МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

КОМП'ЮТЕРНА ЛОГІКА. КУРСОВА РОБОТА

Рекомендації до виконання курсової роботи

Видання друге перероблене і доповнене

Навчальний посібник

Рекомендовано Методичною радою КПІ ім. Ігоря Сікорського
як навчальний посібник для здобувачів ступеня бакалавра
за освітньою програмою «Комп’ютерні системи та мережі»
спеціальності 123 Комп’ютерна інженерія

Укладачі: В. І. Жабін, О. А. Верба

Електронне мережне навчальне видання

Київ
КПІ ім. Ігоря Сікорського
2025

УДК 004.31 (075.8)
К

Укладачі: *Жабін Валерій Іванович, д-р техн. наук, проф.*
Верба Олександр Андрійович, канд. техн. наук

Рецензент: *Дичка І. А., д-р техн. наук, проф.*, декан факультету прикладної математики, науковий керівник кафедри програмного забезпечення комп'ютерних систем КПІ ім. Ігоря Сікорського

Відповідальний редактор *Кулаков Ю.О., д-р. техн. наук, проф.*

Гриф надано Методичною радою КПІ ім. Ігоря Сікорського
*(протокол **№ 5 від 6.03.2025 р**.)*
за поданням Вченої ради факультету інформатики та обчислювальної техніки
*(протокол **№ 8. від 24.02.2025 р**.)*

Комп'ютерна логіка. Курсова робота. Видання друге перероблене і доповнене. [Електронний ресурс]: навч. посібн. для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія» / Укладачі: В. І. Жабін, О. А. Верба; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1,31 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2025. – 81 с.

Навчальний посібник призначено для вивчення, закріплення, поглиблення, узагальнення теоретичних знань та практичних навичок, які отримують студенти під час вивчення дисципліни «Комп'ютерна логіка», арифметичних основ побудови вузлів і блоків цифрових систем, прикладних питань теорії цифрових автоматів, а також придбання вмінь та навиків у застосуванні логічних методів аналізу і синтезу цифрових схем, навчитись користуватися довідковою літературою і вивчити процес створення проектно-конструкторської документації у відповідності діючим стандартам. Призначено для здобувачів ступеня бакалавра за освітньою програмою «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія».

УДК 004.31 (075.8)

Реєстр. № НП XX/XX-XXX. Обсяг 3,85 авт. арк.
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
проспект Перемоги, 37, м. Київ, 03056 https://kpi.ua
Свідоцтво про внесення до Державного реєстру видавців, виготовлювачів
і розповсюджувачів видавничої продукції ДК № 5354 від 25.05.2017 р.

© КПІ ім. Ігоря Сікорського, 2025

Зміст

Вступ…….……..……...…….3

1. Завдання на курсову роботу ….……..……..……..…3

2..Дослідження способів виконання операцій. …….…….4

3. Побудова функціональних схем…….…….…….…….17

4. Методичні вказівки до виконання курсової роботи……...….19

5. Критерії оцінювання курсової роботи…….22

Література……...……23

Додаток А. Титульний аркуш…….……..24

Додаток Б. Оформлення креслення, таблиць та рисунків....……25

Додаток В. Приклад виконання курсової роботи……..….27

ВСТУП

Курсова робота (КР) з дисципліни „Комп'ютерна логіка” є окремим навчальним модулем. КР призначена для розширення, закріплення, узагальнення і практичного застосування знань, умінь і навичок, отриманих студентом при вивченні дисципліни. У процесі виконання роботи студент повинен також навчитися користуватися довідковою літературою та діючими стандартами подання технічної документації. КР виконується за індивідуальним завданням і є самостійною роботою студента.

Варіант індивідуального завдання визначається вихідними значеннями операндів X і Y, які визначаються номером групи і номером студента у групі наступним чином.

Перевести номер варіанту в двійкову систему. Наприклад, номер групи 31, номер студента 15. Варіант $3115_{10} = 110000101011_2$

Записати два 10-розрядних двійкових числа:

$$X = -x_7x_6 1 x_5x_4 0, x_3 1 x_2x_1 \quad \text{і} \quad Y = +x_9 1 x_8x_7x_6x_5, x_4x_3x_2x_1,$$

де x_i - двійкові цифри номера варіанту у двійковій системі числення (x_1 - молодший розряд).

Визначені числа повністю визначають варіант КР.

1. Завдання на курсовую роботу

1.1 Числа X і Y в прямому коді записати у формі з плаваючою комою у класичному варіанті (з незміщеним порядком і повною мантисою). На мантису відвести 6 основних розрядів, на порядок студент самостійно вибирає необхідну кількість розрядів, щоб не було переповнення порядку при виконання чергової операції.

В пам’яті зберігається один знаковий розряд для мантиси та один знаковий розряд для порядку числа.

Записати числа X і Y також за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі).

1.2 Виконати 8 операцій з числами, що подані з плаваючою комою в класичному варіанті (чотири способи множення, два способи ділення, додавання та обчислення кореня додатного числа). Номери операцій (для п.3) відповідають порядку переліку, починаючи з нуля (наприклад, 0 – множення першим способом; 5 – ділення другим способом). Операндами для першого способу множення є задані числа X та Y. Для кожної наступної операції першим операндом є результат попередньої операції, а другим операндом завжди є число Y. (Наприклад, для ділення першим способом першим операндом є результат множення за четвертим способом, для операції обчислення кореня операндом є результат додавання зі знаком плюс).

Після виконання чергової операції результат може збільшитись за модулем. Студент може самостійно збільшити розрядність порядку, якщо це необхідно.

2. Дослідження способів виконання операцій

Перед виконанням чергової операції треба представити необхідні операнди в класичному вигляді з порядком та мантисою. Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу виконання операції;

2.2 операційну схему;

2.3 змістовний мікроалгоритм виконання операції;

2.4 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 6 основних розрядів мантиси результату;

2.5 обробку порядків (показати у довільній формі);

2.6 форму запису нормалізованого результату з плаваючою комою в пам'ять комп'ютера в прямому коді.

2.7 отриманий результат перевести у короткий формат за стандартом ANSI/IEEE 754-2008

Вказані пункти для операції додавання виконати тільки для етапу нормалізації результату з урахуванням можливого нульового результату. Інші дії крім етапу нормалізації результату можна проілюструвати у довільній формі.

Теоретичні відомості

Форми подання чисел у ЕОМ

В ОТ переважно використаються дві форми подання чисел:

– подання чисел з фіксованою комою;

– подання чисел із плаваючою комою.

Будь-яке число X у позиційній системі числення з основою k можна записати як

$$X = k^P M ,$$

де M –*мантиса* числа X, а P - *порядок* числа X.

Якщо порядок фіксований для всіх чисел, то говорять, що число подане у *формі з фіксованою комою*. В цьому випадку числа задаються тільки одним об'єктом – мантисою.

При поданні чисел у формі з фіксованою комою положення коми залишається незмінним для всіх чисел, з якими оперує машина, тобто спеціальні розряди для коми не виділяються. З метою спрощення

обчислення масштабних коефіцієнтів кому звичайно фіксують перед старшим розрядом мантиси або після молодшого розряду.

При фіксації коми перед старшим розрядом мантиси ЕОМ оперує із числами, які менше одиниці. Якщо число розрядів для запису мантис становить n, то мінімальне (за абсолютною величиною) число, що може бути подане в машині, дорівнює k^{-n}, а максимальне дорівнює $1-k^{-n}$.

У другому випадку, при фіксації коми після молодшого розряду, в машині виконуються операції над цілими числами, абсолютні величини яких перебувають у межах від 0 до $k^{-n}-1$.

Розрядна сітка для подання чисел у ЕОМ з фіксованою комою складається із двох частин: один розряд для подання знака, інші розряди для подання мантиси.

Якщо кожне число задається парою *P* і *M*, то таке подання називають формою *із плаваючої комою*.

При поданні чисел у формі із плаваючою комою порядок *P* може бути додатним або від'ємним цілим числом. Мантиса ж у більшості випадків є додатним або від'ємним правильним дробом, причому

$$k^{-1} \le M \le 1. \qquad (1)$$

Це означає, що старший розряд модуля мантиси завжди дорівнює 1 (мантиса нормалізована). Якщо в ЕОМ для запису порядку використається *m* розрядів, а для запису мантиси – *n* розрядів, то в цій машині може бути подане наступне максимальне (за абсолютною величиною) число

$$k^{k^m-1}(1-k^{-n}) = k^{k^m-1} - k^{k^m-n-1}.$$

У свою чергу, мінімальне за абсолютною величиною число, що відрізняється від нуля, у такій машині дорівнює

$$k^{-k^m+1}k^{-1} = k^{-k^m}.$$

Крім зазначених вище розрядів, для подання порядку й мантиси, у розрядній сітці ЕОМ із плаваючою комою є також розряди для подання знаків порядку й мантиси (рис. 2.1).

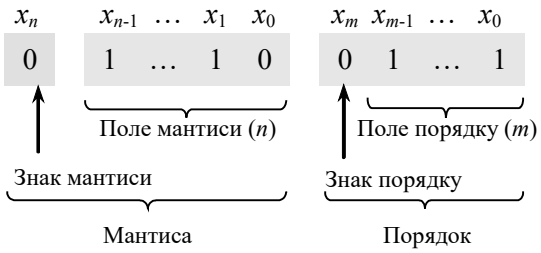


Рис. 2.1. Приклад запису числа у формі з плаваючою комою

Приклад.

Записати у формі із плаваючою комою (m=n=4) двійкові доповнювальні коди чисел: X=9$_{(10)}$, Y = – 9$_{(10)}$, Z = – 9/256$_{(10)}$.

$P_{(X)}$ =0,1 0 0 $\quad$ $M_{(X)}$=0 , 1001

$P_{(Y)}$ =0,1 0 0 $\quad$ $M_{(Y)}$=1 , 0111

$P_{(Z)}$ =1, 1 0 0 $\quad$ $M_{(Z)}$=1 , 0111

У ЕОМ, де використається подання чисел у формі із плаваючою комою, арифметичні операції виконуються як над мантисами чисел, так і над їхніми порядками. Часто також необхідно виконувати операцію нормалізації чисел, сутність якої складається у виконанні умови $k^{-1} \le M \le 1$. Тому машини із плаваючою комою є більш складним і менш швидкодіючим, ніж машини з фіксованою комою. З іншого боку, у машинах з фіксованою комою через масштабування виникають труднощі при програмуванні.

Для універсальних комп’ютерів розроблений стандарт ANSI/IEEE 754-2008 на подання чисел із плаваючою комою. В цьому стандарті визначені різні формати чисел. У кожному форматі використовують схований старший розряд мантиси з вагою 2^{-1}, який завжди дорівнює 1 для додатних

чисел. Таким чином, мантиси змінюються в межах $1 \le M < 2$. Вводиться "зміщений порядок" $P_{ЗМ} = P + (2^{m-1} - 1)$, де m – кількість розрядів для подання зміщеного порядку. Для 32-розрядної та 64-розрядної сітки визначені два основні базові формати: короткий та довгий

Короткий формат (рис. 2.2) має 32 розряди, з яких 8 розрядів призначені для подання порядку, 23 розряди - для подання мантис й один розряд - для знака мантиси. Спеціального розряду для знаку порядку немає. Таким чином, коди мулевого та від'ємних порядків *P* будуть подані "*зміщеними порядками*" $P_{зм}$, які є додатними числами з нулем в старшому розряді, а коди додатних порядків – з одиницями в старшому розряді. Введення "зміщеного порядку" дозволяє звести операції над порядками до арифметичної дії над цілими додатними числами. Для короткого формату $P_{ЗМ} = P + (2^{8-1} - 1) = P + 127$.

Отже, максимальний додатний порядок числа в короткому форматі дорівнює $11111111_{(2)} - 01111111_{(2)} = 10000000_{(2)} = 128_{(10)}$,

а максимальний за модулем від'ємний порядок

$$00000000_{(2)} - 01111111_{(2)} = -01111111_{(2)} = -127_{(10)}.$$

ЗМ	2^7	2^6	…	2^0	2^{-2}	2^{-3}	…	2^{-24}
Знак мантиси	Зміщений порядок (8 розрядів)				Мантиса (23 розряди)			

а

ЗМ	2^{10}	2^9	…	2^0	2^{-2}	2^{-3}	…	2^{-53}
Знак мантиси	Зміщений порядок (11 розрядів)				Мантиса (52 розряди)			

б

Рис. 2.2. Формати чисел із плаваючою комою: а – короткий формат; б – довгий формат.

Довгий формат, що використовується для обчислень із підвищеною точністю, має 64 розряди. Для порядку виділяється 11 розрядів, а для мантиси – 52 розряди.

Як видно з рис. 2, старший розряд мантиси з вагою $\mathbf{2^{-1}}$ не записується. В цьому розряді, якщо мантиса нормалізована, завжди буде одиниця і вона не записується для економії числа розрядів. Ця одиниця носить назву «прихована одиниця».

Додавання чисел із плаваючою комою

Суму двох чисел $X = 2^{P_X} M_X$ і $Y = 2^{P_Y} M_Y$, поданих у форматі із плаваючою комою, можна записати у вигляді

$$2^{P_X} M_X + 2^{P_Y} M_Y = 2^{P_Z} M_Z .$$

У загальному випадку дійсні ваги однакових розрядів мантиси різних чисел із урахуванням порядків чисел можуть відрізнятися. Тому для додавання чисел із плаваючою комою необхідно привести їх до загального порядку $P_{\text{заг}}$, в якості якого зручно обрати більший порядок з двох доданків

$$P_{\text{заг}} = \max(P_X, P_Y) .$$

При цьому зменшенню за рахунок зсуву праворуч підлягає мантиса числа з меншим порядком. В противному випадку виникає переповнення розрядної сітки мантиси числа, що перетворюється. Після цього суму двох чисел можна подати у вигляді

$$2^{P_{\text{заг}}} M_X + 2^{P_{\text{заг}}} M'_Y = 2^{P_{\text{заг}}} (M_X + M'_Y),$$

де за M'_Y прийнято перетворену мантису числа з меншим порядком.

Можна визначити наступні етапи додавання чисел із плаваючою комою. Будемо вважати, що мантиси чисел зберігаються у пам’яті у прямих кодах.

1. *Зрівняння порядків.* На цьому етапі виконується віднімання порядків доданків з метою визначення доданку з меншим порядком. Отримана різність порядків і її знак вказують, порядок якого доданку менше і на скільки розрядів необхідно зсунути його мантису праворуч. Далі у якості загального обирається більший порядок, а мантиса числа з меншим порядком зсувається вправо на різність порядків. При цьому молодші розряди мантиси можуть втрачатися, вслід чого у доданок вноситься похибка.

2. *Додавання мантис.* Виконується додавання мантис доданків за правилами складання чисел із знаками в модифікованих обернеіх або доповнювальних кодах.

3. *Нормалізація результату.* Мантиса результату може оказатися відразу нормалізованою або подана з порушенням нормалізації вліво чи вправо. В двох останніх випадках проводиться нормалізація результату, тобто приведення мантиси до вигляду що відповідає виразу (1).

Якщо при додаванні мантис виникає переповнення розрядної сітки, то таке переповнення говорить о *порушенні нормалізації вліво*. Ознакою порушення нормалізації вліво є розбіжність у знакових розрядах модифікованих кодів чисел. Для приведення результату до нормалізованої форми необхідно зсунути мантису результату вправо на один розряд, а загальний порядок збільшити на одиницю.

Нормалізовані мантиси додатних чисел в розряді з вагою 2^{-1} завжди мають одиницю. Нормалізовані мантиси від'ємних чисел, що подані обернеім або доповнювальним кодом, мають в цьому розряді нуль. Збіжність цифр у знаковому і старшому розряді мантиси у додатних і від'ємних чисел і говорить о *порушенні нормалізації вправо*. Для приведення до нормалізованого вигляду мантиса отриманого результату зсувається вліво до появи одиниці або нуля (залежно від використовуємих кодів) у

старшому розряді, а із загального порядку результату віднімається число одиниць, що дорівнює кількості зсувів мантиси.

