

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"
ФАКУЛЬТЕТ ІНФОРМАТИКИ І ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра обчислювальної техніки

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА

з дисципліни ” Комп’ютерна логіка 2. Комп’ютерна арифметика ”

Виконав

Лисенко Дмитро Вадимович

Факультет ІОТ,

Група ІО-61,

Залікова книжка № 6116

Керівник _____

(підпис керівника)

Київ - 2017р.

Вибір варіанту.

Перевести номер залікової книжки в двійкову систему. Записати два 10-розрядних двійкових числа:

$$X = -x_7x_61x_5x_40,x_31x_2x_1 \quad \text{і} \quad Y = +x_91x_8x_7x_6x_5,x_4x_3x_2x_1,$$

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

$$6116_{10} = 1011111100100_2$$

$$x_{10} = 1; x_9 = 1; x_8 = 1; x_7 = 1; x_6 = 1; x_5 = 0; x_4 = 0; x_3 = 1; x_2 = 0; x_1 = 0.$$

$$X_2 = -111000,1100$$

$$Y_2 = +111110,0100$$

Завдання.

1. Числа X_2 і Y_2 в прямому коді записати у формі з плаваючою комою у класичному варіанті (з незміщеним порядком і повною мантисою). На порядок відвести 4 розряди, на мантису 7 розрядів (з урахуванням знакових розрядів). Записати числа X і Y також за стандартом ANSI/IEEE 754-2008 в короткому 32-розрядному форматі).

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). Нумери 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).

Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу;

2.2 операційну схему;

2.3 змістовний (функціональний) мікроалгоритм;

- 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.

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

3 Для операції з номером $x_3x_2x_1$ додатково виконати:

3.1 побудувати функціональну схему з відображенням управляючих сигналів, входів для запису операндів при ініціалізації пристрою і схем формування внутрішніх логічних умов;

3.2 розробити закодований (структурний) мікроалгоритм (мікрооперації замінюються управляючими сигналами виду W,SL,SR тощо);

3.3 для операції з парним двійковим номером $x_3x_2x_1$ додатково подати граф управляючого автомата Мура з кодами вершин, а для непарного номера $x_3x_2x_1$ – автомата Мілі;

3.4 побудувати управляючий автомат на тригерах та елементах булевого базису. Вибрати JK - тригери для автомата Мура та RS - тригери для автомата Мілі.

Завдання №1

$$X_{ПК} = 1.111000,1100$$

$$Y_{ПК} = 0.111110,0100$$

Представлення чисел у формі з плаваючою точкою з порядком і округленою мантисою:

$$P_x = +110_2; \quad M_x = -0,111001_2;$$

$$P_y = +110_2; \quad M_y = +0,111110_2;$$

$X_2 :$

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

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

$Y_2 :$

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

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

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

$$E_x = P_x + (2^{m-1} - 1) = P_x + (2^7 - 1) = 110_2 + 1111111_2 = 10000101_2$$

$$E_y = P_y + (2^{m-1} - 1) = P_y + (2^7 - 1) = 110_2 + 1111111_2 = 10000101_2$$

X_2 :

1	0	0	0	0	1	0	1	1	1	1	0	0	0	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Y_2 :

1	0	0	0	0	1	0	1	0	1	1	1	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Завдання №2

2.1 Перший спосіб множення.

2.1.1 Теоретичне обґрунтування першого способу множення:

Під час множення *першим способом* в першому такті i -го циклу аналізується значення $RG2[1]$ – молодшого (n -го) розряду регістру $RG2$, в якому знаходиться чергова цифра множника. Вміст $RG3$ додається до суми часткових добутоків, що знаходяться в регістрі $RG1$, якщо $RG2[1]=1$, або не додається, якщо $RG2[1]=0$. В другому такті здійснюється правий зсув у регістрах $RG1$ і $RG2$, що еквівалентно множенню їхнього вмісту на 2^{-1} . При зсуві цифра молодшого розряду регістру $RG1$ записується у вивільнюваний старший розряд регістру $RG2$. Після виконання n циклів молодші розряди $2n$ -розрядного добутку будуть записані в регістр $RG2$, а старші – у $RG1$.

2.1.2 Операційна схема:

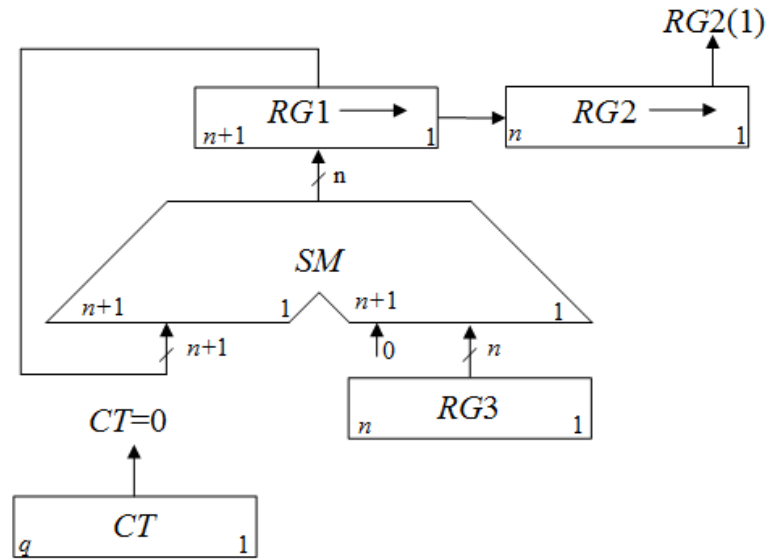


Рисунок 2.1.1 Операційна схема пристрою для множення першим способом

2.1.3 Змістовний мікроалгоритм:

