

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ ТА ТЕЛЕКОМУНІКАЦІЙ

УДК 621.3.037.372.3

«Допущено до захисту»

Завідувач кафедри біомедичної
інженерії та телекомунікацій
к.т.н., доц. Никітчук Т.М.

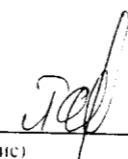
« 14 » грудня 2021 р.

КВАЛІФІКАЦІЙНА РОБОТА

на тему Дослідження можливостей побудови елементної бази трійкової логіки

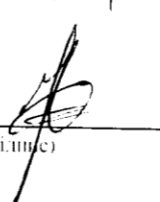
за спеціальністю 172 «Телекомунікації та радіотехніка»
освітній ступінь «магістр»

Виконавець роботи
студент 2 курсу, гр.ТРм-20-1

12.12.2021 
(дата, підпис)

Присяжнюк С.О.

Керівник роботи
к.т.н., доц.

10.12.2021 
(дата, підпис)

Коломієць Р.О.

Житомир
2021

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ «ЖИТОМИРСЬКА ПОЛІТЕХНІКА»
ФАКУЛЬТЕТ ІНФОРМАЦІЙНО-КОМП'ЮТЕРНИХ ТЕХНОЛОГІЙ
КАФЕДРА БІОМЕДИЧНОЇ ІНЖЕНЕРІЇ ТА ТЕЛЕКОМУНІКАЦІЙ

Пояснювальна записка

до кваліфікаційної магістерської роботи

на тему **Дослідження можливостей побудови елементної бази трійкової логіки**

Виконав: студент 2 курсу,
групи ТРМ-20-1 спеціальності
172 «Телекомунікації та радіотехніка»

Присяжнюк С. О.

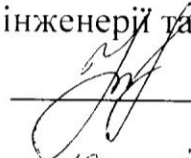
Керівник Коломієць Р.О

Рецензент Ципоренко В.В.

Житомир
2021

«Затверджено»

Завідувач кафедри біомедичної
інженерії та телекомунікацій

 Т.М. Нікітчук

« 18 » січня 2021 р.

Завдання на кваліфікаційну магістерську роботу

Присяжнюка Сергія Олександровича
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження можливостей побудови
елементної бази трійкової логіки

затверджено наказом по університету від « 08 » листопада 2021 р. № 546 /с

2. Мета дослідження: дослідити принципову можливість реалізації
цифрових схем з основою числення 3.

3. Вихідні дані (характеристики об'єкта, умов дослідження та ін.
Тип сигналу: дискретний, трох-рівневий. Елементна база: нів-
провідникова. Операції: інверсія, додавання по модулю 3.

4. Основні задачі дослідження 1) провести аналіз дискретної алгебри з
трьохма стійкими станами; 2) побудувати схеми, що реалізують цифрові
операції в цій алгебрі.

5. Термін подання роботи до захисту Грудень 2021 р.

6. Дата видачі завдання « 18 » січня 2021 р.

Науковий керівник

Завдання прийняв
до виконання


(прізвище)

(прізвище)

Коломієць Р.О.
(прізвище, ініціали)
Присяжнюк С.О.
(прізвище, ініціали)

Анотація

Дана кваліфікаційна робота присвячена дослідженню, розробці та моделюванню можливостей побудови елементної бази трійкової логіки. Основною областю використання буде створення на основі елементів трійкової логіки пристроїв де потрібна підвищена швидкодія.

В роботі було проведено огляд трійкової логіки наявних проектів з її використанням. Проведено розробку, моделювання та аналіз елементної бази трійкової логіки. Також проведена розробка арифметико-логічного пристрою на базі трійкових елементів.

Результатом цього дослідження є змодельовані принципові схеми елементів трійкової логіки, а також проведений аналіз кожної з них.

Abstract

This master's thesis is devoted to research, development and modeling of the possibilities of constructing the element base of ternary logic. The main area of use will be the creation of devices based on elements of ternary logic where increased speed is required.

The paper reviews the ternary logic of existing projects using it. The development, modeling and analysis of the element base of ternary logic is carried out. The development of an arithmetic-logic device based on ternary elements was also carried out.

The result of this study is a simulated schematic diagram of the elements of ternary logic, as well as an analysis of each of them.

Реферат

Кваліфікаційна магістерська робота присвячена дослідженню, розробці а також моделюванню елементної бази трійкової логіки. Основною областю використання буде створення на основі елементів трійкової логіки пристроїв де потрібна підвищена швидкодія .

Робота складається з вступу і трьох розділів, з висновками до кожного з них. Робота містить 22 таблиці ,78 рисунків. Повний обсяг роботи становить 119 сторінок.

У вступі обґрунтовано актуальність розробки та сформульовано мету, об'єкт та предмет розробки, а також поставлені завдання.

У першому розділі проведено огляд та аналіз трійкової логіки, обґрунтування розробки елементів трійкової логіки наявних проектів з її використанням, також проведено порівняльну характеристику з двійковою та основні переваги і недоліки в порівнянні з двійковою.

У другому розділі детально розглянуто особливості розробки елементів для трійкової логіки та основні проблеми такої реалізації.

У третьому розділі проведено розробку та дослідження елементів трійкової логіки також розроблено та досліджено арифметико-логічний пристрій на основі елементів трійкової логіки.

У висновках коротко описано досягнення при виконанні поставлених задач.

ЗМІСТ

ВСТУП	7
РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ	9
1.1 Основні визначення, що використовуються для даного дослідження	9
1.2 Обґрунтування необхідності розробки трійкових логічних елементів	11
1.3 Наявні методи реалізації та проекти з використанням трійкової логіки	14
1.4 Переваги та недоліки трійкової логіки в порівнянні з двійковою	18
1.5 Висновки до розділу 1	27
РОЗДІЛ 2 АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ЛОГІЧНИХ ЕЛЕМЕНТІВ ДЛЯ ТРІЙКОВОЇ ЛОГІКИ	28
2.1 Аналіз особливостей роботи наявних пристроїв трійкової логіки	28
2.2 Аналіз особливостей роботи трійкових логічних елементів	33
РОЗДІЛ 3 РОЗРОБКА ТРІЙКОВИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ	39
3.1 Вибір методів та компонентів для розробки трьохрівневої логіки	39
3.2 Розробка та моделювання роботи елементів трійкової логіки	50
3.3 Розробка арифметико-логічного пристрою на базі трійкових елементів	82
3.4 Висновки до розділу 3	115
ВИСНОВКИ	116
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ	119

ВСТУП

Незважаючи на традиційне використання в автоматичі та обчислювальній техніці елементів із двома стійкими станами, свого часу у світовій практиці виникла тенденція створення елементів із великою кількістю стійких станів. Наприклад, починаючи з середини двадцятого століття кількість робіт даного періоду, присвячених дослідженню багатостійких елементів і структур, подвоювалася кожні 4-5 років.

Один із найбільш важливих розділів більш загального завдання, пов'язаного з розробкою елементної бази цифрових автоматів з багатозначним структурним алфавітом – це принципи побудови багатозначних фізичних систем.

Розробка зазначеної елементної бази є складнішим завданням, ніж розробка елементної бази цифрових автоматів з двозначним структурним алфавітом. Причина цього насамперед у тому, що у випадку двозначного алфавіту практично зазвичай мають справу з порівняно невеликим числом повних систем логічних функцій. Вигляд функцій, а також особливості функціонування фізичних схем з двоцифровим структурним алфавітом зазвичай виявляються такими, що реалізація цих функцій на основі відомих схем, як правило, не викликає практичних труднощів. Причому через те, що кількість повних систем, що забезпечують можливість побудови відомих аналітичних уявлень, у двозначному випадку невелика, вибір повної системи, найбільш ефективно з точки зору заздалегідь сформульованих вимог, що реалізується в заданому класі фізичних схем може бути виконаний, наприклад, у результаті прямого перебору. Так, повний набір двох змінних складається з 16 логічних функцій для двозначних (бістабільних) систем. А для тризначних елементів (тристабільних) число можливих функцій двох змінних (тобто двовходових тристабільних елементів) становить уже 19683! (3 у ступені 9). Тому в багатозначному випадку при реалізації повних систем функцій у

заданому класі фізичних систем у зв'язку з вельми швидким зростанням числа функцій m -значної логіки зі збільшенням m практично повністю виключається можливість розв'язання задачі шляхом перебору систем, що включають різні функції багатозначної логіки. Крім того, зі збільшенням числа стійких станів складнішими виявляються і процеси, що протікають у таких схемах, що також ускладнює розробку багатозначних логічних елементів. внаслідок прямого перебору.

Актуальність теми дослідження полягає у розробці та дослідженні елементів трійкової логіки.

Мета і задачі дослідження. Метою дослідження є дослідження можливостей побудови елементної бази під роботу з трійковою симетричною системою числення.

Об'єкт дослідження: процес моделювання розроблюваної елементної бази трійкової логіки.

Предмет дослідження: електричні характеристики базових елементів трійкової логіки.

РОЗДІЛ 1

АНАЛІТИЧНИЙ ОГЛЯД ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1.1 Основні визначення, що використовуються для даного дослідження

При розробці та дослідженні в роботі буде використовуватись доволі багато визначень, які стосуються дуже вузького напрямку в електроніці. Тому в цьому підрозділі ми розглянемо основні визначення, що стосуються як двійкової логіки так і трійкової, а також загальні поняття. Засоби обчислювальної техніки призначені для переробки інформації. Будь-яке керування ґрунтується на інформації. Поняття інформації має узагальнений вигляд і відображає одну з об'єктивних властивостей матеріального світу. Воно використовується в різних значеннях і має ряд визначень. Дамо, на наш погляд, більш узагальнене означення інформації.

Інформація - це сукупність повідомлень про властивості об'єктів, зміну їх стану та протікання процесів.

Інформація передається за допомогою повідомлень (показники приладів, команди керування, слова, картинки, звуки тощо). Матеріальною оболонкою повідомлень є сигнали різних видів, тобто повідомлення передаються за допомогою сигналів (радіосигнали, звукові сигнали, тощо).

Отже, загальне означення інформації, яке було б строго науковим і задовольняло б усі області науки і техніки дати дуже важко.

Двійкова система числення — позиційна система числення з основою 2. Завдяки безпосередній реалізації цифрових електронних схем на логічних елементах, двійкова система використовується практично у всіх сучасних комп'ютерах та інших обчислювальних електронних пристроях. Таким чином можна описати будь-яку ситуацію за допомогою одиниць та нулів. В електроніці все набагато простіше. Є сигнал – це одиниця, сигналу немає – це нуль. Один символ у двійковій системі називається бітом. Кожна цифра у

двійковому числі називається біт (або розряд). Чотири біти - це напівбайт, 8 біт - байт, 16 біт - слово, 32 біти - подвійне слово.

Інформація передається за допомогою сигналів, які можуть бути представлені в аналоговій або цифровій формах. Різниця між ними у тому, що перші безперервні, а другі дискретні. Сучасні комп'ютерні системи використовують цифрові сигнали, тому для обробки інформації необхідно перетворювати безперервні сигнали (в природі існують тільки аналогові сигнали) в дискретні.

За допомогою 1 байта можна кодувати 256 різних значень, тобто представити (передати, зберігати) інформацію про 256 можливих станів системи:

$$I = \log_2 256 = 8 \text{ біт} = 1 \text{ байт} . \quad (1.1)$$

Таким чином, різноманітна інформація (текст, таблиці, звуки, малюнки тощо) представляються у вигляді байтів. Однак комп'ютер повинен розрізняти, яку інформацію несе кожен байт, де вона починається і закінчується, який формат мають дані, тобто кожна послідовність байтів інформації визначеного типу повинна реєструватись.

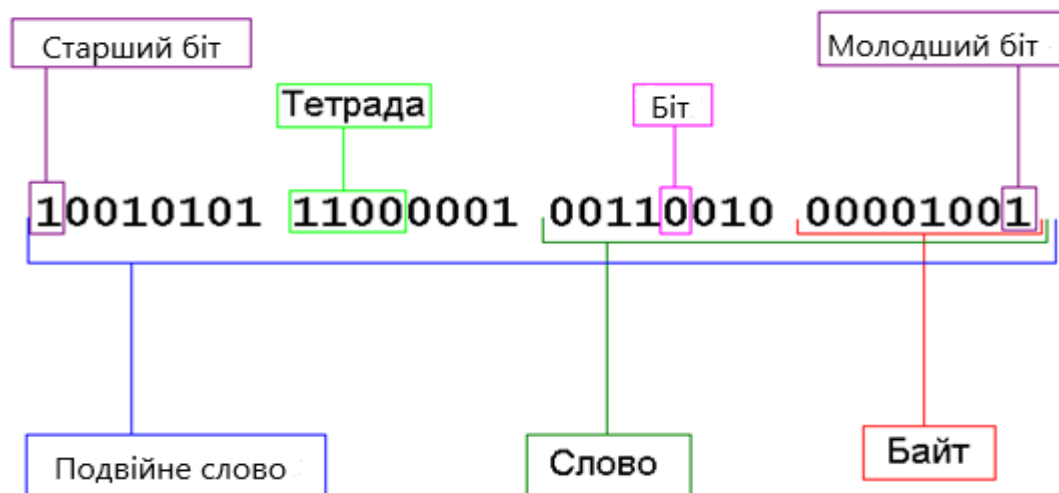


Рисунок 1.1 – Формат байту

Біт — це один знак у двійковій системі числення. За допомогою одного біта ми можемо зашифрувати дві інформації: ТАК чи НІ. А скільки інформації

можна показати за допомогою двох біт: два біти - це два знаки разом у двійковій системі числення. Це означає, що за допомогою двох бітів ми можемо закодувати 4 інформації. Припустимо, ми використовуємо два біти, отже, ми можемо закодувати $2^2 = 2 \times 2 = 4$ інформації, тобто 4 можливі варіанти, якщо ж використовуємо три біти, то $2^3 = 2 \times 2 \times 2 = 8$, значить 8 інформацій ми можемо закодувати за допомогою трьох бітів і т.д. . Неважко порахувати, що з допомогою п'яти бітів можна закодувати $2^5 = 2 \times 2 \times 2 \times 2 \times 2 = 32$.

Трійкова система числення - позиційна система числення з цілою чисельною основою, рівною 3. Існує у двох варіантах: несиметрична і симетрична.

У несиметричній трійковій системі числення частіше застосовуються цифри $\{0,1,2\}$, а в симетричній трійковій системі числення знаки $\{-,0,+\}$, $\{-1,0,+1\}$.

У деяких людей ця логіка викликає труднощі. Вони кажуть, наприклад, наведіть приклад подібної логіки у житті.

Людина, яка трохи подумала над цією логікою, зрозуміє, що вона більш життєва ніж двійкова. Звичайний приклад трійкової логіки в житті пов'язаний з постійним струмом: струм рухається в один бік, в інший бік його немає.

1.2 Обґрунтування необхідності розробки трійкових логічних елементів

Як ми знаємо будь який електронний пристрій працює за допомогою логічних елементів, але на даний час всі логічні елементи реалізовано за допомогою двійкових елементів. Відповідно до нашої задачі ми будемо використовувати трійкову систему числення, яка має ряд переваг над двійковою. Але на жаль логічних компонентів для неї реалізовано не було на сучасній компонентній базі. Тому в даному підрозділі обґрунтуємо використання саме останньої систем числення для електроніки.

Відомо, що трійкова арифметика має істотні переваги в порівнянні з двійковою, яка використовується в сучасних комп'ютерах. Дональд Кнут у зв'язку з цим висловив сподівання, що одного чудового дня заміна "фліп-флопа" на "фліп-флеп-флоп" все-таки відбудеться [1]. Ця заміна, в реальність якої при теперішньому засиллі двійкової техніки важко повірити, спростить і вдосконалисть не тільки комп'ютерну арифметику, а й інформатику в цілому, Тризначна логіка краще відповідає природі речей та неформальному мисленню людини. На жаль, сучасні дослідження за багатозначними (недвійковими) логіками носять, як правило, формальний характер і не пов'язані із запитами практики.

Інтерес до трійкової логіки виник задовго до появи комп'ютерів, у зв'язку з властивостями симетричного коду чисел. Останнім часом цей інтерес відроджується завдяки новим можливостям напівпровідникової технології. Наслідком цього буде поява дешевих інтегральних елементів із трьома станами. Крім того, проводяться теоретичні дослідження, що стосуються трирівневої техніки. Очікується, що їхні результати знайдуть своє оригінальне практичне застосування. До основних характеристик, що визначають практичну цінність трійкового коду та тризначної логіки, належить таке:

існує природне подання чисел зі знаком, тобто не потрібно користуватися штучними прийомами типу прямого, зворотного або додаткового коду;

знак числа визначається знаком старшої ненульової цифри, при цьому не потрібно використовувати спеціальний знаковий біт, як у двійковій системі;

доволі просто здійснюється порівняння чисел по величині, при цьому не потрібно звертати увагу на знак числа;

відповідно до цього команда розгалуження за знаком у трійковій машині займає у два рази менше часу, ніж у двійковій;

зменшення довжини числа рівносильне правильному округленню, а способи округлення, використовувані у двійкових машинах, не забезпечують цього;

трійковий суматор здійснює обчислення під час інвертування одного з доданків, звідки випливає, що трійковий лічильник автоматично є реверсивним;

у тривходовому трійковому суматорі перенос у наступний розряд виникає у восьми ситуаціях з 27, а у двійковому суматорі – в 4 з 8. У чотириходовому суматорі перенос також відбувається тільки в сусідній розряд;

таблиці множення та ділення майже такі ж прості, як і у двійковій системі, множення на -1 інвертує множник.

Реалізація двійкових операції І, АБО, НІ за переваги використання трійкової логіки описуються нижче.

Принципи, операції та функції трійкової логіки. Нехай ми маємо трійкову логіку типу $[0,1,2]$.

Операція трійкового мінімуму виконується так:

$$y = x_a \wedge x_b = \min(x_a, x_b). \quad (1.2)$$

Операція трійкового максимуму виконується так:

$$y = x_a \vee x_b = \max(x_a, x_b). \quad (1.3)$$

Операція трійкової інверсії виконується так:

$$y = 2 - x. \quad (1.4)$$

1.3 Наявні методи реалізації та проекти з використанням трійкової логіки

На сьогодні відомо один з найбільших проектів з використанням трійкової логіки Сетунь і також один емулятор трійкового коду Ternac.

Почнемо огляд проекту який був створений ще в СРСР і його назва «Сетунь». Вже 1956 року виникла потреба у створенні практичного зразка цифрової обчислювальної машини, яка б змогла використовуватися у вузах, лабораторіях. Для таких цілей була потрібна проста в освоєнні, надійна, недорога, але в той же час ефективна мала ЕОМ, розрахована на масове використання.

Вимоги, пред'явлені до такої машини: швидкість роботи мала дорівнювати кільком сотням операцій за секунду, «віталася» простота і зручність програмування, точність обчислень — 6-8 вірних десяткових знаків, високонадійність у експлуатації й технічне обслуговування, помірні габарити, економне споживання енергії, використання недорогих та «недефіцитних» матеріалів та деталей. Такі вимоги для того часу були «суперечливими», адже створення машини зручнішою для роботи програмістів спричинило б збільшення кількості обладнання, що, у свою чергу, спричинило б зниження надійності та підвищення вартості не тільки самої машини, а й її експлуатації.

Забезпечення зручності для програмістів окрім інженерного шляху, пов'язаного з ускладненням машини, було реалізовано програмним шляхом (розробкою систем стандартних підпрограм, введенням компілюючих та інтерпретуючих систем, програмуючих програм), застосуванням двоступеневої системи пам'яті, побудовою схем на магнітних елементах.

Використання трійкової системи числення:

У XIII столітті італійський математик Фібоначчі висунув ідею використання обчислень трійкової системи. Він сформулював і вирішив «завдання про гирі» (відому під назвою Баше-Менделєєва). Суть завдання така: за якої системи гир, маючи їх по одній, можна безкоштовно зважити всілякі вантажі Q від 0 до максимального вантажу $Q_{\max}$, щоб абсолютне

значення максимального вантажу $Q_{\max}$ було б найбільшим серед усіх можливих різновидів. Дві умови для вирішення цього завдання: якщо гирі класти на вільну чашу авторитетів; якщо гирі дозволяється класти в обидві чаші терезів. У першому випадку «оптимальна система гир» зводиться до двійкової системи гир: 1, 2, 4, 8, 16 і т.д., а «оптимальний» алгоритм, що з'являється при цьому, або спосіб вимірювання породжує двійкову систему числення, що лежить в основі «модних» комп'ютерів. У другий випадок найкращою є система гир: 1, 3, 9, 27, 81 тощо. Якщо можна класти гирі тільки на одну чашу терезів, то зручніше, швидше і економічніше робити підрахунки в двійковій системі, а якщо можна класти гирі на обидві чаші, то доцільніше вдатися до трійкової системи.

Особливість трійкової системи числення - цілочночисленна основа в ній дорівнює трьом. Тобто всі безліч цілих чисел можна записати за допомогою всього трьох цифр, наприклад 0, 1, 2, 10, 11, 12 і так далі (10 відповідає цифрі 3 зі звичної нам десяткової системи).

Протягом року фахівці аналізували та вивчали вже наявні обчислювальні машини та їх технічні можливості, в результаті було прийнято рішення – застосувати в машині не двійковий, а трійковий симетричний код. Це було б реалізацією врівноваженої системи числення. Розробка трійкової ЕОМ здійснювалася вперше.

Порівняно з двійковим кодом із цифрами 0, 1, трійковий код із цифрами -1, 0, 1 забезпечує оптимальну побудову арифметики чисел зі знаком. При такому застосуванні арифметика має ряд переваг:

- однаковість коду чисел
- варіювана довжина операндів
- єдиність операції зсуву
- тризначність функції знак числа
- оптимальне заокруглення чисел простим відсіканням молодших розрядів
- взаємокомпенсованість похибок округлення у процесі обчислення

Історія обчислювальної машини Сетунь, як і сама ця машина, незвичайна - все відбувалося всупереч загальноприйнятим підходам та методам. Можна подумати, що діяли за принципом "Роби не так, як усі". Але принцип був інший - «Чим природніше і простіше, тим краще».

Робота була проведена досить у короткий термін, що свідчило про простоту трійкової цифрової техніки, досягалося суттєве спрощення, природність архітектури трійних пристроїв. Така архітектура — раціонально побудована система програмування, яка включала інтерпретуючі системи ІІ-2 (кома, 8 десяткових знаків), ІІ-3 (кома, 6 десяткових знаків), ІІ-4 (комплексні числа, 8 десяткових знаків), ІІ-5 (плаваюча кома, 12 десяткових знаків), автокод ПОЛІЗ (мова зворотного польського запису) з операційною системою та бібліотекою стандартних підпрограм (плама кома, 6 десяткових знаків) робили малу машину Сетунь простою у освоєнні.

Як і згадувалося раніше числа і команди надані трійковим кодом (з цифрами 1, 0, -1), Сетунь оперувала 18-розрядним і 9-розрядними трійними кодами, кома завжди після другого розряду, що означало, що всі числа по модулю менше 4,5. Команди представлялися 9 трійними розрядами, 5 старших з них - адресна частина, три - це код операції і один, молодший розряд - як ознака модифікації адреси. При виконанні команд, що містять у цьому розряді 1 або -1, їх адресна частина автоматично змінювалася відповідно додаванням або відніманням числа, що зберігається в спеціальному 5-розрядному індекс-реєстрі.

Оперативна пам'ять складалася з пристрою на феритових сердечниках ємністю в 162 9-трійних осередки, тобто розділена на 3 сторінки по 54 осередки для посторінкового обміну з основною пам'яттю. Основна пам'ять - це пристрій на магнітному барабані, ємністю 36 або 72 сторінки (1944 9-трійних осередки). Передача інформації між пристроями, що запам'ятовують, проводилася зонами, що містять по 54 дев'ятирозрядних коди. За допомогою п'ятипозиційної паперової перфострічки через фотоелектричний вступний пристрій вводилися дані в машину. Такий вступний пристрій працював зі

швидкістю 800 знаків за секунду. Виведення даних здійснювалося завдяки друку та перфорації на паперовій стрічці зі швидкістю 7 знаків за секунду (виведення як трійних кодів, так і буквено-цифрового тексту із завданням довільної форми бланка).

Також давайте розглянемо реалізацію «Сетунь 70». "Сетунь 70" є двостековою машиною: стек операндів є як би розвитком регістра-акумулятора одноадресної машини, стек адрес повернення складає основу автомата, що управляє вкладеністю підпрограм. Надалі знадобилася [8] незначна модифікація цієї схеми, щоб перетворити "Сетунь 70" на машину запропонованого Е.Дейкстрою структурованого програмування [9]. Було введено операційні трайти, що дозволяють конструювати команди:

JSR A - виклик підпрограми-процедури A;

BRT A1 A2 A3 - альтернативного виклику однієї з трьох процедур, залежно від знака поточного значення вершини стека операндів;

DOW A - циклічного виклику процедури A, поки поточне значення вершини стека не дорівнює нулю.

Досягнута таким шляхом адекватна реалізація ідей Дейкстри, названа процедурним програмуванням, повністю виправдала його надії радикально вдосконалити програмістську справу (чого, як відомо, "structured revolution" так і не досягла [10]) - трудомісткість створення, обслуговування та модифікації програм на "процедурній машині" зменшилася в 3-5 разів при гарантованій їх надійності.

Проте "Сетунь 70" виявилася останнім "ternac"ом, оскільки роботи на цьому напрямі були припинені. Досвідчений зразок машини склав основу автоматизованої системи навчання "Наставник" . Процедурне програмування перетворилося на Діалогову систему структурованого програмування (ДССП), що у своїй емулює архітектуру " Сетуні 70 " , реалізуючи переваги процедурного структурування на двійкових машинах . Результатом подальшого розвитку ДССП з'явився Інтелектуальний інструментарій

інформаційних систем, що розвивається (РІПС) і високорівневе конструктне програмування, що виникло на його основі .

Всупереч поширеній думці, ніби головна перевага потрійного коду перед двійковим полягає в його економності, яка обумовлена тим, що 3 ближче ніж 2 до основи натурального логарифму $e = 2,718 \dots$, Слід сказати, що ця економія не перевищує 5%, тобто . практично не має значення. Разом з тим, може бути виправдана і суттєва неекономність кодування триту парою бітів задля реалізації дійсних переваг потрійного коду. Ці переваги обумовлені тим, що 3 - мінімальна основа системи числення з природним уявленням чисел зі знаком.

Ternac

У 1973 році в США створили Ternac - це був трійковий емулятор, написаний на fortran і працюючий на двійковому комп'ютері Burroughs B1700. Це було те, що зараз прийнято називати "proof of concept", тобто експериментом, метою якого було довести можливість реалізації обробки трійкової логіки на двійковій машині в принципі. Йдеться також про те, що швидкість і витрата пам'яті були того ж порядку, що й при двійкових обчисленнях.

1.4 Переваги та недоліки трійкової логіки в порівнянні з двійковою

Для початку про всяк випадок потрібно коротко пояснити, що ж таке трійкова система числення і чим вона відрізняється від двійкової та звичної нам десяткової. У побуті ми звикли використовувати десяткову систему і легко оперуємо цифрами від нуля до дев'яти. Є припущення, що якби еволюція замість десяти пальців дала б нам, скажімо, 12, то й основною системою числення у нас, ймовірно, була б дванадцяткова і це було б набагато зручніше, адже на відміну від десяти, дванадцять ділиться і на 2 і на 3 і на 4 і на 6, а десять - тільки на 2 і на 5. Принаймні нам набагато частіше було б зручніше ділити щось на думку. Для когось думка про те, що цифр може бути більше десяти, можливо здається дивною, але в принципі не так важливо, як ми вирішимо відображати якусь конкретну цифру на листі. У шістнадцятковій

системі числення перші десять цифр збігаються з цифрами десяткової системи, тобто, від нуля до дев'яти, а для запису цифр, що відсутні, прийнято використовувати літери англійського алфавіту від A до F. У цьому випадку це вже не літери, а саме цифри. Цифра A, цифра B тощо. Кількість цих цифр називається базою чи основою системи числення. Для трійкової та будь-якої іншої системи числення з непарною основою, існують два різні варіанти - симетрична система числення і несиметрична. Перша є симетричною щодо нуля і у випадку з трійковою системою використовує як цифри мінус (або мінус одиницю), нуль і плюс (або одиницю). Несиметрична - нуль, одиницю та двійку.

Що ж залежить від того, яким набором цифр ми оперуємо – двома, десятьма чи якоюсь іншою кількістю? Принаймні це те, скільки розрядів поміститься те чи інше число. Наприклад, число менше десяти для нас укладається в один розряд, число від десяти до дев'яносто дев'яти - у два розряди і таке інше. Тобто, кількість розрядів для збереження одного й того ж обсягу інформації - це тобто кількість цифр, які нам потрібні для того, щоб це число записати. Для десяткової системи два більше двадцяти в 10 разів, тобто, підстава системи числення говорить про те, в яку кількість разів цифра в старшому регістрі (тобто стоїть ліворуч) більша за аналогічну цифру в молодшому регістрі (тобто стоїть правіше). Наприклад, одиниця у двійковому записі це одиниця. Одиниця і нуль праворуч - вже вдвічі більше, тобто двійка, а одиниця з двома нулями - це два в третьому ступені - два, помножене на два, помножене на два, тобто вісім і так далі. Системи числення, побудовані згідно цьому принципу називаються позиційними.

Ну і ще один приклад, який дозволить відчувати різницю на основі систем числення. Якщо ми візьмемо десять розрядів для зберігання числа з використанням несиметричної системи числення, то максимальне число в двійковій системі, що міститься в ці десять розрядів, буде дорівнює 1024 (насправді 1023, тому що нуль - це теж число, але припустимо, ми рахуємо з

одиниці), у трійковій системі числення це буде число 59049, а в десятковій це вже десять мільярдів.

Чому ж практично всі і сучасні та архаїчні процесори, мікроконтролери, системи зберігання та обробки даних використовують саме двійкову систему числення? Адже якби вона була потрібною, то за один такт якийсь мікроконтролер зміг би оперувати числами, що мають набагато більші абсолютні значення. Якщо ж перевести системи зберігання на трійкові, то й тут, начебто, теж відкриваються непогані перспективи.

Насправді все дуже просто і тут знову можна повернутися до аналогії з десятьма пальцями - звичний комп'ютер має очевидні апаратні обмеження - у нього для рахунку спочатку було всього два, якщо можна так висловитися, "віртуальних пальця" - нуль і одиниця. Колись існували комп'ютери, побудовані на базі реле, потім були лампові комп'ютери, ну і нарешті, комп'ютери на транзисторах, якими ми користуємося і до цього дня, просто транзистори дуже компактно упаковані в процесори (це звичайно спрощено), але протягом усього цього еволюційного процесу будь-яка мінімальна логічна одиниця могла набувати лише двох значень. Якщо розглядати реле, воно або включено, або вимкнено, воно не може бути включено наполовину. Транзистор, наприклад, може бути відкритий не повністю, але в нього нелінійна характеристика внутрішнього опору. Створити процесор, всередині якого можна буде утримувати безліч транзисторів у напіввідкритому стані - інженерно набагато складніше завдання, ніж використовувати транзистори як ключі - або транзистор відкритий, або закритий. Магнітне поле має лише дві орієнтації полярності - плюс та мінус. Тобто ділянка магнітної поверхні на диску або стрічці також може приймати всього два стани. Отвори в перфокарті або є, або їх немає. Флеш пам'ять спочатку була побудована на транзисторах з плаваючим затвором, що приймає два стани.

Мені стало цікаво і я вирішив розібратися, що можна було б використати зараз для задоволення першого та другого пунктів.

Магнітний спосіб зберігання інформації, очевидно, не підходить через особливості фізичних законів. А ось про флеш пам'яті я дізнався для себе багато нового. З того, що стосується нашої теми, потрібно сказати, що елементарний осередок флеш пам'яті буває двох типів - SLC (single-level cell) та MLC (multi-level cell). Осередок першого типу зберігає в собі один біт інформації і може приймати лише один рівень заряду, а от осередок другого типу - може приймати кілька рівнів заряду. Під MLC зазвичай розуміють комірку з чотирма можливими рівнями заряду, яка зберігає два біти, але існують також і TLC (Triple-level cell) та експериментальні QLC (Quad-level cell), які можуть зберігати 3 та 4 біти інформації відповідно, тобто вони мають можливість зберігати 8 чи 16 рівнів заряду. Отже, зробити комірку для зберігання трійкових даних у чистому вигляді - зараз цілком реально. Достатньо реалізувати зберігання трьох, шести або дев'яти рівнів заряду. Чим більша кількість рівнів заряду на комірці зберігається, тим менша її надійність і менша кількість можливих циклів перезапису. Тобто, шість рівнів на даний момент найімовірніше можна назвати оптимальним числом.

