 1 | 0 | 0 | $t_3 = t_2 + t_{\text{затр.}}$ |
| T_2 | 1 | 0 | 1 | 0 | $t_4 = t_3 + t_{\text{затр.}}$ |
| T_1 | 1 | 1 | 0 | 1 | |

Якщо використовувати парафазний (такий, у якому присутні одночасно прямі та інверсні значення) код, то нема потреби у лініях затримки.

Затримку можна виконати також за допомогою тригера, що значно підвищує надійність роботи схеми. Зсув можна робити як праворуч, так і ліворуч. Такі регістри називають *реверсивними*.

Якщо вихід T_1 подати на вхід T_4 то легко зауважити, що в разі подання імпульсів зсуву інформація у регістрі буде циркулювати. Такі регістри називають *кільцевими*.

План виконання лабораторної роботи.

1. За допомогою D-тригерних схем синтезувати у робочому полі логічні схеми чотирирозрядних регістрів паралельної та послідовної дії.
2. Входи синтезованих схем приєднати до вихідних клем молодших розрядів генератора слів, а виходи – до аналізатора логічних рівнів або індикатора (рис. 15).

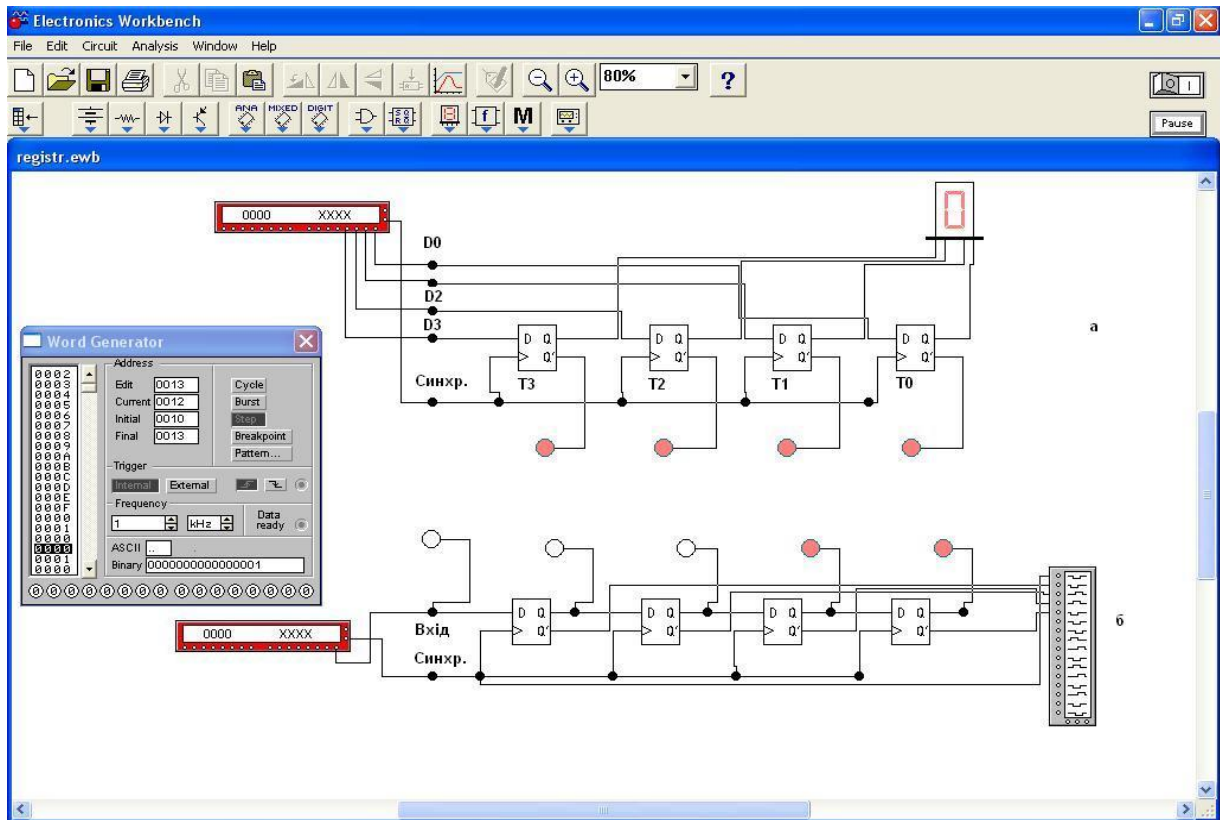


Рис. 15. Приклади синтезованих чотирирозрядних регістрових схем у робочому полі пакета EWB. Регістри: а) – паралельний, б) – послідовний.

3. Деталізувати зображення генератора слів і на відповідних місцях комбінаційної матриці генератора слів записати всі можливі комбінації входних сигналів.
4. Синхронізувальний імпульс подавати від клем Data Ready генератора слів. Під час дослідження регістра паралельної дії на входи D0-D3 подавати відповідні виходи генератора слів. Під час дослідження регістра послідовної дії (зсувального) на вхід подавати серію з логічних 1 та 0.

5. За допомогою клавiші STEP генератора слiв покроково перебирати всi комбiнацiї вхiдних сигналiв. Результати роботи схеми спостерiгати за допомогою аналiзатора логiчних рiвнiв або iндикатора i записати їх у таблицю iстинностi.
6. Створити макроеlementи – R-регiстр для регiстра паралельної дiї та S-регiстр для регiстра послiдовної дiї. Записати у пам'ять файли зi створеними макроеlementами (функцiя меню Create Subcircuit).
7. Оформити звіт про виконання лабораторної роботи у якому подати:
 - тему, мету та завдання лабораторної роботи;
 - прiзвище, iнiцiали та назву групи студента, що виконав роботу;
 - зображення досліджуваних схем;
 - логiчні таблицi iстинностi та знятi часовi дiаграми, отриманi в результатi дослідження схеми;
 - висновки.
8. Звіт оформити у виглядi файла з iменем: Прiзвище_ac5.doc.

ЛАБОРАТОРНА РОБОТА №6

Тема: Побудова i дослідження лiчильних схем

Мета роботи. З використанням можливостей пакета EWB побудувати логiчні схеми лiчильних схем на базi тригерних елементiв пам'ятi. Перевiрити роботу схем та створити макроеlementи кожної схеми.

Теоретичнi вiдомостi.

Лiчильник – пристрiй, призначений для пiдрахунку кiлькостi iмпульсiв. Лiчильники бувають пiдсумовувальнi, вiднiмальнi та реверсивнi. Реверсиний лiчильник залежно вiд перекомутацiї може бути пiдсумовувальним або вiднiмальним. Будують лiчильники на основi тригерiв, використовують у пристроях керування та в арифметичних пристроях для рахунку номерiв команд, кiлькостi циклiв програми, кiлькостi тактiв у разi

множення і ділення, а також як суматори. Приклад 4-розрядного підсумовувального двійкового лічильника, зібраного на синхронних jk-тригерах, які працюють у режимі Т-тригерів, показано на рис.16.

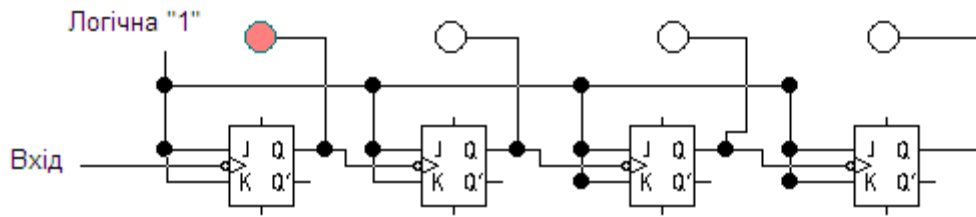


Рис. 16. Схема 4-розрядного підсумовувального двійкового лічильника

Такий лічильник порахує до 16 (1111) і знову стане в "0". Максимальна кількість імпульсів, яку може підрахувати двійковий лічильник, що складається з n розрядів, дорівнює $2^n - 1$.

Віднімальний лічильник отримаємо тоді, коли сигнал знімати з інверсного виходу тригера.

Лічильники, які однаково можна використовувати як для додавання, так і для віднімання імпульсів, називають *реверсивними*.

Входи S , R застосовують для занесення початкового коду у лічильник або скидання в "0".

Іноді потрібно отримати значення $K \neq 2^n$. Тоді будують спеціальні перерахункові схеми, де від певних розрядів уводять обернений зв'язок. Перерахункові схеми з $K=10$ називають декадними лічильниками, які застосовують для побудови десяткових лічильників (рис. 17). На схемі елемент D «диференціює» вихідний сигнал тригера, тобто перетворює перепад логічних рівнів у імпульс.