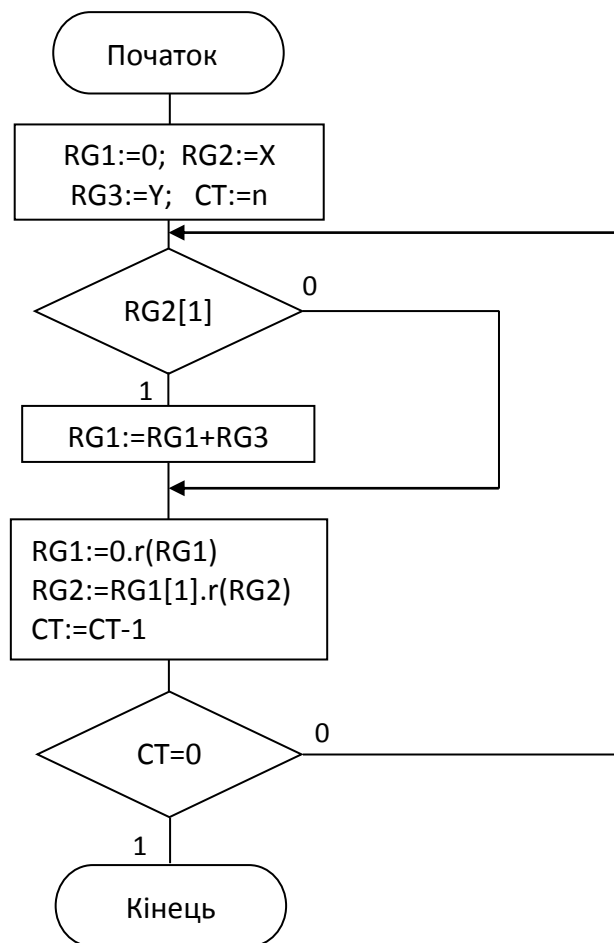


Рисунок 2.1.2 Змістовний мікроалгоритм виконання операції множення першим способом.

2.1.4 Таблиця станів реєстрів:

Таблиця 2.1.1. Таблиця станів реєстрів пристрою множення першим способом

№	RG1→	RG2→	RG3	СТ
П.С.	0000000	111001	111110	110
1	+ 0011111 = 0011111	011100	111110	101
2	0011111	001110	111110	100
3	0011111	000111	111110	011
4	+ 0111110 = 1000101 0100010	000011	111110	010
5	+ 0111110 = 1100000 0110000	100001	111110	001
6	+ 0111110 = 1101110 0110111	010000	111110	000 кінець

2.1.5 Обробка порядків і нормалізація:

$$P_z = P_x + P_y = 110_2 + 110_2 = 1100_2.$$

Отримали результат: 0,110111010000

Округлена мантиса: $M_z = 0,110111$; $P_z = 1100$.

Знак мантиси: $1 \oplus 0 = 1$.

2.1.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	1 1 0 0	1	1 1 1 0 1 1 1

2.2 Другий спосіб множення

2.2.1 Теоретичне обґрунтування другого способу множення:

Перед початком множення *другим способом* множник X записують в реєстр $RG2$, а множене Y – в молодші розряди реєстру $RG3$ (тобто в реєстрі $RG3$ установлюють $Y_0 = Y2^{-n}$). В кожному i -му циклі множення додаванням

кодів $RG3$ і $RG1$ управляє цифра $RG2[1]$, а в регістрі $RG3$ здійснюється зсув вліво на один розряд, в результаті чого формується величина $Y_i = 2Y_{i-1}$. Оскільки сума часткових добутків в процесі множення нерухома, зсув в регістрі $RG3$ можна виконати суміщення в часі з підсумовуванням (як правило, $t_n \geq t_3$). В цьому випадку $t_m = nt_n$. Завершення операції множення визначається за нульовим вмістом регістру $RG2$, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

2.2.2 Операційна схема

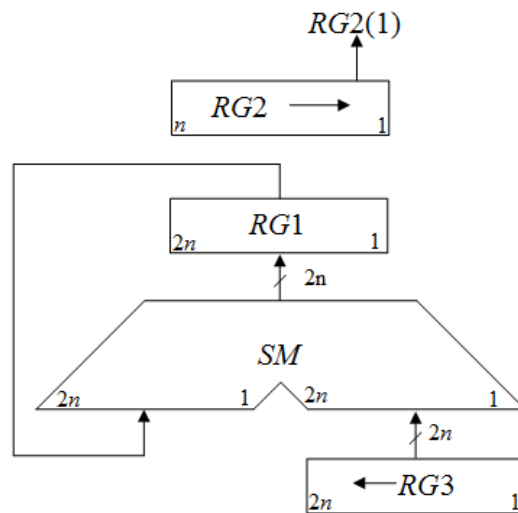


Рисунок 2.2.1. Операційна схема пристрою множення другим способом

2.2.3 Змістовний мікроалгоритм

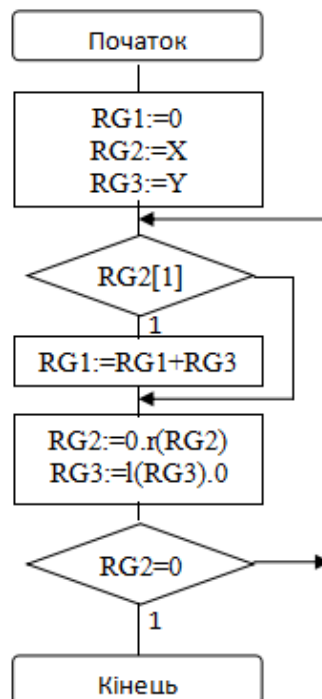


Рисунок 2.2.2. Змістовний мікроалгоритм пристрою множення другим способом

2.2.4 Таблиця станів регістрів:

В ЕОМ при роботі із дробовими числами часто потрібно обчислювати не $2n$, а тільки $(n+1)$ цифр добутку й округляти його до n розрядів. В цьому випадку при реалізації другого способу можна зменшити довжину SM і $RG1$, а при реалізації четвертого – зменшити довжину SM , $RG1$ і $RG3$. Для того щоб похибка від відкидання молодших розрядів не перевищила половини ваги n -го розряду результату, в перерахованих вузлах досить мати тільки по l додаткових молодших розрядів, де l вибирається з умови

$$l \geq 1 + \log_2(n - l - 1).$$

При $n=6$ мінімальне l , яке задовольняє дану умову дорівнює $l=2$. Отже розрядність $RG1$ та суматора замінюємо на $n+l=6+2=8$.

