

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
Тернопільський національний технічний університет  
імені Івана Пулюя

*Кафедра комп'ютерних систем та мереж*

МЕТОДИЧНІ ВКАЗІВКИ ДО ЛАБОРАТОРНИХ  
РОБІТ ТА КУРСОВОГО ПРОЕКТУВАННЯ

з дисципліни:

***"Комп'ютерна схемотехніка"***

для студентів денної та заочної форми навчання  
напряму підготовки 6.05010201 «Комп'ютерна інженерія»

Тернопіль – 2015

Паламар М.І., Паламар А.М. Комп'ютерна схемотехніка: Методичні вказівки до лабораторних робіт та курсового проектування. – Тернопіль: ТНТУ, 2015. – 95 с.

У методичних вказівках викладені лабораторні роботи для практичного засвоєння апаратного забезпечення комп'ютерних схемотехнічних пристроїв на основі використання програмного пакету Multisim для моделювання та дослідження вузлів комп'ютерної схемотехніки. Приведені теми для курсового проектування.

Методичні вказівки призначені для підготовки бакалаврів за напрямом підготовки 6.050102 «Комп'ютерна інженерія» спеціальністю 8.05010201 «Комп'ютерні системи та мережі».

**Укладачі:** доц., д.т.н. Паламар М.І.,  
Паламар А.М.

**Рецензенти:**

професор Євтух П. С.  
професор Ткачук Р. А.

**Відповідальний за випуск:** проф., д.т.н., Лупенко С.А.

Затверджено на засіданні кафедри “Комп'ютерних систем та мереж”

Протокол № від 2015р.

Схвалено та рекомендовано до друку методичною комісією факультету комп'ютерно-інформаційних систем і програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя.

Протокол № від 2015р.

## З М І С Т

1. Ознайомлення з пакетом *NI Multisim*
2. Елементи програми *NI Multisim*
3. Порядок виконання роботи для розробки принципової електричної схеми
  - 3.1. Моделювання інтегруючого *RC*-кола
  - 3.2. Моделювання диференціюючого *RC*-кола
  - 3.3. Моделювання транзисторного автогенератора
  - 3.4. Моделювання трьохкаскадного транзисторного підсилювача
  - 3.5. Моделювання тригера *Шмідта* на аналогових елементах
  - 3.6. Моделювання мультівібратора на елементах *I-НІ*
  - 3.7. Моделювання цифрового генератора
  - 3.8. Моделювання алгебраїчного суматора на *ОП*
  - 3.9. Моделювання роботи дешифратора
4. Завдання до виконання лабораторних робіт

### Комбінаційні схеми

- 4.1. Виконання аналізу в середовищі пакету *Multisim*
- 4.2. Проектування найпростіших логічних схем
- 4.3. Доведення основних тотожностей булевої алгебри за допомогою найпростіших логічних схем

### Абстрактний аналіз логічних схем

- 4.4. Проектування дешифраторів і шифраторів
- 4.5. Проектування мультиплексора і демультиплексора
- 4.6. Проектування суматора

### Аналіз послідовних схем

- 4.7. Проектування тригерів
- 4.8. Проектування регістрів
- 4.9. Проектування двійкових лічильників
- 4.10. Проектування ЦАП і АЦП

### Аналіз аналогових схем

- 4.11. Проектування операційних підсилювачів
  - 4.12. Проектування суматорів на *ОП*
  - 4.13. Проектування інтеграторів на *ОП*
  - 4.14. Проектування мультівібраторів на *ОП*
- Використана література

## 1. Ознайомлення з пакетом NI Multisim

Пакет *NI Multisim* призначений для моделювання й аналізу електротехнічних і схемотехнічних схем. Даний пакет надзвичайно точно моделює побудову реальних схем на реальних компонентах. *NI Multisim* є додатком під операційні системи *Windows 95/98/NT/2000/XP*.

Метою даної лабораторної роботи є ознайомлення з правилами і методами роботи в *NI Multisim*. Лабораторна робота переслідує ознайомчі цілі. Студенту необхідно засвоїти принципи і методи побудови схем за допомогою *NI Multisim*, розміщення елементів на робочому полі, вибір та проведення аналізу, інтерпретація його результатів.

Запуск *Electronics Multisim*. Пакет *NI Multisim* запускається як будь-який інший додаток за допомогою піктограми в *START MENU*.

Пакет *NI Multisim* являє собою засіб програмної розробки і імітації електричних кіл. Інтерфейс користувача складається з панелі меню, панелі інструментів і робочої області (рис. 1).

### Горизонтальне командне меню:

File - організація роботи з файлами. За допомогою цього розділу меню користувач здійснює роботу з файлами (відкриття, створення, роздрук файлів та інше).

Edit - за допомогою цього розділу користувач здійснює редагування і роботу з поточним документом. Опції розділу дозволяють копіювати, видаляти переміщати елементи або блоки схеми. Крім того можливе настроювання візуальних параметрів схеми (розташування й орієнтація елементів схеми, настроювання кольорів і шрифту, пошук та інші стандартні функції).

Circuit - розділ, який дозволяє обертати, змінювати властивості, наближати і віддаляти елементи схеми.

Analysis - розділ служить для проведення різного роду аналізів схем.

Window - розділ призначений для настроювання екранних видів при роботі з документами.

Help - розділ служить для доступу до довідкової системи.

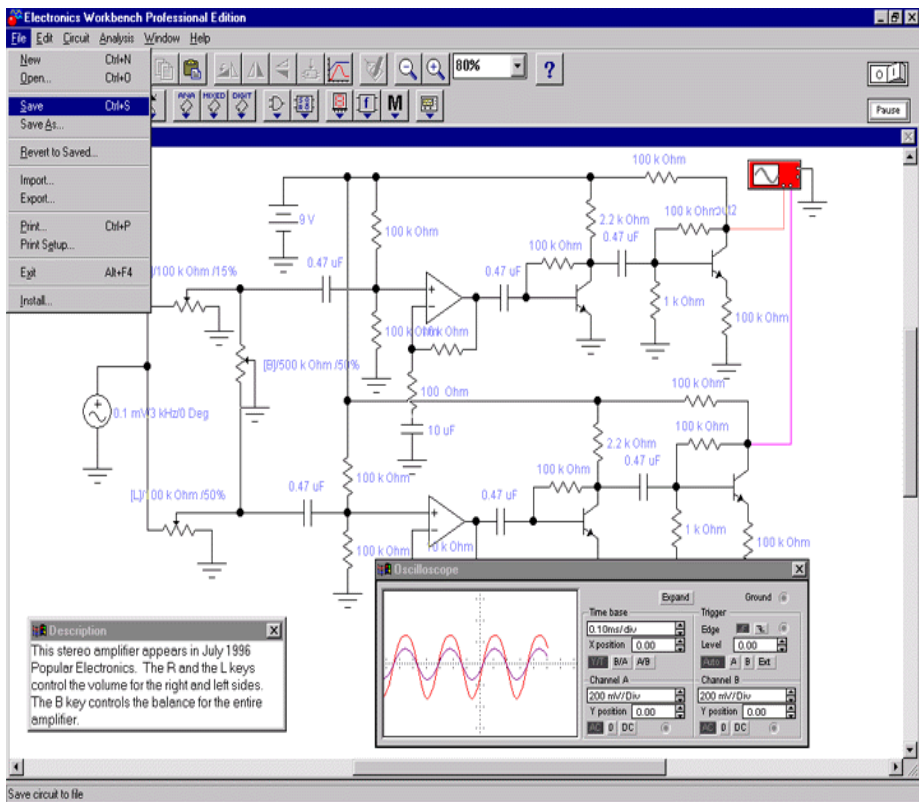


Рис. 1 – Зовнішній вигляд екрана комп'ютера при роботі з програмою EWB

## 2. Елементи програми NI Multisim

Всі елементи розміщені в 13 панелях, розділених на групи:

1. Аналогові елементи
2. Інтегральні схеми
3. Цифрові і логічні елементи
4. Індикаторні пристрої
5. Інструменти

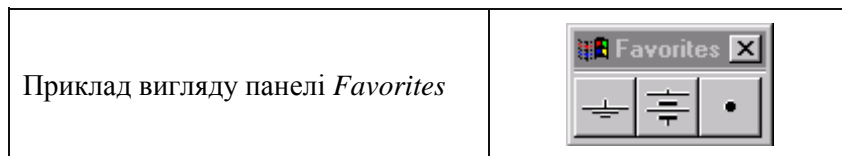
### 3. Порядок виконання роботи для розробки принципової електричної схеми

1. Запустити *NI Multisim*.

2. Підготувати файл для роботи. Для цього необхідно виконати наступні операції з меню: **File/New** і **File/Save as**. При виконанні операції **Save as** необхідно вказати ім'я файлу і каталог, у якому буде зберігатися схема.


3. Перенести необхідні елементи з заданої схеми на робочу область *NI Multisim*. Для цього необхідно вибрати розділ на панелі інструментів (*Sources, Basic, Diodes, Transistors, Analog Ics, Mixed Ics, Digital Ics, Logic Gates, Digital, Indicators, Controls, Miscellaneous, Instruments*), у якому знаходиться потрібний елемент, потім перенести його на робочу область (клацнути мишкою на потрібному елементі і, не відпускаючи кнопки, перенести в потрібне місце схеми).

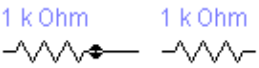
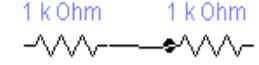
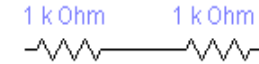
*Multisim* також надає можливість використовувати настроювану панель інструментів *Favorites*. Панель своя для кожного файлу схеми.



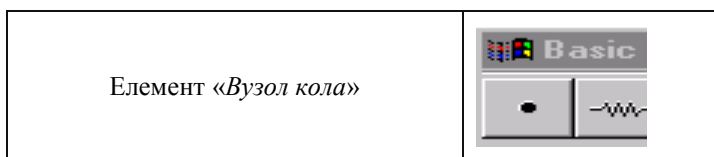
Для того, щоб додати в панель елемент, необхідно активізувати його зображення на панелі правою кнопкою і вибрати *Add to Favorites*. Щоб забрати з панелі *Favorites* - активізувати правою кнопкою елемент на панелі *Favorites* і вибрати *Remove from Favorites*.

4. З'єднати контакти елементів і розмістити елементи в робочій області для одержання необхідної схеми. Для з'єднання двох контактів необхідно клацнути по одному з контактів активною кнопкою миші і, не відпускаючи клавішу, довести курсор до другого контакту. Розглянемо ці дії в покроковому режимі:

1		Навести курсор мишки на вивід елемента так, щоб з'явилася чорна точка контакту
---	---	--

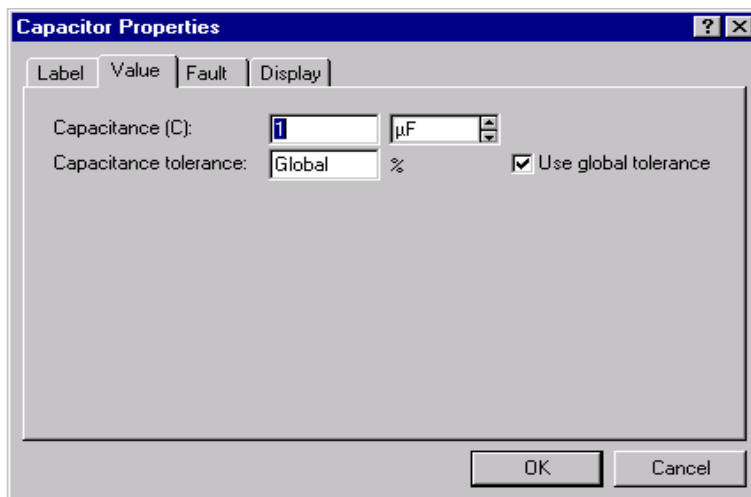
2		Зафіксувати клавішу миші, і не відпускаючи її, провести провідник до елемента, з яким треба виконати з'єднання
3		Коли провідник досягне виводу іншого елемента, з'явиться його точка контакту, тоді кнопку миші можна відпустити
4		Два елементи електрично з'єднані


При необхідності можна додати додаткові вузли (розгалуження). Для цього перетягують елемент із панелі на місце провідника туди, де треба його розгалузити.



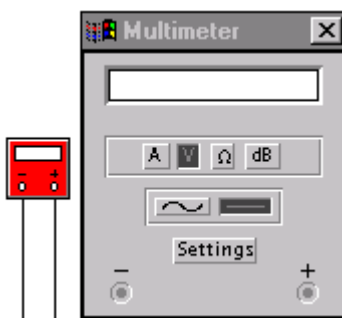
Активізацією на елементі правою кнопкою миші можна одержати швидкий доступ до найпростіших операцій над положенням елемента, таким як обертання (*rotate*), дзеркальне зображення (*flip*), копіювання/вирізання (*copy/cut*), вставка (*paste*), а також до його довідкової інформації (*help*).

**5.** Проставити необхідні номінали і властивості кожному елементу. Для цього потрібно двічі клацнути мишкою на елементі:



6. Коли схема зібрана і готова до запуску, потрібно натиснути кнопку включення живлення на панелі інструментів . У випадку серйозної помилки в схемі (замикання елемента живлення накоротко, відсутність нульового потенціалу в схемі) буде видане відповідне попередження.

7. Виконати аналіз схеми, використовуючи інструменти індикації. Вивід терміналу здійснюється подвійним натисканням клавіші мишки на елементі:



У випадку потреби можна скористатися кнопкою *Pause* .



8. При необхідності можна виконати доступні аналізи в розділі меню *Analysis*.

### 3.1. Моделювання інтегруючого RC-кола

Спочатку потрібно нанести на робочу область *NI Multisim* моделі деталей, які необхідні для моделювання даної схеми. Це виконується за допомогою активізації лівої кнопки мишки на потрібному наборі деталей, після чого буде виведене додаткове вікно, що включає в себе деталі набору. Далше вибираємо відповідну деталь, при цьому на кнопці з малюнком елемента натискається ліва кнопка мишки й елемент переноситься на робочу область (кнопку мишки необхідно тримати натиснутою до вибору місця розташування елемента). В даному випадку необхідні: джерело імпульсів (*Function Generator*), резистор (*Resistor*), конденсатор (*Capacitor*), осцилограф (*Oscilloscope*) і заземлення (*Ground*). Резистор і конденсатор знаходяться в наборі *Basics*, заземлення - в наборі *Sources*, осцилограф і генератор імпульсів - в наборі *Instruments*.

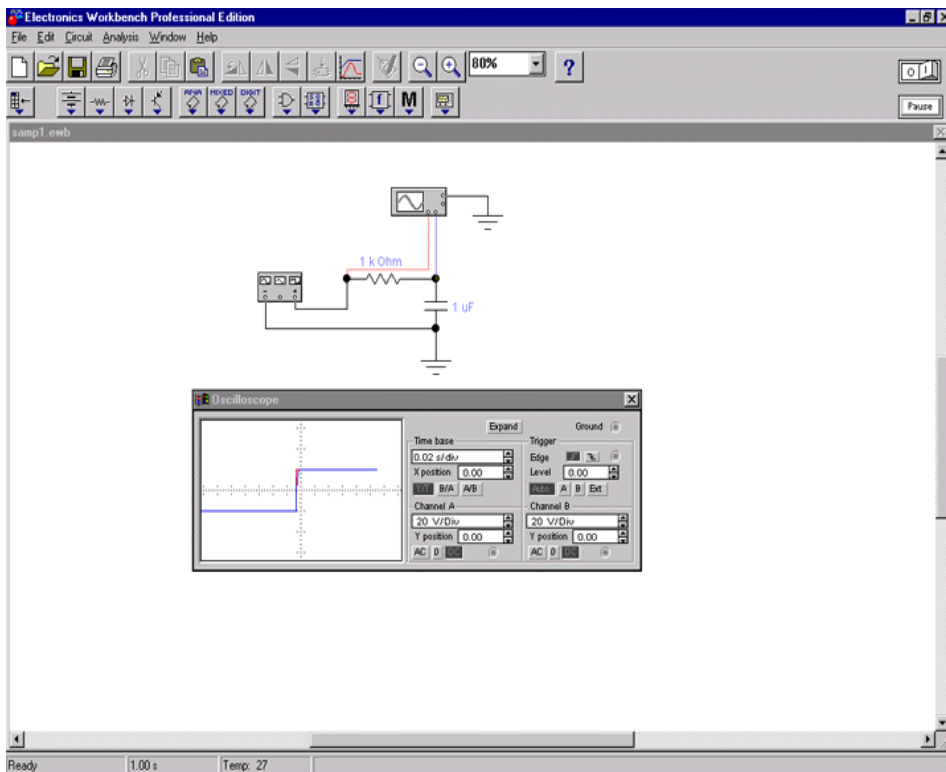


Рис. 3.1.1 - *Схема інтегруючого RC-кола в пакеті NI Multisim*

Кожен елемент має точки з'єднання, які потрібно з'єднати для одержання потрібної схеми. Це здійснюється вибором контакту лівою кнопкою мишки і переносом її до іншого контакту: при цьому створюється з'єднуючий їх провідник. При необхідності на провідник можна нанести вузол (*Connector* у наборі *Basics*). Потім для наочності можна перенести елементи в необхідні місця робочої області. Ця дія також здійснюється активізацією на елементі лівою кнопкою мишки і переносом при утриманні в натиснутому стані кнопки. При цьому з'єднувальні провідники будуть переміщені автоматично. При необхідності провідник можна також переміщати. На рис. 3.1.1 представлений вид інтерфейсу *NI Multisim* після зборки RC-кола.

Коли схема створена і готова до роботи, то для початку імітації процесу роботи необхідно виконати команду меню - активізувати кнопку включення живлення на панелі інструментів. Дана дія приведе в робочий стан схему й в одному з вікон рядка стану буде відображатися час роботи схеми (*час не відповідає реальному, залежить від швидкості процесора і системи персонального комп'ютера. Тому для розробки складних схем рекомендується використовувати комп'ютери Pentium II з тактовою частотою процесора не менше 266 МГц*). Перервати імітацію можна двома способами. Після закінчення роботи і перегляду результатів імітації можна повторно клацнути перемикач живлення. Якщо ж потрібно тимчасово перервати роботу схеми, наприклад, для детального розгляду осцилограми, а потім продовжити роботу, то можна скористатися кнопкою *Pause*, що також розташована на панелі інструментів.

Тепер, для проведення аналізу імітації можна змінювати номінали елементів, виводити і налаштовувати термінали приладів. В даному випадку можна переглянути осцилограму на виході RC-кола. Для цього потрібно вивести вікно терміналу осцилографа подвійним натисканням на компоненті *Oscilloscope*. Вид панелі осцилографа представлений на рис. 3.1.2.

При використанні осцилографа в *NI Multisim* є можливість перегляду сигналу протягом усього часу імітації. Для цього можна скористатися кнопкою *Expand* (*розширений режим*) і кнопками прокрутки зображення; щоб перевести панель у нормальний режим,

використовується кнопка **Reduce**. Вид розширеної панелі осцилографа показаний на рис. 3.1.3.

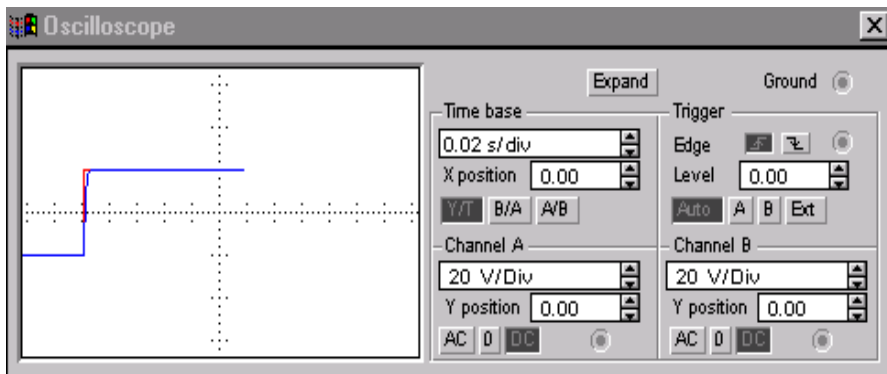


Рис. 3.1.2 – Вид нормальної панелі осцилографа NI Multisim

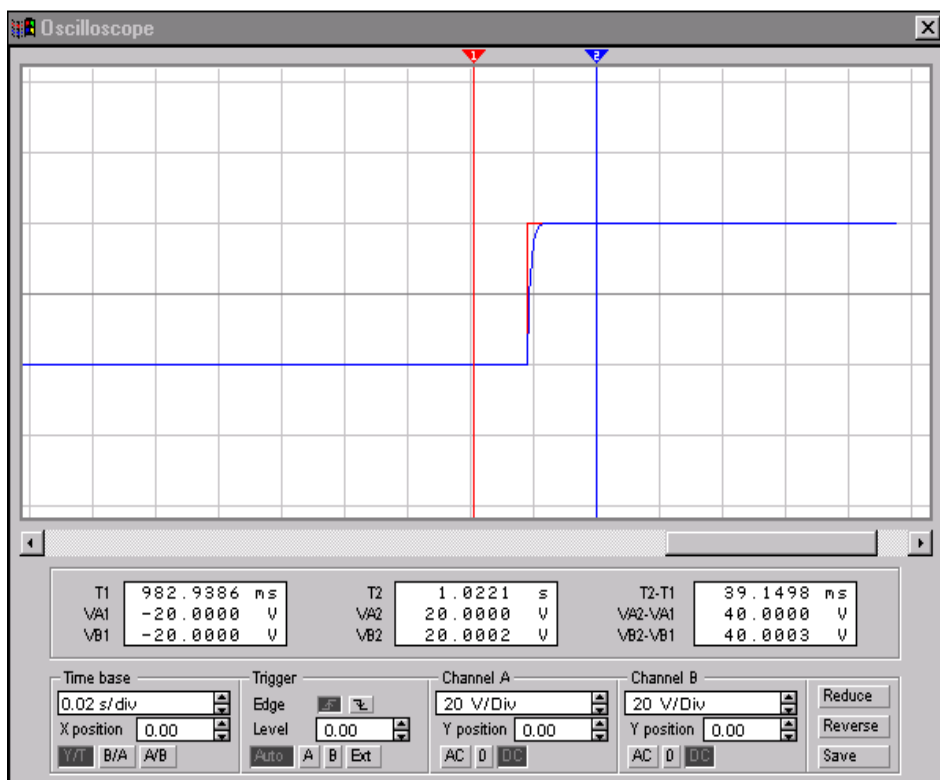


Рис. 3.1.3 – Розширена панель осцилографа

Тепер для вивчення властивостей  $RC$ -кола можна змінити сигнал на його вході. Для цього потрібно вивести на екран панель генератора імпульсів. Її вигляд представлений на рис. 3.1.4.

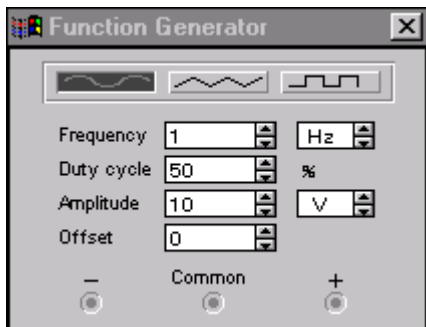


Рис. 3.1.4 – Панель генератора імпульсів

За допомогою генератора імпульсів можна формувати три види сигналів: *синусоїдний*, *пилподібний* і *прямокутний*. У даному випадку для аналізу потрібний прямокутний імпульс. Для переведення генератора в потрібний режим потрібно натиснути відповідну кнопку на панелі. Також можна змінити інші параметри – частоту й амплітуду сигналу. Перед зміною будь-яких параметрів варто відключати джерела живлення схеми, інакше можливе одержання невірних результатів.

Для того, щоб змінити будь-які параметри елементів схеми, потрібно двічі клацнути лівою кнопкою миші на потрібному елементі: при цьому буде виведено вікно властивостей елемента. Приклад такого вікна приведений на рис. 3.1.5. Крім аналізу прямого спостереження за терміналами інструментів *NI Multisim* дозволяє виконати додаткові види аналізу. Наприклад, для даної схеми можна привести одержання *АЧХ* і *ФЧХ* схеми як чотиріполюсника. Тобто, при розрахунку на вхід схеми буде подаватися сигнал різної частоти і буде виконаний аналіз залежності виду вихідного сигналу від вхідного. При цьому потрібно буде задати початкову і кінцеву частоти, на яких буде проведений аналіз. Для проведення цього аналізу потрібно припинити роботу кола, тобто скористатися перемикачем живлення або кнопкою *Pause* і виконати команду меню

*Analysis/AC Frequency*. Перед розрахунком буде виведене вікно параметрів аналізу.

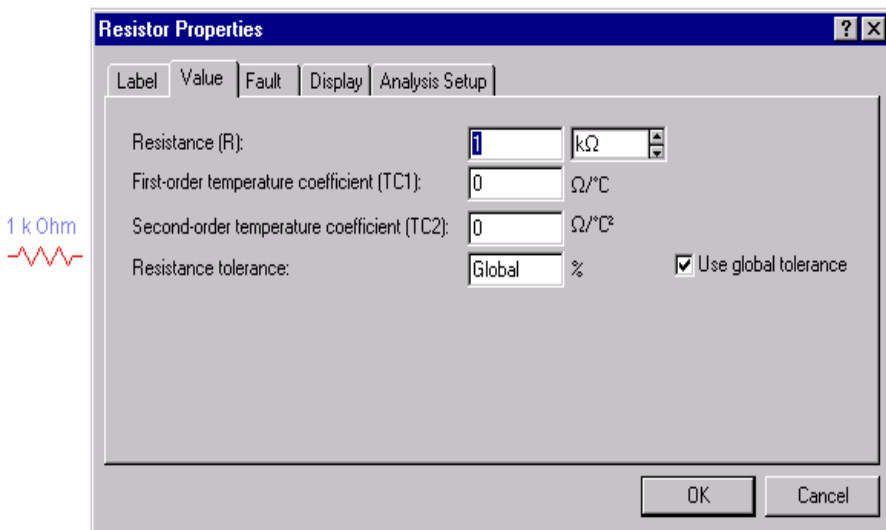


Рис. 3.1.5 - Вікно параметрів резистора

Вид цього вікна приведений на рис. 3.1.6. При необхідності можна змінити деякі з параметрів: *Start frequency* (початкова частота), *End frequency* (кінцева частота), *Sweep type* (тип горизонтальної осі на кінцевому графіку), *Number of points* (кількість крапок аналізу). У даному випадку зручно встановити кількість досліджуваних точок рівною 1000 для одержання більш згладженого графіка, тип горизонтальної осі – *логарифмічною* і діапазон частот від 1 Гц до 100 кГц.

Для одержання графіків *АЧХ* і *ФЧХ* потрібно натиснути кнопку *Simulate* у вікні параметрів аналізу, після чого буде виведене вікно результатів, яке представлено на рис. 3.1.7.