Слід зазначити, якщо у результаті додавання отриманий нульовий результат, процес зсуву мантиси може виявитись нескінченним, тому кількість зсувів обмежується кількістю розрядів мантиси і результат операції подається *машинним нулем*.

4. *Формування результату.* На останньому етапі мантиса результату, що отримана у доповнювальному або обернеому коді подається у прямому коді. Далі до мантиси приписується загальний порядок доданків.

Вслід зсуву мантиси на етапі зрівняння порядків частина розрядів виходить з меж розрядної сітки і втрачається, тому після підсумовування може бути зроблено округлення результату, для цього до розряду, що вийшов із розрядної сітки останнім, додається одиниця з розповсюдженням переносив у старші розряди, що представляють результат у рамках розрядної сітки. Округлення може визвати порушення нормалізації вліво, тобто повторну нормалізацію.

Приклад.

Завдання. Виконати додавання чисел $Z = X + Y$, що подані у формі із плаваючою комою.

$$X = 0\ \ 1000\ \ 0\ \ 10111,$$

$$Y = 0\ \ 1010\ \ 0\ \ 11100.$$

Виконання завдання.

$$P_{(X)} = 0,1000 \qquad M_{(X)} = 0,10111$$

$$P_{(Y)} = 0,1010 \qquad M_{(Y)}\ \ 0,11100$$

1. Зрівняння порядків.

Виконаємо віднімання порядків доданків у вигляді $\Delta = P_{(X)} + (-P_{(Y)})$ з використанням доповнювального коду.

$$\begin{array}{ll} P_{(Y)\,[ДК]} & 1,0110 \\ & + \\ P_{(X)} & \underline{0,1000} \\ (P_{(X)}-P_{(Y)})_{[ДК]} & 1,1110 \end{array}$$

$$(P_{(X)}-P_{(Y)})_{[ПК]} \quad 1,0010$$

Отримана в результаті від'ємна різність порядків ($\Delta=-2_{(10)}$) вказує на те, що порядок числа *Y* більший за порядок числа *X*. Тому порядок цього числа обираємо як загальний порядок:

$$P_{(заг)}=0,1010$$

Приводимо порядок числа *X* до загального порядку, для чого збільшуємо його на різність порядків і зсуваємо мантису числа *X* на два розряди вправо:

$$P_{(заг)}=0,1010 \qquad M_{(X)}=0,0010111$$

2. Додавання мантис:

Мантиси є додатними числами, тому підсумовування виконуємо за правилами додавання звичайних додатних чисел, при чому застосовуємо модифікований код подання чисел:

$$\begin{array}{ll} M_{(X)} & 00,0010111 \\ & + \\ M_{(Y)} & \underline{00,1110000} \\ M_{(Z)} & 01,0000111 \end{array}$$

3. Нормалізація результату.

Отримана мантиса результату $M_{(Z)}$ не є нормалізованою, оскільки присутня розбіжність у знакових розрядах. Це говорить про переповнення розрядної сітки, тобто відбулося порушення нормалізації вліво.

Для приведення результату до нормалізованої форми зсунемо мантису результату на один розряд вправо, збільшимо загальний порядок на одиницю.

$$P'_{(заг)} = P_{(заг)} + 1$$

$$P'_{(заг)}=0,1011 \qquad M'_{(Z)}=0,10000111$$

4. Формування результату.

Округляємо мантису результату, для чого до розряду що вийшов із розрядної сітки останнім додається одиниця:

$$\begin{array}{rl|l} M'_{(Z)} & 0\,,\,10000 & 111 \\ + & & 1 \\ \hline M_{(Z)} & 0\,,\,10001 & \end{array}$$

До отриманої мантиси результату приписуємо загальний порядок:

$$P'_{(\text{заг})} = P_{(Z)}$$

$$P_{(Z)} = 0\,,\,1011 \qquad M_{(Z)} = 0\,,\,10001$$

Відповідь:

$Z = 0\ \ 1011\ \ 0\ \ 10001.$

Множення чисел із плаваючою комою

Множення двох чисел $X = 2^{P_X} M_X$ і $Y = 2^{P_Y} M_Y$, що задані у форматі із плаваючою комою, можна подати у вигляді

$$2^{P_Z} M_Z = 2^{P_X} M_X \cdot 2^{P_Y} M_Y = 2^{(P_X + P_Y)} \cdot (M_X \cdot M_Y)$$

При виконання операції множення мантис можна отримати результат із порушенням нормалізації вправо. Дійсно, якщо мантиси множників подані у нормалізованій формі

$$2^{-1} \le M_X, M_Y < 1,$$

то результат може знаходитися у границях

$$2^{-2} \le M_Z < 1.$$

Етапи множення чисел із плаваючою комою.

1. Одержання порядку результату у вигляді суму порядків множників

$$P_Z = P_X + P_Y.$$

2. Знаходження мантиси результату

$$M_Z = M_X \cdot M_Y .$$

3. Нормалізація результату, тобто приведення мантиси результату до вигляду

$$2^{-1} \le M_Z < 1.$$

Приклад.

Завдання. Подати числа у формі з плаваючою комою и знайти добуток Z = X ·Y, якщо:

X = 10,101,

Y = 0,1011.

Виконання завдання.

Приведемо задані числа до форми:

$X = 2^{P_X} M_X = 2^2 \cdot 0{,}10101$

$Y = 2^{P_Y} M_Y = 2^0 \cdot 0{,}1011$

Отримали порядки і мантиси:

$P_{(X)}$ =0 , 10 $M_{(X)}$=0 , 10101

$P_{(Y)}$ =0 , 00 $M_{(Y)}$=0 , 10110

Визначимо порядок результату, виконавши додавання порядків множників $P_Z = P_X + P_Y = 2 + 0 = 2$. Після множення мантис отримаємо:

P_Z =0 , 10 M_Z =0 , 0111001|1|1|0

Мантиса результату не є нормалізованою. Для виконання нормалізації мантису результату зсуваємо ліворуч на один розряд і виконуємо корекцію порядку $P_Z = P_Z - 1 = 1$. Отримаємо:

$P_{(Z)}$ =0 , 01 M_Z =0 , 1110011|1

Відповідь одержуємо після округлення мантиси:
$Z = 0\ \ 01\ \ 0\ \ 11101.$

Ділення чисел із плаваючою комою

Результат ділення двох чисел $X = 2^{P_X} M_X$ і $Y = 2^{P_Y} M_Y$, що задані у форматі із плаваючою комою, можна записати у вигляді

$$2^{P_Z} M_Z = \frac{2^{P_X} M_X}{2^{P_Y} M_Y} = 2^{(P_X - P_Y)} \cdot \frac{M_X}{M_Y}$$

Ділення мантис повинно виконуватись при виконанні умови

$$M_X < M_Y, \qquad (2)$$

яка не завжди виконується при поданні мантис у нормалізованій формі. Тому перед началом ділення мантису діленого завжди зсувають вправо, чим забезпечують зменшення її у два рази. Відповідно до порядку діленого додається одиниця, тобто

$$2^{P_X} M_X = 2^{P_X+1} \cdot M_X \cdot 2^{-1}. \qquad (3)$$

Після цього маємо

$$2^{-2} \le M_X < 2^{-1},$$

$$2^{-1} \le M_Y < 1,$$

$$2^{-2} \le M_Z < 1.$$

Етапи ділення чисел із плаваючою комою.

1. Одержання порядку результату із урахуванням виконання умови (2) шляхом віднімання

$$P_Z = P_X - P_Y.$$

2. Одержання мантиси результату. На цьому етапі виконується ділення мантис: $M_Z = \frac{M_X}{M_Y}$.

3. Нормалізація результату. Виконується аналогічно нормалізації при множенні чисел із плаваючою комою.

Приклад.

Завдання. Виконати ділення чисел $Z = \frac{X}{Y}$, що подані у формі із плаваючою комою.

X = 0 10 0 1011,

Y = 0 01 0 1001.

Виконання завдання.

Маємо ділене і дільник, задані у формі:

$$P_X = 0,10 \qquad M_X = 0,1011$$

$$P_Y = 0,01 \qquad M_Y = 0,1001$$

Застосуємо перетворення (3) для забезпечення умови (2). Отримаємо змінену мантису і порядок діленого *X*:

$$P_X = 0,11 \qquad M_{(X)} = 0,01011$$

Визначимо порядок результату.

$P_Z = P_X - P_Y = 3 - 1 = 2$, тобто

$$P_Z = 0,10$$

Визначимо мантису результату діленням мантис і одержуємо результат:

$$M_Z = 0,1100|1$$

Отриману в результаті мантису частки округляємо і усікаємо до розмірів розрядної сітки:

$$M_Z = 0,1101|$$

Мантиса результату є нормалізованою, тому сформуємо результат обчислення:

$$P_Z = 0,10 \qquad M_Z = 0,1100$$

Відповідь:

Z = 0 01 0 1100.

3. Побудова функціональних схем

Для операції з номером $x_3x_2x_1$ додатково виконати:

3.1 побудувати на базі операторної схеми функціональну схему з відображенням управляючих сигналів, входів для запису операндів при ініціалізації пристрою і схем формування внутрішніх логічних умов; схему подати як рисунок у пояснювальній записці.

3.2 розробити структурний мікроалгоритм (мікрооперації замінюються управляючими сигналами виду W, SL, SR тощо), а також закодований мікроалгоритм (сигнали, що завжди формуються разом, позначаються одним символом);

3.3 якщо в отриманому мікроалгоритмі в двох суміжних операторних вершинах виконуються різні мікрооперації на одному і тому же регістрі/лічильнику, то між вказаними операторними вершинами треба додати порожню вершину; при цьому різні управляючи сигнали на регістрі не будуть перекриватися в часі, тобто не виникне збій в роботі пристрою

3.4 для операції з парним номером $x_3x_2x_1$ подати граф управляючого автомата Мура з відображенням кодів вершин, а для непарного номера – граф автомата Мілі;

3.5 побудувати управляючий автомат на тригерах та елементах булевого базису. Відповідно із кодом x_2x_1, що приймає значення $00, 01, 10, 11$, вибрати відповідно JK-, RS-, T-, D-тригери для побудови автомата; виконати сумісну мінімізацію функцій управління тригерами та вихідних сигналів;

3.6 у форматі А2 або А3 виконати креслення функціональної схеми управляючого автомата з урахуванням діючих стандартів для схем Е2.

3.7 у моделюючій програмі ПРОГМОЛС (AFDK) побудувати операційний пристрій і керуючий автомат виконання даної операції. Надати

часову діаграму роботи автомата. Виконати числовий приклад за варіантом. Зробити скріншот побудованої схеми з числовим результатом.

4. Методичні вказівки до виконання курсової роботи

КР повинна містити наступні документи (в порядку їх комплектування):

- титульний лист (додаток 1);
- опис альбому;
- сторінка з написом у середині аркушу "Технічне завдання".
- технічне завдання;
- сторінка з написом у середині аркушу "Автомат керуючий. Схема електрична функціональна";
- автомат керуючий, схема електрична функціональна;
- сторінка з написом у середині аркушу "Пояснювальна записка";
- пояснювальна записка.

Усі скомплектовані документи виконуються в рамках формату А4 з відповідними штампами і скріплюються в один альбом.

Технічне завдання розробляється на підставі вихідних даних відповідно до діючих стандартів. У технічному завданні повинні бути наступні розділи:

- *призначення* розроблювального об'єкта, в якому розкриваються його області застосування;
- *вихідні дані* для розробки;
- *склад* пристроїв, в якому приводиться перелік основних складових частин проектованого пристрою;
- *етапи* проектування і терміни їх виконання;
- *перелік* текстової і графічної документації.

Технічне завдання повинне бути підписане виконавцем і керівником проекту.

У пояснювальній записці повинні бути наступні розділи:

ВСТУП

1 ЗАВДАННЯ НА КУРСОВУ РОБОТУ

2 ДОСЛІДЖЕННЯ СПОСОБІВ ВИКОНАННЯ ОПЕРАЦІЙ

3 ПОБУДОВА ФУНКЦІОНАЛЬНИХ СХЕМ

ВИСНОВКИ

СПИСОК ЛІТЕРАТУРИ

Порядок побудови розділів і підрозділів пояснювальної записки, правила подання тексту повинні цілком відповідати вимогам діючих стандартів.

Перед комплектацією всі документи курсової роботи повинні бути підписані виконавцем і керівником на титульному листі і у відповідних штампах на документах і креслень.

Функціональна схема визначає основні функціональні частини виробу, їх призначення і взаємозв'язок, роз'ясняє визначені процеси, що протікають в окремих функціональних частинах або у виробі в цілому. Функціональні частини таких схем зображуються, як правило, у виді прямокутників.

У функціональних схемах використовуються умовні графічні позначення (УГП) функціональних частин. УГП елемента має форму прямокутника, до якого підводять лінії виводів. УГП елемента в загальному випадку може містити три поля: основне і два додаткових, котрі розташовують ліворуч і праворуч від основного (рис. 4.1).

В основному полі УГП поміщають позначення функції, яка реалізується елементом (табл. 4.1). В додаткових полях поміщають інформацію про призначення виводів (мітки виводів, вказівники).

Крім виду, зазначеного на рис 4.1, УГП може також складатися:

- тільки з основного поля;
- з основного поля і одного додаткового (праворуч або ліворуч від основного);

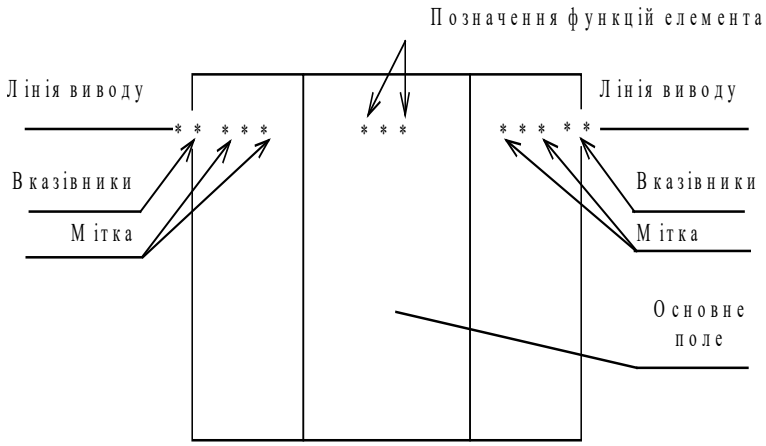


Рисунок 4.1 - Умовне графічне позначення елементу

Допускаються додаткові поля розділяти на зони, відокремлені горизонтальною рискою. Основне і додаткове поля можуть бути не відділені лінією. При цьому відстань між буквенними, цифровими і буквенно-цифровими позначеннями, поміщеними в основне і додаткові поля, визначається однозначністю розуміння кожного позначення.

Входи елемента зображують з лівої сторони УГП, виходи – з правої сторони УГП. Двонаправлені виводи і виводи, що не несуть логічної інформації, зображують з правої або лівої сторони УГП.

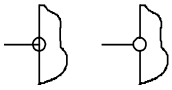
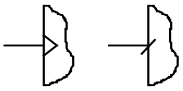
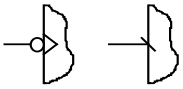
При підведенні ліній виводів до контуру УГП не допускається:

- проводити їх на рівні сторін прямокутника;
- проставляти на них стрілки, що вказують напрямок передачі інформації.

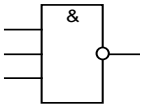
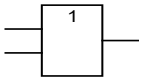
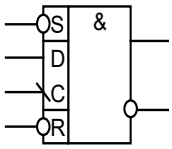
Таблиця 4.1 - Позначення елементів

Найменування	Позначення
Тригер	T
Двоступінчастий тригер	TT
Логічне “І”	&
Логічне “АБО”	1
Виключне “АБО”	=1
Генератор імпульсів	G

Таблиця 4.2 - Позначення входів елементів

Найменування	Позначення
Прямий статичний вхід	
Інверсний статичний вхід	
Прямий динамічний вхід	
Інверсний динамічний вхід	

Таблиця 4.3 – Приклади УГП елементів

Найменування	Позначення
Елемент 3І-НЕ	
Елемент 2АБО	
D-тригер	

Розміри сторін УГП і відстані між виводами повинні бути кратні М. Діаметр вказівника інверсного виводу повинний дорівнювати М. При виконанні курсової роботи прийняти М=5 мм.