Таблиця 2.2.1. Таблиця станів регістрів пристрою множення другим способом

№	RG1	RG2→	RG3←
П.С.	00000000	110111	000000111110
1	+ 00000011 = 00000011	011011	000001111100
2	+ 00000111 = 00001010	001101	000011111000
3	+ 00001111 = 00011001	000110	000111110000
4	00011001	000011	001111100000
5	+ 00111110 = 01010111	000001	011111000000
6	+ 01111100 = 11010011 ($n+1$)	000000 кінець	111110000000

2.2.5 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 1100_2 + 110_2 = 10010_2.$$

Отримали результат: 0,1101001

Округлена мантиса: $M_z = 0,110101$; $P_z = 10010$.

Знак мантиси: $1 \oplus 0 = 1$.

2.2.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	1 0 0 1 0	1	1 1 0 1 0 1

2.3 Третій спосіб множення.

2.3.1 Теоретичне обґрунтування третього способу множення:

При множенні *третім способом* множник X записується в старші розряди $RG2$, при цьому $RG2[1]=0$. Вага молодшого розряду $RG3$ дорівнює 2^{-2n} , тому код в регістрі $RG3$ являє собою значення $Y2^{-n}$. В кожному циклі множення підсумування виконується при $RG2[n+1]=1$. В регістрах $RG1$ і $RG2$ виконується лівий зсув. В результаті підсумовування вмісту $RG3$ і $RG1$ може виникнути перенос в молодший розряд регістру $RG2$, що реалізується на SM . Збільшення довжини $RG2$ на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розряди добутку будуть знаходитися в регістрі $RG1$, а старші – в регістрі $RG2$. Час множення третім способом визначається аналогічно першому способу.

2.3.2 Операційна схема

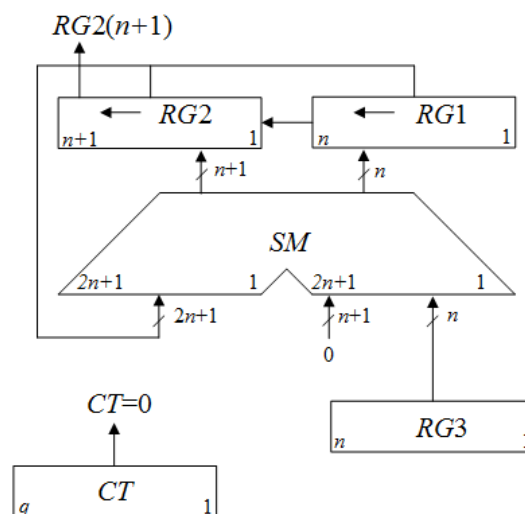


Рисунок 2.3.1. Операційна схема пристрою множення третім способом

2.3.3 Змістовний мікроалгоритм

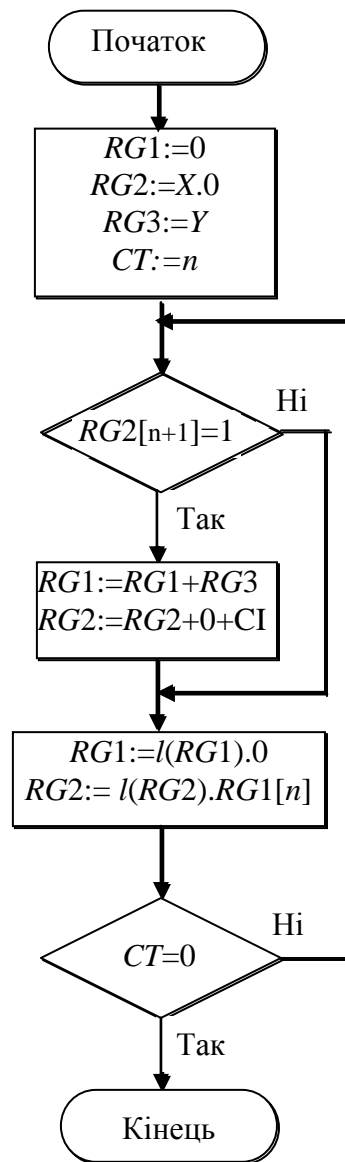


Рисунок 2.3.2. Змістовний мікроалгоритм пристрою множення третім способом

2.3.4 Таблиця станів регістрів

Таблиця 2.3.1. Таблиця станів регістрів пристрою множення третім способом

№ ц.	RG2←	RG1←	RG3	CT
П.С.	1101010	000000	111110	110
1	+ 0000000 = 1101010 1010101	+ 111110 = 111110 111100	111110	101
2	+ 0000001 = 1010110 0101101	+ 111110 = 111010 110100	111110	100
3	1011011	101000	111110	011

4	$ \begin{array}{r} + \\ 0000001 \\ = \\ 1011100 \\ 0111001 \\ \hline 1110010 \end{array} $	$ \begin{array}{r} + \\ 111110 \\ = \\ 100110 \\ 001100 \\ \hline 111110 \end{array} $	111110	010
5	1110010	011000	111110	001
6	$ \begin{array}{r} + \\ 0000001 \\ = \\ 1110011 \\ 1100110 \\ \hline 1100110 \end{array} $	$ \begin{array}{r} + \\ 111110 \\ = \\ 010110 \\ 101100 \\ \hline 1100110 \end{array} $	111110	000 кінець

2.3.5 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 10010_2 + 110_2 = 11000_2.$$

Отримали результат: 0,110011010110

Округлена мантиса: $M_z = 0,110011$; $P_z = 11000$.