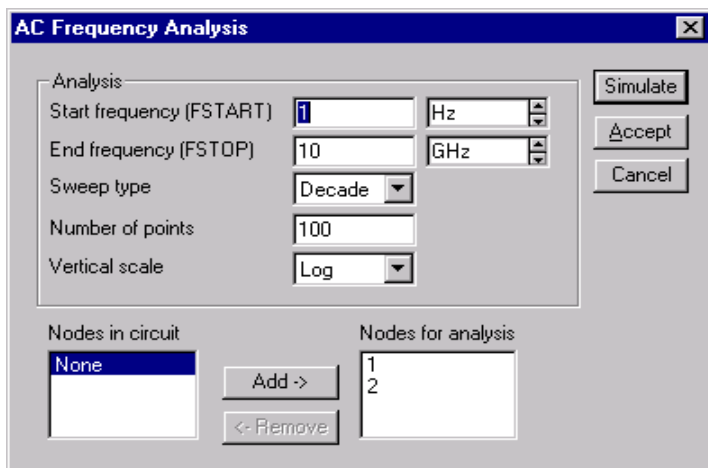


Рис. 3.1.6 – Параметри аналізу AC Frequency

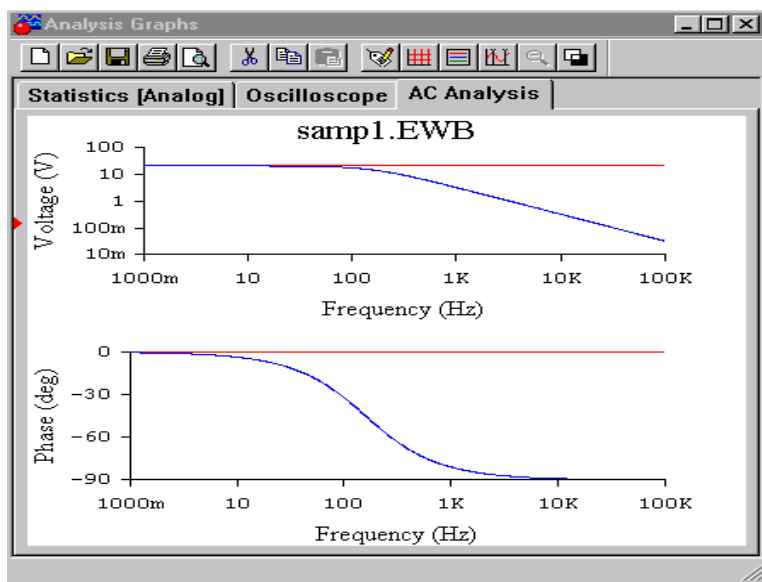


Рис. 3.1.7 – АЧХ і ФЧХ інтегруючого RC-кола

### 3.2. Моделювання диференціюючого RC-кола

Для моделювання диференціюючого RC-кола необхідно зібрати схему, яка приведена на рис 3.2.1. З диференціюючим RC-колом можна проробити ті ж види аналізу, що були описані в попередньому пункті. На рис. 3.2.2 показаний сигнал на виході кола, а на рис. 3.2.3 - АЧХ і ФЧХ схеми.

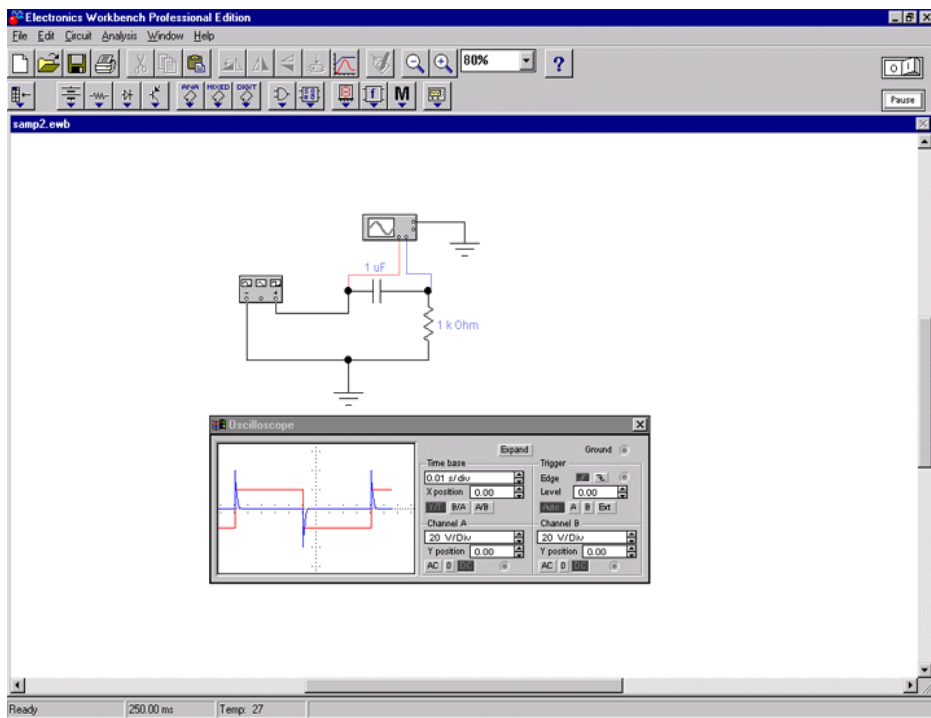


Рис. 3.2.1 – Диференціююче RC-коло

Після внесення змін або закінчення робіт схему при необхідності можна зберегти, скориставшись командою меню *File/Save*.

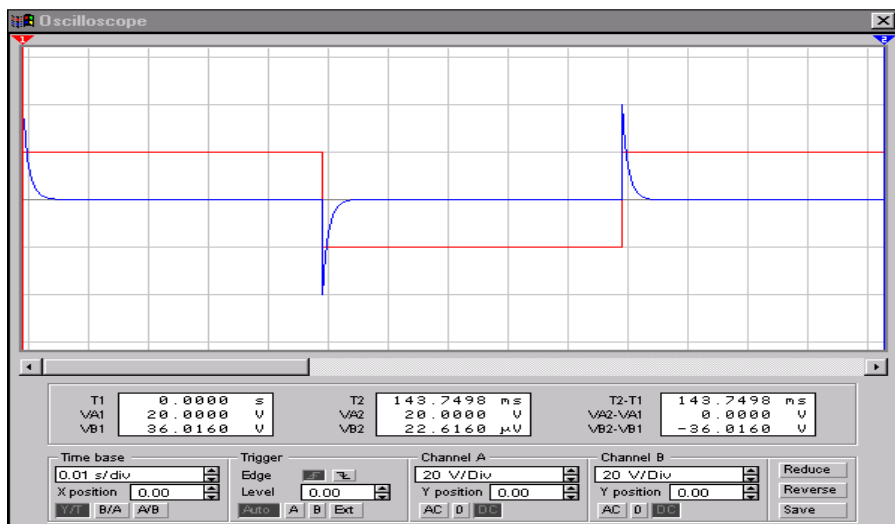


Рис. 3.2.2 – Сигнал на виході диференціюючого RC-кола

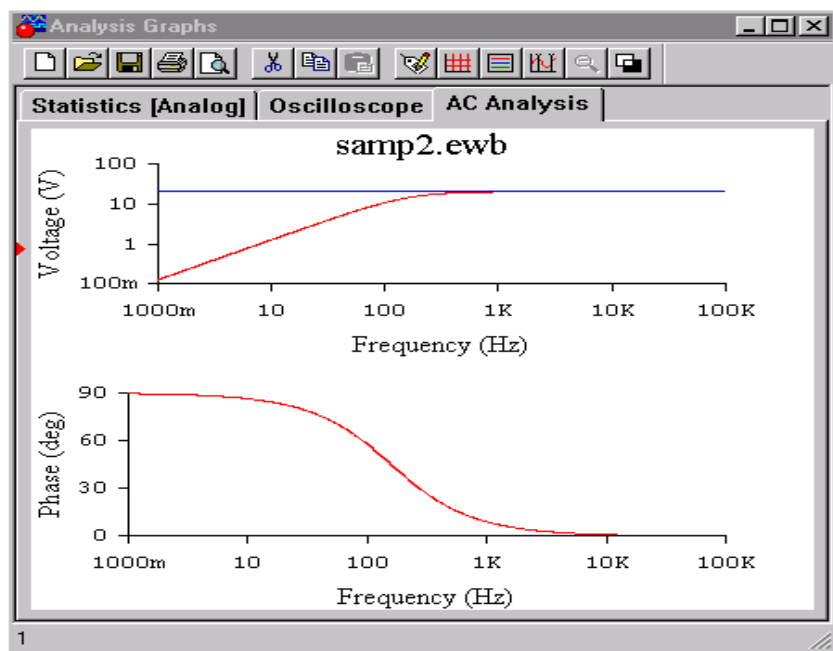


Рис. 3.2.3 – АЧХ і ФЧХ диференціюючого RC-кола



### 3.3. Моделювання транзисторного автогенератора

Для моделювання транзисторного автогенератора необхідно зібрати схему, представлену на рис. 3.3.1. Для цього потрібно нанести всі компоненти схеми на робочу область *NI Multisim* і з'єднати їх контакти провідниками. Модель транзистора знаходиться в наборі деталей *Transistors*, а джерело живлення в наборі *Sources*. Після з'єднання моделей деталей в схемі необхідно настроїти параметри кожного компонента шляхом його активізації клавішею мишки і заповненням вікон параметрів. Після настроювання параметрів включають джерело живлення і спостерігають за панеллю осцилографа - чи виходить генератор в режим генерації. Якщо ж на виході не з'являється сигнал змінної напруги, то потрібно повторити розрахунок параметрів схеми і змінити їх відповідним чином.

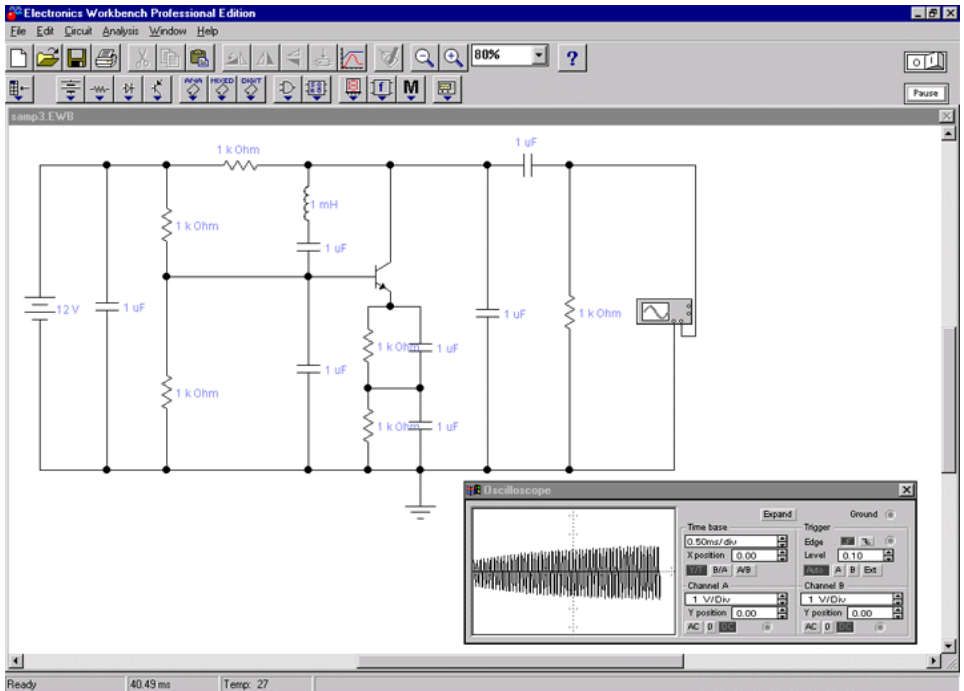


Рис. 3.3.1 – Модель транзисторного автогенератора

За сигналом на виході генератора зручно спостерігати, використовуючи розширене вікно терміналу осцилографа. На рис. 3.3.2 показаний момент початку генерації сигналу і момент

встановлення стабільного режиму транзисторного автогенератора. Для більш детального вивчення можна використати кнопки прокрутки і змінити параметри термінала.

Для одержання більш точної моделі можна замінити компоненти реальними моделями існуючих, тобто замінити, наприклад, ідеальний транзистор моделлю реально існуючого транзистора. Бібліотека *NI Multisim* має досить велику кількість реальних моделей деталей широко відомих виробників.

Після завершення роботи з програмним комплексом потрібно закрити програму, попередньо зберігши при потребі створену схему.

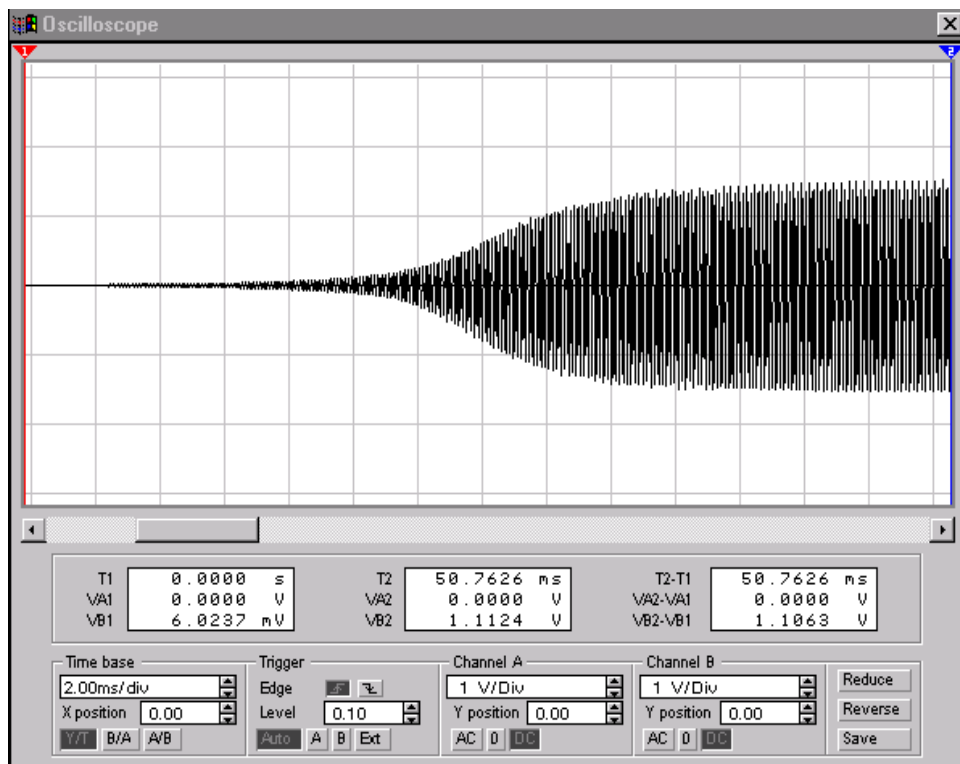


Рис. 3.3.2 – Сигнал на виході транзисторного автогенератора

### 3.4. Моделювання трьохкаскадного транзисторного підсилювача

Як джерело сигналу необхідно взяти елемент “AC Voltage Source” (джерело змінної напруги). Для розміщення його на схемі досить перейти у вікно *Passive*, помістити на необхідний елемент (у нашому випадку - це джерело напруги) курсор, активізувати ліву кнопку мишки і перетягнути елемент на вільне місце вікна *Circuit*.

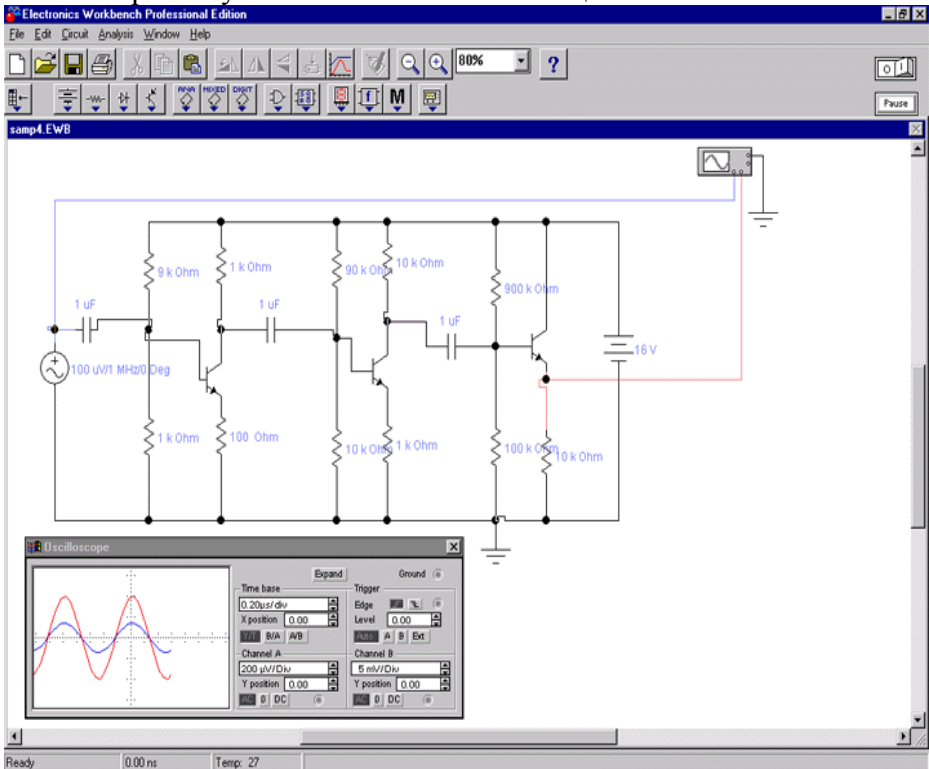


Рис. 3.4.1 - Схема підсилювача

Те ж саме треба зробити з іншими елементами, необхідними для схеми – 11 резисторами і 3 конденсаторами (з набору *Passive*), 3 транзисторами (набір *Active*), батареєю на 16 вольт (набір *Passive*) і елементом «земля». Для того, щоб з'єднати елементи між собою, досить провести мишкою лінію між ними. У випадку необхідності об'єкт можна повернути за допомогою комбінації клавіш Ctrl-R. Щоб ввести параметри елемента, досить двічі клацнути на ньому мишкою.

Для виміру характеристик створеного приладу треба ввести в схему вимірювальний прилад. Для наших цілей найкраще підходить звичайний осцилограф. Один з його каналів підключаємо до входу схеми, виділяємо з'єднання синім кольором. Другий канал підключаємо до виходу підсилювача, виділивши з'єднання червоним кольором.

Для початку експерименту досить клацнути тумблером у правому верхньому куті екрана. На екрані осцилографа (рис. 3.4.2) бачимо дві синусоїди: одну – синього, іншу - червоного кольору, що по раніше прийнятим позначенням відповідають вхідному і вихідному сигналам зібраного пристрою.

Тепер можна підібрати номінали резисторів і конденсаторів, а також виконати оптимальне поєднання транзисторів, постійно відслідковуючи ефективність відображених змін по станах вхідного і вихідного сигналів, спотворенням і т.д.

Не слід забувати, що змінюючи настройки в меню **Circuit/Analysis Options**, можна підібрати таку конфігурацію системи, яка буде найбільш зручна для дослідження саме цієї схеми.

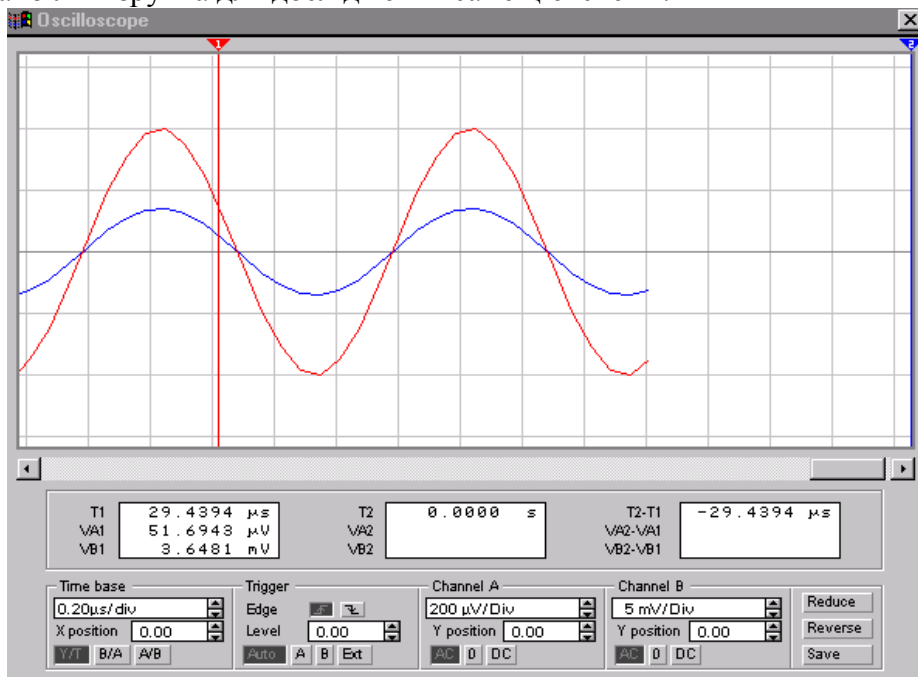


Рис. 3.4.2 – Сигнали на вході і виході

### 3.5. Моделювання тригера Шмідта на аналогових елементах

Особливістю цього електронного пристрою є те, що перехід з одного стану, коли, наприклад, транзистор  $VT1$  закритий, а  $VT2$  відкритий, в інший, коли транзистор  $VT1$  відкритий, а  $VT2$  закритий, відбувається стрибком при визначеній напрузі на базі транзистора  $VT1$ , яка встановлюється змінним резистором  $R3$ . Збираємо пристрій за принциповою схемою (рис 3.5.1).

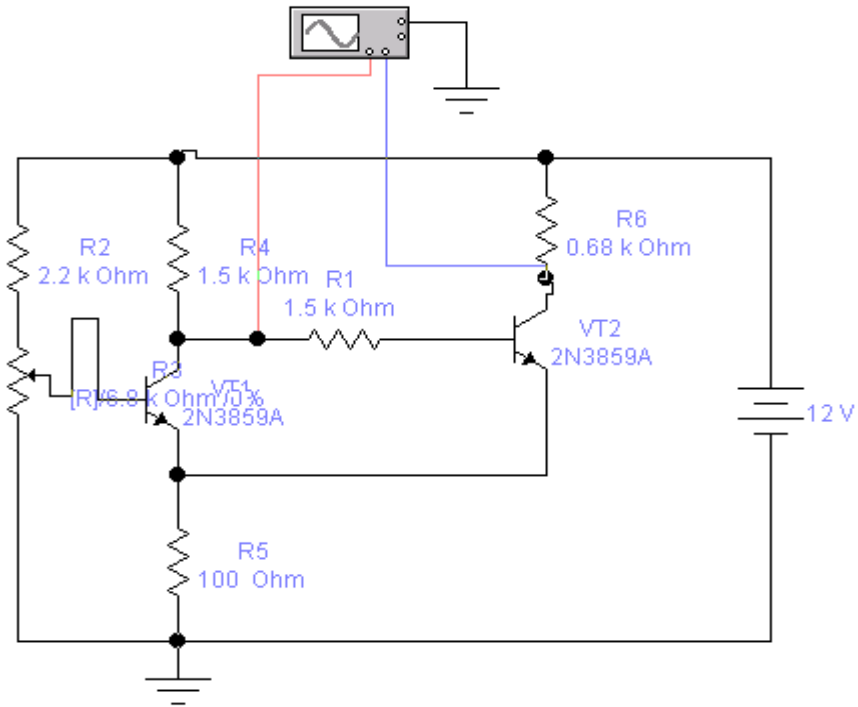


Рис. 3.5.1 – Принципова схема тригера Шмідта

Важливо вибрати правильні моделі і номінали елементів. Як відомо, пакет *Multisim* дозволяє змінювати опори змінних резисторів під час емуляції роботи схеми шляхом натискання клавіш, зазначених в полі **Key** закладки **Value** параметрів змінного резистора для зменшення його опору, і цю ж клавішу одночасно з клавішею *Shift* для

збільшення опору. Початкові значення параметрів змінного резистора в нашій схемі повинні бути такими:

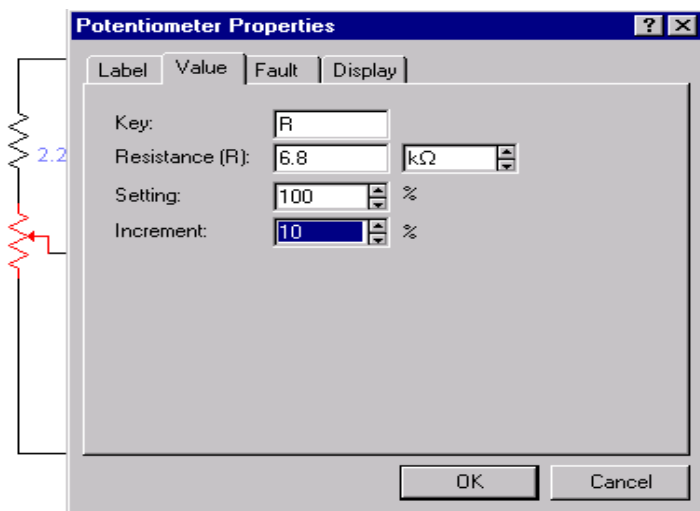


Рис 3.5.2 – Початкові параметри змінного резистора

Модель транзистора можна вибрати наступним чином:

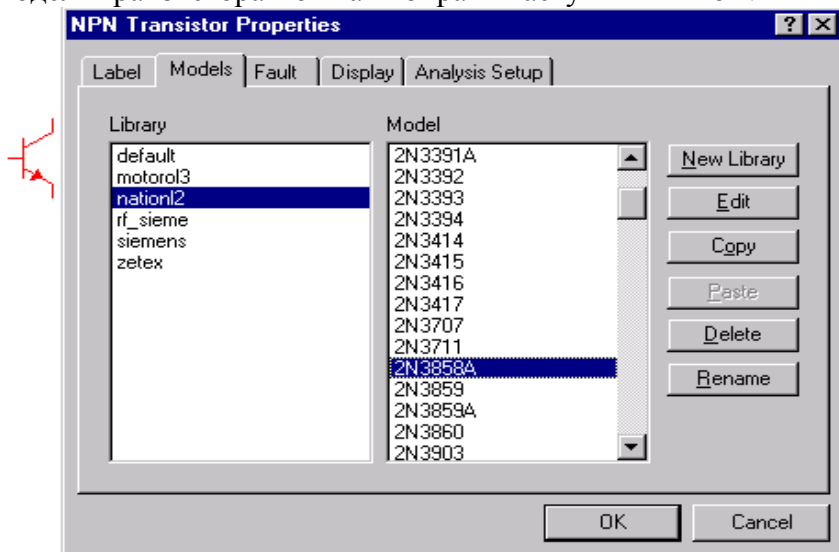


Рис. 3.5.3 – Вибір моделі транзистора

В процесі емуляції схеми зменшуємо опір змінного резистора  $R3$  натисканням клавіші  $R$  на  $10\%$  за кожен раз.

