

SECTION 5. ELECTRONICS

DOI: 10.46299/ISG.2025.MONO.TECH.1.5.1

5.1 Схемотехнічна реалізація операцій над логіко-часовими функціями

Сучасний розвиток обчислювальної техніки, цифрової схемотехніки та систем автоматизованого управління все частіше стикається із завданнями, що вимагають опрацювання складних логіко-часових залежностей [133, 134, 135].

Використання методів математичного моделювання в поєднанні з комп'ютерними технологіями для розв'язання інженерних і наукових завдань дозволяє значно підвищити ефективність складних процесів, таких як проектування технічних систем, розпізнавання образів, обробка великих масивів даних та управління динамічними об'єктами. Завдяки високій швидкості обчислень, можливості обробки складних нелінійних залежностей та імітації різних сценаріїв, математичне моделювання із застосуванням сучасних комп'ютерних засобів стало основним інструментом аналізу та оптимізації складних динамічних систем. Це дозволяє не лише глибше розуміти поведінку об'єктів і процесів у різних умовах, а й прогнозувати їхній розвиток, що є особливо важливим у науці, техніці та промисловості.

Сучасні комп'ютери, за винятком спеціалізованих систем, розроблених для керування технологічними процесами, є універсальними обчислювальними пристроями. Ця універсальність дозволяє використовувати їх у широкому спектрі завдань, але водночас вимагає створення спеціалізованих алгоритмів, здатних ефективно розв'язувати конкретні проблеми. У багатьох випадках розроблені алгоритми не повною мірою враховують динамічний характер змін, що відбуваються в керованих системах або досліджуваних процесах. Це може призводити до неточностей у моделюванні, обробці даних та прогнозуванні їхнього подальшого розвитку.

Особливо це актуально для складних систем, де процеси тісно взаємопов'язані та змінюються у часі під впливом різноманітних факторів. Неповний облік цих взаємодій може стати причиною накопичення помилок,

некоректних оцінок і неправильних рішень. Для мінімізації таких ризиків необхідне застосування математичного апарату, який здатний адекватно описувати поведінку складних, взаємозалежних процесів у часовому просторі. Використання методів математичного моделювання, зокрема диференціальних рівнянь, стохастичних моделей та нейромережевих підходів, дозволяє більш точно аналізувати, прогнозувати та оптимізувати складні динамічні системи.

На рівні обчислювальних пристроїв час відіграє роль критерію для впорядкування послідовності виконання операцій або програм, які обробляють дані. При цьому його вплив є неявним, оскільки він визначає логіку та порядок виконання інструкцій без прямої фіксації абсолютних часових значень.

Водночас на рівні окремих апаратних вузлів час є ключовим параметром, який враховується безпосередньо і відіграє критичну роль у забезпеченні узгодженої роботи цифрових компонентів. У цьому контексті він представлений у вигляді тактових сигналів або синхронізуючих імпульсів, які задають ритм обчислювальних процесів та гарантують коректне виконання операцій. Тактові імпульси координують роботу окремих блоків, синхронізуючи обмін даними між процесором, пам'яттю та іншими периферійними пристроями, що забезпечує узгодженість і стабільність роботи системи загалом.

Логіко-часові функції (ЛЧФ), які поєднують у собі властивості класичних булевих функцій із динамічними характеристиками, є основою для моделювання процесів, аналізу складних технічних систем і побудови ефективних апаратних рішень для цифрової обробки інформації.

З огляду на зростаючі вимоги до швидкості, точності й енергоефективності обчислень, питання оптимальної реалізації операцій над ЛЧФ у схемотехнічній площині стають особливо актуальними. Це стосується не лише обробки даних у системах реального часу, а й моделювання часових процесів у цифрових пристроях, таких як комутаційні схеми, синхронізовані логічні блоки та модулі програмованих логічних інтегральних схем (FPGA).

У контексті схемотехнічної реалізації операцій над ЛЧФ перед дослідниками постають такі ключові завдання:

1. **Формалізація ЛЧФ та вибір методів їх опрацювання:** це включає опис функцій у математичних моделях, визначення алгоритмів виконання основних операцій і аналіз їх оптимальності.

2. **Проектування базових схемотехнічних блоків:** до них належать модулі, що реалізують базові операції над ЛЧФ, зокрема логічні операції, операції затримки, складання часових інтервалів, диференціювання, інтегрування тощо.

3. **Оптимізація апаратної реалізації:** мінімізація витрат апаратних ресурсів, енергоспоживання та часових затримок. Особлива увага приділяється вибору архітектури: від традиційних жорстко заданих схем до гнучких програмованих рішень.

4. **Практична реалізація та тестування:** створення прототипів, перевірка їх працездатності на реальних завданнях і оцінка ефективності.

В даній роботі розглянемо проектування базових схемотехнічних блоків, які базуються на формалізації ЛЧФ та операцій над ними, розглянуті в роботах автора [136 – 139].

Багатозначними логіко-часовими функціями (БЛЧФ) будемо називати такі логічні k -значні ($k \geq 2$) функції, які змінюються в дискретному часі і можуть набувати одне із значень від 0 до $k-1$ на кожному Δ -інтервалі, що входить до якогось фіксованого кінцевого часового інтервалу існування змінних, за межами цього інтервалу функція дорівнює нулю [136]. Моменти початку часових інтервалів існування t_i та їх тривалість T_i дискретні. Значення t_i та T_i називають часовими координатами змінних.

Аналіз можливих варіантів БЛЧФ дозволив виділити три основних класи функцій (повна система):

- 1) клас БЛЧФ, що між двома нулями приймають стале значення (елементарні ЛЧФ);
- 2) клас БЛЧФ, які мають n часових координат, відрізки існування яких не перетинаються;
- 3) клас монотонних функцій.

Наприклад, елементарна БЛЧФ (див. рис.1), область визначення якої складається лише з одного відрізка існування, в загальному вигляді має вид:

$$f(t, t_1, T_1, a_1) = \begin{cases} a_1, & \text{якщо } t_1 \leq t \leq t_1 + T_1 \\ 0, & \text{якщо } t_1 + T_1 < t < t_1 \end{cases}, \quad (1)$$

де t – поточне дискретне значення параметру;

T_1 – інтервал існування функції;

a_1 – одне із k значень функції

t_1 – початок часового інтервалу

$t_1 + T_1$ – кінець часового інтервалу.

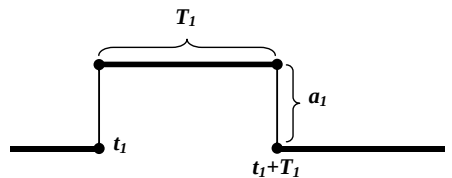


Рисунок 1. Елементарна логіко-часова функція

На рис. 2 наведено можливий варіант монотонної БЛЧФ (а – зростаючої, б – спадної).

Можна показати, що функції другого та третього класів можна подати у вигляді накладання (суперпозиції, логічного додавання) БЛЧФ першого класу.

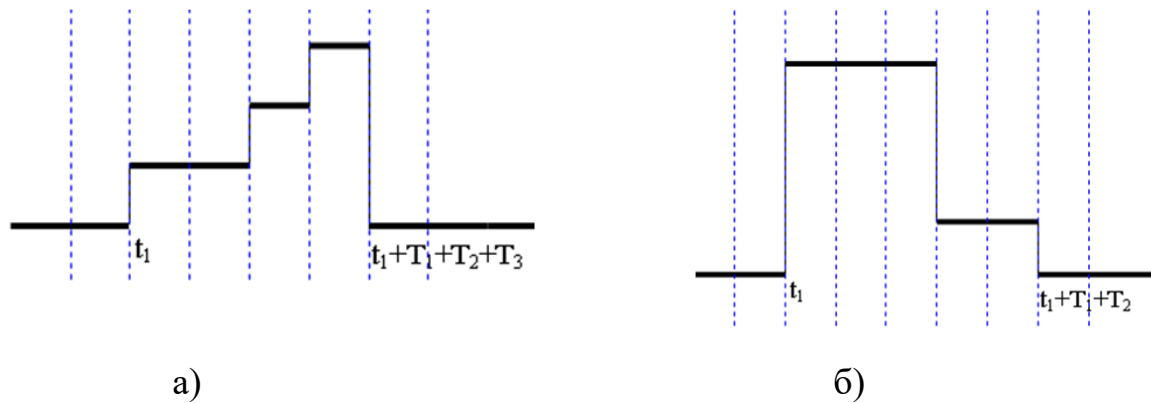


Рисунок 2. Можливі варіанти монотонної ЛЧФ : а) зростаюча ЛЧФ, б) спадна ЛЧФ

У працях [134, 138-141] було досліджено основні логічні операції над багатозначними логіко-часовими функціями (БЛЧФ), поданими в індексній формі, зокрема і операцію нерівнозначності Θ , яку в індексному позначенні можна подати так:

$$\left({}_k x_{t_x}^{T_x} \Theta {}_k y_{t_y}^{T_y} \right) = \bigvee_{i=0}^{T_{\max}-1} \left({}^x a_i \Theta {}^y c_i \right) \Big|_k (x \Theta y)_{\min(t_x, t_y)}^{T_{\max}}, \quad (2)$$

де W – оператор впорядкування за часом та розбиття значень аргументів БЛЧФ на одиничні Δ -інтервали [140].

На рис. 3 наведено графічні результати операції додавання по модулю 2 або нерівнозначності.

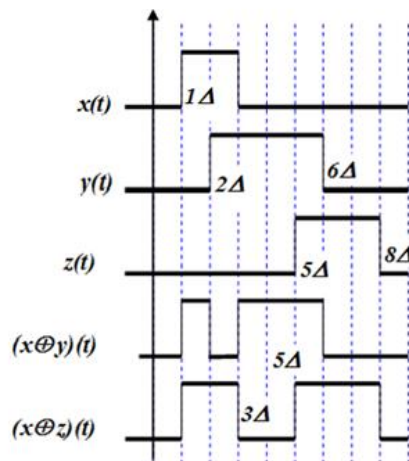


Рисунок 3. Двійкова нерівнозначність

У роботі [136] операцію диференціювання здійснювали графічним методом. Символьний підхід до обчислення похідної виявився надмірно складним, неочевидним і незручним для практичного використання.

Похідна БЛЧФ є також БЛЧФ, що показує величину зміни поточного (в момент t) значення функції відносно будь-якого іншого (в момент $t+s$) попереднього чи наступного значення функції, що віддалене від поточного на s Δ -інтервалів [141].

Похідна БЛЧФ, при $dt = \Delta = 1$ визначається формулою:

$$\left[\frac{dz(t)}{dt} = dz(t)|_{dt=1} = z'(t) = z(t) \ominus z(t+1) \right] \quad (3)$$

В цьому випадку похідна і диференціал співпадають. Можна довести, що похідна суперпозиції (логічної суми) логіко-часових функцій дорівнює суперпозиції (логічній сумі) похідних цих функцій.

Якщо $z(t) = x(t) \oplus y(t)$, де $x(t) = x_{t_x}^{T_x}$ та $y(t) = y_{t_y}^{T_y}$, то похідну можна визначити так:

$$z'(t) = x_{t_x}^{'T_x} \oplus y_{t_y}^{'T_y}. \quad (4)$$

Зазначимо, що символ « $\oplus$ » позначає суперпозицію часових відрізків (накладання відрізків на одну часову координату), що еквівалентне булевому логічному додаванню n відрізків (або відліків БЛЧФ) $f(t, (t_1, T_1, a_1 \vee t_2, T_2, a_2 \vee \dots \vee t_n, T_n, a_n))$. Фактично, логічне додавання в останній формулі має сенс суперпозиції значущих інтервалів БЛЧФ на часовій осі. Суперпозиція це семантично ширше значення логічної операції «АБО» і враховуючи, що функція «АБО» в k -значній логіці є вибором більшого, то для позначення логічного додавання, як суперпозиції підінтервалів БЛЧФ, доцільно використати інший символ, зокрема такий, як запропоновано автором.