Позначення функцій, виконуваних елементом, утворюють із прописних букв латинського алфавіту, арабських цифр і спеціальних знаків, записаних без пробілів. Кількість знаків у позначенні функції не обмежена, однак варто прагнути до їх мінімального числа при збереженні однозначності розуміння кожного позначення.

В таблиці 4.1 приведені деякі стандартні позначення функцій, яких варто дотримуватись при виконанні курсової роботи.

Виводи елементів, які несуть логічну інформацію, підрозділяють на статичні і динамічні, а також на прямі та інверсні. Властивості виводів позначають за допомогою міток відповідно до табл. 4.2.

Приклади деяких УГП приведені в табл. 4.3.

Захист курсових робіт є особливою формою перевірки якості виконання роботи і знань студента в даній області. Захист полягає в короткій доповіді (5-7 хв.) студента по виконаній роботі і відповідях на питання.

5. Критерії оцінювання курсової роботи

- Оформлення документів згідно стандартам: 0 – 20 балів;
- Дослідження способів виконання операцій 0-30 балів;
- Розробка схеми автомата керуючого з пам`яттю 0-40 балів;
- Відповіді на теоретичні запитання викладача: 0-10 балів.

Якщо в конструкторській документації проекту будуть виявлені грубі порушення стандартів або виявиться, що спроектований пристрій непрацездатний, робота повертається на доробку.

Література

Базова:

1. Жабін В.І., Верба О.А. Комп'ютерна логіка. Курсова робота. Навчальний посібник. https://ela.kpi.ua/handle/123456789/50134

2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. Прикладна теорія цифрових автоматів: Навч. посібник.– К.: Вид-во НАУ, 2009.– 364 с. https://campus.kpi.ua/tutor/index.php?mode=mob&show&irid=200206

3. Жабін В.І., Ткаченко В.В. Цифрові автомати. Практикум. – К.: ВЕК+, 2004.– 160 с.

Допоміжна:

5. Матвієнко М.П. Комп'ютерна логіка. Підручник. Вид. 2-ге перероб. та доп. – Київ: Видавництво Ліра – К, 2017. – 324 с.

6. Лупенко С.А., Пасічник В.В., Тиш Є.В. Комп'ютерна логіка. Навчальний посібник для ВНЗ. Вид. Магнолія, 2017.– 354 с.

7. ДСТУ 3008-2015 «Державний стандарт України. Документація. Звіти в сфері науки і техніки. Структура і правила оформлення».

8. ДСТУ ISO 5457:2006 (ISO 5457:1999, IDT) Національний стандарт України. Документація технічна на вироби. Креслеників. Розміри та формати.

9. ДСТУ ГОСТ 2.702:2013 ЄСКД. Правила виконання електричних схем (ГОСТ 2.702-2011, IDT).

10. ДСТ 2.104-2006. ЄСКД. Основні написи (Міждержавний. В Україні -ДСТУ 2.104-2006).

11. Рамки і штампи. https://sprosi.xyz/articles/ramka-dlya-kursovyh-diplomov-i-referata/

Додаток А. Титульний аркуш

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

КУРСОВА РОБОТА

з дисципліни "Комп’ютерна логіка"

на тему: Пристрій виконання арифметичної операції

Виконав _____

(прізвище, ініціали)

Група _____

Варіант № _____

Допущений до захисту _____

_____

(оцінка, підпис керівника)

Київ – 202_ р.

Додаток Б. Оформлення креслення, таблиць та рисунків

Креслення «Автомат керуючий, схема електрична функціональна Е2» виконується на аркуші формату А2 або А3 у редакторі Visio 2010

Вимоги до функціональної схеми автомата управління.

Елементи розміщуються колонками.

На логічних елементах основне поле не менше 10 мм.

Додаткові поля не менше 5 мм.

Всі розміри елементів по висоті кратні 5 мм.

Між лініями входів та виходів елементів не менше 5 мм.

Лінії можна збирати у джгути. Кожна лінія знаходиться по номеру входу в джгут та номеру виходу із джгута. В різних джгутах своя нумерація. Входи в джгут обов'язково в порядку 1, 2, 3 … Виходи по будь якому порядку.

Елементи схеми не можна розміщувати нижче верхньої границі штампу (це місце для пояснень розробників та служб контролю).

Вимоги до надписів таблиць та рисунків.

Шрифт "Times new roman", 12 (в таблицях можна менше), інтервал 1,5, відступ 1,25 см., рамки, штампи.

Всі розділи, рисунки, таблиці нумеруються. Текст роздруковується на принтері. Далі приклади.

Таблиця 3.3 – Структурна таблиця автомата (крап. немає)

Рисунок 1.1 – Операційна схема множення першим способом **(крап.немає)**

Додаток В. Приклад виконання курсової роботи

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

КУРСОВА РОБОТА

з дисципліні "Комп’ютерна логіка"

на тему: Пристрій виконання арифметичної операції

Виконав:

Група ІО-36

Варіант №3607

Допущений до захисту

_____

(оцінка, підпис керівника)

Київ – 2024 р.

Опис альбому

№ рядка	Формат	Позначення	Найменування	Кіль-сть	Примітка
1			Документація загальна		
2					
3			Наново розроблена		
4					
5	А4	КПІ.ФІОТ.ІО3607.001 ОА	Курсова робота Опис альбому	1	
6					
7	А4	КПІ.ФІОТ.ІО3607.002 ТЗ	Курсова робота Технічне завдання	5	
8					
9					
10	А2	КПІ.ФІОТ.ІО3607.003 Е2	Автомат керуючий. Схема електрична функціональна	1	
11					
12	А4	КПІ.ФІОТ.ІО3607.004 ПЗ	Курсова робота Пояснювальна записка	46	
13					
14					
15					
16					
17					
18					
22					
23					
24					
25					

Зм.	Арк.	№ докум.	Підпис	Дата	**КПІ.ФІОТ.ІО3607.001 ОА**
Розробив	Данилюк В.І.				*КУРСОВА РОБОТА* *Опис альбому*
Перевірив	Верба О.А.				Літера / Аркуш: 1 / Аркушів: 1
Реценз.					**НТУУ «КПІ» ФІОТ** **Група ІО-36**
Н. контр.					
Затверд.	Жабін В.І.				

Технічне завдання

Зміст

1. ***ПРИЗНАЧЕННЯ РОЗРОБЛЮВАЛЬНОГО ОБ'ЄКТУ..... 2***
2. ***ВИХІДНІ ДАНІ ДЛЯ РОЗРОБКИ 2***
3. ***СКЛАД ПРИСТРОЇВ 4***
4. ***ЕТАПИ І ТЕРМІНИ ПРОЕКТУВАННЯ 4***
5. ***ПЕРЕЛІК ТЕКСТОВОЇ І ГРАФІЧНОЇ ДОКУМЕНТАЦІЇ 5***

1. ПРИЗНАЧЕННЯ РОЗРОБЛЮВАНОГО ОБ'ЄКТУ

У курсовій роботі необхідно виконати арифметичні операції між числами у форматі з плаваючою комою.

Призначенням арифметичних операцій є дослідження та визначення способів, за допомогою яких комп'ютери виконують арифметичні операції. Ці розділи допомагають розуміти, як комп'ютери здійснюють обчислення, зокрема, додавання, віднімання, множення та ділення. Вони вивчаються з метою розробки та оптимізації арифметичних операцій у комп'ютерах, що важливо для підвищення їх продуктивності та ефективності.

2. ВИХІДНІ ДАНІ ДЛЯ РОЗРОБКИ

Варіант індивідуального завдання визначається вихідними значеннями операндів X і Y, які визначаються номером групи і номером студента у групі наступним чином. Перевести номер варіанту в двійкову систему.

Номер залікової книжки – $3607_{10} = 1110\ 0001\ 0111_2$.

Записати два 10-розрядних двійкових числа:

$X = -001100{,}1111$ і $Y = +010001{,}0111$

Числа і в прямому коді записати у формі з плаваючою комою у класичному варіанті (з незміщеним порядком і повною мантисою). На мантису відвести 6 основних розрядів, на порядок студент самостійно вибирає необхідну кількість розряді, один знаковий розряд для мантиси та один знаковий розряд для порядку числа.

Виконати 8 операцій з числами, що подані з плаваючою комою в класичному варіанті: чотири способи множення, два способи ділення, додавання та обчислення кореня додатного числа. Операндами для першого способу множення є задані числа X та Y. Для кожної наступної операції першим операндом є результат попередньої операції, а другим операндом завжди є число Y.

Перед виконанням чергової операції треба представити необхідні операнди в класичному вигляді з порядком та мантисою.

Для операції з номером $x_3x_2x_1$ додатково виконати:

1. побудувати на базі операторної схеми функціональну схему з відображенням управляючих сигналів, входів для запису операндів при ініціалізації пристрою і схем формування внутрішніх логічних умов; схему подати як рисунок у пояснювальній записці.
2. розробити структурний мікроалгоритм (мікрооперації замінюються управляючими сигналами виду W, SL, SR тощо), а також закодований мікроалгоритм (сигнали, що завжди формуються разом, позначаються одним символом);
3. якщо в отриманому мікроалгоритмі в двох суміжних операторних вершинах виконуються різні мікрооперації на одному і тому же регістрі/лічильнику, то між вказаними операторними вершинами треба додати порожню вершину; при цьому
   різні управляючи сигнали на регістрі не будуть перекриватися в часі, тобто не виникне збій в роботі пристрою
4. для операції з парним номером $x_3x_2x_1$ подати граф управляючого автомата Мура з відображенням кодів вершин, а для непарного номера – граф автомата Мілі;
5. побудувати управляючий автомат на тригерах та елементах булевого базису.
   Відповідно із кодом x_2x_1, що приймає значення 00, 01, 10, 11, вибрати відповідно
   JK-, RS-, T-, D-тригери для побудови автомата; виконати сумісну мінімізацію функцій управління тригерами та вихідних сигналів;
6. у форматі А2 або А3 виконати креслення функціональної схеми управляючого автомата з урахуванням діючих стандартів для схем Е2.
7. у моделюючій програмі ПРОГМОЛС (AFDK) побудувати операційний пристрій і керуючий автомат виконання даної операції. Надати часову діаграму роботи автомата. Виконати числовий приклад

за варіантом.

Зробити скріншот побудованої схеми з числовим результатом.

3. СКЛАД ПРИСТРОЇВ

Керуючий цифровий автомат:

Керуючий автомат містить у собі комбінаційну схему (КС) і пам'ять, що складається з тригерів. Входами КС є виходи тригерів і логічні умови, входами тригерів є логічні умови.

4. ЕТАПИ І ТЕРМІНИ ПРОЕКТУВАННЯ

1. Дослідження способів виконання операцій:
   1.1. теоретичне обґрунтування способу виконання операції;
   1.2. операційну схему;
   1.3. змістовний мікроалгоритм виконання операції;
   1.4. таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 6 основних розрядів мантиси результату;
   1.5. обробку порядків;
   1.6. форму запису нормалізованого результату з плаваючою комою в пам'ять комп'ютера в прямому коді;
   1.7. форму запису нормалізованого результату з плаваючою комою в пам'ять комп'ютера в прямому коді.
2. Для операції з номером $x_3x_2x_1$ додатково зробити:
   2.1. фукнціональну схему;
   2.2. структурний мікроалгоритм (якщо в отриманому мікроалгоритмі в двох суміжних операторних вершинах виконуються різні мікрооперації на одному і тому же регістрі/лічильнику, то між вказаними операторними вершинами треба додати порожню вершину);

2.3. для операції з парним номером $x_3x_2x_1$ подати граф управляючого автомата Мура з відображенням кодів вершин, а для непарного номера – граф автомата Мілі;

2.4. побудувати управляючий автомат на тригерах та елементах булевого базису.

Відповідно із кодом x_2x_1, що приймає значення 00, 01, 10, 11, вибрати відповідно JK-, RS-, T-, D-тригери для побудови автомата;

2.5. у форматі А2 або А3 виконати креслення функціональної схеми управляючого автомата з урахуванням діючих стандартів для схем Е2.

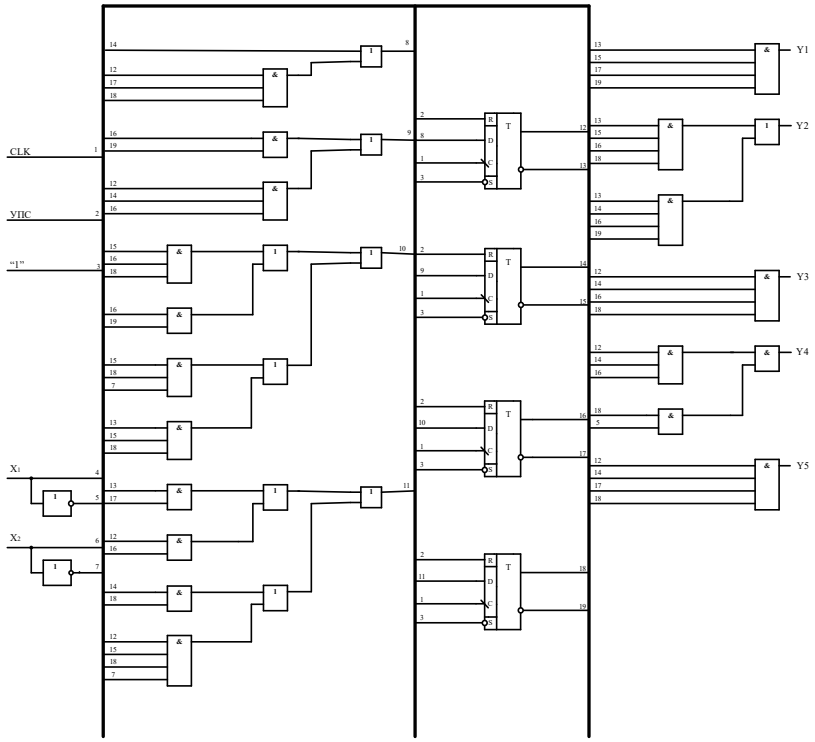
2.6. у моделюючій програмі ПРОГМОЛС (AFDK) побудувати операційний пристрій і керуючий автомат виконання даної операції. Надати часову діаграму роботи автомата. Виконати числовий приклад за варіантом. Зробити скріншот побудованої схеми з числовим результатом.

5. ПЕРЕЛІК ТЕКСТОВОЇ І ГРАФІЧНОЇ ДОКУМЕНТАЦІЇ

- Титульний лист;
- Сторінка з написом “Опис альбому”;
- Опис альбому;
- Сторінка з написом “Технічне завдання”;
- Технічне завдання;
- Сторінка з написом “Автомат керуючий. Схема електрична функціональна”;
- Керуючий автомат, схема електрична функціональна;
- Сторінка з написом “Пояснювальна записка”;
- Пояснювальна записка.

Автомат керуючий

Схема електрична функціональна



					ФІОТ.ІО3607.003 Е2			
Змн	Арк	№ докум	Підпис	Дата	Автомат керуючий Схема електрична функціональна	Літ	Маса	Масштаб
Розроб.		Данилюк В. І.						
Перевірив						Аркуш 1	Аркушів 1	
Н. Контр.					НТУУ «КПІ» ФІОТ ГРУПА ІО-36			
Затверд.		Жабін В.І.						