При визначеному опорі  $R3$  відбувається переключення тригера:

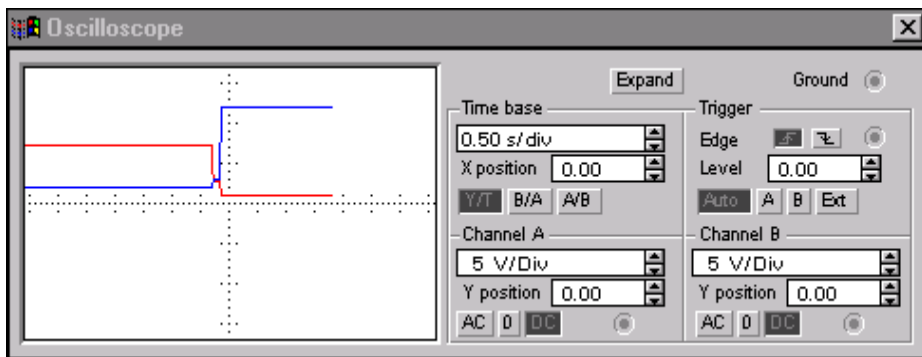


Рис. 3.5.4 – Осцилограма при переключенні тригера

Якщо ж і далі збільшувати значення опору  $R3$ , натискаючи  $Shift-R$ , то можна спостерігати обернене переключення:

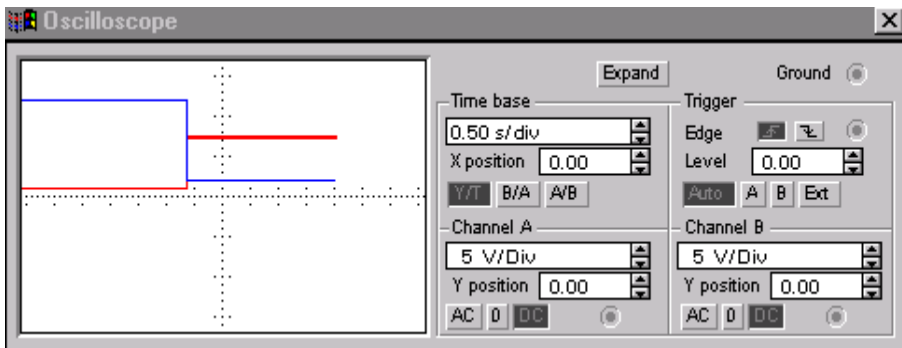


Рис. 3.5.5 – Осцилограма при оберненому переключенні тригера

### 3.6. Моделювання мультивібратора на елементах І-НІ

Збираємо схему:

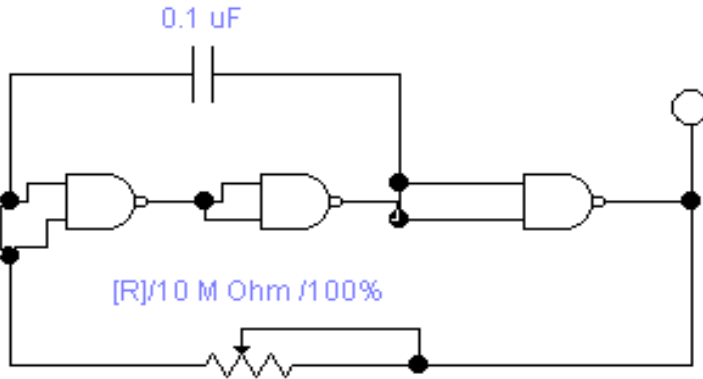


Рис. 3.6.1 – Принципова схема мультивібратора

Змінюючи ємність конденсатора і опір резистора, можна змінювати частоту генерації в широких межах.

### 3.7. Моделювання цифрового генератора

Програмування логічних сигналів в пакеті *Multisim* є процесом трудомістким. Потрібно прописувати картину нулів і одиниць на кожну ітерацію генератора шляхом безпосередньо двійкового коду або ж кодів *ASCII*.

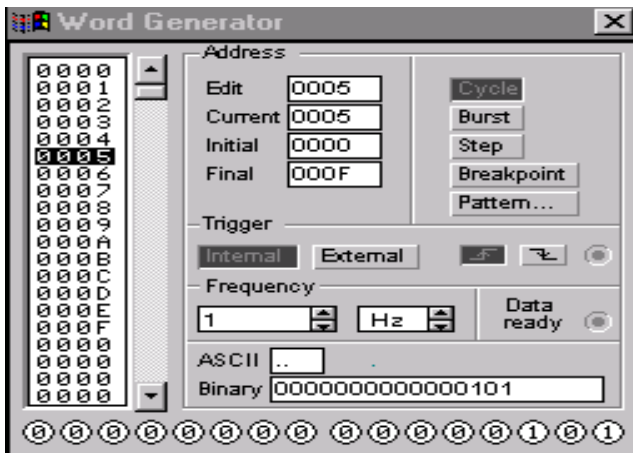


Рис. 3.7.1 – Вікно налаштування цифрового генератора



Збираємо схему, яка містить цифровий генератор, семисегментний цифровий індикатор з дешифратором і чотири логічних індикатори. Програмуємо генератор двійковими значеннями шістнадцяткових цифр. В процесі роботи схема відображає всі шістнадцяткові цифри.

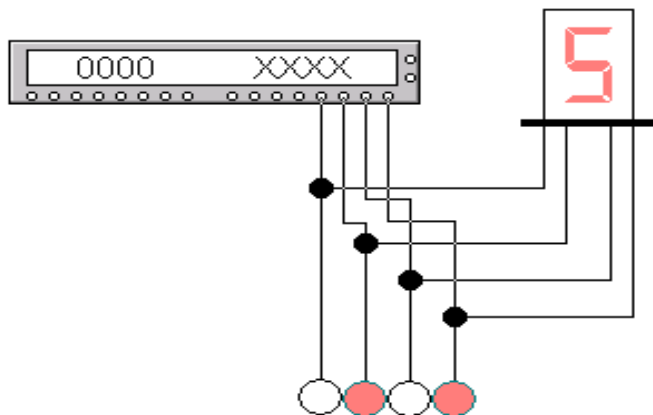


Рис. 3.7.2 – Приклад схеми з цифровим генератором

### 3.8. Моделювання алгебраїчного суматора на ОП

Збираємо в програмі *Multisim* наступну схему:

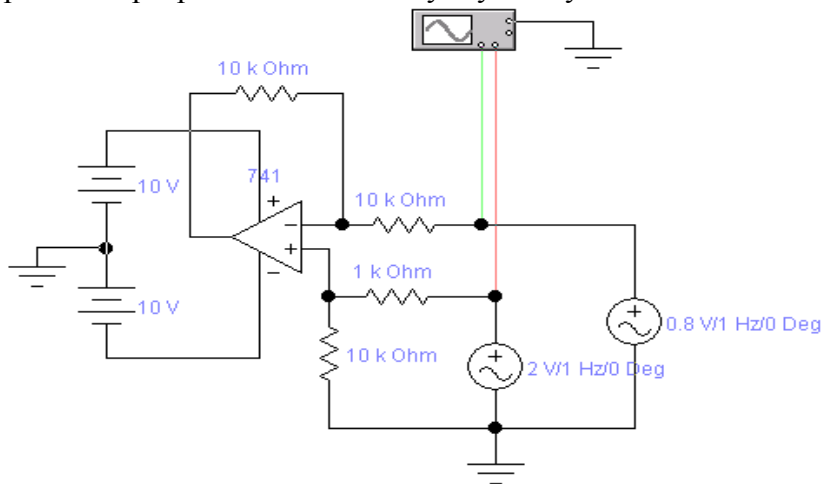


Рис. 3.8.1 – Принципова схема алгебраїчного суматора на ОП

Настроювання компонентів *AC Voltage Source* здійснюється наступним чином:

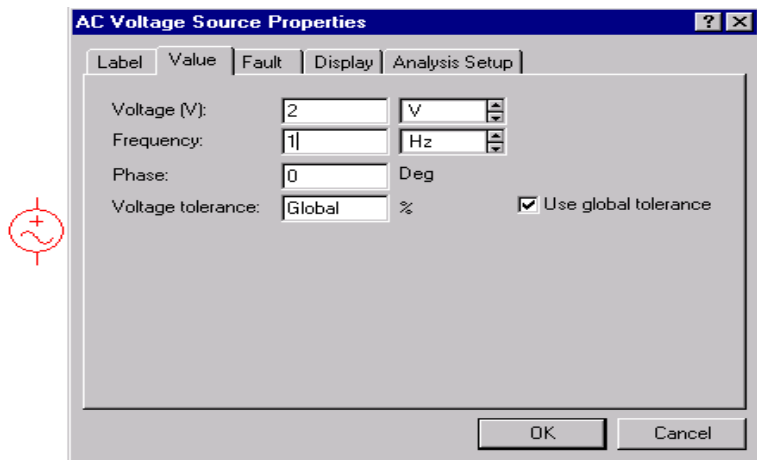


Рис. 3.8.2 – Настроювання компонентів *AC Voltage Source*

Одне з джерел настраюється на діюче значення напруги  $2\text{ V}$ , а інше – на  $0.8\text{ V}$ . Перевіряємо працездатність схеми, активізувавши вікно осцилографа в розширеному режимі і включивши живлення:

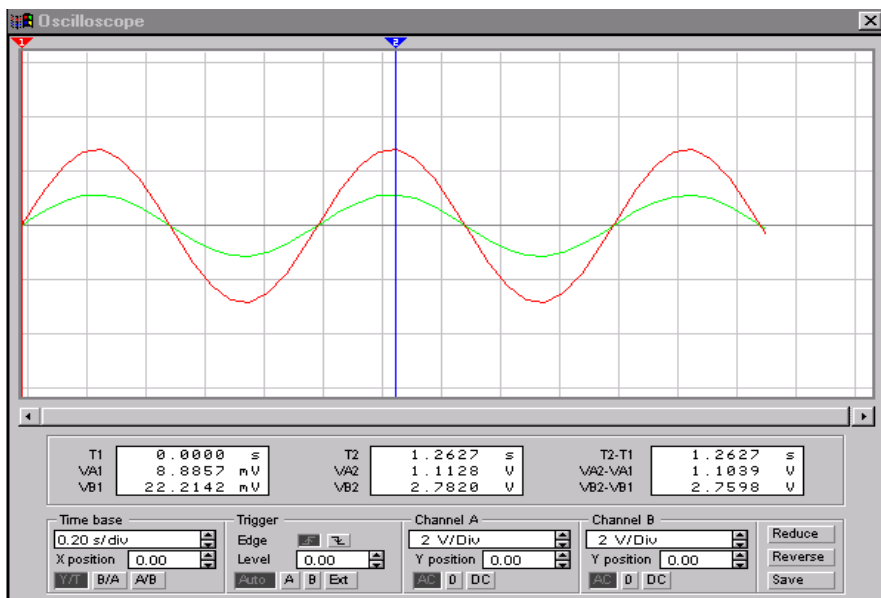


Рис. 3.8.3 – Осцилограма вихідних напруг генераторів

Як бачимо, генератори працюють коректно. Підключаємо один з виводів осцилографа до виходу схеми (рис. 3.8.4).

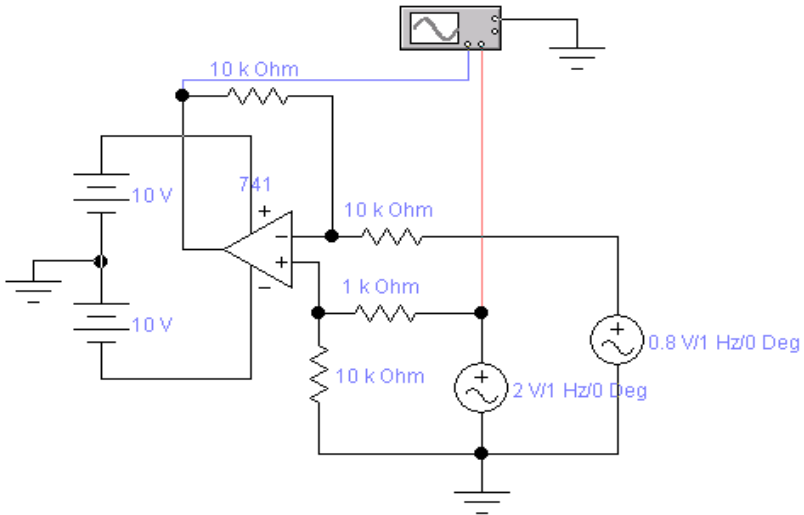


Рис. 3.8.4 - Підключення осцилографа для перевірки роботи суматора

І перевіряємо правильність сумування:

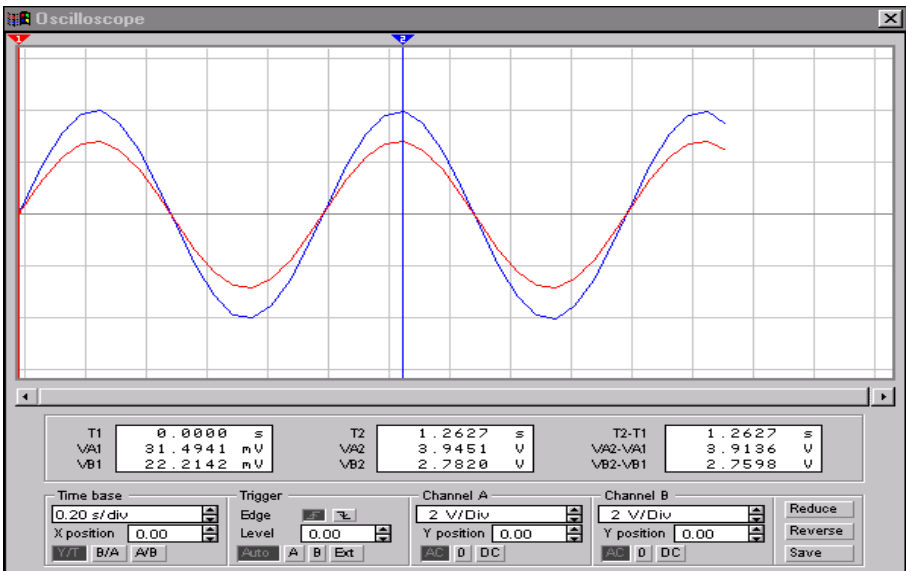


Рис. 3.8.5 – Перевірка роботи суматора

Виконана перевірка показала правильність роботи суматора ( $1.11 + 2.78 \approx 3.9$ ).

### 3.9. Моделювання роботи дешифратора

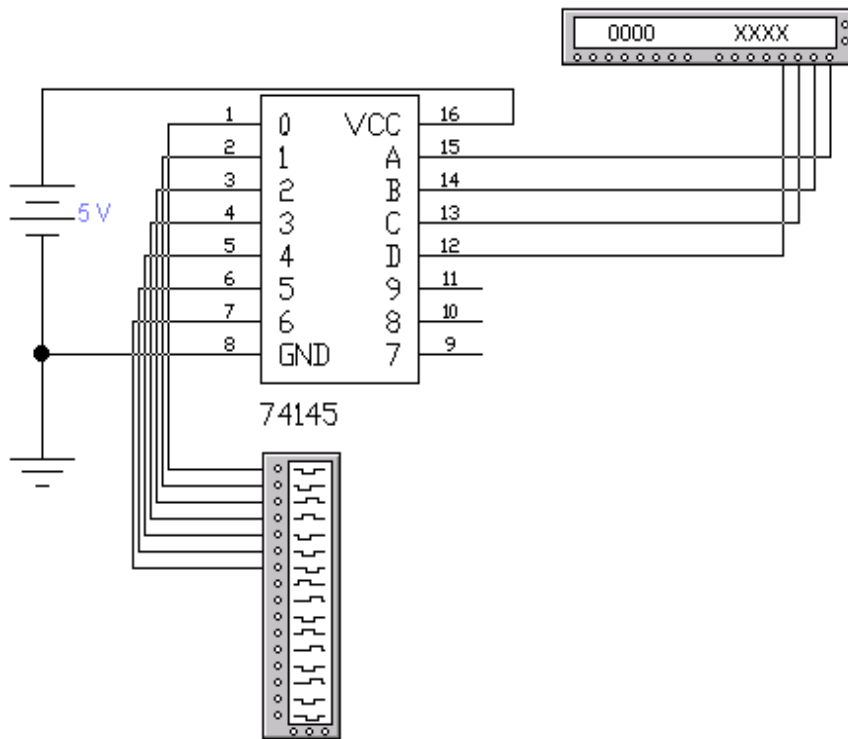


Рис. 3.9.1 - Схема з інтегральним дешифратором

Збираємо схему згідно рис. 3.9.1, а цифровий генератор прорахуємо наступним чином (рис. 3.9.2):

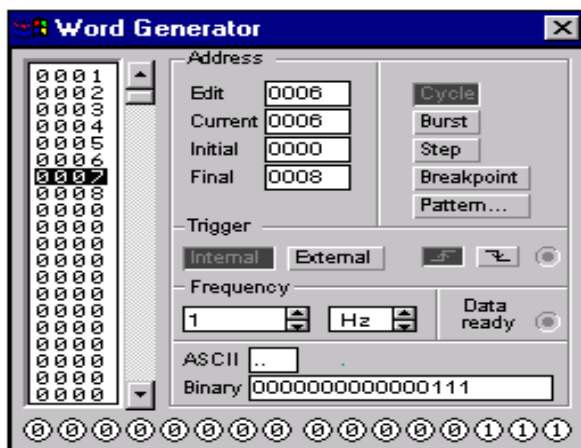


Рис. 3.9.2 - Налаштування цифрового генератора

Активізуємо вікно логічного аналізатора і включаємо живлення схеми.

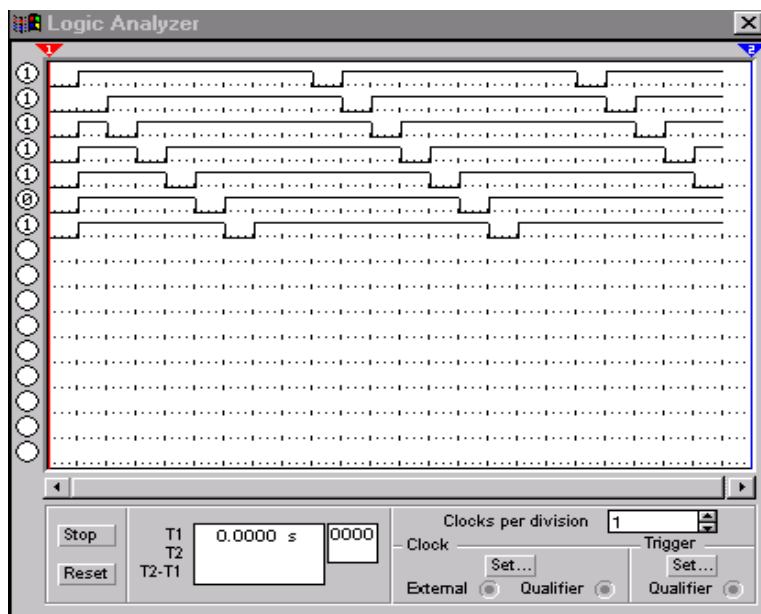


Рис. 3.9.3 – Логічний аналіз вихідних сигналів схеми

В схемі використовується інтегральний дешифратор 74145. Щоб переглянути призначення його виводів, необхідно клацнути мишкою на обраній мікросхемі і вибрати на панелі інструментів «*знак питання*» або *Help* з контекстного меню. З'явиться вікно, яке характеризує дану мікросхему.

#### 4. Завдання до виконання лабораторних робіт

##### Комбінаційні схеми

##### 4.1. Виконання аналізу в середовищі пакету NI Multisim

Метою даної частини лабораторної роботи є освоєння правил і методів проведення різного роду аналізів, а також освоєння і розуміння результатів проведеного аналізу та формування висновків по роботі аналізованої схеми. Усі види аналізу здійснюються за допомогою елементів панелі інструментів *Instruments* (рис. 4.1.1).



Рис. 4.1.1 - Панель інструментів

##### Прилади, які використовуються в процесі моделювання

**Multimeter.** Цей прилад використовується для виміру змінної або постійної напруги чи струму, або опору чи втрати децибелів між двома точками в схемі. *Multimeter* автоматично виставляє діапазони, тому нема потреби самим вказувати діапазон вимірів. Внутрішні опір і струм мають значення, які наближені до ідеальних. Ці значення можуть бути змінені за допомогою активізації кнопки *Settings*.

**Oscilloscope.** Двохканальний осцилограф відображає амплітуду і коливання електронних сигналів. Даний прилад здатний відображати інтенсивність одного або двох сигналів в часі або ж порівнювати одну часову діаграму з іншою.

**Bode plotter.** *Графопобудувач Бодє* рисує графіки частоти відповідей схеми і він корисний для аналізу схем фільтрації. Даний прилад використовується для вимірювання збільшення напруги

сигналу або зсуву фази. Коли *графобудувач Бодє* підключений до схеми, виконується спектральний аналіз.

**Logic Analyzer.** Аналізатор логіки відображає рівні до 16 цифрових сигналів. Він використовується для швидкого збору даних про логічний стан і розширеного часового аналізу, щоб допомогти спроектувати великі системи і надає допомогу при виникненні можливих проблем.

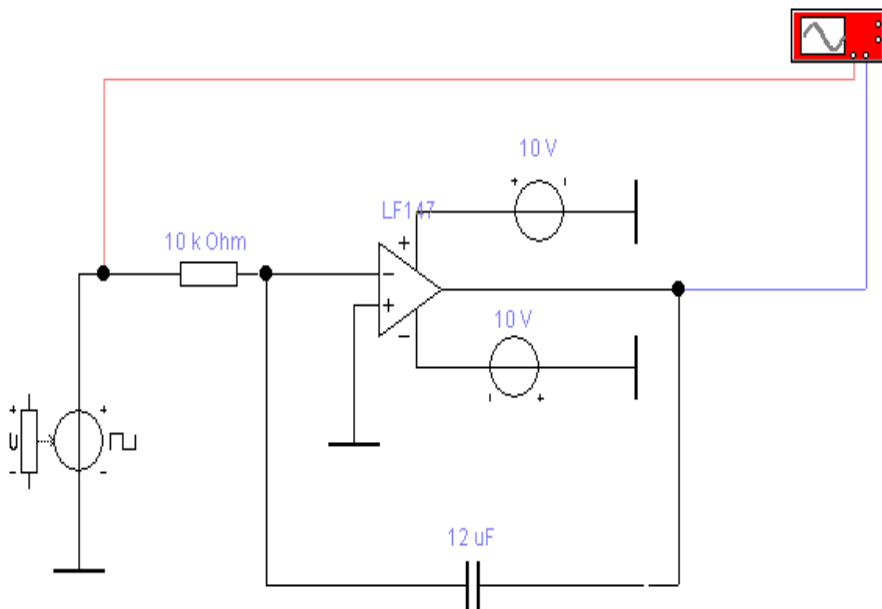


Рис. 4.1.2 - Приклад схеми

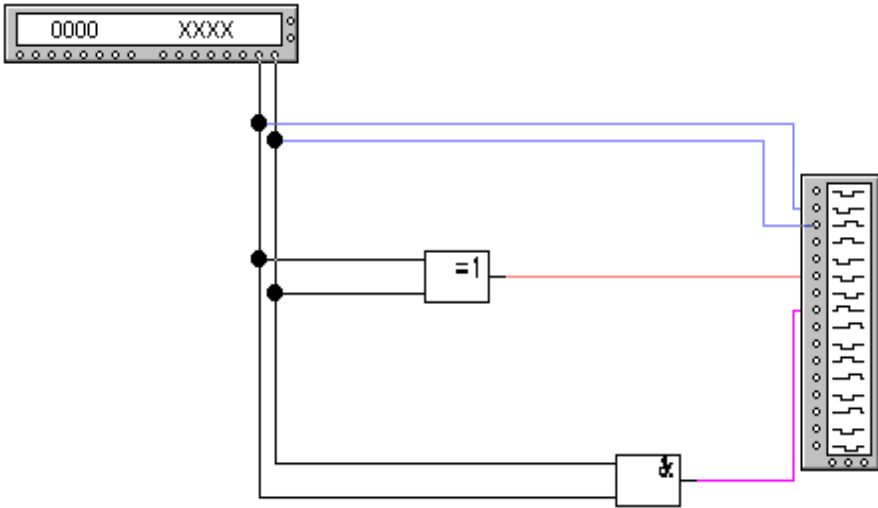


Рис. 4.1.3 - Приклад схеми

### Завдання:

1. Зібрати схеми, приведені на рис. 4.1.2 і 4.1.3.
2. Провести різного роду аналізи з кожною із схем, (особливу увагу необхідно приділити розумінню функцій і вмінню настроїти різні опції).
3. В звіті привести графіки для кожного проведеного аналізу.

## 4.2. Проектування найпростіших логічних схем

Усі пристрої ЕОМ складаються з елементарних логічних схем. Робота цих схем базується на законах і правилах алгебри логіки, яка оперує двома поняттями: істинності і хибності висловлювань. Відповідно до такої двійкової природи висловлень домовилися називати їх *логічними двійковими змінними* і позначати **1** у випадку істинності і **0** у випадку хибності. Прикладами логічних змінних є висловлювання: **A** = “Земля плоска”, **B** = “Автомобіль має двигун”. На підставі цих висловлювань можна записати: **A** = **0**; **B** = **1**, тому що висловлювання **A** помилкове, а висловлювання **B** істинне. Висловлю-



вання можуть бути простими і складними: прості містять одне закінчене твердження, складні утворюються з двох або більшого числа простих висловлювань, зв'язаних між собою деякими логічними зв'язками. Формалізація і перетворення зв'язків між логічними змінними здійснюється відповідно до правил *алгебри логіки* або так званої *алгебри Буля*.

Дві логічні змінні **A** і **B**, які приймають значення **0** або **1**, можуть утворювати логічні функції. В алгебрі логіки будь-які функції зручно зображувати у вигляді таблиці відповідності всіх можливих комбінацій вхідних логічних змінних і вихідних логічних функції – так званою *таблицею істинності*.

Нижче приведено повний перелік функцій двох аргументів (Таблиця 4.2.1).

Функція	Назва функції	X1	0	0	1	1
		X2	0	1	0	1
<b>F1=x1 ∧ x2</b>	Кон'юнкція – логічне множення (І)		0	0	0	1
<b>F2=x1 ∨ x2</b>	Диз'юнкція – логічне додавання (АБО)		0	1	1	1
<b>F3=x1 → x2</b>	Імплікація x1 в x2		1	1	0	1
<b>F4=x1 ← x2</b>	Імплікація x2 в x1		1	0	1	1
<b>F5=x1 ⇒ x2</b>	Заборона x2		0	0	1	0
<b>F6=x1 ⇐ x2</b>	Заборона x1		0	1	0	0
<b>F7=x1 ~ x2</b>	Еквівалентність		1	0	0	1
<b>F8=x1 ⊕ x2</b>	Додавання по модулю 2		0	1	1	0
<b>F9=x1/x2</b>	І-НІ – Штрих Шеффера		1	1	1	0
<b>F10=x1 ↓ x2</b>	АБО-НІ – Стрілка Пірса		1	0	0	0
<b>F11=x1</b>	Повторення x1		0	0	1	1
<b>F12=x2</b>	Повторення x2		0	1	0	1
<b>F13=1</b>	Константа 1		1	1	1	1
<b>F14=0</b>	Константа 0		0	0	0	0
<b>F15=x1^</b>	Інверсія x1 - НІ x1		1	1	0	0
<b>F16=x2^</b>	Інверсія x2 - НІ x2		1	0	1	0

Таблиця 4.2.1 - Таблиця істинності

Функції, утворені логічними змінними, можна перетворювати відповідно до правил або законів алгебри логіки. При цьому прагнуть мінімізувати логічний вираз, тобто привести його до виду, зручного для практичної реалізації на логічних елементах.