Наприклад, функцію, зображену на рис. 4

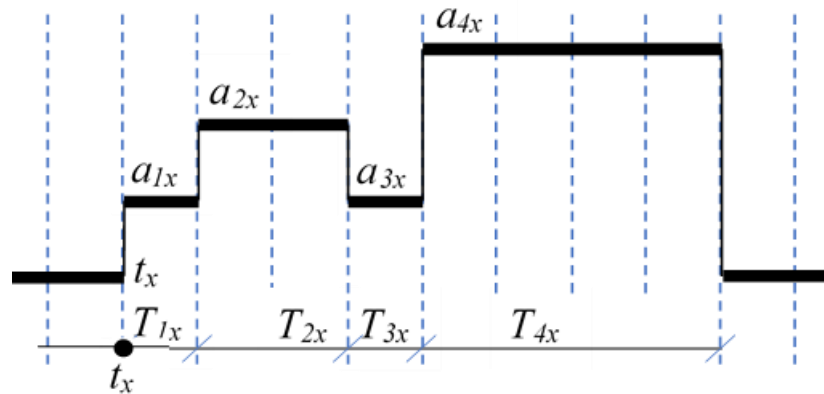


Рисунок 4. Багатозначна логіко-часова функція

аналітично можна подати так:

$$x(t) = a_{1x} x_{t_x}^{T_{1x}} \oplus a_{2x} x_{t_x+T_{1x}}^{T_{2x}} \oplus a_{3x} x_{t_x+T_{1x}+T_{2x}}^{T_{3x}} \oplus a_{4x} x_{t_x+T_{1x}+T_{2x}+T_{3x}}^{T_{4x}}. \quad (5)$$

Багатозначну, k -значну, логіко-часову функцію (БЛЧФ) $x(t)$, для уникнення іншого трактування будемо позначати нижнім попереднім індексом k : ${}_k x(t)$.

В загальному випадку маємо:

$${}_k z(t) = a_1 {}_k z_{t_z}^{T_{a_{1z}}} \oplus a_2 {}_k z_{t_z+T_{a_{1z}}}^{T_{a_{2z}}} \oplus a_3 {}_k z_{t_z+T_{a_{1z}}+T_{a_{2z}}}^{T_{a_{3z}}} \oplus a_N {}_k z_{t_z+T_{a_{1z}}+T_{a_{2z}}+T_{a_{3z}}+\dots+T_{a_{Nz}}}^{T_{a_{Nz}}}, \quad (6)$$

де $a_i \in \forall \{0, 1, 2, \dots, k-1\}$,

Якщо прийняти, що a може дорівнювати «нулю», тоді інтервал існування БЛЧФ може включати в себе нульові підінтервали в залежності від процесу, що описується.

Далі, підінтервали, де співпадають кінець поточного з початком наступного, можна з'єднати і тоді, використовуючи для k -значних функцій індексний запис, для неперервного інтервалу T_x існування функції отримаємо:

$${}_k x(t) = {}^{T_{a_1}, T_{a_2}, T_{a_3}, T_{a_4}, \dots, T_{a_N}}_{a_1, a_2, a_3, a_4, \dots, a_N} {}_k x_{t_x}^{T_x}, \quad (7)$$

де $T_x = T_{a_1} + T_{a_2} + T_{a_3} + T_{a_4} + \dots + T_{a_N}$

Деякі індекси можна опустити, якщо це не викликає інших трактувань. Крім того, з (7) видно, що розміщення значень a_i , в межах інтервалу існування, прив'язані до початку часового інтервалу через тривалості підінтервалів (6). Іншими словами, значення БЛЧФ повністю задають через відліки a_i , значенням початку функції на часовому інтервалі та тривалості підінтервалів.

Якщо необхідно вказати конкретне значення часових координат для a , записуємо їх нижнім та верхнім часовими індексами, наприклад, ${}^x a_{\max(t_x, t_y)+5}^{T_x}$. Тут x вказує приналежність a конкретній змінній, якщо це не ясно з інших параметрів. Для кращого розуміння деяких виразів замість a можна використати іншу букву. t_x вказує початок інтервалу існування БЛЧФ $x(t)$ і може приймати будь-які цілі додатні чи від'ємні значення. T_x описує тривалість інтервалу існування і при дискретному часі є також цілим додатнім числом.

Зауважимо, що БЛЧФ з від'ємним значенням T_x не існують. Від'ємне значення T_x означає зворотній відлік часу, коли закінчення БЛЧФ настає раніше початку.

Фактично індексний запис тривалості амплітуд a_i , k -значних функцій, разом з початком інтервалу існування t_x , повністю характеризують БЛЧФ.

В індексній формі можна одержати значення похідної логіко-часової функції [141]:

$${}_k z'(t) = \left. W_{t=0}^{T_{\max}-1} (a_t \Theta a_{t+1}) \right| z_{t_z}^{T_{\max}} \quad (8)$$

В формулі (8) використаний зсув вправо (затримка). Таку похідну, що використовує наступні значення функції, назвемо «правою». Відповідно «ліва похідна», з попередніми значеннями, матиме вигляд:

$${}_k z'(t) = \left. W_{t=0}^{T_{\max}-1} (a_t \Theta a_{t-1}) \right| z_{t_z}^{T_{\max}} \quad (9)$$

Наприклад, для функції ${}_3 x(t) = 1212101021110220 \big| x_{t_x}^{16}$ маємо відповідно праву ${}_3 x'(t)R = 1111111210012020 \big| x_{t_x}^{16}$ та ліву ${}_3 x'(t)L = 1111111121001202 \big| x_{t_x}^{16}$ похідні. Бачимо, що права похідна це зсунута вліво ліва похідна і навпаки.

Графічний варіант БЛЧФ і її похідних представлено на рис. 5.

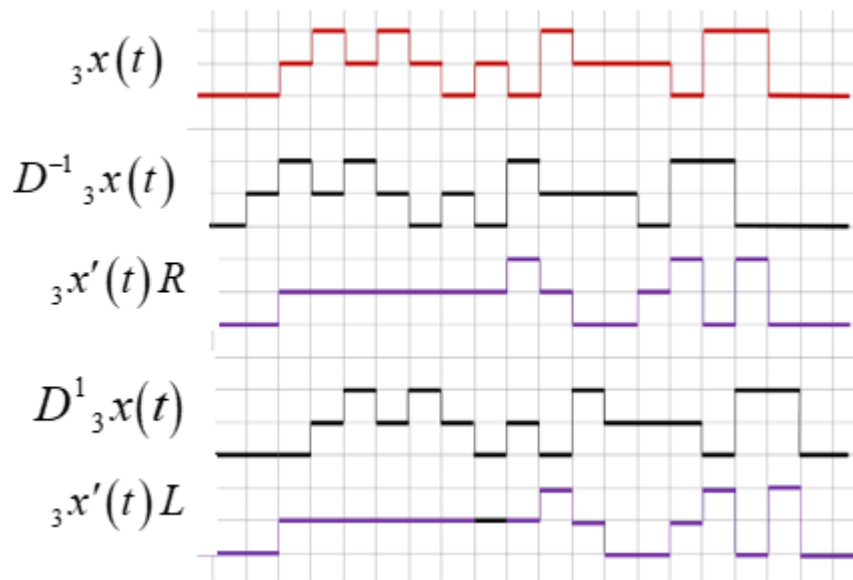


Рисунок 5. Похідна БЛЧФ

Підкреслимо, що операція нерівнозначності при обчисленні похідної, виконується над двома сусідніми розрядами продукуючого слова.

На сьогодні для схмотехнічної реалізації БЛЧФ немає загальноприйнятих k -значних логічних елементів. Найбільш поширеним є підхід, де використовуються двійкові логічні елементи і відповідне двійкове кодування. На такому принципі побудована сучасна флеш пам'ять, коли всередині кристалу використовується k -значність а зовнішній інтерфейс чисто двійковий.

Для прикладу розглянемо знаходження мінімального чи максимального значень із двох функцій при k -значній кон'юнкції чи диз'юнкції. Тут немає сенсу використовувати якісь арифметичні вузли. Простіше синтезувати логічну схему на підставі таблиці істинності, що однозначно задає ці функції.

Нижче розглядається структурна схема операцій кон'юнкції та диз'юнкції чотиризначної (четвіркової) системи з двійковим кодуванням типу «0» – 00; «1» – 01; «2» – 10, «3» – 11 де a, b - вхідні змінні, k - кон'юнкція d - диз'юнкція. Індекси: 1-молодший розряд; 2 -старший.

Синтез схеми зроблено на основі таблиці 1 з відповідною мінімізацією. Кон'юнкція описується виразами:

$$k_2 = b_2 a_2$$

$$k_1 = b_1 a_1 \vee b_2 \bar{a}_2 a_1 \vee \bar{b}_2 b_1 a_2$$

диз'юнкція:

$$d_2 = b_2 \vee a_2 \vee b_1 \bar{a}_1$$

$$d_1 = b_2 b_1 \vee a_2 a_1 \vee \bar{b}_2 a_1$$

Таблиця 1 Таблиця істинності для операцій кон'юнкції та диз'юнкції чотиризначної логіки

	b_2	b_1	a_2	a_1	k_2	k_1	d_2	d_1
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	0	1	0
5	0	1	0	1	0	1	0	1
6	0	1	1	0	0	1	1	0
7	0	1	1	1	0	1	1	1
8	1	0	0	0	0	0	1	0
9	1	0	0	1	0	1	1	0
10	1	0	1	0	1	0	1	0
11	1	0	1	1	1	0	1	1
12	1	1	0	0	0	0	1	1
13	1	1	0	1	0	1	1	1
14	1	1	1	0	1	0	1	1
15	1	1	1	1	1	1	1	1

Відповідні функціональні схеми на два входи і їх умовне позначення для кон'юнкції та диз'юнкції представлені на рис.6 та рис. 7 відповідно.

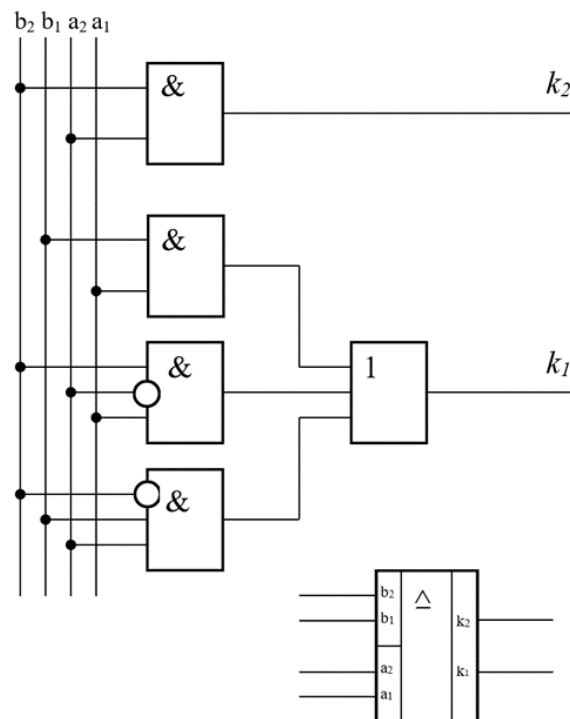


Рисунок 6. Функціональна схема
четвіркової кон'юнкції на два четвіркових входи

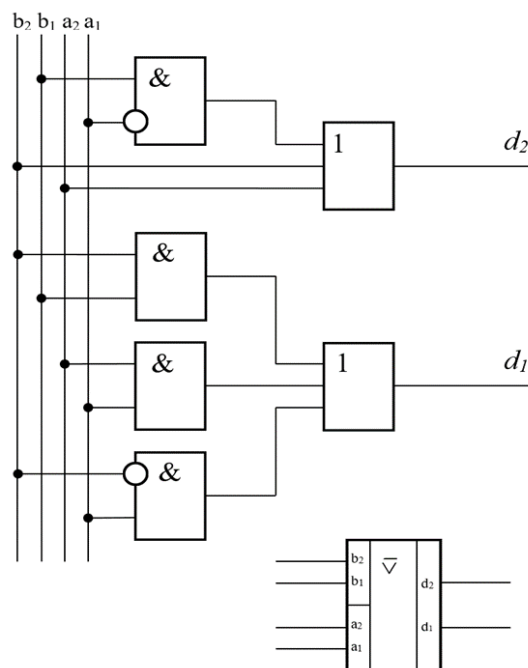


Рисунок 7. Функціональна схема четвіркової диз'юнкції
на два четвіркових входи

Аналогічно можна розробити схеми і для інших операцій k -значної логіки. Для побудови схем з більшим числом входів використаємо пірамідальну структуру (див. рис.8), робота якої не потребує пояснень.