Тепер про заміну транзисторів: із цим все набагато складніше. У принципі, транзистори давно використовують у складі підсилювачів. Тобто теоретично ми могли б використовувати звичайні транзистори, які будуть у кожний момент часу або закриті, або відкриті, або відкриті наполовину. Але так як у наборах машинної логіки транзистори працюють у ланцюжках і логічний вихід одного транзистора може бути з'єднаний з логічним входом іншого, то струм, що виходить навіть з відкритого наполовину транзистора, спрямований на вхід іншого, відкриє його вже повністю і проміжний стан перенести не вдасться. Можливо, якщо змінити співвідношення площ переходів база-колектор і база-емітер, вдасться отримати пропорційний електронний ключ, транзистор, який не посилює сигнал, а лише відкривається повністю або наполовину, але і в цьому випадку є проблеми - замість цифрового комп'ютера ми фактично будемо збирати аналоговий, а похибка в рівні сигналу при проходженні через кожен такий транзистор буде

накопичуватися і в результаті призведе до перетворення одного значення на інше і, як наслідок, до помилок.

У мережі є безліч схем логічних блоків із трьома стійкими станами, побудованих на парі - трійці окремих транзисторів та невеликій обв'язці і всі описані вище завдання реалізовані, але процес їх реалізації набагато складніший, ніж звичні нам пристрої, що базуються на двійковій логіці, які теж простими назвати зовсім не можна.

Тому слід розуміти, що якщо зараз створити трійковий комп'ютер, то через фундаментальні відмінності у своїй архітектурі він не зможе працювати на жодній з існуючих операційних систем. Розробку доведеться починати з самого нуля – з мов програмування та ядра операційної системи, а це сотні тисяч годин роботи кваліфікованих фахівців.

Чому ж трійкові комп'ютери не зустрічаються зараз? Вони банально нерентабельні. Коли я казав, що деякі операції на трійковому комп'ютері будуть виконуватися швидше, ніж на двійковому, я зробив припущення. Це було б правдою, якщо припустити, що трійковий процесор працюватиме на тій же тактовій частоті і за кожен такт оброблятиме не меншу кількість інструкцій на не меншому наборі даних, що й двійковий процесор.

Величезні корпорації виробляють процесори, пам'ять та накопичувачі, побудовані на двійковій логіці. Як транзистори, так і самі процесори пройшли довгий еволюційний шлях, вони ставали все швидше, менше в розмірах, все енергоефективніше. Якщо ми навіть знайдемо елемент, що приймає три стабільні стани і здатний замінити собою транзистор, то навряд чи зможемо відразу досягти від нього тієї ж ефективності, що й від сучасних транзисторів. Структура троїчного процесора буде набагато складніша, він неминуче матиме більший розмір і менші частоти, що швидше за все зведе нанівець вигреш у продуктивності. Виробництво трійкової машини буде настільки дорожчим, що вигідніше буде придбати набагато потужнішу двійкову машину за меншу ціну і симулювати на ній трійкову логіку, або вирішити необхідне завдання алгоритмічно.

Тепер давайте більш детально розглянемо трійкову систему обґрунтовуючи плюси і мінуси, відштовхуючись від математичних досліджень. Завдяки наявності в трійковій системі числення "позитивної" і "негативної" цифр, в коді числа немає особливого розряду знака, а це дуже спрощує логіку арифметичних операцій. Трійкове слово в 1,6 рази коротше двійкового, отже і операції, наприклад операція додавання в трійковому послідовному арифметичному пристрої виконуються також в 1,6 рази швидше, ніж у двійковому. Перевагою трійкової системи числення з цифрами -1, 0, 1 є:

1) заокруглення числа x до k вірних трійкових знаків, що виходить відкиданням усіх молодших знаків, починаючи з $(k+1)$.

Наслідок. Найближчим цілим числа x ,

$$X = E_1, E_2 \dots E_{m+1} \dots XE$$

є число:

$$[X]_{\text{бл}} = E_1, E_2 \dots E_m$$

де E_i – трійкові цифри. Виходить, що простий спосіб виділення найближчої цілої частини числа полегшує алгоритми обчислення експотенційної та тригонометричних функцій.

Цій трійковій системі не потрібно спеціального знакового розряду, знак числа визначається знаком старшої трійкової цифри. Це спрощує події з відносними числами.

Вміст будь-якого регістру автоматично сприймається як відносне число. Операції зсуву та нормалізації – універсальні і до того ж сильно спрощують програмування дії з масштабами та з плаваючою комою.

Все це «сприятливо» для побудови підпрограми для виконання дій з плаваючою комою та обчислення елементарних функцій з повною

обчислювальною похибкою, що не перевищує двох одиниць молодшого розряду мантиси.

Як і двійкова система, трійкова система обчислення заснована на позиційному принципі кодування чисел, але вага i -ї позиції/розряду в ній дорівнює не 2 і, а 3 і. Розряди тризначні, а чи не двозначні: до 0 і ще допускається третє значення -1, тому одноманітно представні як позитивні, і негативні числа.

Значення n - тритного цілого числа N визначається аналогічно до значення n - бітного:

$$N = \sum_{i=0}^{n-1} \alpha_i 3^i = \alpha_{n-1} 3^{n-1} + \alpha_{n-2} 3^{n-2} + \alpha_{n-3} 3^{n-3} + \dots + \alpha_1 3 + \alpha_0, \quad (1.4)$$

де $\alpha_i \in \{1, 0, -1\}$ значення цифри i розряду.

Ось як виглядатимуть десяткові числа 13, 7, 6, -6 у трійковому записі $13 = +++$, $7 = +-+$, $6 = +-0$, $-6 = -+0$. Зміна знака числа в симетричному коді дорівнює потритної інверсії, відбувається взаємозаміна "+" на "-", і навпаки.

Нижче наведено таблицю з операціями додавання та множення в трійковому симетричному коді.

Таблиця 1.1.

Операції додавання та множення

	+	0	-
+	+-	+	0
0	+	0	-
-	0	-	-+

	+	0	-
+	+	0	-
0	0	0	0
-	-	0	+

У потрібному симетричному коді відсутня проблема чисел зі знаком.

На відміну від двійкової, це арифметика чисел зі знаком. Знак числа - це цифра старшого з його значних (ненульових) розрядів. Проблеми чисел зі знаком, що не має в двійковому коді досконалого рішення, у трійковому симетричному коді просто не може бути, а це вже важлива перевага.

Тритом було названо три види сигналів, що формуються базовим елементом майбутнього трійкового комп'ютера. Якщо за міру кількості інформації приймати біт, то інформаційна ємність триту дорівнюватиме приблизно 1,6. Тому трійковий комп'ютер обробляє в одиницю часу більше інформації, ніж двійковий.

Мінімальною адресованою одиницею пам'яті проєктованого потрібного комп'ютера став трайт, що дорівнює шести тритам і приймає значення від -364 до 364. Робота з діапазоном негативних значень - особливість, що відрізняє трайт від двійкового байта, значення якого поширюються від 0 до 255.

Завдяки інформаційній ємності трайта з його допомогою легко можна закодувати всі великі та малі символи російського та латинського алфавітів, математичні та службові символи.

Переваги трійкової системи числення перед двійковою.

Трійкова логіка має низку переваг перед двійковою. Нижче наведено основні:

Трійкова СЧ дозволяє вміщувати більший діапазон чисел у пам'яті трійкового комп'ютера, оскільки:

$$3^n > 2^n. \quad (1.5)$$

Очевидно, що трійкова СЧ використовує менше розрядів для запису чисел, порівняно з двійковою СЧ. Наприклад:

$$11101012 = 111003$$

$$10002 = 223$$

Для трійкової СЧ використовується несиметричний набір $\{0,1,2\}$.

Ці дві важливі переваги перед двійковою системою числення говорять про більшу економічність потрійної системи числення.

Економічність системи числення — можливість представлення якнайбільшої кількості чисел з використанням як найменшої загальної кількості знаків.

Доведемо економічність потрійної системи числення математично.

Нехай p — основа системи числення, а n — кількість потрібних знаків. Для запису n знаків знадобиться n/p розрядів, а кількість чисел, яку можна записати, дорівнюватиме $p^{n/p}$.

Розглянемо функцію :

$$f(p) = p^{n/p} \quad (1.6)$$

Для того щоб визначити максимальне значення функції, знайдемо її похідну:

$$f'(p) = -n(p^{n/p-2})(\ln p - 1) \Rightarrow \ln p - 1 = 0, \ln p = 1, p = e \quad (1.7)$$

$e \approx 2,71$, найближче число до e — 3. Таким чином, трійкова СЧ не тільки економічніша за двійкову, а й економічніша за будь-яку іншу СЧ.

Трійкова логіка включає майже всі можливості двійкової логіки.

Комп'ютер, заснований на трійковій логіці, має велику швидкодію. Наприклад, трійковий суматор і напівсуматор у трійковому комп'ютері при

складанні тритів виконує приблизно в 1,5 рази менше операцій складання порівняно з двійковим комп'ютером.

1.4 Висновки до розділу 1

В даному розділі було проведено аналіз літературних джерел. Було детально розглянуто трійкову систему, проведено порівняння з іншими. Також було виявлено основні переваги та недоліки як двійкових систем числення так і трійкових, а також логічних елементів. Тепер давайте розглянемо детальніше про що йдеться в підпунктах першого розділу. В першому підпункті розглянуто основні визначення, що використовуються для даного дослідження, це такі визначення які використовуються в доволі вузькій сфері електроніки, тому такий пункт має місце в розгляді літературних джерел. В другому підпункті було проведено обґрунтування необхідності розробки трійкових логічних елементів це також важливий пункт, оскільки в ньому розглянуті можливості та переваги трійкових логічних елементів. Наступним став один із самих інформативних пунктів в якому йдеться про наявні методи реалізації та проекти з використанням трійкової логіки. В ньому розглянули один із найбільших проектів з використанням трійкової логіки, а саме проект «Сетунь». І в останньому дослідили переваги та недоліки трійкової логіки в порівнянні з двійковою і цим самим створили міцний фундамент для подальшої розробки та дослідження згідно нашої теми.

РОЗДІЛ 2

АНАЛІЗ ОСОБЛИВОСТЕЙ РОЗРОБКИ ЛОГІЧНИХ ЕЛЕМЕНТІВ ДЛЯ ТРІЙКОВОЇ ЛОГІКИ

2.1 Особливості роботи наявних пристроїв трійкової логіки

Як видно з того що було проаналізовано при аналізі літературних джерел трійкова логіка не була поширена і майже не розвивалася. Є тільки певні згадки про неї, наприклад такою згадкою є інформація про розробку ЄВМ основою на трійковій логіці «Сетунь». Але на захист трійкової логіки можна сказати що іноді для виконання якихось специфічних завдань дійсно вигідніше розробити вузькоспеціалізований пристрій, який виконуватиме суворо обмежений набір логічних операцій, зате це робитиме набагато швидше. Коли був бум майнінгу криптовалют, серед іншого розроблялися і так звані ASIC майнери, виконані у формі фактора невеликого пристрою, що підключається USB і нагадує велику флешку. На борту вони несли спеціалізований чіп, заточений виключно для обчислення хешсум. Таким чином деякі люди майнули криптовалюту, підключивши кілька десятків таких пристроїв до якоїсь Raspberry PI через USB хаби із зовнішнім живленням.

Оскільки нам потрібно зробити аналіз логічних елементів які використовувались при роботі з трійковою системою ми будемо розглядати саме наявні логічні компоненти єдиного проекту про який нам відомо. Тому давайте розглянемо і проведемо повний аналіз схеми блок-схеми обчислювальної машини "Сетунь".

Машина складається з шести функціональних пристроїв: 1) арифметичного пристрою, 2) пристрою управління, 3) оперативного пристрою, 4) пристрою введення, 5) пристрою виведення, 6) пристрою на магнітному барабані.

Передача інформації між блоками, у тому числі введення та виведення її з машини, проводиться через оперативний пристрій. Цей пристрій має 9-розрядний зсувний регістр, за допомогою якого інформація, що надходить на його вхід у послідовній формі, перетворюється на паралельну форму для запису в запам'ятовуючий пристрій, а інформація, що зчитується паралельно з пристроєм, що запам'ятовує, перетворюється в послідовну форму. Ця інформація направляється у той чи інший пристрій залежно від стану перемикача П1, який у кожний момент визначається програмою, що виконується. Арифметичний пристрій машини здійснює виконання команд додавання, віднімання, множення, порозрядного множення, зсуву, нормалізації, а також команд посилки чисел до регістру арифметичного пристрою та запису результату в оперативний пристрій.

В арифметичному пристрої є два регістри: регістр множника R і регістр результату S (акумулятор). Регістр R складається з 18 тригерів, що управляють ключами помножувального пристрою. Регістр S являє собою 18-розрядний тригерний регістр зі зсувом вліво та вправо.

Число з регістру S за допомогою перемикача П2 може бути спрямоване одним з чотирьох каналів. При виконанні команди додавання або віднімання воно подається на вхід суматора арифметичного пристрою одночасно з надходженням на другий вхід цього суматора числа, вибраного з оперативної пам'яті. При множенні число з регістра 5 може бути послано або в регістр R як множник, або на вхід множника в якості множника. У разі запису вмісту регістра 5 в оперативну пам'ять перемикач П2 з'єднує вихід регістра 5 з входом регістра пристрою.

У регістр S число може бути прийнято з пристрою, що запам'ятовує, з розмножувального пристрою і з лічильника зрушень при нормалізації, причому у всіх випадках воно проходить через суматор.

Прийом числа в регістр S супроводжується виробленням ознаки w , $w(S)$, за правилом: якщо число позитивно, то $w=1$, якщо негативно, то $w=-1$, якщо дорівнює нулю, то $w=0$.

При надходженні числа в реєстр R або реєстр модифікації F пристрою управління виробляється за тим же правилом ознака w , $w(R)$ або $w(F)$.

Залежно від значення цієї ознаки відбувається передача управління при виконанні команд умовного переходу. При операціях, які пов'язані з виробленням ознаки z , зберігається його значення, вироблене попередньої операцією.

У пристрої управління є три реєстри: 9-розрядний реєстр команди K, 5-розрядний реєстр адреси команди C та 5-розрядний реєстр модифікації (індекс-реєстр) F.

Реєстр До складається з двох частин: 5-розрядного тригерного реєстру зі зсувом вправо, в який поміщаються адреси чисел і команд при вибірці цих чисел і команд з оперативного пристрою і чотирьох тригерів коду операції, з яких три визначають характер послідовності керуючих імпульсів залежно від прийнятого ними коду, а четвертий управляє механізмом модифікації адресної частини команди.

Пристрій управління має власний суматор, за допомогою якого проводиться послідовна зміна адрес виконуваних команд і здійснюються операції, пов'язані з реєстром модифікації F, в тому числі і сама модифікація адресних частин команд.

На момент початку операції адресу команди, що підлягає виконанню, приймається в реєстр C, а сама команда — в реєстр K, причому першими надходять молодші розряди команди, що містять ознаку модифікації та код операції. Адресна частина команди проходить у реєстр D0 через суматор, на другий вхід якого може бути подано вміст індексу-реєстру F з позитивним або негативним знаком залежно від ознаки модифікації.

Встановлена в реєстрі D0 модифікована адресна частина команди в залежності від характеру виконуваної операції використовується або як адреса числа, що вибирається з оперативної пам'яті, або як адреса, за якою в пам'ять проводиться запис числа, або як адреса команди, якій має бути передано

управління, або як умовне число, що містить інформацію про обмін групою кодів між оперативною пам'яттю та магнітним барабаном, а також про роботу із зовнішніми пристроями.

Під час роботи машини з довгими осередками (18 розрядів) читання (запис) проводиться двічі: спочатку зчитуються чи записуються 9 молодших розрядів числа, а потім — 9 старших розрядів. Адреса довгого осередку пам'яті містить у п'ятому розряді -1; адресу її половини, що відповідає молодшим розрядам, містить у цьому розряді 1; адреса старшої половини - 0.

При виконанні команд безумовної передачі управління, так само як і умовної, у разі якщо передача управління дійсно проводиться, прийнята в регістр адресна частина команди передачі управління сприймається як адреса нової команди; відбувається вибірка цієї команди з пам'яті і прийом її в регістр. Тоді як сама адреса передається з регістру до регістру С. Якщо ж передача управління не повинна проводитися, то механізм вибірки чергової команди працює так, як при будь-якій іншій операції, а саме: адреса виконаної команди направляється з регістру через суматор в регістр. Причому у відповідний момент на другий вхід суматора подається імпульс, під дією якого відбувається послідовне збільшення адреси, завдяки чому в регістр надходить адреса наступної по порядку команди.

Структура циклу, відповідного тієї чи іншої операції, задається розподільником імпульсів (на блок-схемі не показаний), який являє собою замкнуте кільце лінії затримки, в різних точках якої є відгалуження, що видають керуючі імпульси в потрібний момент в залежності від коду виконуваної операції. Час пробігу імпульсу по розподільнику переважно операцій дорівнює 180 мксек. При виконанні операцій множення, зсуву, нормалізації та запису з регістра 5 на згадку про довжину кільця розподільника імпульсів і відповідно тривалість циклу збільшуються. При виконанні операцій передачі управління у разі, якщо передача відбувається, довжина цього кільця та тривалість циклу зменшуються до 100 мксек.

Під час роботи з магнітним барабаном або з пристроями введення - виведення основний пристрій управління стає в режим очікування, підкоряючись імпульсам, що надходять з автономних пристроїв управління магнітним барабаном, введенням і виведенням.

При роботі в однотактному режимі кільце розподільника імпульсів розмикається, і для кожної команди необхідно подати на вхід розподільника пусковий імпульс. Цей імпульс формується схемою одного імпульсу при кожному натисканні кнопки “пуск” пульта управління. У випадку, коли машина знаходиться в автоматичному режимі, пусковий імпульс, що надійшов у розподільник, циркулює в його кільці до тих пір, поки робота машини не буде перервана одною з "зупинок": командою "зупинка", переповненням в регістрі S, кнопкою "зупинка" або ключем “однотактний режим” на пульті керування.

Система команд машини дана в наведеній нижче таблиці. У цій таблиці код операції разом із ознакою модифікації, покладеним рівним нулю, записується за допомогою двох дев'яткових цифр. Цифри з рискою нагорі (у цій публікації верхня риска замінена префіксом “знак оклику” — прим. ред.) позначають відповідні “негативні” цифри (наприклад, символ !3 позначає -3).

2.2 Аналіз особливостей роботи трійкових логічних елементів

Ми розглядатимемо просту трійкову функціональну схему — трійковий суматор. У ньому використовуються такі позначення: $\{0,1,2\}$ (несиметрична трійкова система числення).

Одномісні операції. За аналогією з двійковою логікою, в трійковій логіці існує всього 3^{3^n} операцій для n аргументів. Таким чином, у трійковій логіці всього існує $3^{3^1} = 27$ одномісних операцій.

Інверсія:

NOT-, NOT та NOT+ — оператори інверсії, що зберігають стан -, 0 і + відповідно, коли воно відповідає типу оператора, або звертають у значення, не

рівне вихідному стану і не відповідає типу оператора інверсії, тобто в третє, що залишилося.

Наприклад, якщо $a=(-)$, $\text{NOT}+a=0$. Оскільки вихідний стан $(-)$, тип інверсії $\text{NOT}+$, методом виключення можна дійти результуючого стану 0.

Усі можливі варіанти для цієї одномісної операції наведено у таблиці.

Таблиця 2.1

Варіанти одномісної операції

aa	NOT-	NOT	NOT+
—	—	+	0
0	+	0	—
+	0	—	+

Операція вибору:

$S-$, S та $S+$ - оператори вибору. Перетворюють стан, відповідний типу оператора $(+)$, у разі будь-якого з інших двох станів змінна набуває значення $(-)$.

Таблиця 2.2

Операція вибору

a	$S-$	S	$S+$
—	+	—	—
0	—	+	—
+	—	—	+

Модифікація:

INC та DEC — оператори модифікації, відповідно збільшення та зменшення триту на одиницю за модулем три. При переповненні триту рахунок починається знову ($\text{INC}(+)=(-)$).

Таблиця 2.3

Оператори модифікації

a	INCa	DECa
–	0	+
0	+	–
+	–	0

Порогове збільшення та зменшення:

$\nearrow$, $\searrow$ — дані оператори працюють аналогічно операторам модифікації лише з тією відмінністю, що при переповненні трита цикл станів не повторюється, і значення і залишається мінімальним чи максимальним.

Таблиця 2.4

Порогове збільшення та зменшення

a	$\nearrow a$	$\searrow a$
–	0	–
0	+	–
+	+	0

Інші одномісні функції:

$+$, 0 і $-$ функції, не залежать від аргументу a , вони ж вироджені.

Функція a — тотожна і вироджена функція.

Інші функції від однієї змінної утворюються шляхом поєднання операторів вибору з операторами інверсії та модифікації, тому вони не мають власних назв.

Двомісні операції:

Легко бачити, що в трійковій логіці існує $3^{3^2} = 19683$ двомісні операції.

У таблиці наведено найголовніші і практично корисні з них.

Таблиця 2.5

Найголовніші двомісні операції в трійковій логіці

a	b	$a \wedge b$	$a \vee b$	$a \cdot b$	$a \oplus b$	$a b$	$a + b$	$a \uparrow b$	$a \rightarrow b$	$a \text{ cmp } b$	$a \&_L b$
—	—	—	—	+	+	0	—	—	—	0	—
—	0	—	0	0	—	+	—	0	0	—	—
—	+	—	+	—	0	—	0	+	—	—	—
0	—	—	0	0	—	+	—	0	0	+	—
0	0	0	0	0	0	+	0	—	+	0	—
0	+	0	+	0	+	—	+	+	0	—	0
+	—	—	+	—	0	—	0	+	—	+	—
+	0	0	+	0	+	—	+	+	0	+	0
+	+	+	+	+	—	—	+	—	—	0	+

Таблиця 2.6

Найголовніші двомісні операції в трійковій логіці

$a \rightarrow_L b$	$a \wedge_+ b$	$a \rightarrow_+ b$	$a \rightarrow_G b$	$a \rightarrow_M b$	$a \rightarrow_B b$	$a \equiv b$
+	—	+	+	+	+	+
+	0	0	+	+	0	—
+	—	—	+	+	0	—
0	0	+	—	0	0	—
+	0	0	+	0	0	+
+	0	0	+	+	0	—
—	—	+	—	—	—	—
0	0	0	0	0	0	—
+	+	+	+	+	+	+

Алгебраїчні властивості:

Усі нижчеперелічені закони та властивості легко доводяться шляхом перебору всіх значень змінних, що входять до них. Алгебраїчний підхід полягає в тому, щоб визначити над безліччю $\{-, 0, +\}$ двомісні ($\wedge$, $\vee$) і одномісні ($'$, S , $\neg$) операції за допомогою законів, а властивості, що залишилися, вже виводити з них алгебраїчно.

Властивості констант:

$$\begin{aligned}
 a \wedge (+) &= a \\
 a \wedge (-) &= (-) \\
 a \vee (+) &= (+) \\
 a \vee (-) &= a \\
 \overline{(-)} &= (+) \\
 \overline{(+)} &= (-)
 \end{aligned}
 \tag{2.1}$$

Для кон'юнкції та диз'юнкції у трійковій логіці зберігаються комутативний, асоціативний та дистрибутивний закони, закон ідемпотентності.

Закон подвійного заперечення (заперечення Лукасевича) та потрійного (циклічного) заперечення:

$$\begin{aligned}
 \overline{\overline{a}} &= a \\
 a''' &= a
 \end{aligned}
 \tag{2.2}$$

Буквальне визначення циклічного заперечення впливає з таких властивостей:

$$\begin{aligned}
 (-)' &= 0 \\
 0' &= (+) \\
 (+)' &= (-)
 \end{aligned}
 \tag{2.3}$$

Незмінність третього стану (0) при запереченні Лукасевича:

$$\begin{aligned}
 \overline{0} &= 0 \\
 \overline{(a \wedge 0)} &= \overline{a} \vee 0
 \end{aligned}
 \tag{2.4}$$

Для законів двійкової логіки, не справедливих для потрійної, існують їхні трійкові аналоги. Закон несумісності станів (аналог закону суперечності у двійковій логіці):

$$\begin{aligned}
Sa \wedge Sa'' &= (-) \\
Sa' \wedge Sa'' &= (-) \\
Sa' \wedge Sa &= (-)
\end{aligned}
\tag{2.5}$$

Закон виключеного четвертого (замість закону виключеного третього),
він же закон повноти станів:

$$\begin{aligned}
Sa' \vee Sa \vee Sa'' &= (+), \\
S^{-}a \vee Sa \vee S^{+}a &= (+)
\end{aligned}
\tag{2.6}$$

Тричленний закон Блейка-Порецького:

$$\begin{aligned}
a \vee Sa' \wedge b \vee Sa \wedge b &= a \vee b, \\
a \vee S^{-}a \wedge b \vee Sa \wedge b &= a \vee b
\end{aligned}
\tag{2.7}$$

Закон тричленного склеювання:

$$\begin{aligned}
a \wedge Sb' \vee a \wedge Sb \vee a \wedge Sb'' &= a, \downarrow \\
a \wedge S^{-}b \vee a \wedge Sb \vee a \wedge S^{+}b &= a
\end{aligned}
\tag{2.8}$$

Закон узагальненого тричленного склеювання:

$$\begin{aligned}
a \wedge Sd' \vee b \wedge Sd \vee c \wedge Sd'' \vee a \wedge b \wedge c &= a \wedge Sd' \vee b \wedge Sd \vee c \wedge Sd'', \downarrow \\
a \wedge S^{-}d \vee b \wedge Sd \vee c \wedge S^{+}d \vee a \wedge b \wedge c &= a \wedge S^{-}d \vee b \wedge Sd \vee c \wedge S^{+}d
\end{aligned}
\tag{2.9}$$

РОЗДІЛ 3

РОЗРОБКА ТРІЙКОВИХ ЛОГІЧНИХ ЕЛЕМЕНТІВ

3.1 Вибір методів та компонентів для розробки трьохрівневої логіки

Апаратну реалізацію елементів тризначної логіки можна розглядати насправді з двох позицій. Перша – моделювання трійкових функцій за допомогою кодування у двійковому вигляді (binary coded ternary, BCT representation). У цьому уявленні існує кілька варіантів кодування, але у разі на два біта припадає один трит і ще один рівень залишається невизначеним. Його можна використовувати для позначення розділової коми в числах або індикації високоімпедансного стану на виході. Двійкове моделювання обумовлено відсутністю доступних трирівневих елементів та використанням замість них стандартних бінарних. Загальний підхід до синтезу BCT схем розглянуто у роботах [10,11] і полягає в наступному. Складається таблиця істинності для тризначної функції відповідно до обраного кодування. На основі цієї таблиці для кожного вихідного розряду складаються карти Карно (Karnaugh maps) для мінімізації моделюючої функції. За результатами мінімізації провадиться синтез бінарних схем. Як приклад наведемо схему дворозрядного BCT суматора. Вибране кодування представлене у таблиці 3.1.

Зазначимо, що вибір розрядності BCT пристроїв доцільно виробляти орієнтуючись на розрядність двійкової пам'яті: 8, 16, 32 біти. Це пов'язано з тим, що вона входить до складу практично будь-якого сучасного цифрового пристрою відносно дешевих мікросхем. Найбільш раціонально використовувати розрядність 16 біт, виходячи з параметра, який характеризує скорочення розрядної ємності. Очевидно, що цей параметр є великим недоліком даного уявлення і обмежує збільшення розрядності BCT пристроїв. Зазначимо також, що 8 кодованих трит еквівалентні 1267 біт. Цього цілком достатньо таких застосувань як обробка звуку.

Таблиця 3.1

Таблиця істиності для тризначної функції

3-bit binary	000	001	010	011	100	101	110	111
Decimal	0	1	2	3	4	5	6	7
2-digit ternary	00	01	02	10	11	12	20	21
BCT	0000	0001	0010	0100	0101	0110	1000	1001

Другий підхід реалізації трьохрівневих пристроїв полягає у використанні керованих електронних ключів при симетричному живленні. Схема показана на рис. 3.1 може працювати як повторювач або як інвертор трирівневого сигналу. Умова її функціонування очевидна: у певному логічному стані замкнутий лише один ключ при розімкнутих інших. Допустимо це пара ключів $S(1)$ і $S(-1)$, які являють собою біполярні транзистори. В результаті отримаємо наочну схему повторювача на основі двотактного каскаду.

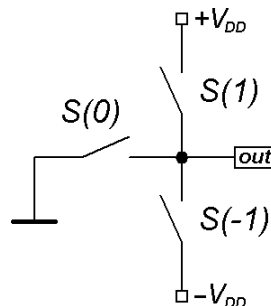


Рис. 3.1 – Схема керованих електронних ключів

Великий інтерес до створення трирівневих пристроїв представляють схемотехнічні аналоги напівпровідникових однокристальних негатронів. Вони побудовані на транзисторах і поділяються також на дві групи на вигляд вольт-амперної характеристики. Серед аналогів, що мають N-подібну ВАХ, найбільш відомий лямбда-діод, що складається з двох польових транзисторів. Його схема в оригіналі містить n-канальний та р-канальний польові

транзистори з керуючим переходом. Як відомо, при напрузі $U_{GS} = \pm 0,7 \text{ V}$ цей р-n перехід відкривається і транзистор перестає працювати. З цієї причини, а також тому, що КМОН-технологія домінує в даний час, більш практично використовувати у схемі (рис. 3.2) два нормально відкритих польових транзистора з ізольованим затвором. Якщо на один з них подавати напругу, що управляє, то отримаємо можливість задавати ВАХ лямбда-діода.

Таким чином, є два варіанти лямбда-транзистора (рис. 3.3, 3.4), назовемо їх NLT і PLT. Вихідні вольт-амперні характеристики цих приладів наведено на рис. 3.5, 3.6 відповідно. Зауважимо, що відповідно до своєї ВАХ транзистор PLT є обмежувачем струму. З використанням таких лямбда-транзисторів можна збудувати бінарні логічні елементи. До них входять повторювач, інвертор, двовходові елементи "І", "АБО", а також RS- та D-тригер. Якщо використовувати ідею керованих ключів при симетричному живленні разом із запропонованими схемами, можна побудувати тривірневі логічні елементи.

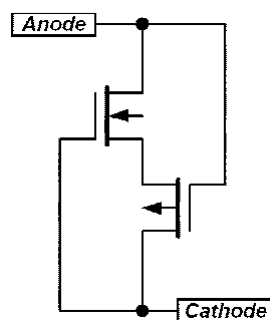


Рис. 3.2 – Лямбда діод

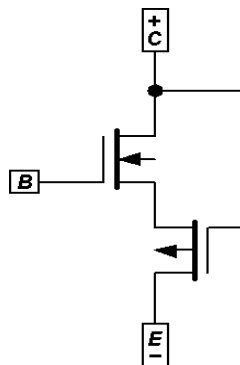


Рис 3.3 Лямбда – транзистор (NLT)

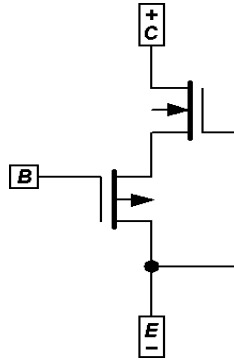


Рис. 3.4 – Лямбда- транзистор (PLT)

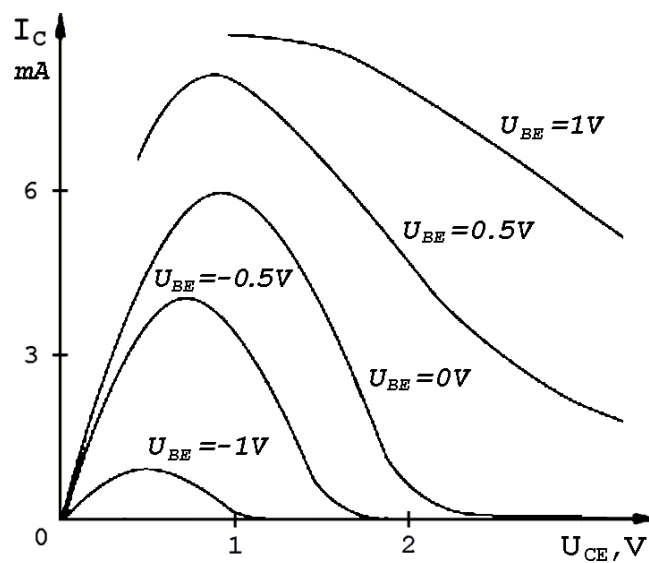


Рис. 3.5 – ВАХ лямбда-транзистора NLT

Технічна реалізація досягнутих теоретико-прикладних результатів у сфері багатозначної логіки поки що викликає труднощі, зокрема, через відсутність елементної бази, що має характеристики такого ж порядку, як і база двозначної логіки, що обумовлено проблемами сучасної технології.

Крім того, в цій прикладній області не вирішеним виявився клас завдань, пов'язаний з відображенням алгоритмів у структурі обчислювальних засобів, що максимально використовують можливості підвищення продуктивності та надійності. Це є наслідком жорсткої орієнтації математичних моделей обробки

на апарат теорії багатозначних функцій логіки алгебри, в результаті це питання знову-таки зводиться до проблеми елементної бази.