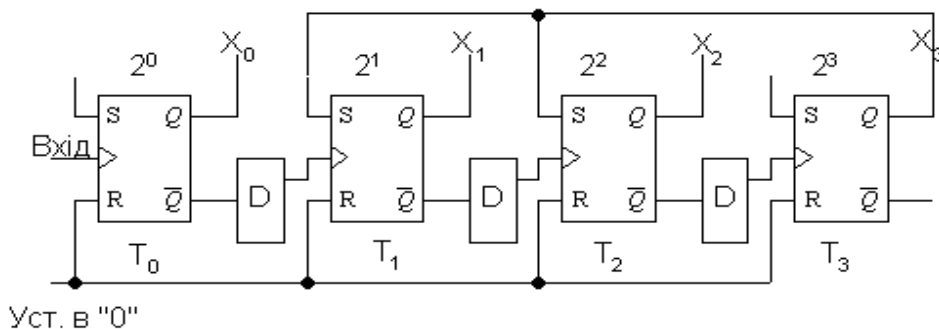


Рис. 17. Декадний лічильник

У разі подання перших семи імпульсів схема працює у звичайному режимі. Восьмий імпульс установить тригери T_0 , T_1 і T_2 в стан "0", а тригер T_3 – у стан "1". У цьому випадку внаслідок оберненого зв'язку сигнал перенесення з T_3 встановить в одиничний стан T_1 і T_2 : отримаємо стан лічильника 1110. Дев'ятий імпульс установить в "1" T_0 , а десятий скине в "0" усі тригери схеми. Після цього схема знову готова рахувати до 10.

План виконання лабораторної роботи.

1. За допомогою Т-тригерних схем синтезувати у робочому полі логічну схему чотирирозрядного двійкового лічильника та десяткового лічильника (рис. 18). Для правильного функціонування схеми десяткового лічильника обернений зв'язок з останнього каскаду необхідно запровадити через спеціальну схему перетворення логічного рівня у імпульс, як це показано на рис.18б.
2. Імпульси на вхід лічильника подавати від клеми Data Ready генератора слів. Під час дослідження лічильників на вхід JK подавати логічну 1, а виходи приєднати до аналізатора логічних рівнів або індикатора (рис.18).
3. За допомогою клавіші STEP генератора слів покроково перебирати всі комбінації вхідних сигналів. Результати роботи схеми спостерігати за допомогою аналізатора логічних рівнів або індикатора і записати їх у таблицю істинності.

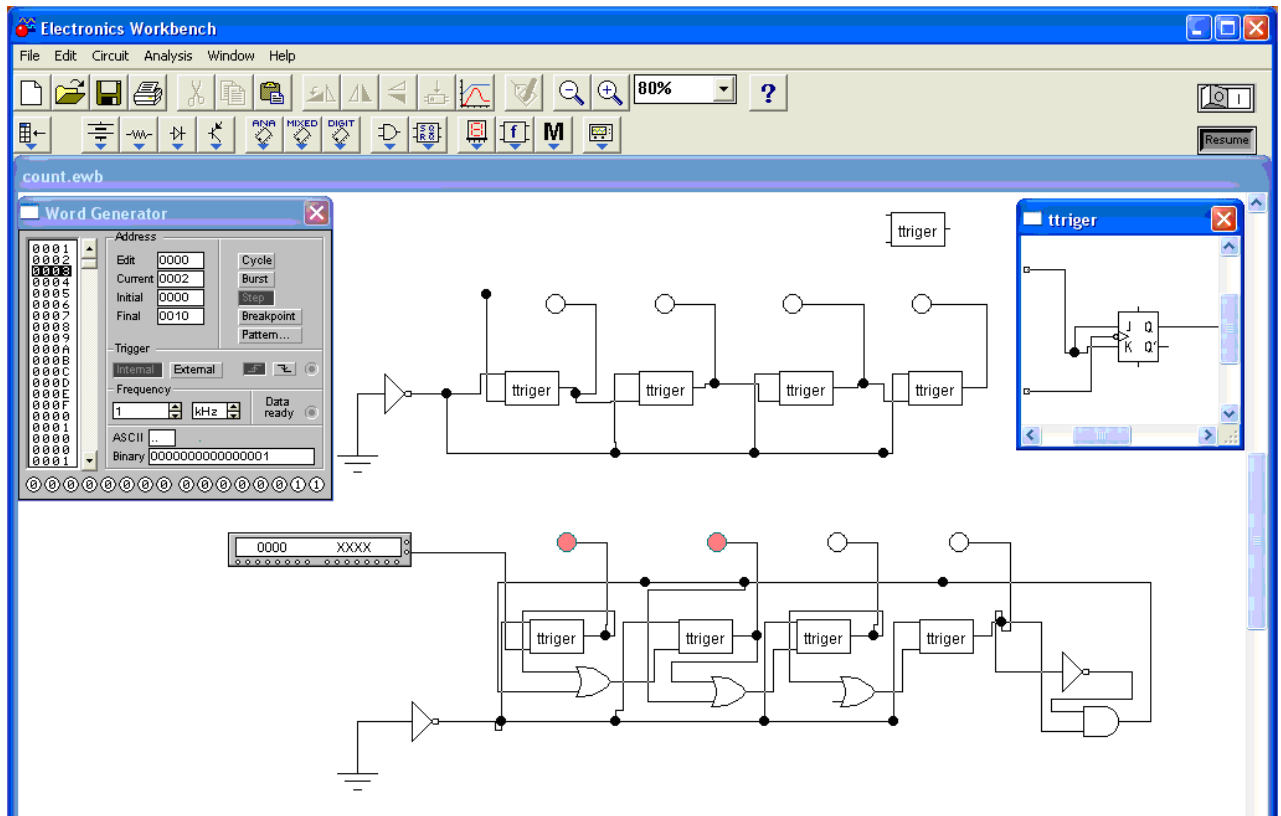


Рис. 18. Приклад синтезованих схем лічильників сигналів в робочому полі пакета EWB: а) – двійковий , б) – десятковий .

4. Створити макроеlementи – N=16 для двійкового лічильника та N=10 для десяткового лічильника. Записати у пам'ять файли зі створеними макроеlementами (функція меню Create Subcircuit, див.рис. 18).
5. Оформити звіт про виконання лабораторної роботи у якому подати:
 - а. тему, мету та завдання лабораторної роботи;
 - б. прізвище, ініціали та назву групи студента, що виконав роботу;
 - с. зображення досліджуваних схем;
 - д. логічні таблиці істинності та зняті часові діаграми, отримані в результаті дослідження схеми;
 - е. висновки.
6. Звіт оформити у вигляді файла з іменем: Прізвище_асб.doc.

ЛАБОРАТОРНА РОБОТА №7

Тема: Моделювання роботи оперативного запам'ятовуючого пристрою (статичного)

Мета роботи. З використанням можливості пакета EWB побудувати логічну схему комірки статичної пам'яті для зберігання 1 біта інформації на базі логічних елементів. Створити макроелемент комірки пам'яті та дослідити роботу 4-бітової матриці комірок пам'яті.

Теоретичні відомості.

Для зберігання інформації у ЕОМ використовують запам'ятовуючі пристрої. За призначенням та виконуваними функціями розрізняють оперативні (ОЗП), постійні (ПЗП) та перепрограмовувані запам'ятовуючі пристрої (ППЗП). ОЗП вирізняються з поміж інших пристроїв високою швидкістю і за способом реалізації діляться на статичні та динамічні. Статичні ОЗП переважно використовують для кеш-пам'яті і будують на тригерних схемах, а динамічні ОЗП – для решти оперативної пам'яті а інформацію запам'ятовують на конденсаторах. Розрізняють три режими роботи ОЗП: режим запису, режим зберігання і режим читання інформації. Структурно ОЗП поділена на комірки, у яких міститься інформація. Розмір комірки у різних ЕОМ може бути довільний. Найменша комірка містить 1 біт інформації. Кожна комірка пам'яті має свою адресу, за допомогою якої здійснюють запис, читання чи контроль її вмістимого. Адресу комірки задають сукупністю двох координат, які формують за допомогою дешифраторів. Крім того кожна комірка пам'яті повинна мати інформаційний вхід, інформаційний вихід та вхід, на який подають команди запису або читання.

Розглянемо роботу елементарної комірки пам'яті для зберігання 1 біта інформації. Побудуємо її на основі D-тригера (рис. 19).

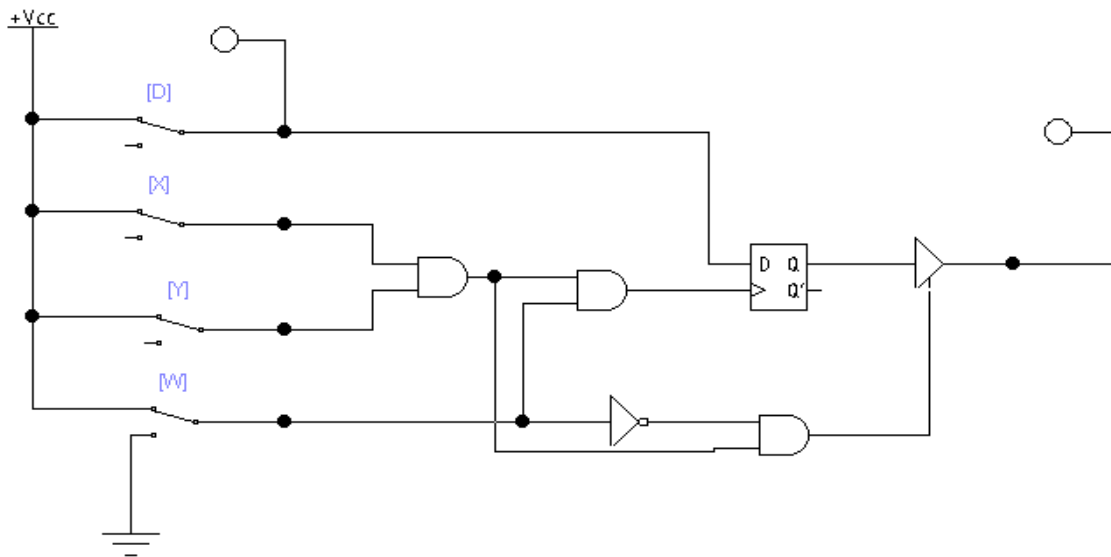


Рис. 19. Схема для зберігання 1 біта інформації.