Для перетворення схеми рис.8 на три входи видаляємо верхню схему першого рівня на два входи, а входи b_1 та b_2 другого рівня відповідно стають входами z_1 і z_2 першого рівня. Так само будується і диз'юнкція на більше входів.

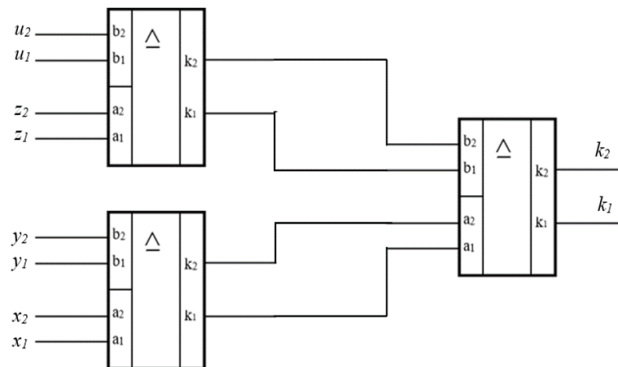


Рисунок 8. Функціональна схема кон'юнкції на чотири входи

На рис. 9 подано можливий варіант функціональної схеми додавання за модулем 4.

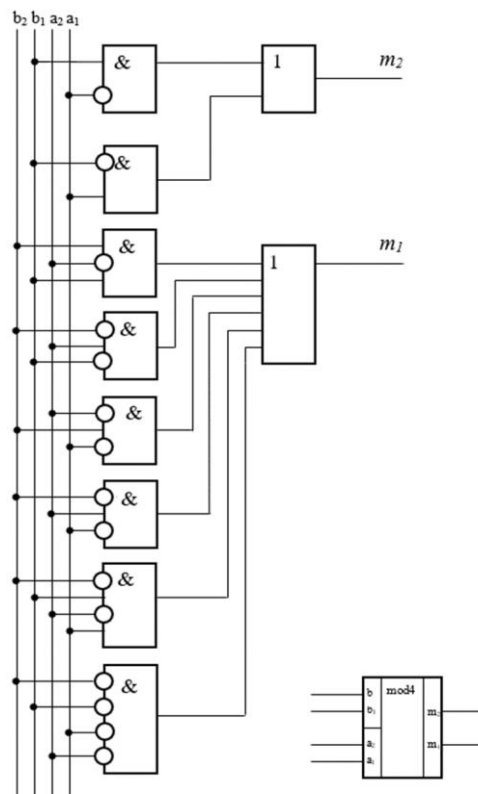


Рисунок 9. Функціональна схема додавання по mod 4

Аналогічно можна побудувати і схеми інших операцій. Окремі логічні вузли здебільшого проектуються на підставі логічних рівнянь і з'єднуються в систему на підставі алгоритму роботи пристрою в цілому.

Для прикладу, розглянемо можливий варіант рівняння для розробки схеми, що визначає «фронт» БЛЧФ (фронт – це характеристика сигналу, а у нашому випадку – логічні змінні, тому «фронт» в лапках).

Нехай маємо БЛЧФ $0(x_t^T)0$, що задана на неперервному інтервалі існування. Продукуюче слова такої функції може включати в довільному порядку всі значення від 1 до $k-1$. Нас цікавить лише логічне значення a_0 . Для цього потрібно з вхідної функції $0(x_t^T)0$ створити допоміжну константу $(k-1)$ з часовими координатами t, T . Досягти це, з використанням комбінації широковідомих k -значних функцій авторів не вдалося, через проблему нуля за межами існування функції.

Найпростіше тут ввести функцію формування константи (позначимо її символом «й»), яка на вхідних значеннях нуль формує також нуль, а на всіх вхідних значення від 1 до $(k-1)$ формує значення $(k-1)$. Остання операція дасть $0(Const(k-1)_t^T)0$. Кон'юнкція цієї константи і її інверсії Лукасевича, затриманої на один Δ -інтервал, дає нам шаблон $(k-1)_t^1$ для визначення «фронт» БЛЧФ шляхом кон'юнкції шаблону і вихідної функції. Остаточно маємо:

$$a_0 \mid x_t^1 = \left(\text{й}x \triangle D \text{й}x \right) \triangle x \mid x_t^1 \quad (10)$$

Операція формування константи $0(Const(k-1)_t^T)0$ для четвіркової функції на базі двійкових елементів і кодування, викладеного раніше, описується формулою $k_{i1} = k_{i2} = a_{i1} \vee a_{i2}$. Повна схемотехнічна реалізація формули (10) не

викликає якихось ускладнень. Затримка на один інтервал може бути реалізована на бінарних D-тригерах або іншим способом.

На рис. 7 показані діаграми роботи алгоритму.

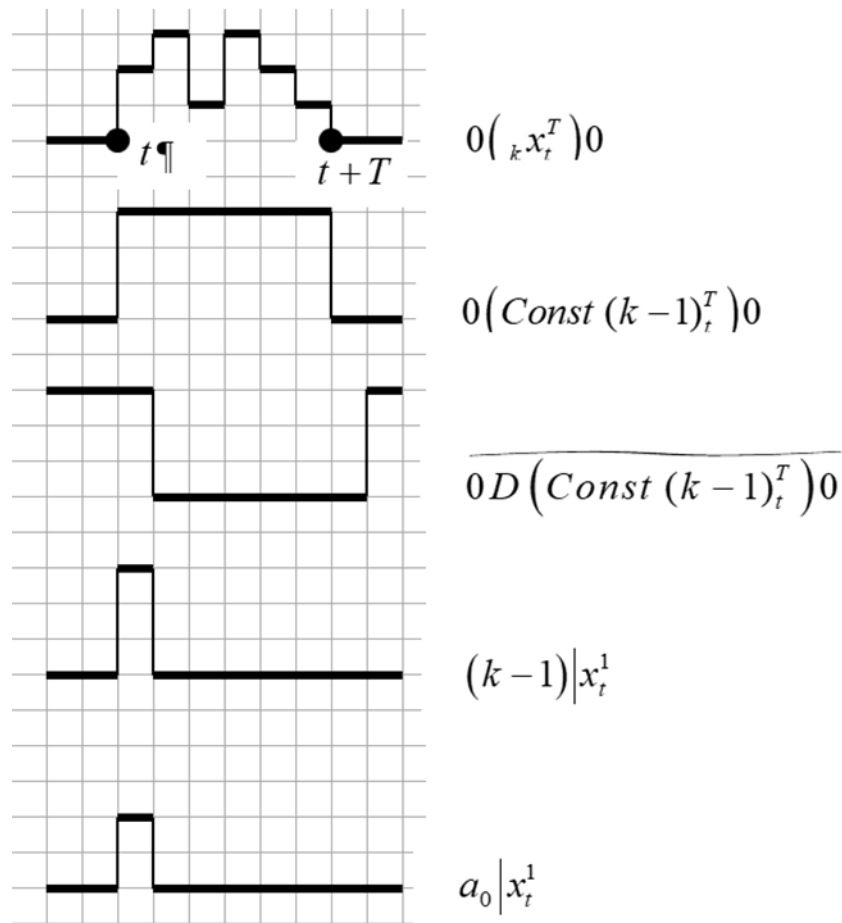


Рисунок 10. Виділення переднього «фронту» БЛЧФ

Рівняння (10) дозволяє виділити і задній «фронт» БЛЧФ.

$$a_{T-1}|x_{t+T}^1 = (D \text{ й} x \triangle \text{й} x) \triangle x |x_{t+T}^1 \quad (11)$$

Подібні алгоритми можуть бути використані при аналізі роботи електронних пристроїв.

Можливі інші рівняння і схеми типу: «сигнал x був», «сигналу x не було», «сигнал x має тривалість m » і т. п. Функція «й» фактично означає: «був якийсь сигнал».

На рис.12 представлена електрична схема інтегратора з функцією диференціювання, розробленого з використанням програмного пакету NI Multisim 14.0 на віртуальних компонентах, що реалізує формулу інтегралу отриману шляхом аналізу і узагальнення рис.11.

$$\int \left(\bigoplus_{k=0}^{a_i} x_0^{t_x} \bigoplus_{k=0}^{a_j} x_{t_x}^{\infty} \right) dt = \left(\bigoplus_{r=0}^{(t_x/2)-1} a_i^0 x_{(t_x-2)-2r}^2 \bigoplus_{r=0}^{\infty} a_j^0 x_{t_x+2r}^2 \right) \ominus K, \quad (12)$$

де $r \in 0, 1, 2, \dots, n$ та виконується одна або обидві умови $(a_j + a_i) \leq (k-1)$, $(a_j - a_i) \geq 0$, $a_i, a_j \in \{0, 1, \dots, (k-1)\}$

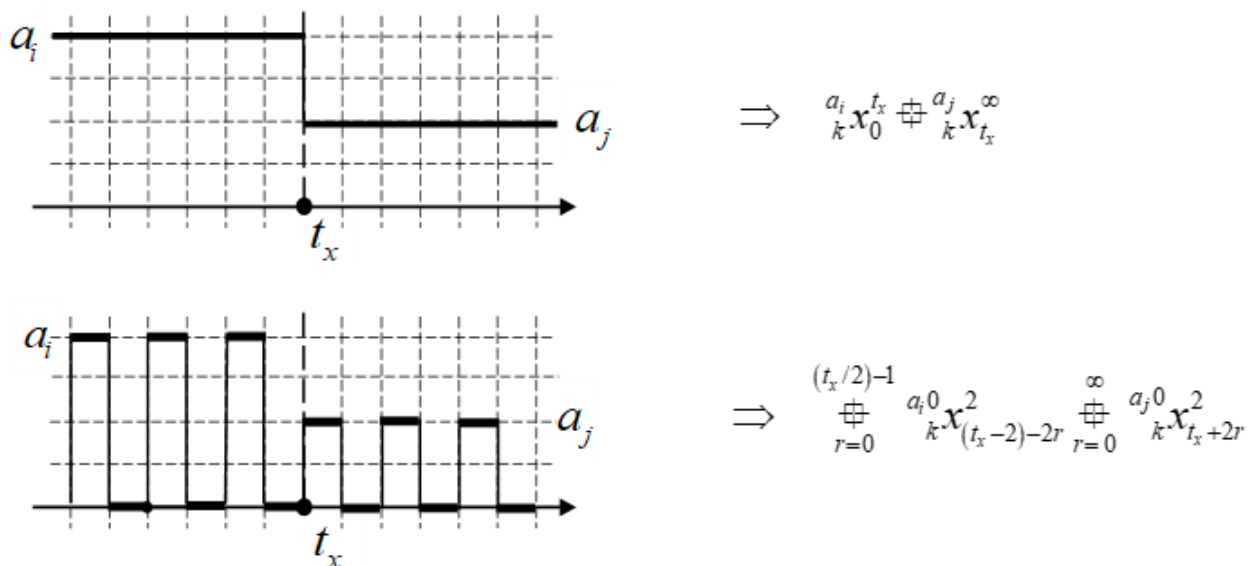


Рисунок 11. Первісна переходу (стрибка значення) констант

та формулу похідної з рахуванням особливостей бінарних ЛЧФ. В наведеній схемі (див. рис. 12) блок перемикачів $S1$ служить для установки параметрів інтервалу існування БЛЧФ. Регістр зсуву $U5$ $U6$ $U1$ формує власне ЛЧФ. Модуль нерівнозначності $U2$ та тригер $U7$, що сумують по $\text{mod}2$ поточне і зсунуте на одиницю значення вихідної функції ${}_2X_t^T$, формують похідну. Первісна формується модулем нерівнозначності $U9$ та тригером $U8$, сумуючи поточне значення функції і попереднє значення первісної.