Пояснювальна записка

Зміст

1. ***ВСТУП*** 2
2. ***ЗАВДАННЯ НА КУРСОВУ РОБОТУ*** 2
   - *2.1 Формат числа з плаваючою комою* 2
3. ***ДОСЛІДЖЕННЯ СПОСОБІВ ВИКОНАННЯ ОПЕРАЦІЙ*** 3
   - *3.1 Перший спосіб множення* 3
   - *3.2 Другий спосіб множення* 6
   - *3.3 Третій спосіб множення* 10
   - *3.4 Четвертий спосіб множення* 13
   - *3.5 Перший спосіб ділення* 18
   - *3.6 Другий спосіб ділення* 22
   - *3.7 Додавання чисел* 26
   - *3.8 Добування кореню додатного числа* 28
4. ***ПОБУДОВА ФУНКЦІОНАЛЬНИХ СХЕМ*** 32
   - *4.1 Функціональна схема* 32
   - *4.2 Структурний мікроалгоритм* 32
   - *4.3 Автомат Мілі* 34
5. ***ВИСНОВОК*** 42
6. ***СПИСОК ЛІТЕРАТУРИ*** 43

1. ВСТУП

Курсова робота виконана за номером технічного завдання 3607_{10} => **111000010111_2**, і складається з двох частин: дослідження операційних способів виконання операцій та побудова функціональних схем. Вихідними даними при дослідженні операційних способів виконання операцій є результат операцій. Вхідними даними при синтезі автомата є заданий алгоритм, тип тригера та елементна база.

2. ЗАВДАННЯ НА КУРСОВУ РОБОТУ

2.1 Формат числа з плаваючою комою

Отже, операнд X=-001100.1111; Y=+010001.0111;

Для операції з номером $x_3x_2x_1 = 111_2 = 7_{10}$ виконати четвертий пункт. Також подати граф автомата Мілі. Відповідно з кодом $x_2x_1 = 11$, тобто на тригерах D побудувати автомат.

В ПК:

X = 1.001100.1111

Y = 0.010001.0111

$P_X = 4$

$P_Y = 5$

$M_X = 110011$

$M_Y = 100010$

Числа у формі плаваючої коми:

X = 0 00100 1 110011

Y = 0 00101 0 100010

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 3 = 131_{10} = 10000011_2$

$E_Y = 128 + (P_Y - 1) = 128 + 4 = 132_{10} = 10000100_2$

X = 1 10000011 10011110000000000000000

Y = 0 10000100 00010111000000000000000

3. ДОСЛІДЖЕННЯ СПОСОБІВ ВИКОНАННЯ ОПЕРАЦІЙ

3.1 Перший спосіб множення

Операнди:

X = 0 00100 1 110011

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

Множення здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим.

Операційна схема

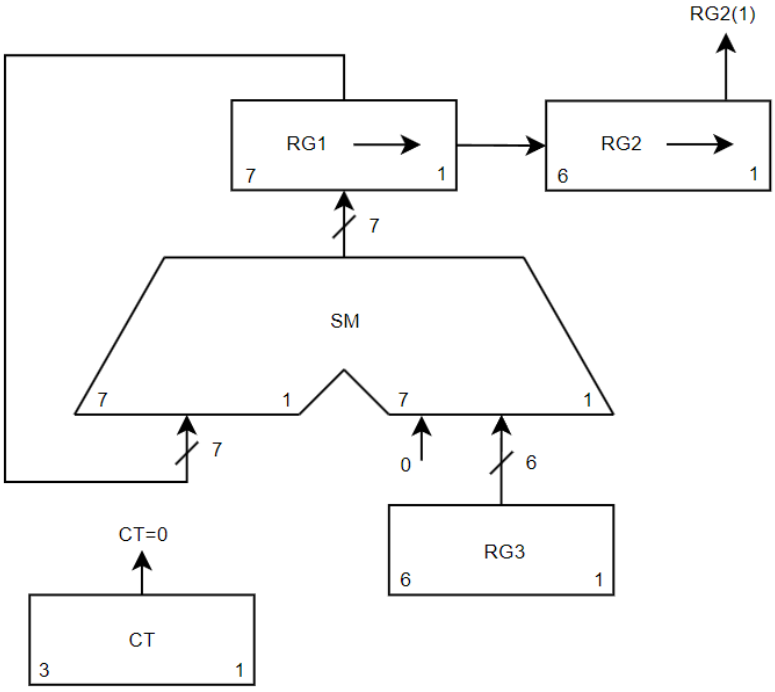


Рисунок 1 – Операційна схема першого способу множення

Змістовний мікроалгоритм

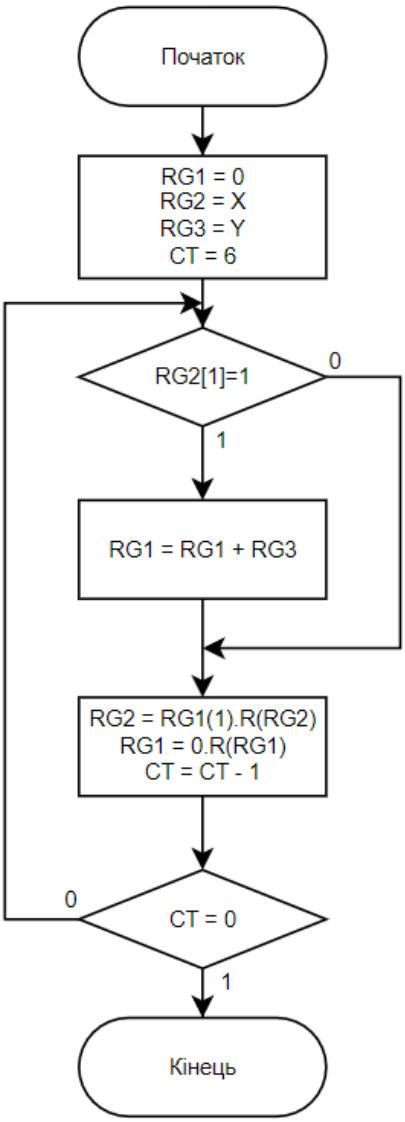


Рисунок 2 – Змістовний мікроалгоритм першого способу множення

Таблиця станів регістрів

$M_X * M_Y = 110011 * 100010$
1) $RG_1 = 0$; $RG_2 = X$; $RG_3 = 0.Y$; $CT = 6$

2) $RG_1 = RG_1 + RG_3$; $RG_2 = RG_1(1).R(RG_2)$; $RG_1 = 0.R(RG_1)$; $CT = CT - 1$

3) $RG_2 = RG_1(1).R(RG_2)$; $RG_1 = 0.R(RG_1)$; $CT = CT - 1$

Таблиця 1 – Стани регістрів під час першого способу множення

№	RG_1	RG_2	RG_3	CT	Мікрооперації
-	0000000	110011	0100010	110	1
1	0000000 + 0100010 0100010 0010001	110011 011001		101	2
2	0010001 + 0100010 0110011 0011001	011001 101100		100	2
3	0011001 0001100	101100 110110		011	3
4	0001100 0000110	110110 011011		010	3
5	0000110 + 0100010 0101000 0010100	011011 001101		001	2
6	0010100 + 0100010 0110110 0011011	001101 000110		000	2

Відповідь: 0,011011000110

Обробка порядків

$P_Z = P_X + P_Y$

$P_X = 4$

$P_Y = 5$

$P_Z = 9_{10} = 01001_2$

Форма запису нормалізованого результату

Результат множення: 0,011011000110

Знак мантиси: $1 \oplus 0 = 1$

Нормалізований результат:

X = 0 01000 1 110110

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 7 = 135_{10} = 10000111_2$

X = 1 10000111 10110001100000000000000

3.2 Другий спосіб множення

Операнди:

X = 0 01000 1 110110

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

Множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

Операційна схема

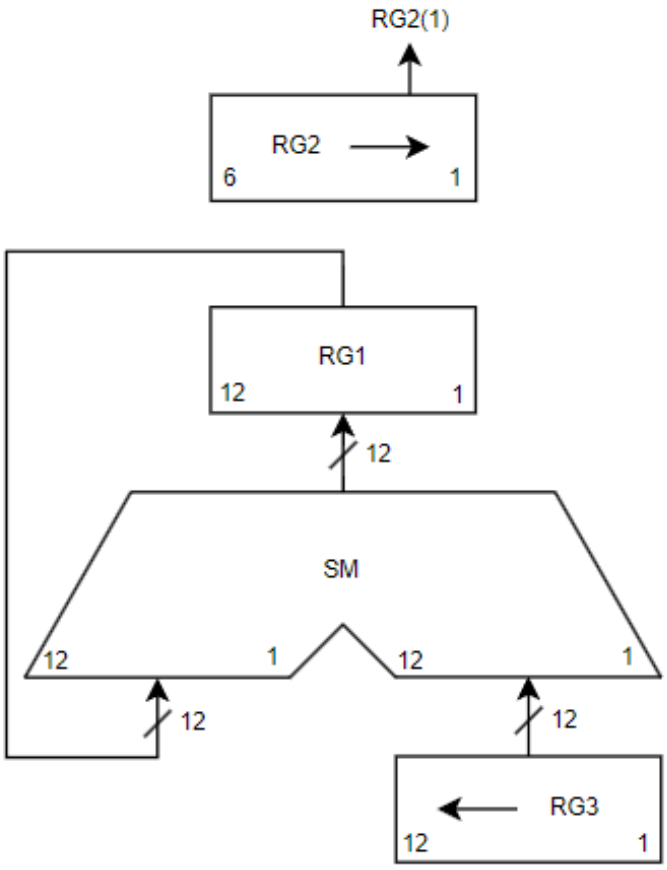


Рисунок 3 – Операційна схема другого способу множення

Змістовний мікроалгоритм

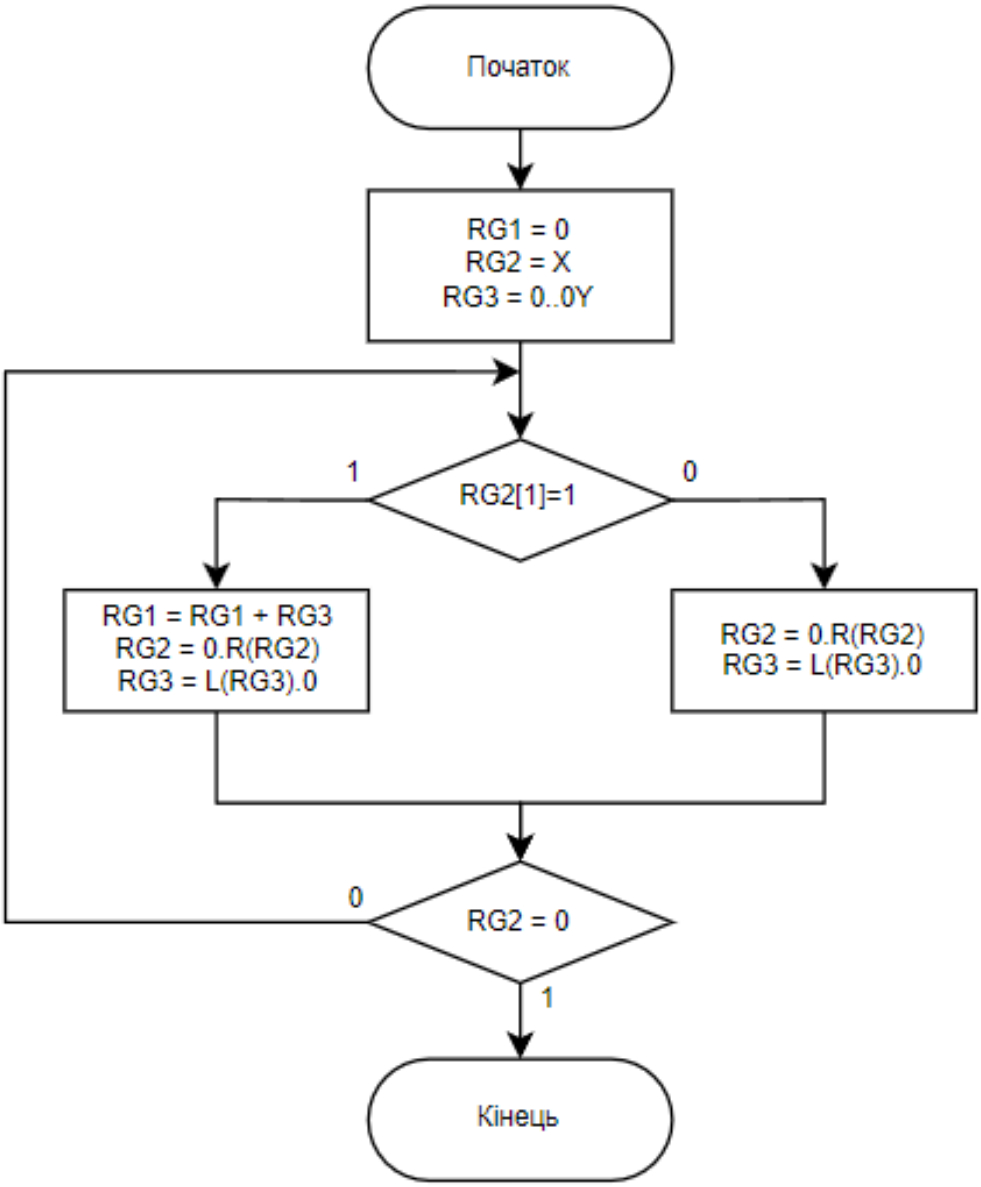


Рисунок 4 – Змістовний мікроалгоритм другого способу множення

Таблиця станів регістрів

$M_X * M_Y = 110110 * 100010$
1) $RG_1 = 0$; $RG_2 = X$; $RG_3 = 0..0Y$;

2) $RG_1 = RG_1 + RG_3$; $RG_2 = 0.R(RG_2)$; $RG_3 = L(RG_3).0$

3) $RG_2 = 0.R(RG_2)$; $RG_3 = L(RG_3).0$

Таблиця 2 – Стани регістрів під час другого способу множення

№	RG_1	RG_2	RG_3	Мікрооперації
-	000000000000	110110	000000100010	1
1	000000000000	110110 011011	000000100010 000001000100	3
2	000000000000 + 000001000100 000001000100	011011 001101	000001000100 000010001000	2
3	000001000100 + 000010001000 000011001100	001101 000110	000010001000 000100010000	2
4	000011001100	000110 000011	000100010000 001000100000	3
5	000011001100 + 001000100000 001011101100	000011 000001	001000100000 010001000000	2
6	001011101100 + 010001000000 011100101100	000001 000000	010001000000 100010000000	2

Відповідь: 0,011100101100

Обробка порядків

$P_Z = P_X + P_Y$

$P_X = 8$

$P_Y = 5$

$P_Z = 13_{10} = 01101_2$

Форма запису нормалізованого результату

Результат множення: 0,011100101100

Знак мантиси: $1 \oplus 0 = 1$

Нормалізований результат:

X = 0 01100 1 111001

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 11 = 139_{10} = 10001011_2$

X = 1 10001011 11001011000000000000000

3.3 Третій спосіб множення

Операнди:

X = 0 01100 1 111001

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

Множення здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме.