Знак мантиси: $1 \oplus 0 = 1$.

2.3.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	11000	1	110011

2.4 Четвертий спосіб множення.

2.4.1 Теоретичне обґрунтування четвертого способу множення:

Перед множенням *четвертим способом* множник записують в регістр $RG2$, а множене – в старші розряди регістру $RG3$ (тобто в $RG3$ установлюють $Y_0=Y2^{-1}$). В кожному циклі цифра $RG2[n]$, що знаходиться в старшому розряді регістру $RG2$, управляє підсумовуванням, а в $RG3$ здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2^{-1} . Час виконання множення четвертим способом складає $t_m=nt_{\Pi}$, визначається аналогічно другому способу.

2.4.2 Операційна схема

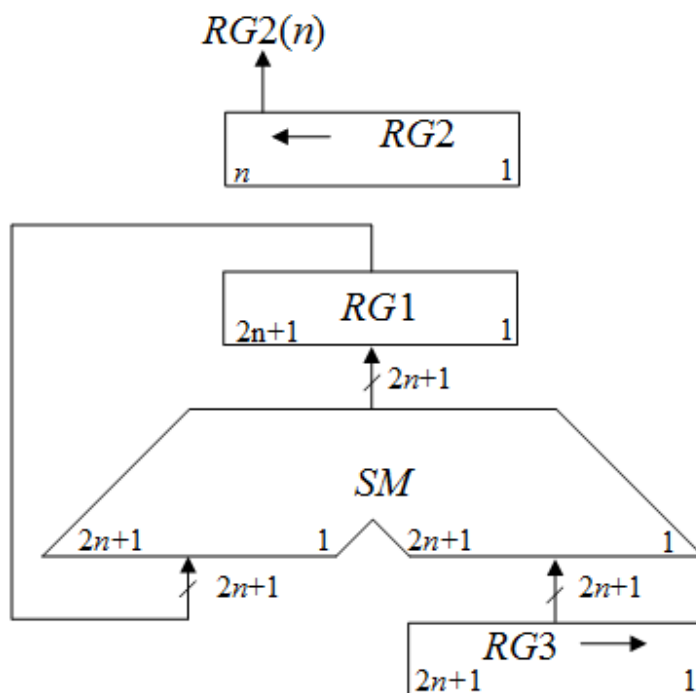


Рисунок 2.4.1. Операційна схема пристрою множення четвертим способом

2.4.3 Змістовний мікроалгоритм

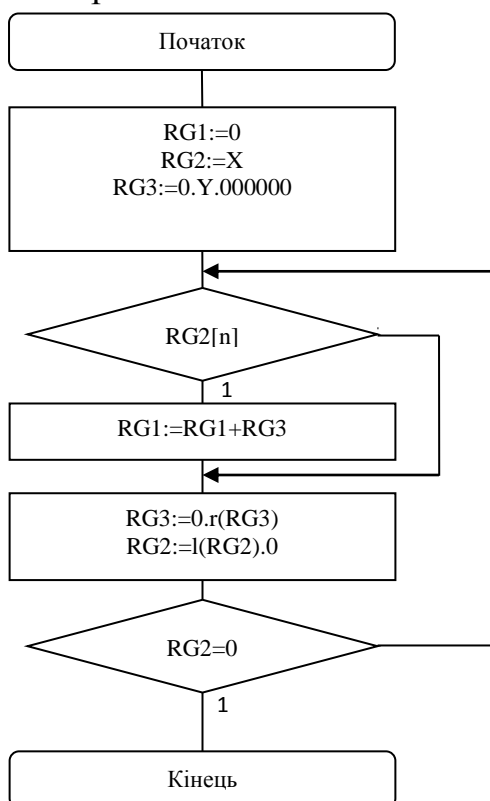


Рисунок 2.4.2. Змістовний мікроалгоритм пристрою множення четвертим способом

2.4.4 Таблиця станів регістрів

Замінюємо розрядність RG1, RG3 та SM на $n+l+1 = 6+2+1=9$

Таблиця 2.4.1. Таблиця станів регістрів пристрою множення четвертим способом

№ п.	RG1	RG2←	RG3→
П.С.	000000000	110011	011111000
1	$ \begin{array}{r} + \\ 011111000 \\ = \\ 011111000 \end{array} $	100110	001111100
2	$ \begin{array}{r} + \\ 001111100 \\ = \\ 101110100 \end{array} $	001100	000111110
3	101110100	011000	000011111
4	101110100	110000	000001111
5	$ \begin{array}{r} + \\ 000001111 \\ = \\ 110000011 \end{array} $	100000	000000111
6	$ \begin{array}{r} + \\ 000000111 \\ = \\ \mathbf{110001010} \\ (n+1) \end{array} $	000000 кінець	000000011

2.4.5 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 11000_2 + 110_2 = 11110_2.$$

Отримали результат: 0,1100010

Округлена мантиса: $M_z = 0,110001$; $P_z = 11110$.

Знак мантиси: $1 \oplus 0 = 1$.

2.4.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	1 1 1 1 0	1	1 1 1 0 0 0 1

2.5 Перший спосіб ділення

2.5.1 Теоретичне обґрунтування способу

При реалізації ділення за *першим* варіантом здійснюється зсув вліво залишку при нерухомому дільнику. 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. Максимальний час одержання цифри результату визначається виразом $t_{\Pi} = t_{\text{Д}} + t_3$, де $t_{\text{Д}}$ – тривалість виконання мікрооперації додавання/віднімання; t_3 – тривалість виконання мікрооперації зсуву. Час для одержання $n+1$ цифри частки визначається виразом $t = (n+1) t_{\Pi}$.