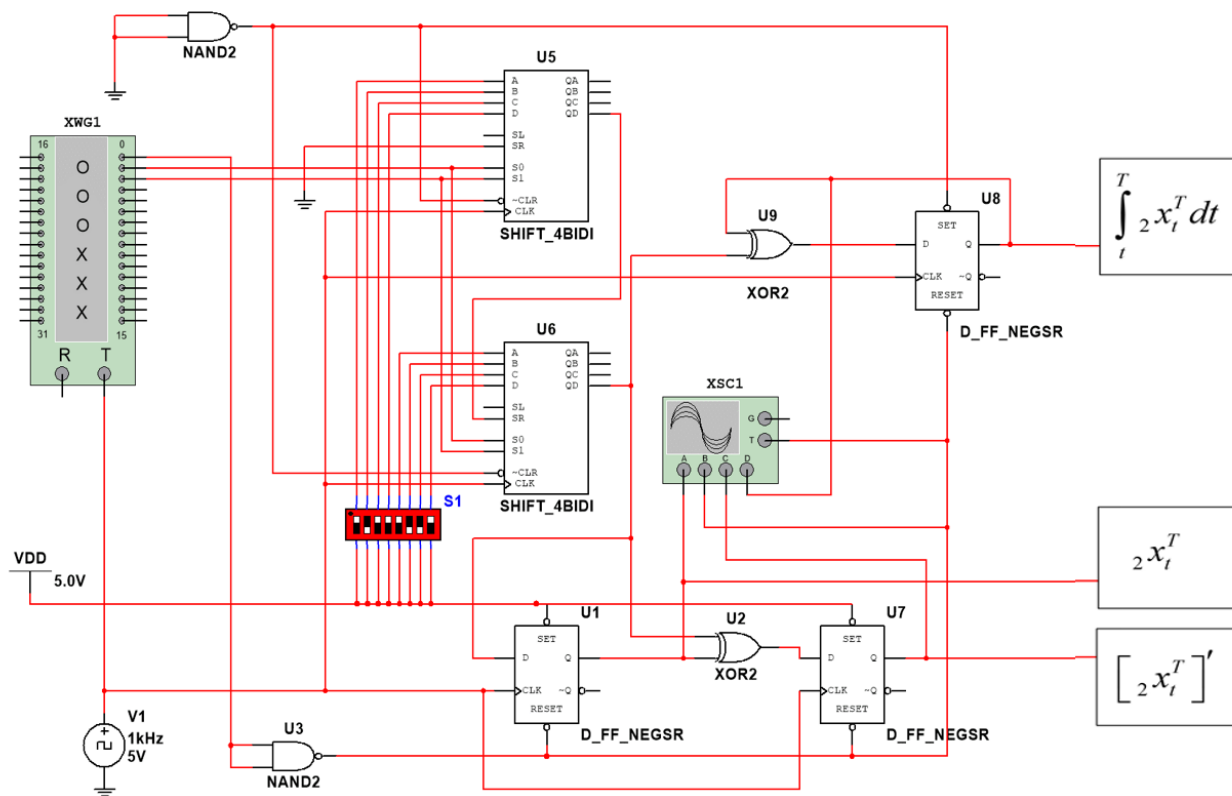


Рисунок 12. Двійковий інтегратор-диференціатор

Генератор слів XWG1 забезпечує формування управляючих сигналів для скидання схеми в вихідне положення, запису в регістр зсуву параметрів інтервалу існування функції ${}_2x_t^T = 10011101|{}_2x_t^T$, включення режиму зсуву. Призначення решти елементів і зав'язків зрозуміла зі схеми.

Моделювання схеми k – значної логіки, наприклад, трійкової можливе в базисі бінарних логічних елементів, на базі яких розробляються потрібні трійкові логічні елементи, тригери, регістри і т. п. Часова діаграма роботи бінарного інтегратора подана на рис. 13.

Враховуючи думку авторитетного американського вченого Дональда Кнута [142, 143] який сказав, що «заміна двійкового тригера на трійковий в один прекрасний день обов'язково станеться», все ж дещо зупинимось на трійковій схемотехніці.

Трійкові елементи та вузли можуть будуватись як з використанням різних кодових систем так і з використанням різних фізичних принципів роботи елементів.

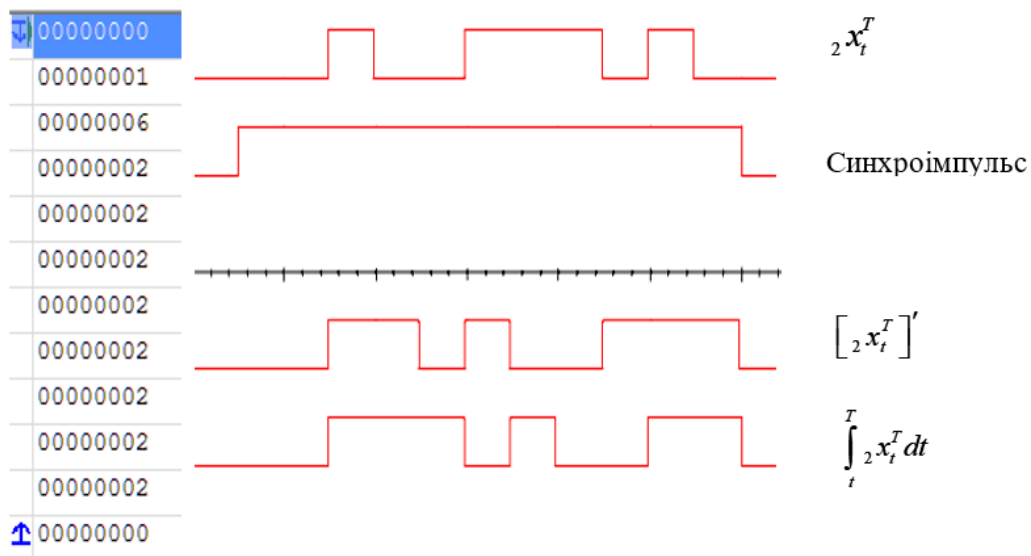


Рисунок 13. Часова діаграма роботи бінарного інтегратора

В даній роботі нас цікавлять трійкові вузли, що використовують звичайний двійковий код. Візьмемо за основу трійковий тригер «ТТДЛ-22» (ТТBL-22) на базі двійкових логічних елементів і коду 0, 1, 2 (00, 01, 10) англійською «zero», «one», «two» або скорочено – ZOT (можливо RST краще).

Установка стану виходів здійснюється за таблицею переходів (табл.2).

Таблиця 2. Таблиця переходів

Q_2	Q_1	Q_2^{*+1}	Q_1^{*+1}	S_2	S_1
Q_2	Q_1	0	1	0	1
Q_2	Q_1	1	0	1	0
Q_2	Q_1	0	0	1	1
Q_2	Q_1	Q_2	Q_1	0	0

Схема тригера з мінімальною вхідною логікою, осцилограми його роботи представлена на рис. 14 та рис. 15 та таблиця генератора слів (табл. 3).