План виконання лабораторної роботи.

1. За допомогою елементарних логічних схем синтезувати у робочому полі логічну схему комірки ОЗП для зберігання 1 біта інформації.
2. Входи синтезованої схеми приєднати до вихідних клем перемикачів, а вихід – до аналізатора логічних рівнів або індикатора (рис. 19).
3. Вхідні клеми перемикачів: (D) - інформаційний вхід, (X,Y) - адресні входи, (W) - вхід для вибору команди запису або читання приєднати до джерела постійного живлення $+V_{CC}$ з напругою +5 вольт, що відповідає рівню логічної одиниці.
4. Дослідити роботу комірки пам'яті та перевірити виконання функцій запису, зберігання і зчитування інформації.
5. Створити макроелемент 1bit. Записати у пам'ять файл зі створеним макроелементом (функція меню Create Subcircuit, рис. 26).
6. За допомогою макроелементів та перемикачів синтезувати у робочому полі схему чотирибітової матриці комірок пам'яті (див. рис. 20).

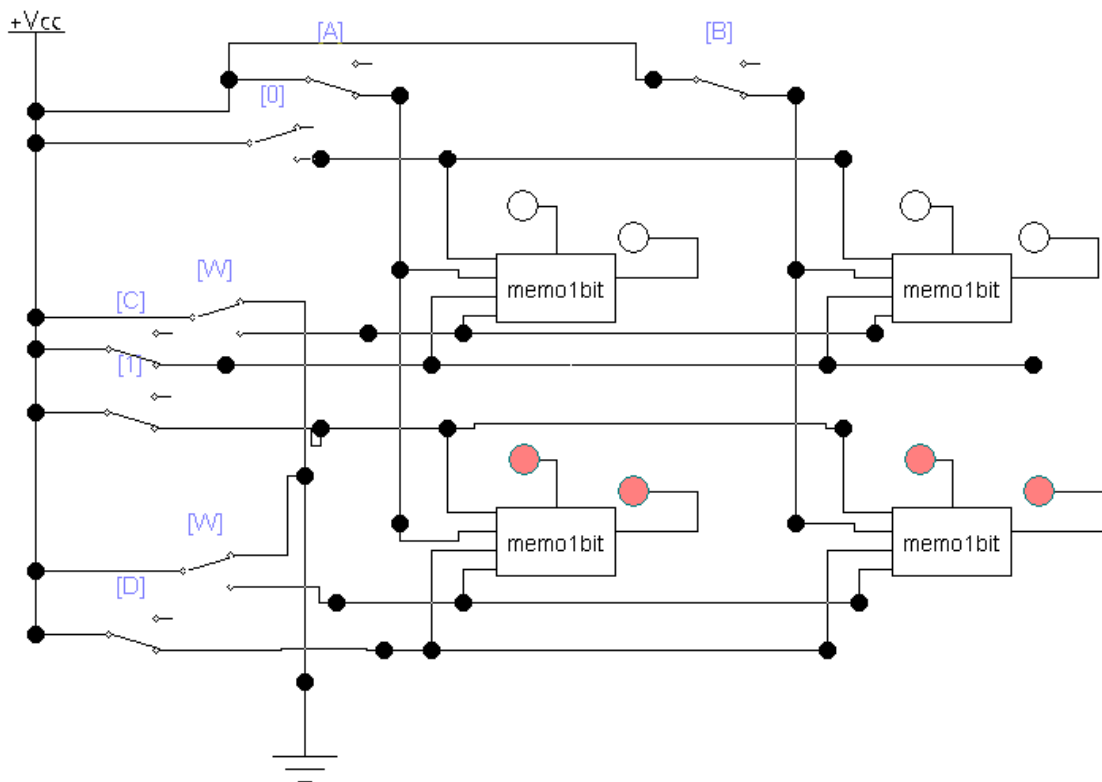


Рис. 20. Приклад синтезованої схеми чотирибіткової матриці комірок пам'яті в робочому полі пакета EWB.

7. Дослідити роботу чотирибіткової матриці комірок пам'яті та перевірити виконання функцій запису, зберігання та зчитування інформації у кожній окремо взятій комірці. Тут входи A,B відповідають координатам Y0, Y1; C,D – координатам X0, X1; W=1 – запис, W=0 – читання; 0,1 – інформаційні входи нульового і першого розрядів пам'яті.
8. Оформити звіт про виконання лабораторної роботи у якому подати:
 - а. тему, мету та завдання лабораторної роботи;
 - б. прізвище, ініціали та назву групи студента, що виконав роботу;
 - с. зображення досліджуваних схем;
 - д. пояснити результати, отримані в результаті дослідження схеми;
 - е. висновки.
9. Звіт оформити у вигляді файла з іменем: Прізвище_ac7.doc.

ЛАБОРАТОРНА РОБОТА №8

Тема: Моделювання роботи арифметико-логічного пристрою

Мета роботи. Ознайомлення з можливостями моделювання роботи арифметико-логічного пристрою (АЛПр) з використанням пакета EWB. Дослідження роботи АЛПр на прикладі виконання арифметичних і логічних операцій.

Тривалість роботи: 4 години.

Теоретичні відомості.

Арифметико-логічний пристрій призначений для виконання арифметичних і логічних операцій над багаторозрядними операндами залежно від кодів, що подають на керуючі входи.

Залежно від типу даних, які опрацьовуються, існує кілька типів АЛПр: з фіксованою комою, з плаваючою комою і для десяткових чисел.

За характером використання елементів і вузлів АЛПр поділяють на блочні (для кожного типу числа свій блок) та багатофункційні (опрацювання різних типів чисел виконується у одному блоці).

За способом організації виконання операцій над кожним операндом програми АЛПр поділяють на послідовні, паралельні та послідовно-паралельні.

За видом зв'язків між головними вузлами АЛПр бувають з безпосередніми зв'язками та з магістральною структурою.

Спрощена блок-схема арифметико-логічного пристрою зображена на рис.21.

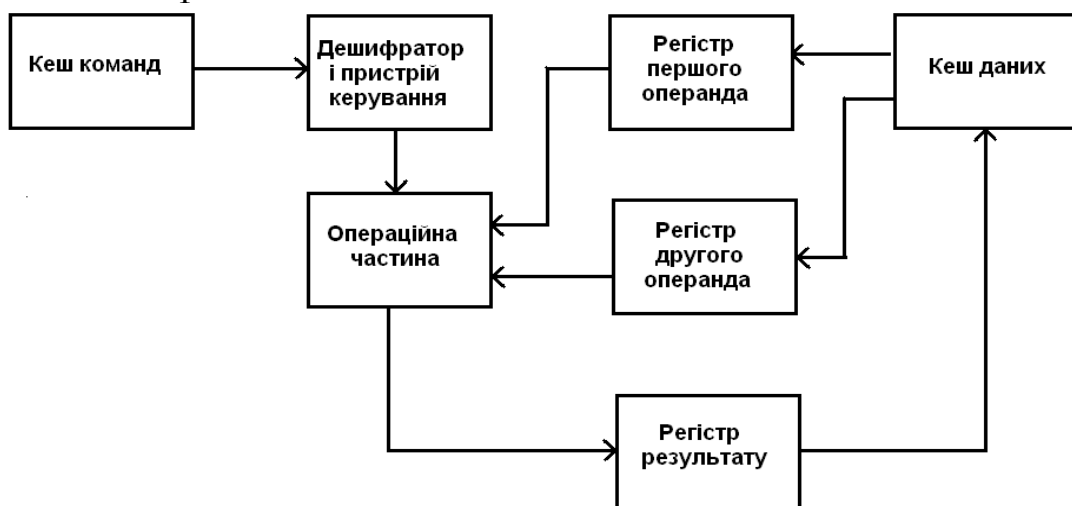


Рис.21 Спрощена блок-схема арифметико-логічного пристрою.

Команда, яку потрібно виконати, подається із кешу команд на дешифратор команд. Дешифратор команд виробляє послідовність мікрокоманд, які потрібно виконати. Пристрій керування забезпечує виконання мікрокоманд операційною частиною. Операнди виконуваної команди завантажують із кешу даних у регістри першого і другого операндів, звідки операнди у відповідний момент надходять у операційний пристрій. Результат виконання команди записується у регістр результату. Структура регістрів АЛПр, куди поміщають вхідні дані і результати, а також розмір регістрів (число двійкових розрядів) формують поняття розрядної сітки ЕОМ, яке є визначальним для організації обчислень.