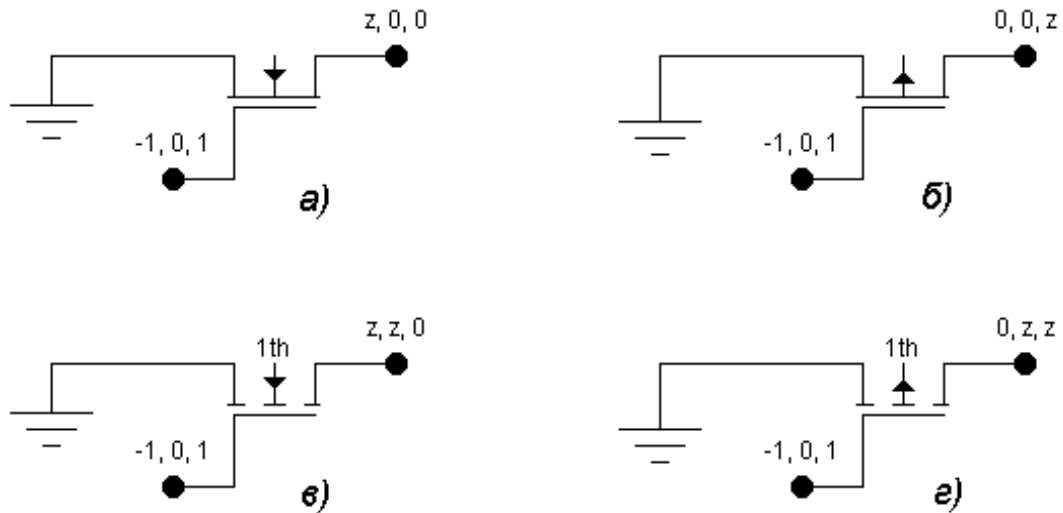


Рис. 3.6 МОН-транзистори, що комутують інформаційний сигнал "0", показані стани на виході (стоку) залежно від керуючих сигналів.

Транзистор із вбудованим каналом n-типу на рис. 3.7 виконує функцію, а з вбудованим каналом р-типу виконує функцію.

Транзистори з індукованими каналами n- і р-типів виконують функції відповідно.

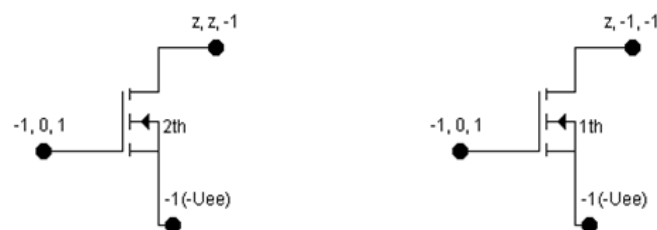


Рис.3.7 – МОН-транзистори, що здійснюють комутацію інформаційного сигналу "-1". Показано стан на стоку залежно від керуючих сигналів на затворі.

На рис. 3.7 показані транзистори, що здійснюють комутацію інформаційного сигналу "-1".

Транзистор із високим порогом 2th на рис.3.7 здійснює функцію $A^{-1}(1)$, а транзистор із низьким порогом 1th – функцію $A^{-1}(0,1)$. На підкладку МОН - транзисторів зазвичай подається зворотнзміщувальна напруга джерела живлення, що замикає перехід канал-підкладка. При цьому потрібно враховувати, що величина потенціалу підкладки щодо витоку (каналу) впливає на рівень граничної напруги пропорційно коефіцієнту впливу підкладки. У польових структурах типу КНС (кремній на сапфірі) та плівкових МОН-структурах питання підключення підкладки відпадає. Якщо будь-якому стані схеми не виникне прямого зміщення р-п-переходу закритого транзистора, здійснює комутацію нульового інформаційного сигналу, то підкладку допустимо з'єднувати з витоком.

Для умови симетричності схем природно вибрати рівні напруги джерел живлення обох полярностей рівними одне одному, тобто. $|+U_{dd}| = |-U_{ee}| = U_{ж}$. Тоді висока гранична напруга $U_{02} = 1,5 U_{ж}$, а низька гранична напруга $U_{01} = 0,5 U_{ж}$ у МОН-транзисторів, позначених відповідно 2th і 1th .

Трирівневі логічні схеми, синтезовані в роботі, були досліджені у програмному середовищі Multisim, зокрема, за допомогою процедури Parameter sweep (варіація параметрів) було побудовано передавальні характеристики розроблених схем, що підтверджують їхню працездатність.

Вибрані напруги двополярного джерела живлення схем, що моделюються: $+U_{dd} = 4 \text{ В}$, $-U_{ee} = -4 \text{ В}$ (що не є принциповим).

Для синтезу трирівневих схем у середовищі Multisim було створено таку бібліотеку моделей МОН-транзисторів:

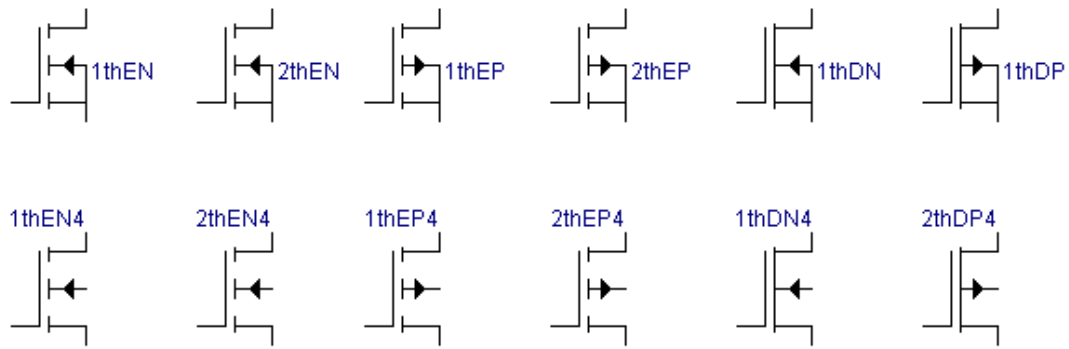


Рис. 3.8 – Бібліотека моделей МОН-транзисторів створена в середовищі Multisim

Основний параметр, яким вони розрізняються – це рівень порогового напруги:

1thEN – МОН-транзистор з індукованим n-каналом (Enhancement N-MOSFET) з граничною напругою $U_0 = 2\text{В}$;

2thEN – МОН-транзистор з індукованим n-каналом (Enhancement N-MOSFET) з граничною напругою $U_0 = 6\text{В}$;

1thEP – МОН-транзистор з індукованим p-каналом (Enhancement P-MOSFET) з граничною напругою $U_0 = -2\text{В}$;

2thEP – МОН-транзистор з індукованим p-каналом (Enhancement P-MOSFET) з пороговою напругою $U_0 = -6\text{В}$;

1thDN – МОН-транзистор із вбудованим n-каналом (Depletion N-MOSFET) з пороговою напругою $U_0 = -2\text{В}$;

1thDP – МОН-транзистор із вбудованим p-каналом (Depletion P-MOSFET) з граничною напругою $U_0 = 2\text{В}$.

Інші транзистори відрізняються від перерахованих (нижній ряд на рис. 3.8) лише відсутністю з'єднання між контактами підкладки та витоку.

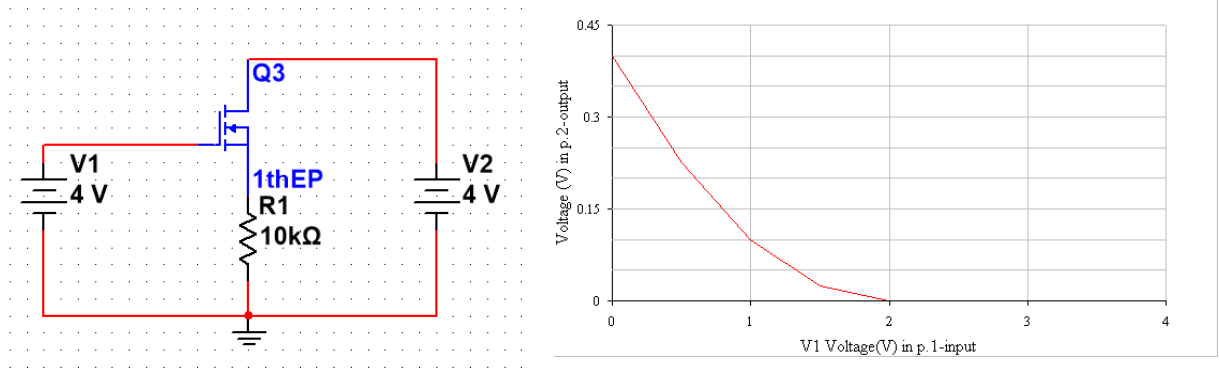


Рис. 3.9 – Сток-затворна характеристика МОН-компонента 1thEP.

У програмі Multisim було змодельовано вимірювання сток-затворних характеристик МОН-компонентів із створеної бібліотеки. На рис.3.9 -3.14 показані схеми зі зняттям характеристик та графіки, побудовані програмою даних схем за допомогою процедури Parameter sweep електронного моделювання, що дозволяє “вимірювати” напругу в будь-якій обраній точці схеми при варіюванні напруги будь-якого джерела живлення схеми в заданих межах .

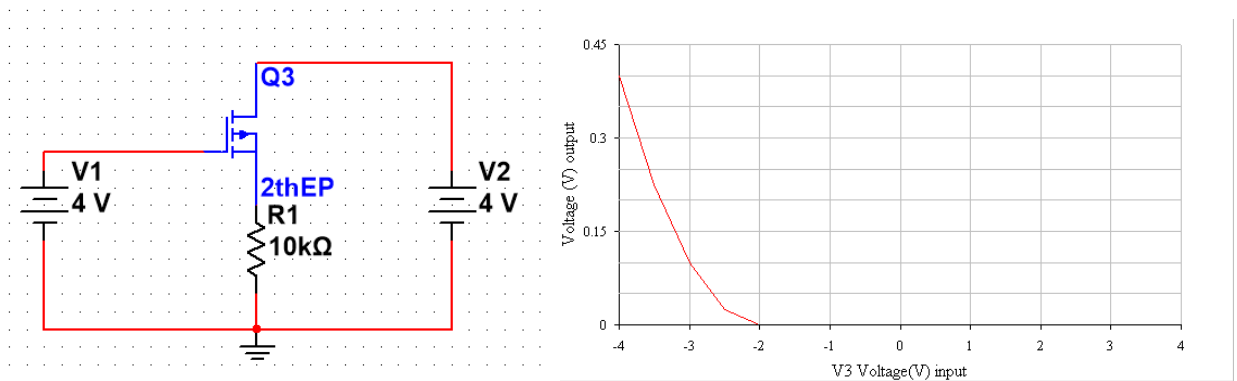


Рис. 3.10 – Сток-затворна характеристика МОН-компонента 2thEP.

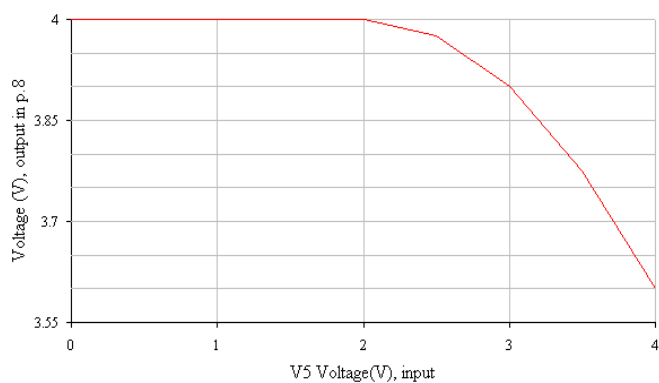
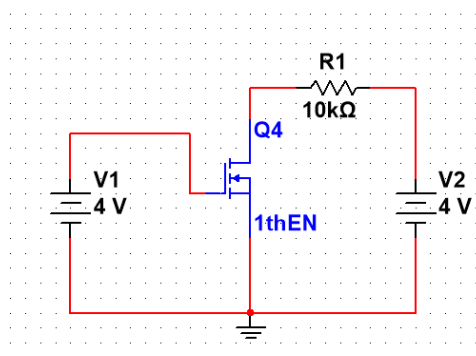


Рис. 3.11 – Сток-затворна характеристика МОН-компонента 1thEN.

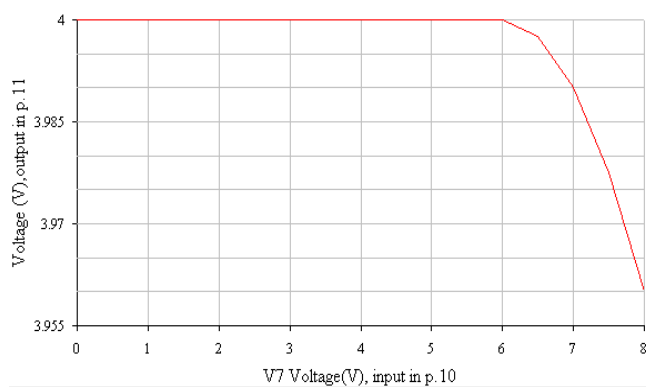
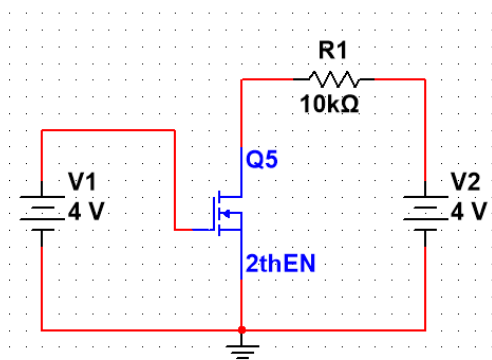


Рис. 3.12 – Сток-затворна характеристика МОН-компонента 2thEN

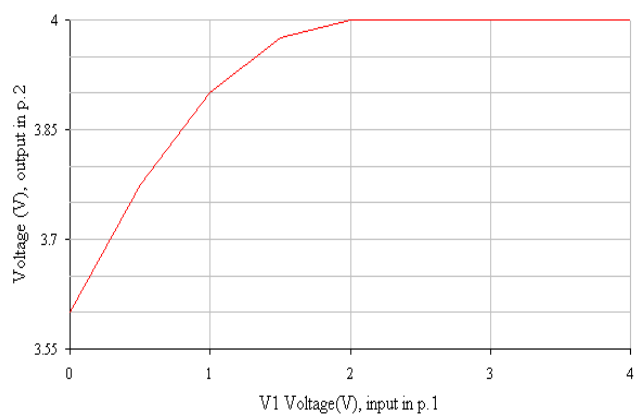
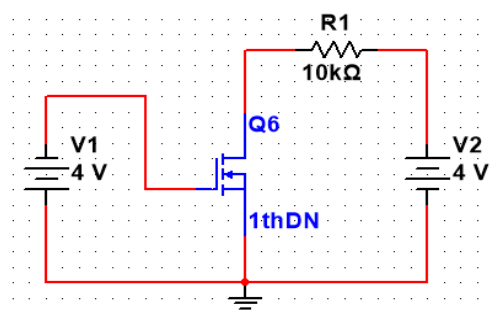


Рис. 3.13 – Сток-затворна характеристика МОН-компонента 1thDN

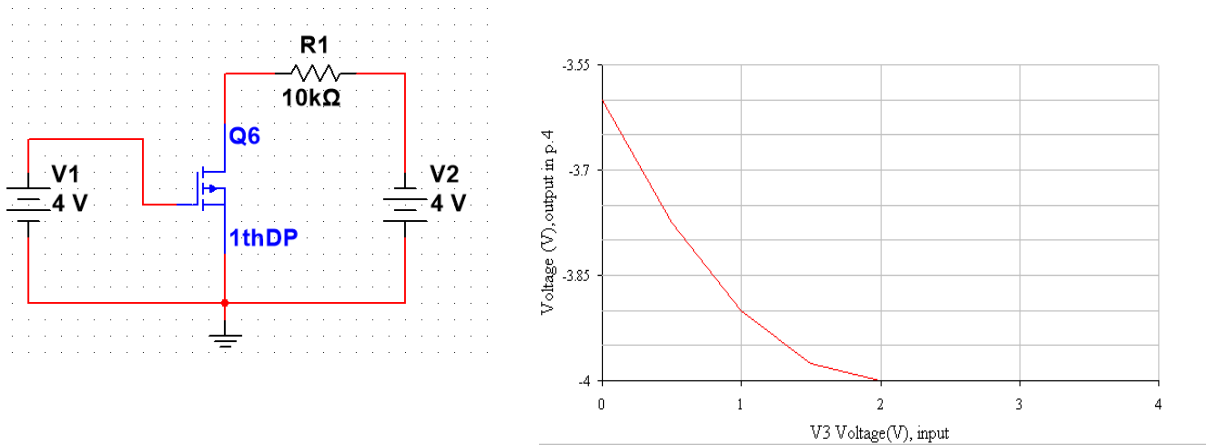


Рис. 3.14 – Сток-затворна характеристика МОН-компонента 1thDP.

При використанні процедури *Parameter sweep* можна задавати будь-яку дискретність збільшення параметра, що варіюється. При зменшенні кроку збільшення збільшується точність розрахунку графіка та час, що витрачається програмою на схемотехнічне моделювання.

У попередньому підрозділі були введені логічні функції базових компонентів трирівневих структур виду $Ax(y)$, де A - вхід для керуючого сигналу компонента або їх з'єднання (затвор або з'єднання затворів МОН-транзисторів), x - інформаційний сигнал на початку МОН-транзистора з безлічі $\{-1, 0, 1\}$, y - безліч керуючих сигналів, при подачі яких на вхід (затвор) компонента. A останній переходить у відкритий стан, комутуючи інформаційний сигнал на вихід (стік) МОН-компонента або групи з'єднання компонентів. У випадку, якщо сигнал, що управляє, не належить безлічі y , компонент знаходиться в непровідному стані (z -стані). Введена логічна функція подібна до узагальнених характеристичних функцій тризначної логіки.

Послідовне з'єднання транзисторів відповідає кон'юнкції інформаційного сигналу і сигналів керуючих, поданих на затвори МОН-транзисторів, тому ланцюг послідовно з'єднаних транзисторів може бути еквівалентно представлена одним транзистором, на вхід якого подається вхідний сигнал, відповідний кон'юнкції вихідних функцій. Паралельне поєднання компонентів здійснює диз'юнкцію функцій. З'єднання стоків

транзисторів, що виконують комутацію різних інформаційних сигналів в одну точку, може бути формально представлено як диз'юнкція функцій різних інформаційних сигналів, при цьому необхідно дотримання умови присутності на виході лише одного з інформаційних сигналів у разі кожного з сигналів, що управляють, тобто. вимкнути змагання інформаційних сигналів.

Таким чином, щоб синтезувати будь-яку тризначну логічну функцію, необхідний повний набір дев'яти базових функцій, наведений у табл.3.1, для всіх комбінацій інформаційного та сигналів, що управляють.

Таблиця 3. 1

Повний набір базових функцій.

Інформаційний сигнал	Керуючий сигнал		
	-1	0	1
-1	$A^{-1}(-1)$	$A^{-1}(0)$	$A^{-1}(1)$
0	$A^0(-1)$	$A^0(0)$	$A^0(1)$
1	$A^1(-1)$	$A^1(0)$	$A^1(1)$

З даних функцій базовими компонентами здійснюються такі 4 функції: : $A^{-1}(1)$, $A^0(-1)$, $A^0(1)$, $A^1(-1)$.

Крім них базовими компонентами виконуються ще такі 4 функції: $A^{-1}(0,1)$, $A^0(-1,0)$, $A^0(0,1)$, $A^1(-1,0)$, які комутують на вихід інформаційний сигнал при керуючому дорівнює одному з двох сигналів, визначених функцією. Дані базові функції дозволяють провести мінімізацію функцій, що синтезуються.

У таблиці 3.2 зведено функціональні характеристики базових компонентів з бібліотеки МОН-транзисторів, створеної серед програми Multisim для синтезу і моделювання трирівневих логічних схем.

Таблиця 3.2

Функції компонентів бібліотеки базових елементів середовища
моделювання Multisim

Тип елементу	Інформаційни й сигнал	Стан виходу при керуючому сигналі			Функція елементу
		-1	0	1	
1thEP	+1	+1	+1	z	$A^1(-1,0)$
2thEP	+1	+1	z	z	$A^1(-1)$
1thDN	0	z	0	0	$A^0(0,1)$
1thDP	0	0	0	z	$A^0(-1,0)$
1thEN	0	z	z	0	$A^0(1)$
1thEP	0	0	z	z	$A^0(-1)$
1thEN	-1	z	-1	-1	$A^{-1}(0,1)$
2thEN	-1	z	z	-1	$A^{-1}(1)$

3.2 Розробка та моделювання роботи елементів трійкової логіки

Розробку почнемо з створення трирівневого інвертора. Він повинен виконувати функцію інверсії тризначної логіки чи так звану операцію діаметрального заперечення, тобто. значення “ІСТИНА”(+1) він повинен переводити у значення “БРЕХНЯ”(-1) і навпаки, а значення “НЕВИЗНАЧНІСТЬ” передавати без зміни. Отже, йому вірна наступна таблиця істинності.

Таблиця 3.3

Таблиця істинності інвертора.

x	-1	0	1
$\overline{x}$	1	0	-1

Розширена логічна функція, що відповідає таблиці істинності інвертора і складається з диз'юнкцій базових функцій, записується так:

$$\overline{x} = x -1 (1) \vee x 0 (0) \vee x 1 (-1) \quad (3.1)$$

Базовими МОН-компонентами виконуються функції $x-1(1)$ (компонентом 2thEN з бібліотеки елементів Multisim) та $x 1 (-1)$ (компонентом 2thEP). Базова функція $x 0 (0)$ є кон'юнкцією двох функцій МОН-транзисторів із вбудованими n та p каналами:

$$x 0 (0) = x 0 (-1,0) \& x 0 (0,1) \quad (3.2)$$

Ця кон'юнкція відповідає послідовному з'єднанню МОН-транзисторів 1thDN і 1thDP бібліотеки елементів.

В результаті ми маємо схемотехнічну формулу для реалізації інвертора:

$$x = x -1 (1) \vee x 0 (-1,0) \& x 0 (0,1) \vee x 1 (-1) \quad (3.3)$$

За схемотехнічною формулою (3.3) здійснюється безпосередній синтез інвертора із МОН-транзисторів базової бібліотеки. На рис. 3.15 наведена схема інвертора, синтезована за даною схемотехнічною формулою.

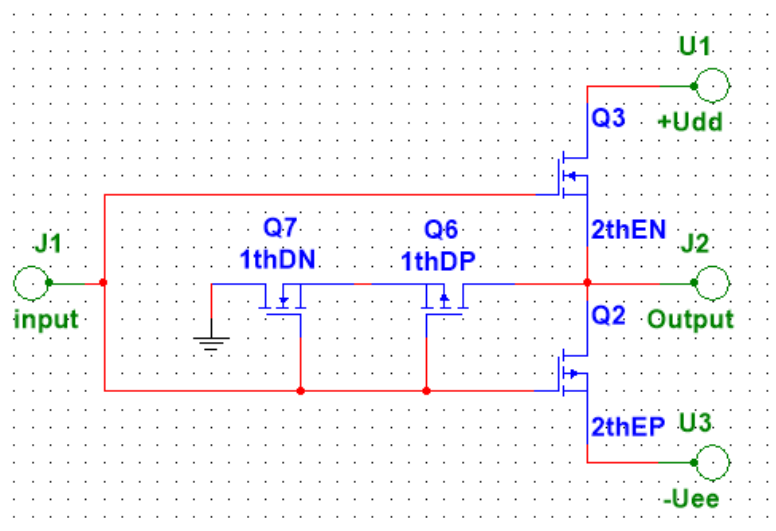


Рис. 3.15 – Схема інвертора, синтезованого за схемотехнічною формулою. Схема скопійована із середовища схемотехнічного моделювання Multisim

Розглянемо роботу трирівневого інвертора. На рис. 3.16 наведені вольтамперні характеристики транзисторів, на яких реалізований інвертор. При зміні вхідного сигналу від U_{ee} до 0 у протифазі працюють транзистори 2thEP і 1thDN, перший при цьому закривається, а другий відкривається, забезпечуючи зміну вихідного сигналу з U_{dd} до 0. У процесі зміни сигналу транзистор 1thDP відкритий, а транзистор 2thEN закритий. При вхідному сигналі, що дорівнює 0, транзистори з індукованими каналами 2thEP і 2thEN закриті, а транзистори з вбудованими каналами 1thDN і 1thDP відкриті, і на виході є сигнал 0. При зміні вхідного сигналу від 0 до $+U_{dd}$ у протифазі вже працюють транзистори 1thDP і 2thEN, при цьому перший закривається, а другий відкривається і сигнал на виході змінюється з 0 до $-U_{ee}$. В результаті передавальна характеристика трирівневого інвертора матиме вигляд, показаний на рис.3.16 б. Підкладки транзисторів з вбудованим каналом 1thDN і 1thDP в цій схемі необхідно з'єднувати або з їх витоками, або з шинами, що живлять $+U_{dd}$ для р-канального транзистора та $-U_{ee}$ для n-канального. Якщо підкладка з'єднана з витоком, то при високому вихідному сигналі будь-якої полярності у транзистора з вбудованим каналом, який в цьому випадку закритий, рп перехід стік-підкладка зміщений активним високим вихідним сигналом. Низькопорогові компоненти 1th при моделюванні схеми інвертора в середовищі програми Multisim мають граничну напругу $|0,5U_{dd}|$, а високопорогові $|1,5U_{dd}|$ за модулем.

Робота схеми синтезованого трирівневого інвертора була змодельована серед програми схемотехнічного моделювання Multisim. За допомогою процедури Parameter sweep (варіація параметрів) програмою була побудована передатна характеристика схеми, що моделюється. На рис. 3.17 наведено схему моделі з джерелами живлення V5, V6 та джерелом вхідного сигналу V7, напруга якого в процесі моделювання змінюється від $-U_{ee}$ до $+U_{dd}$. На рис.3.18 наведено передатну характеристику інвертора, яка скопійована із середовища програми Multisim.

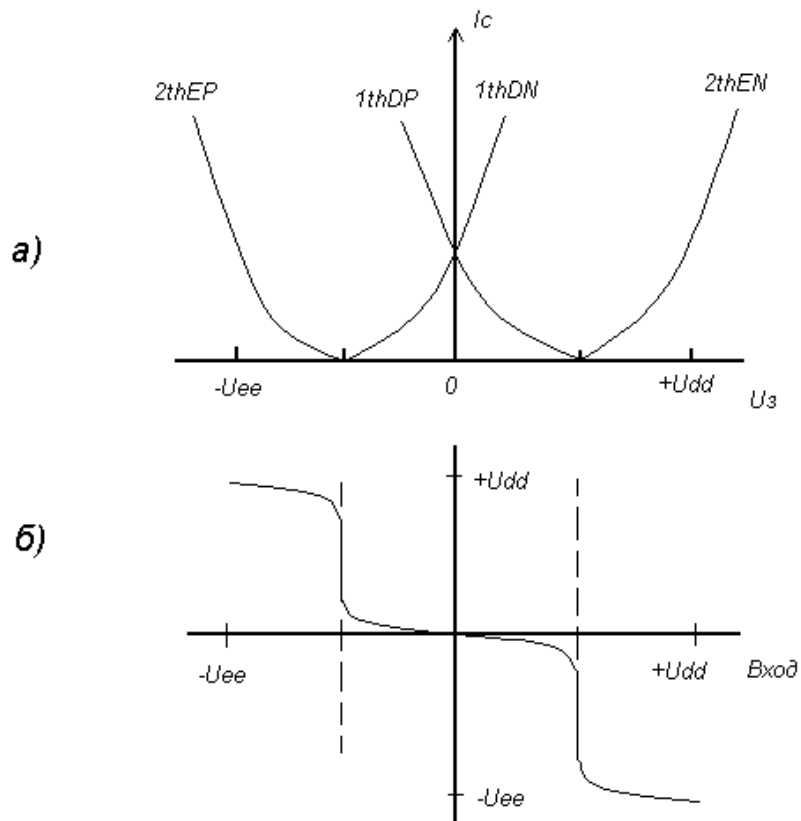


Рис. 3.16 – Вольтамперні характеристики МОН-компонентів інвертора (а) та його передатна характеристика (б).

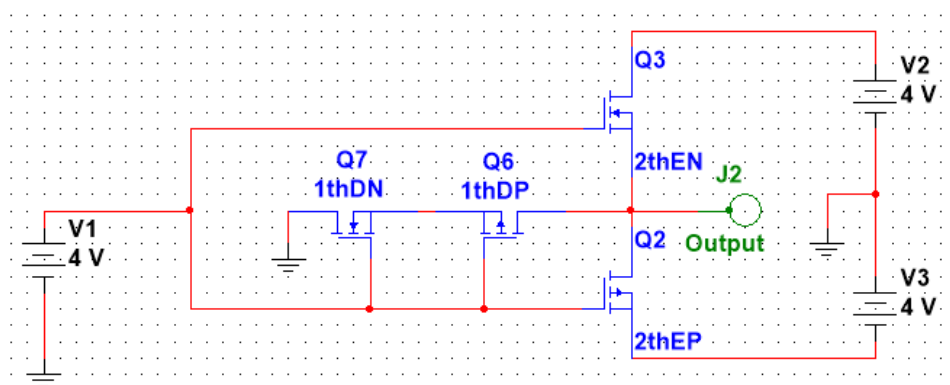


Рис. 3.17. – Схема моделювання трирівневого інвертора для зняття його передавальної характеристики та вимірювання параметрів у середовищі програми Multisim.

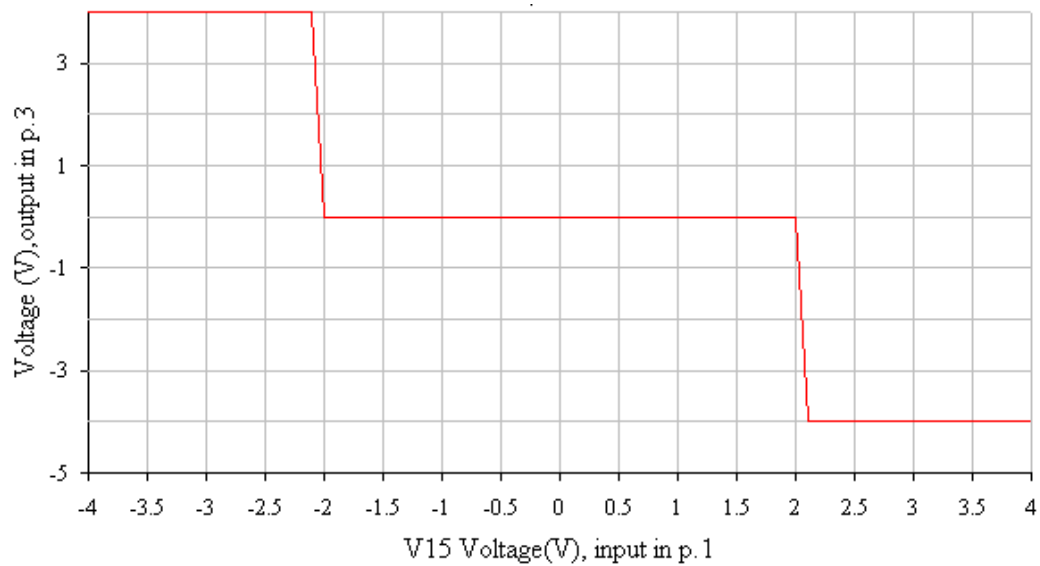


Рис. 3.18 – Передавальна характеристика інвертора.

Серед програми Multisim було змодельовано осцилографічне дослідження схеми інвертора.

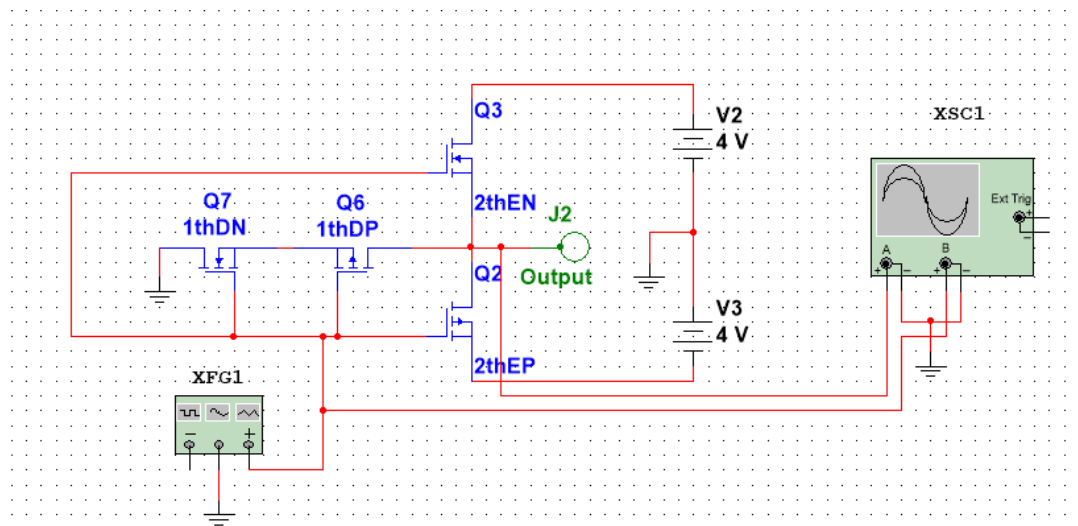


Рис. 3.19 – Дослідження роботи схеми трирівневого інвертора у середовищі програми Multisim за допомогою віртуальних генераторів та двопробного осцилографа.