Операційна схема

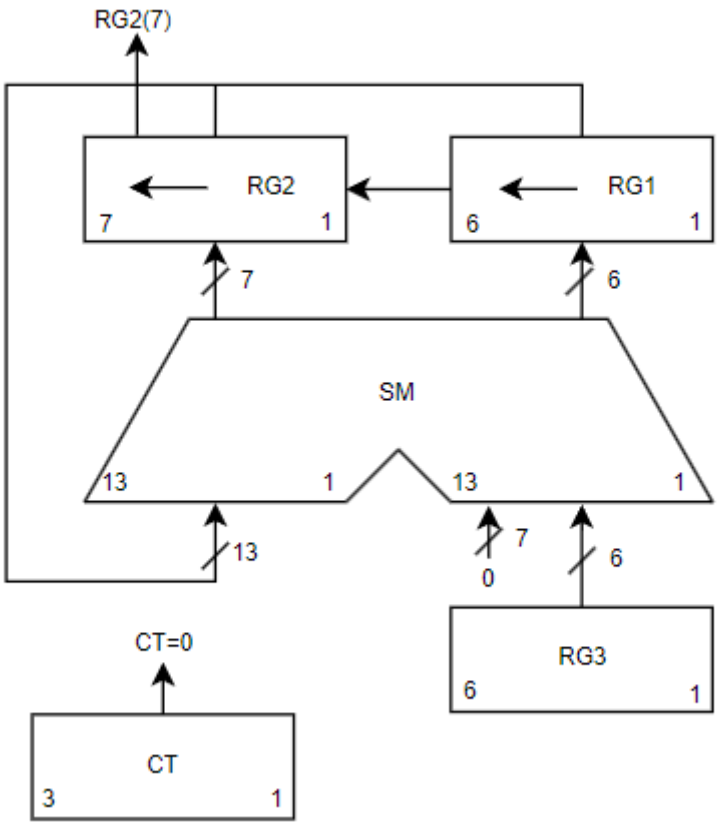


Рисунок 5 – Операційна схема третього способу множення

Змістовний мікроалгоритм

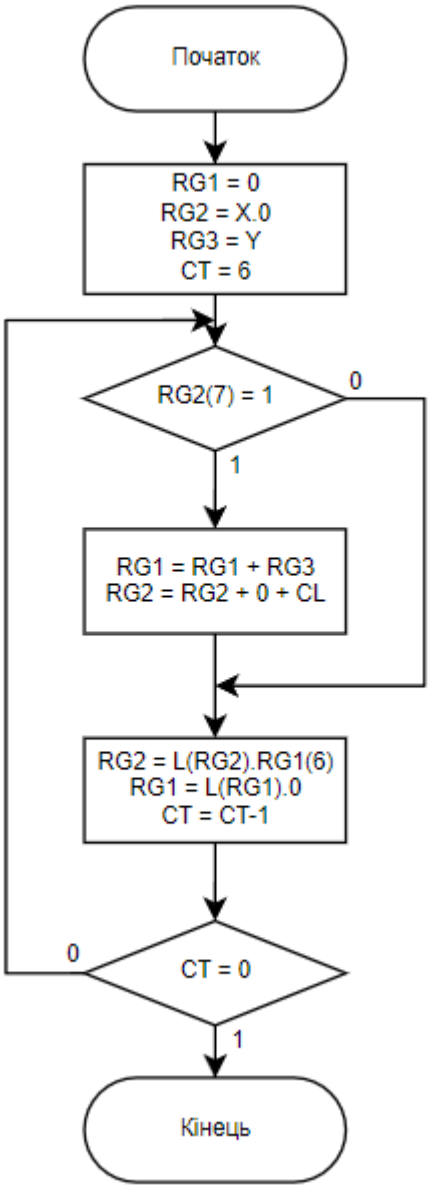


Рисунок 6 – Змістовний мікроалгоритм третього способу множення

Таблиця станів регістрів

$M_X * M_Y = 111001 * 100010$
1) $RG_1 = 0$; $RG_2 = X.0$; $RG_3 = Y$; $CT = 6$

2) $RG_1 = RG_1 + RG_3$; $RG_2 = RG_2 + 0 + CL$; $RG_2 = L(RG_2).RG_1(6)$; $RG_1 = L(RG_1).0$;

$CT = CT - 1$

3) $RG_2 = L(RG_2).RG_1(6)$; $RG_1 = L(RG_1).0$; $CT = CT - 1$

Таблиця 3 – Стани регістрів під час третього способу множення

№	RG_2	RG_1	RG_3	CT	Мікрооперації
-	1110010	000000	100010	110	1
1	1110010 + 0000000 1110010 1100101	000000 + 100010 100010 000100		101	2
2	1100101 + 0000000 1100101 1001011	000100 + 100010 100110 001100		100	2
3	1001011 + 0000000 1001011 0010111	001100 + 100010 101110 011100		011	2
4	0010111 0101110	011100 111000		010	3
5	0101110 1011101	111000 110000		001	3
6	1011101 + 0000001 1011110 0111100	110000 + 100010 010010 100100		000	2

Відповідь: 0,011110010010

Обробка порядків

$P_Z = P_X + P_Y$

$P_X = 12$

$P_Y = 5$

$P_Z = 17_{10} = 10001_2$

Форма запису нормалізованого результату

Результат множення: 0,011110010010

Знак мантиси: $1 \oplus 0 = 1$

Нормалізований результат:

X = 0 10000 1 111100

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 15 = 143_{10} = 10001111_2$

X = 1 10001111 11100100100000000000000

3.4 Четвертий спосіб множення

Операнди:

X = 0 10000 1 111100

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

Множення виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо.

Операційна схема

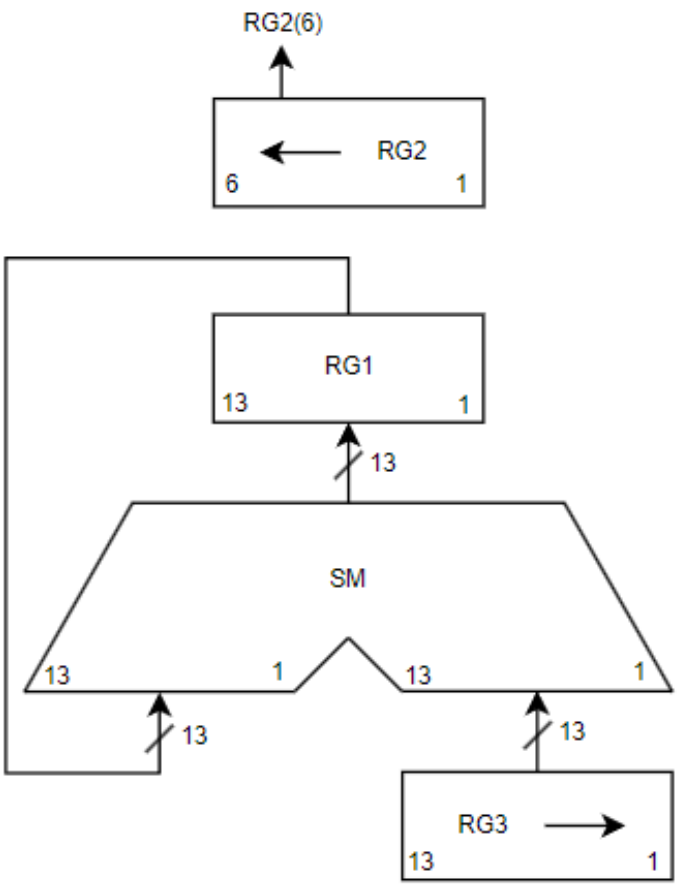


Рисунок 7 – Операційна схема четвертого способу множення

Змістовний мікроалгоритм

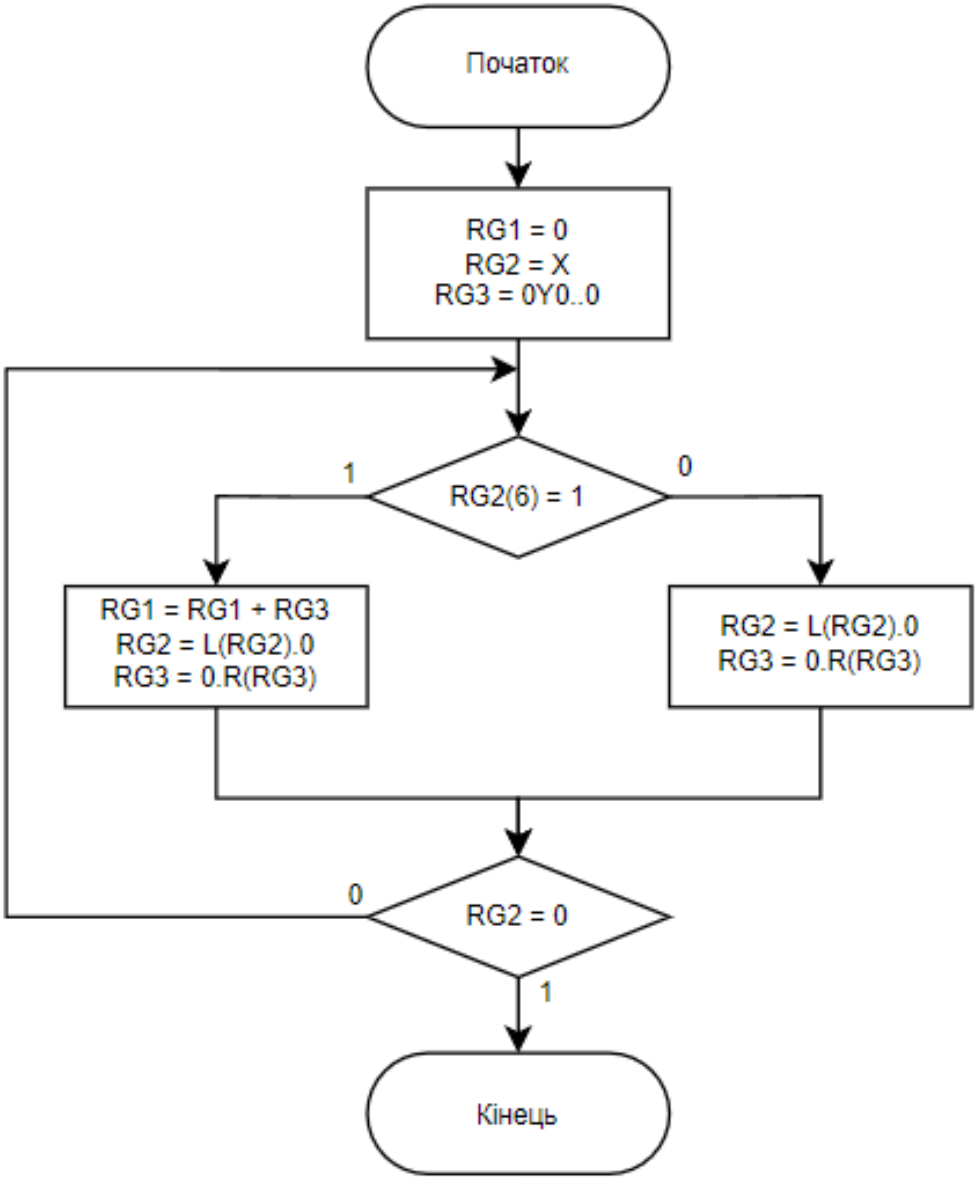


Рисунок 8 – Змістовний мікроалгоритм четвертого способу множення

Таблиця станів регістрів

$M_X * M_Y = 111100 * 100010$
1) $RG_1 = 0$; $RG_2 = X$; $RG_3 = 0Y0..0$;

2) $RG_1 = RG_1 + RG_3$; $RG_2 = L(RG_2).0$; $RG_3 = 0.R(RG_3)$

3) $RG_2 = L(RG_2).0$; $RG_3 = 0.R(RG_3)$

Таблиця 4 – Стани регістрів під час четвертого способу множення

№	RG_1	RG_2	RG_3	Мікрооперації
-	0000000000000	111100	0100010000000	1
1	0000000000000 + 0100010000000 0100010000000	111100 111000	0100010000000 0010001000000	2
2	0100010000000 + 0010001000000 0110011000000	111000 110000	0010001000000 0001000100000	2
3	0110011000000 + 0001000100000 0111011100000	110000 100000	0001000100000 0000100010000	2
4	0111011100000 + 0000100010000 0111111110000	100000 000000	0000100010000 0000010001000	2

Відповідь: 0,011111111000

Обробка порядків

$P_Z = P_X + P_Y$

$P_X = 16$

$P_Y = 5$

$P_Z = 21_{10} = 10101_2$

Форма запису нормалізованого результату

Результат множення: 0,011111111000

Знак мантиси: $1 \oplus 0 = 1$

Нормалізований результат:

X = 0 10100 1 111111

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 19 = 147_{10} = 10010011_2$

X = 1 10010011 11111110000000000000000

3.5 Перший спосіб ділення

Операнди:

X = 0 10100 1 111111

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

При реалізації ділення за першим варіантом здійснюється зсув вліво залишку при нерухомому дільнику. На рис. 13 показана можлива побудова пристрою ділення. Чергова остача формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний X). Виходи RG2 підключені до входів суматора SM безпосередньо, тобто ланцюги видачі коду з RG2 не потрібні. Дільник Y знаходиться в регістрі RG1. Результат формується в регістрі RG3 за (n + 1) циклів. Знак остачі визначається розрядом RG2[n+2]. Розряд RG3[n+1] використовується для визначення кінця операції, ознакою цього є маркерний нуль на виході розряду.

Операційна схема

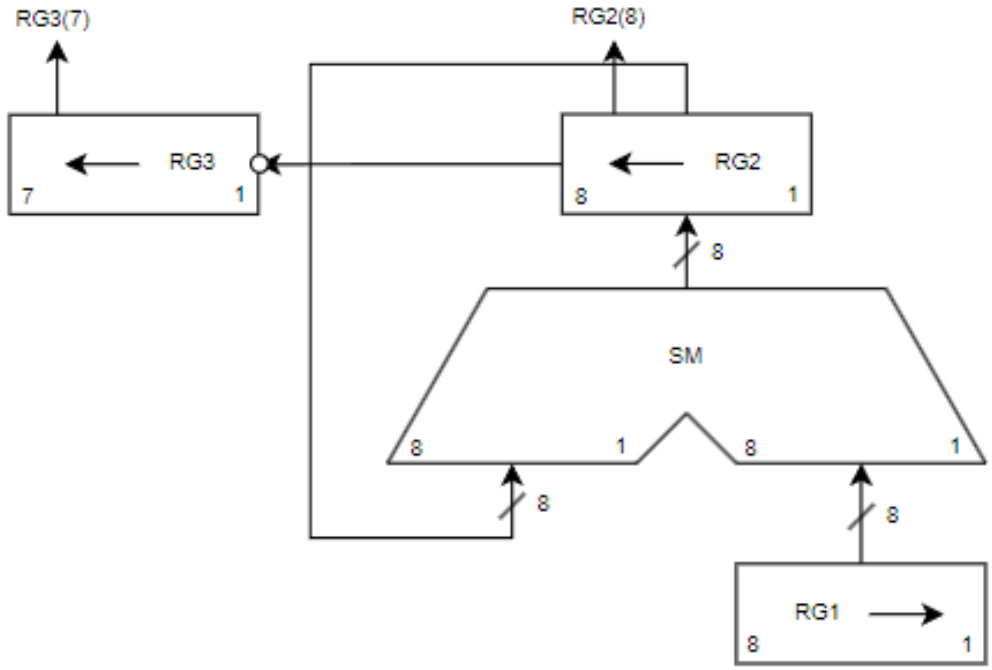


Рисунок 9 – Операційна схема першого способу ділення

Змістовний мікроалгоритм

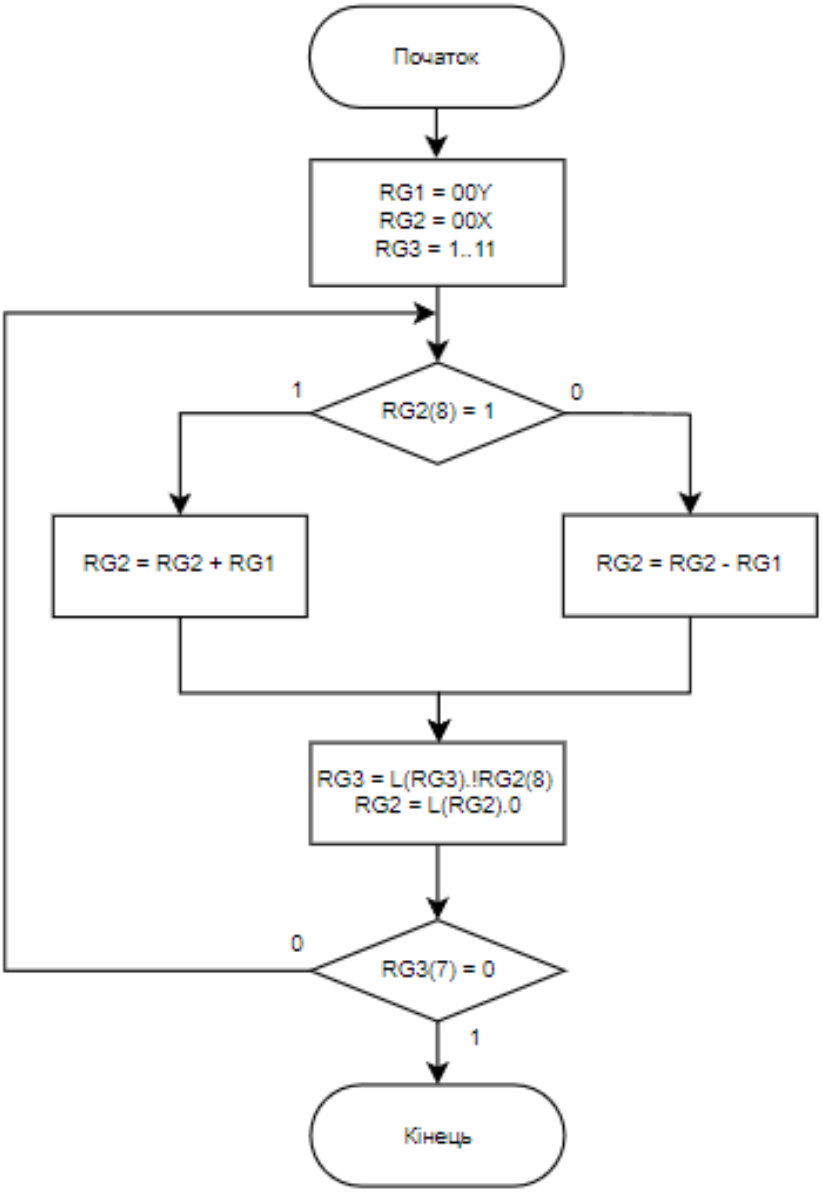


Рисунок 10 – Змістовний мікроалгоритм першого способу ділення

Мантиса діленого більша за мантису дільника, через це зсунемо її на один розряд вправо та збільшимо порядок на 1