2.5.2 Операційна схема

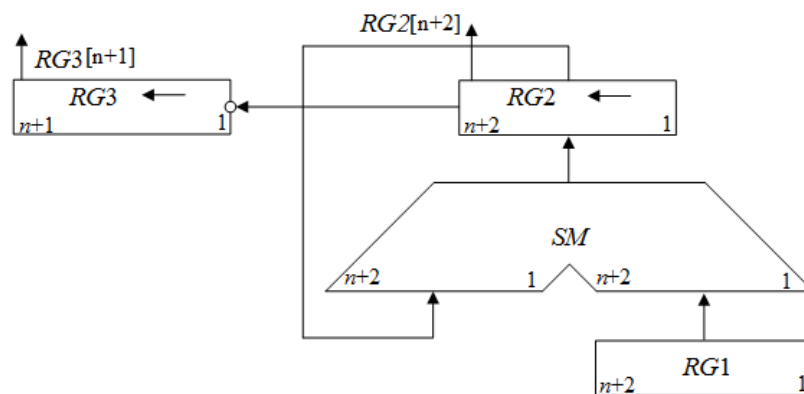


Рисунок 2.5.1. Операційна схема пристрою ділення першим способом

2.5.3 Змістовний мікроалгоритм

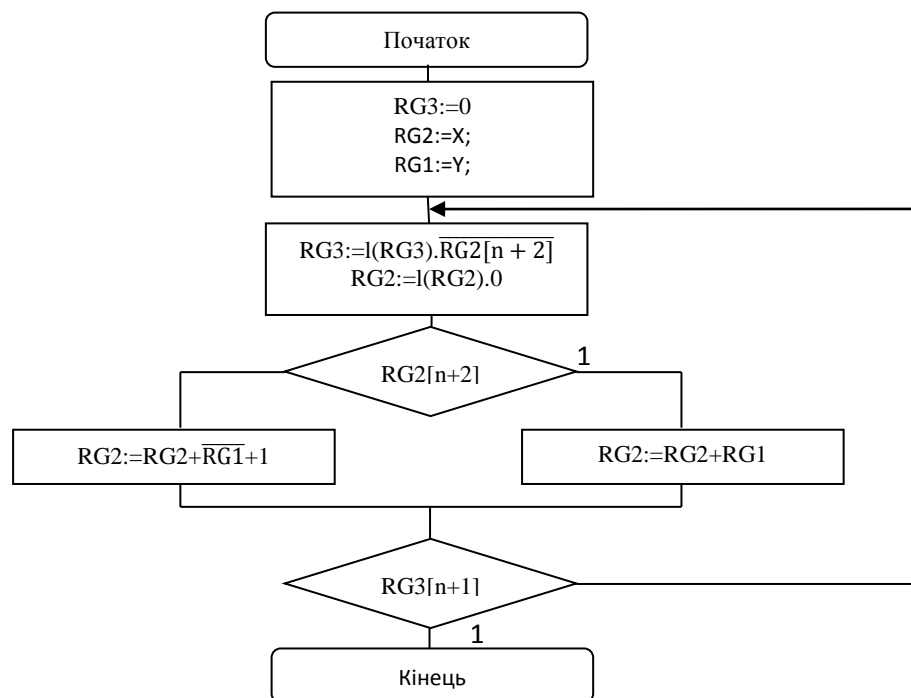


Рисунок 2.5.2. Змістовний мікроалгоритм пристрою ділення першим способом

2.5.4 Таблиця станів регістрів

Таблиця 2.5.1. Таблиця станів регістрів пристрою ділення першим способом

№ циклу	RG3←	RG2←	RG1
П.С.	0000000	00110001	00111110
1	0000001*	01100010 + 11000010 = 00100100	00111110
2	000001*1	01001000 + 11000010 = 00001010	00111110
3	00001*11	00010100 + 11000010 = 11010110	00111110
4	0001*110	10101100 + 00111110 = 11101010	00111110
5	001*1100	11010100 + 00111110 = 00010010	00111110
6	01*11001	00100100 + 11000010 = 11100110	00111110
7	1*110010 кінець	11001100 + 00111110 = 00001010	00111110

2.5.5 Обробка порядків і нормалізація

$$P_z = P_x - P_y = 11110_2 - 110_2 = 11000_2.$$

Мантиса: $M_z = 0,110010$; $P_z = 11000$.

Знак мантиси: $1 \oplus 0 = 1$.

2.5.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	110000	1	110010

2.6. Другий спосіб ділення.

2.6.1 Теоретичне обґрунтування другого способу ділення:

При реалізації ділення другим способом (із зсувом дільника) збільшується розрядність регістрів $RG2$, $RG3$ і суматора SM (рис. 3.2). В даному випадку процеси додавання/віднімання і зсуву можуть бути суміщені у часі. Отже, для ділення за другим способом час одержання цифри результату дорівнює $t_{Ц} = t_{Д}$. Цифра результату формується на виході переносу суматора $SM(p)$. Загальний час ділення визначається як $t = (n + 1)t_{Д}$.