Історично першим АЛПр на мікросхемах великого ступеню інтеграції був чотирирозрядний Intel 4004. У програмі EWB чотирирозрядний АЛПр представлений мікросхемою 74181, яку ми пропонуємо дослідити докладніше.

АЛПр працює у двох режимах: в режимі виконання логічних операцій при значенні керуючого сигналу модифікатора **M=1** і в режимі виконання арифметичних операцій при значенні керуючого сигналу **M=0**. Кодові комбінації подають на керуючі входи S0, S1, S2, S3. Схема має 8 інверсних інформаційних входів (2 чотирирозрядних числа) A₀, A₁, A₂, A₃, B₀, B₁, B₂, B₃. Вхід C_n є входом перенесення з попереднього розряду: при C=1 – врахування сигналу перенесення, при C=0 – без врахування перенесення. Результат операцій над розрядами чисел A і B отримують на інформаційних виходах F₀, F₁, F₂, F₃ (інверсних). Для організації прискореного перенесення передбачено спеціальні виходи: P – підтвердження перенесення, C_{n+4} – перенесення 4-розрядного каскаду, G – вихід утворення перенесення 4-розрядного каскаду. Виходи P і G використовують для об'єднання декількох схем АЛПр за допомогою блоку прискореного перенесення. Вихід A=B є індикатором порівняння чисел A і B (при A=B на виході маємо логічну 1).

Повний набір операцій містить 32 арифметичні операції (16 при наявності перенесення та 16 при відсутності перенесення) і 16 логічних операцій. У наведеній таблиці наведено логічні та арифметичні операції залежно від значень кодових комбінацій на керуючих входах S0, S1, S2, S3.

| S3 | S2 | S1 | S0 | Логічна функція M=1 | Арифметико-логічна функція M=0, C'=1 (вхідний перенос відсутній) |
|-----------|-----------|-----------|-----------|----------------------------|---|
| 0 | 0 | 0 | 0 | A' | A або $A+1$ (вх.перенос) |
| 0 | 0 | 0 | 1 | $(A+B)'$ | $A+B$ або $A+B+1$ |
| 0 | 0 | 1 | 0 | $A' B$ | $A+B'$ або $(A+B')+1$ |
| 0 | 0 | 1 | 1 | 0 | -1 або 0 при переносі |
| 0 | 1 | 0 | 0 | $(AB)'$ | $A+AB'$ або $(A+AB')+1$ |
| 0 | 1 | 0 | 1 | B' | $(A+B)+AB'$ або $((A+B)+AB')+1$ |
| 0 | 1 | 1 | 0 | $(A \ll B)'$ викл. АБО | $A-B-1$ або $A-B$ |
| 0 | 1 | 1 | 1 | AB' | $AB'-1$ або AB' |
| 1 | 0 | 0 | 0 | $A'+B$ | $A+AB$ або $A+B+1$ |
| 1 | 0 | 0 | 1 | $A \ll B$ | $A+B$ або $A+B+1$ |
| 1 | 0 | 1 | 0 | B | $(A+B')+AB$ або $(A+B')+AB+1$ |
| 1 | 0 | 1 | 1 | AB | $AB-1$ або AB |
| 1 | 1 | 0 | 0 | 1 | $A+A$ або $A+A+1$ |
| 1 | 1 | 0 | 1 | $A+B'$ | $(A+B)+A$ або $(A+B)+A$ |
| 1 | 1 | 1 | 0 | $A+B$ | $(A+B')+A$ або $(A+B')+A+1$ |
| 1 | 1 | 1 | 1 | A | $A-1$ або A |

На рис.22 наведена схема для дослідження АЛПр 74181.

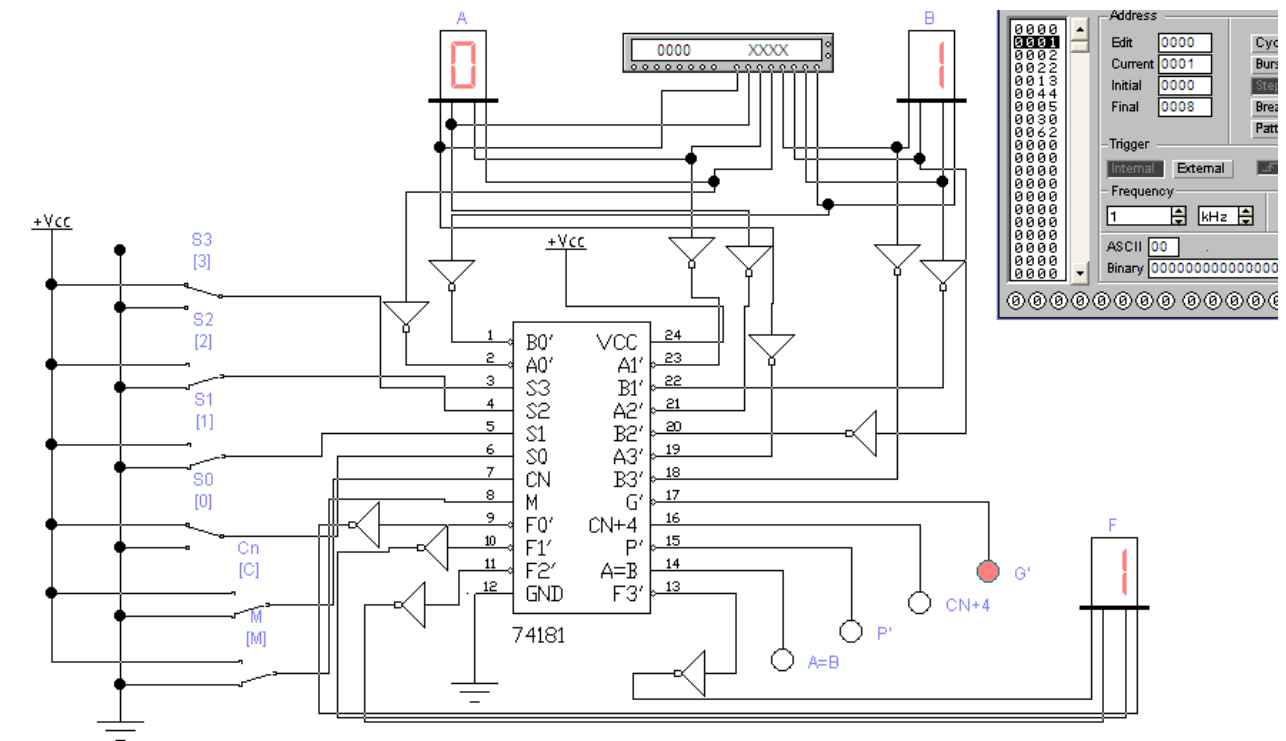


Рис. 22. Схема для демонстрації роботи арифметико-логічного пристрою 74181.

План виконання лабораторної роботи.

1. Підготувати на робочому столі мікросхему АЛПр 74181 з бібліотеки EWB, перемикачі, джерело логічної одиниці +Vcc та логічного нуля – заземлення.
2. Входи схеми S0, S1, S2, S3, C_n, M приєднати до вихідних клем перемикачів, а виходи A₀, A₁, A₂, A₃, B₀, B₁, B₂, B₃, F₀, F₁, F₂, F₃ – до аналізатора логічних рівнів або індикаторів (рис. 22).
5. Значення чотирирозрядних операндів A і B задають за допомогою генератора слів і у шістнадцятковому коді відображують на індикаторах. Результат операції відображують на алфавітно-цифровому індикаторі F.
6. Дослідити роботу синтезованої схеми та перевірити виконання арифметичних та логічних операцій згідно до наведених кодових комбінацій у таблиці.
8. Оформити звіт про виконання лабораторної роботи у якому подати:
 - f. тему, мету та завдання лабораторної роботи;

- g. прізвище, ініціали та назву групи студента, що виконав роботу;
- h. зображення досліджуваних схем;
- i. пояснити результати, отримані в результаті дослідження схеми;
- j. висновки.

9. Звіт оформити у вигляді файла з іменем: Прізвище_ac8.doc.

ГОЛОВНІ ВІДОМОСТІ ПРО РОБОТУ ПАКЕТА ELECTRONICS WORKBENCH

Пакет Electronics Workbench (EWB) є типовим симулятором. Він має добре розвинений редактор схем та програму емуляції з орієнтацією на інтегральні схеми (SPICE), що дозволяє легко будувати екранні сценарії і практично миттєво їх аналізувати. Повнота аналізу схем у різних режимах робить дослідження продуктивними. Програма емуляції дає змогу моделювати і досліджувати аналогові, цифрові та змішані електричні кола.

Пакет містить обширну бібліотеку промислових мікросхем. У цьому практикумі використано версію Electronics Workbench Version 5.12 для операційних систем Windows 3.x та Windows 95/NT.

У середовищі пакета вмонтована система підказок (Help) у вигляді гіпертексту, для якого відведені спеціальні інформаційні вікна.

Робоча панель пакета EWB.

Після активації програми в операційній системі (наприклад, у Windows XP , рис.1д) на екрані монітора ПЕОМ з'являється робоча панель, яку формально можна розбити на кілька полів. Загальний вигляд робочої панелі показано на рис. 2д.