Таблиця станів регістрів

M_X / M_Y = 011111 / 100010
1) RG_1 = 00Y; RG_2 = 00X; RG_3 = 1..11;

2) $RG_2 = RG_2 + RG_1$; $RG_2 = L(RG_2).0$; $RG_3 = L(RG_3).!RG_2(8)$

3) $RG_2 = RG_2 - RG_1$; $RG_2 = L(RG_2).0$; $RG_3 = L(RG_3).!RG_2(8)$

Таблиця 5 – Стани регістрів під час першого способу ділення

№	RG_3	RG_2	RG_1	Мікрооперації
-	1111111	00011111	SUM: 00100010 SUB: 11011110	1
1	1111111 1111110	00011111 + 11011110 11111101 11111010		3
2	1111110 1111101	11111010 + 00100010 00011100 00111000		2
3	1111101 1111011	00111000 + 11011110 00010110 00101100		3
4	1111011 1110111	00101100 + 11011110 00001010 00010100		3
5	1110111 1101110	00010100 + 11011110 11110010 11100100		3
6	1101110 1011101	11100100 + 00100010 00000110 00001100		2
7	1011101 0111010	00001100 + 11011110 11101010 11010100		3

Відповідь: 0,111010

Обробка порядків

$P_Z = P_X - P_Y$

$P_X = 21$

$P_Y = 5$

$P_Z = 16_{10} = 10000_2$

Форма запису нормалізованого результату

Результат ділення: 0,111010

Знак мантиси: $1 \oplus 0 = 1$

Нормалізація результату не потрібна

Нормалізований результат:

X = 0 10000 1 111010

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 15 = 143_{10} = 10001111_2$

X = 1 10001111 11010000000000000000000

3.6 Другий спосіб ділення

Операнди:

X = 0 10000 1 111010

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

При реалізації ділення другим способом (із зсувом дільника) збільшується розрядність регістрів RG1, RG3 і суматора SM (рис. 3.2). Цифра результату формується на виході переносу суматора SM(p). При діленні чисел з фіксованою комою повинна бути передбачена можливість фіксації переповнення розрядної сітки. Ознака переповнення формується, якщо в першому циклі цифра результату дорівнює одиниці.

Операційна схема

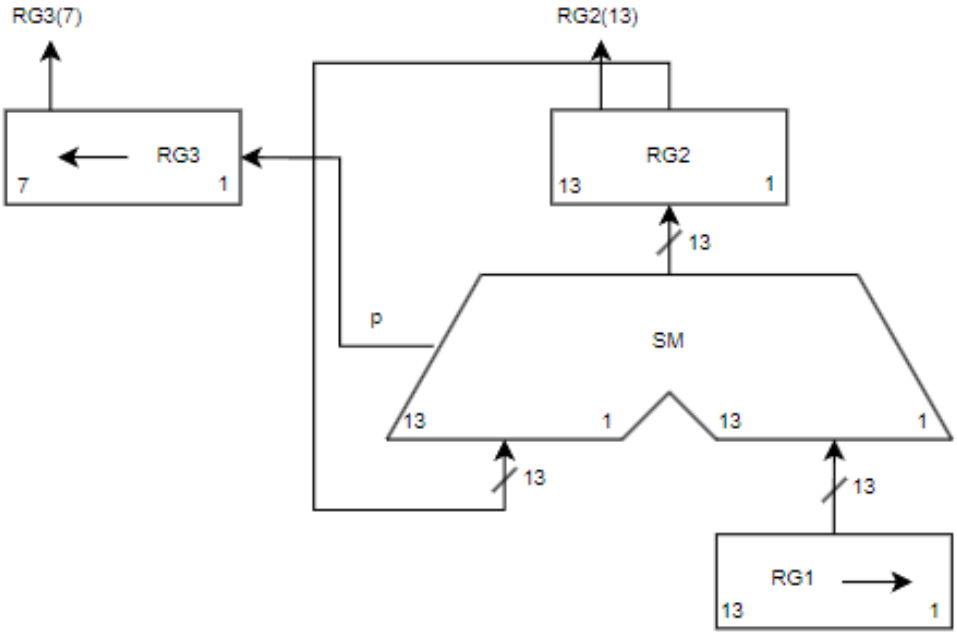


Рисунок 11 – Операційна схема другого способу ділення

Змістовний мікроалгоритм

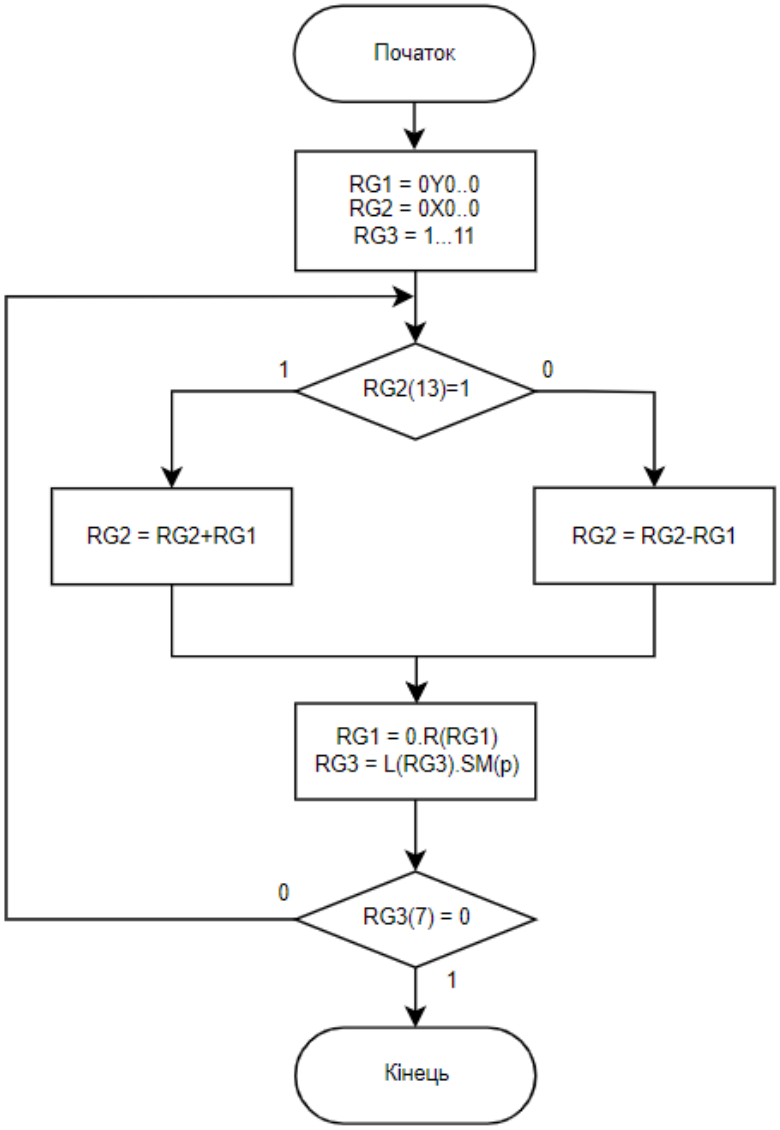


Рисунок 12 – Змістовний мікроалгоритм другого способу ділення

Мантиса діленого більша за мантису дільника, через це зсунемо її на один розряд вправо та збільшимо порядок на 1

Таблиця станів регістрів

M_X / M_Y = 011101 / 100010

1) RG_1 = 0Y0..0; RG_2 = 0X0..0; RG_3 = 1..11;

2) $RG_2 = RG_2 + RG_1$; RG_1 = 0.R(RG_1); RG_3 = L(RG_3).SM(p)

3) $RG_2 = RG_2 - RG_1$; RG_1 = 0.R(RG_1); RG_3 = L(RG_3).SM(p)

Таблиця 6 – Стани регістрів під час другого способу ділення

№	RG_3	RG_2	RG_1	Мікрооперації
-	1111111	0011101000000	SUM: 0100010000000 SUB: 1011110000000	1
1	1111111 1111110	0011101000000 + 1011110000000 1111011000000	0100010000000 SUM: 0010001000000 SUB: 1101111000000	3
2	1111110 1111101	1111011000000 + 0010001000000 0001100000000	0010001000000 SUM: 0001000100000 SUB: 1110111100000	2
3	1111101 1111011	0001100000000 + 1110111100000 0000011100000	0001000100000 SUM: 0000100010000 SUB: 1111011110000	3
4	1111011 1110110	0000011100000 + 1111011110000 1111111010000	0000100010000 SUM: 0000010001000 SUB: 1111101111000	3
5	1110110 1101101	1111111010000 + 0000010001000 0000001011000	0000010001000 SUM: 0000001000100 SUB: 1111110111100	2
6	1101101 1011011	0000001011000 + 1111110111100 0000000010100	0000001000100 SUM: 0000000100010 SUB: 1111111011110	3
7	1011011 0110110	0000000010100 + 1111111011110 1111111110010	0000000100010 SUM: 0000000010001 SUB: 1111111101111	3

Відповідь: 0,110110

Обробка порядків

$P_Z = P_X - P_Y$

$P_X = 17$

$P_Y = 5$

$P_Z = 12_{10} = 01100_2$

Форма запису нормалізованого результату

Результат ділення: 0,110110

Знак мантиси: $1 \oplus 0 = 1$

Нормалізація результату не потрібна

Нормалізований результат:

X = 0 01100 1 110110

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 11 = 139_{10} = 10001011_2$

X = 1 10001011 10110000000000000000000

3.7 Додавання чисел

Операнди:

X = 0 01100 1 110110

Y = 0 00101 0 100010

Теоретичне обґрунтування способу виконання операцій

Для додавання чисел у форматі з плавачою комою потрібно:

а. Порівняти порядки

б. Вирівняти порядки

а. Додати мантиси

б. Зробити обробку порядків

Порівняння порядків

$P_X = 12_{10} = 01101_2$

$P_Y = 5_{10} = 00101_2$

$P_X > P_Y$ ➔ $\Delta = P_X - P_Y = 7_{10} = 00111_2$

Вирівнювання порядків

Для вирівнювання порядків потрібно зсунути мантису числа Y на різницю порядків, тобто на 7.

Таблиця 7 – Зсув мантиси

M_Y	Операція	Δ
0.100010	→	00111
0.010001	→	00110
0.001000	→	00101
0.000100	→	00100
0.000010	→	00011
0.000001	→	00010
0.000000	→	00001
0.000000	→	00000

Додавання мантис

$M_X = 11.110110$

$M_Y = 00.000000$

$M_Z = M_{X(ДК)} + M_{Y(ПК)}$ (оскільки M_X від'ємне, потрібно перевести в ДК)

$M_{X(ДК)} = 11.001010$

$M_{Y(ПК)} = 00.000000$

$$\begin{array}{r} 11.001010 \\ +\ \underline{00.000000} \\ 11.001010 \end{array}$$

Оскільки результат ДК, потрібно перевести в ПК:

$M_X = 11.110110$

Обробка порядків

$P_Z = P_X$

$P_X = 12$

$P_Z = 12_{10} = 01100_2$

Форма запису нормалізованого результату

Результат додавання: 11.110110

Нормалізація результату не потрібна

Нормалізований результат:

X = 0 01100 1 110110

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 11 = 139_{10} = 10001011_2$

X = 1 10001011 10110000000000000000000

3.8 Добування кореню додатного числа

Операнд:

X = 0 01100 1 110110

Теоретичне обґрунтування способу виконання операцій

Найбільш простий алгоритм обчислення квадратного кореня з n-розрядної мантиси числа зводиться до підбору шифр результату розряд за розрядом, розпочинаючи зі старшого 2^{-1}-го розряду. За цього обчислення i-ї цифри результату X відбувається наступним чином. Після отримання чергової (i-1)-ї цифри a_{i-1} в i-й розряд A розміщується одиниця. Обчислюється різність $(B - A_i^2) = R_i$. Якщо $R_i \geq 0$, то A_i і є число, де цифри всіх розрядів збігаються з цифрами результату A. Якщо $R_i < 0$, то в i-му розряді a_i необхідно поставити нуль і переходити до обчислення (i+1)-го розряду.