При проектуванні логічних схем в пакеті *NI Multisim* використовуються елементи панелі компонентів **Logic Gates**: *Or, Nor, And, Nand, Xor, Xnor, Inverters*, на необхідне число входів. Для імітації роботи спроектованої схеми входи підключаються до генератора тактових імпульсів. Він вибирається з панелі *Instruments* і називається *Word Generator*. Генератор має 16 виходів.

Розглянемо приклад оформлення лабораторної роботи.

Функція  $F_2(A, B)$  - диз'юнкція: логічне додавання двох змінних  $A$  і  $B$  є логічна функція  $F_2$ , яка істинна тоді, коли істинна одна з двох вхідних змінних.  $F_2 = A+B$ . Для функції логічного додавання таблиця істинності має вигляд:

<b>A</b>	0	0	1	1
<b>B</b>	0	1	0	1
<b>F<sub>2</sub></b>	0	1	1	1

Таблиця 4.2.2 – Диз'юнкція - логічне додавання

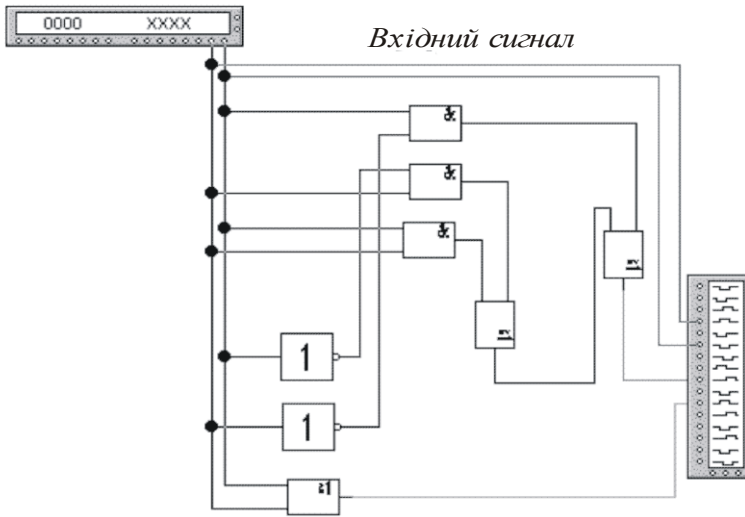


Рис. 4.2.1 - Графічна схема проектування функції додавання

Графічна схема проектування функції додавання, створена в програмі *NI Multisim*, приведена на рис. 4.2.1

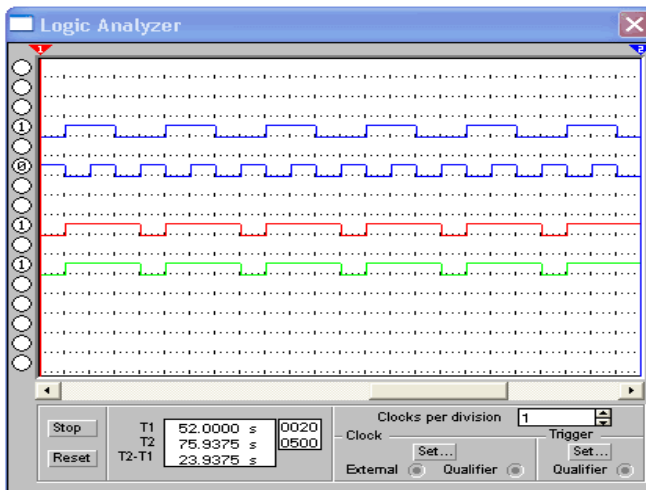


Рис. 4.2.2 - Діаграми вхідних і вихідних сигналів

Крім представлення самої схеми, зібраної в програмі *Multisim*, необхідно привести діаграми вхідних і вихідних сигналів (рис. 4.2.2).

Для перевірки можна порівняти діаграми вхідних і вихідних сигналів з таблицею істинності. Як видно на 1-у такті  $A=1$ ;  $B=0$ ;  $F_2=1$ . Порівняємо це з таблицею істинності і побачимо, що  $F_n$  збігаються. При повному збігу вихідної функції на інших тактах і при іншому поєднанні змінних можна зробити висновок, що побудована схема працює правильно. На цьому опис цієї функції закінчується. Інші функції описуються і збираються аналогічно.

При побудові і проектуванні інших завдань перевірка працездатності побудованої схеми виконується так само. Перевіряються діаграми вхідних і вихідних сигналів і порівнюються по тактах з таблицею істинності.

### Завдання:

1. Використовуючи пакет *NI Multisim*, спроектувати схеми логічних функцій згідно свого варіанту (Таблиця 4.2.3).

2. Оформити звіт, який повинен містити побудовані схеми та діаграми вхідних і вихідних сигналів кожної з виконаних схем.

<i>N</i> <i>варіанту</i>	$F_0$	$F_1$	$F_2$	$F_3$	$F_4$	$F_5$	$F_6$	$F_7$	$F_8$	$F_9$	$F_{10}$	$F_{11}$	$F_{12}$	$F_{13}$	$F_{14}$	$F_{15}$
1	+					+					+					
2		+					+					+				
3			+					+					+			
4				+					+					+		
5					+					+					+	
6						+					+					+
7	+						+					+				
8		+						+					+			
9			+						+					+		
10				+						+					+	
11					+						+					+
12	+					+						+				
13		+					+						+			
14			+					+						+		
15				+					+						+	
16					+					+						+

Таблиця 4.2.3 - Варіанти завдань

### 4.3. Доведення основних тотожностей булевої алгебри за допомогою найпростіших логічних схем

Робота полягає в доведенні основних тотожностей булевої алгебри (Таблиця 4.3.1) за допомогою пакета моделювання *NI Multisim*. Для виконання роботи необхідно використати досвід попередньої роботи в тій частині, де розглядається побудова технічних аналогів логічних рівнянь. Для доказу тотожності необхідно побудувати схеми аналогів логічних рівнянь обох частин тотожності і проаналізувати вихідні параметри кожної схеми.

Для прикладу розглянемо сьому тотожність з таблиці 4.3.1. На рис. 4.3.1 приведені схеми для обох частин тотожності, а на рис. 4.3.2 приведена часова діаграма вхідних і вихідних характеристик для схеми.

№	Логічний вираз	Формулювання
1	$F1 = X * 0 = 0$	Логічний добуток будь-якого аргумента на 0 дорівнює 0
2	$F2 = X * 1 = X$	Логічний добуток будь-якого аргумента на 1 дорівнює значенню аргумента
3	$F3 = X * X = X$	Логічний добуток тих самих аргументів дорівнює аргументу
4	$F4 = X * X^{\wedge} = 0$	Логічний добуток аргумента з його інверсією дорівнює 0
5	$F5 = X + 0 = X$	Логічна сума будь-якого аргумента з 0 дорівнює аргументу
6	$F6 = X + 1 = 1$	Логічна сума будь-якого аргумента з 1 дорівнює 1
7	$F7 = X + X = X$	Логічна сума аргумента із самим собою дорівнює аргументу
8	$F8 = X + X^{\wedge} = 1$	Логічна сума аргумента з його інверсією дорівнює 1
9	$F9 = X^{\wedge\wedge} = X$	Подвійна інверсія аргумента дає його істинне значення
10	$F10 = X1 * X2 = X2 * X1$	Закон перестановки
11	$F11 = X1 + X2 = X2 + X1$	Закон перестановки
12	$F12 = (X1 * X2) * X3 = X1 * (X2 * X3)$	Закон поєднання
13	$F13 = (X1 + X2) + X3 = X1 + (X2 + X3)$	Закон поєднання

14	$F14=X1*(X2+X3)=X1*X2+X1*X3$	Розкриття дужок
15	$F15=X1+(X2*X3)=(X1+X2)*(X1+X3)$	Виключене третє
16	$F16 = X1 + X1 * X2 = X1$	Поглинання
17	$F17 = X1 + X1^{\wedge} * X2 = X1 + X2$	Поглинання
18	$F18 = (X1 * X2)^{\wedge} = X1^{\wedge} + X2^{\wedge}$	1 правило де Моргана
19	$F19 = (X1 + X2)^{\wedge} = X1^{\wedge} * X2^{\wedge}$	2 правило де Моргана

Таблиця 4.3.1 - Тотожності для перетворення логічних виразів

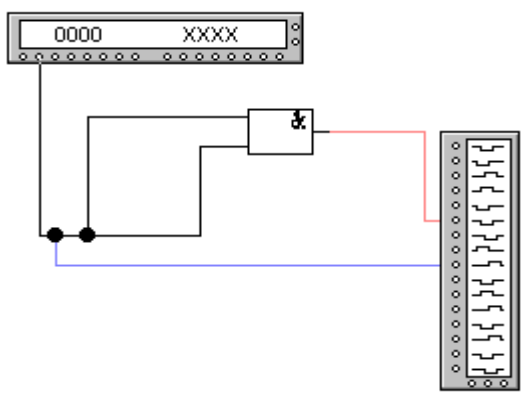


Рис. 4.3.1 - Схема тотожності

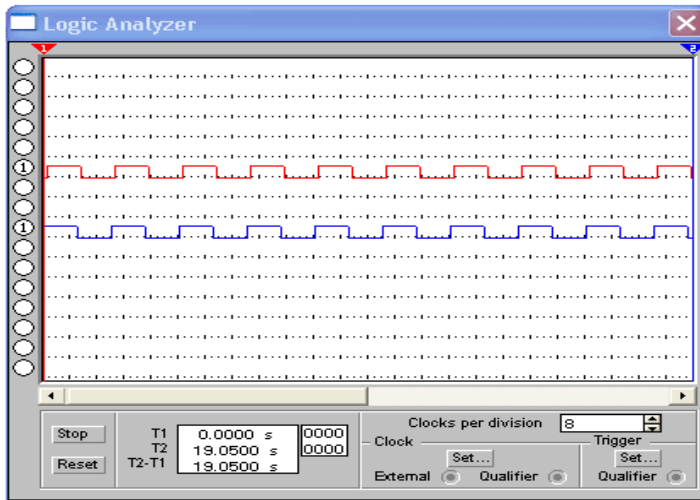


Рис. 4.3.2 - Діаграми вхідних і вихідних сигналів

**Завдання:**

1. Використовуючи пакет *NI Multisim* спроектувати схеми, які відповідають обом частинам тотожності, і за допомогою проведення аналізу довести тотожність.
2. Оформити звіт, який повинен містити побудовані схеми та діаграми вхідних і вихідних сигналів кожної з виконаних схем.
3. Лабораторна виконується відповідно до варіантів (Таблиця 4.3.2).

N варіанта	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19
1	+									+									+
2		+									+								+
3			+									+						+	
4				+									+			+			
5					+									+	+				
6						+								+	+				
7							+						+			+			
8								+				+					+		
9									+		+								+
10					+					+									+

11	+								+		+								
12		+							+			+							
13			+					+					+						
14				+		+								+					
15					+							+			+				
16				+		+										+			
17			+				+										+		
18		+						+										+	
19	+								+										+

Таблиця 4.2.3 - Варіанти завдань

## Абстрактний аналіз логічних схем

### 4.4. Проектування дешифраторів і шифраторів

Дешифратори і шифратори належать до числа перетворювачів кодів. З поняттям шифрації пов'язана уява про компресію даних, з поняттям деширації - обернене перетворення.

Комбінаційна схема, яка перетворює код, який поступає на її вхід, в сигнал тільки на одному з її виходів, називається *дешифратором*.

В умовних позначеннях дешифраторів і шифраторів використовуються букви *DC* і *CD* (від слів *decoder* і *coder* відповідно).

Якщо кількість двійкових розрядів дешифрованого коду позначити через *n*, то число виходів дешифратора повинно бути  $2^n$ . Так як за допомогою *n*-розрядного двійкового коду можна відобразити  $2^n$  кодових комбінацій, число виходів повного дешифратора дорівнює  $2^n$ . Таким чином, дешифратор містить число виходів рівне числу комбінацій вхідних змінних. Наприклад, якщо число входів дорівнює 3, то число виходів дорівнює  $2^3=8$ .

Якщо частина вхідних наборів не використовується, то дешифратор називають неповним і в ньому  $N_{вих} < 2^n$ . В *ЕОМ* за допомогою дешифраторів здійснюється вибірка необхідних комірок запам'ятовуючих пристроїв, розшифровка кодів операцій з видачею відповідних керуючих сигналів і т.д.

Якщо вхідні змінні представити як двійкову систему запису чисел, то логічна одиниця формується на тому виході, номер якого відповідає десятковому запису того ж числа. Наприклад,  $A = 1, B = 0, C = 0, D = 1$  - число *1001* в двійковому коді. В десятковому коді



цьому числу відповідає 9, тобто при даній комбінації вхідних змінних  $F_9 = 1$ . Дешифратори широко використовуються як перетворювачі двійкового коду в десятковий, а також у багатьох інших пристроях. Функціонування дешифратора описується системою логічних рівнянь, складених на основі таблиці істинності.

*Одноступінчатий дешифратор (лінійний)* - найбільш швидкодіючий, але його реалізація при значній розрядності вхідного слова складна, оскільки вимагає застосування логічних елементів з великим числом входів і супроводжується великим навантаженням на джерела вхідних сигналів. Як правило, одноступінчатими виконуються дешифратори на невелике число входів, обумовлене можливостями елементів застосовуваної серії мікросхем (рис. 4.4.1).

В приведеному прикладі на рис. 4.4.1 дешифратор має 3 входи, отже максимальна кількість виходів буде дорівнювати  $2^3=8$ .

Побудова дешифратора здійснюється на основі простих елементів за допомогою таблиці істинності (таблиця 4.4.1) і складених відповідно логічних рівнянь.

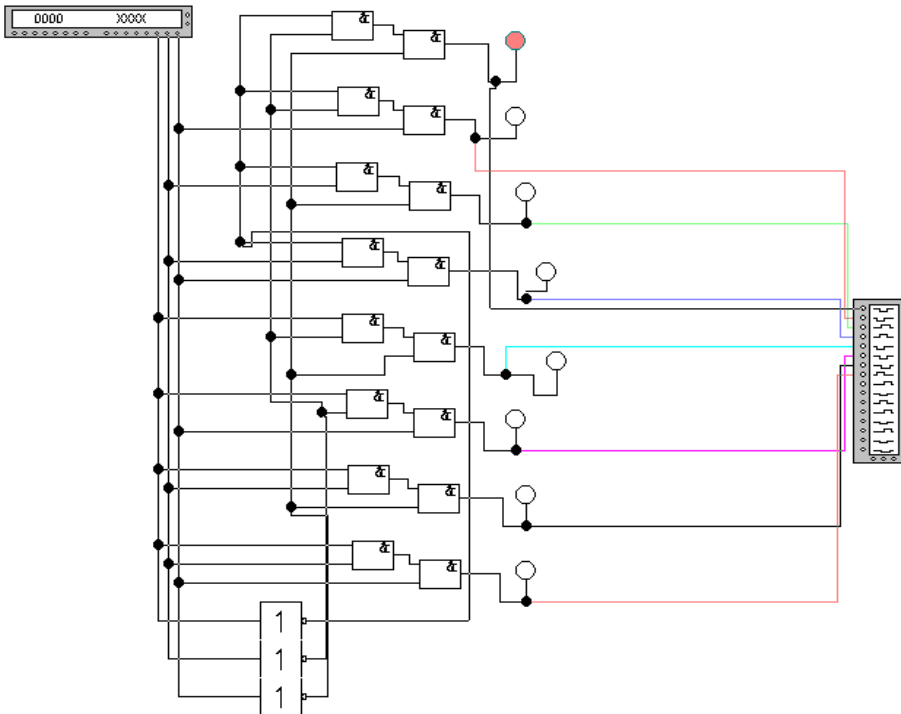


Рис. 4.4.1 - Схема дешифратора на 3 входи і 8 виходів

	A	B	C	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
<b>1</b>	0	0	0	1	0	0	0	0	0	0	0
<b>2</b>	0	0	1	0	1	0	0	0	0	0	0
<b>3</b>	0	1	0	0	0	1	0	0	0	0	0
<b>4</b>	0	1	1	0	0	0	1	0	0	0	0
<b>5</b>	1	0	0	0	0	0	0	1	0	0	0
<b>6</b>	1	0	1	0	0	0	0	0	1	0	0
<b>7</b>	1	1	0	0	0	0	0	0	0	1	0
<b>8</b>	1	1	1	0	0	0	0	0	0	0	1

Таблиця 4.4.1 - Таблиця істинності

Рівняння для побудови:

1.  $Y1=A \wedge B \wedge C \wedge$ ;
2.  $Y5=A B \wedge C \wedge$ ;
3.  $Y2=A \wedge B \wedge C$  ;
4.  $Y6=A B \wedge C$  ;
5.  $Y3=A \wedge B C \wedge$ ;
6.  $Y7=A B C \wedge$ ;
7.  $Y4=A \wedge B C$  ;
8.  $Y8=A B C$  ;

На рис. 4.4.1 приведена часова діаграма роботи дешифратора.

Поява малорозрядних дешифраторів (*пірамідальний* і *матричний*) у вигляді *SIC* (середніх інтегральних схем) порушило питання про застосування їх як засобів побудови дешифраторів більшої розрядності, що дає істотну економію апаратних витрат.

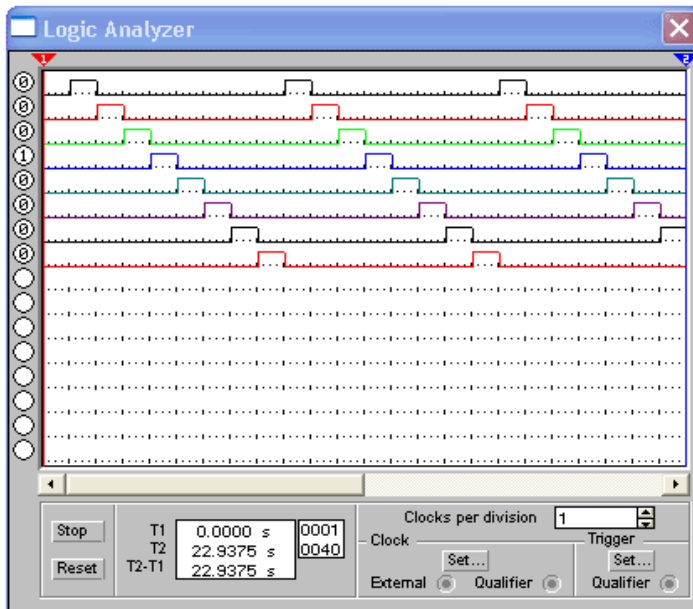


Рис. 4.4.2 - Діаграма роботи дешифратора на 3 входи і 8 виходів

Будь-який потрібний дешифратор може бути побудований за *пірамідальною* структурою. Вхідне слово поділяється на поля,

розрядність яких відповідає числу входів наявних *CIS* дешифраторів, а потім із *CIS* будується пірамідальна структура, що складає сукупність лінійних дешифраторів.

*Матричні* дешифратори формуються на основі простих лінійних дешифраторів меншої розмірності, тобто будуються у вигляді матриці.

*Шифратори.* Двійкові шифратори перетворюють код “ $1$  з  $N$ ” у двійковий код, тобто виконують мікрооперацію, обернену мікрооперації дешифраторів. При збудженні одного із входних кіл шифратора на його виходах формується слово, яке відображає номер збудженого кола. Повний двійковий шифратор має  $2^n$  входів і  $n$  виходів. Одне з основних застосувань шифратора - ввід даних з клавіатури, при якому натискання клавіші з десятковою цифрою повинно приводити до передачі в пристрій двійкового коду даної цифри. На рис. 4.4.3 приведена часова діаграма роботи шифратора, а рис. 4.4.4 - приклад побудови шифратора .

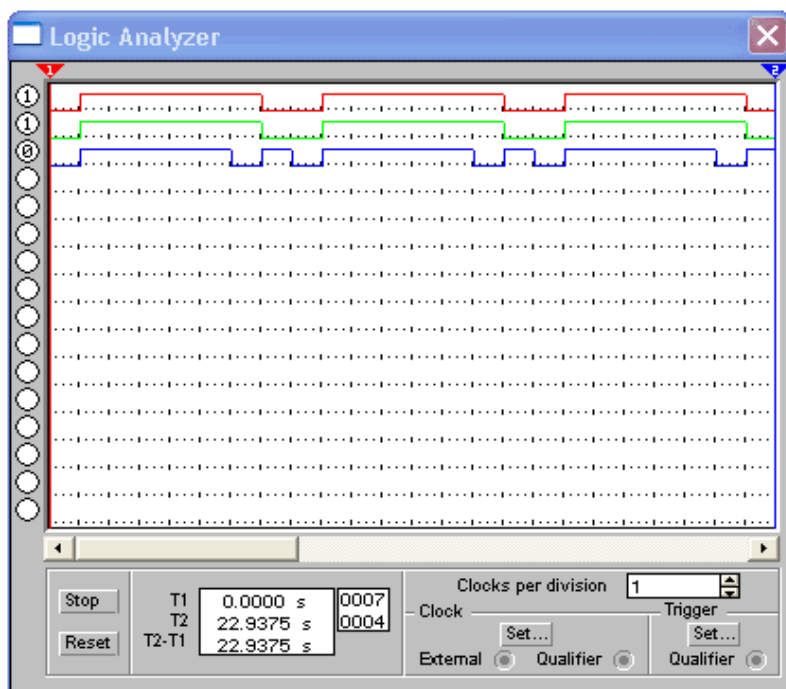


Рис. 4.4.3 - Діаграма роботи шифратора

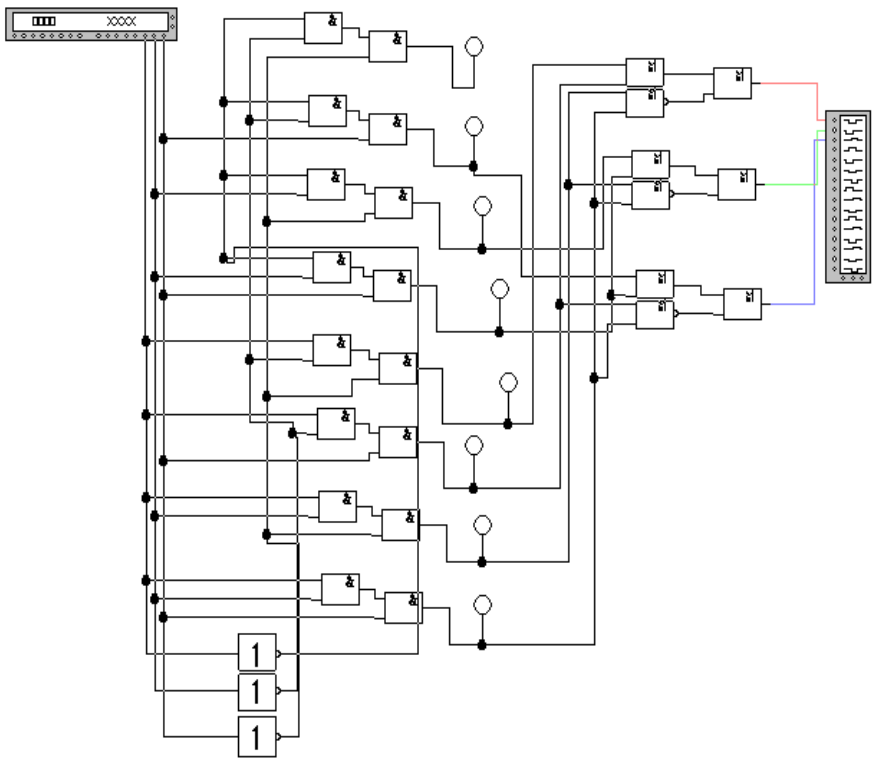


Рис. 4.4.4 - Схема шифратора

**Завдання:**

1. Використовуючи пакет *NI Multisim*, спроектувати схеми на основі найпростіших елементів, використовуючи для складання схеми таблицю істинності і проаналізувати роботу:

- дешифратора;
- шифратора.

2. Оформити звіт, який повинен містити :

- схеми дешифратора і шифратора;
- часові діаграми роботи дешифратора і шифратора.

3. Завдання виконуються відповідно до варіантів:

- а) спроектувати лінійний дешифратор на 4 входи і шифратор;
- б) спроектувати пірамідальний дешифратор на 4 входи і шифратор;

с) спроектувати матричний дешифратор на 4 входи і шифратор.

#### 4.5. Проектування мультиплексора і демультимплексора

Велика частина даних у цифрових системах передається безпосередньо по проводах і провідниках друкованих плат. Як правило, виникає необхідність в багаторазовій передачі інформаційних двійкових сигналів з одного місця в інше. В деяких випадках потрібно передавати дані на великі відстані по телефонних лініях і кабелях. Якби всі дані передавалися одночасно по паралельних лініях зв'язку, загальна довжина таких кабелів була б занадто велика і вони були б занадто дорогими. Замість цього дані передаються по одному проводі в послідовній формі і групуються в паралельні дані на прийомному кінці цієї єдиної лінії зв'язку. Пристрої, що використовуються для послідовної пересилки і прийому даних, називаються відповідно *мультиплексор* і *демультимплексор*. Паралельні дані одного з цифрових пристроїв за допомогою мультиплексора перетворюються в послідовні інформаційні сигнали, які передаються по одному проводу. На виході демультимплексора ці послідовні сигнали знову групуються в паралельні дані. Входи мультиплексора поділяються на дві групи: *інформаційні* і *керуючі* (адресуючі).

*Мультиплексор* (*multiplexor*) – це комбінаційний пристрій, який забезпечує комутацію одного з входів на загальний вихід під керуванням сигналів на адресних входах.

Синтез мультиплексора можна виконати за схемою, приведеною на рис. 4.5.1. Кількість робочих входів  $N_{роб} = 2^{N_{адрес}}$ .

x0	x1	x2	I	Y
0	0	0	i0	i0
0	0	1	i1	i1
0	1	0	i2	i2
0	1	1	i3	i3
1	0	0	i4	i4
1	0	1	i5	i5
1	1	0	i6	i6

1	1	1	i7	i7
---	---	---	----	----

Таблиця 4.5.1 - Таблиця істинності мультимплексора

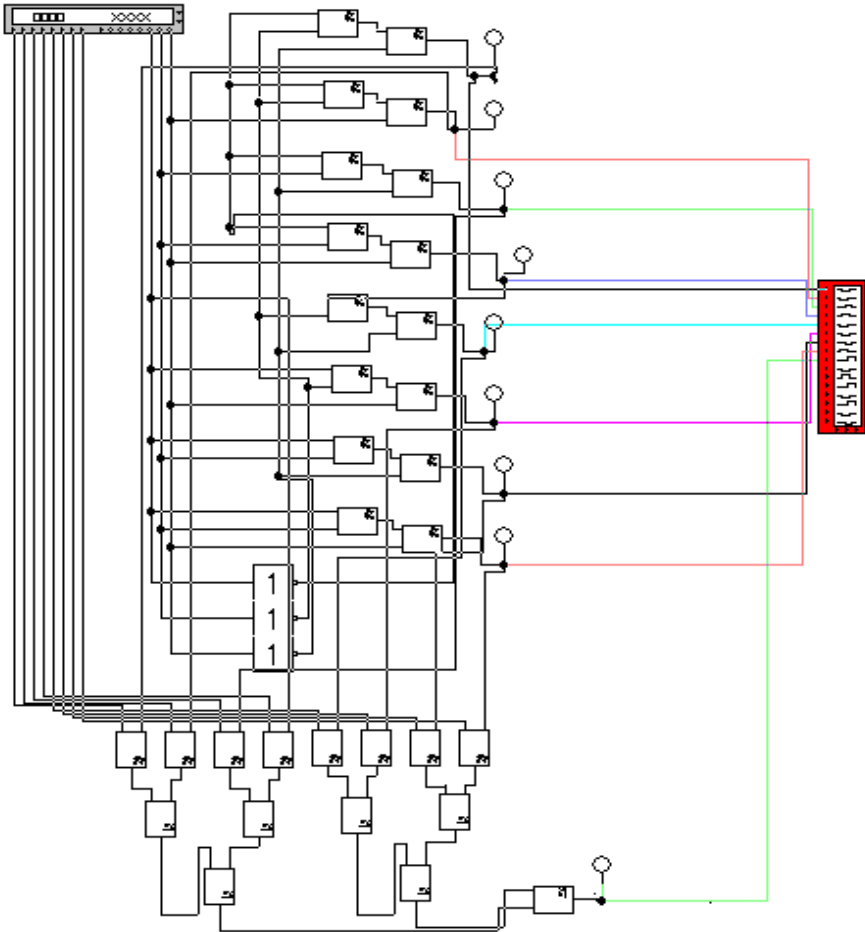


Рис. 4.5.1 - Схема мультимплексора

$$y = x_0^*x_1^*x_2^*i_0 + x_0^*x_1^*x_2^*i_1 + \dots + x_0^*x_1^*x_2^*i_7$$

На рис. 4.5.2 приведена часова діаграма роботи мультимплексора.