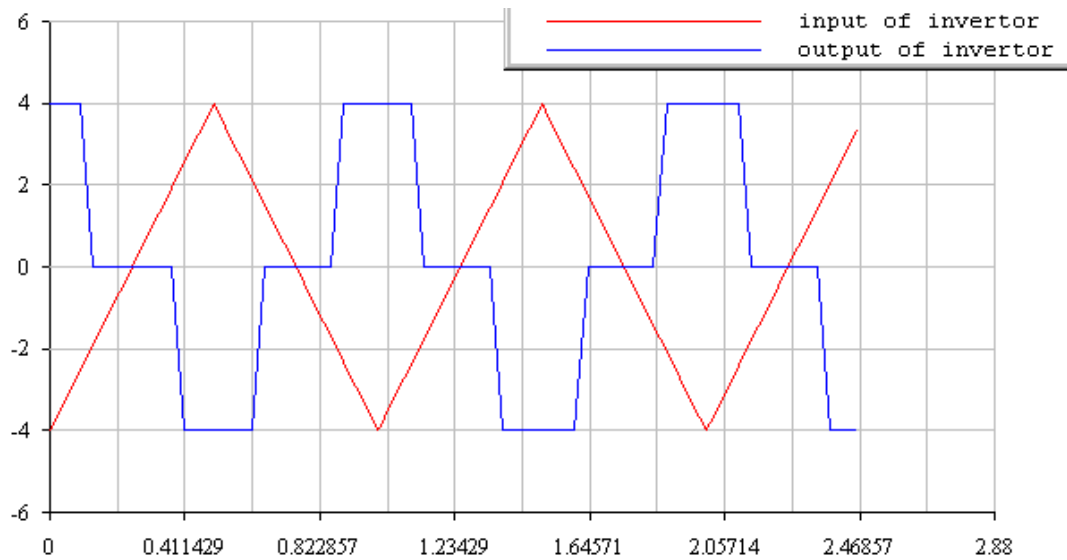


Рис. 3.20 – Осцилограми вхідного (червоний промінь) та вихідного (синій промінь) сигналів трирівневого інвертора

На схемі рис.3.20 При лінійному зміні вхідного сигналу видно чітке перемикання рівнів вихідного сигналу інвертора.

Для синтезу схем, що виконують функції тризначної логіки, важливу роль відіграють характеристичні функції. У тризначній логіці три характерні функції. Їхні таблиці істинності наведені в таблиці 3.4.

Таблиця 3.4

Характеристичні функції.

x	$\varphi_{-1}(x)$	$\varphi_0(x)$	$\varphi_1(x)$
-1	1	-1	-1
0	-1	1	-1
1	-1	-1	1

Характеристична функція $\varphi_{-1}(x)$ просто реалізується на МОН транзисторах з індукованим каналом. Розширена логічна формула цієї функції записується так:

$$\varphi_1(x) = x1(-1) \vee x-1(0,1) \quad (3.4)$$

За формулою (3.4) проводиться схемотехнічна реалізація характеристичної функції $\varphi_1(x)$. На рис. 3.21 наведено синтезовану схему, що виконує характеристичну функцію, скопійовану з середовища схемотехнічного моделювання Multisim

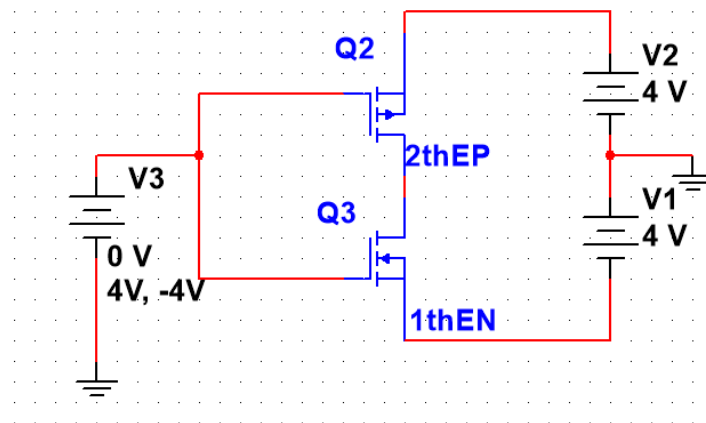


Рис. 3.21 – Схема, що виконує характеристичну функцію $\varphi_1(x)$.

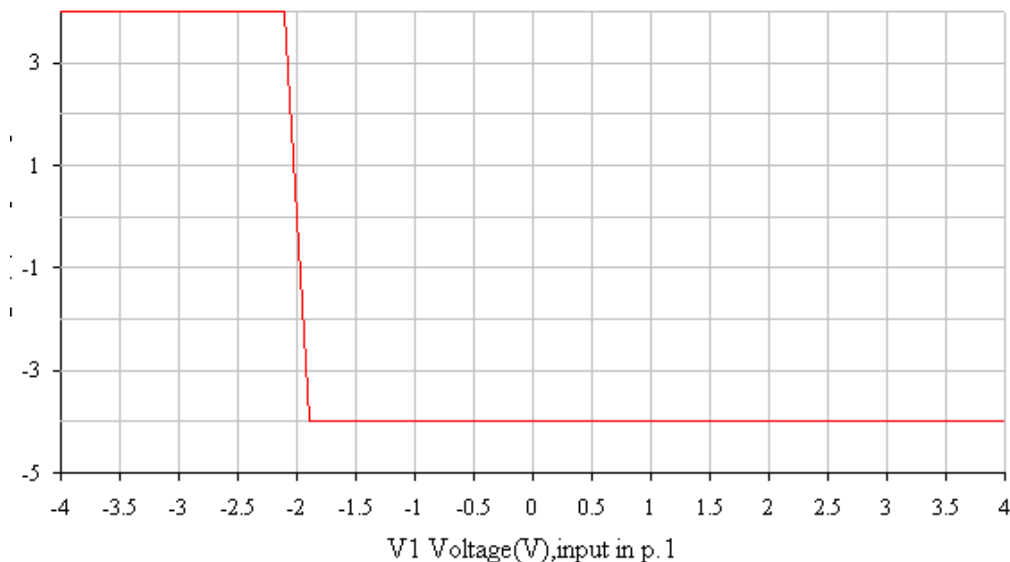


Рис. 3.22 – Передатна характеристика схеми, що зображена на рис. 3.21 і виконує характеристичну функцію $\varphi_1(x)$.

Характеристика побудована при моделюванні схеми в середовищі Multisim за допомогою процедури Parameter sweep і відображає зміну вихідної напруги в схемі у точці з'єднання транзисторів в залежності від вхідної.

На рис.3.22 наведено передатну характеристику синтезованої схеми. Дана характеристика розраховувалася програмою для зміни напруги джерела вхідного сигналу V1 схеми на рис.3.21 від -4 до +4 В при кроці дискретного збільшення 0,1 В. Графік на рис.3.22 показує чітке перемикання вихідних сигналів і неспотвореність їх рівнів, а так а відповідність роботи схеми таблиці істинності характеристичної функції $\varphi_1(x)$.

Також на МОН-компонентах досить просто реалізується інверсія характеристичної функції $\neg\varphi_1(x)$. Розширена логічна формула інверсії цієї функції записується так:

$$\neg\varphi_1(x) = x1(-1,0) \vee x-1(0) \quad (3.5)$$

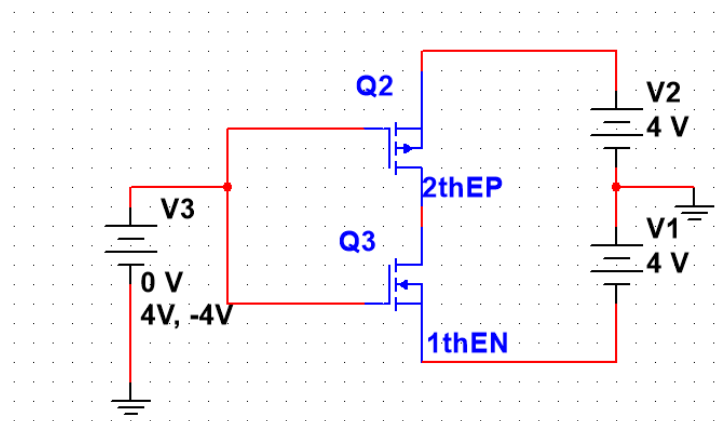


Рис. 3.23 – Схема, синтезована за формулою (3.2)

Синтезована схема, що виконує інверсію характеристичної функції $\neg\varphi_1(x)$. Транзистор Q2, виділений червоним кольором, має високий поріг $U_0 = 6\text{В}$ по модулю (2thEN), транзистор Q4 має порогову напругу $U_0 = 2\text{В}$ по модулю (1thEP).

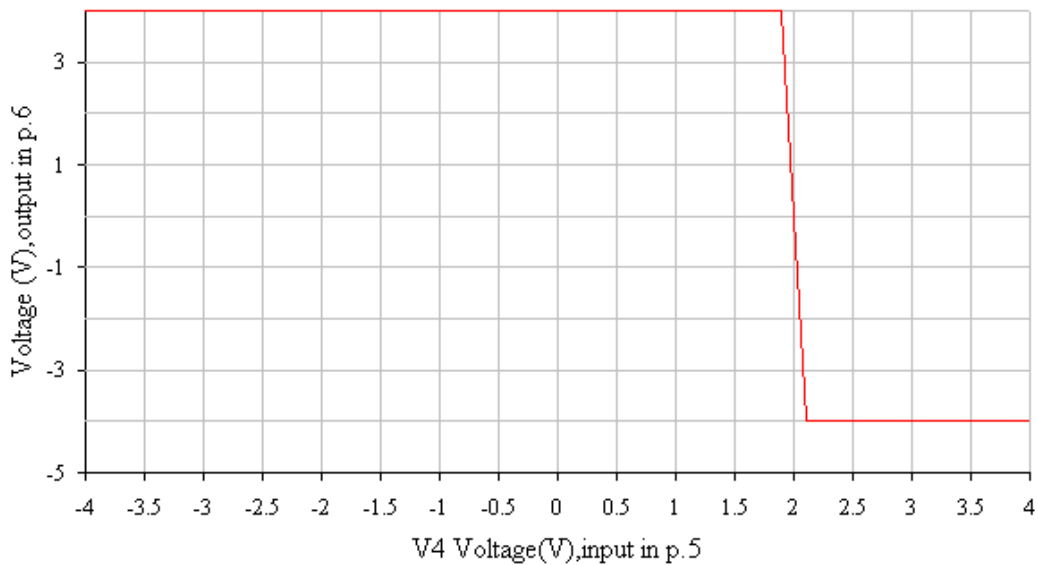


Рис. 3.23 – Передатна характеристика схеми, зображеної на рис.21 та виконує інверсію характеристичної функції $\neg\varphi_1(x)$.

На осі абсцис – зміна вхідної напруги , на осі ординат – вихідна напруга у точці схеми, наведеної на рис.3.23. Характеристика побудована серед програми Multisim при кроці дискретного збільшення вхідної напруги $0,1\text{ В}$.

За допомогою схеми, що виконує інверсію максимальних активних сигналів $+1$ і -1 , та схеми виконує функцію $\neg\varphi_1(x)$, можна отримати характеристичну функцію $\varphi_1(x)$, проінвертувавши вихідний сигнал схеми, зображеної на рис.3.24.

Функцію інверсії максимальних активних сигналів може виконувати будь-яка із схем, наведених на рис.3.21 та рис.3.23.

Для синтезу трирівневих пристроїв, що виконують функції тризначної логіки, особливий інтерес має характеристична функція нуля. $\varphi_0(x)$. Її розширена логічна формула, що складається з диз'юнкцій базових логічних функцій, записується так:

$$\varphi_0(x) = x-1(1) \vee x1(0) \vee x-1(-1) \quad (3.6)$$

У розширеній логічній формулі (3.6) дві останні базові логічні функції $x1(0)$ і $x-1(-1)$ не реалізуються безпосередньо будь-яким із базових компонентів трирівневих структур. Ці логічні функції можна реалізувати на основі суперпозиції базових компонентів.

Синтез трирівневої схеми, що виконує характерну функцію $\varphi_0(x)$, буде вироблений після доповнення до набору базових елементів, що виконують базові логічні функції, схемами, що реалізують базові функції, що залишилися $A-1(-1)$, $A-1(0)$, $A1(0)$, $A1(1)$.

Повний набір базових функцій дозволить синтезувати будь-яку трирівневу схему, яка виконує будь-яку функцію тризначної логіки і залежить від довільного числа змінних.

Якщо із затвором елемента 1thEP, що виконує базову функцію $A1(-1,0)$ або елемента 2thEP, що виконує функцію $A1(-1)$, з'єднати вихід схеми, що здійснює інверсію характеристичної функції $\neg\varphi_1$, то отримаємо схему, яка виконує базову логічну функцію $A1(1) = A1(\neg\varphi_1(A)) = -1$. Ця схема наведена на рис.3.24

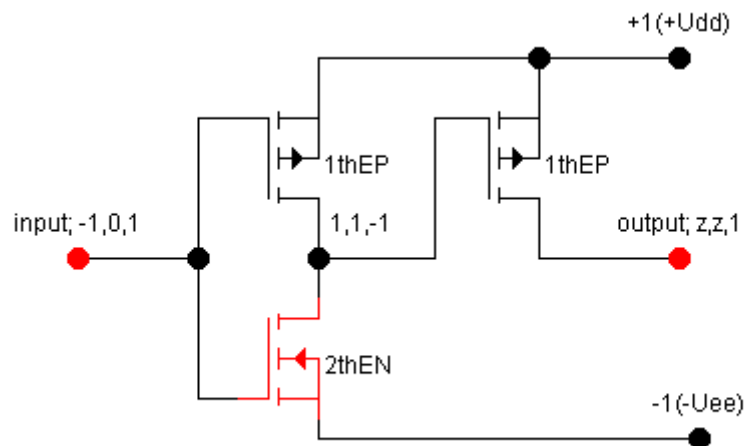


Рис. 3.24 – Схема, що реалізує логічну базову функцію $A1(1)$. При вхідному сигналі, що дорівнює “+1”, на виході є також сигнал “+1”. За інших вхідних сигналах вихід схеми перебуває у z-стані.

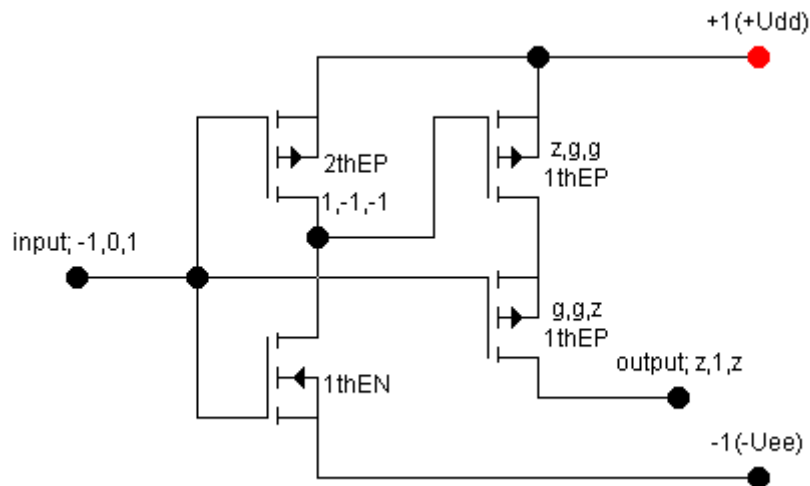


Рис. 3.26 – Схемотехнічна реалізація базової логічної функції $A1(0)$.

При нульовому входньому сигналі на виході є “1”, в інших випадках вихід знаходиться в z-стані. Символ “g” означає відкритий стан каналу.

Ця кон'юнкція реалізується послідовним з'єднанням схем, що виконують функції $A1(0,1)$ та $A1(-1,0)$. Схема, що реалізує базову функцію $A1(0)$, наведено на рис.3.26.

Якщо вихід схеми виконує характеристичну функцію $\varphi -1$ з'єднати з затвором n-канального транзистора 1thEN, що виконує базову логічну функцію $A-1(0,1)$, або транзистора 2thEN, що виконує базову логічну функцію $A-1(1)$, то дана схема буде реалізовувати базову логічну функцію $A-1(-1) = A-1(\varphi-1(A) = 1$, причому виходом її є сток даного транзистора. Схема, що реалізує цю функцію, наведено на рис.3.27.

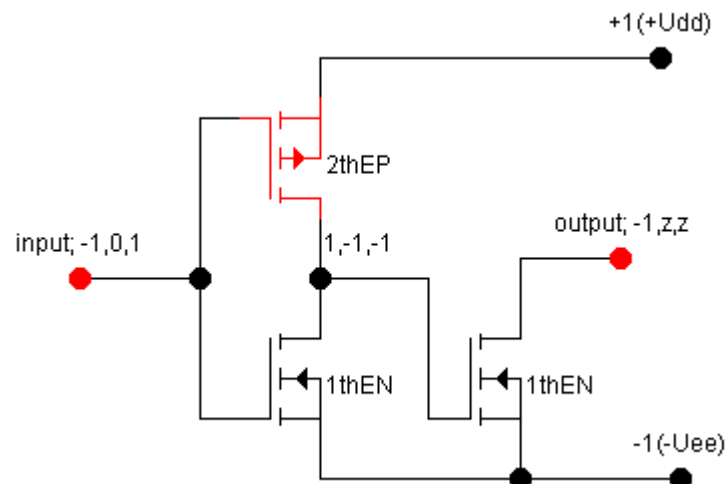


Рис. 3.27 – Схемотехнічна реалізація базової логічної функції $A-1(-1)$. На виході схеми є “-1”, коли на вході також сигнал “-1”. Інакше вихід перебуває у z-стані.

Якщо реалізації базової функції на рис.3.27 замість частини схеми, виконує характеристичну функцію $\varphi -1$, застосувати схему, що виконує інверсію характеристичної функції $\neg\varphi_1$, то в цьому випадку схемотехнічно буде реалізована додаткова базова логічна функція $A-1(-1,0) = A-1(\neg\varphi_1(A) = 1$. Схема, що реалізує додаткову базову функцію $A-1(-1,0)$, наведено на рис.30.

Базова функція $A-1(0)$ є кон'юнкцією додаткових базових функцій $A-1(0,1)$ та $A-1(-1,0)$:

$$A-1(0) = A-1(0,1) \& A-1(-1,0) \quad (3.8)$$

Ця кон'юнкція здійснюється послідовним з'єднанням відповідних схемних реалізацій, що показано на рис.31.

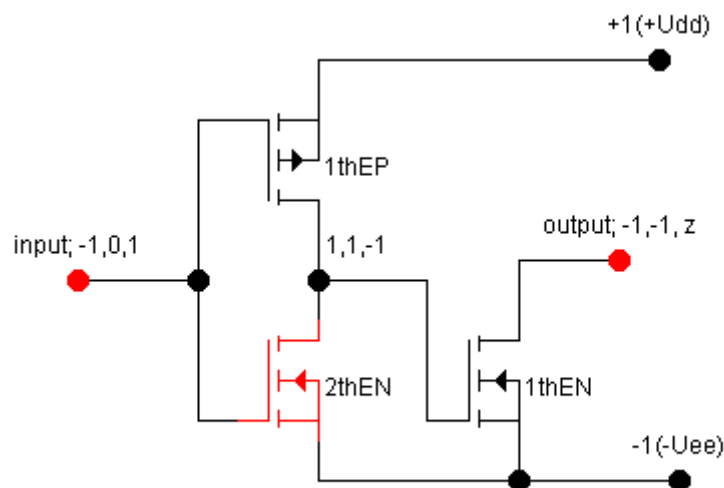


Рис. 3.28 – Схемотехнічна реалізація додаткової базової функції $A-1(-1,0)$. На виході схеми є сигнал “-1”, якщо на вході схеми сигнали “-1” або “0”, інакше вихід знаходиться в z-стані.

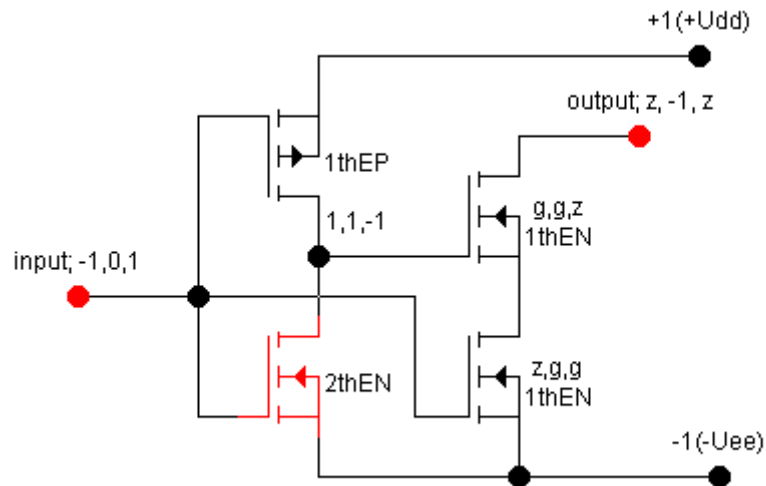


Рис. 3.29 – Схемотехнічна реалізація базової функції $A-1(0) = A-1(0,1) \& A-1(-1,0)$. На виході схеми є “-1”, якщо на вході сигнал “0”, інакше вихід знаходиться в z-стані.

Таким чином, ми маємо повний набір 9 базових функцій, які дозволяють синтезувати будь-яку тризначну логічну функцію:

$$A-1(-1), A-1(0), A-1(1), A0(-1), A0(0), A0(1), A1(-1), A1(0), A1(1). \quad (3.9)$$

Крім того, ми маємо набір додаткових базових функцій, що дозволяють мінімізувати схеми, що синтезуються:

$$A1(-1,0), A1(0,1), A0(0,1), A0(-1,0), A-1(0,1), A-1(-1,0). \quad (3.10)$$

Зробимо синтез характеристичної функції $\varphi_0(x)$. Її розширена логічна формула записується в такий спосіб:

$$\varphi_0(x) = x-1(1) \vee x-1(0) \vee x-1(-1) \quad (3.11)$$

Вона складається із диз'юнкцій базових функцій. Схемотехнічна реалізація характеристичної функції допускає мінімізацію, оскільки у двох

членах диз'юнкції $x_1(0)$ та $x_{-1}(-1)$ використовується загальна характеристична функція $\varphi_{-1}(x)$:

$$x_1(0) = x_1(0,1) \& x_{-1}(-1,0) = x_1(\varphi_{-1}(x) = -1) \& x_{-1}(-1,0) \quad (3.12)$$

$$x_{-1}(-1) = x_{-1}(\varphi_{-1}(x)) = 1 \quad (3.13)$$

З (3.11) (3.12) та (3.13) отримуємо схемотехнічну формулу:

$$\varphi_0(x) = x_{-1}(1) \vee x_1(\varphi_{-1}(x) = -1) \& x_{-1}(-1,0) \vee x_{-1}(\varphi_{-1}(x) = 1) \quad (3.14)$$

За цією схемотехнічною формулою здійснюємо синтез схеми, що реалізує характеристичну функцію $\varphi_0(x)$.

Синтезована схема наведена на рис.3.30 Ця схема скопійована із середовища програми схемотехнічного моделювання Multisim, куди було введено для побудови передавальної характеристики та перевірки працездатності. Передатна характеристика, зображена на рис.3.31, при кроці дискретного збільшення вхідної напруги 0,1В. Графік передавальної характеристики синтезованої схеми показує, що схема має чіткий поріг перемикання, неспотвореність рівнів вихідних сигналів і робота схеми відповідає таблиці істинності для характеристичної функції $\varphi_0(x)$.

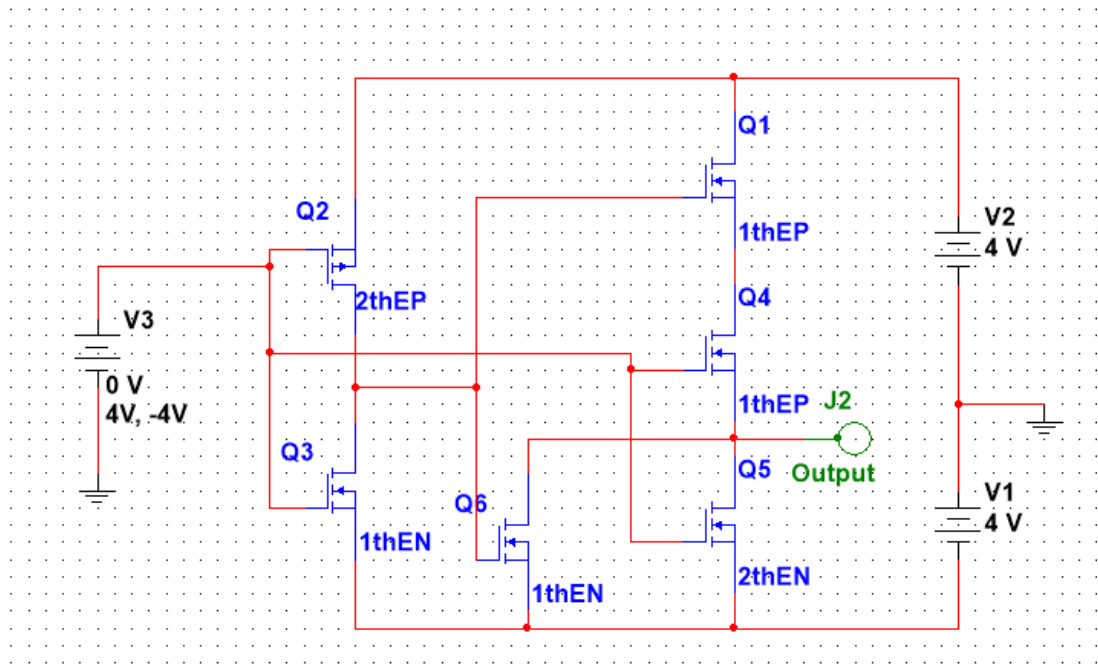


Рис. 3.30 – Схемотехнічна реалізація характеристичної функції $\varphi_0(x)$.

Схема змодельована в середовищі програми схемотехнічного моделювання Multisim для побудови передавальної характеристики.

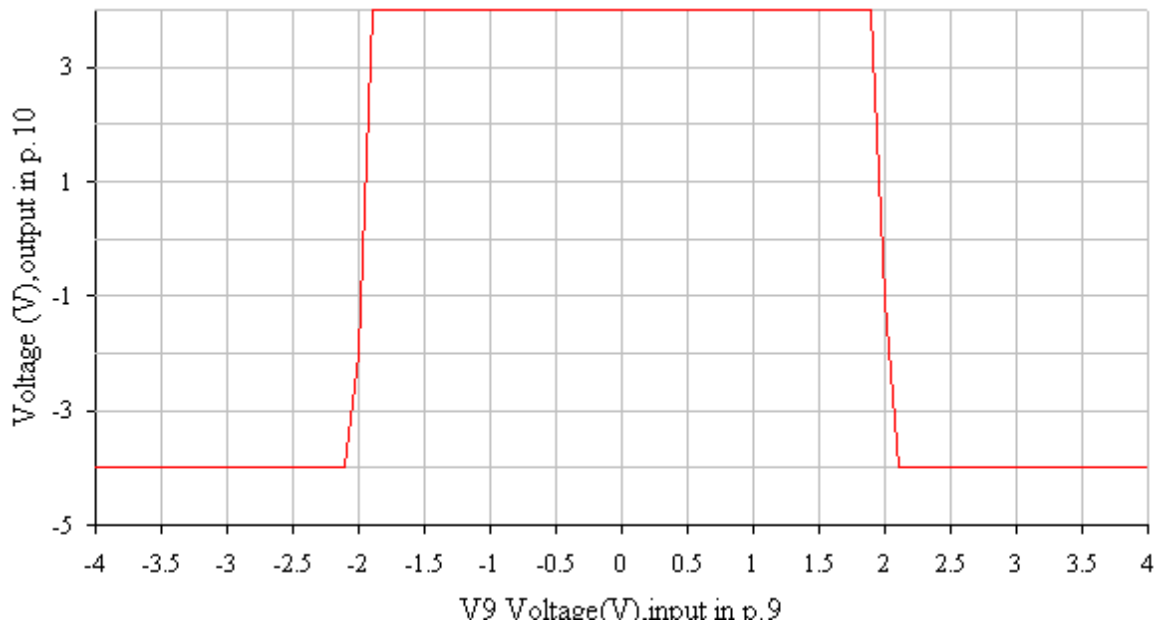


Рис. 3.31 – Передатна характеристика для схеми, зображеної на рис. 3.30 яка реалізує характеристичну функцію $\varphi_0(x)$.

У трирівневих пристроях, що розробляються, при переході з одного логічного стану в інший яка-небудь пара або група МОН-транзисторів працює в протифазі, забезпечуючи зміну логічних рівнів. Відомо, що й *n*-канальний транзистор передає інформаційний сигнал, негативний стосовно інформаційного сигналу, переданого комплементарним йому транзистором, тобто, якщо *n*-канальний транзистор забезпечує зміну сигналу з високого рівня на низький, цей інформаційний сигнал передається на стік без спотворення. Якщо ж *n*-канальний транзистор забезпечує зміну сигналу більш позитивний по відношенню до змінного (тобто. потенціал витоку транзистора позитивний по відношенню до потенціалу витоку ще до зміни логічних рівнів), то цьому випадку переданий інформаційний сигнал спотворюється на величину порогової напруги даного транзистора, тобто. вихідний сигнал буде меншим за інформаційний на величину порогової напруги. Те саме правильне для *p*-канальних МОН-транзисторів, якщо рівень аналізованих сигналів змінити на протилежний. Процес перемикання у випадках буде затяжним під час переходу через пороговий рівень.

Подібна ситуація може виникнути у трирівневих пристроях при комутації МОН-транзисторами нульового інформаційного сигналу. У цьому випадку комутуючий транзистор можна замінити на відповідну схему композицією базових компонентів.

Як приклад розглянемо можливі реалізації базової логічної функції $A0(-1)$, яка схемотехнічно здійснюється низькопороговим МОН-транзистором з індукованим *p*-каналом 1thEP з бібліотеки базових елементів середовища програмного моделювання електронних схем Multisim Цю базову функцію можна реалізувати, використовуючи схему, яка виконує характеристичну функцію $\varphi - I(x)$. Тоді як транзистор, комутує нульовий інформаційний сигнал, можна застосувати транзистори як з вбудованим, так і з індукованим *n*-каналом. Обидві реалізації наведено на рис. 3.32 та 3.35.

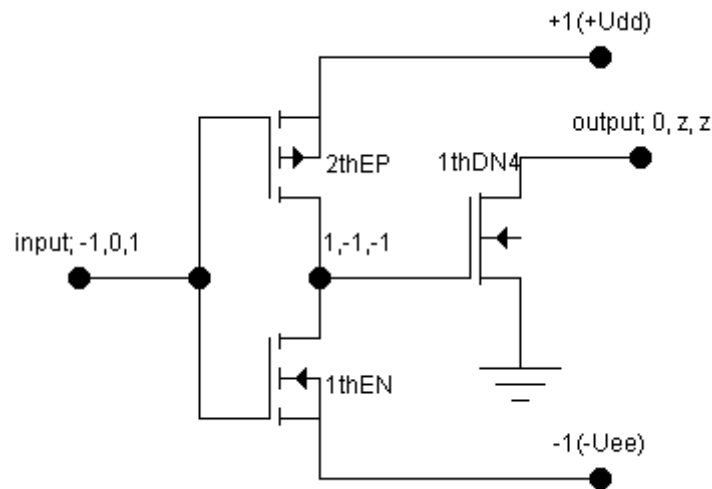


Рис. 3.32 – Реалізація базової логічної функції $A0(-1)$, де як компонент, що комутує нульовий інформаційний сигнал, використовується транзистор з вбудованим n-каналом.

Таким чином, виконання однієї і тієї ж базової логічної функції можна здійснити, використовуючи як нульовий сигнал, що комутує, транзистор як з каналом n-типу, так і р-типу - в залежності від знака протифазного інформаційного сигналу.