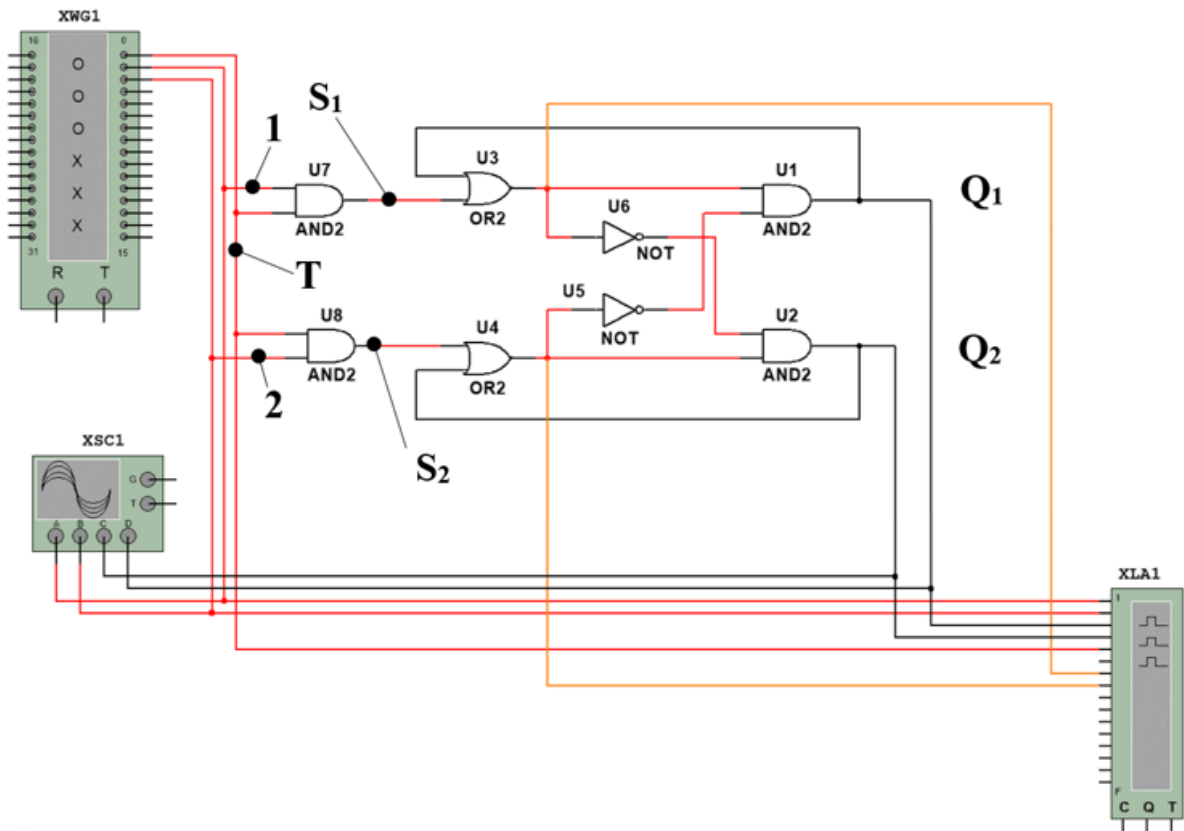


Рисунок 14. Трійковий тригер

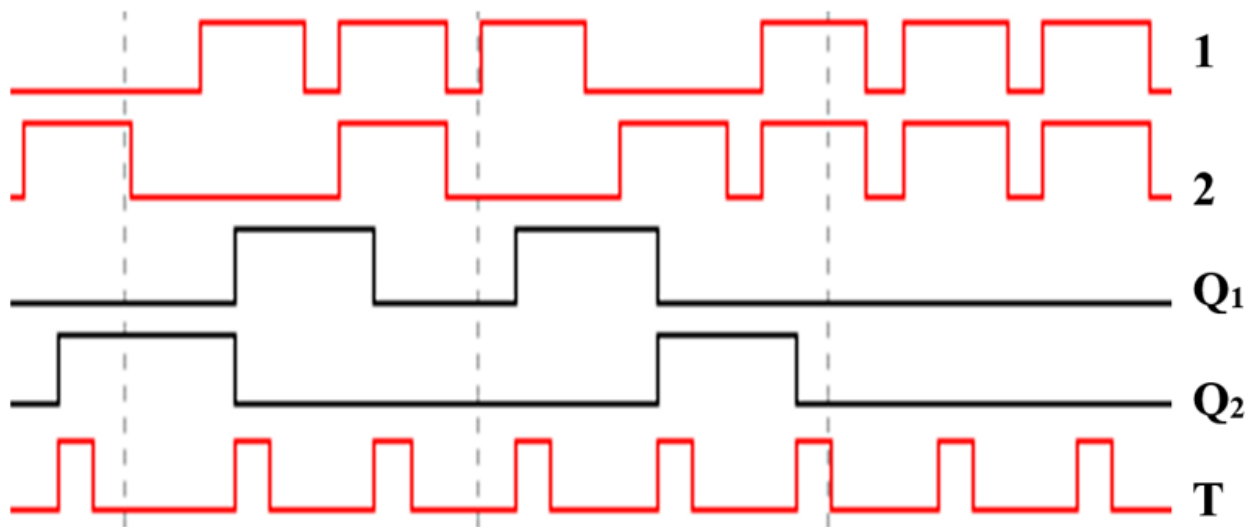


Рисунок 15. Діаграма роботи трійкового тригера

Таблиця 3. Діаграма роботи трійкового тригера

<u>Data:</u>	00000006	00000000
00000000	00000000	00000006
00000004	00000002	00000007
00000005	00000003	00000006
00000004	00000002	00000000
00000000	00000000	00000006
00000000	00000004	00000007
00000002	00000005	00000006
00000003	00000004	<u>Initial:</u>
00000002	00000000	0000
00000000	00000006	<u>Final:</u>
00000006	00000007	0020
00000007	00000006	

Це не єдиний трійковий тригер на двійкових елементах. В контексті багатозначних ЛЧФ внутрішня будова k -значного тригера не принципова. Але він має використовувати двійкові значення стану: 00, 01, 10. Згаданий трійковий тригер (3Т) вперше був використаний в 1968 в трійковій ЕВМ «Сетунь», що мала українське коріння. Зауважимо, що в трійковій системі не біти, а тріти і відповідно не тригер, а трітер. Тож і для нашого випадку будемо використовувати термін трітер.

Трійковий тригер, що працює по фронту імпульсу найпростіше побудувати, наприклад, для регістрів зсуву або лічильників на базі ТТДЛ-22 по схемі M-S (Master-Slave), синтезувавши по таблиці переходів лічильного тритера М потрібні входні сигнали S_1 та S_2 . Тоді маємо: $S_1^M = Q_1 \vee Q_2$, $S_2^M = \bar{Q}_2$. Інвертори U15, U16 забезпечують інверсну передачу стану М - тригера в S - тригер, через запис нуля двома одиницями. Діаграми і схеми зображено на

рис. 16 та рис. 17. Далі будемо використовувати тритери побудовані по схемі М-
S, робота яких більш зрозуміла.

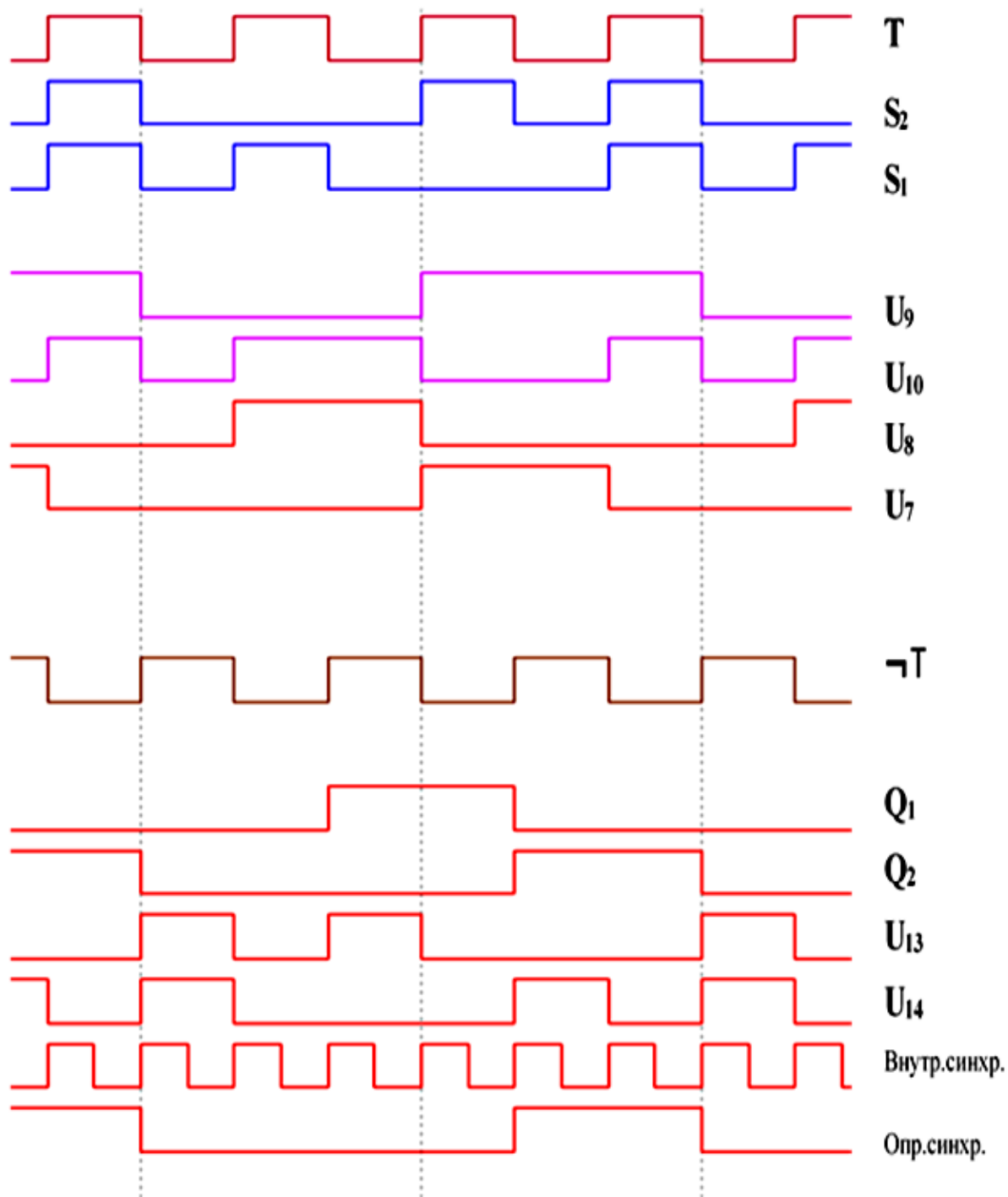


Рисунок 16. Діаграма роботи T – трітера

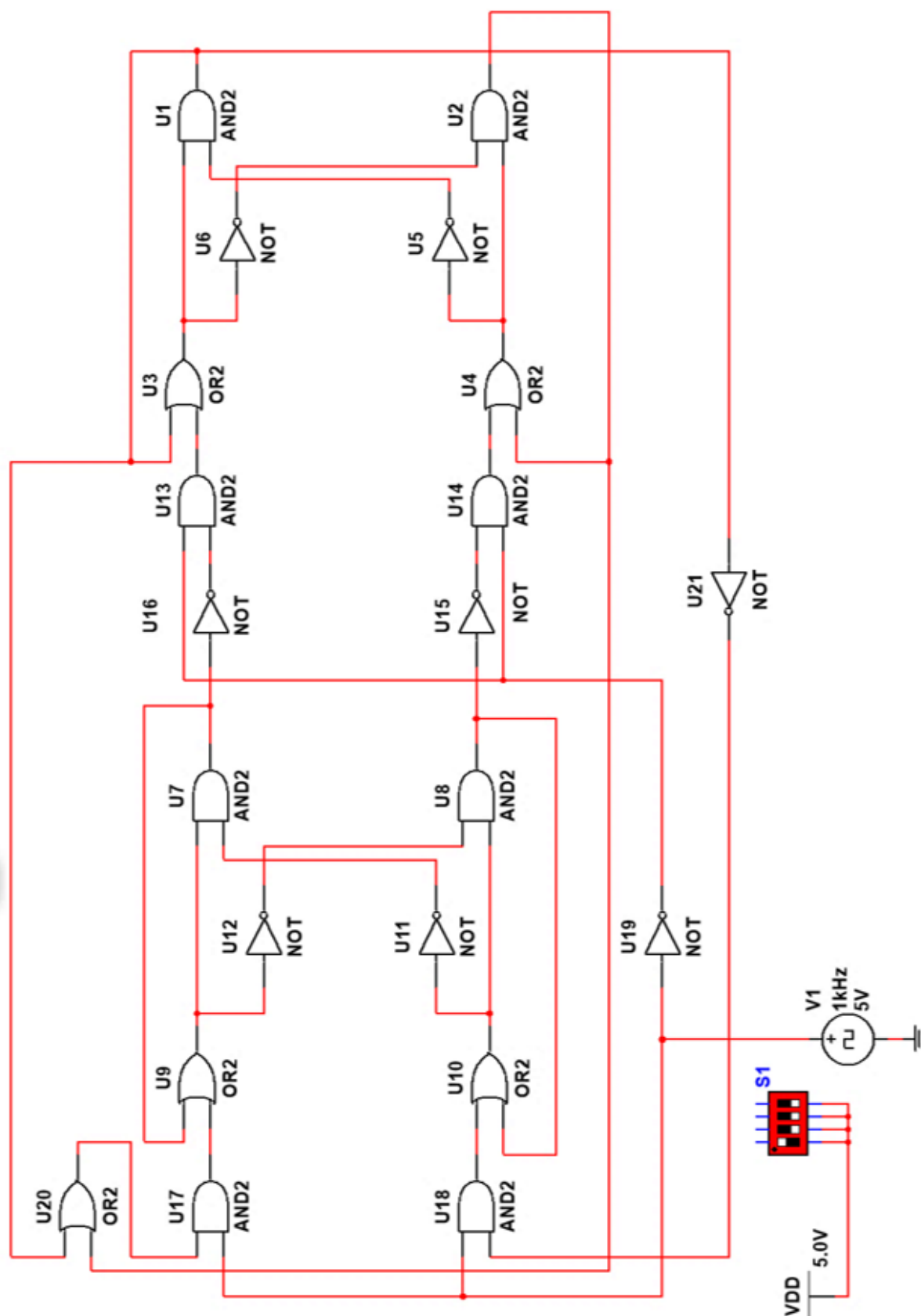


Рисунок 17. Схема T – тритера

Розглянемо синтез і моделювання тризначної схеми «НЕРІВНОЗНАЧНОСТІ МІНУС» та «НЕРІВНОЗНАЧНОСТІ ПЛЮС», що є основою операції інтегрування.

Таблиця 4 Таблиця істинності НЕРІВНОЗНАЧНОСТЬ МІНУС

x_2	x_1	y_2	y_1	n_2	n_1	$\bar{i}$
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	0	1	0	0	0
0	1	1	0	#	#	1
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0

При цьому:

$$n_1 = x_1 \bar{y}_1 \vee \bar{x}_1 y_1$$

$$n_2 = \bar{x}_1 \bar{x}_2 y_2 \vee x_2 \bar{y}_2 \bar{y}_1$$

$\bar{i} = x_1 y_2$ – ознака не інтегровності, що введена як окремий вихід схеми.