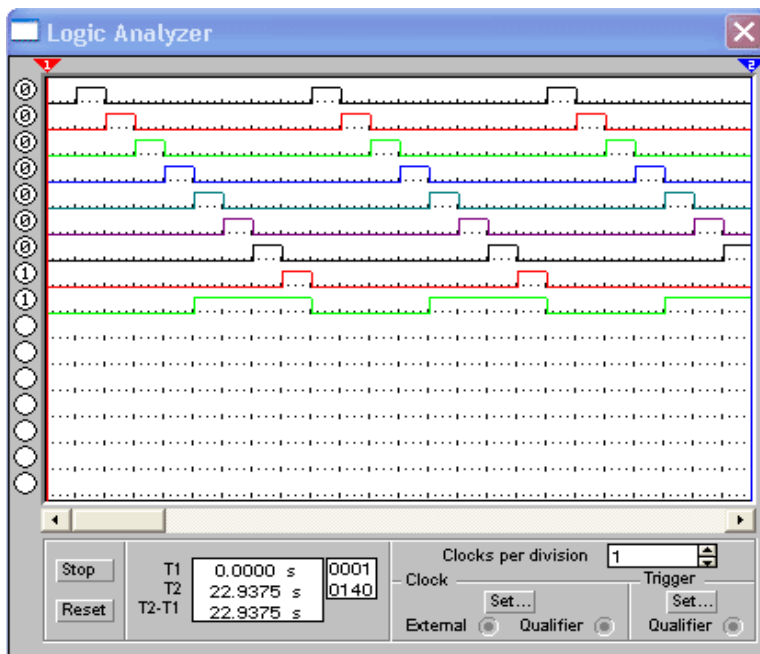


Рис. 4.5.2 - Діаграма роботи мультимплексора

**Демультимплексор** – це комбінаційний пристрій, який забезпечує комутацію єдиного входу на один із виходів під керуванням сигналів на адресних входах (рис. 4.5.3).

Демультимплексор виконує операцію обернену мультимплексору, тобто перетворює послідовний сигнал, який передається, в паралельну комбінацію сигналів.

I	x0	x1	x2	y0	y1	y2	y3	y4	y5	y6	y7
<b>i0</b>	0	0	0	<b>i0</b>	0	0	0	0	0	0	0
<b>i1</b>	0	0	1	0	<b>i1</b>	0	0	0	0	0	0
<b>i2</b>	0	1	0	0	0	<b>i2</b>	0	0	0	0	0
<b>i3</b>	0	1	1	0	0	0	<b>i3</b>	0	0	0	0
<b>i4</b>	1	0	0	0	0	0	0	<b>i4</b>	0	0	0
<b>i5</b>	1	0	1	0	0	0	0	0	<b>i5</b>	0	0
<b>i6</b>	1	1	0	0	0	0	0	0	0	<b>i6</b>	0
<b>i7</b>	1	1	1	0	0	0	0	0	0	0	<b>i7</b>

Таблиця 4.5.2 - Таблиця істинності демультимплексора на 3 адресних входах



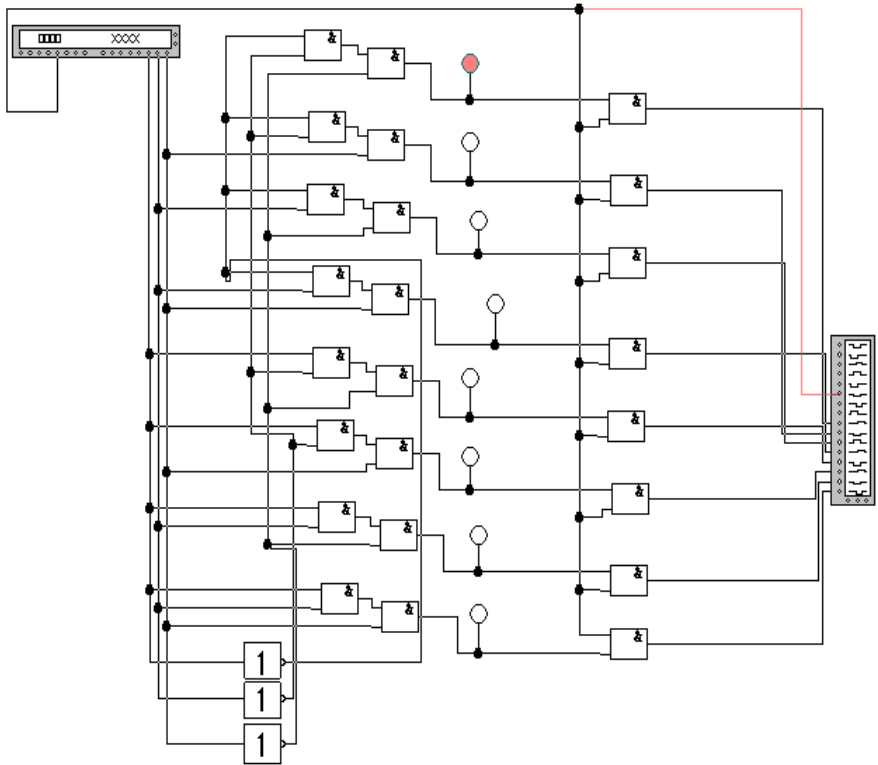


Рис. 4.5.3 - Схема демультиплектора

Логічні рівняння демультиплектора або виходи рівні:

$$y_0 = x_1 \wedge x_2 \wedge x_3 \wedge A$$

$$y_1 = x_1 \wedge x_2 \wedge x_3 \wedge \bar{A}$$

...

$$y_7 = x_1 \wedge x_2 \wedge x_3 \wedge A$$

На рис. 4.5.4 приведена часова діаграма роботи демультиплектора.

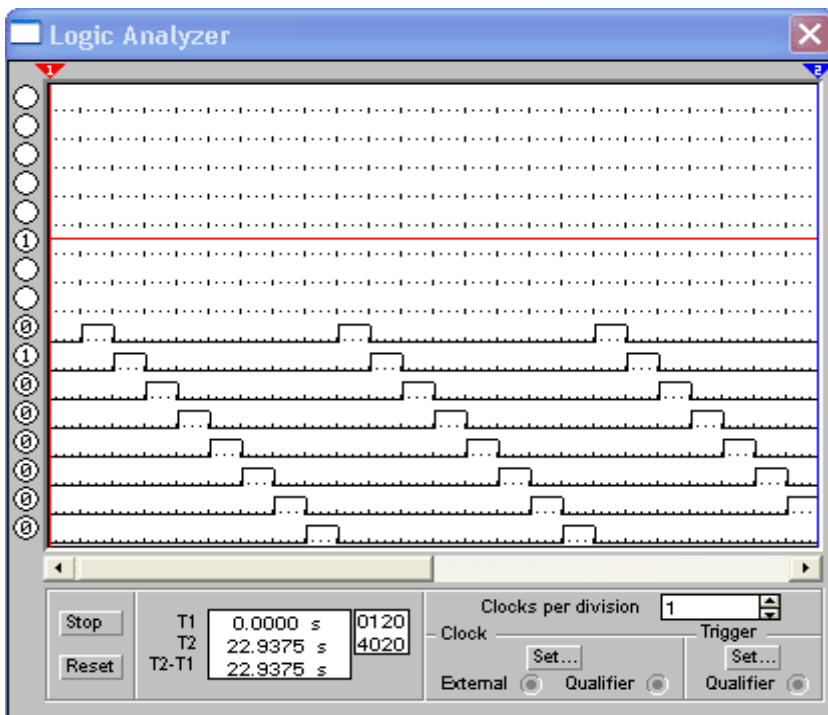


Рис. 4.5.4 - Діаграма роботи демультиплектора

### Завдання:

1. Використовуючи пакет *Multisim*, спроектувати схеми на основі найпростіших елементів, використовуючи для складання схеми таблицю істинності і проаналізувати роботу:

- мультиплектора;
- демультиплектора.

2. Скласти звіт про виконання лабораторної роботи, в якому привести:

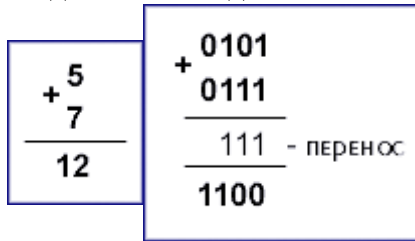
- схеми мультиплектора і демультиплектора;
- часові діаграми роботи мультиплектора і демультиплектора.

*Завдання виконати відповідно до свого варіанту.*

1. Спроектувати мультиплексор на 4 входи.
2. Спроектувати демультиплексор на 4 входи.

#### 4.6. Проектування суматора

Суматор є найпростішим цифровим пристроєм. Це вузол ЕОМ, який виконує арифметичне сумування кодів чисел, тобто він призначений для додавання двох чисел, заданих в двійковому коді. Порівняємо сумування десяткових і двійкових чисел:



#### Правила додавання двійкових і десяткових чисел однакові:

- додавання виконується порозрядно – від молодшого розряду до старшого;
- в молодшому розряді обчислюється сума молодших розрядів доданків  $A_i$  і  $B_i$ . Ця сума в даній системі числення може бути записана однозначним числом  $S_i$  або двохзначним числом  $P_i S_i$ . Функція  $P$  називається *переносом*;
- в усіх наступних розрядах знаходиться сума даних розрядів доданків  $A_i$  і  $B_i$ , причому при  $P_{i-1}=1$  до цієї суми додається одиниця (у числових прикладах, наведених вище, цей випадок виділений жирним шрифтом; результат додавання в  $i$ -му розряді записується у вигляді однозначного  $S_i$  або двохзначного  $P_i S_i$  числа).

Таким чином, в кожному розряді необхідно знайти суму  $A_i$ ,  $B_i$  і  $P_{i-1}$  (якщо  $P_{i-1}=1$ ), тобто визначити  $S_i$  і  $P_i$ . По числу входів розрізняють *напівсуматори*, *однорозрядні суматори (ОС)* і *багаторозрядні суматори*.

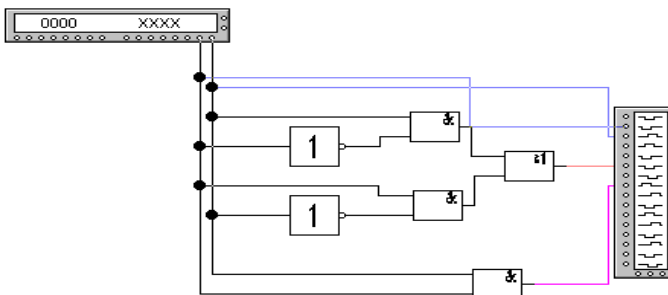


Рис. 4.6.1 - Схема напівсуматора

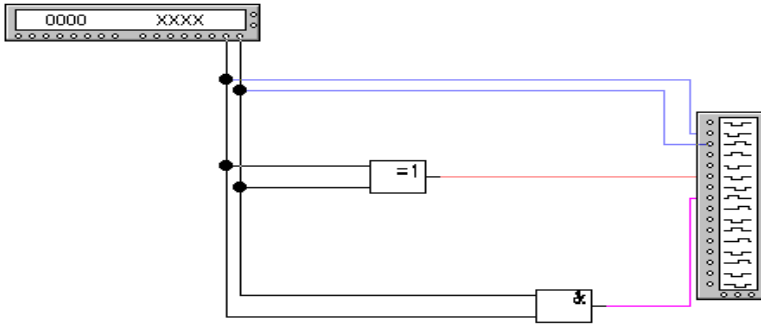


Рис. 4.6.2 - Схема напівсуматора з використанням стандартної схеми додавання по модулю 2

*Напівсуматорами* називаються пристрої з двома входами і двома виходами, на яких виникають сигнали суми і переносу. Напівсуматор реалізує тільки частину задачі сумування, так як не враховує вхідної величини – переносу із сусіднього молодшого розряду в даних (див. рис. 4.6.1 і 4.6.2). В таблиці 4.6.1 приведена таблиця істинності напівсуматора, а на рис. 4.6.3 приведена часова діаграма роботи напівсуматора.

<b>X<sub>i</sub></b>	<b>Y<sub>i</sub></b>	<b>S<sub>i</sub></b>	<b>PI+1</b>
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Таблиця 4.6.1 - Таблиця істинності напівсуматора

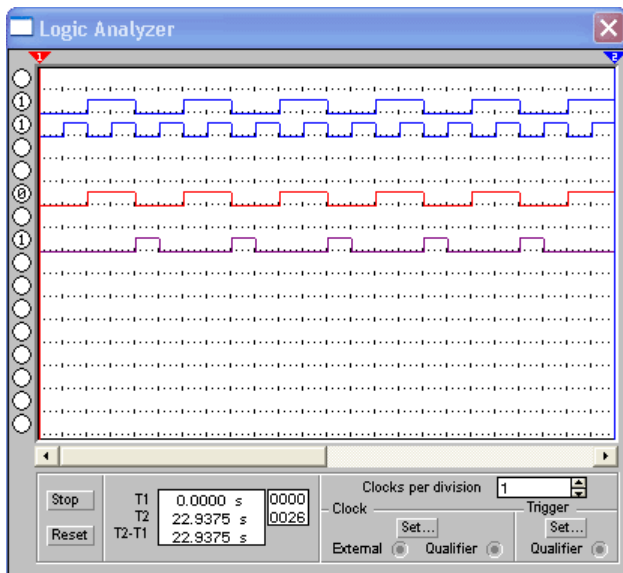


Рис. 4.6.3 - *Діаграма роботи напівсуматора*

Логічне рівняння:  $S = x \wedge y + x * y \wedge P = x * y$

*Однорозрядний двійковий суматор складається з двох комбінаційних схем: одна - для формування  $S_i$ , друга - для визначення  $P_i$  (див. рис. 4.6.4 і 4.6.5).*

*Багаторозрядний суматор будується на основі однорозрядних відповідно до правил додавання.*

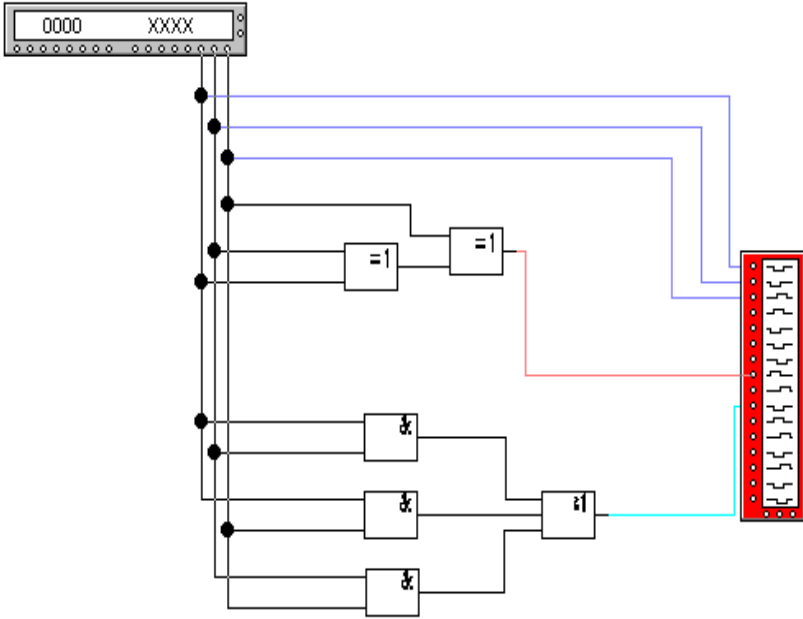


Рис. 4.6.4 - Схема однорядного двійкового суматора

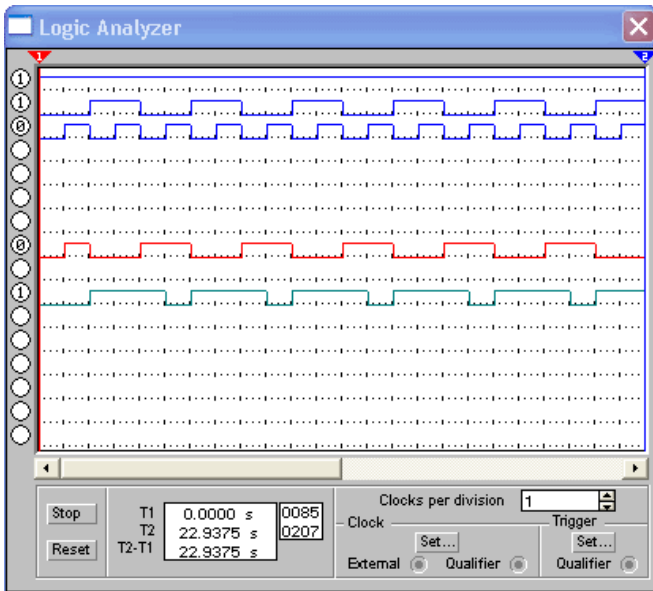


Рис. 4.6.5 - Діаграма роботи однорядного двійкового суматора

Однорозрядні суматори мають три входи і забезпечують додавання розрядів доданків і перенесення з попереднього розряду (Таблиця 4.6.2).

<b>Xi</b>	<b>Yi</b>	<b>Pi</b>	<b>Si</b>	<b>Pi+1</b>
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Таблиця 4.6.2 - Таблиця істинності суматора

Логічні рівняння:

$$S=y1+y2+y3+y4$$

$$Y1=x1^{\wedge}x2^{\wedge}x3$$

$$Y2=x1^{\wedge}x2*x3^{\wedge}$$

$$Y3=x2^{\wedge}x3^{\wedge}x1$$

$$Y4=x1*x2*x3$$

$$Pi+1=y1+y2+y3+y4$$

$$Y1=x1^{\wedge}x2*x3$$

$$Y2=x1*x2^{\wedge}x3$$

$$Y3=x1*x2*x3^{\wedge}$$

$$Y4=x1*x2*x3$$

Мінімізуючи дані рівняння відповідно до формул алгебри логіки, побудуємо суматор в середовищі пакету *NI Multisim*.

В залежності від характеру вводу-виводу кодів і організації переносів багаторозрядні суматори бувають *послідовного* і *паралельного* принципу дії.

В *послідовному* суматорі додавання кодів здійснюється поразрядно, починаючи з молодшого розряду, за допомогою комбінаційного суматора на три входи. Перенос  $P_{j+1}$ , який утворюється в даному розряді, затримується на час  $t$  і поступає на вхід  $P_j$  суматора в момент надходження наступного розряду доданків. Таким чином, послідовно розряд за розрядом виконується додавання

кодів чисел. Перевагою послідовного суматора є простота апаратної реалізації, а недоліком - досить великий час сумування (див. рис. 4.6.6). На рис. 4.6.7 приведена часова діаграма, яка ілюструє роботу послідовного суматора.

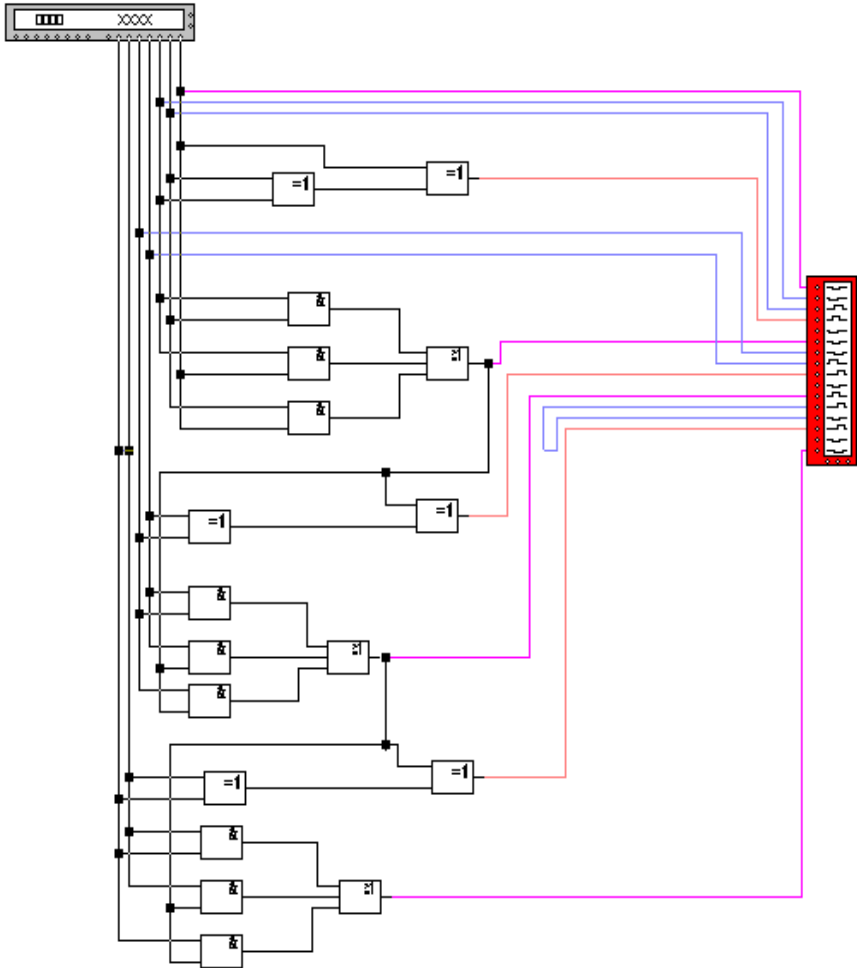


Рис. 4.6.6 - Схема послідовного суматора



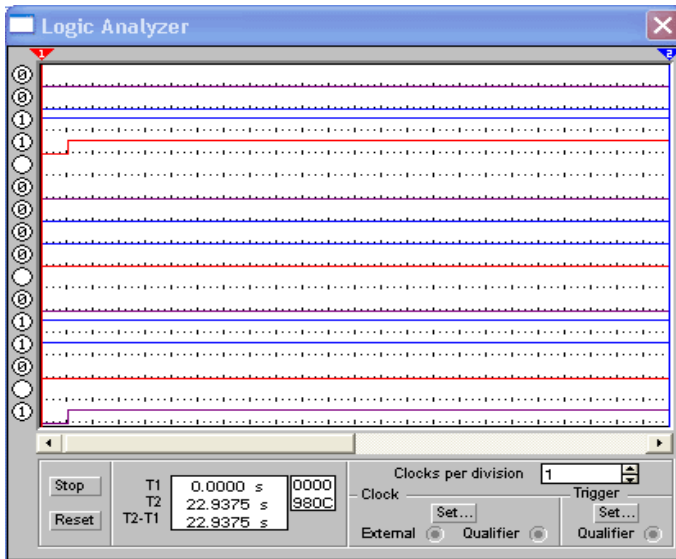


Рис. 4.6.7 - Діаграма роботи послідовного суматора

В паралельному суматорі досягається більш висока швидкодія. Просумовані коди надходять на входи суматора одночасно по всіх розрядах. Для цього в кожному розряді використовується комбінаційний суматор на три входи, на виходах якого утворюються значення суми  $S_j$  даного розряду і переносу  $P_{j+1}$  у старший розряд. В процесі поширення сигналу переносу встановлюється остаточне значення суми в кожному розряді. Очевидно, що протягом цього часу на входах суматора присутні сигнали  $X_i$ ,  $Y_i$ , які відповідають кодам сумування. Максимальне за часом сумування виходить у тому випадку, коли перенос, що виник у першому розряді, поширюється по всіх розрядах (наприклад, при додаванні кодів 11..11 і 00..01). В паралельному суматорі, як правило, застосовуються різні способи прискорення переносу (паралельний перенос, груповий і т.д.).

Цифрові схеми порівняння формують на виході  $F=1$  при рівності поданих на вхід двох двійкових чисел  $A$  (порозрядно записуємо  $a$  і  $b$ ) і  $B$  ( $c$  і  $d$ ). Цифрова схема порівняння - це цифровий аналог компаратора (див. рис. 4.6.8), що є одним з найважливіших пристроїв імпульсної техніки; часова діаграма цифрової схеми порівняння приведена на рис. 4.6.9. На основі таблиці істинності для

компаратора (таблиця 4.6.3) складаємо рівняння (для  $A > B$ ,  $A < B$ ,  $A = B$ ), мінімізуємо їх, використовуючи закони алгебри логіки.