Центральну частину панелі займає робоче поле, у якому відбувається синтез електронної схеми. Над робочим полем розташовані поля базових елементів, промислових схем, приладів, індикаторів та ін.(див рис. 2д).

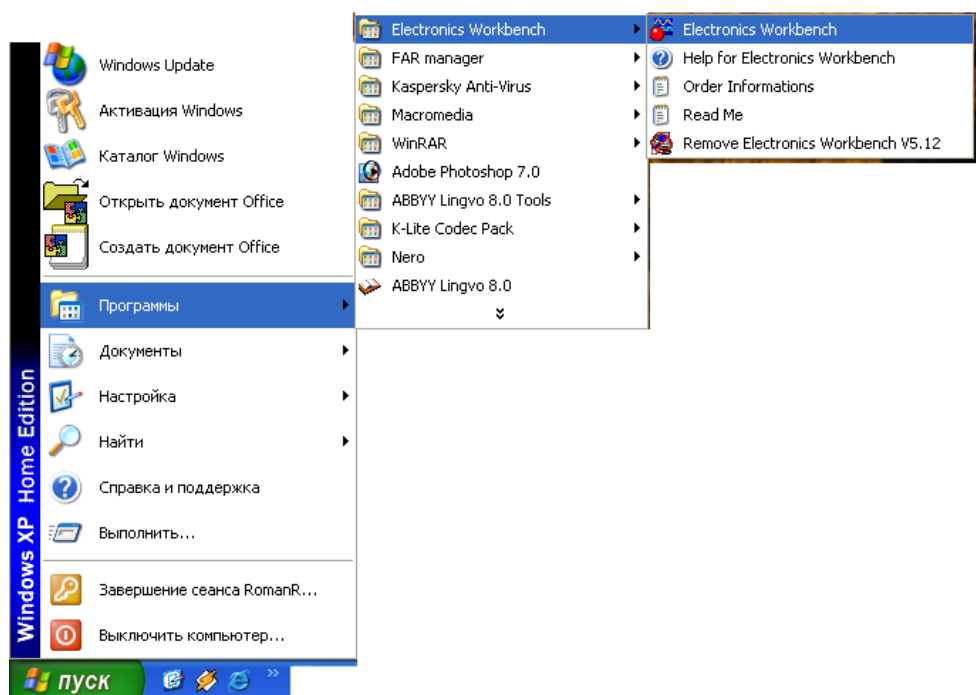


Рис.1д. Вигляд фрагмента вікна операційної системи WindowsXP.

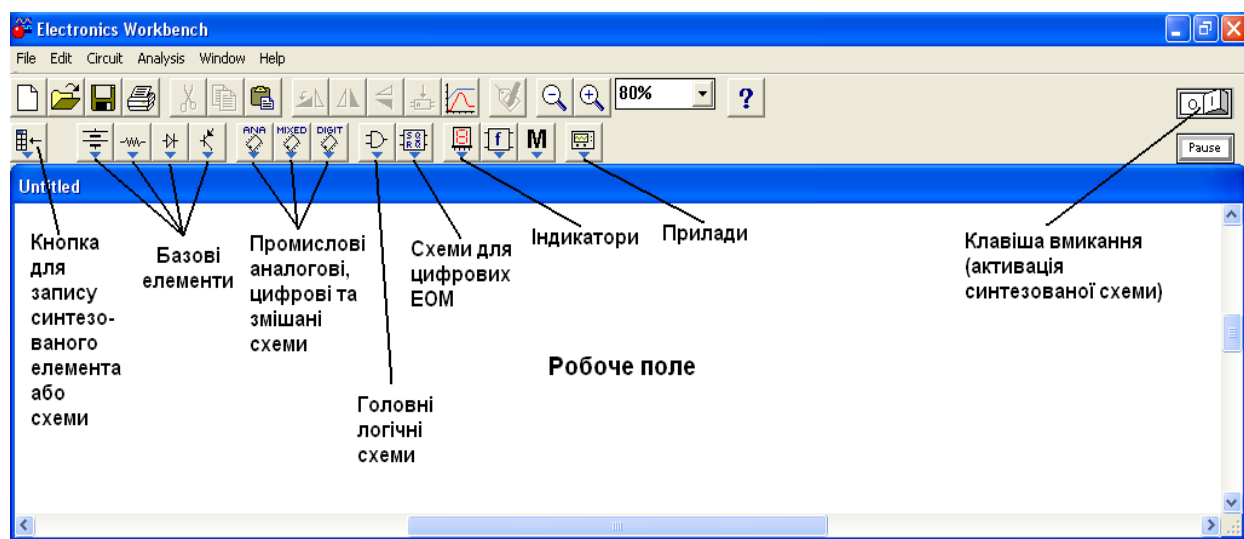


Рис. 2д. Загальний вигляд робочої панелі EWB.

У полі базових елементів розміщені джерела живлення постійної та змінної напруги, головні конструктивні елементи електричних кіл: резистори, конденсатори, індуктивності, реле, діоди, транзистори тощо.

Головні логічні схеми представлені елементами І, АБО, НЕ та їхніми комбінаціями. Схеми цифрових ЕОМ містять тригери,

суматори, мультиплексори, дешифратори, регістри, арифметико-логічні пристрої.

Набір індикаторів та вимірювальних приладів дає змогу контролювати у синтезованих схемах рівні напруг, значення струмів та логічних рівнів.

Поле приладів містить мультиметр, осцилоскоп, генератор сигналів різної форми, плотер, генератор логічних слів, логічний аналізатор та логічний перетворювач.

Вибір елементів та розташування їх на робочому полі. Для синтезу електронних схем використовують готові елементи, які пропонує база пакета EWB, та власноруч створені елементи чи схеми у вигляді макроелементів. З метою дослідження роботи схеми та задання режимів роботи приєднують прилади та індикатори, які вибирають з відповідних полів пакета.

Щоб вибрати потрібний елемент з поля, виконують такі кроки. Переміщують вказівник мишки на зображення відповідного поля і зафіксують вибір натисканням лівої клавіші мишки. У верхній частині робочого поля з'явиться набір елементів. Переміщують вказівник мишки на зображення відповідного елемента, зафіксують вибір елемента натисканням лівої клавіші мишки. Далі, не відпускаючи клавіші, переміщують елемент на робоче поле у потрібне місце. Фіксація елемента відбудеться після відпускання клавіші мишки. За описаною вище методикою можна переміщати елементи у межах робочого поля. Якщо схема містить значну кількість елементів, яка не поміщається в центральній частині робочого поля, то за допомогою скролінгу можна розширювати межі поля.

Елементи робочого поля можна помічати, або іменувати, повертати, копіювати, видаляти та отримувати розширену інформацію про них. Для цього потрібно встановити вказівник мишки на відповідний елемент, натиснути праву клавішу мишки і виконати відповідний пункт меню. Інший спосіб видалення елемента: встановити вказівник мишки на відповідний елемент,

натиснути на ліву клавішу мишки, натиснути на клавішу “Delete” і дати підтвердження про видалення елемента.

З’єднання елементів та приладів у схемі. Елементи та прилади у схемі з’єднують за допомогою ліній та через відповідні точки з’єднань – контакти елементів і клеми приладів. Безпосередньо можна сполучити між собою лише дві точки схеми. Якщо потрібно сполучити між собою більше двох точок, то застосовують спеціальний з’єднувач (Connector) з поля базових елементів (див. рис. 3д).

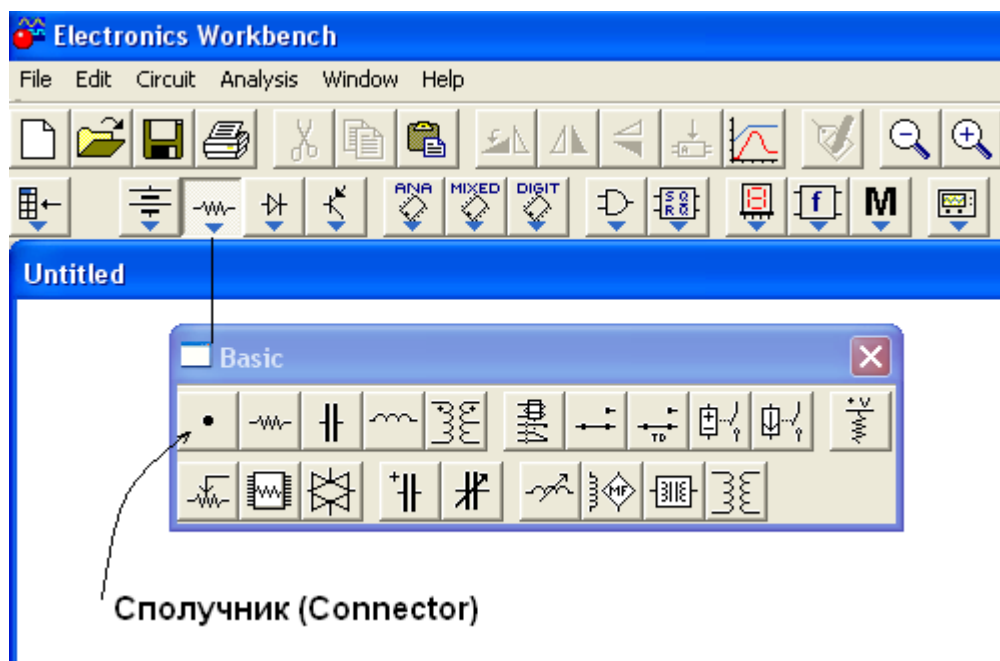


Рис.3д. Базові елементи EWB.