Операційна схема

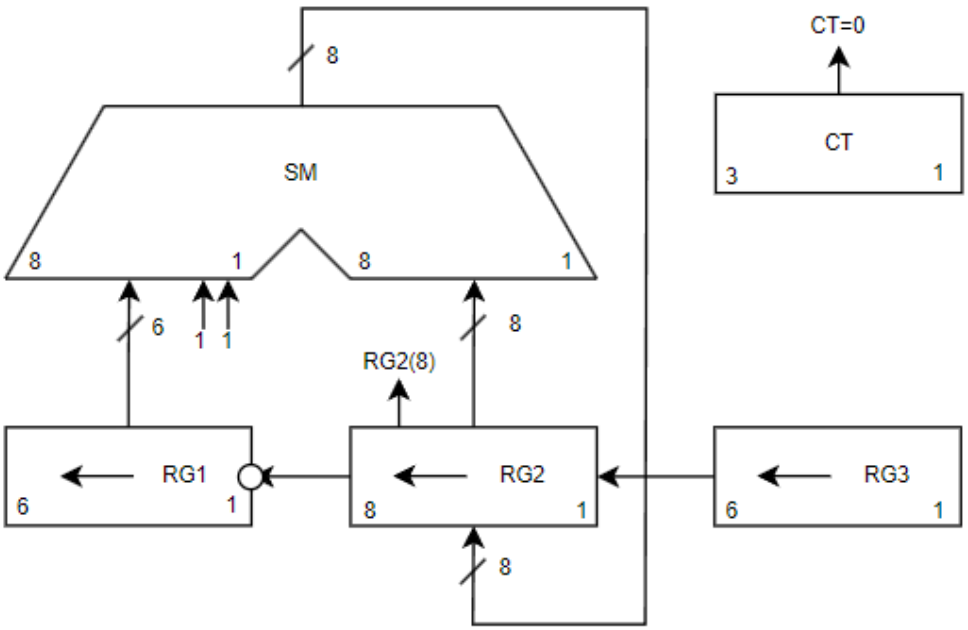


Рисунок 13 – Операційна схема добування кореня

Змістовний мікроалгоритм

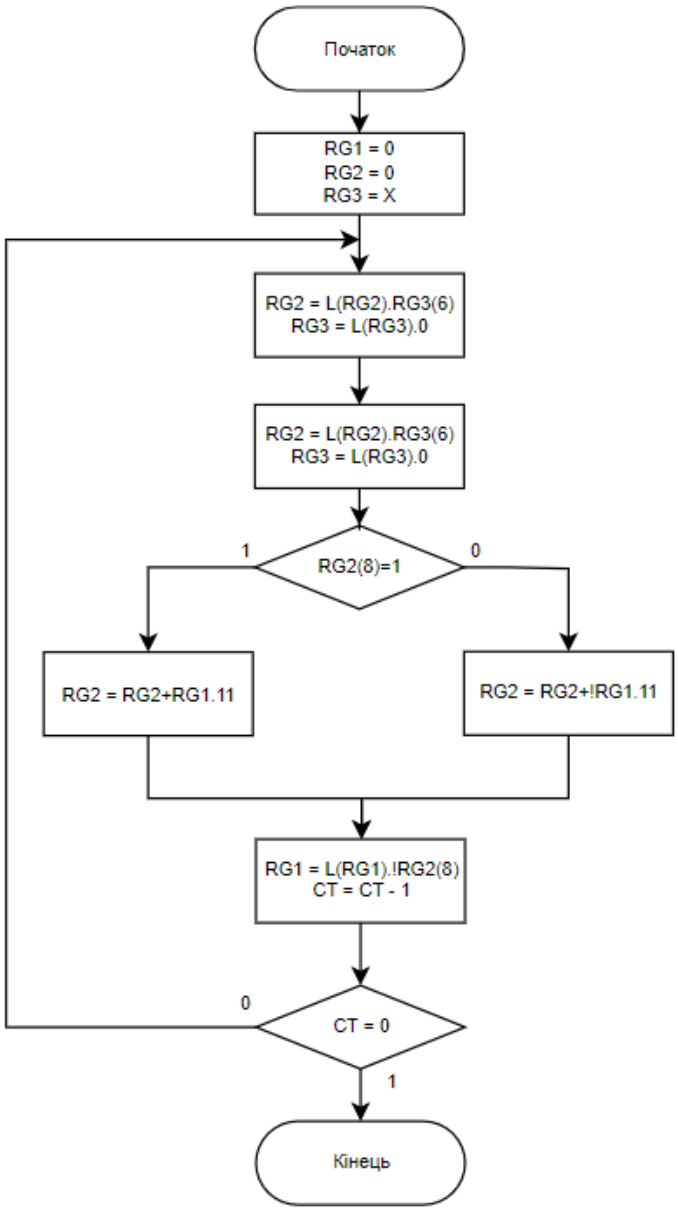


Рисунок 14 – Змістовний мікроалгоритм добування кореня

Таблиця станів регістрів

$\sqrt{|M_X|} = \sqrt{110110}$

1) $RG_1 = 0$; $RG_2 = 0$; $RG_3 = X$;

2) $RG_2 = L(RG_2).RG_3(6)$; $RG_3 = L(RG_3).0$; $RG_2 = L(RG_2).RG_3(6)$; $RG_3 = L(RG_3).0$;

$RG_2 = RG_2 + !RG_1.11$; $RG_1 = L(RG_1).!RG_2(8)$; $CT = CT - 1$

3) $RG_2 = L(RG_2).RG_3(6)$; $RG_3 = L(RG_3).0$; $RG_2 = L(RG_2).RG_3(6)$; $RG_3 = L(RG_3).0$;

$RG_2 = RG_2 + RG_1.11$; $RG_1 = L(RG_1).!RG_2(8)$; $CT = CT - 1$

Таблиця 8 – Стани регістрів під час добування кореня

№	RG_1	RG_2		RG_3	CT	Мікрооперації
		8-3	2-1			
-	000000	000000	00	110110	110	1
1	000000 000001	000000 000000 000000 + 111111 000000	00 01 11 11 10	110110 101100 011000	101	2
2	000001 000011	000000 000001 000010 + 111110 000001	10 00 01 11 00	011000 110000 100000	100	2
3	000011 000111	000001 000010 000100 + 111100 000001	00 01 10 11 01	100000 000000 000000	011	2
4	000111 001110	000001 000010 000101 + 111000 111101	01 10 00 11 11	000000 000000 000000	010	2
5	001110 011101	111101 111011 110111 + 001110 000101	11 10 00 11 11	000000 000000 000000	001	3

6	011101 111010	000101 001011 010111 + 100010 111001	11 10 00 11 11	000000 000000 000000	000	2

Відповідь: 0,111010

Обробка порядків

$P_Z = P_X / 2$

$P_X = 12$

$P_Z = 6_{10} = 00110_2$

Форма запису нормалізованого результату

Результат добування кореню: 111010

Знак мантиси: 0

Нормалізація результату не потрібна

Нормалізований результат:

X = 0 00110 0 111010

Короткий формат за стандартом ANSI/IEEE 754-2008

Представлення чисел за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі:

$E_X = 128 + (P_X - 1) = 128 + 5 = 133_{10} = 10000101_2$

X = 0 10000101 11010000000000000000000

4. ПОБУДОВА ФУНКЦІОНАЛЬНИХ СХЕМ

4.1 Функціональна схема

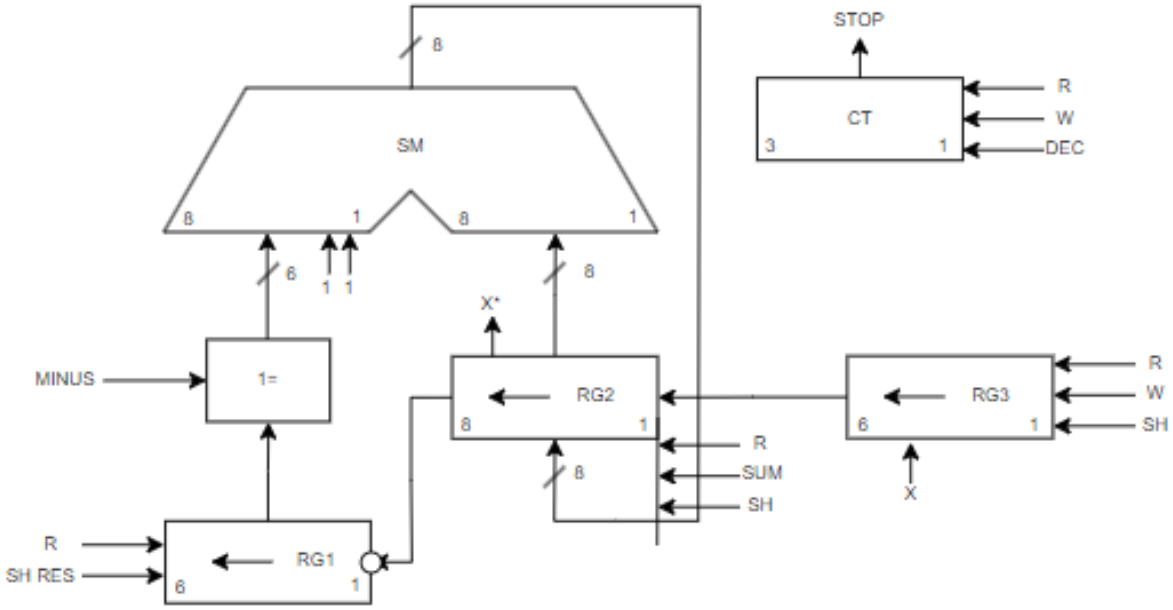


Рисунок 15 – Функціональна схема добування кореня

/

4.2 Структурний мікроалгоритм

Оцінивши функціональну схему та змістовний мікроалгоритм, та включаючи те, що якщо йдуть два поспіль сигнали, які виконуються в одному блоці, додати між ними пустий блок, можна побудувати наступний мікроалгоритм:

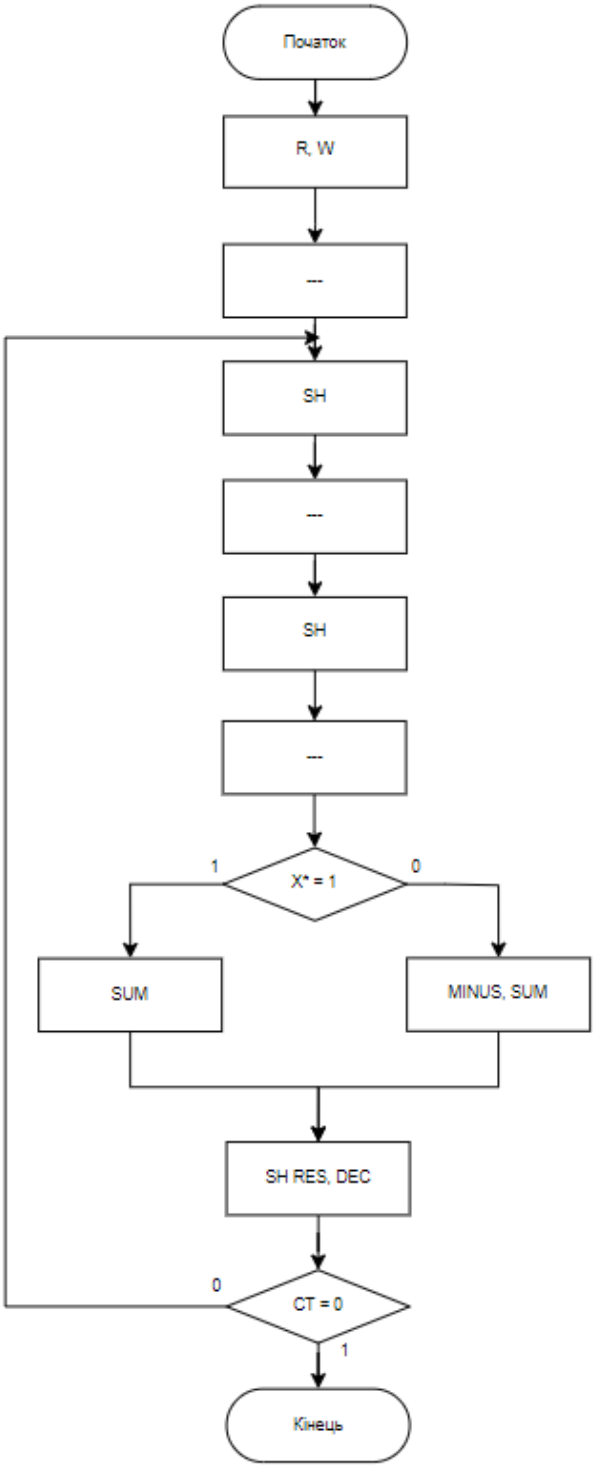


Рисунок 16 – Структурний мікроалгоритм для добування кореня

4.3 Автомат Мілі

Розмітка станів

Стан в автоматах Мілі позначається початок кожного наступного блоку після операційного (рис. 17).

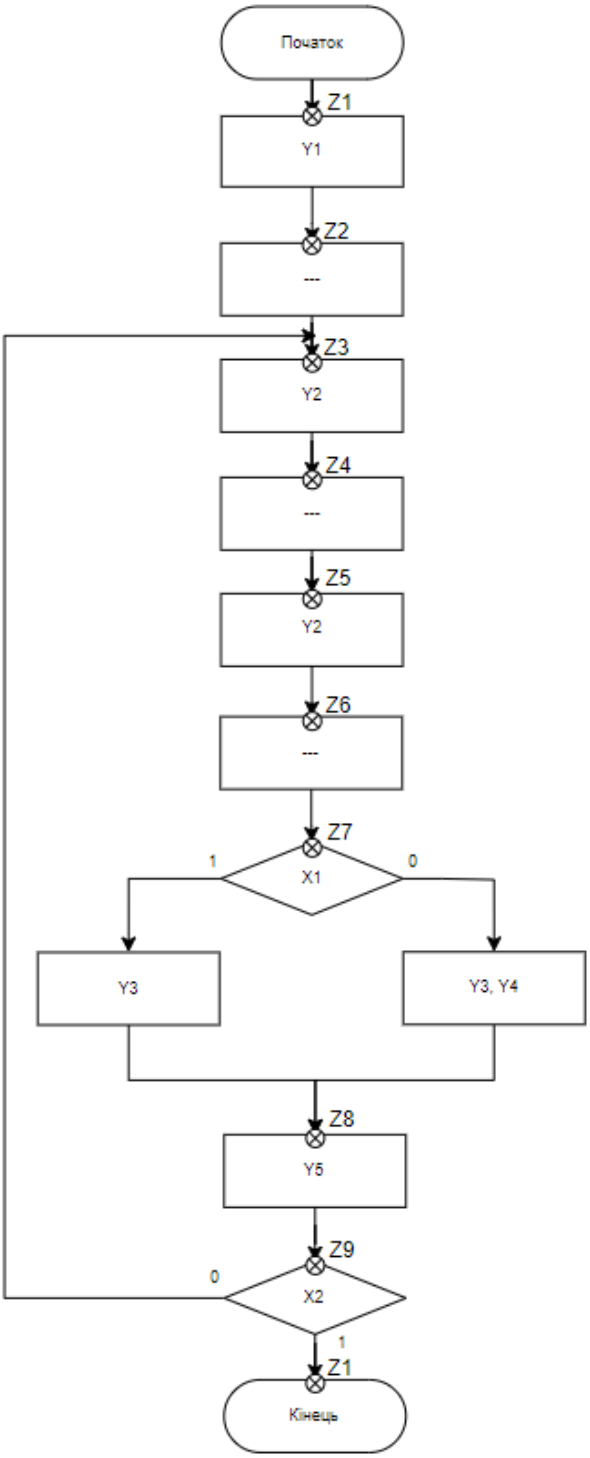


Рисунок 17 – Закодований мікроалгоритм для добування кореня

Граф

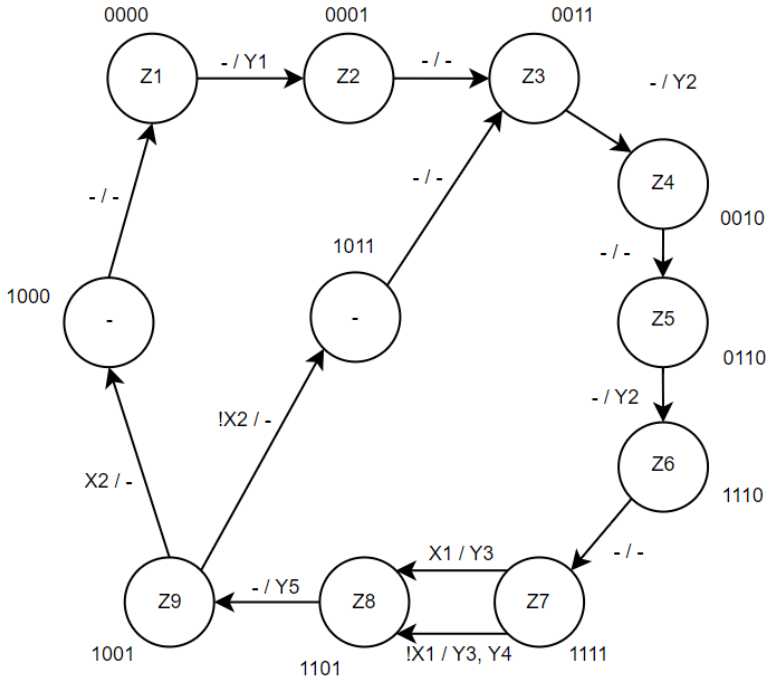


Рисунок 18 – Граф автомата Мілі

Структурна таблиця

Таблиця 9 – Структурна таблиця

Q_1	Q_2	Q_3	Q_{4t}	Q_1	Q_2	Q_3	Q_{4t+1}	X_1	X_2	Y_1	Y_2	Y_3	Y_4	Y_5	D_1	D_2	D_3	D_4
0	0	0	0	0	0	0	1	-	-	1	0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	1	-	-	0	0	0	0	0	0	0	1	1
0	0	1	1	0	0	1	0	-	-	0	1	0	0	0	0	0	1	0
0	0	1	0	0	1	1	0	-	-	0	0	0	0	0	0	1	1	0
0	1	1	0	1	1	1	0	-	-	0	1	0	0	0	1	1	1	0
1	1	1	0	1	1	1	1	-	-	0	0	0	0	0	1	1	1	1
1	1	1	1	1	1	0	1	1	-	0	0	1	0	0	1	1	0	1
1	1	1	1	1	1	0	1	0	-	0	0	1	1	0	1	1	0	1
1	1	0	1	1	0	0	1	-	-	0	0	0	0	1	1	0	0	1
1	0	0	1	1	0	1	1	-	0	0	0	0	0	0	1	0	1	1
1	0	0	1	1	0	0	0	-	1	0	0	0	0	0	1	0	0	0
1	0	1	1	0	0	1	1	-	-	0	0	0	0	0	0	0	1	1
1	0	0	0	0	0	0	0	-	-	0	0	0	0	0	0	0	0	0