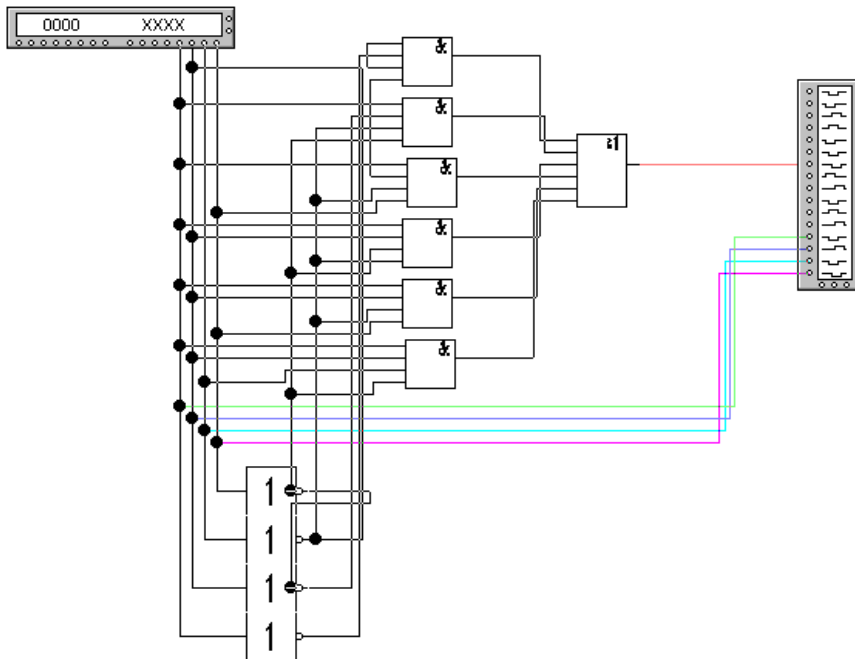


Рис. 4.6.8 - Цифрова схема порівняння

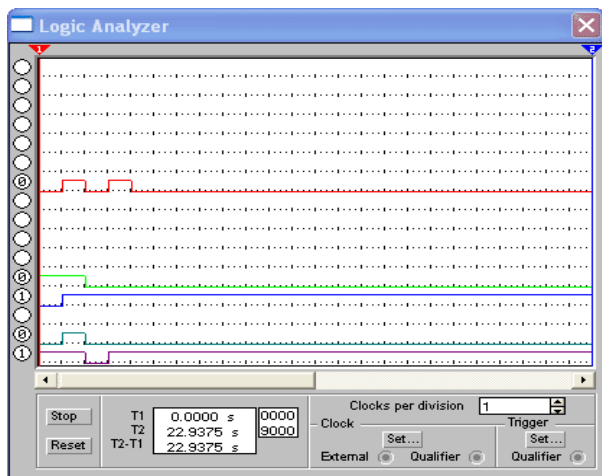


Рис. 4.6.9 - Діаграма роботи цифрової схеми порівняння

A		B		A>B	A<B	A=B
a	b	c	d			
0	0	0	0	0	0	1
0	0	0	1	0	1	0
0	0	1	0	0	1	0
0	0	1	1	0	1	0
0	1	0	0	1	0	0
0	1	0	1	0	0	1
0	1	1	0	0	1	0
0	1	1	1	0	1	0
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	0	1
1	0	1	1	0	1	0
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	0	1

Таблиця 4.6.3 - Таблиця істинності цифрової схеми порівняння

Логічні рівняння:

$$A > B = a^b c^d + a b^c d + a b^c d + a b c^d + a b c^d + a b c d$$

$$A < B = a^b c^d + a^b c d + a^b c d + a^b c d + a^b c d + a b^c d$$

$$A = B = a^b c^d + a b^c d + a b c^d + a b c d$$

Завдання:

1. Використовуючи пакет *NI Multisim*, спроектувати схему на основі найпростіших елементів, застосовуючи для складання схеми таблицю істинності і проаналізувати роботу суматора або схеми порівняння.
2. Скласти звіт про виконану лабораторну роботу, в якій відобразити:
  - схеми, побудовані в пакеті *NI Multisim*;
  - часові діаграми роботи побудованих схем.

Загальне завдання:

- спроектувати однорозрядний суматор.

Завдання по варіантах:

1. Спроектувати 4-розрядний суматор з послідовним переносом.

- |   |             |
|---|-------------|
| 2. Спроекувати цифрову схему порівняння двійкових чисел $A < B$ . | 2-розрядних |
| 3. Спроекувати цифрову схему порівняння двійкових чисел $A > B$ . | 2-розрядних |
| 4. Спроекувати цифрову схему порівняння двійкових чисел $A = B$ . | 2-розрядних |

### Аналіз послідовних схем

#### 4.7. Проектування тригерів

**Тригер** – це цифровий автомат, що має два стійких стани рівноваги: або  $0$ , або  $1$ . Стан тригера розпізнається по його вхідному сигналу. Під впливом вхідного сигналу тригер стрибкоподібно переходить з одного стійкого стану в інший, при цьому стрибкоподібно змінюється рівень напруги його вихідного сигналу. Для зручності використання в схемах обчислювальних пристроїв тригери, як правило, мають два виходи: *прямий*  $Q$  (називається також “вихід  $1$ ”) і *інверсний*  $Q^{\wedge}$  (“вихід  $0$ ”). В одиничному стані тригери на виході  $Q$  мають високий рівень сигналу, а в нульовому – низький. На виході  $Q^{\wedge}$  - навпаки.

Якщо хоча б з одного входу інформація в тригер заноситься примусово під впливом синхронізуючого сигналу, то тригер називається *синхронізуючим (синхронним)*. Якщо занесення інформації в тригер з будь-якого входу здійснюється без синхронізуючого сигналу, то тригер називається *несинхронізуючим (асинхронним)*.

Стан тригера визначається сигналом  $Q$  на прямому виході тригера (або сигналом  $Q^{\wedge}$  на його інверсному виході). Закони функціонування тригерів задаються таблицями переходів з компактним записом, при якому в стовбці станів може бути вказано, що новий стан збігається з попереднім або є його запереченням.

#### Типи тригерів.

*Тригер типу RS* має два входи роздільної установки в нульовий і одиничний стани. Вплив по входу  $S$  (позначений по першій букві слова *set* – установка) приводить тригер в одиничний стан, а вплив

по входу  $R$  (від першої букви слова *reset* – скидання) – в нульовий. Одночасна подача сигналів  $S$  і  $R$  не допускається, що є недоліком для  $RS$ -тригера.

*Асинхронний  $RS$ -тригер* на елементах  $I-HI$  показаний на рис. 4.7.1. Тригер утворений із двох комбінаційних схем  $I-HI$ , з'єднаних таким чином, що виникають позитивні зворотні зв'язки, завдяки яким у стійкому стані вихідний транзистор однієї схеми  $АБО-HI$  закритий, а іншої - відкритий. Таблиця 4.7.1 визначає закон функціонування тригера. На рис. 4.7.2 приведена часова діаграма, яка ілюструє роботу асинхронного  $RS$ -тригера.

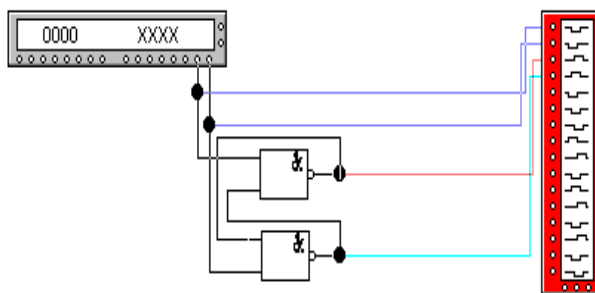


Рис. 4.7.1 - Схема асинхронного  $RS$ -тригера на елементах  $I-HI$

$R$	$S$	$Q$	Примітка
0	0	$Q$	Збереження
0	1	1	Установка 1
1	0	0	Установка 0
1	1	-	Заборонено

Таблиця 4.7.1 - Таблиця переходів асинхронного  $RS$ -тригера на елементах  $АБО-HI$

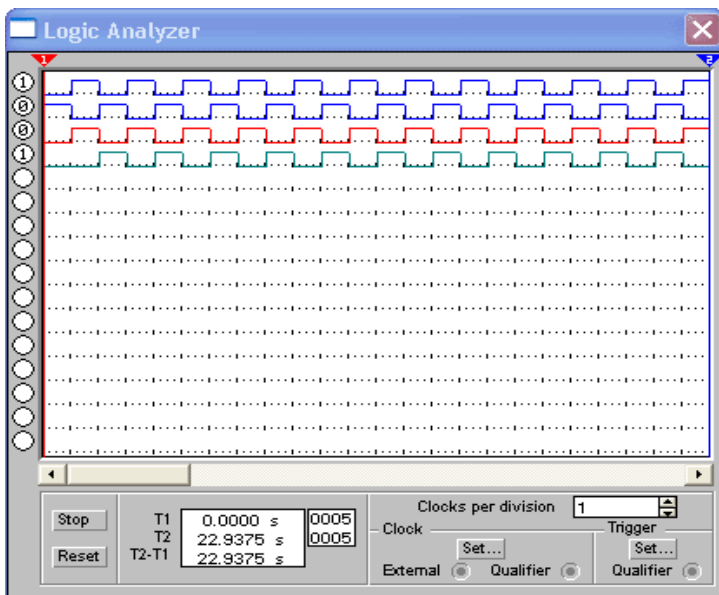


Рис. 4.7.2 - Діаграма роботи асинхронного RS-тригера

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 й інверсному виходах встановлюється нульовий сигнал. Тригерне кільце перетворюється в два незалежних інвертори і при переході до збереження ( $R=S=0$ ) тригер може встановлюватися в будь-який стан. Тому така комбінація вхідних сигналів заборонена.

*Синхронізуючий одноктактний RS-тригер* зображений на рис. 4.7.3. Такі RS-тригери мають два інформаційних входи  $R$  і  $S$  і вхід синхронізації  $C$ . Крім того, тригер може мати несинхронізуючі входи  $R$  і  $S$ . В цьому випадку функціонування тригера здійснюється або під впливом несинхронізуючих входів при  $C=0$ , або під впливом синхронізуючих входів. В останньому випадку на несинхронізуючих входах повинні бути присутні сигнали, які не впливають на стан схеми. На рис. 4.7.4 приведена часова діаграма, яка ілюструє роботу синхронного одноктактного RS-тригера.

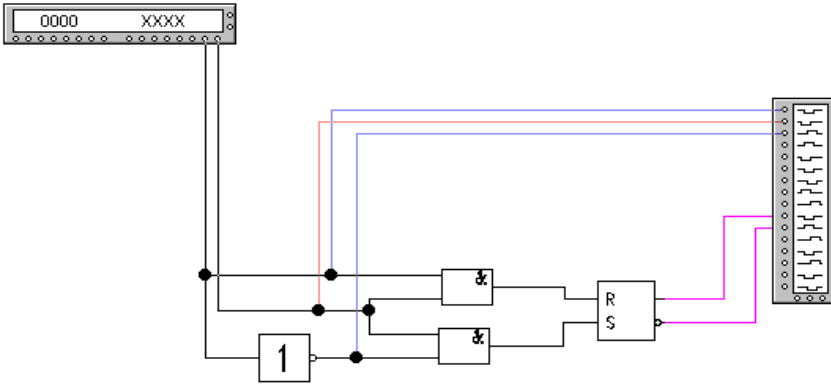


Рис. 4.7.3 - Схема синхронного RS-тригера

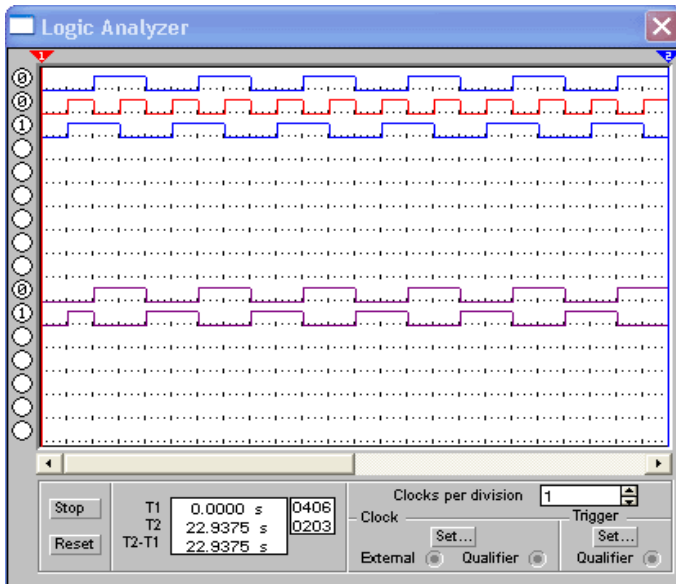


Рис. 4.7.4 - Діаграма роботи синхронного RS-тригера

Таблиця 4.7.2 визначає переходи RS-тригера для синхронізуючих входів  $R$  і  $S$ . Робота відповідно до даної таблиці здійснюється при сигналі несинхронізуючого входу  $R=1$  і при  $C=1$ .

<b><i>R</i></b>	<b><i>S</i></b>	<b><i>Q</i></b>	<i>Примітка</i>
1	1	$\bar{Q}$	Збереження
1	0	1	Установка 1
0	1	0	Установка 0
0	0	-	Заборонено

Таблиця 4.7.2 - Таблиця переходів для одноктного RS-тригера

Вхідна інформація заноситься в синхронізуючий одноктактний RS-тригер через елементи вхідної логіки 1 і 2 в момент надходження сигналу синхронізації *C*. Під час відсутності сигналу синхронізації тригер може бути встановлений в стан 0 шляхом подачі на несинхронізуючий вхід *R* сигналу  $R=0$ .

*Двоктактний RS-тригер.* Стабільна робота одноктактних RS-тригерів у схемі з передачею інформації між тригерами можлива тільки у випадку, якщо занесення в тригер інформації здійснюється після завершення передачі інформації про колишній його стан в інший тригер (див. рис. 4.7.5). Це досить просто забезпечується при використанні двох серій синхросигналів, які знаходяться в протифазі. Таблиця 4.7.3 задає закон функціонування такого двоктактного тригера. Цей тригер змінює свої стани тільки після закінчення дії сигналу синхронізації  $C=1$  (перехід в режим збереження інформації). Тому з двоктактних тригерів можна будувати довільні схеми, в тому числі подавати сигнали з виходу тригера на його вхід.

<b><i>R</i></b>	<b><i>S</i></b>	<b><i>Q</i></b>	<i>Примітка</i>
0	0	$\bar{Q}$	Збереження
1	0	0	Установка 0
0	1	1	Установка 1
1	1	-	Заборонено

Таблиця 4.7.3 - Таблиця переходів для двоктактного RS-тригера

Такий принцип обміну інформації реалізований у двоктактних RS-тригерах.



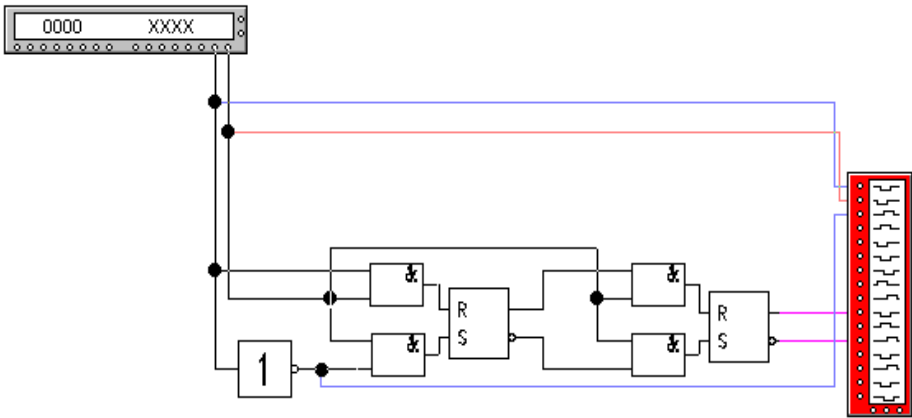


Рис. 4.7.5 - Схема двотактного RS-тригера

Найпростіша схема двохвідного двотактного RS-тригера показана на рис. 4.7.5, яка складається з двох одноктактних RS-тригерів і інвертора в колі синхронізації. При надходженні на вхід RS-тригера сигналу  $C=1$  вхідна інформація заноситься в перший одноктактний RS-тригер, а другий при цьому буде зберігати інформацію, яка відноситься до попереднього періоду представлення. По закінченні дії сигналу синхронізації, коли  $C=0$ , а  $C^{\wedge}=1$ , перший RS-тригер перейде в режим збереження, а другий прийме той же стан, що і перший. В результаті до наступного такту на вході двотактного RS-тригера з'явиться сигнал нового стану. На рис. 4.7.6 приведена часова діаграма, яка ілюструє роботу двотактного RS-тригера.

При проектуванні вищеприведених схем потрібно використати елемент *Панелі інструментів "Digital"* (рис. 4.7.7) і відповідні функціональні елементи: *Or, Nor, And, Nand*.

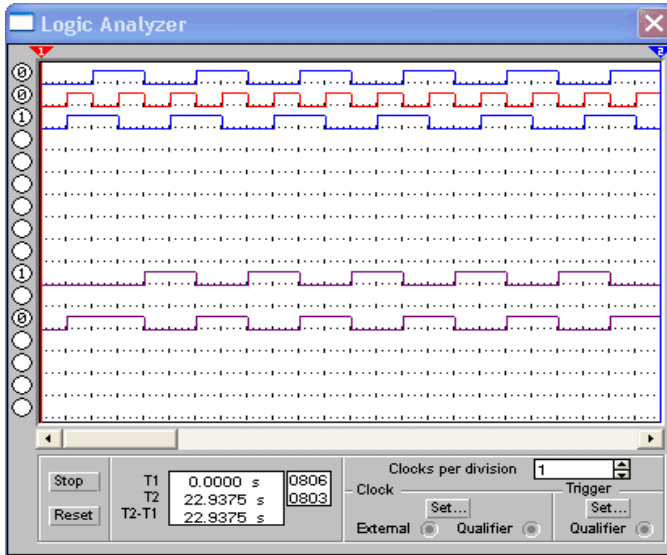


Рис. 4.7.6 - Діаграма роботи двотактного RS-тригера

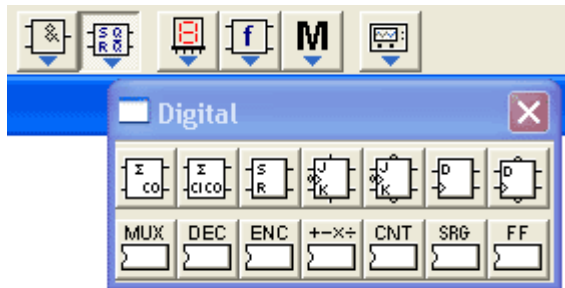


Рис. 4.7.7 - Панель інструментів "Digital"

Тригер типу *D* (від слова *delay* – затримка) приймає інформацію по одному входу і реалізує функцію часової затримки. *D*-тригер має тільки режими установки *1* і *0*. В зв'язку з цим несинхронізуючий *D*-тригер не застосовується, так як його вихід буде просто повторювати вхідний сигнал. Синхронізуючий однокітний *D*-тригер затримує розповсюдження вхідного сигналу на час паузи між синхросигналами (затримка на півперіод). Для затримки на період (на один такт) використовується двотактний *D*-тригер.

Inputs				Outputs		
$D$	$C$	$S$	$R$	$Q$	$Q^{\wedge}$	
X	X		1	0	0	1
X	X		0	1	1	0
X	X		0	0	1	0
X	0		1	1	$Q'$	$QB'$
X	1		1	1	$Q'$	$QB'$
0	RE		1	1	0	1
1	RE		1	1	1	0

Таблиця 4.7.4 - Таблиця істинності  $D$ -тригера

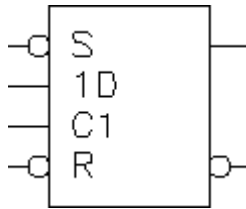


Рис. 4.7.8 - Умовне позначення  $D$ -тригера

Параметри синхронного  $D$ -тригера:

$D$  - інформаційний вхід тригера

$C$  - синхронізуючий вхід

$S$  - установка 1

$R$  – установка 0

$Q$  - інформаційний вихід

$Q^{\wedge}$  - інверсний інформаційний вихід

Параметри  $S$  і  $R$  служать для завдання початкових умов.

Тригер типу  $JK$  - універсальний, з роздільною установкою нульового й одиничного стану, і залежно від з'єднання його входів він може працювати як  $RS$ ,  $T$ ,  $D$  тригер. На відміну від тригера типу  $RS$  в ньому не заборонена одночасна подача сигналів на обидва входи. Входи  $J$  і  $K$  еквівалентні входам  $S$  і  $R$  установки тригера відповідно в стани "1" і "0". При об'єднанні входів  $J$  та  $K$  і при подачі на них імпульсів відрахунку вхід  $J$  при роздільному використанні входів відіграє роль входу установки в "одиницю", а вхід  $K$  - роль входу установки в "нуль".

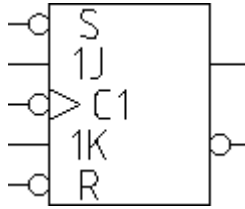


Рис. 4.7.9 - Умовне позначення JK-тригера

Параметри синхронного JK-тригера:

- C** - синхронізуючий вхід
- J і K** - інформаційні входи тригера
- S** - установка 1
- R** - установка 0
- Q** - інформаційний вихід
- Q<sup>^</sup>** - інверсний інформаційний вихід

Параметри **S** і **R** служать для завдання початкових умов (таблиця 4.7.5)

Inputs				Outputs		
J	K	C	S	R	Q	Q <sup>^</sup>
X	X	X	1	0	0	1
X	X	X	0	1	1	0
X	X	X	0	0	1	0
X	X	0	1	1	Q'	QB'
X	X	1	1	1	Q'	QB'
0	0	FE	1	1	Q'	QB'
0	1	FE	1	1	0	1
1	0	FE	1	1	1	0
1	1	FE	1	1	QB'	Q'

Таблиця 4.7.5 - Таблиця істинності синхронного JK-тригера

Тригер типу *T* називається тригером з лічильним входом (або лічильним тригером). Він змінює свій стан на протилежний кожен раз, коли на його вхід приходиться черговий сигнал. Позначення тригера походить від першої букви англійського слова *toggle* - *засувка*.

Так як в *NI Multisim* не приводиться *T*-тригер, то його можна одержати, об'єднавши інформаційні входи *JK*-тригера.

**Завдання:**

1. Використовуючи пакет *NI Multisim*, спроектувати схеми *RS*, *D*, *JK*, *T* тригерів на основі логічних елементів, застосовуючи для складання схеми таблицю істинності і проаналізувати роботу тригерів.

2. Скласти звіт про виконану лабораторну роботу, в якій відобразити:

- схеми тригерів;
- часові діаграми роботи тригерів.

**Завдання відповідно до варіантів:**

1. Дослідити схему двотактного *RS*-тригера на базі елементів *АБО-НІ*.

2. Дослідити схему двотактного *RS*-тригера з використанням мікросхеми, приведеної в *NI Multisim*.

3. Дослідити схему *D*-тригера з використанням мікросхеми, приведеної в *NI Multisim*.

4. Дослідити схему *JK*-тригера з використанням мікросхеми, приведеної в *NI Multisim*.

5. Дослідити схему *T*-тригера з використанням мікросхеми, приведеної в *NI Multisim*.

**Примітка:** Для побудови й аналізу роботи *T*, *D* і *JK* тригерів використати їхні умовні позначення.

*Регістри пам'яті і реєстри зміщення*

**Регістром** називається пристрій, який виконує прийом, збереження, перетворення і видачу чисел у двійковому коді. Інформація в реєстрі зберігається в вигляді числа. Він містить в собі окремі тригери, кількість яких відповідає числу розрядів двійкового коду, і логічні елементи.

Регістри виконують ряд мікрооперацій над словами:

1. Прийом слова в реєстр у прямому і зворотному коді; дані зберігаються в реєстрі доти, поки не з'явиться команда на їхню зміну.

2. Видача слова з реєстра в прямому і зворотному коді.

3. Виконання порозрядних логічних операцій над декількома словами.

4. Зміщення коду вправо або вліво на необхідне число розрядів, перетворення паралельного коду в послідовний і навпаки.

По способу запису і зчитування коду числа в регістрі розрізняють *паралельні* і *послідовні* регістри.

*Паралельний регістр.* В паралельних регістрах операції запису і зчитування інформації здійснюються у всіх розрядах одночасно. Схема трьохрозрядного регістра на *D*-тригерах, побудованого в пакеті *NI Multisim*, приведена на рис. 4.8.1. Інформація надходить у вигляді паралельного коду, тобто всі розряди одночасно по  $n$  ( $n=3$ ) проводах. В такий спосіб інформація в паралельному регістрі зберігається в паралельному коді, тому паралельний регістр називають *регістром пам'яті*. Інформація, що зчитується з виходів тригерів, зображена на часовій діаграмі на рис. 4.8.2.

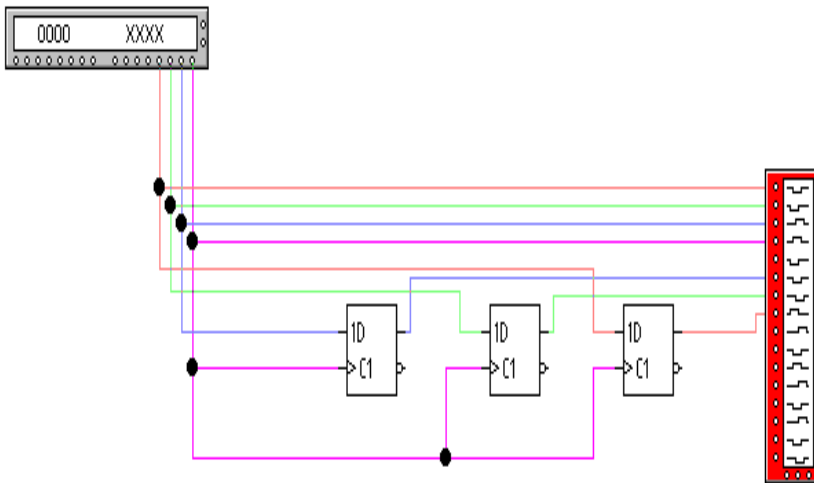


Рис. 4.8.1 - Схема паралельного регістра

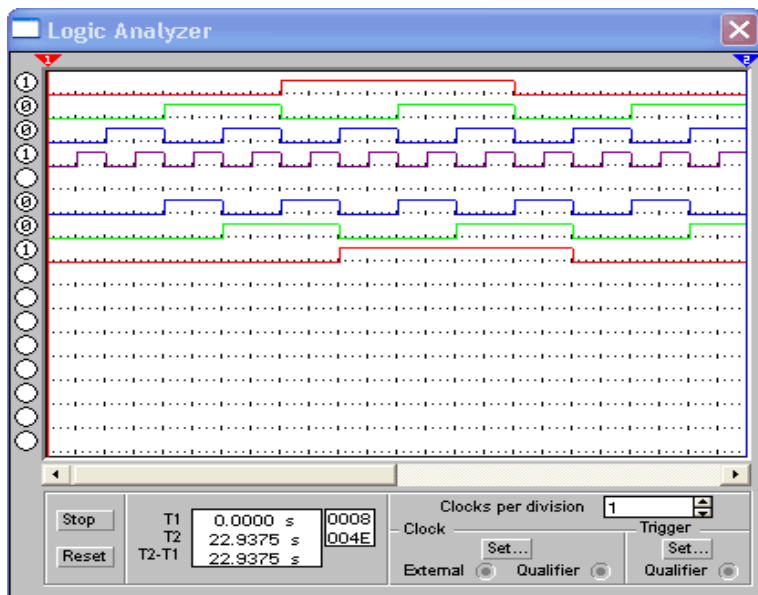


Рис. 4.8.2 - Діаграма роботи паралельного регістра

*Послідовний регістр.* У послідовних регістрах запис коду числа починається з першого розряду шляхом послідовного просування інформації за допомогою тактових імпульсів. Схема трьохрозрядного послідовного регістра, виконаного на *D*-тригерах, приведена на рис. 4.8.3. Часова діаграма, що ілюструє роботу регістра, показана на рис. 4.8.4. Записане число надходить по одному входу у вигляді послідовного коду, тобто значення розрядів передаються аналогічно тому як ми читаємо багаторозрядне число, наприклад: “тисяча чотириста тридцять сім” - 1437. В загальному вигляді *n*-розрядний регістр запам'ятовує *n*-розрядне число за *n*-тактових імпульсів. Послідовний код, що надійшов на вхід, перетвориться в регістрі в паралельний код: число може бути зчитане з виходів тригерів. З надходженням кожного тактового імпульсу записана інформація зміщується в регістрі (рух від входу до виходу), тому послідовний регістр називають *регістром зміщення*.