При синтезі схеми символ «#» було за замовчуванням замінено нулями (необхідність такого підходу продиктована особливістю схемної реалізації інтегрування на двійкових елементах). Фактично, нульовими значеннями первісної вхідна функція розбивається на підінтервали від одного значення не інтегровності до наступного, початковою умовою «0».

Таку функцію назвемо умовно або частково інтегровною, а первісну– доповненою або доповненою до початкової умови. В даному випадку доповненою до «0».

В загальному випадку така заміна можлива на будь-яке допустиме значення БЛЧФ, як на початкову умову інтегрування. І нарешті, за необхідності, можливий синтез схеми, де заміна # на необхідні початкові умови може задаватись.

Наприклад, четвіркова функція

$$f(t) = 10213021132001331032011220322232 \Big|_4 f_t^{32}$$

має подвійні значення на наборах 1-1 та 2-1, що теоретично може для цієї функції сформуванати не менше 32 доповнених первісних. Звичайно розглядати всі варіанти не доцільно, а розглянемо лише варіанти

$$\int_0^{31} f_t^{32} \Big|_{\bar{p}}^{\oplus+} dt \text{ та } \int_0^{31} f_t^{32} \Big|_{\bar{p}}^{\oplus-} dt,$$

тут

- $\oplus+$ та $\oplus-$ означають, що при подвійному значенні первісної за результат приймаємо значення суми (більшого) чи різниці (меншого) відповідно;
- $\bar{p}$ - тут ознака доповненої первісної.

На рис.18 показані вихідна функція $f(t)$, первісна $\int_0^{31} f_t^{32} dt$ та первісні

$$\int_0^{31} f_t^{32} \Big|_{\bar{p}}^{\oplus+} dt \text{ і } \int_0^{31} f_t^{32} \Big|_{\bar{p}}^{\oplus-} dt. \text{ При цьому великі цифри – подвійні значення; заливка –}$$

на цьому інтервалі функція не інтегровна і значення первісною прийнято за нуль.

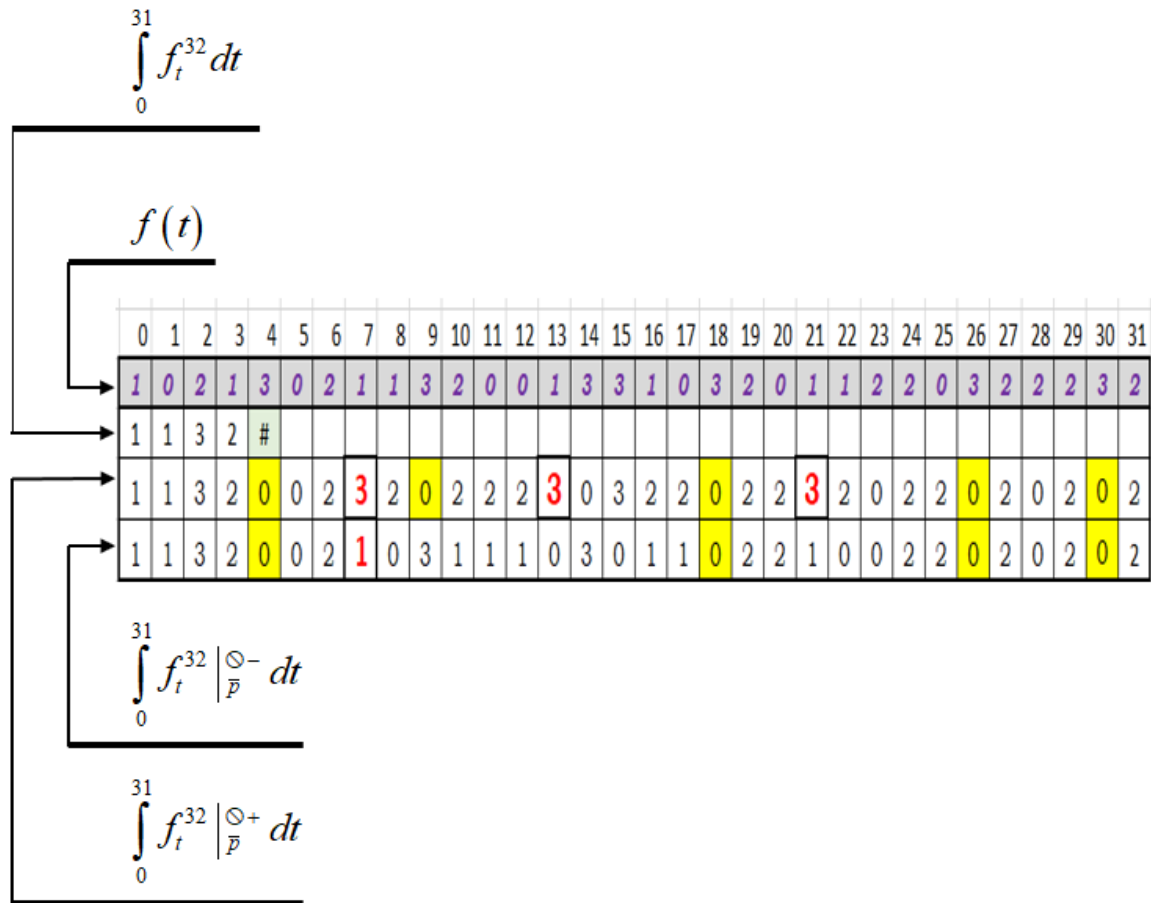


Рисунок 18. Доповнені первісні з початковою умовою «0»

Розглянемо схему $\ominus$ – (див. рис.19) та діаграму її роботи (див. рис.20).

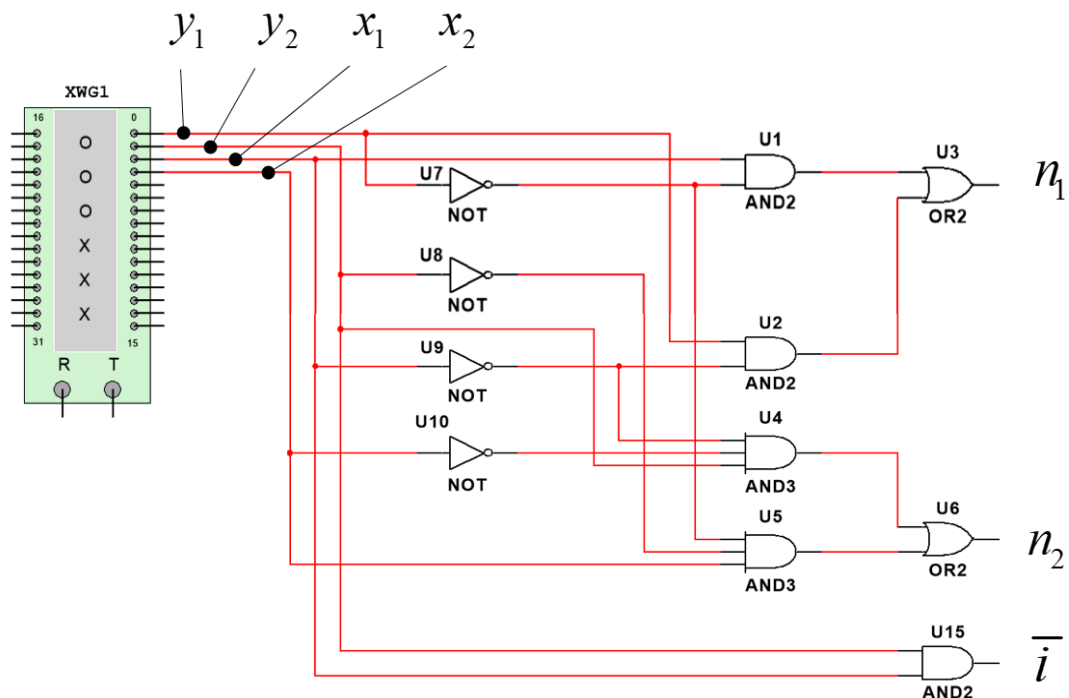


Рисунок 19. Трійкова двофазна схема НЕРІВНОЗНАЧНОСТІ МІНУС

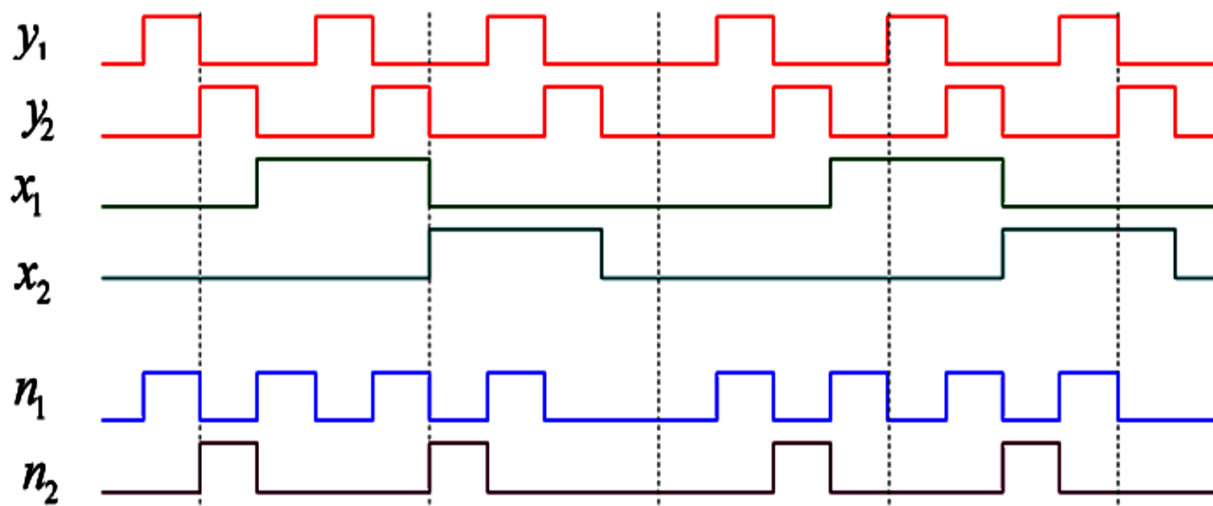


Рисунок 20. Діаграма роботи двофазної схеми НЕРІВНОЗНАЧНОСТІ МІНУС

Для більш ефективного реального використання цієї схеми варто додати вихідні вентиля для зчитування інформації.

Таблиця 5. Таблиця істинності та логічні рівняння схеми НЕРІВНОЗНАЧНОСТІ+

x_2	x_1	y_2	y_1	n_2	n_1	$\bar{i}$
0	0	0	0	0	0	0
0	0	0	1	0	1	0
0	0	1	0	1	0	0
0	1	0	0	0	1	0
0	1	0	1	1	0	0
0	1	1	0	#	#	1
1	0	0	0	1	0	0
1	0	0	1	0	1	0
1	0	1	0	0	0	0

$$n_1^+ = x_1 \bar{y}_1 \bar{y}_2 \vee \bar{x}_1 y_1$$

$$n_2^+ = \bar{x}_2 y_1 \vee \bar{x}_1 \bar{x}_2 y_2 \vee x_2 \bar{y}_2 \bar{y}_1$$

$$\bar{i} = x_1 y_2 - \text{ознака не інтегровної функції}$$

Звернемо увагу, що результат операції $\ominus^+$, який позначений решіткою, як і результат $\ominus^-$, робить ці операції не комутативними, а для синтезу схем функцій n_1, n_2 цей набір не використовуємо, вважаючи його значення нулем.

Іншими словами пари входів x_1, x_2 та входи y_1, y_2 не рівнозначні.

В рівняннях верхній індекс «-» та «+» вказують на приналежність рівняння до відповідної НЕРІВНОЗНАЧНОСТІ.

Розглянемо схеми нерівнозначності для чотиризначних ЛЧФ, в яких будемо використовувати двійкове кодування і звичайні двійкові логічні елементи.