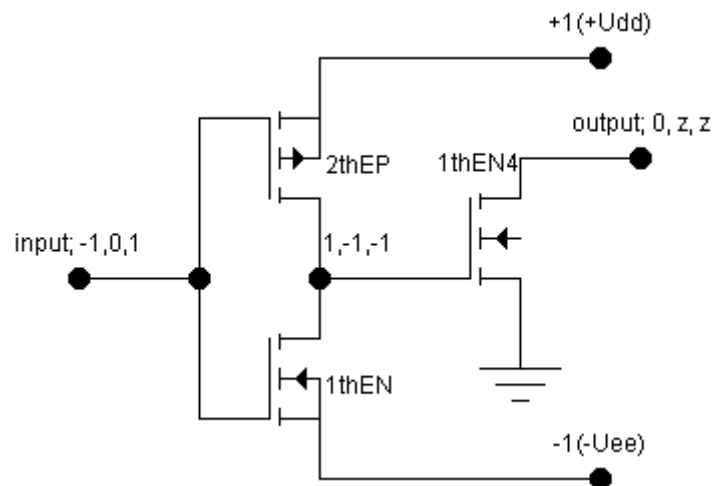


Рис. 3.33 – Схемотехнічна реалізація базової логічної функції $A0(-1)$

Схема з використанням транзистора з індукованим каналом краще, оскільки вона забезпечує гарантоване закривання каналу незалежно від потенціалу стоку. У разі використання транзистора з вбудованим каналом, як на рис.3.32, якщо він працюватиме на навантаження, підключене до негативного полюса джерела живлення, то нульовий інформаційний сигнал спотворюватиметься.

У трирівневих системах, що виконують функції тризначної логіки, важливу роль відіграє функція "модуль", яка є детектором певної події або повідомлення у тризначній логіці. При сигналі на вході функції модуля "ІСТИНА"(+1) або "БРЕХНЯ"(-1) на виході схеми є сигнал "ІСТИНА"(+1), якщо вхідний сигнал "0" – "НЕВИЗНАЧНІСТЬ", то на виході схеми також сигнал "0". Цю функцію можна реалізувати, використовуючи схему трирівневого інвертора та схему, що здійснює тризначну диз'юнкцію, оскільки функція модуля згідно із законом заперечення тризначної логіки (див. (3.11)) записується як

$$|x| = x \vee \neg x \quad (3.15)$$

Але з погляду апаратних витрат оптимально синтезувати схему, що виконує функцію модуля, безпосередньо з базових компонентів трирівневих пристроїв за таблицею істинності цієї функції. Тоді розширена логічна формула даної логічної функції записується так:

$$|x| = x1(-1) \vee x0(0) \vee x1(1) \quad (3.16)$$

За формулою (3.16) складається схемотехнічна формула:

$$|x| = x1(-1) \vee x0(-1,0) \& x0(0,1) \vee x1(\neg \varphi 1(x) = -1) \quad (3.17)$$

За схемотехнічною формулою здійснюється безпосередній синтез схеми базових компонентів тривірневих пристроїв.

Синтезована схема наведена на рис.3.34 для дослідження характеристик та перевірки працездатності. Передавальна характеристика схеми, що здійснює функцію модуля, зображено на рис.3.35. Крок дискретного збільшення вхідної напруги 0,1В. Схема має чіткий поріг перемикання та неспотвореність рівнів вихідних сигналів.

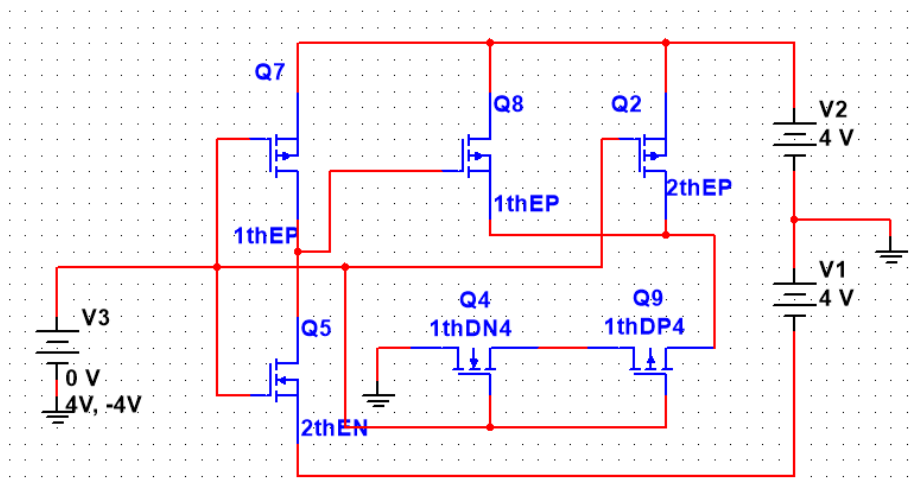


Рис. 3.34 – Синтезована схема, що виконує функцію модуля тризначної логіки x .

До входу схеми підключено джерело вхідного сигналу V1, напруга якого змінюється у діапазоні $(-4 \dots +4)$. В

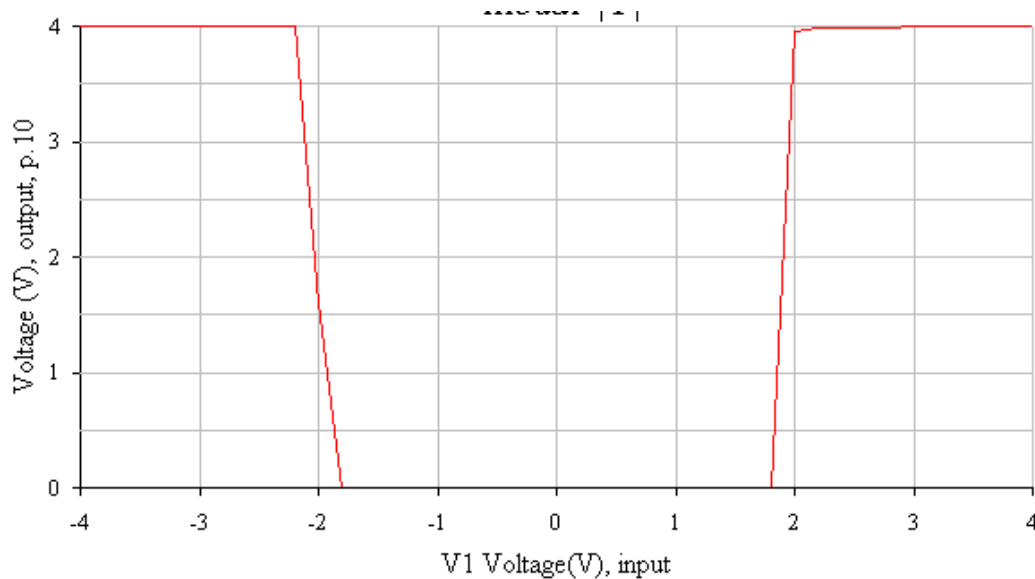


Рис. 3.35 – Передатна характеристика схеми.

Характеристика побудована в Multisim на осі абсцис – напруга, що варіюється, джерела вхідного сигналу схеми V1, на осі ординат – вихідна напруга схеми, зображеної на рис.3.34.

У синтезі трирівневих логічних схем, що виконують функції тризначної логіки, так і арифметичні операції в трійковій системі числення, виняткову роль відіграє функція циклічного заперечення тризначної логіки. Визначення цієї функції та її таблиця істинності наведено у таблиці в попередньому розділі. Зауважимо, що триразове циклічне заперечення аргументу в тризначній логіці дорівнює цьому аргументу. Часто при синтезі трирівневих логічних схем поряд із використанням функції циклічного заперечення виникає необхідність застосування функції подвійного циклічного заперечення. Для оптимізації з погляду апаратних витрат вигідніше використовувати дві різні схеми, що виконують обидві ці функції, ніж одну схему застосовувати два рази для виконання функції подвійного циклічного заперечення. Функцію одноразового циклічного заперечення можна назвати “правим циклом”, оскільки ряд істиннісних значень у таблиці істинності цієї операції циклічно зрушується у напрямі. Аналогічно функцію подвійного циклічного заперечення можна назвати "лівим циклом", оскільки ряд

істиннісних значень у разі циклічно зрушується у зворотному напрямі. Правий цикл позначається стрілкою праворуч над аргументом, лівий цикл, відповідно, - стрілкою в ліву сторону.

Здійснимо синтез трирівневих схем, що реалізують операції правого та лівого циклів тризначної логіки. У таблиці 3.5 наведено таблиці істинності даних функцій.

Таблиця 3.5

Таблиця істинності для циклічних заперечень.

x	$\rightarrow$ x	$\leftarrow$ x
-1	0	1
0	1	-1
1	-1	0

За таблицею істинності для правого циклу складаємо розширену логічну формулу:

$$\rightarrow$$

$$x = x0(-1) \vee x1(0) \vee x -1 \quad (3.18)$$

Усі члени диз'юнкції (3.18) є базовими логічними функціями, які є компонентами базового набору трирівневих структур, тому схема синтезується безпосередньо з розширеної логічної формули. Схемна реалізація функції:

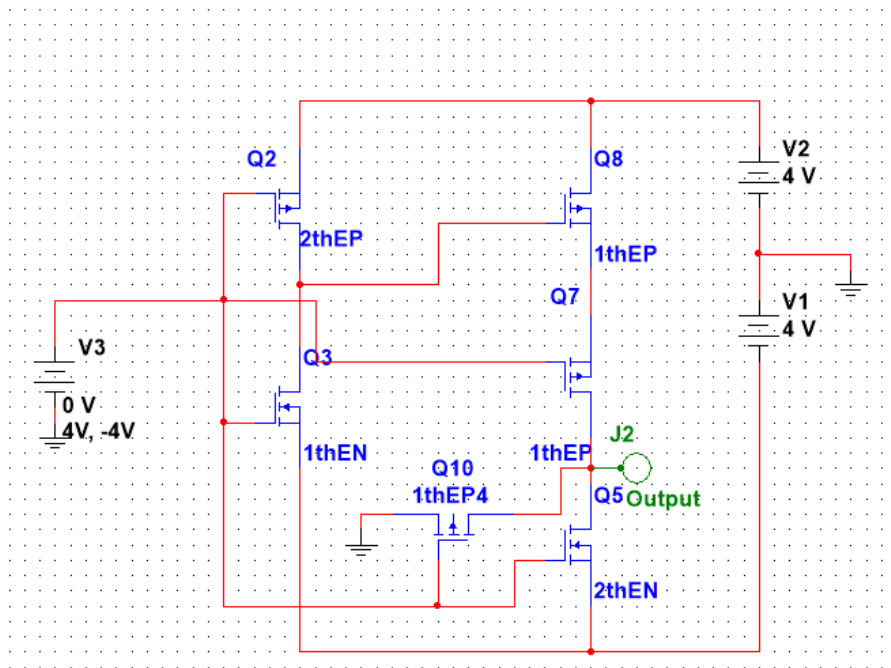


Рис. 3.36 – Схемотехнічна реалізація функції циклічного заперечення (правий цикл) тризначної логіки. Елементи Q1-Q3, Q5 виконують базову логічну функцію $x1(0)$, елемент Q4 – функцію $x0(-1)$, елемент Q6 – функцію $x-1(1)$.

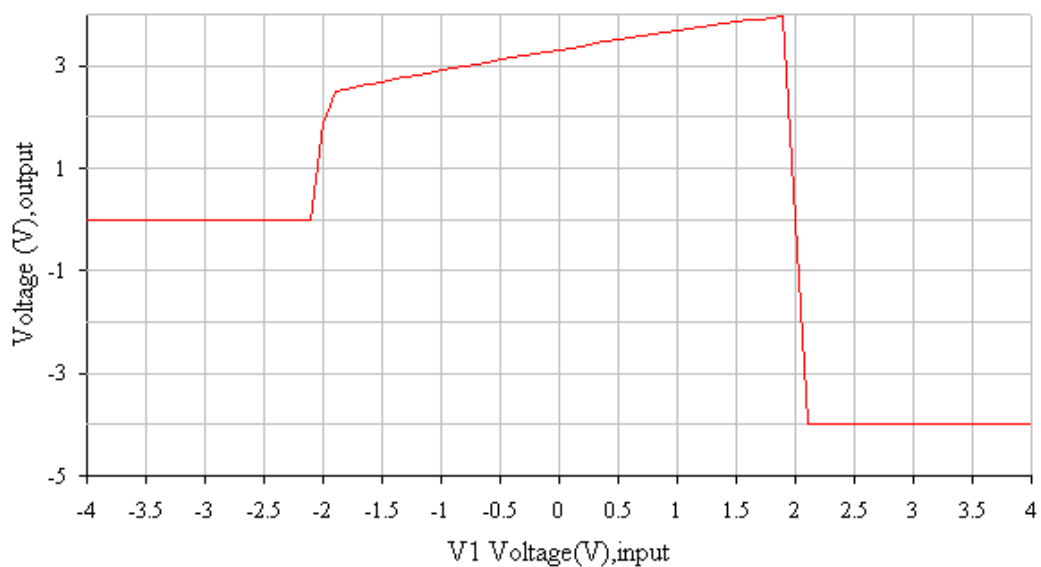


Рис. 3.37 – Передатна характеристика схеми, що здійснює функцію циклічного заперечення.

Характеристика побудована при кроці дискретного збільшення напруги 0,1В джерела вхідного сигналу V1 рис.3.36.

Наведена на рис. 3.37 передавальна характеристика синтезованої схеми, що здійснює функцію правого циклу, показує, що при перемиканні з нульового логічного рівня на рівень +1 відбувається спотворення інформаційного сигналу внаслідок того, що транзистор Q4 закривається не повністю, оскільки стік стає більш позитивним в процесі перемикання, ніж витік. На цій ділянці перехідної характеристики транзистор Q4 поводить себе як джерело струму, що працює на змінне навантаження із послідовно з'єднаних польових транзисторів Q2, Q3. Чим більш позитивним стає стік транзистора Q4 у процесі перемикання в міру відкривання транзисторів Q2, Q3, тим вище потрібна напруга на його затворі, необхідне замикання каналу.

Якщо подібне спотворення неприйнятне, то можна застосувати n-канальний транзистор як нульовий інформаційний сигнал, що комутує, з'єднавши його затвор з точкою з'єднання стоків транзисторів Q1 і Q5. Тоді дана частина схеми здійснюватиме функцію $x0(-1)$. В цьому випадку при перемиканні з нульового логічного рівня на рівень +1 на виході схеми спотворення інформаційних сигналів не буде, оскільки процес перекриття каналу в даному випадку не залежить від потенціалу стоку.

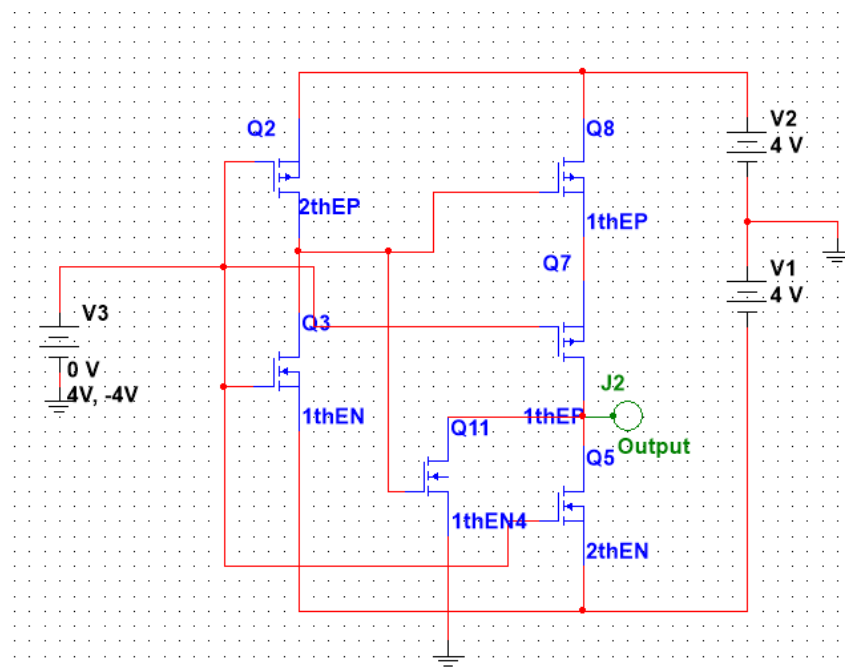


Рис. 3.38 – Модифікований варіант схемної реалізації функції правого циклу, наведеної на рис.3.37. Тут функцію $x0(-1)$ виконують транзистори Q23, Q24 та Q41.

На рис.3.39 наведена передатна характеристика схеми, побудована серед Multisim за допомогою процедури Parameter sweep для кроку дискретного збільшення вхідної напруги в 0,1В. Як видно з графіка, у цьому випадку дійсно не відбувається спотворення інформаційного сигналу при перемиканні схеми з одного стану до іншого.

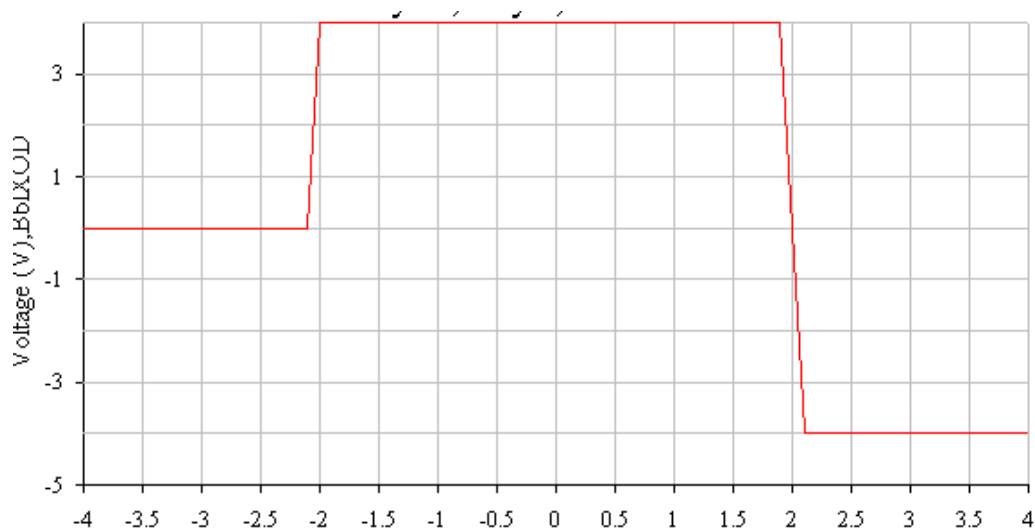


Рис. 3.39 – Передавальна характеристика схеми.

Здійснює операцію циклічного заперечення (правий цикл). Графік побудований програмою Multisim для цієї схеми. З графіка видно чітке перемикання схеми та неспотвореність рівнів вихідних сигналів.

Синтезуємо схему, що здійснює логічну операцію подвійного циклічного заперечення або лівий цикл. За таблицею істинності цієї функції (див. таблицю 3.5) складаємо розширену логічну формулу:

←

$$x = x1(-1) \vee x-1(0) \vee x0(1) \quad (3.19)$$

Щоб при перемиканні логічних станів у схемі, що синтезується, не було спотворень інформаційних сигналів, в ній базову логічну функцію $x0(1)$ слід також реалізувати композицією базових елементів, аналогічною застосованою у схемі правого циклу, де нульовий інформаційний сигнал комутується р-канальним МОН-транзистором з індукованим каналом. Схемотехнічна формула цієї базової логічної функції записується як:

$$x0(1) = x0(\neg\varphi1(x) = -1 \quad (3.20)$$

Схемотехнічна формула базової функції $x-1(0)$ записується як:

$$\begin{aligned} x-1(0) &= x-1(0,1) \& x-1(-1,0) = \\ &= x-1(0,1) \& x-1(\neg\varphi1(x) = 1) \end{aligned} \quad (3.21)$$

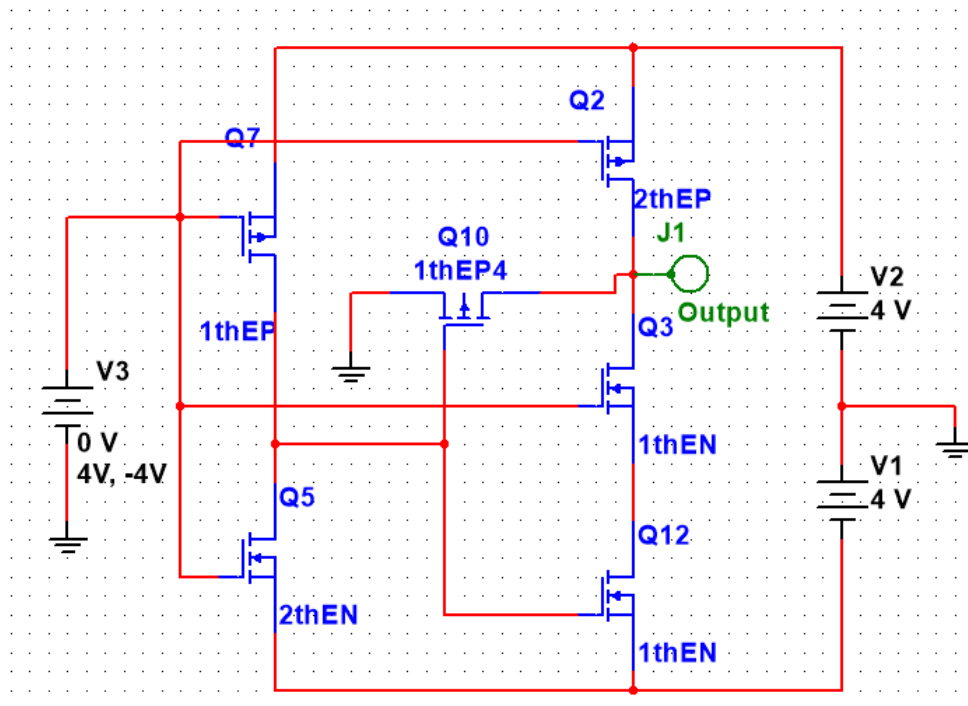


Рис. 3.40 – Синтезована схемотехнічна реалізація логічної функції подвійного циклічного заперечення (лівий цикл).

Базову логічну функцію $x1(-1)$ виконує транзистор Q8, функцію $x0(1)$ – транзистори Q7, Q9 та Q42, функцію $x-1(0)$ – транзистори Q7, Q9 та Q11.

Робота синтезованої схеми була досліджена серед схемотехнічного моделювання Multisim. Передавальна характеристика схеми, розрахована цією програмою, наведено на рис.3.41

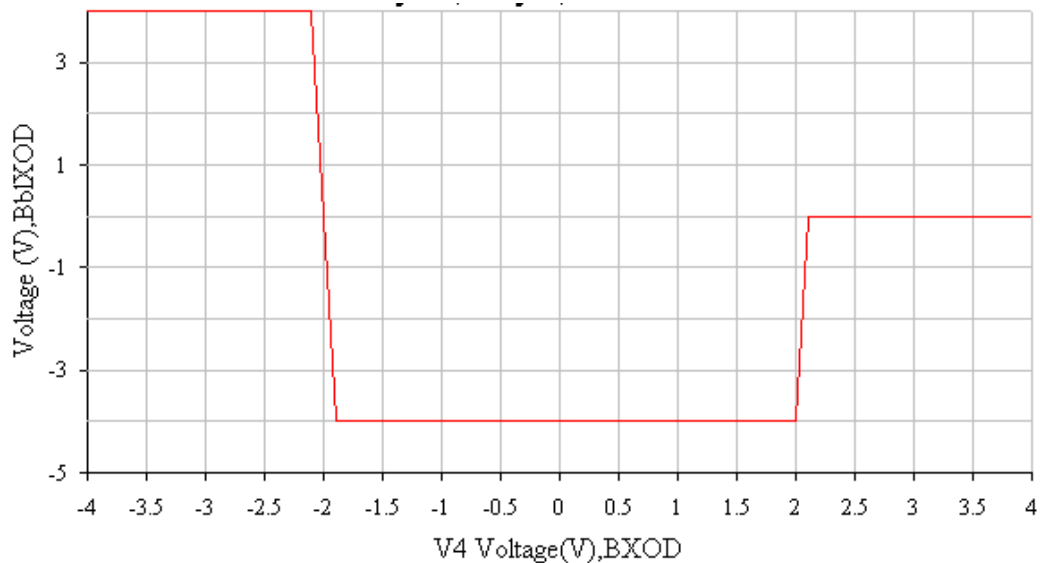


Рис. 3.41 – Передатна характеристика схеми, що здійснює функцію подвійного циклічного заперечення.

Наведена на рис.3.41 (функція лівого циклу). Характеристика побудована у середовищі програми схемотехнічного моделювання Multisim за допомогою процедури Parameter sweep при кроці дискретного збільшення для вхідної напруги від джерела V4 схеми на рис.3.41 0,1В. Графік характеристики показує, що схема, синтезована має чіткий і виражений поріг перемикавання та неспотвореність вихідних логічних рівнів. Робота схеми відповідає таблиці істинності функції подвійного циклічного заперечення (лівого циклу), яка наведена в табл. 3.5

Диз'юнкція ($P \vee Q$) має такі значення істинності: 1) вона вірна, коли вірно P (яке б не було Q) або коли вірно Q (яке б не було P); 2) вона хибна, якщо хибно P і хибно Q ; 3) вона визначена лише у зазначених випадках (а тому не визначена в решті).

Цьому визначенню відповідає таблиця істинності тризначної функції диз'юнкції прийнятої системи кодування істиннісних значень, наведена в табл. 3.6.

Таблиця 3.6

Таблиця істинності тризначної диз'юнкції.

$A \vee Y$	-1	0	1
-1	-1	0	1
0	0	0	1
1	1	1	1

Якщо синтезувати схему, що здійснює тризначну логічну функцію інверсії диз'юнкції двох аргументів, синтез буде простіше і кількість базових компонентів в схемі, що синтезується, буде менше, як і в разі синтезу схеми, що здійснює функцію тризначної кон'юнкції. Таблиця істинності функції інверсії диз'юнкції наведено у табл.3.7.

Таблиця 3.7

Таблиця істинності інверсії диз'юнкції.

$\neg A \vee Y$	-1	0	1
-1	1	0	-1
0	0	0	-1
1	-1	-1	-1

По таблиці 3.7 виробляється синтез трирівневого логічного елемента, що здійснює тризначну логічну функцію заперечення диз'юнкції двох аргументів.

Інформаційний сигнал "+1" визначається в таблиці істинності єдиною кон'юнкцією, що складається з логічних функцій двох базових елементів:

$$F1 = A1(-1) \& B1(-1) \quad (3.22)$$

Ця кон'юнкція є схемотехнічною формулою для частини схеми, що комує вихід інформаційний логічний рівень «+1».

Використовуючи базову логічну функцію $x0(-1,0)$, напишемо МДНФ частини логічної формули, що визначає нульовий інформаційний сигнал:

$$F0 = A0(-1,0) \& B0(0) \vee A0(0) \& B0(-1,0) \quad (3.23)$$

З неї випливає схемотехнічна формула:

$$\begin{aligned} F0 = & A0(-1,0) \& B0(-1,0) \& B0(0,1) \vee \\ & \vee A0(-1,0) \& A0(0,1) \& B0(-1,0) \end{aligned} \quad (3.24)$$

Мінімізуємо вираз виносячи за дужки загальні кон'юнкції:

$$F0 = A0(-1,0) \& B0(-1,0) \& \{B0(0,1) \vee A0(-1,0)\} \quad (3.25)$$

Оскільки за рівності одного з вхідних сигналів позитивного логічного рівня сигнал на виході дорівнює «-1» і залежить від стану другого входу, то інформаційний логічний сигнал «-1» визначається наступною диз'юнкцією:

$$F -1 = A -1(1) \vee B -1(1) \quad (3.26)$$

Оскільки дана диз'юнкція складається з логічних функцій базових логічних компонентів, вона є схемотехнічною формулою частини схеми, що комутує вихід логічний інформаційний сигнал «-1».

Повна схемотехнічна формула для схеми, що здійснює тризначну логічну функцію заперечення диз'юнкції двох аргументів, записується так:

$$F = A1(-1) \& B1(-1) \vee A0(-1,0) \& B0(-1,0) \& B0(0,1) \vee A0(-1,0) \& A0(0,1) \& B0(-1,0) \vee A-1(1) \vee B-1(1) \quad (3.27)$$

За схемотехнічною формулою виробляється синтез схеми. Синтезована схема наведена на рис. 3.42

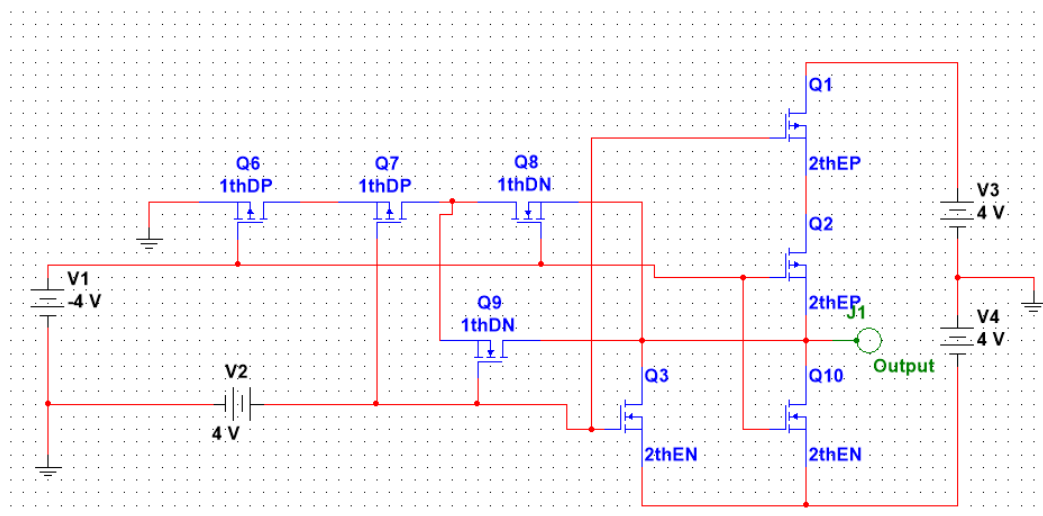


Рис.3.42 – Синтезована схема, що здійснює тризначну логічну функцію інверсії диз'юнкції двох аргументів.

Для схеми, зображеної на рис. 3.42, у середовищі схемотехнічного моделювання Multisim були побудовані передатні характеристики при фіксованих логічних рівнях на одному з входів схеми та лінійній зміні напруги

на іншому вході від -4 до $+4$ В. Дані характеристики наведені на рис. 3.44 – 3.46.

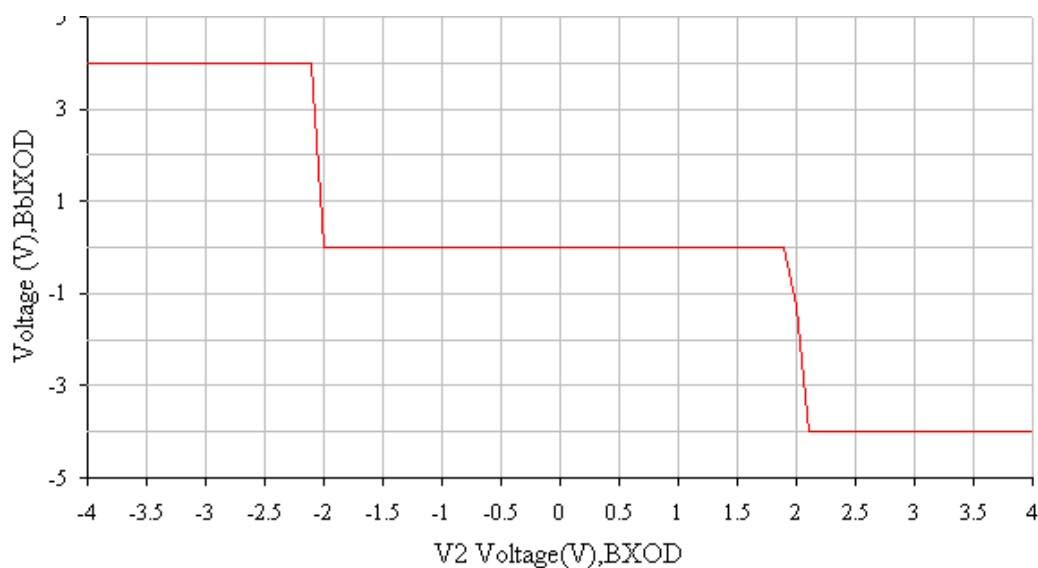


Рис. 3.43 – Передавальна характеристика синтезованої схеми, наведеної на рис. 3.42, при фіксованому рівні напруги на вході А, рівному -4 , що відповідає логічному рівню «-1», для лінійної зміни напруги на вході в діапазоні від -4 до $+4$ в.