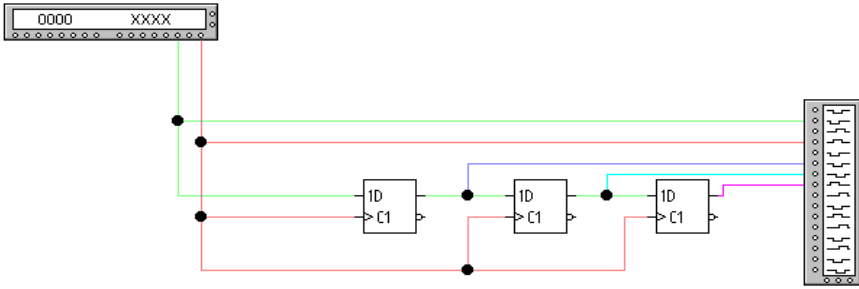
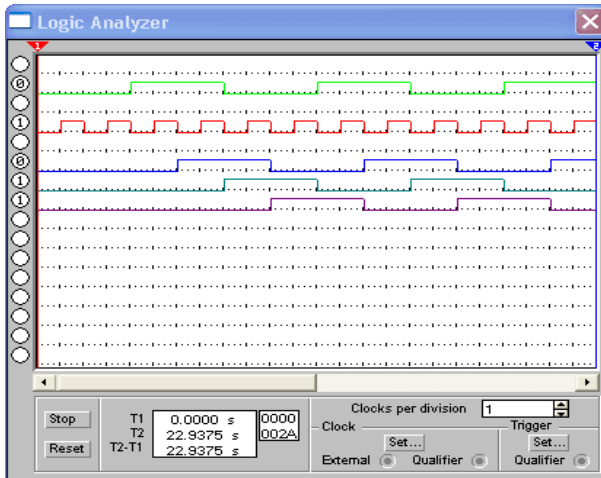


Рис. 4.8.3 - Схема послідовного регістра

Зміщення інформації на один розряд рівнозначне множенню коду на 2. Наприклад, записано число 101 (у десятковому кодi 5), зміщуємо його на один розряд вліво й одержуємо 1010 (десятковому кодi 10). Інформація, записана в послідовному регістрі, може бути зчитана з виходу його старшого розряду у вигляді послідовного коду: якщо після запису в регістр числа знову подати тактові імпульси, число порозрядно буде прочитуватися на виході старшого розряду і звідти може бути передане до інших зчитуючих кіл. Для збереження й обробки інформації в мікро-ЕОМ широко використовуються регістри зміщення. Регістри зміщення, як правило, реалізуються на середніх інтегральних схемах, виконаних із застосуванням *RS*-, *JK*- або *D*-тригерів, і різниця між ними тільки в методі обробки вхідних і вихідних даних.





**Завдання:**

1. Використовуючи пакет *NI Multisim*, спроектувати схеми на основі елементів і проаналізувати роботу регістрів.
2. Скласти звіт про виконану лабораторну роботу, в якій відобразити:
  - схеми регістрів;
  - часові діаграми роботи регістрів.

**Завдання відповідно до варіантів:**

1. Побудувати 4-х розрядний послідовний регістр.
2. Побудувати 4-х розрядний паралельний регістр.

*Проектування двійкових лічильників*

Майже кожна складна цифрова система містить кілька лічильників. **Лічильник** – це функціональний вузол, який призначений для підрахунку числа входних сигналів і запам'ятовування коду цього числа відповідними тригерами. Результат підрахунку в них записується в двійковому коді. Максимальне число  $N$ , що може бути записане в лічильнику, рівне  $2^n - 1$ , де  $n$ -число розрядів лічильника. Кожен розряд лічильника містить у собі тригер. По призначенню лічильники поділяються на *сумуючі* і *віднімаючі*.

Розглянемо будову і часову діаграму роботи сумуючого лічильника (трохрозрядного).

*Сумуючий лічильник* працює за принципом сумування сигналів, що надходять на його вхід (таблиця 4.9.1). На рис. 4.9.1 приведена функціональна схема трохрозрядного сумуючого лічильника і часова діаграма роботи (рис. 4.9.2), в таблиці 4.9.1 – стани його тригерів ( $T_2$ ).

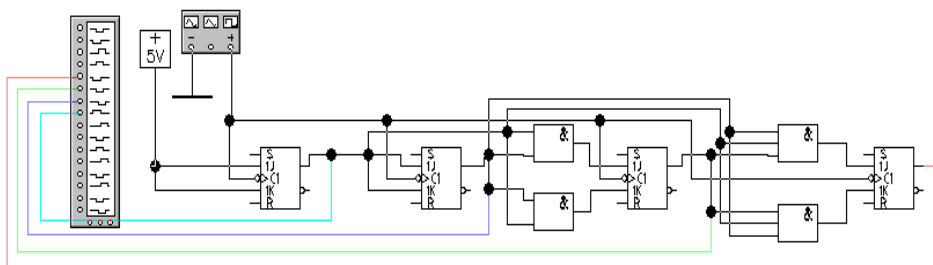


Рис. 4.9.1 - Схема сумуючого двійкового лічильника

Номер імпульсу	Стан тригерів		
	$Q1$	$Q2$	$Q3$
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1
8	0	0	0

Таблиця 4.9.1 - Таблиця істинності сумуючого двійкового лічильника

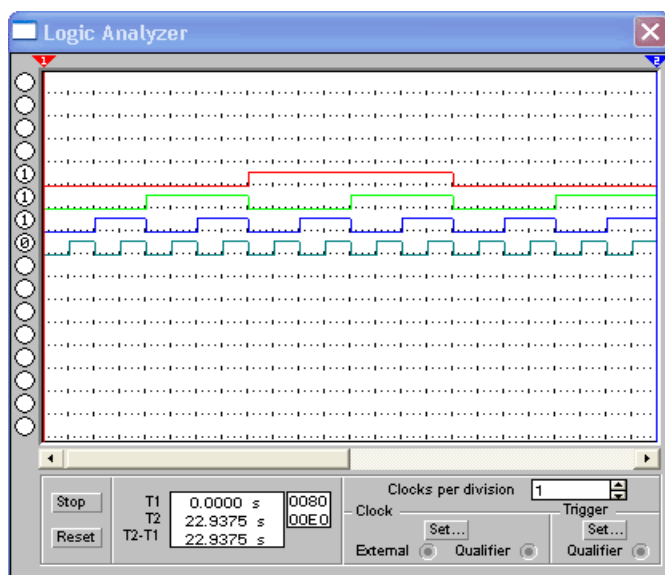


Рис. 4.9.2 - Діаграма роботи сумуючого двійкового лічильника

В початковий момент часу всі тригери встановлюються сигналом *Уст 0* в стан “0”. Після приходу першого рахункового імпульсу тригер  $T_{21}$  перейде в стан “1” і в лічильнику зафіксується код 001.

Другий імпульс, що прийшов на вхід, переведе  $T21$  знову в стан “0”. При цьому виникає імпульс переносу, що встановлює наступний тригер  $T22$  в стан “1” і в лічильнику зафіксується код 010. Після третього вхідного сигналу  $T21$  знову перейде в стан “1”, а інші тригери залишаться в колишньому стані. Так буде продовжуватися доти, поки лічильник не просумує максимальне для трьох розрядів число  $7_{10}=111_2$ . Восьмий імпульс переведе  $T21$  в стан “0”, а перенос, що виник, надійде на  $T22$  і також переведе його в стан “0”. В свою чергу, імпульс переносу з другого розряду переведе в стан “0” і  $T23$ . В результаті цей лічильник перейде у вихідний нульовий стан (000).

У віднімаючому лічильнику перенос від розряду береться не з одиничних, а з нульових виходів тригерів. Можна переконатися в тому, що при такій комутації перенос утвориться при переході відповідного тригера в стан “1”, а не “0”, як це було в сумуючому лічильнику (таблиця 4.9.2).

Номер імпульсу	Стан тригерів		
	$Q1$	$Q2$	$Q3$
0	1	1	1
1	1	1	0
2	1	0	1
3	1	0	0
4	0	1	1
5	0	1	0
6	0	0	1
7	0	0	0
8	1	1	1

Таблиця 4.9.2 - Таблиця істинності двійкового віднімаючого лічильника

У віднімаючому лічильнику кожен сигнал, що надходить на вхід, не збільшує, а зменшує вміст лічильника на одиницю (рис.

4.9.3). Часова діаграма роботи віднімаючого лічильника приведена на рис. 4.9.4.

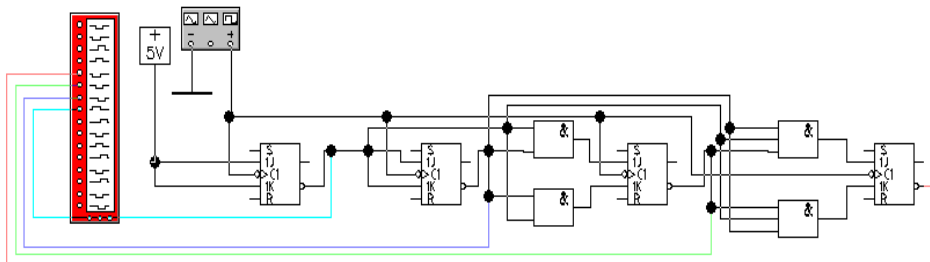


Рис. 4.9.3 - Схема двійкового віднімаючого лічильника

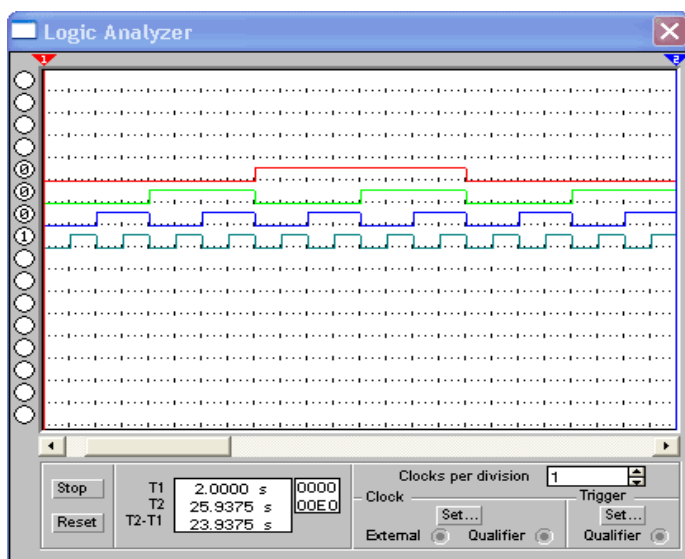


Рис. 4.9.4 - Діаграма роботи двійкового віднімаючого лічильника

### Завдання:

1. Використовуючи пакет *NI Multisim*, спроектувати схеми лічильників, використовуючи для складання схеми таблицю істинності і проаналізувати роботу:

- сумуючого лічильника;
- віднімаючого лічильника.

2. Скласти звіт про виконану лабораторну роботу, в якій відобразити:

- схеми лічильників;
- часові діаграми роботи лічильників.

Завдання відповідно до варіантів:

1. Спроекувати 8-розрядний лічильник на основі RS тригера.
2. Спроекувати 8-розрядний лічильник на основі JK тригера.

*4.10. Проектування ЦАП і АЦП*

1. *Цифро-аналогові перетворювачі (ЦАП)* служать для перетворення інформації з цифрової форми в аналоговий сигнал – сумування струмів і напруг. ЦАП широко застосовується в різних пристроях автоматики для зв'язку цифрових ЕОМ з аналоговими елементами і системами.

Принцип роботи ЦАП полягає в сумуванні аналогових сигналів, пропорційних розрядності вхідного цифрового коду з коефіцієнтами, рівними нулю або одиниці в залежності від значення відповідного розряду коду.

ЦАП перетворює цифровий двійковий код  $Q_4Q_3Q_2Q_1$  в аналогову величину, як правило, в напругу  $U_{вих}$ . Кожен розряд двійкового коду має визначену вагу  $i$ -го розряду - вдвічі більшу, ніж вага  $(i-1)$ -го. Роботу ЦАП можна описати наступною формулою:

$$U_{вих} = e*(Q_1 1 + Q_2 * 2 + Q_3 * 4 + Q_4 * 8 + \dots), \quad (1)$$

де  $e$  - напруга, що відповідає вазі молодшого розряду,  
 $Q_i$  - значення  $i$ -го розряду двійкового коду (0 або 1).

Наприклад, числу 1001 відповідає

$$U_{вих} = e*(1*1 + 0*2 + 0*4 + 1*8) = 9*e, \quad \text{а числу 1100}$$
$$U_{вих} = e*(0*1 + 0*2 + 1*4 + 1*8) = 12*e.$$

На рис. 4.10.1 приведена схема цифро-аналогового перетворювача.

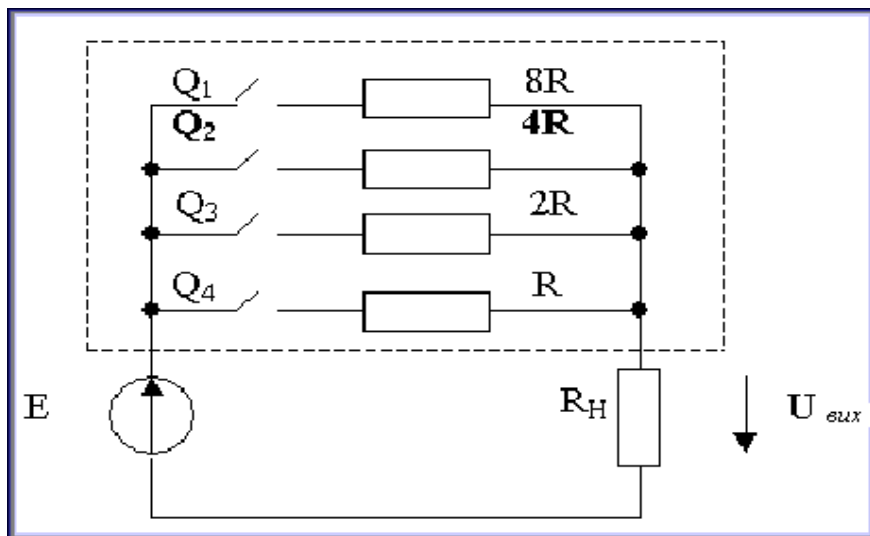


Рис. 4.10.1 - Схема цифро-аналогового перетворювача

На рис. 4.10.1 представлена спрощена схема реалізації ЦАП. В цій схемі  $i$ -й ключ замкнутий при  $Q_i=1$ , при  $Q_i=0$  – розімкнутий. Регістри підбрані таким чином, що  $R \gg R_n$ . Еквівалентний опір обведеного пунктиром двохполюсника  $R_{ек}$  і опір навантаження  $R_n$  утворюють подільник напруги, тоді

$$U_{вих} = E \cdot R_n / R_{ек} + R_n \gg E \cdot R_n / R_{ек} \quad (2)$$

Провідність двохполюсника  $1/R_{ек}$  рівна сумі провідностей частин кіл (при  $Q_i=1$   $i$ -а частина кола включена, при  $Q_i=0$  – відключена):

$$1/R_{ек} = Q_1/8R + Q_2/4R + Q_3/2R + Q_4/R \quad (3)$$

Підставивши (3) в (2), отримаємо вираз, ідентичний (1)

$$U_{вих} = (8 \cdot E \cdot R_n / R) \cdot (Q_1 \cdot 1 + Q_2 \cdot 2 + Q_3 \cdot 4 + Q_4 \cdot 8)$$

Очевидно, що  $e = 8 \cdot E \cdot R_n / R$ . Вибором  $e$  можна встановити необхідний масштаб аналогової величини.

2. Аналогово-цифрові перетворювачі (АЦП). В інформаційних і керуючих системах частина (або вся) інформації від давачів буває

представлена в аналоговій формі. Для її передачі в цифровій *ЕОМ* і цифровий керуючий пристрій широко застосовуються *АЦП*. В більшості випадків *АЦП* виконують перетворення вхідної напруги або струму в двійковий цифровий код. Існують різні типи *АЦП*. Розглянемо тільки ті, які одержали в даний час найбільше поширення.

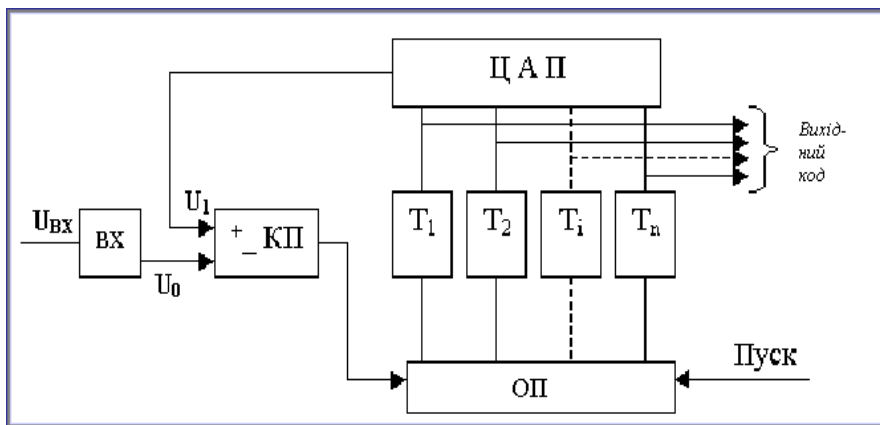


Рис. 4.10.2 - Схема АЦП послідовного наближення

2.1. *АЦП послідовного наближення (АЦППН)*. Структурна схема *АЦППН* приведена на рис. 4.10.2. Схема працює наступним чином. Вхідний аналоговий сигнал  $U_{вх}$  перед початком перетворення запам'ятовується схемою вибірки-збереження *ВХ*, що необхідно, так як в процесі перетворення необхідна зміна аналогового сигналу. Далі по команді “Пуск” за допомогою регістра зміщення послідовно в часі кожен тригер  $T_i$ , починаючи зі старшого розряду, переводить в положення 1 відповідний розряд *ЦАП*. Напруга  $U_1$  (або струм) з виходу *ЦАП* порівнюється з вхідним аналоговим сигналом за допомогою компаратора *КП*. Якщо  $U_0 > U_1$ , то на виході компаратора зберігається низький рівень і в тригері зберігається одиниця, при  $U_0 < U_1$  спрацьовує компаратор і переводить тригер в положення 0. Після закінчення циклу на виходах тригерів з'являється двійковий код, що відповідає (при ідеальних елементах)  $U_0$  з точністю до половини молодшого розряду.

Похибка *АЦППН* визначається неточністю *ЦАП*, зоною нечутливості і зсувом нуля компаратора, а також похибкою схеми вибірки-збереження.

Оскільки в такій схемі помилка в будь-якому розряді надалі не коректується, необхідно щоб час на “зважавання” кожного розряду був достатнім для загасання перехідного процесу до рівня, що відповідає половині молодшого розряду, і щоб при розбалансі  $U1-U0$  на це значення компаратор встиг спрацювати. Загальний час перетворення

$$t_{np} = t_{ex} + n \cdot (t_{z,k} + t_e + t_u) + t_{ск}$$

де  $t_{ex}$  – час, необхідний для фіксації  $U_{ex}$  схемою  $BX$ ;

$n$  – число розрядів;

$t_{z,k}$  – час затримки, внесений компаратором;

$t_e$  – час встановлення  $U1$  на вході  $ЦАП$ ;

$t_u$  – час затримки цифрових елементів в схемі керування і спрацьовування тригера;

$t_{ск}$  – час, необхідний для скидання  $ЦАП$  у вихідний стан, включаючи час, необхідний для синхронізації з початком такту.

Найбільшу частку в  $t_{np}$ , як правило, вносить  $t_e$ , найбільша величина якого може бути оцінена наступним чином:

$$t_e = (1 + n) \cdot T_e \cdot \ln 2,$$

де  $T_e$  – еквівалентна постійна часу на вході  $ЦАП$ . Якщо на його виході включений  $ОП$  (операційний підсилювач), що служить для зменшення вихідного опору і прискорення тим самим перехідного процесу, то  $T_e \gg 1/2p \cdot f_{зр}$  ( $f_{зр}$  – частота зрізу  $ОП$  по контуру зворотнього зв'язку).

При 12-розрядному  $АЦП$  і використанні швидкодіючого  $ЦАП$  з  $t_e = 100$  нс час  $t_{np}$  близький до 1.5 мкс. В більшості випадків  $t_{np}$  такого перетворювача досягає 10-100 мкс.

2.2.  $АЦП$  паралельного типу ( $АЦПП$ ). Істотне зменшення  $t_{np}$  вдається одержати в  $АЦП$  паралельного типу. Його структурна схема приведена на рис. 4.10.3. Тут вхідна аналогова величина  $U_0$  з виходу схеми  $BX$  порівнюється за допомогою  $2^{n+1}-1$  компараторів з  $2 \cdot (2^n - 1)$  еталонними рівнями, утвореними подільниками з резисторів рівного опору. При цьому спрацьовують  $m$  молодших компараторів, які утворюють на виходах схем  $I-III$  нормальний одиничний код, який потім за допомогою спеціального дешифратора  $ДШ$  перетворюється в дійковий вихідний сигнал.



Похибка АЦПП визначається неточністю і нестабільністю еталонної напруги, резистивного подільника і похибками компараторів. Значну роль можуть грати вхідні струми компараторів, якщо подільник недостатньо низькоомний.

Час перетворення складається з наступних складових:

$$t_{np} = t_{вх} + t_{з,к} + k \cdot t_{лс},$$

де  $t_{лс}$  – час затримки логічних схем;

$k$  – число послідовно включених логічних схем.

При використанні компаратрів зі стробуванням АЦПП може бути без схеми ВХ. При цьому він забезпечує найбільшу швидкість у порівнянні з будь-якими іншими АЦП.

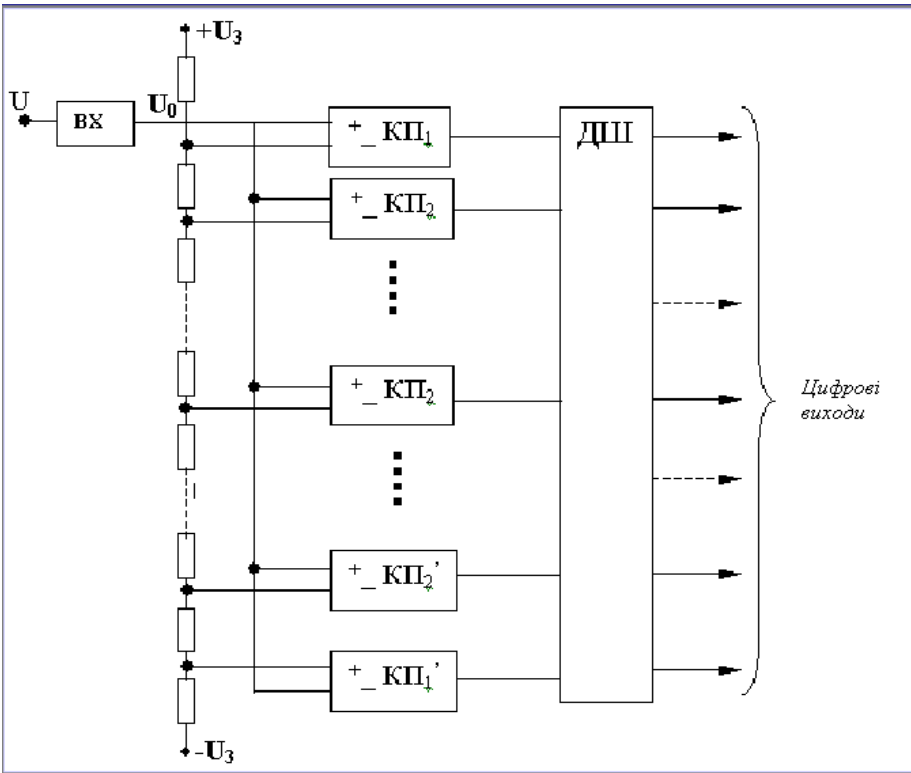


Рис. 4.10.3 - Схема АЦП паралельного типу

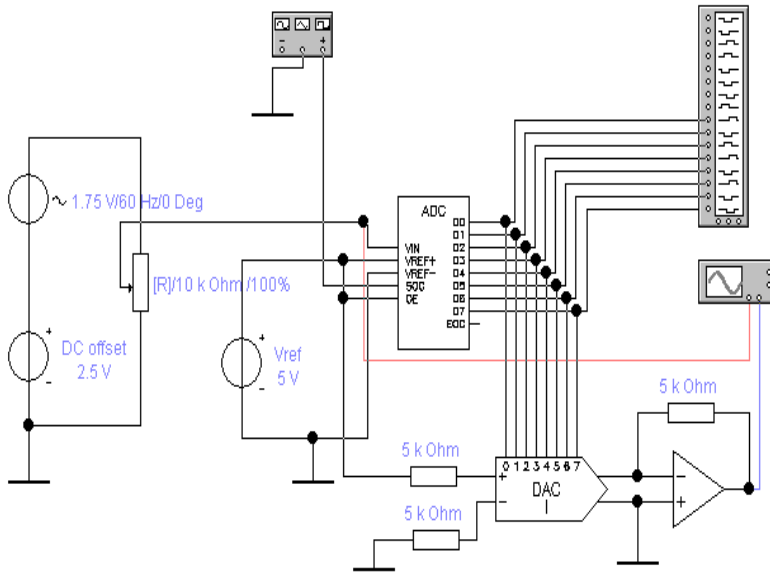
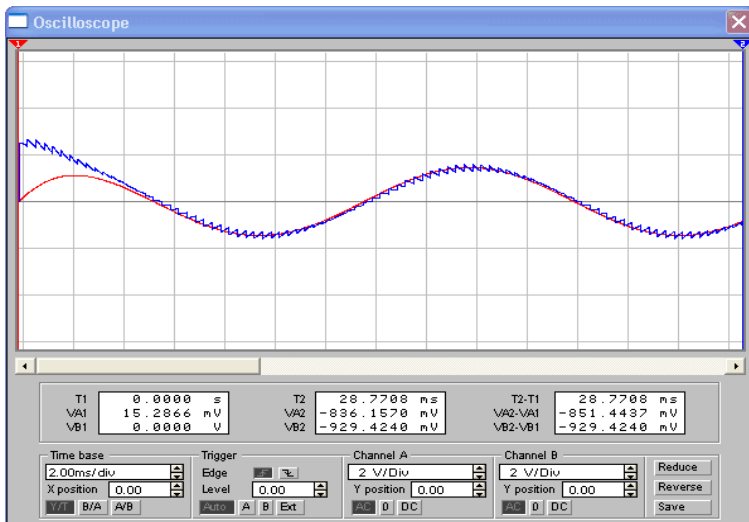


Рис. 4.10.4 - АЦП і ЦАП

На рис. 4.10.4 показана схема для перетворення аналогової величини (напруги) в цифровий код і зворотнє перетворення цифрового коду в аналогову величину. Процес роботи такої схеми показаний на часовій діаграмі (рис. 4.10.5).



**Завдання:**

1. Використовуючи пакет *NI Multisim*, спроектувати схему, представлену на рис. 4.10.4, і провести аналіз зміни напруги, змінюючи частоту й амплітуду, змінити час замикання і розмикання кожного з ключів.
2. Скласти звіт про виконану лабораторну роботу, в якій відобразити:
  - схеми ЦАП і АЦП;
  - часові діаграми роботи ЦАП і АЦП.