Для цього спочатку побудуємо таблиці Келі для операцій додавання та віднімання.

Таблиця 6 Сума

+	0	1	2	3
0	0	1	2	3
1	1	2	3	4
2	2	3	4	5
3	3	4	5	6

Таблиця 7 Різниця

-	0	1	2	3
0	0	-1	-2	-3
1	1	0	-1	-2
2	2	1	0	-1
3	3	2	1	0

Далі будуємо об'єднану таблицю.

Таблиця 8 Об'єднана

$\ominus$	0	1	2	3
0	0	1	2	3
1	1	0 ∇ 2	3	-2:4
2	2	1 ∇ 3	0	-1:5
3	3	2	1	0

Аналіз табл.8 вказує недопустимі значення функції: -2:4 та -1:5 при входніх парах значень 1:3 та 2:3. Це означає, що при таких входніх значеннях чотиризначна ЛЧФ є не інтегровною (скоріше частково або умовно інтегровною).

Крім того, при входніх наборах 1-1 та 2-1 маємо два значення функції, що породжують чотири правила інтегрування $\odot_{01}$, $\odot_{03}$, $\odot_{21}$, $\odot_{23}$ (див. табл. 9-12).

Всі ці операції не комутативні і при використанні схем для їх реалізації входи не взаємозамінні. В інтеграторі на входи x_1, x_2 будемо подавати p_{i-1} .

Таблиця 9. $\odot_{01}$

$\odot_{01}$	0	1	2	3
0	0	1	2	3
1	1	0	3	#
2	2	1	0	#
3	3	2	1	0

Таблиця 10. $\odot_{03}$

$\odot_{03}$	0	1	2	3
0	0	1	2	3
1	1	0	3	#
2	2	3	0	#
3	3	2	1	0

Таблиця 11. $\odot_{21}$

$\odot_{21}$	0	1	2	3
0	0	1	2	3
1	1	2	3	#
2	2	1	0	#
3	3	2	1	0

Таблиця 12. $\odot_{23}$

$\odot_{23}$	0	1	2	3
0	0	1	2	3
1	1	2	3	#
2	2	3	0	#
3	3	2	1	0

На основі цих таблиць будемо таблицю істинності (див. табл.13) і через карти Карно записуємо логічні рівняння виходів схем.

$$n_2^{01} = P_2 P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_2 \vee P_2 \bar{A}_2 \bar{A}_1 \vee \bar{P}_2 \bar{A}_1 A_2$$

$$n_1^{01} = P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_1 \vee \bar{P}_1 \bar{A}_2 A_1$$

$$n_2^{23} = P_2 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_2 \vee \bar{P}_2 A_2 \bar{A}_1 \vee P_1 \bar{A}_2 A_1$$

$$n_1^{23} = P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_1 \vee \bar{P}_1 \bar{A}_2 A_1$$

$$n_2^{03} = P_2 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_2 \vee \bar{P}_2 A_2 \bar{A}_1 \vee \bar{P}_1 A_2 \bar{A}_1$$

$$n_1^{03} = P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_1 \vee \bar{P}_1 \bar{A}_2 A_1$$

$$n_2^{21} = P_2 P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_2 \vee \bar{P}_2 \bar{A}_2 \bar{A}_1 \vee P_1 \bar{A}_2 A_1$$

$$n_1^{21} = P_1 \bar{A}_2 \vee \bar{P}_2 \bar{P}_1 A_1 \vee \bar{P}_1 \bar{A}_2 A_1$$

$$\bar{i} = P_2 \bar{P}_1 A_2 A_1 \vee \bar{P}_2 P_1 A_2 A_1$$

Заслужовує уваги те, що перший розряд функції n_i однаковий для всіх схем. Побудова такої комбінаційної схеми для сучасних технологій є простою задачею. Більше того, при інтегральному виконанні схеми не потрібна і мінімізація логічних рівнянь.

Таблиця 13. Таблиця істинності

	P ₂	P ₁	A ₂	A ₁	n ₂ ⁰¹	n ₁ ⁰¹	n ₂ ²³	n ₁ ²³	n ₂ ⁰³	n ₁ ⁰³	n ₂ ²¹	n ₁ ²¹
0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	1	0	1	0	1	0	1
2	0	0	1	0	1	0	1	0	1	0	1	0
3	0	0	1	1	1	1	1	1	1	1	1	1
4	0	1	0	0	0	1	0	1	0	1	0	1
5	0	1	0	1	0	0	1	0	0	0	1	0
6	0	1	1	0	1	1	1	1	1	1	1	1
7	0	1	1	1	#	#	#	#	#	#	#	#
8	1	0	0	0	1	0	1	0	1	0	1	0
9	1	0	0	1	0	1	1	1	1	1	0	1
10	1	0	1	0	0	0	0	0	0	0	0	0
11	1	0	1	1	#	#	#	#	#	#	#	#
12	1	1	0	0	1	1	1	1	1	1	1	1
13	1	1	0	1	1	0	1	0	1	0	1	0
14	1	1	1	0	0	1	0	1	0	1	0	1
15	1	1	1	1	0	0	0	0	0	0	0	0

Отримані схеми вимагають зауваження в частині не інтегровності вхідних функцій, наприклад для четвіркових наборів 7 та 11 (6 для трійкового набору). Раніше відмічалось, що при синтезі схем значення функції « $\emptyset$ » на цих наборах позначене символом «#» і дорівнює нулю. Це означає, що на виході схеми інтегрування в ці моменти також буде нуль. Не залежно від фізичних принципів роботи логічних елементів і k -значності на виході схеми в момент не інтегровності завжди буде одне із k значень БЛЧФ.

В нашому випадку для бінарної логіки одним із $k \in \{0, 1\}$. Саме схемотехнічна наявність конкретного значення сигналу із множини допустимих при «#» вимагає не інтегровності визначати окремим сигналом, наприклад $\bar{1}$. У випадку надлишково двійково-кодованих k -значних функцій можуть бути недопустимі значення функції, наприклад значення три для трійкової функції або 5, 6 та 7 для п'ятіркової і т. п. Тоді такі значення можна використати як $\bar{1}$. В нашому випадку нуль допустиме значення первісної і тоді $\bar{1} = 1$.

Якщо значення $\bar{1}$ ігнорувати, то отримаємо первісну умовно інтегрованої функції. Таким чином, нульовими значеннями первісної при $\bar{1} = 1$ вхідна функція розбивається на підінтервали з початковою умовою «0». При інших значеннях первісної на не інтегровному наборі отримаємо і інші початкові умови на вхідних інтервалах. Будемо вважати, що первісна на не інтегровних наборах завжди дорівнює нулю, якщо окремо не вказано інше значення.

Розглянемо трійковий інтегратор (див. рис. 21) з каналом диференціювання. Основу схеми складає чотирирозрядний трійковий регістр зсуву, що побудований на ZOT-трітерах HB1-HB4 і доповнений трітером HB5. Передача інформації між розрядами відбувається по спаду вхідного сигналу С «прямими» входами S1 та S2. Цим же сигналом початкове вхідне слово з перимикача S3 записується по додаткових входах S1A і S2A.

На «прямих» входах нуль записується комбінацією 00, а на додаткових – комбінацією 11.

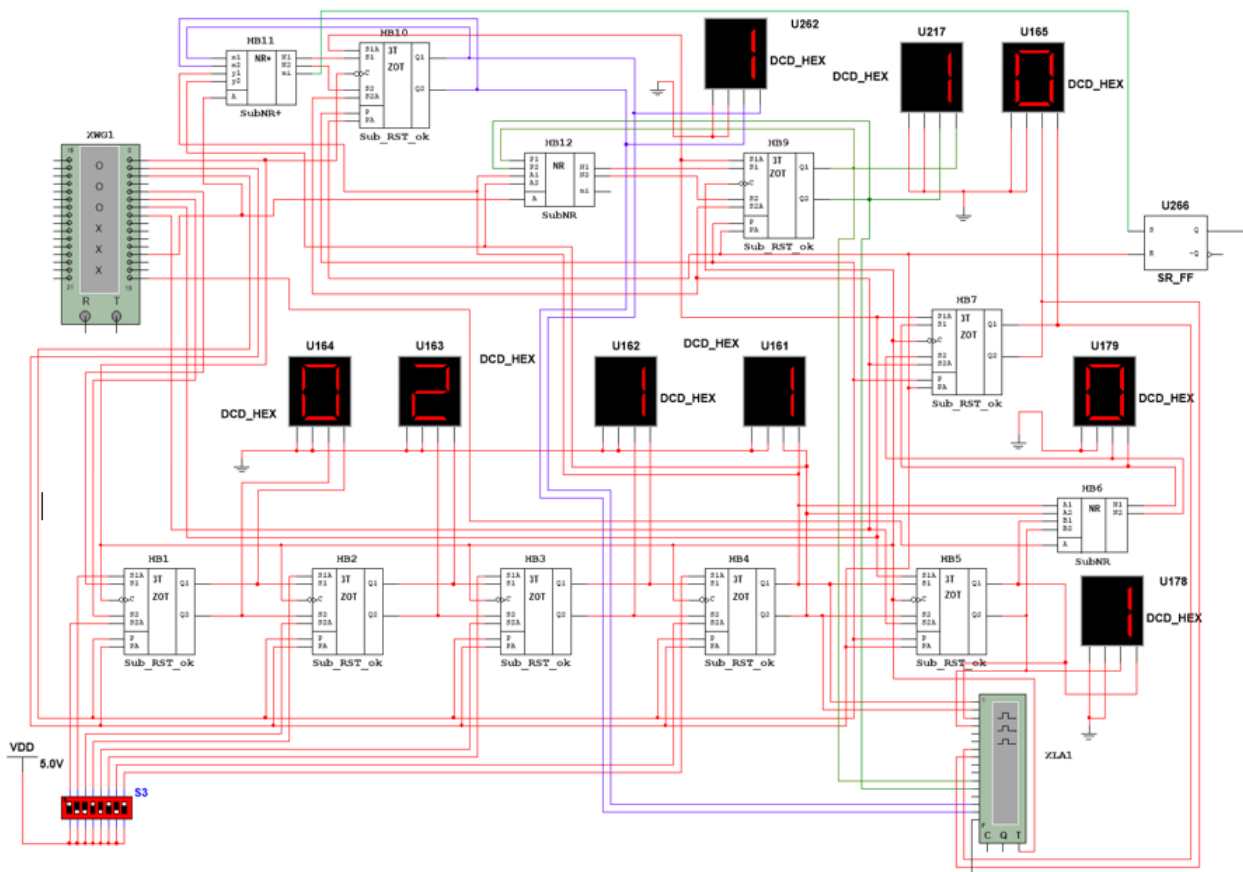


Рисунок 21. Схема трійкового інтегратора

Вхідне слово задається лінійкою перемикачів S4. Відповідно зі схемою трітера для запису нуля на ці входи потрібно подати одиниці. Запис вихідного слова і його зсув в регістрі керуються потенціалами на входах PA та P.

Запис нулів при зсуві в старший розряд забезпечується генератором слів XWG1 (див. табл. 14). Похідна вхідного слова формується схемою нерівнозначності HB6, яка стробується сигналом A і потім записується в тритер HB7. З його виходу похідна в послідовному коді може передаватись далі для використання.

Первісна через НЕРІВНОЗНАЧНІСТЬ— формується схемою нерівнозначності HB12 і трітером HB9, що охоплені зворотним зв'язком $p_i = p_{i-1} \ominus - a_i$.

Первісна через НЕРІВНОЗНАЧНІСТЬ+ формується елементами HB10, HB11 за формулою $p_i = p_{i-1} \ominus + a_i$.

Ознака того, що БЛЧФ не інтегрована формується елементом HB11 ($\Theta+$) та тригером U266 (SR-FF) за формулою $(p_{i-1}\Theta + a_i) = \#$.