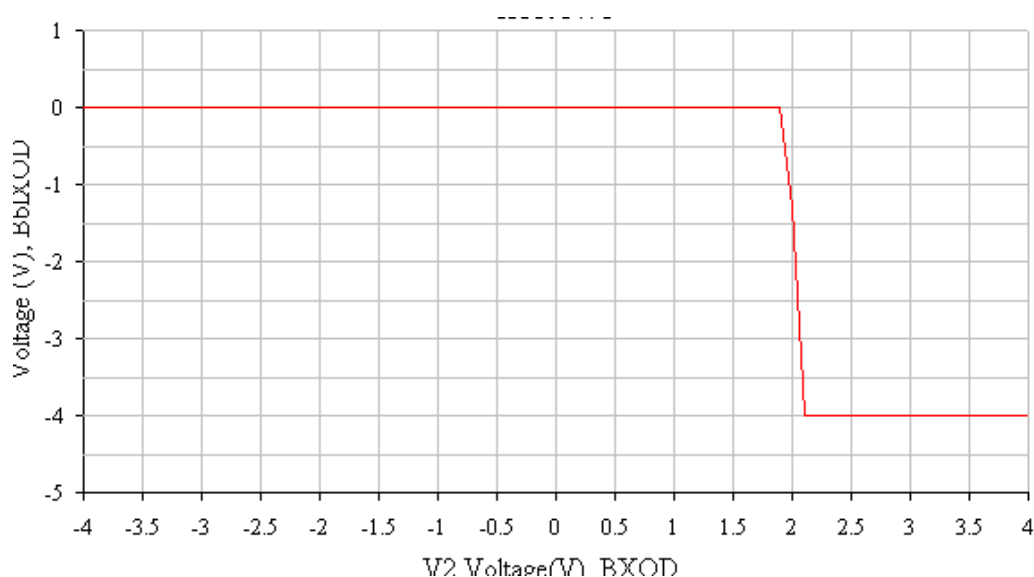


Рис. 3.44 – Передатна характеристика синтезованої схеми при лінійній зміні напруги на вході в діапазоні від -4 до $+4$ В при фіксованому нульовому логічному рівні 0В на вході А.

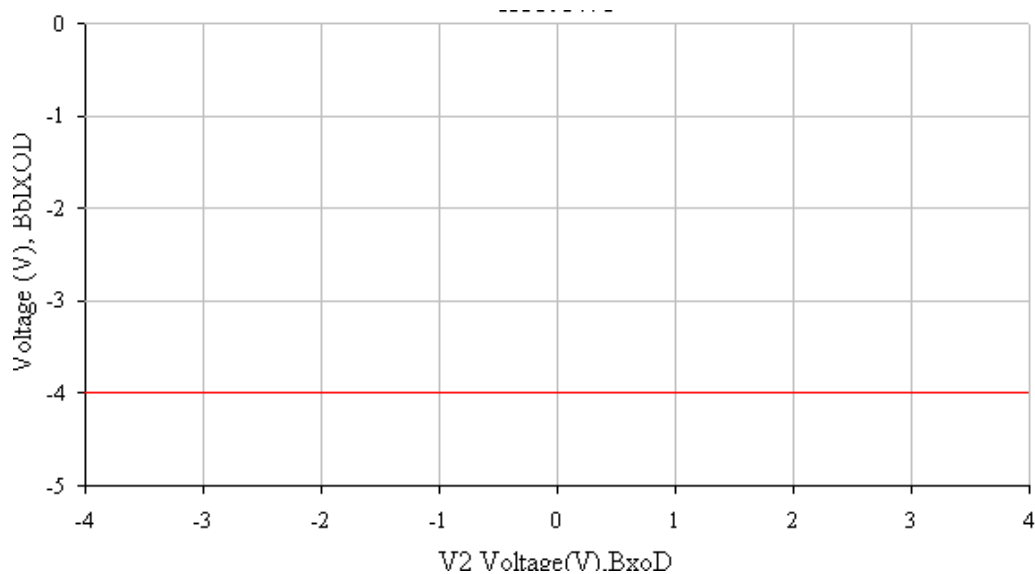


Рис. 3.45 – Передавальна характеристика схеми.

Передавальні характеристики, наведені на рис. 3.43-45, побудовані серед схемотехнічного моделювання Multisim за допомогою процедури Parameter sweeper для схеми, зображеної на рис. 3.43 при кроці дискретного збільшення напруги джерела вхідного сигналу на вході (батарея V2 на рис.3.43), рівному 0,1 В. Наведені характеристики показують, що розроблена схема має чіткі пороги перемикавання і неспотворені рівні вихідних логічних сигналів. Робота схеми відповідає таблиці істинності тризначної логічної функції інверсії диз'юнкції двох аргументів (табл. 3.5). Власне, диз'юнкцію двох аргументів можна отримати, якщо проінвертувати вихідний сигнал синтезованої схеми.

Функція диз'юнкції двох аргументів і циклічна заперечення утворюють функціонально повний базис тризначної логіки, тобто. за їх допомогою можна синтезувати будь-яку трирівневу схему.

3.3 Розробка арифметико-логічного пристрою на базі трійкових елементів

Трирівнева двовходова схема, що здійснює тризначну функцію додавання по модулю три ($\text{mod } 3$), може бути основою для побудови трійкових суматорів у трирівневих арифметичних пристроях.

У таблиці 15 наведено правило додавання по модулю три в трійковій системі числення, де прийнято лише три цифри – 0, 1 та 2.

Таблиця 3.8

Додавання за модулем три

$\oplus$	0	1	2
0	0	1	2
1	1	2	0
2	2	0	1

Якщо прийняти наступне кодування трьох цифр логічними рівнями трійкових схем, коли цифра 0 кодується рівнем логічного «-1», цифра 1 – рівнем логічного нуля і цифра 2 кодується логічною «+1» ($-1 \equiv 0$, $0 \equiv 1$, $1 \equiv 2$), тоді таблиця істинності для двовходової трирівневої схеми, що здійснює додавання по модулю три, буде виглядати наступним чином (табл. 3.8).

По таблиці істинності функції складання по модулю три (табл.3.6) видно, схемотехнічна формула, що описує цю функцію, і, отже, сама схема, що реалізує функцію, будуть досить складними. За будь-яких варіантів кодування цифр складність не зменшиться – у будь-якому випадку необхідне використання складних базових функцій з використанням характеристичних функцій та композиції базових елементів для визначення кожного інформаційного сигналу за будь-якого набору вхідних сигналів. Дана схема не дозволить провести мінімізацію з використанням додаткових базових

логічних функцій, оскільки в таблиці істинності немає навіть двох однакових сусідніх кодів.

Таблиця 3.8

Таблиця істинності функції додавання по модулю три відповідно до кодування логічними рівнями.

$\oplus$			
	-1	0	1
-1	-1	0	1
0	Вхід А		-1
1			0

Запишемо функції СДНФ для інформаційного сигналу «-1» (функція $A \oplus Y \bmod 3$ позначатиметься як S):

$$S - 1 = A - 1(-1) \& B - 1(-1) \vee A - 1(1) \& B - 1(0) \vee \vee A - 1(0) \& B - 1(1) \quad (3.28)$$

Для кожної кон'юнкції даної СДНФ запишемо схемотехнічні формули, що складаються з функцій базового набору компонентів трирівневих схем і наведемо схемотехнічну реалізацію кожного фрагмента схеми:

$$A - 1(-1) \& B - 1(-1) = A - 1 \{ \varphi - 1(A) = 1 \} \& \& B - 1 \{ \varphi - 1(B) = 1 \} \quad (3.29)$$

Схемотехнічна реалізація кон'юнкції наведено на рис. 3.47. У цій схемі транзистори Q1, Q3 мають високу порогову напругу, інші – низька. Пара транзисторів Q1 і Q2, а також пара транзисторів Q3 і Q4 реалізують характеристичні функції $\varphi^{-1}(A)$ та $\varphi^{-1}(B)$ відповідно. Послідовне з'єднання транзисторів Q5 і Q6 здійснюють кон'юнкцію вхідних сигналів та інформаційного сигналу «-1».

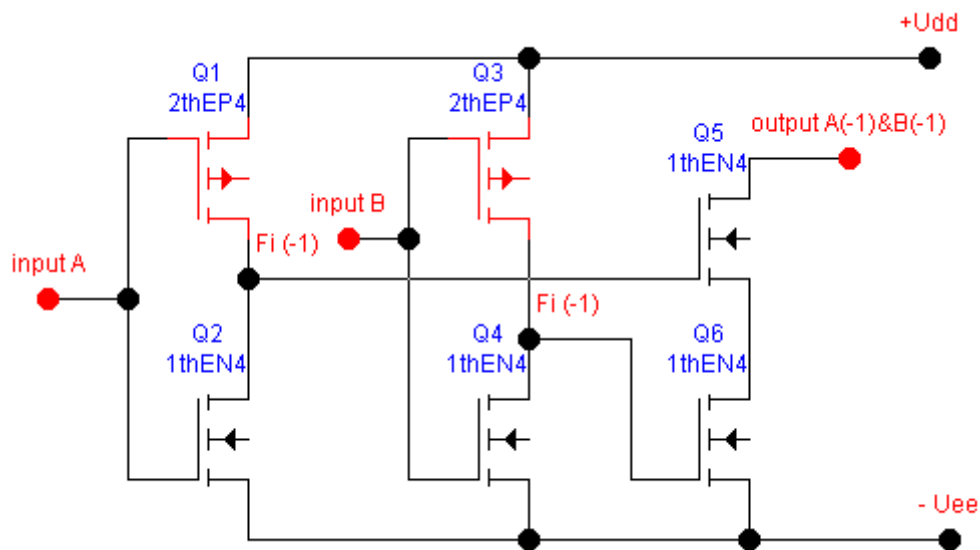


Рис. 3.47 – Схемотехнічна реалізація кон'юнкції $A^{-1}(-1) \& B^{-1}(-1)$ для інформаційного сигналу «-1» ($-U_{EE}$).

Запишемо схемотехнічну формулу для кон'юнкції $A^{-1}(1) \& B^{-1}(0)$:

$$A^{-1}(1) \& B^{-1}(0) = A^{-1}(1) \& B^{-1}(0,1) \&$$

$$\& B^{-1}(\neg \varphi 1(B) = 1 \quad (3.30)$$

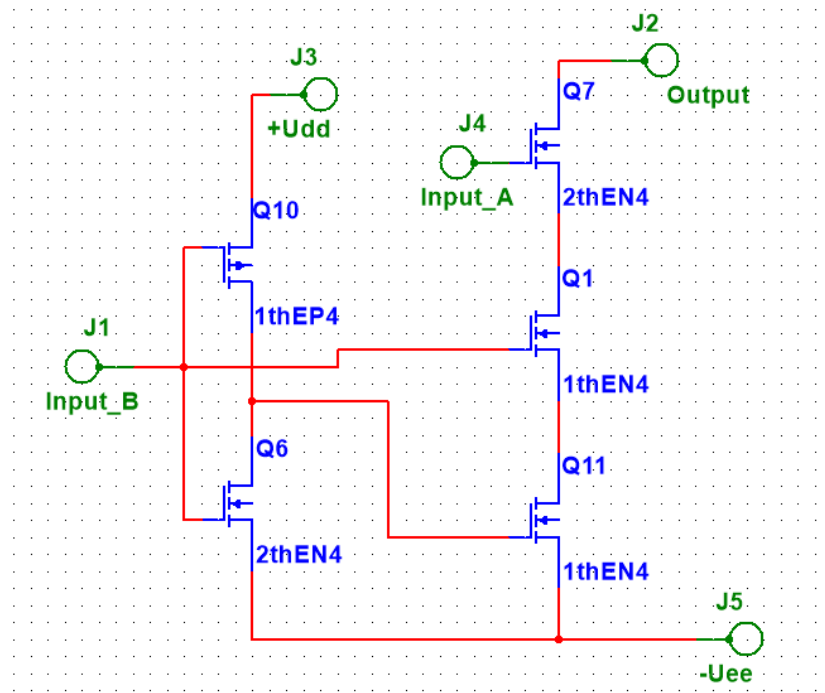


Рис. 3.48 – Реалізація кон'юнкції $A-1(1) \& B-1(0)$.

У схемі на рис. 3.48 транзистори Q6 і Q7 мають високий поріг відкриття (індукування) каналу, решта – низький. На транзисторах Q10, Q6 здійснено заперечення характеристичної функції $\neg\phi 1 (B)$. Транзистори Q1, Q6, Q10, Q11 здійснюють базову логічну функцію $B-1(0)$, а транзистор Q9 – базову логічну функцію $A-1(1)$.

Запишемо схемотехнічну формулу для кон'юнкції $A-1(0) \& B-1(1)$, що визначає інформаційний сигнал “-1” на одному з наборів:

$$A-1(0) \& B-1(1) = A-1(0,1) \&$$

$$\& A-1 \{ \neg\phi 1 (A) = 1 \} \& B-1(1) \quad (3.31)$$

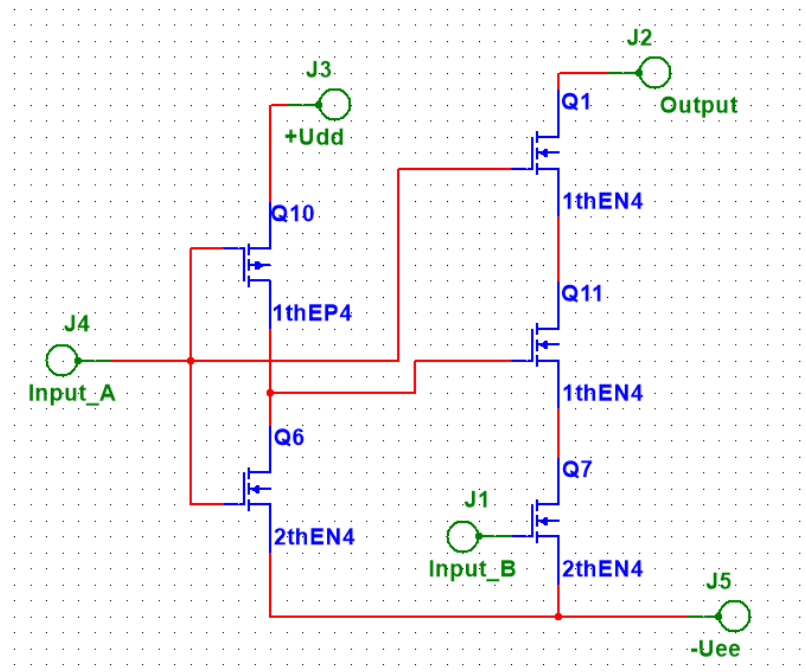


Рис. 3.49 – Схемотехнічна реалізація кон'юнкції $A-1(0) \& B-1(1)$ для інформаційного сигналу “-1” (-UEE).

У схемі на рис. 3.49 транзистори Q6 і Q7 мають високу граничну напругу. Транзистори Q10, Q6 реалізують інверсію характеристичної функції $\neg\phi 1(A)$. На транзисторах Q12 – Q15 здійснено базову логічну функцію $A-1(0)$, а транзистор Q16 виконує базову логічну функцію $B-1(1)$.

Таким чином, схемотехнічна формула для інформаційного сигналу “-1”, яка описує частину схеми, що синтезується, здійснює комутацію інформаційного сигналу “-1” (-UEE) на вихід при заданих в табл.3.8 вхідних наборах, записується наступним чином:

$$S-1 = A-1 \{ \phi-1(A) = 1 \} \& B-1 \{ \phi-1(B) = 1 \} \vee$$

$$\vee A-1(1) \& B-1(0,1) \& B-1(\neg\phi 1(B) = 1) \vee$$

$$\vee A-1(0,1) \& A-1 \{ \neg\phi 1(A) = 1 \} \& B-1(1) \quad (3.32)$$

Схемотехнічна формула реалізується паралельним поєднанням схем, наведених на рис. 3.47 – 3.49, тобто. об'єднуються однойменні входи та всі виходи зазначених схем, утворюючи тим самим частину схеми, що визначає інформаційний сигнал «-1».

Розглянемо синтез частини схеми, яка здійснює комутацію на вихід нульового інформаційного сигналу. СДНФ для нульового інформаційного сигналу записується так:

$$S_0 = A_0(0) \& B_0(-1) \vee A_0(-1) \& B_0(0) \vee A_0(1) \& B_0(1) \quad (3.33)$$

Схемотехнічні формули для кожної кон'юнкції СДНФ записуються так:

$$A_0(0) \& B_0(-1) = A_0(-1,0) \& A_0(0,1) \& B_0(-1) \quad (3.34)$$

$$A_0(-1) \& B_0(0) = A_0(-1) \& B_0(-1,0) \& B_0(0,1) \quad (3.35)$$

Кон'юнкція $A_0(1) \& B_0(1)$ є схемотехнічною формулою, т.к. складається із функцій базових логічних елементів.

Виходячи з формул (3.33) - (3.35), повна схемотехнічна формула для нульового інформаційного сигналу записується наступним чином:

$$S_0 = A_0(-1,0) \& A_0(0,1) \& B_0(-1) \vee \\ \vee A_0(-1) \& B_0(-1,0) \& B_0(0,1) \vee A_0(1) \& B_0(1) \quad (3.36)$$

Схемотехнічна реалізація формули наведено на рис. 3.50.

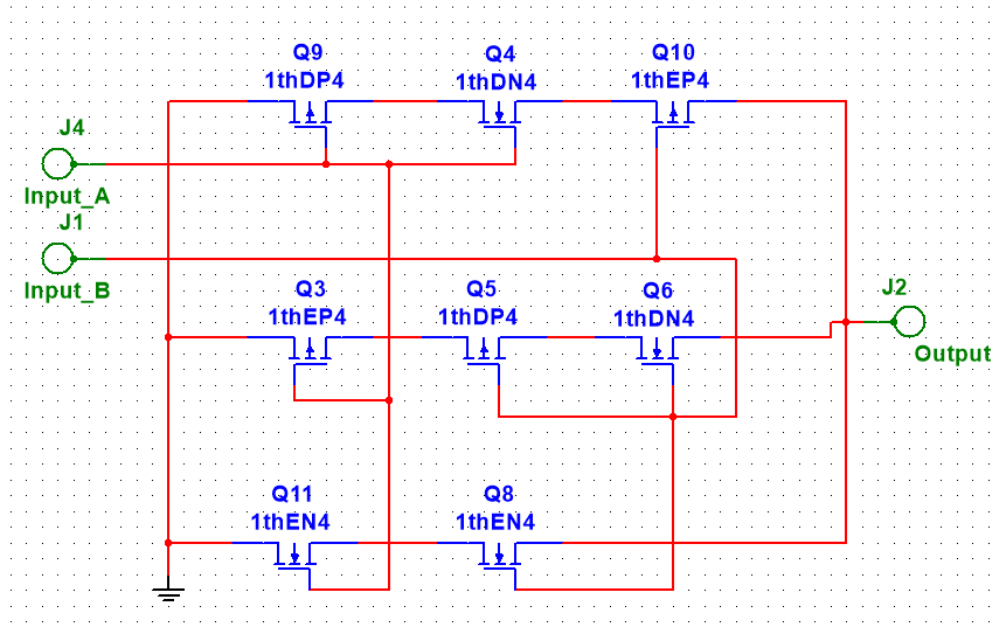


Рис. 3.50 – Схемотехнічна реалізація частини схеми, що здійснює комутацію нульового інформаційного сигналу на вихід на нульових вхідних наборах функції додавання по модулю три.

Ланцюг послідовно з'єднаних транзисторів Q9 – Q10 у схемі на рис.3.50 здійснюють кон'юнкцію $A0(0) \& B0(-1)$, ланцюг послідовно з'єднаних транзисторів Q3, Q5, Q6 реалізують кон'юнкцію $A0(-1) \& B0(0)$, послідовне з'єднання транзисторів Q11 та Q8 реалізує кон'юнкцію $A0(1) \& B0(1)$. Паралельне з'єднання даних ланцюгів реалізує СДНФ інформаційного сигналу “0” S 0 (3.33).

Внаслідок застосування транзисторів з індукованим каналом Q19, Q20, Q23 і Q24 як комутують нульовий інформаційний сигнал при перемиканнях з нульового логічного стану в інші спостерігатимуться спотворення рівнів інформаційних сигналів.

Розглянемо послідовно синтез частини схеми додавання по модулю три, що здійснює комутацію інформаційного сигналу «+1» ($+UDD$) на вихід схеми.

СДНФ функції, що визначає одиничні набори в таблиці істинності (табл.3.8), записується так:

$$S1 = A1(1) \& B1(-1) \vee A1(0) \& B1(0) \vee A1(-1) \& B1(1) \quad (3.37)$$

Запишемо схмотехнічну формулу для кон'юнкції $A1(1) \& B1(-1)$:

$$A1(1) \& B1(-1) = A1 \{ \neg \varphi 1 (A) = -1 \} \& B1(-1) \quad (3.38)$$

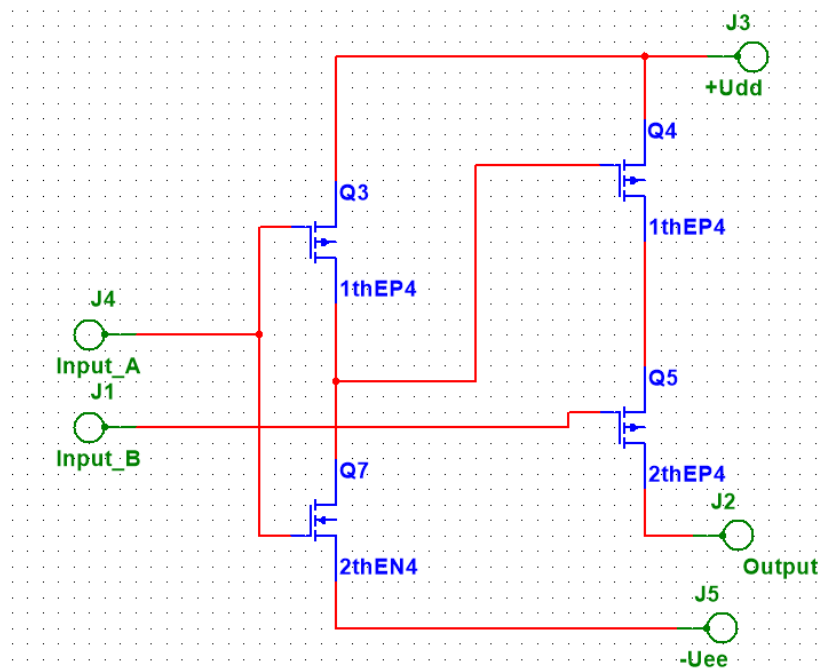


Рис. 3.51 – Схмотехнічна реалізація кон'юнкції $A1(1) \& B1(-1)$ для інформаційного сигналу “+1” (+UDD).

У схемі на рис. 3.51 транзистори Q7, Q5 мають високу прогову напругу. Транзистори Q3, Q7 здійснюють інверсію характеристичної функції $\neg \varphi 1 (A)$, а разом із транзистором Q4 вони реалізують базову логічну функцію $A1(1)$. Транзистор Q5 реалізується базова логічна функція $B1(-1)$.

Запишемо схмотехнічну формулу для наступної кон'юнкції із СДНФ (3.39):

$$A1(0) \& B1(0) = A1 \{ \varphi^{-1}(A) = -1 \} \& A1(-1,0) \&$$

$$\& B1 \{ \varphi^{-1}(B) = -1 \} \& B1(-1,0) \quad (3.39)$$

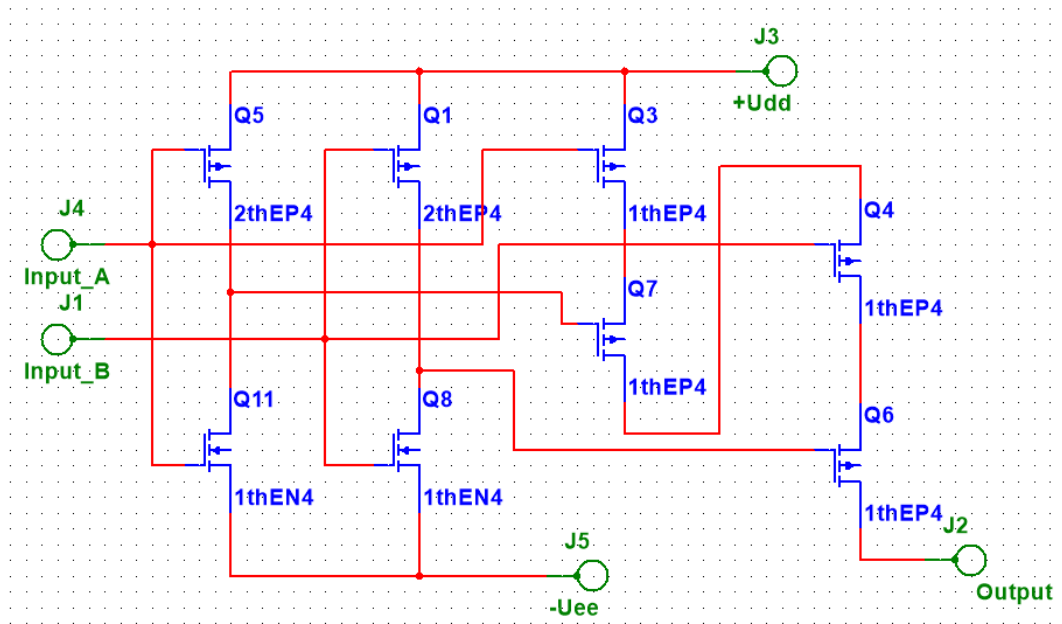


Рис. 3.52 – Схемотехнічна реалізація кон'юнкції $A1(0) \& B1(0)$ для інформаційного сигналу +1 (+UDD).

У схемі на рис. 3.52 транзистори Q5 і Q1 мають високу граничну напругу. Пари транзисторів Q5-Q11 та Q1-Q8 реалізують характеристичні функції φ^{-1} . Транзисторами Q5, Q11, Q3, Q7 здійснюється базова логічна функція $A1(0)$, транзисторами Q1, Q8, Q4, Q6 – базова логічна функція $B1(0)$.

Запишемо схемотехнічну формулу для кон'юнкції $A1(-1) \& B1(1)$:

$$A1(-1) \& B1(1) = A1(-1) \& B1 \{ \neg \varphi^{-1}(B) = -1 \} \quad (3.40)$$

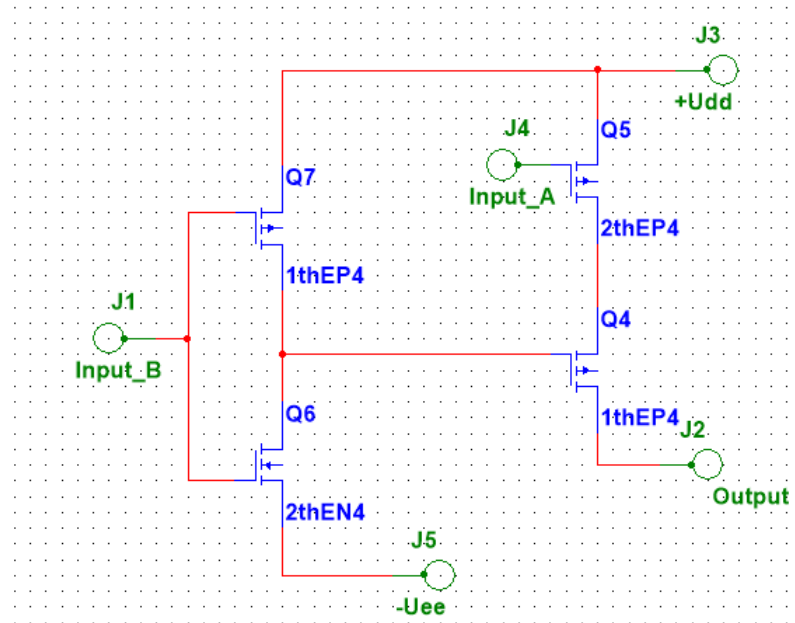


Рис. 3.53 – Схемотехнічна реалізація кон'юнкції $A1(-1) \& B1(1)$.

У схемі на рис.3.53 транзистори Q7, Q6 здійснюють інверсію характеристичної функції $\neg\phi 1(B)$, і разом із транзистором Q4 – базову логічну функцію $B1(1)$. Транзистор Q5 здійснює логічну базову функцію $A1(-1)$. Транзистори Q6 і Q5 мають високу порогову напругу.

Паралельне поєднання схем, зображених на рис. 3.51 – 3.53, утворює частину схеми, що здійснює комутацію інформаційного сигналу «+1» на вихід схеми для відповідних входних наборів табл.3.7 та реалізує повну схемотехнічну формулу для даного інформаційного сигналу:

$$S1 = A1 \{ \neg\phi 1 (A) = -1 \} \& B1(-1) \vee$$

$$\vee A1 \{ \phi -1(A) = -1 \} \& A1(-1,0) \& B1 \{ \phi -1(B) = -1 \} \& B1(-1,0) \vee$$

$$\vee A1(-1) \& B1 \{ \neg\phi 1(B) = -1 \} \quad (3.41)$$

Схемотехнічна формула, що описує всю схемну реалізацію функції додавання по модулю три, є диз'юнкцією трьох схемотехнічних формул для

кожного інформаційного сигналу $S-1 \vee S0 \vee S1$. У цій схемотехнічній формулі кожна характеристична функція зустрічається двічі, що дозволяє скоротити загальну кількість транзисторів у схемі з 40 до 32, використовуючи одні й самі характеристичні функції керувати різними транзисторами. Повна схема, синтезована за наведеними вище схемотехнічних формул.

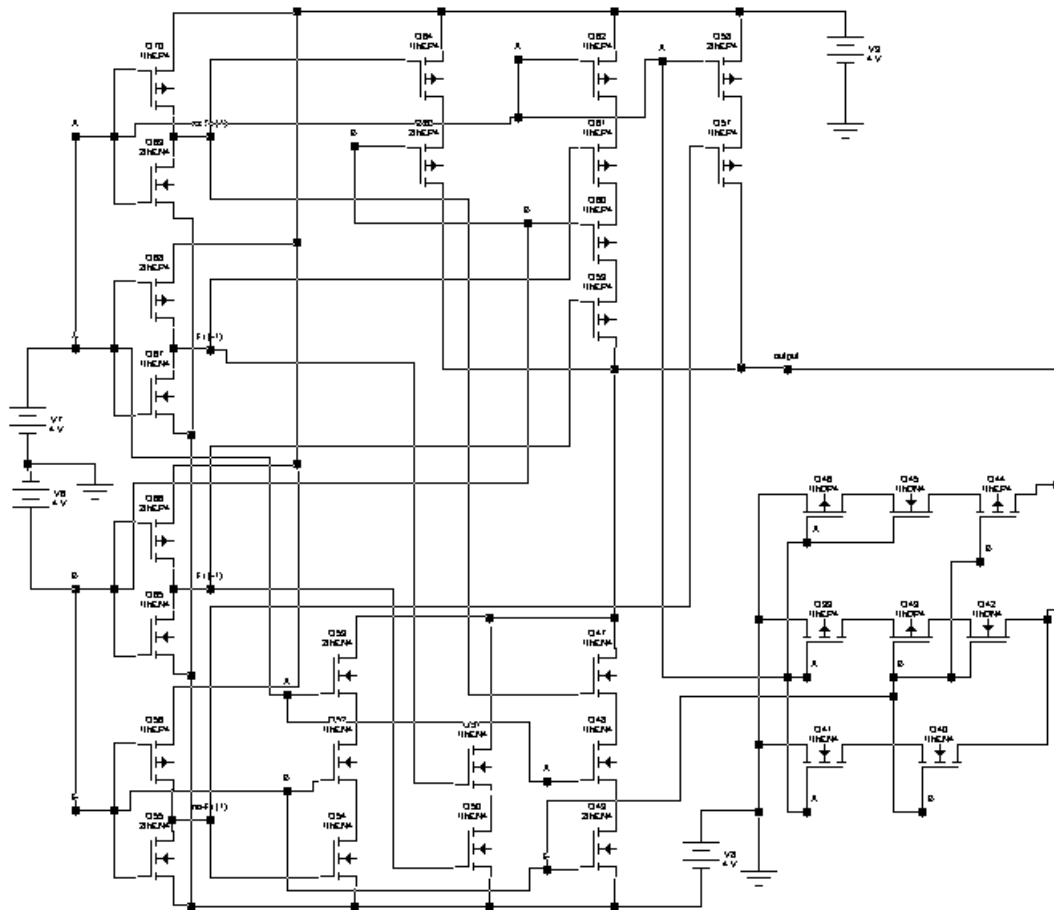


Рис. 3.54 – Повна тривірнева логічна схема, що здійснює функцію додавання двох аргументів за модулем три.

Робота схеми, наведеної на рис.3.54 була досліджена в середовищі програмного схемотехнічного моделювання Multisim були побудовані передатні характеристики схеми при фіксованих логічних рівнях на одному з входів і лінійній зміні напруги на іншому вході в діапазоні напруги від -4В до

+4В при кроці дискретної зміни вхідної напруги 0,2 В. Передавальні характеристики для трьох побудов наведено на рис. 3.55 – 3.56.