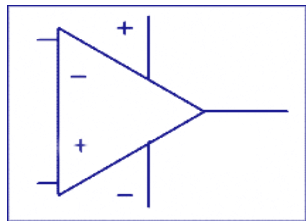
Аналіз аналогових схем

4.11. Проектування операційних підсилювачів

Операційний підсилювач (ОП) - найбільш розповсюджений вид підсилювальних інтегральних мікросхем (ІМС). В ньому зосереджені всі основні переваги підсилювальних схем. Ідеальний операційний підсилювач має надзвичайно великий коефіцієнт підсилення по напрузі,

$$K_U = U_{вих} / U_{вх} \rightarrow \infty$$

На схемах операційні підсилювачі мають відповідне позначення:



Незважаючи на те, що ОП поєднує в собі кращі властивості підсилювальних пристроїв, безпосередньо як підсилювач він не застосовується. Тут є декілька причин, і однією з них є те, що коефіцієнт підсилення змінюється від екземпляра до екземпляра в дуже широких межах. І дуже сильно залежать від режиму роботи, в першу чергу від температури, що обумовлено сильною залежністю від температури транзисторів, які входять до складу ІМС. Ця

нестабільність коефіцієнта підсилення затруднює створення підсилювальних пристроїв.

Для покращення параметрів застосовують операційні підсилювачі зі зворотнім зв'язком, тобто вихідний сигнал подається на один із входів *ОП*. З виходу *ОП* напругу зворотного зв'язку  $U_{oc}$  подають на той же вхід *ОП*. Таким чином, на входах *ОП* діє вхідна напруга  $U_{ex}$  і напруга  $U_{oc}$ .

Основними характеристиками операційних підсилювачів є коефіцієнт підсилення і частотна характеристика, тобто залежність коефіцієнта підсилення *ОП* від частоти поданого на вхід сигналу.

Коефіцієнт підсилення для неінвертованого операційного підсилювача зі зворотнім зв'язком:

$$K_{U_{oc}} = \frac{R_1 + R_2}{R_1}$$

На рис. 4.11.1 приведена схема неінвертованого *ОП* зі зворотнім зв'язком.

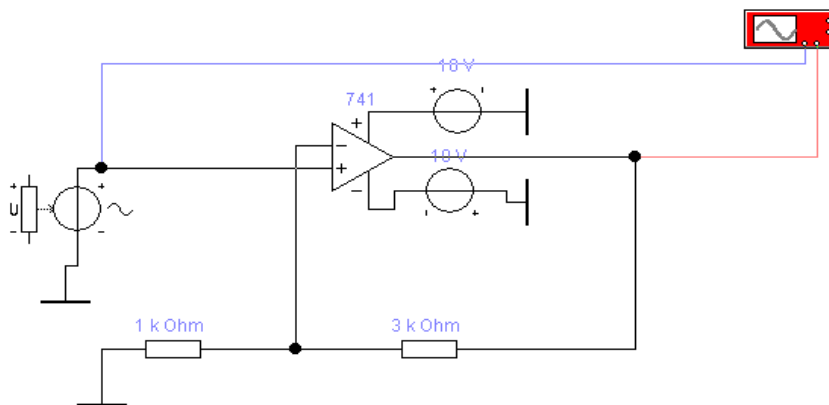


Рис. 4.11.1 - Схема неінвертованого *ОП*

Приклади графіків залежності напруги від часу вхідного і вихідного сигналів для зразкового підсилювача приведені на рис. 4.11.2.

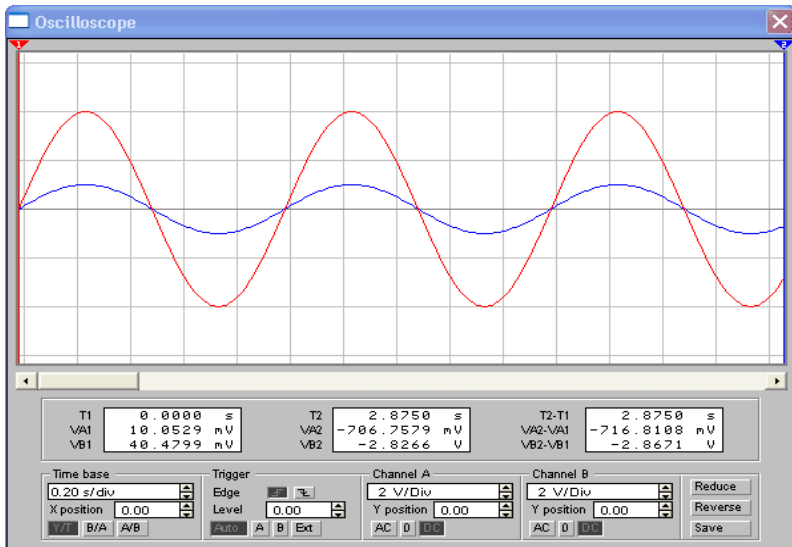


Рис. 4.11.2 - Діаграма роботи неінвертованого ОП

Коефіцієнт підсилення для інвертованого ОП зі зворотнім зв'язком:

$$K_{U_{oc}} = \frac{R_{oc}}{R_n}$$

Схема інвертованого ОП зі зворотнім зв'язком приведена на рис. 4.11.3.

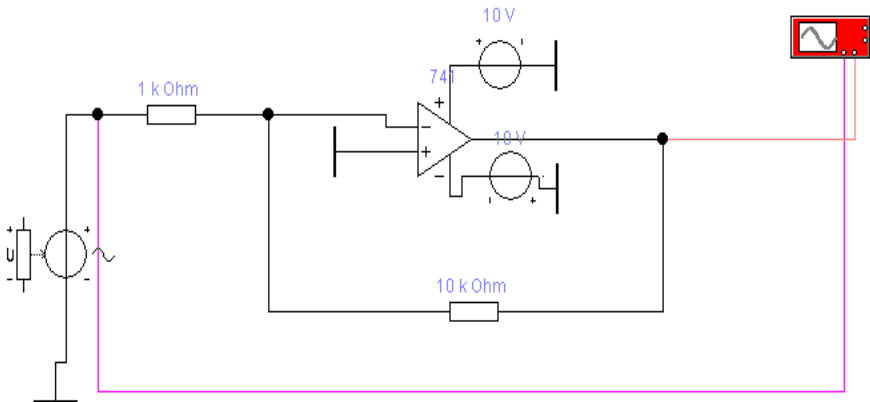


Рис. 4.11.3 - Схема інвертованого ОП

Приклади графіків залежності напруги від часу вхідного і вихідного сигналів для зразкового підсилювача приведені на рис. 4.11.4.

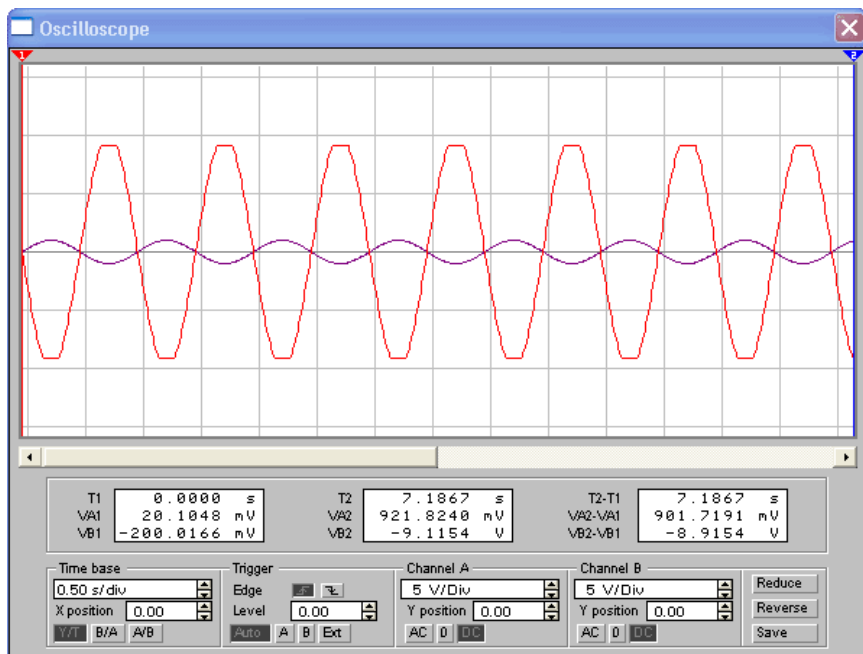


Рис. 4.11.4 - Діаграма роботи інвертованого ОП

### Завдання:

1. Використовуючи пакет *NI Multisim*, зібрати схему неінвертованого ОП. Дослідити схему ОП в режимі аналізу перехідних процесів. Змінюючи амплітуду коливаний джерела струму (параметр *A*), приблизно визначити, при якій амплітуді відбувається насичення (при постійному значенні коефіцієнта підсилення). Включити в звіт графіки перехідних процесів для трьох значень амплітуди:

- амплітуди, при якій ОП працює в нормальному режимі;
- амплітуди, при якій ОП має яскраво виражену затримку;
- граничне значення амплітуди, при якій ОП починає працювати з затримкою.

При постійному значенні амплітуди, змінюючи коефіцієнт підсилення, досягнути роботи підсилювача в:

- *нормальному режимі без насичення;*
- *в режимі з яскраво вираженим насиченням;*
- *в граничному режимі.*

2. Дослідити схему в режимі малосигнального аналізу в частотній області. Включити в звіт графіки залежності коефіцієнта підсилення і фази від частоти. Наближено визначити, при якій частоті ОП втрачає свої підсилювальні властивості, а також частоту, при якій коефіцієнт підсилення починає зменшуватися.

3. Дослідити схему в режимі аналізу по постійному струму. Включити в звіт отриманий графік.

4. Побудувати схему інвертованого ОП. Привести в звіті графік перехідних процесів.

#### 4.12. Проектування суматорів на операційних підсилювачах

На ОП виконують схеми, призначені для виконання математичних операцій над входними сигналами (*додавання, віднімання, інтегрування, виділення модуля функції і т.д.*). Найбільш розповсюдженими є сумуючі й інтегруючі схеми на ОП. Суматори можуть бути *інвертуючими, неінвертуючими або алгебраїчними (інакше віднімаючими)*. Схема інвертуючого суматора на підсилювачі приведена на рис. 4.12.1

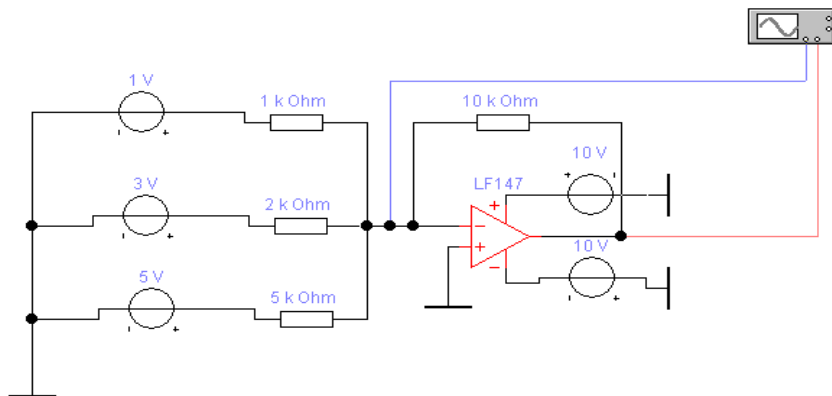


Рис. 4.12.1 - Схема суматора на ОП

В цій схемі знак мінус показує, що разом із сумуванням відбувається інвертування полярності сигналів. Часову діаграму сигналів на вході і виході суматора при використанні зразкового підсилювача приведено на рис. 4.12.2.

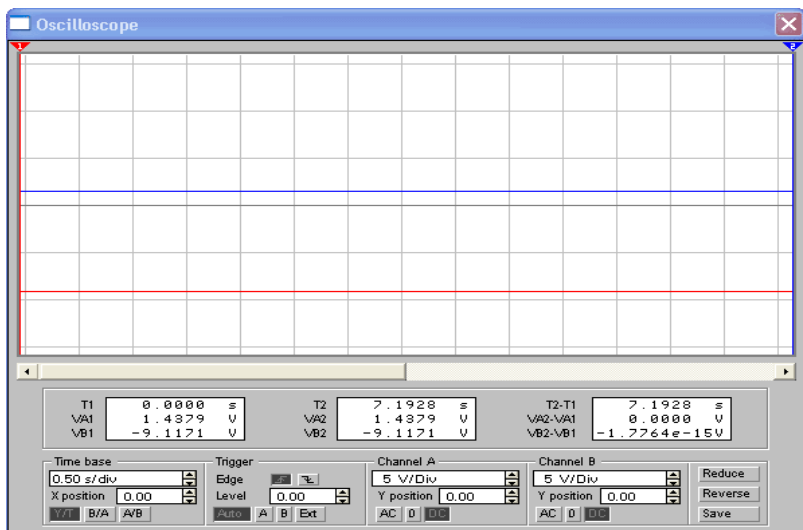


Рис. 4.12.2 - Діаграма роботи суматора на ОП

Схема неінвертуючого ОП відрізняється від схеми з інвертуючим тим, що сигнал подається на неінвертуючий вхід, і зворотний зв'язок здійснюється трохи іншим способом (див. попередню лабораторну роботу). Схема алгебраїчного суматора приведена на рис. 4.12.3.

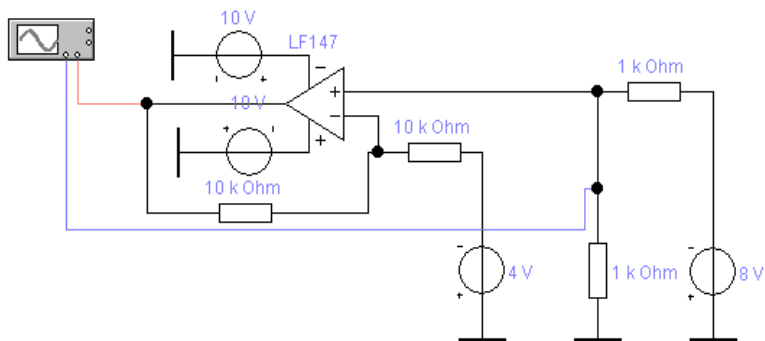


Рис. 4.12.3 - Схема алгебраїчного суматора



Розглянемо його роботу методом *суперпозиції*. Спочатку приймемо  $U_{\text{ex}2}=0$ , тобто закоротимо джерело  $U_{\text{ex}2}$ . Тоді схема зводиться до неінвертуючого *ОП*, на вході якого включений подільник напруги з коефіцієнтом передачі

$$U'_{\text{вих}} = -\left(U_{\text{ex}1} \frac{R_{oc}}{R_1} + U_{\text{ex}2} \frac{R_{oc}}{R_2} + U_{\text{ex}3} \frac{R_{oc}}{R_3}\right)$$

з врахуванням того що

$$\gamma_{\text{ex}} = R_4 / (R_3 + R_4)$$

маємо

$$K_{U_{oc}} = 1/\gamma = (R_1 + R_2)/R_1$$

Тепер приймемо  $U_{\text{ex}1}=0$  (закоротим джерело  $U_{\text{ex}1}$ ). Схема зводиться до інвертуючого *ОП*, оскільки підключення до прямого входу резисторів  $R_3$  і  $R_4$  не змінює потенціал на прямому вході ідеального *ОП*, в якого вхідний струм відносно малий. Тоді відповідно до

$$K_{U_{oc}} = U_{\text{вих}}/U_{\text{ex}} = -R_2/R_1$$

маємо

$$U''_{\text{вих}} = -U_{\text{ex}} \cdot R_2/R_1$$

В результаті впливу двох сигналів вихідна напруга *ОП* буде рівна

$$U_{\text{вих}} = U'_{\text{вих}} + U''_{\text{вих}} = U_{\text{ex}1} \cdot \left[ \frac{R_4}{(R_3 + R_4)} \right] \cdot \left[ \frac{(R_1 + R_2)}{R_1} \right] - U_{\text{ex}2} \cdot R_2/R_1$$

При  $R_1 = R_3, R_2 = R_4$  одержимо

$$U_{\text{вих}} = R_2/R_1 \cdot (U_{\text{ex}1} - U_{\text{ex}2})$$

### Завдання:

1. Використовуючи пакет *NI Multisim*, зібрати один із трьох суматорів, в схеми яких включити модель **LF147**:
  - суматор із синусоїдальними джерелами струмів;
  - суматор із джерелами постійного струму;
  - суматор з обома типами джерел струмів.
2. Дослідити схеми суматорів в режимі аналізу перехідних процесів. Включити в звіт графіки перехідних процесів для кожної схеми. Побудувати графік залежності результату роботи одного із суматорів від зміни коефіцієнта підсилення однієї з ланок схеми.
3. Побудувати один із трьох алгебраїчних суматорів на базі зразкового підсилювача:
  - суматор із синусоїдальними джерелами струмів;
  - суматор із джерелами постійного струму;
  - суматор з обома типами джерел струмів.
4. Дослідити схеми алгебраїчних суматорів в режимі аналізу перехідних процесів. Отримані графіки представити в звіті лабораторної роботи.

#### 4.13. Проектування інтеграторів на операційних підсилювачах

Інтегратори на ОП будуються на базі інвертуючих ОП. В коло зворотнього зв'язку включений конденсатор **C**. Як відомо з курсу **ТОЕ**

$$U_c = \frac{1}{C} \int i_c(t) dt \quad (1)$$

Схема інтегратора на ОП приведена на рис. 4.13.1.

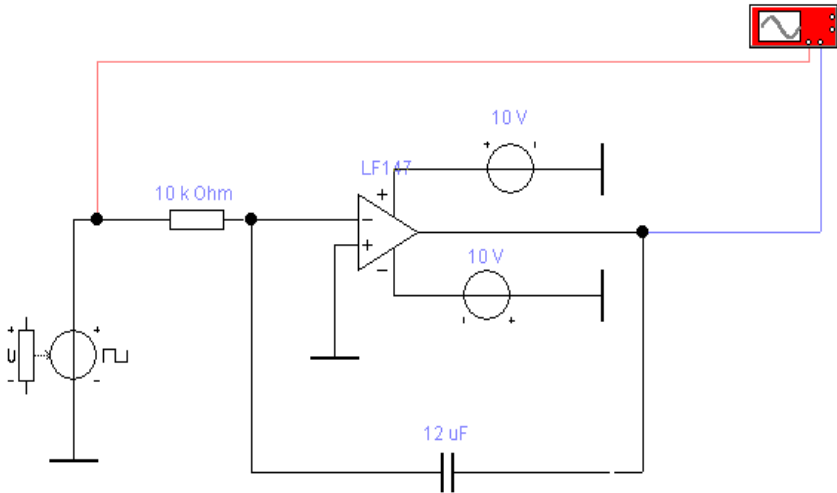


Рис. 4.13.1 - Схема інтегратора на ОП

Оскільки  $R_{ex} = U_{ex}/i_{ex}$ , то

$$i_C = -i_{ex} = -U_{ex} / R \quad (2)$$

Напруга між входами ІМС ОП рівна нулю, тому  $U_{ex} = U_C$ . З огляду на (1) і (2), одержуємо

$$U_{вих} = -\frac{1}{C} \int \frac{U_{ex}(t)}{R} dt = -\frac{1}{RC} \int U_{ex}(t) dt$$

Схема виконує математичну операцію інтегрування. Перейдемо від невизначених інтегралів до визначених і тоді отримаємо

$$U_{вих} = U_{вих}(0) - \frac{1}{RC} \int_0^t U_{ex}(t) dt$$

Вихідна напруга  $U_{вих}$  залежить від початкових умов, тобто від початкової напруги на конденсаторі в момент  $t = 0$   $U_{вих}(0)$ .

Часові діаграми, що ілюструють роботу конденсатора, приведені на рис. 4.13.2.

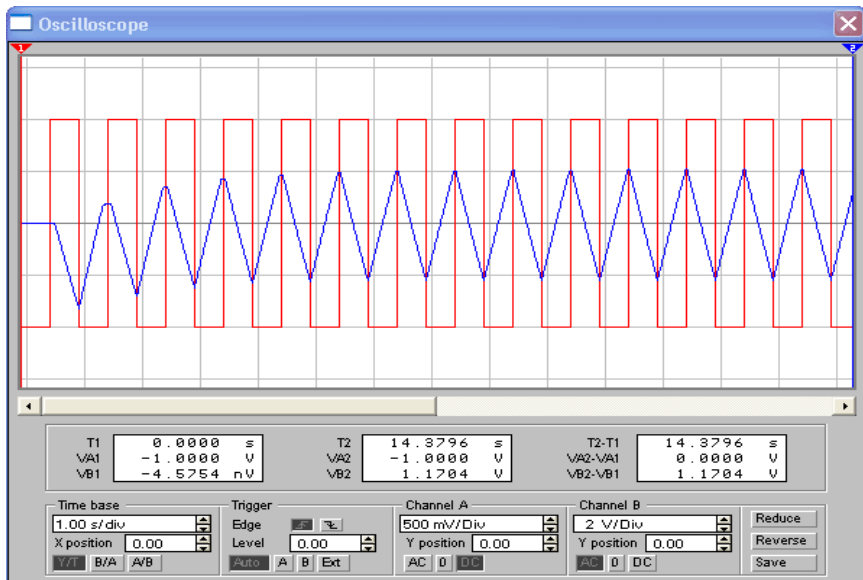


Рис. 4.13.2 - Діаграма роботи інтегратора на ОП

### Завдання:

1. Використовуючи пакет *NI Multisim*, зібрати інтегратор, включивши в нього модель LF147 (коли замінити параметр *Level* на *1*, то можна отримати зразкову модель *ОП*).
  2. Дослідити схему інтегратора в режимі аналізу перехідних процесів. Змінюючи характеристики імпульсного генератора при постійному значенні ємності досягнути роботи генератора:
    - в нормальному режимі (прямолінійна пилка);
    - в режимі експонентного перетворювача;
    - в режимі, коли конденсатор не встигає розрядитися і зарядитися цілком за час періоду (зменшення амплітуди) на порядок.
- Отримані графіки відобразити в звіті по лабораторній роботі.

## 4.14. Проектування мультівібраторів на операційних підсилювачах

Мультивібратором називається генератор періодично повторюваних імпульсів прямокутної форми. Мультивібратор є автогенератором і працює без подачі вхідного сигналу.

Конденсатор  $C$  і резистори  $R_1, R_2$  утворюють інтегруюче  $RC$ -коло: при заряді конденсатора відкритий діод  $V_1$ , струм проходить через  $R_1$ ; при розряді – відкритий діод  $V_2$ , струм йде через  $R_2$ . Джерелом напруги  $E$  є вхідне коло ОП. Компаратор виконаний на ОП з позитивним зворотнім зв'язком через коло  $R_3R_4$ . При переключенні компаратора на його виході відбувається комутація кіл заряду і розряду конденсатора  $C$ , тобто ОП виконує відразу кілька функцій: джерела напруг розряду і заряду конденсатора, компаратора і ключа.

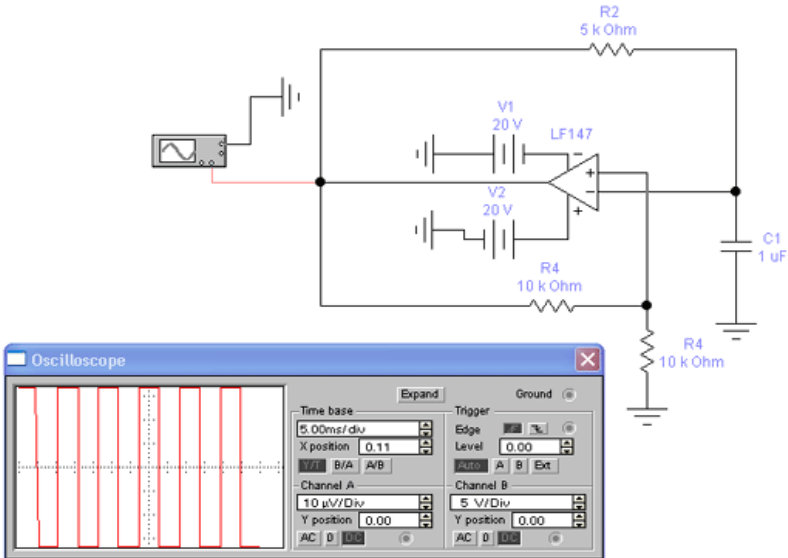


Рис. 4.14.1 - Схема симетричного мультивібратора на ОП і часова діаграма, яка ілюструє його роботу

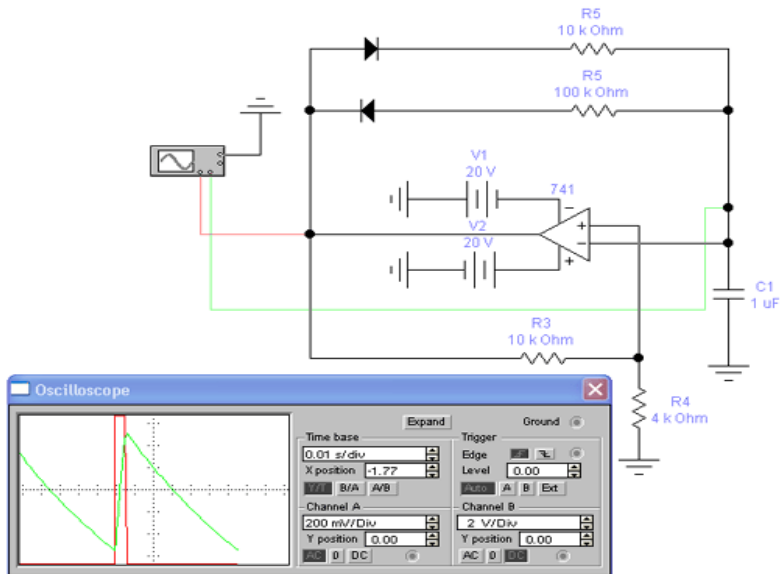


Рис. 4.14.2 - Схема мультивібратора на ОП і часова діаграма, яка ілюструє його роботу

### Завдання:

1. Використовуючи пакет *NI Worbench*, зібрати мультивібратор, включивши в нього модель 741 (або LF147 для симетричного мультивібратора).
2. Дослідити схему мультивібратора в режимі аналізу перехідних процесів.
3. Скласти звіт про виконану лабораторну роботу, в якій відобразити: схеми мультивібраторів і графіки залежності напруги від часу для входного і вихідного сигналу.
4. Змінюючи постійну часу кола заряду і розряду конденсатора, досягнути “скважності” імпульсів рівній 2 при різних частотах мультивібратора.

## *Використана література*

1. Файл допомоги *NI Multisim 5.12*
2. Internet: <http://Multisim.online.kg/>
3. Баскаков С.И. Радиотехнические цепи и сигналы. – М. Высш. шк., 1988.
4. Д. Жарников. Радиоловитель. Ваш компьютер, № 12, 1999.
5. В. Разевиг. Электронная лаборатория  
<http://softline.perm.ru/interactive/articles/art1.htm>
6. Цвеляя И. А. Использование системы компьютерного моделирования *NI Multisim* при изучении электротехнических дисциплин в неинженерных ВУЗах  
<http://bitpro.aha.ru/ITO/ITO98/2/TSVELAJA.html>