Діаграми Вейча

Q_3 Q_3

Q_1

Q_4 Q_2 Q_4

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

X_2 X_2 D_1

X_1 X_1

Q_3 Q_3

Q_1

Q_4 Q_2 Q_4

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

D_2 X_2

X_1 X_1

Рисунок 19 – Діаграми Вейча для функцій D1, D2

Q_3 Q_3

Q_1

Q_4 Q_2 X_2

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

D_3

Q_4 X_2

X_1 X_1

Q_3 Q_3

Q_1

Q_4 Q_2 X_2

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

D_4

Q_4 X_2

X_1 X_1

Рисунок 20 – Діаграми Вейча для функцій D3, D4

Одержання МДНФ

Запишімо МДНФ (І/АБО):

$D_1 = Q_2 \vee Q_1\overline{Q}_3Q_4$

$D_2 = Q_3\overline{Q}_4 \vee Q_1Q_2Q_3$

$D_3 = \overline{Q}_2Q_3Q_4 \vee Q_3\overline{Q}_4 \vee \overline{Q}_2Q_4\overline{X}_2 \vee \overline{Q}_1\overline{Q}_2Q_4$

$D_4 = \overline{Q}_1\overline{Q}_3 \vee Q_1Q_3 \vee Q_2Q_4 \vee Q_1\overline{Q}_2Q_4\overline{X}_2$

$Y_1 = \overline{Q}_1\overline{Q}_2\overline{Q}_3\overline{Q}_4$

$Y_2 = \overline{Q}_1\overline{Q}_2Q_3Q_4 \vee \overline{Q}_1Q_2Q_3\overline{Q}_4$

$Y_3 = Q_1Q_2Q_3Q_4$

$Y_4 = Q_1Q_2Q_3Q_4\overline{X}_1$

$Y_5 = Q_1Q_2\overline{Q}_3Q_4$

Операторні форми

Запис в формі І/АБО:

$D_1 = Q_2 \vee Q_1\overline{Q}_3Q_4$

$D_2 = Q_3\overline{Q}_4 \vee Q_1Q_2Q_3$

$D_3 = \overline{Q}_2Q_3Q_4 \vee Q_3\overline{Q}_4 \vee \overline{Q}_2Q_4\overline{X}_2 \vee \overline{Q}_1\overline{Q}_2Q_4$

$D_4 = \overline{Q}_1\overline{Q}_3 \vee Q_1Q_3 \vee Q_2Q_4 \vee Q_1\overline{Q}_2Q_4\overline{X}_2$

$Y_1 = \overline{Q}_1\overline{Q}_2\overline{Q}_3\overline{Q}_4$

$Y_2 = \overline{Q}_1\overline{Q}_2Q_3Q_4 \vee \overline{Q}_1Q_2Q_3\overline{Q}_4$

$Y_3 = Q_1Q_2Q_3Q_4$

$Y_4 = Q_1Q_2Q_3Q_4\overline{X}_1$

$Y_5 = Q_1Q_2\overline{Q}_3Q_4$

AFDK та часова діаграма

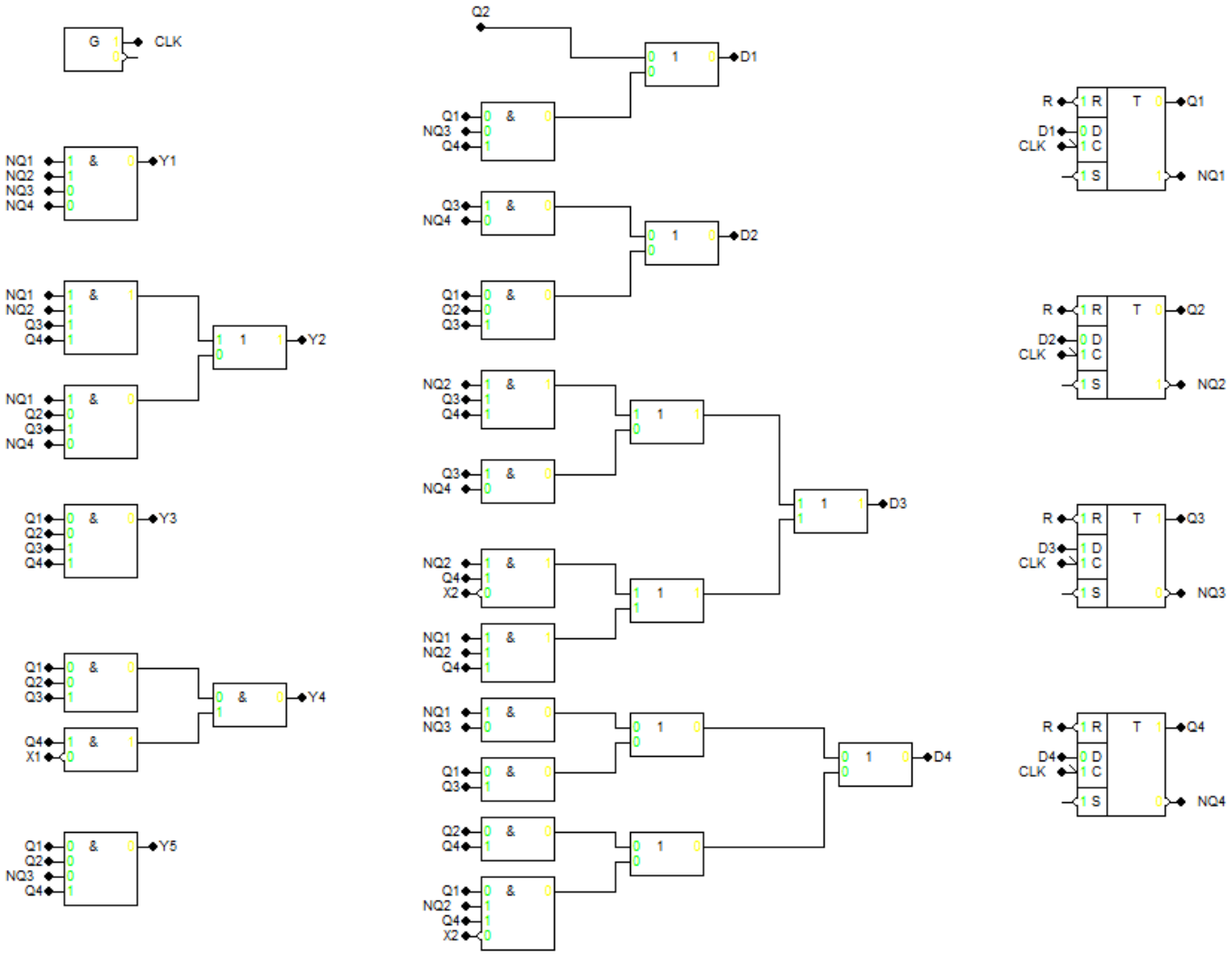


Рисунок 21 – Керуючий автомат

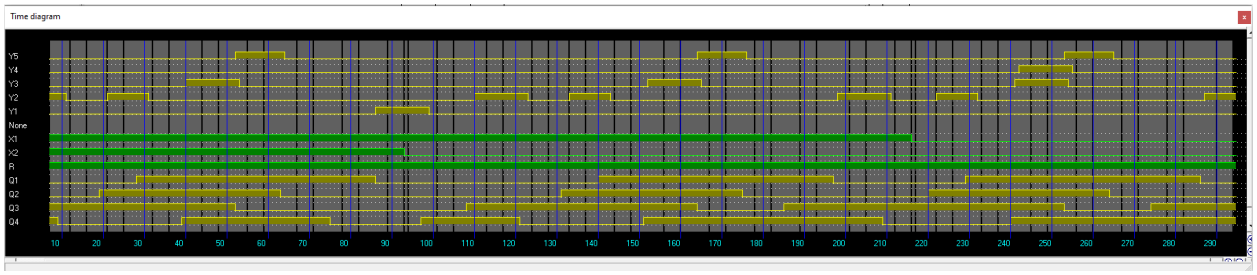


Рисунок 22 – Часова діаграма керуючого автомату

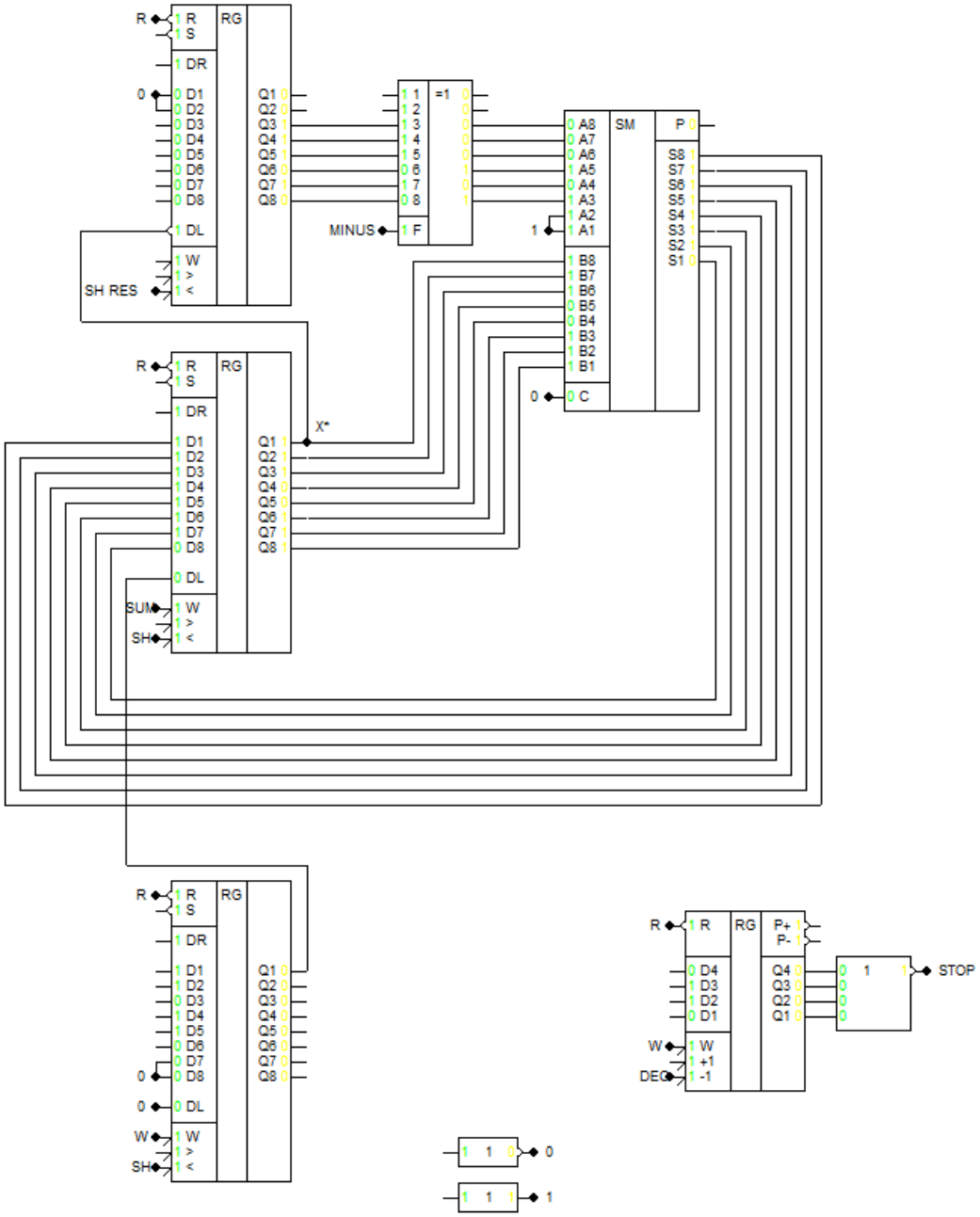


Рисунок 23 – Пристрій обчислення добування кореня

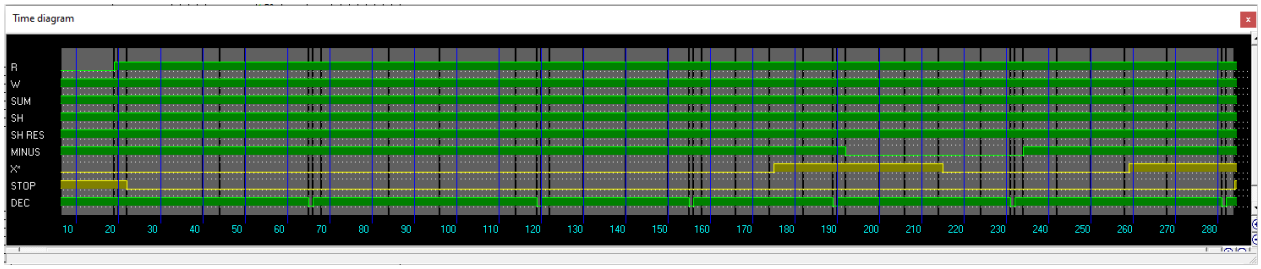


Рисунок 24 – Часова діаграма обчислювального пристрою

5. ВИСНОВОК

Завдання курсової роботи полягало у виконанні абстрактного і структурного синтезу автомата одного з способів арифметичних операцій. Функціональна схема автомата приведена у документі «Керуючий автомат. Схема електрична функціональна» і виконана згідно з вимогами єдиної системи конструкторської документації. Автомат працює по заданому алгоритму і може бути використаний у сфері обчислювальної техніки. Під час виконання роботи були закріплені мої знання теоретичного курсу, отримані навички їх практичного застосування. Також були покращені навички роботи з конструкторською документацією та інструментарієм.

6. СПИСОК ЛІТЕРАТУРИ

1. Жабін В.І., Верба О.А. Комп'ютерна логіка. Курсова робота. Навчальний посібник. https://ela.kpi.ua/handle/123456789/50134

2. Жабін В.І., Жуков І.А., Клименко І.А., Ткаченко В.В. Прикладна теорія цифрових автоматів: Навч. посібник.– К.: Вид-во НАУ, 2009.– 364 с.

3. Жабін В.І., Ткаченко В.В. Цифрові автомати. Практикум. – К.: ВЕК+, 2004.– 160 с.

4. ДСТУ 3008-2015 «Державний стандарт України. Документація. Звіти в сфері науки і техніки. Структура і правила оформлення».

5. Матвієнко М.П. Комп'ютерна логіка. Підручник. Вид. 2-ге перероб. та доп. – Київ: Видавництво Ліра – К, 2017. – 324 с.

6. ДСТ 2.104-2006. ЄСКД. Основні написи (Міждержавний. В Україні - ДСТУ 2.104-2006).

7. ДСТУ ГОСТ 2.702:2013 ЄСКД. Правила виконання електричних схем (ГОСТ 2.702-2011, IDT).

8. ДСТУ ISO 5457:2006 (ISO 5457:1999, IDT) Національний стандарт України. Документація технічна на вироби. Креслеників. Розміри та формати.