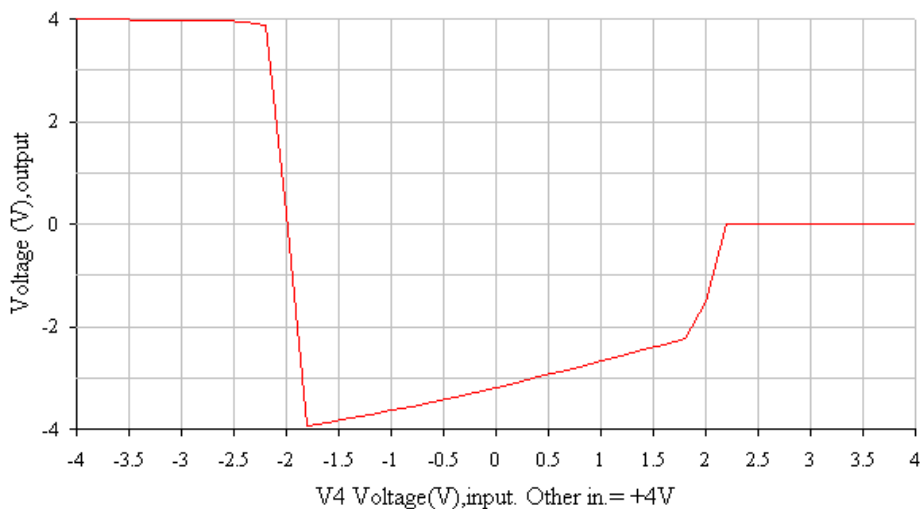
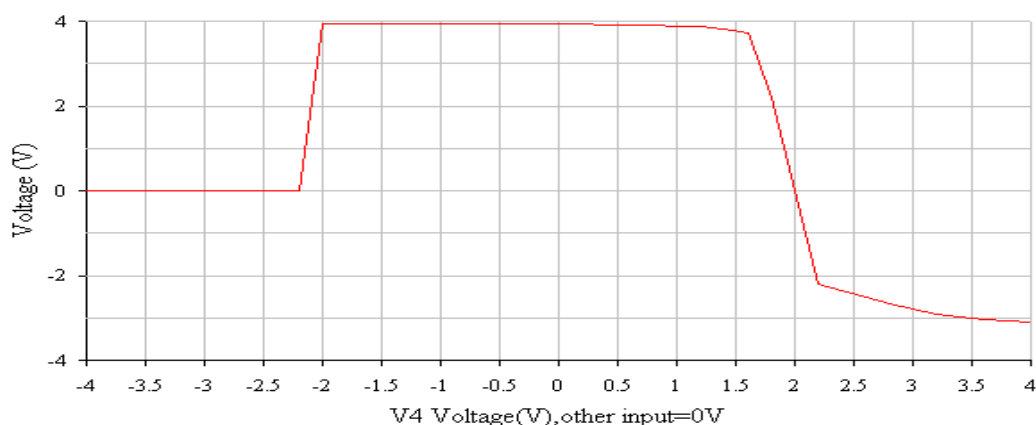


Рис. 3.55 – Передавальна характеристика схеми.

При фіксованому рівні напруги на одному з входів, що дорівнює +4 В (рівень логічної «+1») і лінійній зміні напруги на іншому вході в діапазоні від – 4 до + 4 В. Крок дискретного збільшення вхідної напруги на одному з входів 0,2 В.

Рис. 3.56 – Передавальна характеристика схеми



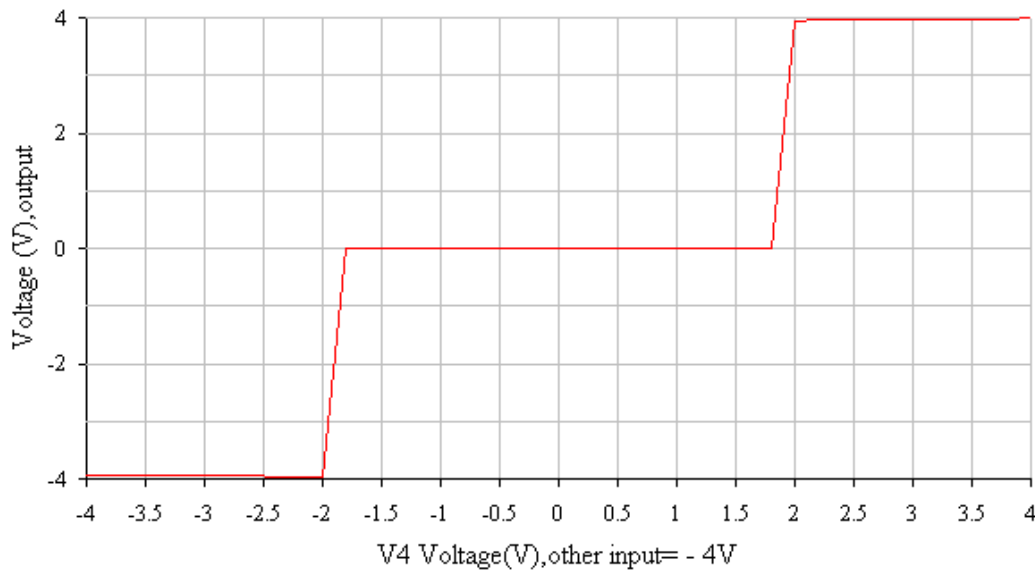


Рис. 3.57 – Передавальна характеристика схеми, наведеної на рис.3.54, при фіксованому рівні напруги на одному з входів, що дорівнює -4 В (рівень логічної «-1») і лінійній зміні напруги на іншому вході в діапазоні від -4 до $+4$ В.

Графіки наведені на рис. 3.55 та 3.57 показують, що схема працює відповідно до заданої таблиці істинності, але при цьому при перемиканні логічних рівнів на виході відбувається спотворення інформаційних логічних сигналів. У принципі, спотворення логічних рівнів лежать у допустимих межах (0,5) і дана схема працездатна, але такий варіант може вимагати додаткових формувачів вихідних логічних рівнів сигналів. Спотворення відбувається внаслідок того, що як ключі, що комутують нульовий логічний інформаційний сигнал на вихід, використовуються польові транзистори з індукованим каналом. Спотворення вихідних логічних рівнів можна усунути, n -канальний – з характеристичної функції $\varphi -1$ вхідного сигналу, але в p -канальний – з інверсії характеристичної функції $\neg\varphi 1$ вхідного сигналу. Ця заміна проілюстрована на рис. 3.58 та 3.59.

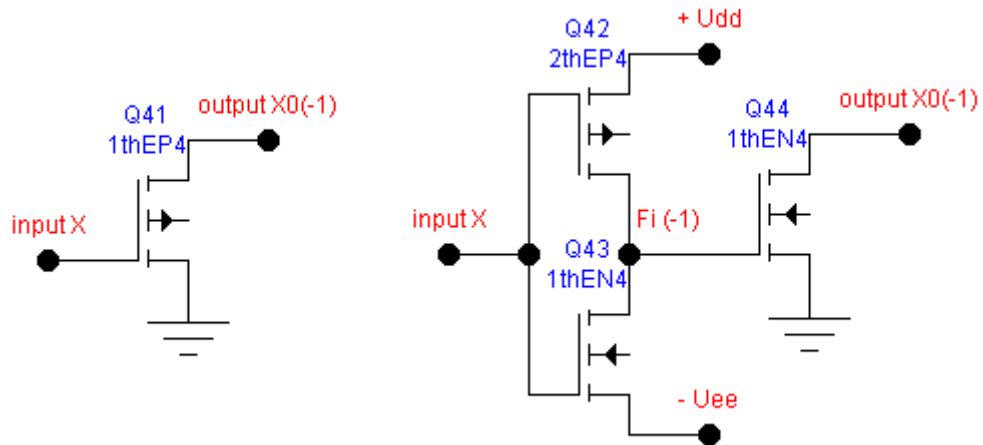


Рис. 3.58 – Заміна р-канального транзистора Q41, що виконує базову логічну функцію $X0(-1)$, на еквівалентну схему з транзисторів Q42 - Q44, що виконують ту саму функцію.

Дана заміна дозволяє уникнути спотворення вихідних логічних рівнів та їх зтяжного перемикавання. Транзистори Q43, Q44 реалізують характеристичну функцію $\varphi - 1(X)$ і разом із транзистором Q45 виконують функцію $X0(-1) = X0\{\varphi - 1(X) = 1$.

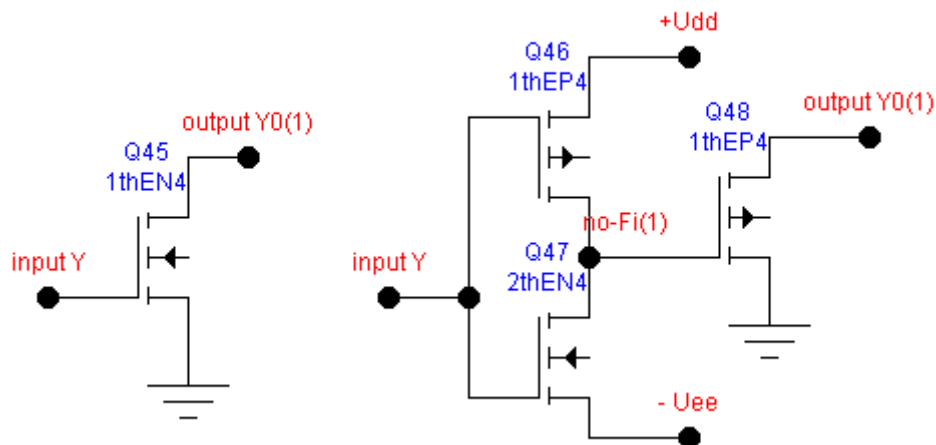
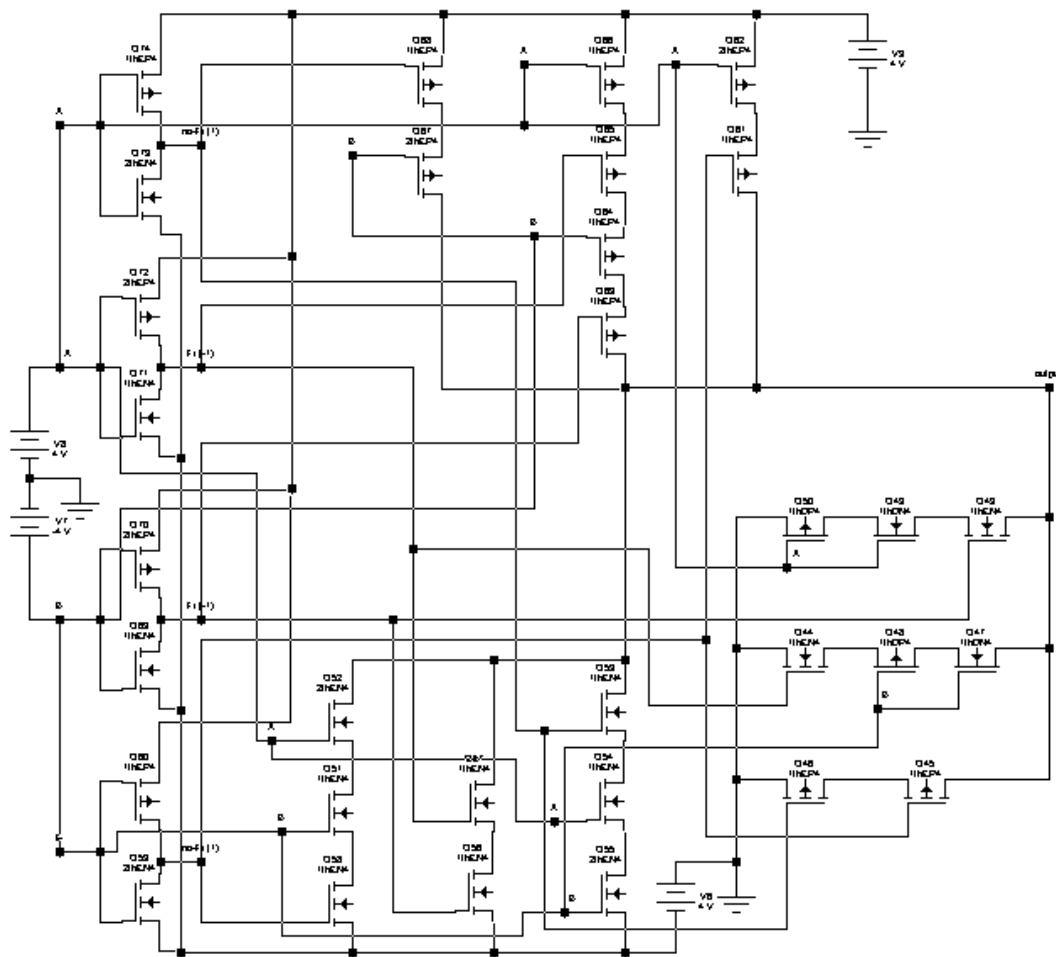


Рис. 3.59 – Заміна п-канального транзистора Q45, що виконує базову логічну функцію $Y0(1)$, на еквівалентну схему з транзисторів Q46 - Q48, що виконують ту саму функцію.

Дана заміна дозволяє уникнути спотворення вихідних логічних рівнів та їх зтяжного перемикання. Транзистори Q46, Q47 реалізують інверсію характеристичної функції $\neg\varphi 1(Y)$ та спільно з транзистором Q48 виконують функцію $Y0(1) = Y0\{\neg\varphi 1(Y) = -1$.

Модифікована схема наведена на рис. 3.60. Оскільки характеристичні функції у цій схемі вже використовуються, заміна базових логічних компонентів, проілюстрована на рис. 3.58 і 3.59 не призводить до збільшення числа транзисторів у схемі.



фіксованих логічних рівнях на одному з входів і лінійній зміні напруги на іншому вході в діапазоні напруг від -4В до +4В при кроці дискретної зміни вхідної напруги 0,2 В.

Передавальні характеристики наведено на рис.3.61 – 3.63.

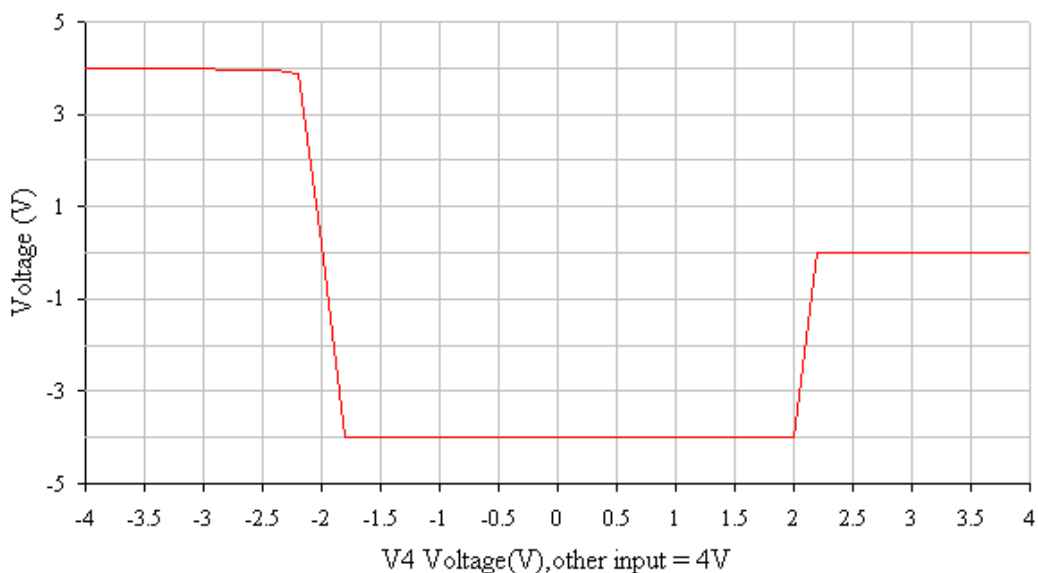


Рис.3.61 – Передавальна характеристика схеми.

При фіксованому рівні напруги на одному з входів, що дорівнює +4 В (рівень логічної «+1») і лінійній зміні напруги на іншому вході в діапазоні від -4 до +4 В. Крок дискретного збільшення вхідної напруги на одному з входів 0,2 В. Рівні вихідних логічних сигналів не спотворюються.

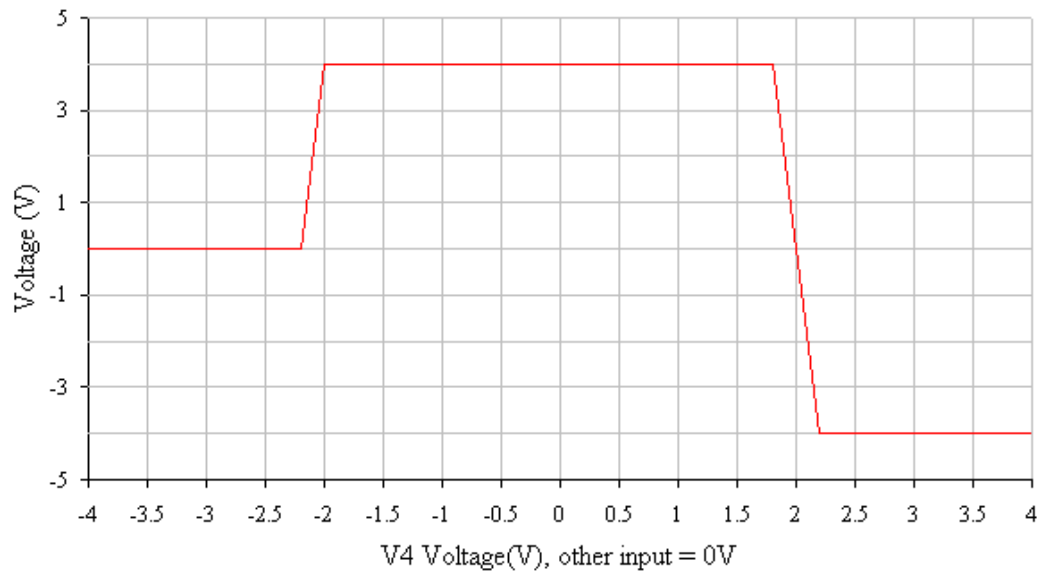


Рис. 3.62 – Передавальна характеристика схеми, наведеної на рис. 3.60, при фіксованому рівні напруги на одному з входів, що дорівнює 0 (рівень логічного нуля) і лінійній зміні напруги на іншому вході в діапазоні.

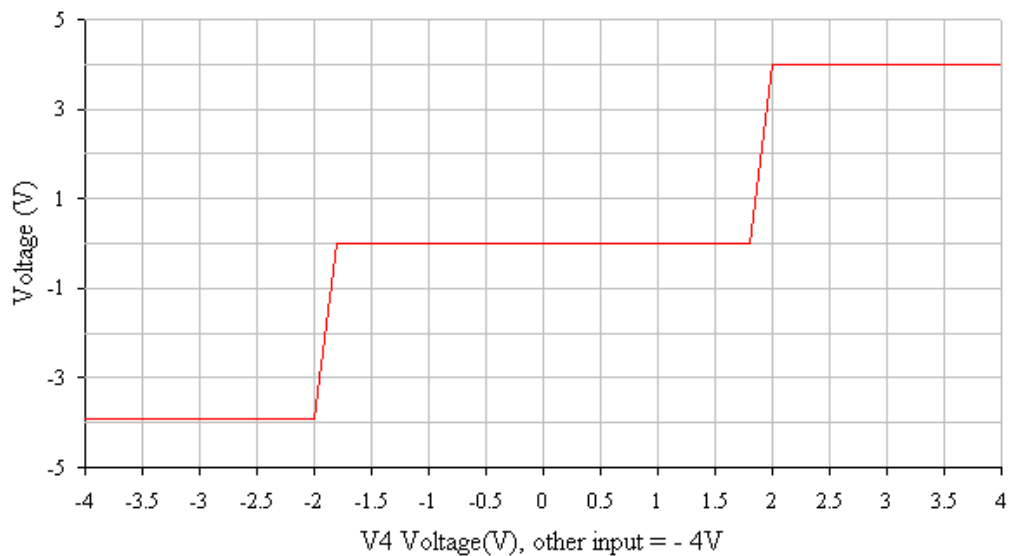


Рис. 3.63 – Передавальна характеристика схеми.

При фіксованому рівні напруги на одному з входів, що дорівнює - 4 В (рівень логічної «-1») і лінійній зміні напруги на іншому вході в діапазоні від - 4 до + 4 В. Графік побудований в середовищі схемотехнічного моделювання Multisim з за допомогою процедури Parameter sweep для кроку дискретного

збільшення вхідної напруги на одному з входів 0,2 В. Рівні вихідних логічних сигналів не спотворюються.

Графіки передавальних характеристик, наведені на рис.3.61 – 3.63, побудовані програмою схемотехнічного моделювання Multisim для схеми, зображеної на рис. 3.60 показують відповідність роботи схеми таблиці істинності функції додавання по модулю три (табл. 3.7). Синтезована схема має чіткий поріг перемикування. Рівні інформаційних логічних сигналів передаються вихід без спотворень.

Функція складання по модулю три без урахування переносів разом із функцією множення по модулю три і константою одиниця утворюють повну систему тризначної логіки. Ця теорема доведена в [4, стор 339].

Для застосування синтезованої схеми в арифметичних суматорах, що оперують числами, представленими в трійковій системі числення, схему додавання по три модулі необхідно доповнити схемою переносу в наступний розряд.

Найбільш типовими логічними вузлами арифметичних пристроїв обчислювальних машин та різних систем обробки інформації є суматори.

Основу схеми суматора становить так званий напівсуматор, що складається зі схеми, що реалізує функцію додавання по три модулю, синтезовану в попередньому розділі, і схеми переносу.

Синтезуємо трійкову схему переносу, що доповнює схему додавання по модулю три до схеми напівсуматора. Таблиця істинності для схеми перенесення кодів $-1, 0, 1$ наведено у таблиці 3.8.

Таблиця 3.8

Таблиця істинності функції перенесення.

3	-1	0	1
-1	-1	-1	-1
0	-1	-1	0
1	-1	0	0

Синтезована схема вийде простіше, якщо синтезувати її для перетвореної таблиці істинності до виду, наведеному в таблиці 3.9.

Таблиця 3.9

Перетворена таблиця істинності схеми перенесення.

C_m	-1	0	1
-1	1	1	1
0	1	1	-1
1	1	-1	-1

Схема даної таблиці істинності вийде найбільш оптимальної. І якщо вихід даного варіанту схеми подати на перетворювач логічних сигналів, який перетворює сигнал «1» на сигнал «-1», а сигнал «-1» на сигнал «0», то вихідні сигнали відповідатимуть таблиці 3.8. Такий перетворювач логічних сигналів може складатися лише з двох транзисторів.

Запишемо СДНФ функції, що визначає таблицю 3.9. Для інформаційного сигналу «+1» СДНФ записується так:

$$C_{m1} = A1(-1) \& B1(-1) \vee A1(-1) \& B1(0) \vee A1(-1) \& B1(1) \vee A1(0) \& B1(-1) \vee A1(1) \& B1(-1) \quad (3.42)$$

СДНФ для інформаційного сигналу «-1»:

$$C_{m-1} = A-1(1) \& B-1(1) \vee A-1(1) \& B-1(0) \vee A-1(0) \vee B-1(1) \quad (3.43)$$

Використовуючи основні та додаткові базові логічні функції, можна мінімізувати наведені вище СДНФ. Запишемо мінімальну ДНФ функції, що визначає таблицю 18, що включає базові логічні функції для обох інформаційних сигналів:

$$Cm = A1(-1) \vee B1(-1) \vee A1(-1,0) \& B1(-1,0) \vee$$

$$A-1(1) \& B-1(0,1) \vee A-1(1) \& B-1(0,1) \quad (3.44)$$

Дана МДНФ є схемотехнічною формулою, оскільки складається з базових логічних функцій, що виконуються базовими компонентами трирівневих пристроїв. Схема, що описується формулою (3.44), реалізує таблицю істинності, наведену у табл. 18. Для того, щоб реалізувати функцію перенесення, таблицю істинності якої наведено в табл. 3.8 вихід даної схеми потрібно подати на перетворювач логічного коду, який описується наступною формулою:

$$C = Cm-1(1) \vee Cm0(-1) \quad (3.45)$$

Таблиця істинності цієї функції наведено у табл. 3.8. Схема скопійована із середовища схемотехнічного моделювання Multisim, куди була введена з метою перевірки правильності функціонування та побудови передавальних характеристик.

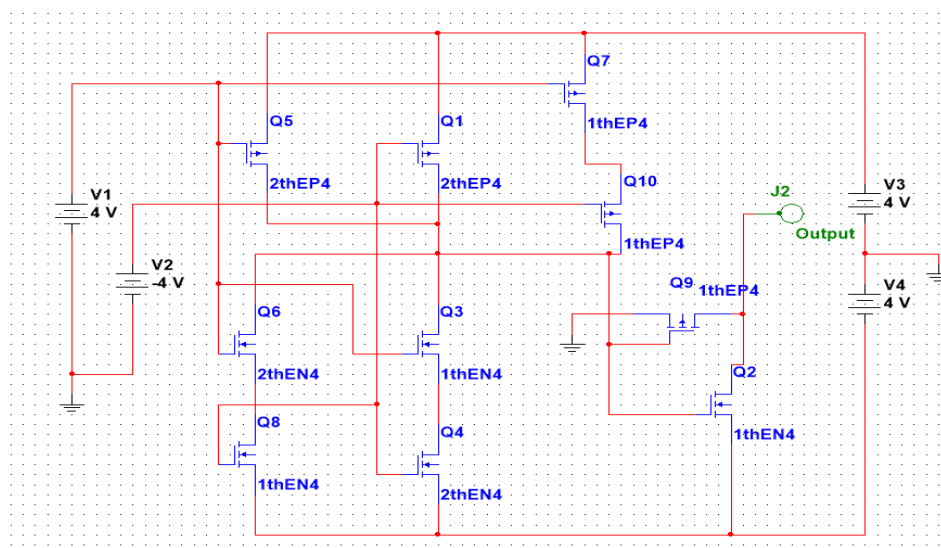


Рис. 3.64 – Синтезована схема, що реалізує функцію перенесення для тривірневого напівсуматора.

У схемі на рис. 3.64 транзистори Q10 – Q6 реалізують функцію, що описується формулою (3.44) та таблицею істинності (табл.3.8). Транзистори Q2 та Q9 реалізують функцію перетворення коду, описану формулою (3.45). У результаті виході схеми здійснюється функція перенесення, задана таблицею істинності (табл.3.8). Для даної схеми були побудовані передатні характеристики для лінійної зміни напруги на одному з входів в діапазоні від -4 до $+4$ при фіксованих логічних рівнях на іншому вході. Графіки трьох побудов наведено на рис.3.65 – 3.66.

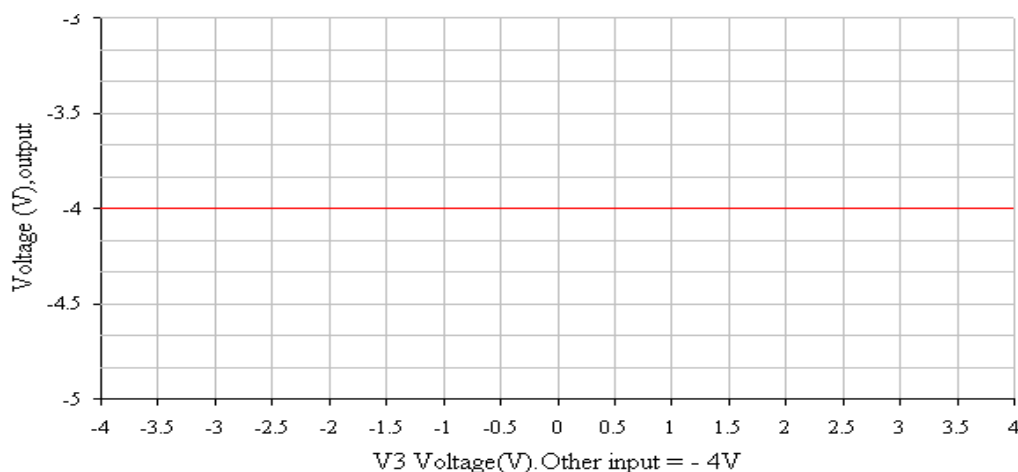


Рис. 3.65 – Передатна характеристика синтезованої схеми.

Перенесення при лінійній зміні напруги на одному з входів у діапазоні від -4 до $+4$ В і фіксованому напрузі на іншому, рівному -4 В, що відповідає логічному рівню «-1».

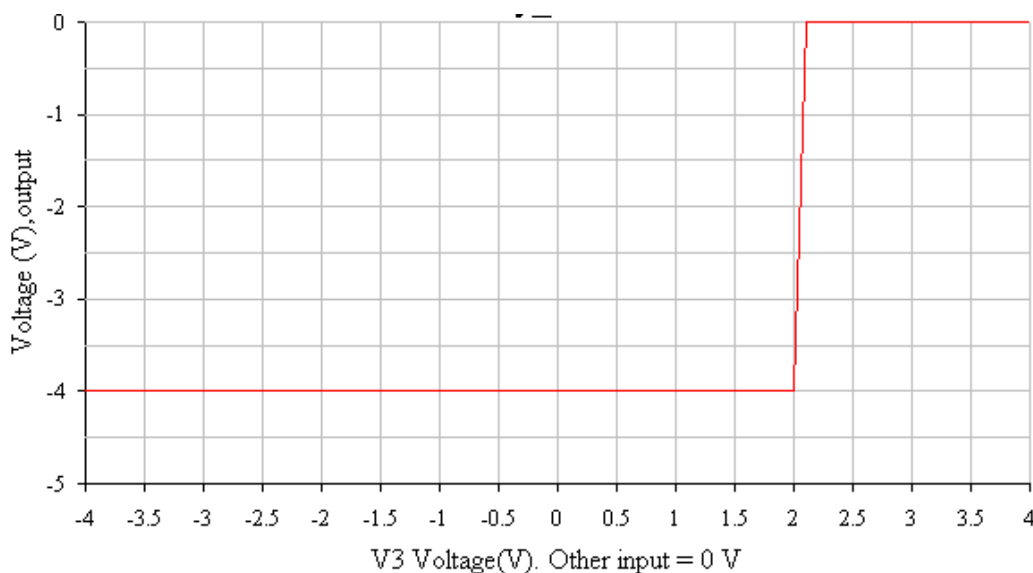


Рис. 3.66 – Передатна характеристика схеми перенесення при лінійній зміні напруги на одному з входів від -4 до $+4$ В.

При фіксованому рівні напруги на іншому вході, що дорівнює 0, що відповідає нульовому логічному рівню. Характеристика побудована за допомогою процедури Parameter sweep для кроку дискретного збільшення вхідної напруги на одному з входів, що дорівнює 0,1 Ст.

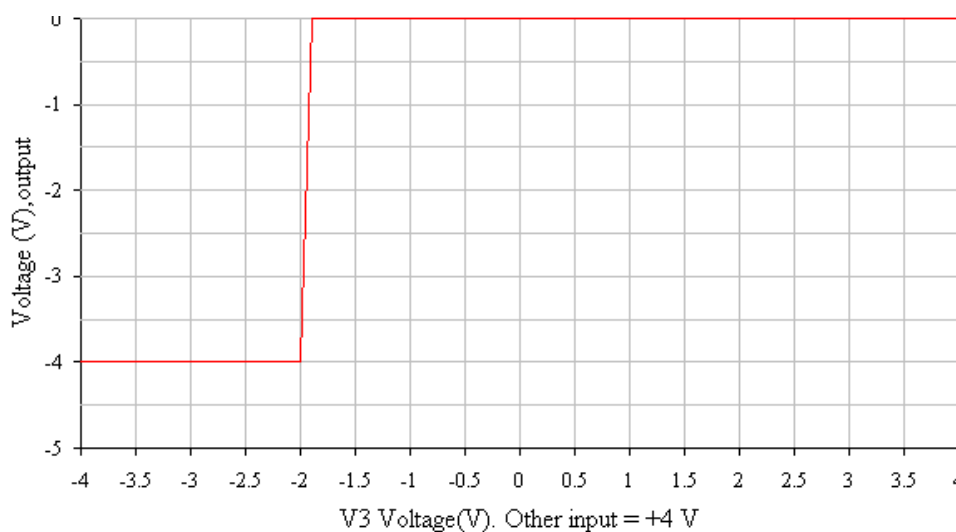


Рис. 3.67 – Передатна характеристика схеми перенесення при лінійній зміні напруги на одному з входів від -4 до $+4$.

При фіксованому рівні напруги на іншому вході, що дорівнює $+4\text{ В}$, що відповідає логічному рівню «+1». Для кроку дискретного збільшення вхідної напруги на одному з входів, що дорівнює $0,1\text{ Ст}$.

Передавальні показники, наведені на рис. 3.65 – 3.67, показують, що робота синтезованої схеми відповідає таблиці істинності (табл. 3.8), схема має чіткий поріг перемикавання та неспотвореність рівнів вихідних логічних сигналів.