2.6.2 Операційна схема

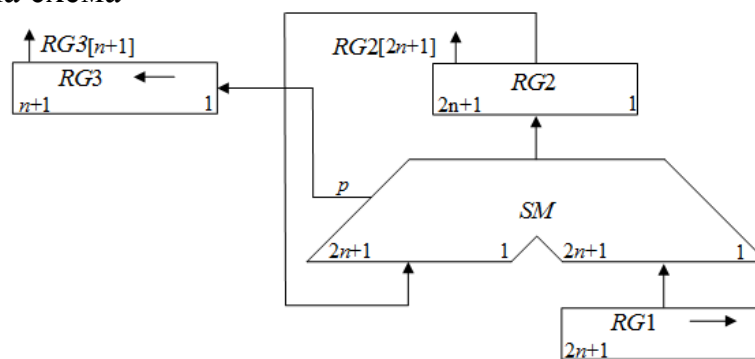


Рисунок 2.6.1. Операційна схема пристрою ділення другим способом

2.6.3 Змістовний мікроалгоритм

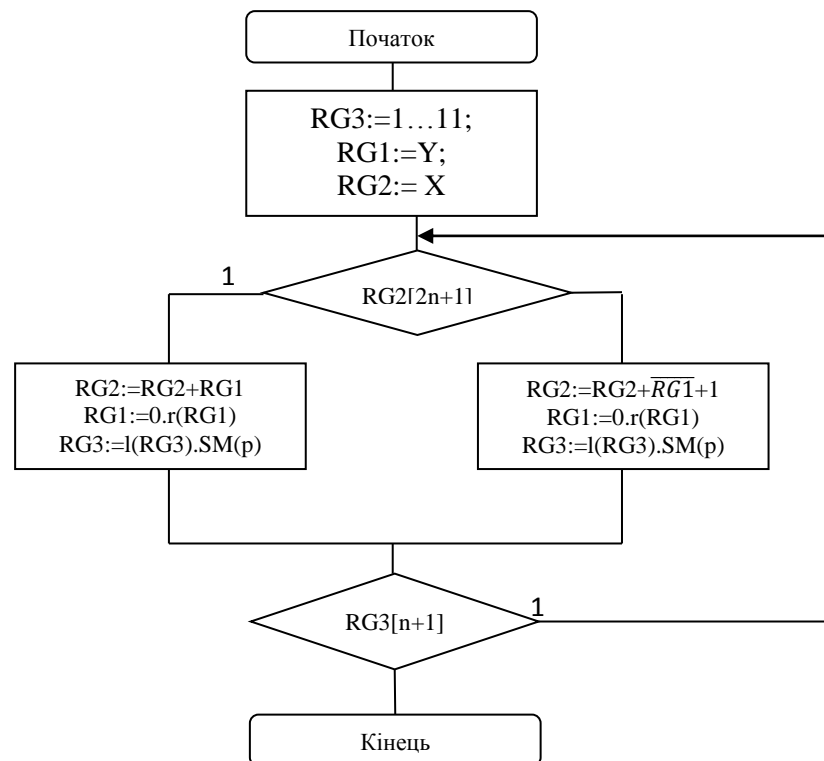


Рисунок 2.6.2. Змістовний мікроалгоритм пристрою ділення другим способом

2.6.4 Таблиця станів регістрів

Таблиця 2.6.1. Таблиця станів регістрів пристрою ділення другим способом

№ ц. П.С.	RG3←	RG2	RG1→
	1111111	0110010000000	0111110000000
1		+ 1000010000000 = 1110100000000	
	1111110*		0011111000000
2		+ 0011111000000 = 0010011000000	
	111110*1		0001111100000
3		+ 1110000100000 = 0000011100000	
	11110*11		0000111110000
4		+ 1111000010000 = 1111011110000	
	1110*110		0000011111000
5		+ 0000011111000 = 1111111101000	
	110*1100		0000001111100
6		+ 0000001111100 = 0000001100100	
	10*11001		0000000111110
7		+ 1111111000010 = 0000000100110	
	0*110011 кінець		0000000011111

2.6.5 Обробка порядків і нормалізація

$$P_z = P_x - P_y = 11000_2 - 110_2 = 10010_2.$$

Мантиса: $M_z = 0,110011$; $P_z = 10010$.

Знак мантиси: $1 \oplus 0 = 1$.

2.6.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z	P_z	Зн. M_z	M_z
0	1 0 0 1 0	1	1 1 0 0 1 1

2.7 Операція додавання чисел

2.7.1 Теоретичне обґрунтування способу

В пам'яті числа зберігаються у ПК.

На першому етапі додавання чисел з плаваючою комою виконують вирівнювання порядків до числа із старшим порядком.

На другому етапі виконують додавання мантий. Додавання мантий виконується у *доповняльних кодах*.

Додавання виконується порозрядно на n -розрядному суматорі з переносом. Останній етап – нормалізація результату. Виконується за допомогою зсуву мантий результату і коригування порядку результату. Порушення нормалізації можливо вліво і вправо, на 1 розряд вліво і на n розрядів вправо.