Шістнадцятирічні індикатори зручно використовувати при покроковому аналізі роботи схеми. Вся робота схеми керується генератором слів.

Таблиця 14. Генератор слів XWG1

<u>Data:</u>	00009004
00000000	00008004
000000CA	00008005
000000CB	00008004
000000CA	00000004
00009004	00000005
00009005	00000004
00009004	00000004
00009004	00000005
00009005	00000004
00009004	00000004
00009004	00000005
00009005	00000004
00009004	<u>Initial:</u>
00009004	0000
00009005	<u>Final:</u>
	001B

На часовій діаграмі (див. рис. 22) показані три послідовності вхідних слів: 1021, 1102 та 2112.

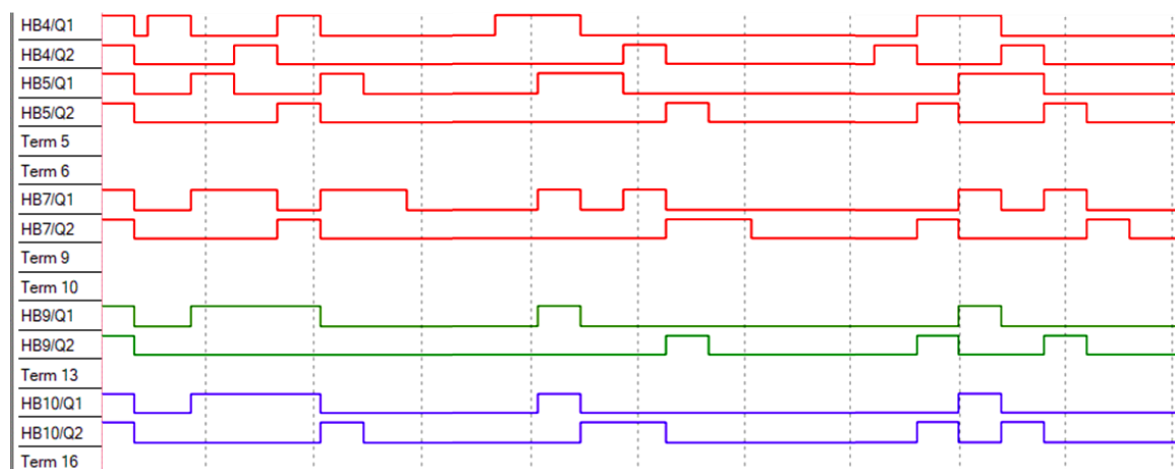


Рисунок 22. Часові діаграми трійкового інтегратора

Вихідні сигнали різних елементів схеми записуються логічним аналізатором XLA1 та представлені нижче. Їх зміст зрозумілий по позначенню виходів на схемі.

Перемикач S3 задає вхідне слово з молодшим розрядом справа. Зсув в регістрі організований також вправо. Відповідно на часовій діаграмі перший розряд зліва.

В подальшому, результати інтегрування та диференціювання БЛЧФ можна записати в регістри зсуву для отримання паралельного коду.

На рис. 23 наведено часові діаграми для БЛЧФ 2121 та 1112, що не є інтегровними. Вихід тригера SR-FF формує ознаку не інтегровності або ознаку недостовірності (достовірності) результату інтегрування за формулою $(p_{i-1} \ominus + a_i) = \#$. Одиничне значення тригера SR-FF на виході Q говорить про те, що функція не інтегровна і отримане значення первісної не достовірне і має ігноруватись.

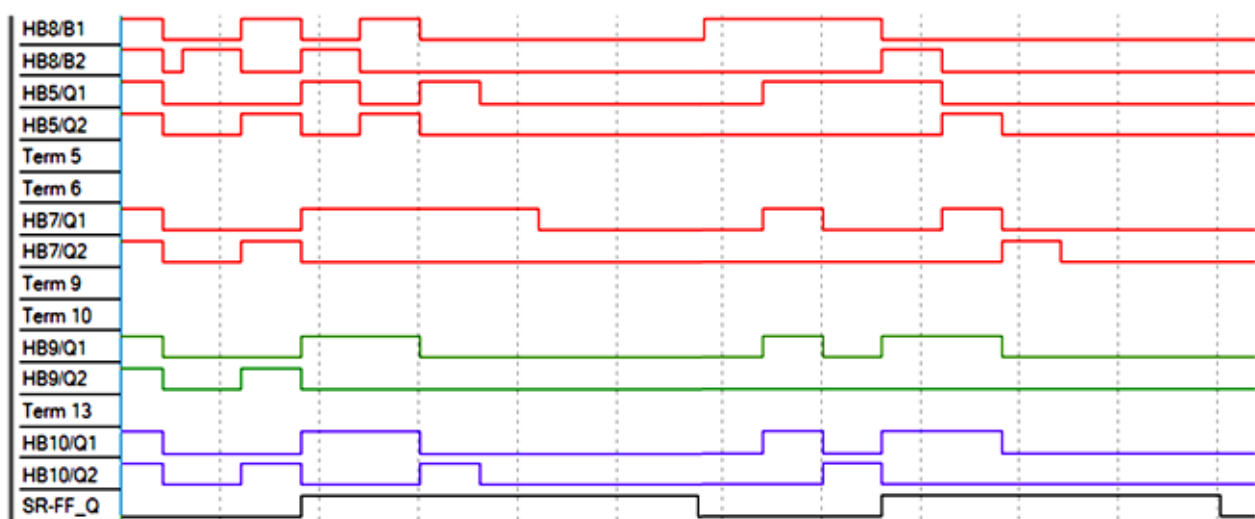


Рисунок 23. Діаграми трійкового інтегратора для не інтегровних функцій

Моделювання роботи інтегратора-диференціатора проводилось в пакеті NI Multisim 14.0 на віртуальних компонентах під управлінням операційної системи Windows 10 Pro на процесорі Intel Core i3-2120 з тактовою частотою 3,3 GHz та об'ємом оперативної пам'яті 8 ГБ.

Схема, що моделювалась, включає в себе 1414 компонентів із них 1325 елементи типу I, АБО, НІ та має 4170 виводів, що з'єднані між собою відповідно до схеми.

Тривалість моделювання одного циклу інтегратора-диференціатора становить 2' 30" ... 3' 30".

Для трійкового слова $a = 20222112$ маємо дві ЛЧФ:

ліві тріти вперед ${}_3(aL)_t^8 = 20222112|_3(aL)_t^8 = 202^3 1^2 2|_3(aL)_t^8$;

праві тріти вперед ${}_3(aR)_t^8 = 21122202|_3(aR)_t^8 = 21^2 2^3 02|_3(aR)_t^8$

В результаті диференціювання отримаємо (справа результат моделювання):

$${}_3(aL)_t^8 = 20222112|_3(aL)_t^8 \rightarrow$$


$$\left[{}_3(aL)_t^8\right]' = 22001012|_3(aL)_t^8 \rightarrow$$


$$\int_t^8 \left[{}_3(aL)_t^8\right]|\ominus^- dt = 22020102|_3(aL)_t^8 \rightarrow$$


$$\int_t^8 \left[{}_3(aL)_t^8\right]|\ominus^+ dt = 22020120|_3(aL)_t^8 \rightarrow$$


$$\left[{}_3(aL)_t^8\right]^{(4)} = 01100112|_3(aL)_t^8 \rightarrow$$


Якщо початкова умова $p_{-1} = 2$ для $20222112|_3(aL)_t^8$ отримаємо:

$$\int_t^8 \left[{}_3(aL)_t^8\right]|\ominus^- dt = 00202102|_3(aL)_t^8$$


А при $p_{-1} = 1$ отримаємо не інтегровну функцію.

$${}_3(aR)_t^8 = 21122202|_3(aR)_t^8 \rightarrow$$


$$\left[{}_3(aR)_t^8\right]' = 10100222|_3(aR)_t^8 \rightarrow$$


$$\int_t^8 \left[{}_3(aR)_t^8\right] dt = 21020220|_3(aR)_t^8 \rightarrow$$


$$\int_t^8 \left[{}_3(aR)_t^8 \right] |^{\odot+} dt = 21202002 | {}_3(aR)_t^8 \rightarrow$$

21202002

Нижче показані результати обчислення похідної, а первісних за формулами $\ominus-$ та $\ominus+$ при початкових умовах $p_{-1} = 0$, $p_{-1} = 1$, $p_{-1} = 2$

$${}_3x_t^8 = 11110101 | {}_3x_t^8$$

$$\left[{}_3x_t^8 \right]' = 00011111 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_0^{\odot-} dt = 10100110 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_0^{\odot+} dt = 12122112 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_1^{\odot-} dt = 01011001 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_1^{\odot+} dt = 21211221 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_2^{\odot-} dt = 10100110 | {}_3x_t^8$$

$$\int_t^8 \left[{}_3(x)_t^8 \right] |_2^{\odot+} dt = 12122112 | {}_3x_t^8$$

Результати моделювання співпадають з результатами отриманими теоретично. Крім того, схема працює і з двійковими вхідними словами, беручи до уваги той факт, що $\int_t^T {}_2x_t^T dt$ не існує.

Дослідницьку схему, що працює в трійковій логіці, не складно перетворити в схему чотиризначної логіки.

Для цього достатньо замінити трітер на два двійкових D-тригери з відповідною вхідною логікою рис. 24.

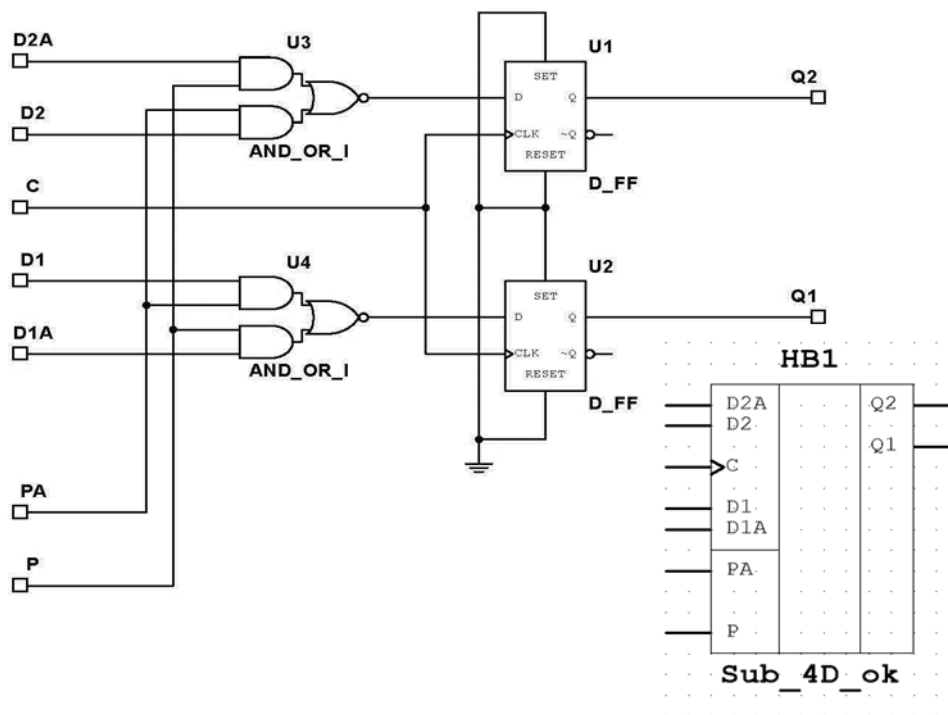


Рисунок 24. Четвірковий тригер

Далі, на базі такого четвіркового тригера створити підсхеми відповідних регістрів із подібними входами і ідентичними функціями та побудувати схеми відповідні схемі нерівнозначності. Потім провести відповідну заміну потрібних підсхем з урахуванням особливостей їх роботи та уточнити генератор слів.