Для з’єднання двох точок схеми встановлюють вказівник мишки на одну з точок. У момент фіксації точки з’явиться темний кружечок (з’єднувач). Натискають на ліву клавішу мишки і, не відпускаючи її, переміщують вказівник до необхідної точки з’єднання. Під час протягування вказівника за ним вимальовується лінія сполучення. Процес завершують відпусканням лівої клавіші мишки у точці з’єднання. Для ліпшої візуалізації з’єднань лінії, що їх сполучають, набувають ламаної форми. Лінії і вузли з’єднань можна фарбувати в різні кольори. Для цього потрібно встановити вказівник мишки на відповідне

з'єднання і двічі натиснути на ліву клавішу мишки. З'явиться меню, з якого можна вибрати колір забарвлення відповідного з'єднання.

Щоб ліквідувати лінії сполучення встановлюють вказівник мишки на будь-яку точку з'єднання, натискають на ліву клавішу мишки і, утримуючи її, відводять від точки з'єднання. Після відпускання клавіші лінія сполучення буде видалена.

Використання приладів та індикаторів. Розгорнуті панелі приладів та індикаторів показані на рис. 4д. Прилади та індикатори вибирають за методикою вибору і розташування елементів на робочому полі, яка описана вище. Для проведення лабораторних робіт зі схемотехніки найчастіше використовують пробники, семисегментні індикатори цифр, генератор сигналів, генератор логічних слів, осцилоскоп, аналізатор логічних рівнів та логічний перетворювач.

Індикатор типу “пробника” приєднують до точок контролю схеми без додаткових сполучників. Одиничний логічний стан точки відповідає червоному забарвленню індикатора, а нульовий – білому. Семисегментні індикатори є двох типів: один передбачає керування кожним сегментом окремо, інший – подання на вхід індикатора чотирирозрядного двійкового коду, що відповідає числам у шістнадцятковій системі числення від 0 до F.

Прилади (інструменти) задають робочі режими, активізують роботу та забезпечують реєстрацію реакції (головно вихідних станів) синтезованої схеми. Застосовують узагальнене та деталізоване зображення приладів. Прилади можна приєднати до точок синтезованої схеми тільки в узагальненому зображенні, а контролювати за роботою схеми – у деталізованому.

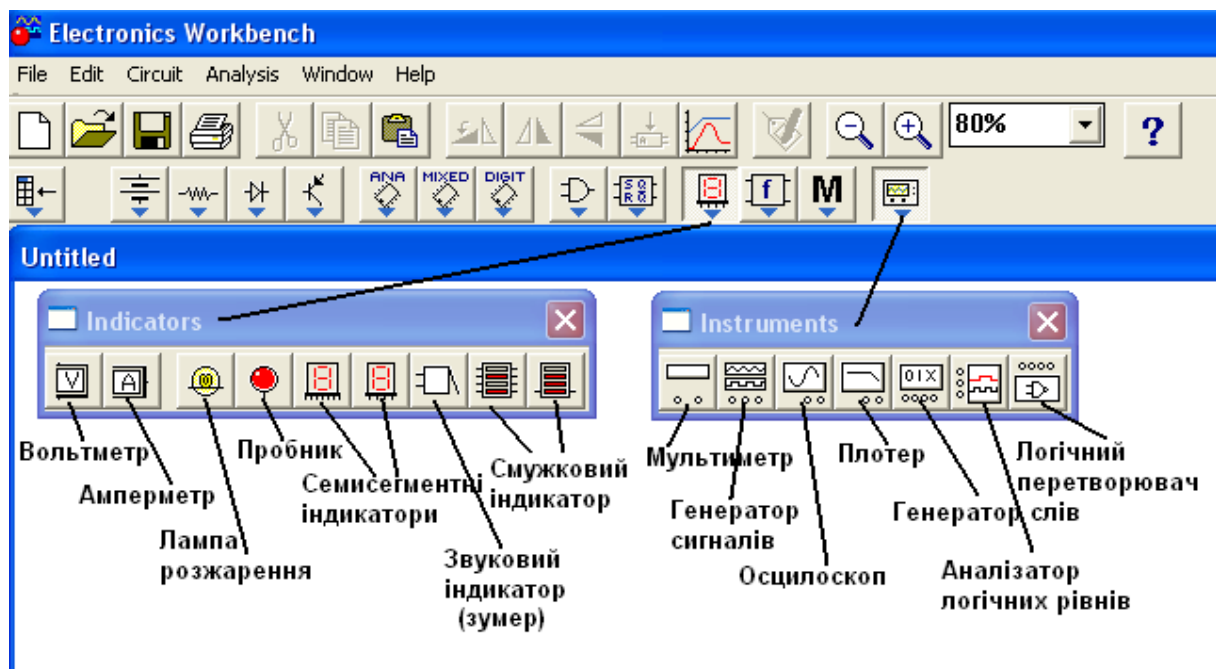


Рис. 4д. Індикатори та прилади.

Мультиметр призначений для вимірювання напруги, струму, опору та загасання між двома точками електричного кола. В полі індикатора видається значення вимірюваної величини змінного (AC) чи постійного (DC) струму.

Двоканальний осцилоскоп відображає амплітуду і частоту змінних сигналів у часі.

Плотер Бode продукує частотну характеристику електричного кола, його застосовують для аналізу роботи фільтрів.

Генератор слів призначений для задання комбінації двійкових чисел (логічних нулів і одиниць), які подають на вхід синтезованих електронних схем. Двійковій логічній одиниці на схемному рівні відповідає високий потенціал, а двійковому логічному нулю – низький. Узагальнене та деталізоване зображення генератора слів показано на рис. 5д.

Ліворуч розташована чотирирозрядна *комбінаційна матриця* шістнадцяткових чисел. Розмір матриці чисел є в межах від 0000 до FFFF (від 0 до 65535 у десятковій системі числення). Рядок матриці репрезентує 16-бітове слово, яке у випадку активації генератора висвітлене у нижній частині приладу. Щоб задати комбінацію певного розряду, потрібно

позиціонувати вказівник мишки на потрібний розряд і за допомогою клавіш “0” або “1” клавіатури набрати потрібний код. Контроль за пересуванням у матриці відбувається за допомогою індикаторів праворуч від комбінаційної матриці. Тут висвітлюються:

- адреса редагованого слова (Edit);
- адреса поточного слова (Current);
- початкова адреса блоку (Initial);
- кінцева адреса блоку чисел (Final).

Два останні поля можна редагувати подібно до матриці чисел.

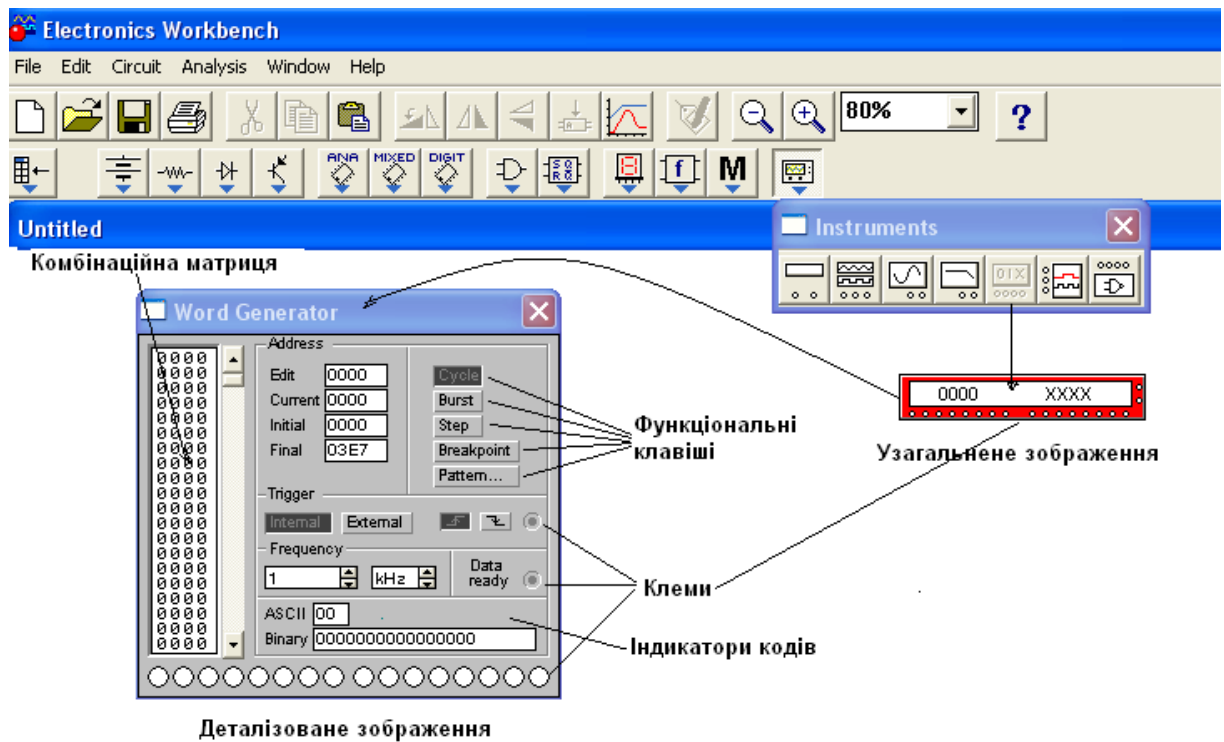


Рис. 5д. Узагальнене та деталізоване зображення генератора логічних слів.