2.7.2 Операційна схема

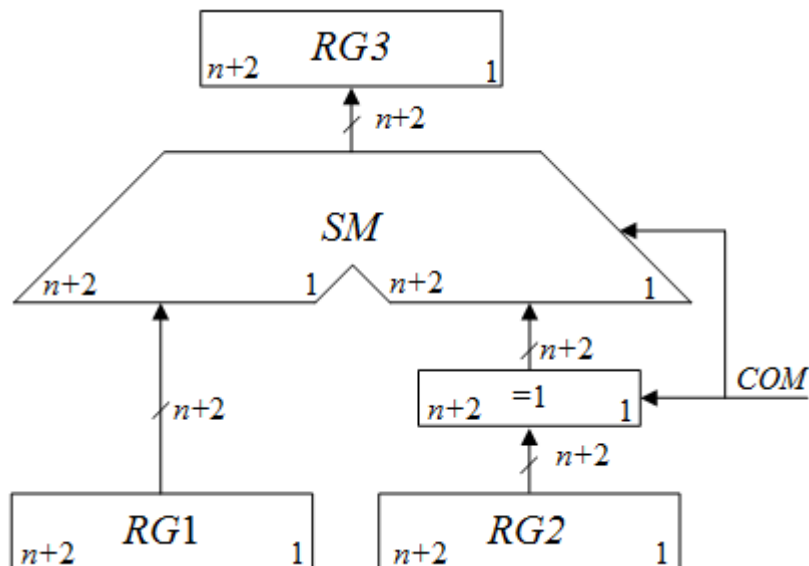


Рисунок 2.7.1. Операційна схема пристрою додавання/віднімання у МДК

2.7.3 Змістовний мікроалгоритм

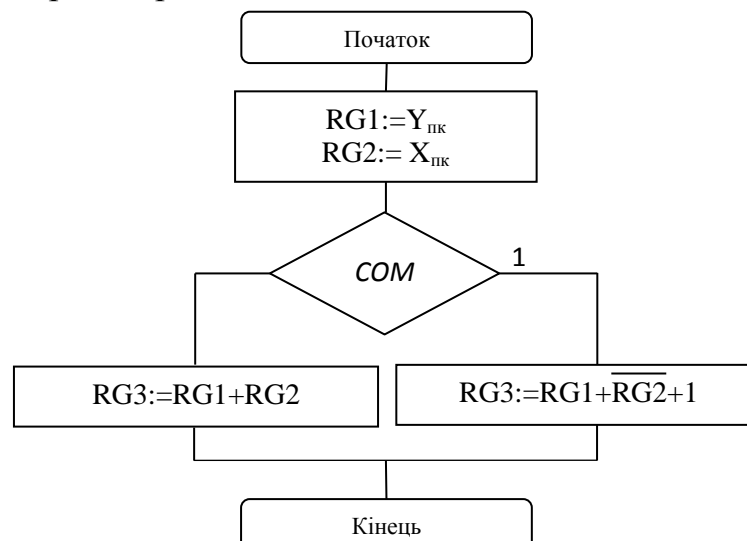


Рисунок 2.7.2. Змістовний мікроалгоритм пристрою додавання/віднімання у МДК

2.7.4 Таблиця станів регістрів

Вирівнювання порядків:

$$P_x > P_y \rightarrow P_z = P_x = 10010_2,$$

$$P_x - P_y = 10010_2 - 110_2 = 1100_2 = 12_{10}.$$

Мантиси з вирівняними порядками:

$$M_x = 0,110011$$

$$M_y = 0,0000000000000111110 \approx 0,000000$$

Числа у модифікованому ДК:

$$X_{\text{МДК}} = 11.001101$$

$$Y_{\text{МДК}} = 00.000000$$

Таблиця 2.7.1. Таблиця станів регістрів пристрою додавання/віднімання у МДК двох чисел

RG1	RG2	RG3
11.001101	00.000000	11.001101 + 00.000000 = 11.001101

2.7.5 Обробка порядків

$$P_z = 10010_2$$

Мантиса: $M_z = 0,110011$; $P_z = 10010$.

Знак мантиси: 1.

2.7.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z

P_z

Зн. M_z

M_z

0.	1	0	0	1	0
----	---	---	---	---	---

1.	1	1	0	0	1	1
----	---	---	---	---	---	---

2.8 Операція добування кореня додатного числа

2.8.1 Теоретичне обґрунтування способу

Аргумент вводиться зі старших розрядів. Порядок результату дорівнює поділеному на два порядку аргумента. З мантиси добувається корінь завдяки нерівностям:

$$Z_i \leq \sqrt{X} \leq Z_i + 2^{-i};$$

$$Z_i^2 \leq X \leq Z_i^2 + 2^{-i}Z_i + 2^{-2i};$$

$$0 \leq 2^{i-1}(X - Z_i^2) \leq Z_i + 2^{-i-1}.$$

Виконання операції зводиться до послідовності дій:

1. Одержання остачі.

$$R_{i+1}' = 2R_i - Z_i - 2^{-i-2};$$

2. Якщо $R_{i+1}' \geq 0$, то $Z_{i+1} = 1$, $R_{i+1} = R_{i+1}'$.

3. Якщо $R_{i+1}' < 0$, то $Z_{i+1} = 0$, $R_{i+1} = R_{i+1}' + Z_i - 2^{-i-2}$.

Відновлення остачі додає зайвий такт, але можна зробити інакше:

$R_{i+2} = 2R_{i+1}' + Z_i + 2^{-i-2} + 2^{-i-3}$, тоді корінь добувається без відновлення залишку.

Для цього R_i зсувається на 2 розряди ліворуч, а Z_i - на 1 розряд ліворуч, і формується як при діленні.