У результаті, об'єднавши дві схеми, що виконують функцію додавання по модулю три і функцію перенесення у старший розряд, ми отримуємо трирівневий напівсуматор, що працює в трійковій системі числення і має два входи для доданків і два виходи: один генерує значення суми, інший - значення перенесення. Даний напівсуматор працює з числами в їх абсолютному представленні без урахування знака числа.

Схематичне уявлення трирівневого напівсуматора наведено на рис. 3.68. На ній входи A і B – для сумованих трійкових чисел, вихід S – вихід їх суми ($S = \text{sum}$), вихід C – значення перенесення ($C = \text{Carry}$).

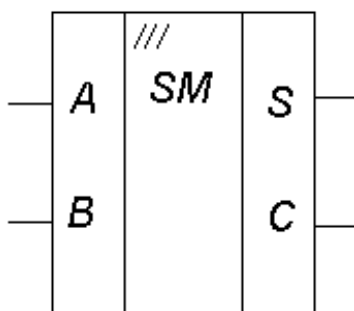


Рис. 3.68 Схематичне уявлення трирівневого напівсуматора. A і B – входи доданків, S – вихід суми, C – вихід перенесення.

Таблиця 3.9

Таблиці істинності для виходів суми S та перенесення C троїчного напівсуматора у коді $-1, 0, 1$.

S	-1	0	1	C	-1	0	1
-1	-1	0	1	-1	-1	-1	-1
0	0	1	-1	0	-1	-1	0
1	1	-1	0	1	-1	0	0

Таблиця 3.10

Таблиці істинності для суми за модулем три та функцією перенесення в трійковій системі числення при поданні чисел цифрами $0, 1, 2$.

S	0	1	2	C	0	1	2
0	0	1	2	0	0	0	0
1	1	2	0	1	0	0	1
2	2	0	1	2	0	1	1

Таблиці істинності, що описують роботу трирівневого напівсуматора в прийнятому коді $-1, 0, 1$ наведені в таблиці 3.10.

У таблиці істинності 19 символи $-1, 0, 1$ кодують цифри $0, 1, 2$ трійкової системи. Дані таблиці еквівалентні таблицям, наведеним у табл.20 при поданні чисел у трійковій системі числення цифрами $0, 1, 2$.

Для того, щоб синтезувати багаторозрядні суматори, необхідно мати схему повного однорозрядного суматора, який повинен мати три входи – для введення першого доданку, для введення другого доданку і значення перенесення з попереднього розряду. Тоді за допомогою каскадного з'єднання

однорозрядних суматорів можна скласти суматор двох трійних чисел будь-якої розрядності. Повний однорозрядний суматор можна синтезувати безпосередньо із базових логічних трирівневих компонентів, проте такий синтез буде складним. Найпростіше з погляду синтезу використовувати вже готову схему троїчного напівсуматора.

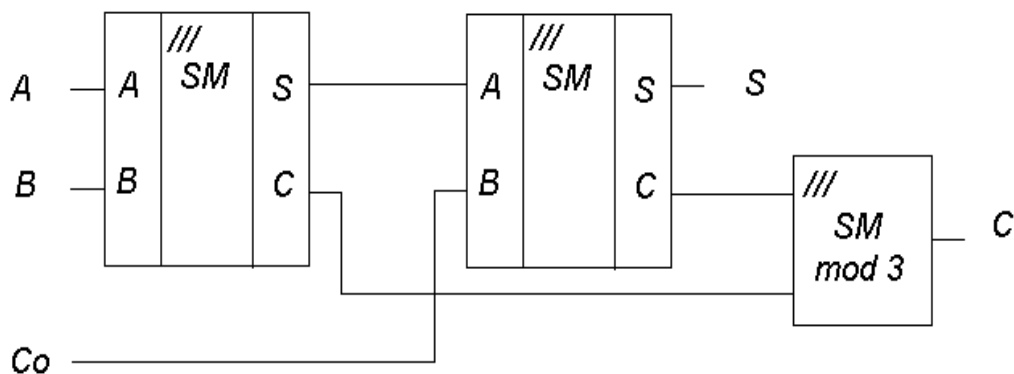


Рис.3.69 – Схема повного потрійного суматора.

Використовуючи два трійних напівсуматора і схему додавання по модулю три, можна реалізувати схему повного суматора, що підсумовує три однорозрядні числа і має два виходи - вихід старшого розряду, який є також і виходом переносу в наступний розряд при каскадному з'єднанні таких суматорів, і молодшого виходу, що є розрядом суми при каскадному з'єднанні повних трирівневих суматорів. Цей варіант суматора наведено на рис. 3.70.

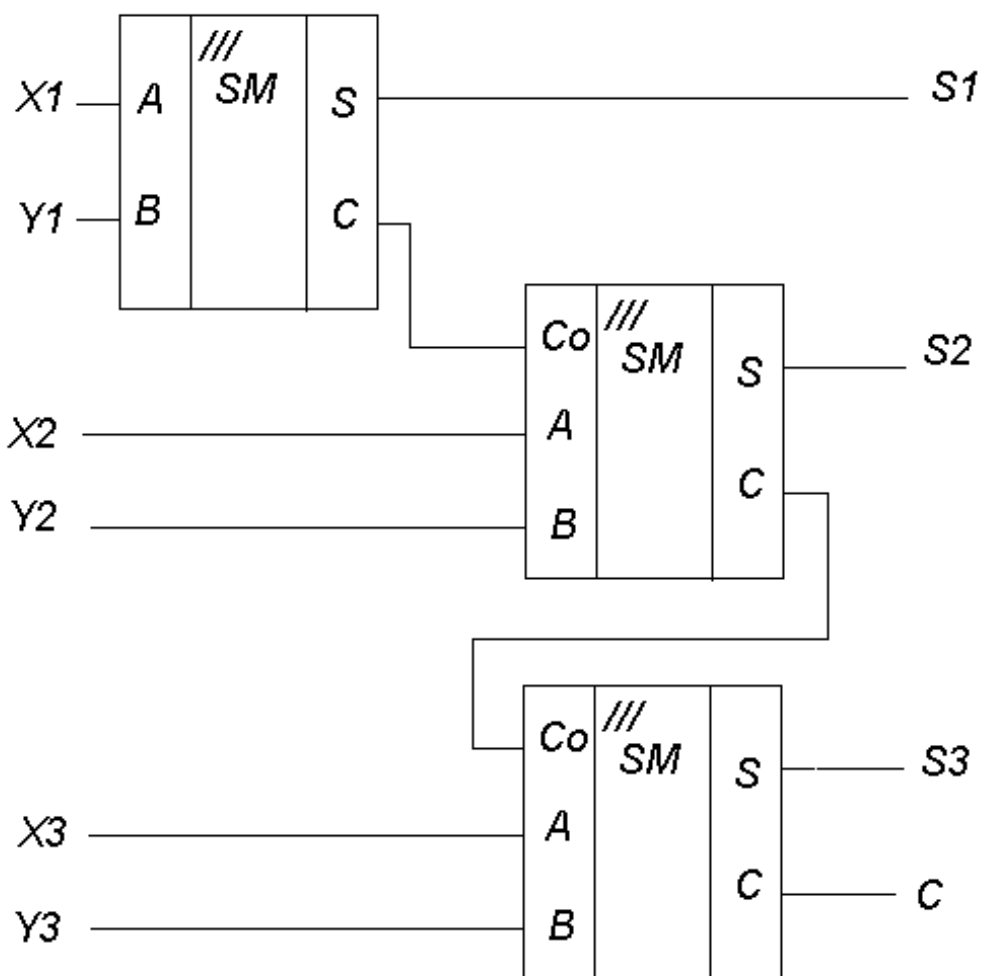


Рис.3.70 – Каскадне з'єднання трійкових суматорів для підсумовування двох трирозрядних чисел у трійковій системі числення.

Усі розглянуті вище суматори, побудовані на розробленому напівсуматорі, оперують із трійковими числами, представленими в абсолютному форматі. Але завдяки тому, що основа системи числення непарна, у трійковій системі можливе симетричне щодо нуля розташування цифр: -1, 0, 1, з чим пов'язана цінна властивість природності уявлення відносних чисел.

Наявність позитивної та негативної цифр дозволяє представляти безпосередньо як позитивні, і негативні числа. При цьому немає необхідності в спеціальному розряді знака - знак числа визначається знаком старшої цифри

числа: якщо вона позитивна, то і число позитивно, якщо негативна, то і число негативно. Для виконання арифметичних операцій з відносними числами, представленими у такому форматі, не треба вводити додатковий (або зворотний) код. Усі дії над числами, представленими в трійчній системі числення з цифрами $-1, 0, 1$, виконуються з урахуванням знаків чисел. Для зміни знака числа треба змінити знаки його цифр, тобто. проінвертувати кожен розряд числа. Віднімання зводиться до додавання з попередньою зміною знака (інвертуванням) віднімається.

Таблиця 3.11

Подання відносних десяткових чисел у трійковій системі для трьох трійних розрядів.

Десятичне	Трійкове відносне			Десятичне	Трійкове відносне		
	x3	x2	x1		x3	x2	x1
-13	-1	-1	-1	0	0	0	0
-12	-1	-1	0	1	0	0	1
-11	-1	-1	1	2	0	1	-1
-10	-1	0	-1	3	0	1	0
-9	-1	0	0	4	0	1	1
-8	-1	0	1	5	1	-1	-1
-7	-1	1	-1	6	1	-1	0
-6	-1	1	0	7	1	-1	1
-5	-1	1	1	8	1	0	-1
-4	0	-1	-1	9	1	0	0
-3	0	-1	0	10	1	0	1
-2	0	-1	1	11	1	1	-1
-1	0	0	-1	12	1	1	0
				13	1	1	1

Для того, щоб оперувати числами, представленими в відносному форматі, як у наведеній вище таблиці, необхідні суматори, побудовані на схемах, що відрізняються від синтезованої схеми додавання по модулю три і схеми перенесення, синтезованої для напівсуматора чисел, представлених в абсолютному форматі. Виходячи з таблиці подання відносних чисел у трійковій системі (табл.3.11), складемо таблиці істинності для напівсуматора, що оперує відносними числами. Дані таблиці для суми відносних чисел у трійковій системі та функції перенесення наведені у таблиці

Таблиця 3.12

Таблиці істинності для напівсуматора

Sr	-1	0	1	Cr	-1	0	1
-1	1	-1	0	-1	-1	0	0
0	-1	0	1	0	0	0	0
1	0	1	-1	1	0	0	1

Оперує числами, представленими у відносному форматі з урахуванням знака, у трійковій системі числення для виходу значення суми Sr та виходу значення перенесення Cr у старший розряд (індекс r означає «відносний» - relative).

Для функції обчислення суми відносних чисел Sr можна синтезувати свою трирівневу схему, але можна використовувати вже синтезовану схему напівсуматора для чисел, представлених в абсолютному форматі, тобто. схему додавання за модулем три. Таблиця істинності для функції суми відносних чисел Sr виходить з таблиці істинності для суматора по модулю три (див табл. 19 для S), якщо -1 перетворити на 1 , 0 перетворити на -1 і 1 перетворити на 0 ($-1 \rightarrow 1$; $0 \rightarrow -1$; $1 \rightarrow \rightarrow 0$). Така операція перетворення логічних значень є ні що інше, як подвійне циклічне заперечення чи лівий цикл, схема якого синтезована розділ 6.5 і наведено на рис. 42. Таким чином, схема функції сумування чисел у відносному форматі Sr може бути отримана, якщо вихід схеми додавання по модулю три подати на схему, що реалізує лівий цикл, що показано на рис. 76. Функцію Sr можна записати як

$$\leftarrow \leftarrow \text{-----} \\ Sr = S = A \oplus B \pmod{3} \quad (3.46)$$

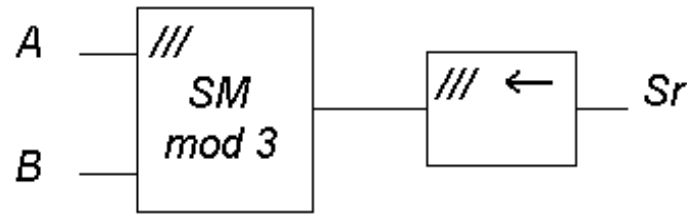


Рис. 3.71 – Отримання функції суми трійкових чисел, представлених у відносному форматі.

Із схеми додавання за модулем три та схеми, що здійснює функцію лівого циклу (подвійного циклічного заперечення).

Синтезуємо схему, що здійснює функцію перенесення C_r таблиці істинності для неї (табл. 3.11).

Якщо синтезувати схему безпосередньо за таблицею істинності, наведеною у таблиці.3.12, схема буде містити характеристичні функції і складатися з 16 польових транзисторів. Схема вийде простіший і міститиме менше елементів, якщо її синтезувати для інверсії функції перенесення, і отриману функцію інвертувати з допомогою трирівневого інвертора. У такому варіанті вся схема міститиме 12 польових транзисторів.

Схемотехнічна формула для інверсії функції переносу записується так:

$$\neg C_r = A1(-1) \& B1(-1) \vee A0(-1,0) \& B0(0,1) \vee A0(0,1) \& B0(-1,0) \vee A-1(1) \& B-1(1) \quad (3.47)$$

Якщо вихід схеми, синтезованої по схемотехнической формулі подати на трирівневий інвертор, то виході інвертора буде реалізовуватися функція перенесення суми відносних трійних чисел C_r , чия таблиця істинності наведено у таблиці.

Синтезована схема перенесення C_r наведено на рис. 3.71. Більш детальний аналіз синтезу схеми за формулою та дослідження такої схеми

будуть наведені в розділі, присвяченому питанням застосування трирівневих пристроїв у системах обробки недостовірної інформації.

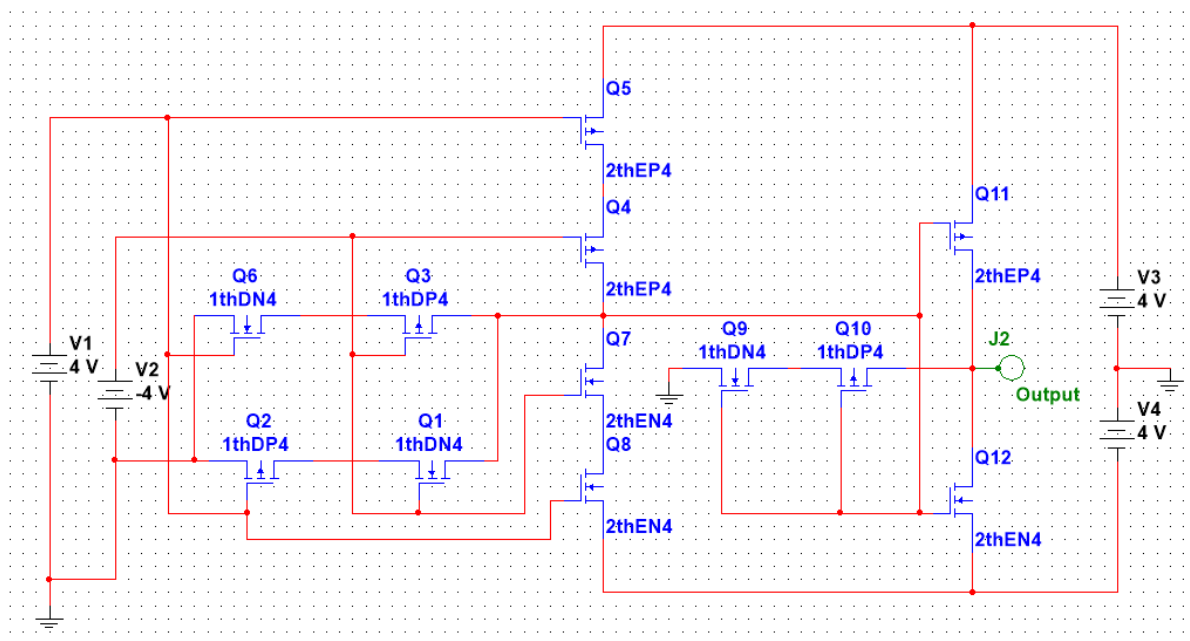


Рис. 3.72 – Синтезована схема, що здійснює функцію перенесення до старшого розряду під час підсумовування трійкових чисел, представлених у відносному форматі.

На транзисторах Q1 – Q8 виконано схему для інверсії цієї функції, але в транзисторах Q9 – Q12 реалізований інвертор.

Для схеми, зображеної на рис. 3.71 в середовищі програми схемотехнічного моделювання Multisim були побудовані передатні характеристики при зміні напруги на одному з входів і фіксованих логічних рівнях на іншому. Графіки двох побудов наведено на рис. 3.73 ,3.74.

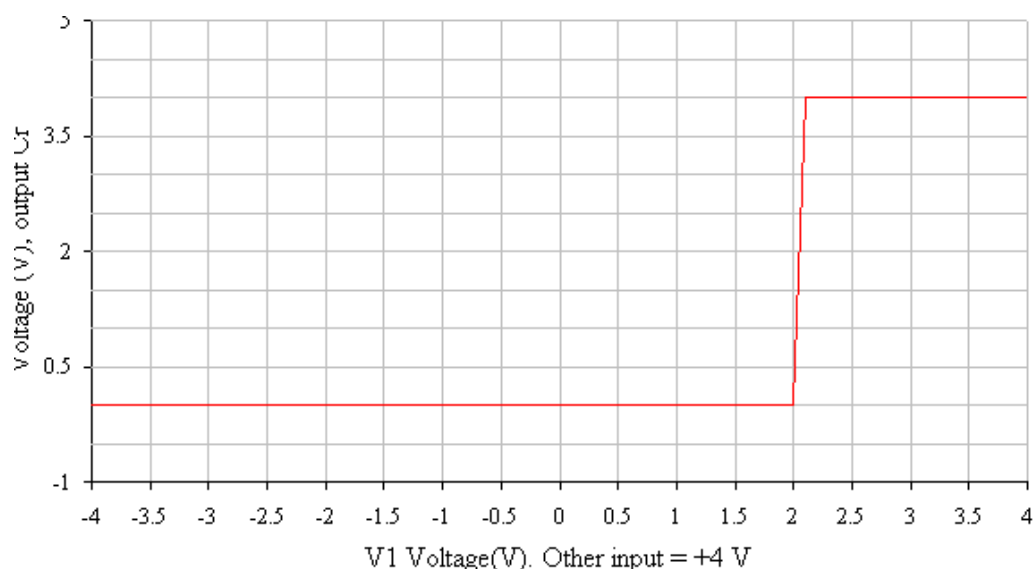


Рис. 3.73 – Передатна характеристика схеми переносу Cr при фіксованому напрузі на одному з входів, що дорівнює +4 В.

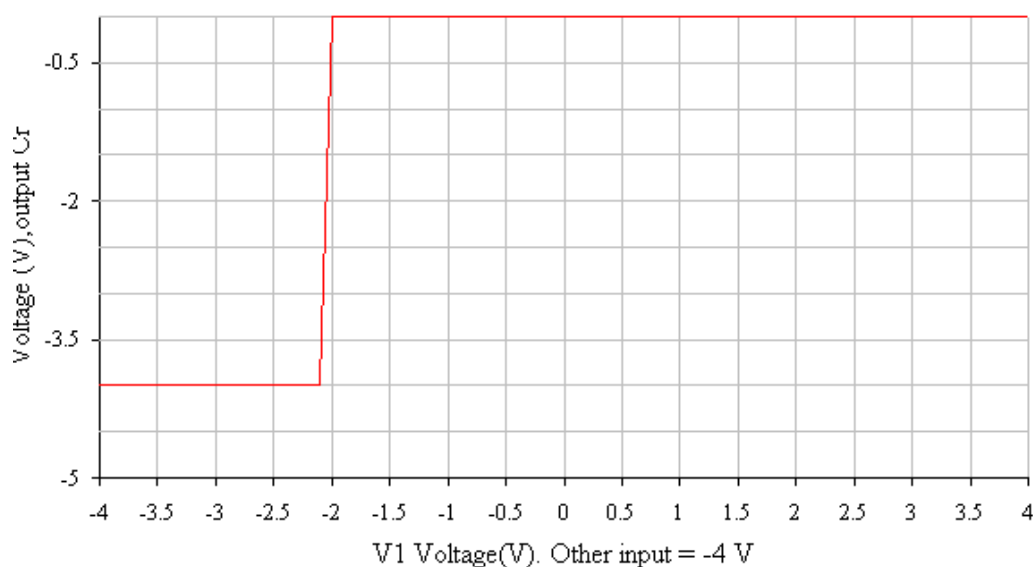


Рис. 3.74 – Передатна характеристика схеми перенесення.

Для лінійної зміни напруги напруги на одному з входів в діапазоні від – 4 до + 4 В при фіксованому рівні напруги на іншому, рівному – 4 В (рівень логічної «-1»).

Характеристики наведені на рис. 3.77 – 3.78, показують, що схема має чіткий поріг перемикання та неспотвореність логічних рівнів на виході.

Таким чином, поєднавши схеми, наведені на рис. 3.71 (схема для суми двох трійкових чисел Sr у відносному форматі) та на рис.3.72 (схема перенесення Cr), ми отримуємо напівсуматор для трійкових чисел, представлених у відносному форматі. Схема повного суматора для відносних чисел подібна до схеми повного суматора для абсолютних трійних чисел і наведена на рис.3.75.

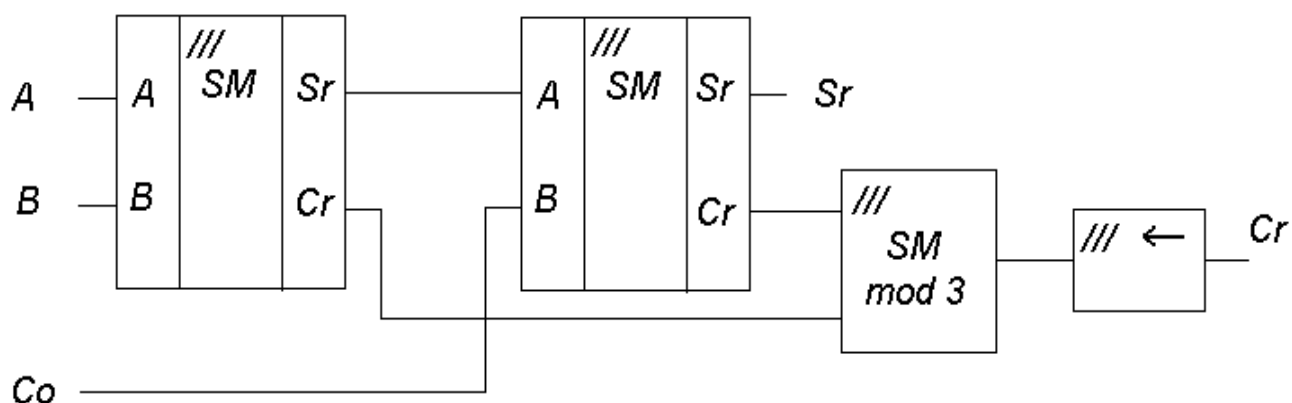


Рис. 3.75 – Повний потрійний суматор відносних чисел.

Каскадне з'єднання суматорів трійних чисел, представлених у відносному форматі, здійснюється аналогічно до з'єднання суматорів чисел в абсолютному поданні, як показано на рис. 3.76.

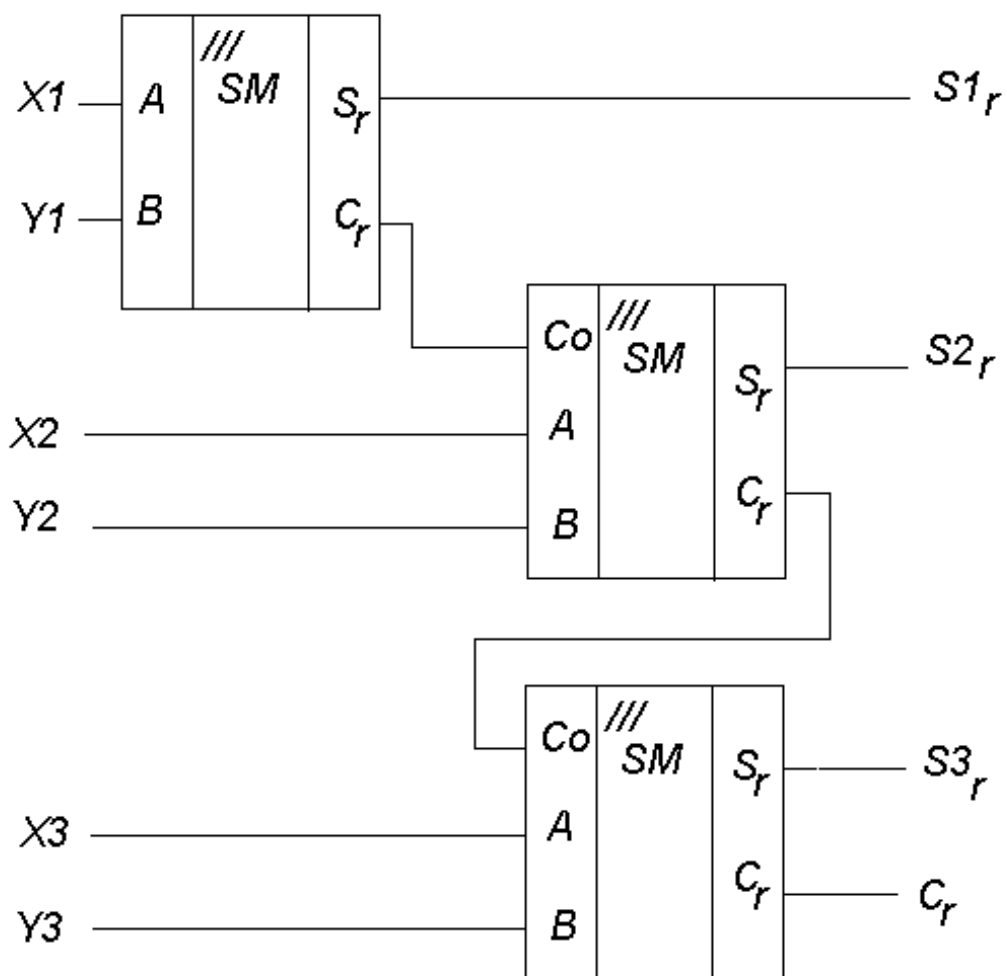


Рис.3.76 – Каскадне з'єднання суматорів для здійснення арифметичних операцій над відносними числами.

Представленими трирозрядним трійним кодом. Підсумовування провадиться з урахуванням знака чисел. Віднімання чисел зводиться до додавання з попередньою зміною знака (інвертуванням) віднімається шляхом інверсії кожного його розряду. Каскад складається з напівсуматора та двох суматорів.

3.4 Висновки до розділу 3

В цьому розділі було проведено розробку та моделювання компонентів та базових елементів трійкової логіки. Було створено в програмі схемотехнічного моделювання транзистори з певними заданими характеристиками які в свою чергу були використанні для створення базових елементів трійкової логіки. Також проведено моделювання роботи базових елементів в схемотехнічному програмному середовищі зняті осцилограми в різних точках схем. Також ми дослідили що реалізація базових елементів трійкової логіки дійсно можлива про що свідчать вище наведені схеми та результати моделювання. За допомогою розроблених базових елементів був створений арифметико-логічний пристрій а саме суматор характеристики якого також були дослідженні.

ВИСНОВКИ

В роботі було проведено огляд і аналіз трійкової логіки. Також проведено її порівняння з іншими та обґрунтовано необхідність створення базових елементів трійкової логіки. Було розроблено базові елементи та арифметико-логічний пристрій з використанням трійкової логіки.

Результатом цього дослідження є змодельовані схеми базових елементів згідно функцій яку вони мають виконувати. Трирівневі пристрої, що виконують функції тризначної логіки, можуть знайти застосування при побудові арифметичних та логічних пристроїв. Елементи тризначної логіки особливо цікаві з погляду застосування в інформаційних системах, що працюють з не повністю певними даними і з суперечливими даними.

Розглянемо залежність апаратних витрат від заснування системи числення для n розрядів двійкових та трійкових пристроїв. Нехай φ_2 – функція, що характеризує кількість обладнання, необхідного для побудови одного розряду бінарного пристрою, а φ_3 – характеризує кількість устаткування, необхідного для побудови одного розряду трирівневого пристрою. Витрати будуть пропорційні числу розрядів: $\varphi_2 * n$ і $\varphi_3 * n$, але кількість чисел, яке може бути передане за допомогою n розрядів, дорівнює n^2 для двійкової системи та n^3 для трійкової системи числення. Тоді кількість витрат за число при n розрядах визначиться як:

$$\varphi_2 * n / n^2 = \varphi_2 / n - \text{для двійкових пристроїв,}$$

$$\varphi_3 * n / n^3 = \varphi_3 / n^2 - \text{для трійкових пристроїв.}$$

Таким чином, навіть якщо $\varphi_3 > \varphi_2$, у багаторозрядних пристроях витрати можуть бути меншими у трирівневих системах, ніж у бістабільних.

Знайдемо залежність між кількостями двійкових та трійкових розрядів, необхідними для подання чисел з однаковою точністю. Нехай n – кількість двійкових розрядів, необхідні представлення деякого максимального числа N ,

а m – кількість трійкових розрядів, необхідні представлення тієї самої числа N . Тоді

$$N = 2^n = 3^m \quad (4.1)$$

Звідки

$$n = \log_2 N = \log_2 3^m = m * \log_2 3 = 1,6 * m \quad (4.2)$$

Тобто для подання деякого числа в двійковій формі потрібно в 1,6 разів більше розрядів, ніж для подання того ж числа в потрійній формі. Звідси випливає, що економія обладнання можлива в тому випадку, якщо регістровий трирівневий одноразрядний пристрій не перевищуватиме за апаратними витратами еквівалентний бінарний пристрій більш ніж на 60%.

Таким чином, потрійне подання інформації має такі переваги: економніше кодування; зручність оперування з відносними числами – не потрібен розряд знака числа, зміна знака числа здійснюється простим інвертуванням кожного його розряду, відсутня необхідність застосування спеціальних кодів для віднімання чисел внаслідок інваріантності правил додавання та віднімання щодо знаків чисел; відсутність необхідності округлення, що з тим, що найкраще при заданій кількості старших розрядів наближення числа досягається простим відкиданням його молодших розрядів. Виключення перетворення кодів може призвести до спрощення арифметичних пристроїв та скорочення часу виконання арифметичних операцій.

Загалом, застосування недвійкового кодування у пристроях переробки дискретної інформації може призвести до економії обладнання, необхідного для їх побудови, збільшення швидкодії, підвищення надійності, а також спрощення структури пристроїв і більш ефективного їх використання в складних інформаційних системах. Багаторівневі схеми дозволяють скоротити

кількість логічних елементів і, що особливо важливо, міжкомпонентних сполук, проблема яких гостро постає при проектуванні.

Трирівневі пристрої, розглянуті в дипломній роботі, побудовані на польових транзисторах з вбудованим та індукованим каналом різної провідності та відмінними пороговими напругами. Використання еліонної технології дозволяє на одному кристалі створювати МОН-структури з індукованими і вбудованими каналами, зі скільки завгодно малими пороговими напругами зі стійким відсотком виходу придатних кристалів у процесі виробництва. Малі порогові напруги транзисторів дозволяють знизити напругу живлення схем. Незалежне підганяння порогової напруги різних транзисторів одного кристала може здійснюватися шляхом іонної імплантації крізь підзатворний оксид іонів домішок.

Перелік використаних джерел

1. Багатозначні елементи та структури. - Збірник статей / Под ред. В.П. Сигірського. - М.: Радянське радіо, 1967.
2. Брусенцов Н. П., Маслов С.П., Розін В.П., Тишуліна А.М. Мала цифрова обчислювальна машина "Сетунь". - М.: Изд.МГУ, 1965.
3. Іваськів Ю.Л. Принципи побудови багатозначних фізичних схем. - Київ.: Науова думка, 1971.
4. Поспелов Д.А. Логічні методи аналізу та синтезу схем. Вид. 3-тє, перераб. та дод. - М.: Енергія, 1974.
5. Кармазінський О.М. Синтез важливих схем цифрових елементів на МОН-транзисторах. - М.: Радіо і зв'язок, 2009.
6. Алексенко А.Г., Шагурін І.І. Мікросхемотехніка: Навчальний посібник для вузів. - 2-ге вид., перероб. та дод. - М.: Радіо і зв'язок, 1990.
7. Пухальський Г.І., Новосельцева Т.Я. Цифрові пристрої: Навчальний посібник для втузів. - СПб.: Політехніка, 2006.
8. Кухарєв Г.А., Шмерко В.П., Зайцева О.М. Алгоритми та систолічні процесори для обробки багатозначних даних. – Мінськ: Наука і техніка, 1990.
9. Пономарьов М.Ф., Конопльов Б.Г. Конструювання та розрахунок мікросхем та мікропроцесорів: Навч. посібників для вузів. - М.: Радіо і зв'язок, 1986.
10. Ігумнов Д.В., Громов І.С. Експлуатаційні параметри та особливості застосування польових транзисторів. - М.: Радіо і зв'язок, 2001.
11. Кондаков Н.І. Логічний словник-довідник - М.: Наука, 1975.