Керують роботою генератора за допомогою *функціональних клавіш*:

Step – покрокового перебирання комбінацій (натискання на клавішу спричинює пересування у комбінаційній матриці на одне слово);

Burst – автоматичного одноразового перебирання усіх комбінацій у межах заданих адрес;


Cycle – циклічного перебирання усіх комбінацій у межах заданих адрес. Примусова зупинка відбувається після натискання на будь-яку функціональну клавішу через позиціонування мишкою на неї або натискання на комбінацію клавіш Ctrl+T на клавіатурі.

Breakpoint – організації зупинки потоку слів і його рестарт за допомогою вибраного слова. Слово вибирають шляхом його виділення у комбінаційній матриці. Точок зупинки може бути декілька.

Pattern – створення, збереження і перевизначення шаблонів слів генератора. Операції виконують за допомогою діалогового вікна, яке відкривається після натискання на клавішу Pattern.

Перемикач режимів синхронізації (Trigger) Internal відповідає внутрішньому вибору режиму синхронізації, а External – синхронізації від зовнішнього джерела.

Клеми забезпечують зв'язок приладу з електронною схемою у двох напрямках:

- 16 контактів у нижній частині приладу, які відповідають 16 розрядам згенерованого слова:
- клема Data Ready слугує для зняття сигналу готовності даних на виході генератора, а клема  – для зняття синхронізуючого імпульсу по наростаючому передньому чи спадному задньому фронтах.

Аналізатор логічних рівнів призначений для відображення логічного стану розрядів двійкового слова.

Зліва розташовані 16 клем для приєднання точок схеми, навпроти яких розміщені 16 горизонтальних рядків дисплея (див. рис. 6д).

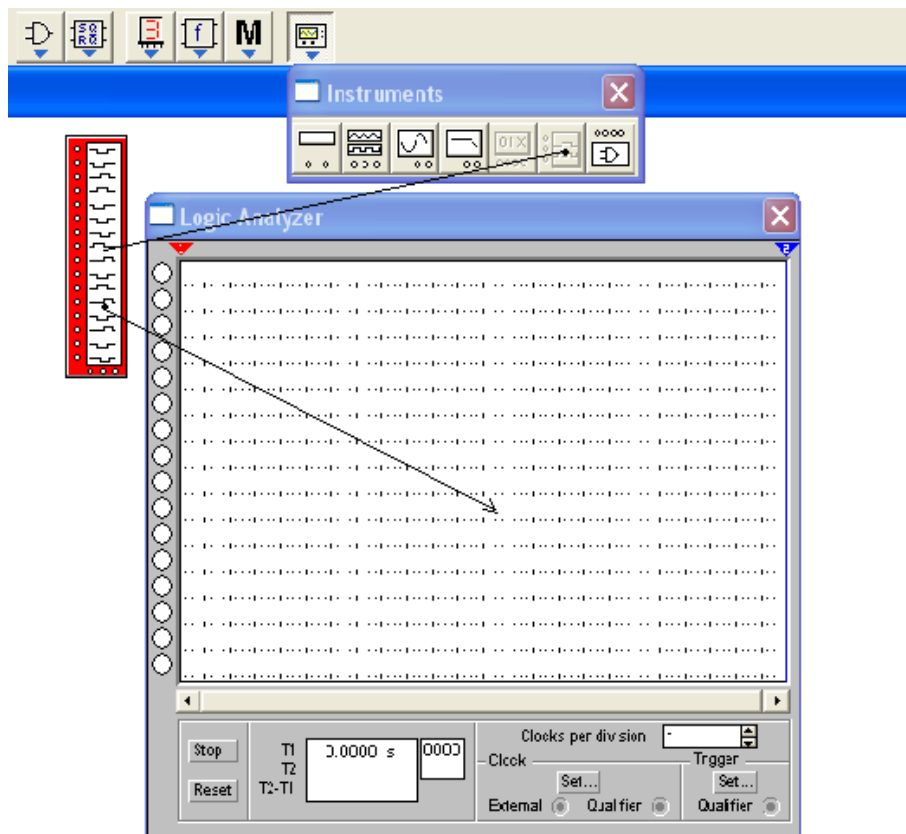


Рис. 6д. Узагальнене та деталізоване зображення аналізатора логічних рівнів

Коли електронна схема активована, аналізатор логічних рівнів почне реєстрацію сигналу на вході. Дані відображаються у вигляді прямокутних сигналів. Нумерація каналів така: верхній рядок відповідає каналу 0, наступний – каналу 1 і т.д. Двійкові значення кожного біта поточного слова відображаються зліва у кружечках. Для задання параметрів відображуваних сигналів використовують опції меню Analysis/Analysis Options. Режим синхронізації вибирають з меню Clock/Set.

Логічний перетворювач – це багатфункційний пристрій, який дає змогу синтезувати логічні вирази та комбінаційні схеми, виконувати логічні перетворення, аналізувати логічні схеми. Детальне зображення пристрою показано на рис. 7д.

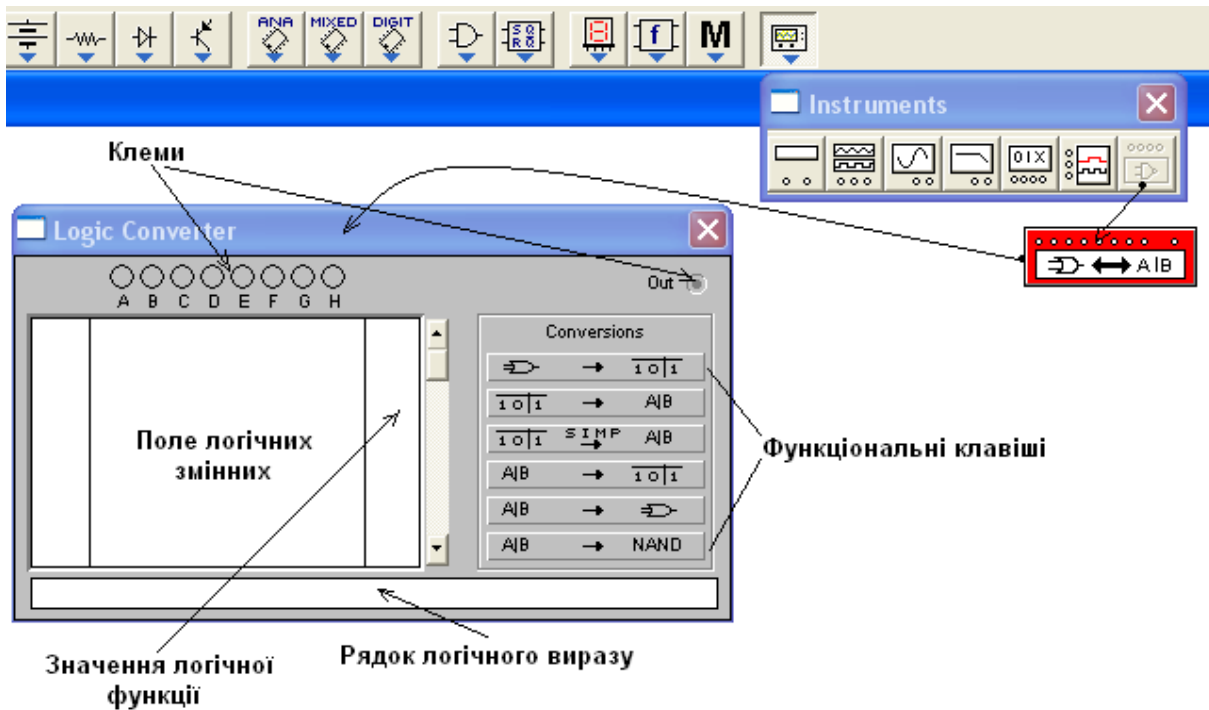








Рис. 7д. Узагальнене та деталізоване зображення логічного перетворювача.

У центрі логічного перетворювача є *таблиця істинності*, яка складається з поля логічних змінних та поля значень логічної функції. У лівій частині поля відображається трирозрядний десятковий еквівалент n -розрядного двійкового числа, записаного у середній частині (полі логічних змінних). Поле значення логічної функції призначене для введення і редагування цих значень. Кількість розрядів n -розрядного двійкового числа визначається шляхом активації відповідної кількості вхідних клем, які містяться над полем (клеми позначені латинськими літерами A, B, C, D, E, F, G, H).


У нижній частині конвертора розташований рядок для запису логічного виразу.

Права частина приладу містить функціональні клавіші (Conversions), за допомогою яких виконують:

- запис логічної таблиці істинності для синтезованої схеми
;
- запис логічного виразу згідно зі сформованою логічною таблицею істинності
;

- запис спрощеного логічного виразу згідно зі сформованою логічною таблицею істинності ;
- формування логічної таблиці істинності для записаного логічного виразу ;
- синтез схеми згідно зі записаним логічним виразом ;
- синтез схеми на основі логічних елементів І-НЕ згідно з записаним логічним виразом .


З метою створення таблиці істинності для синтезованої схеми потрібно виконати таке:

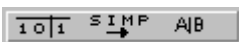
- приєднати сигнальні входи логічного конвертора до відповідних входних точок схеми;
- приєднати вибрану вихідну клему схеми до клеми конвертора Out;
- натиснути на клавішу .

Таблиця істинності для схеми з'явиться в полі логічних змінних.

Конструювання таблиці істинності.

1. Натисніть на номер клеми входного каналу, які позначені від А до Н. Поле логічних змінних заповниться одиничками та нулями. Права частина таблиці (значення функції) буде заповнена нулями.
2. Відредагуйте праву колонку зі значеннями функції відповідно до заданих умов. Функція може мати значення 1, 0 або х. (Позначення х означає, що значення 1 або 0 є однаково прийнятні).

Для перетворення таблиці істинності у логічний (булівський) вираз натисніть на клавішу . Булівський вираз буде зображений у нижньому рядку.

Спрощений логічний вираз згідно зі сформованою логічною таблицею істинності отримують натисканням на клавішу . Для спрощення застосовано метод Квіна - МакКласкі, який у цьому випадку є ефективнішим, ніж карти Карно, однак потребує значного обсягу вільної пам'яті.

Для синтезу електронної схеми за заданим логічним виразом необхідно записати відповідний вираз у нижньому рядку конвертора. Правила запису логічних змінних та операцій додавання і множення такі:

- 1) логічні змінні позначають великими літерами латинського алфавіту;
- 2) операцію додавання позначають символом “+”;
- 3) операцію множення не позначають жодним символом, а дві чи більше змінних записують поряд;
- 4) інверсне значення змінної позначають символом апострофа після символу змінної.

Переміщення у робочому полі. Створення макроелементів.

Схему в робочому полі переміщують з метою кращої деталізації окремих її частин. Переміщення виконують за допомогою стандартних засобів скролінгу. Зображення схеми на робочому столі масштабують за допомогою стандартних функцій меню.

Під час синтезу схеми часто виникає потреба у виконанні певних маніпуляцій як з окремими елементами, так і з частинами схеми (підсхемами). Конкретно об'єкт маніпуляції можна визначити шляхом його виділення. Щоб виділити окремий елемент необхідно перевести на нього вказівник мишки і натиснути на ліву клавішу. В разі натиснутої клавіші елемент можна переміщати в будь-яке місце робочого поля. Після натискання правої клавіші мишки з'явиться меню, функції якого дають змогу отримати характеристики елемента, вирізати, копіювати, видаляти, повертати його.

Окремий об'єкт (частину схеми, окремий вузол) виділяють так. Переводять вказівник мишки в уявний початок межі цієї підсхеми і натискають на ліву клавішу. Не відпускаючи клавіші, пересувають вказівник так, щоб рамка, яка утворюється в робочому полі, охопила потрібну підсхему. Після відпускання клавіші рамка зникне, а виділений об'єкт змінить забарвлення. Виділену підсхему можна переміщати у робочому полі, копіювати, вирізати, видаляти за визначеними правилами у

меню Edit. Якщо виділений об'єкт є окремим вузлом схеми, то іноді доцільно створити з нього копію багаторазового використання – макроелемент.

Для створення макроелемента потрібно:

- 1) виділити частину схеми, яка відповідає певному вузлу.
- 2) вибрати з меню Circuit/Create Subcircuit, і виконати вимоги меню, що з'явиться:
 - Copy from Circuit поміщає копію виділеної схеми у макроелемент. Оригінал схеми залишиться у вікні;
 - Move from Circuit переносить виділений фрагмент з робочого стола у макроелемент;
 - Replace in Circuit поміщає виділений фрагмент у макроелемент і заміщає виділений фрагмент схеми новоствореним макроелементом.

Зауваження: для подальшого використання макроелемента потрібно правильно зафіксувати вхідні та вихідні клеми. Цього вдасться досягти, якщо в разі виділення об'єкта позначення конекторів залишатимуться поза межами виділення.

Активізувати роботу синтезованої схеми можна за допомогою функціональної клавіші у верхньому правому кутку робочого стола або за допомогою клавіш керування роботою генератора логічних слів.

ВАРІАНТИ ЗАВДАНЬ ДО ЛАБОРАТОРНОЇ РОБОТИ №1

1. $ab+acd$
2. $ab+c(\bar{a}+d)$
3. $\bar{a}b+acd$
4. $ab+\bar{b}cd$
5. $ab+\bar{b}cd$
6. $abc+\bar{a}d$
7. $\bar{a}bc+ad$
8. $abc+\bar{a}d$
9. $abc+\bar{a}d$
10. $abc+ad$
11. $abc+\bar{a}d$
12. $abd+\bar{b}c$
13. $\bar{a}bd+\bar{b}c$
14. $ab\bar{d}+\bar{b}c$
15. $abd+ab$
16. $bcd+ab$
17. $bcd+\bar{a}b$
18. $bcd+\bar{a}c$
19. $bcd+\bar{a}c$
20. $bcd+\bar{b}c$
21. $acd+\bar{a}d$
22. $acd+\bar{b}c$
23. $acd+\bar{b}c$
24. $\bar{a}cd+ab$
25. $acd+\bar{a}c$
26. $acd+\bar{a}d$
27. $acd+\bar{b}c$
28. $ac\bar{b}+ad$
29. $acd+\bar{a}b$
30. $acd+\bar{a}b$

ВАРІАНТИ ЗАВДАНЬ ДО ЛАБОРАТОРНОЇ РОБОТИ №3

1. $a+6$
2. $c+2$
3. $b+7$
4. $b+6$
5. $8+d$
6. $6+e$
7. $b+a$
8. $8+9$
9. $9+a$
10. $b+5$
11. $c+9$
12. $d+b$
13. $b+c$
14. $a+c$
15. $b+a$
16. $b+8$
17. $c+5$
18. $d+a$
19. $b+9$
20. $8+b$
21. $7+d$
22. $5+c$
23. $6+b$
24. $d+b$
25. $c+a$
26. $6+d$
27. $7+c$
28. $8+e$
29. $9+b$
30. $4+d$

СПИСОК ЛІТЕРАТУРИ

1. Биркгоф Г., Барти Т. Современная прикладная алгебра / Пер.с англ. – М., 1976.
2. Горбунов В.А., Панфилов Д.И., Преснухин Д.А. Справочное пособие по микропроцессорам и микроЭВМ. – М., 1988.
3. Гук М. Аппаратные средства IBM PC. Энциклопедия – СПб., 1999.
4. Злобін Г.Г., Рикалюк Р.Є. Архітектура та апаратне забезпечення ПЕОМ: Навч.посіб. –К., 2006.
5. Енциклопедія кібернетики у 2 т. – К., 1977.
6. Каган Б.М. Электронные вычислительные машины и системы. – М., 1991.
7. Каім С.Д. Комп'ютерна схемотехніка (симуляційний практикум). – Одеса, 2004.
8. Колодницький М.М. Технічне та програмне забезпечення комп'ютерних інформаційних технологій. – Житомир, 1995.
9. Ланцов А.А., Зворыкин Л.Н., Осипов И.Ф. Цифровые устройства на комплементарных МПД интегральных микросхемах. – М., 1983.
10. Локазюк В.М. Мікропроцесори та мікроЕОМ у виробничих системах. – К., 2002.
11. Мельник Б.К., Жовтанецький М.І. Лабораторний практикум з основ побудови і технології виробництва ЕОМ. – Львів, 1998.
12. Мячев А.А., Степанов В.Н. Персональные ЭВМ и микроЭВМ. Основы организации: Справочник. – М., 1991.
13. Рикалюк Р.Є. Архітектура комп'ютерів. – Львів, 2002.
14. Смирнов А.Д., Архитектура вычислительных систем. – М., 1990.
15. Таненбаум Э. Архитектура компьютера. – СПб., 2003.
16. Толковый словарь по вычислительным системам / Под ред. В. Иллингуорта. – М., 1991.

17. Цифровые ЭВМ: практикум /Под общ. ред. К.Самофалова.
– К.: Вища шк., 1990.
18. Electronics Workbench. User Manuel. Interactive Image
Technologies, 1997.

НАВЧАЛЬНЕ ВИДАННЯ

Роман Євстахович Рикалюк

ЛАБОРАТОРНИЙ ПРАКТИКУМ з курсу «АРХІТЕКТУРА ЕОМ»

Редактор М. М. Мартиняк

Підп. до друку **.**.05. Формат 60x84/16. Папір друк.№2. Друк на різогр.
Умовн.фарбовідб. 0.9.
Обл.-вид.арк. 1.0. Тираж 200 прим. Зам. **.

Видавничий центр Львівського національного університету імені Івана Франка
79000, Львів, вул.Дорошенка, 41