2.8.2 Операційна схема

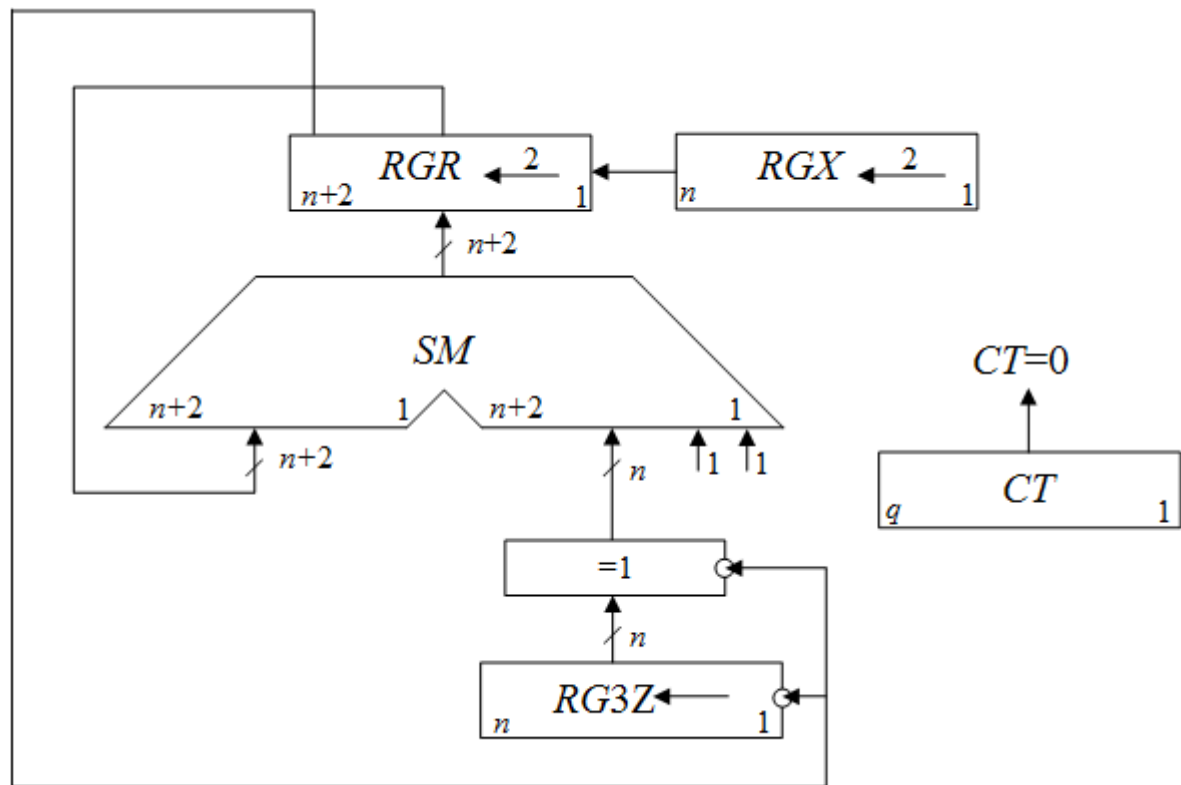


Рисунок 2.8.1. Операційна схема пристрою знаходження кореня

2.8.3 Змістовний мікроалгоритм

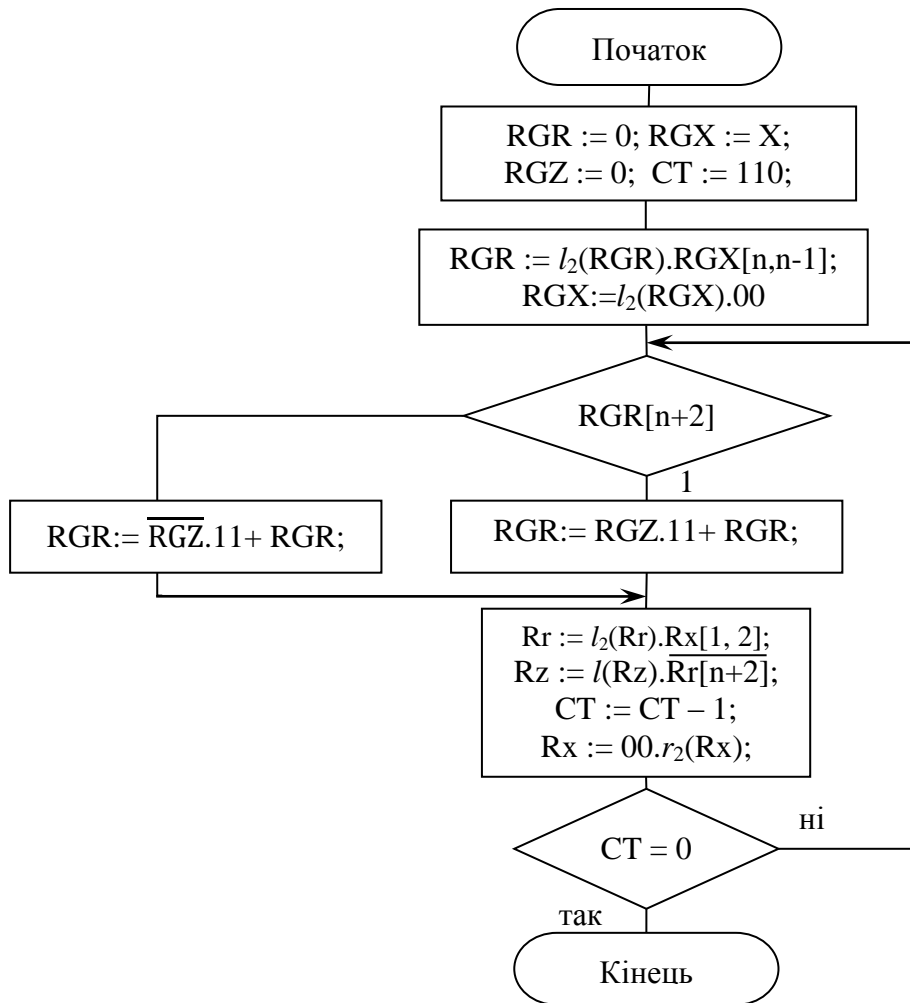


Рисунок 2.8.2 Змістовний мікроалгоритм пристрою знаходження кореня

2.8.4 Таблиця станів регістрів

Таблиця 2.8.1. Таблиця станів регістрів пристрою добування кореня

№ ц.	RGZ←	RGR←	RGX←	CT
ПС	000000	00000000 00000011	110011 001100	110
1	000001	+ 11111111 = 00000010 00001000	110000	101
2	000011	+ 11111011 = 00000011 00001111	000000	100
3	000111	+ 11110011 = 00000010 00001000	000000	011

4	001110	$\begin{array}{r} + \\ 11100011 \\ = \\ 11101011 \\ 10101100 \end{array}$	000000	010
5	011100	$\begin{array}{r} + \\ 00111011 \\ = \\ 11100111 \\ 10011100 \end{array}$	000000	001
6	111001	$\begin{array}{r} + \\ 01110011 \\ = \\ 00001111 \\ 00111100 \end{array}$	000000	000 кінець

2.8.5 Обробка порядків

$$P_z = \frac{P_x}{2} = \frac{10010_2}{10_2} = 1001_2$$

Мантиса: $M_z = 0,111001$; $P_z = 1001$.

Шукали корінь з модуля, тоді знак мантиси: 0.

2.8.6 Форма запису нормалізованого результату з плаваючою комою в пам'ять

Зн. P_z

P_z

0.	1	0	0	1
----	---	---	---	---

Зн. M_z

M_z

0.	1	1	1	0	0	1
----	---	---	---	---	---	---

3. Управляючий автомат Мура на тригерах

$x_3x_2x_1=100$ – операція ділення першим способом.

3.1 Функціональна схема з відображенням управляючих сигналів

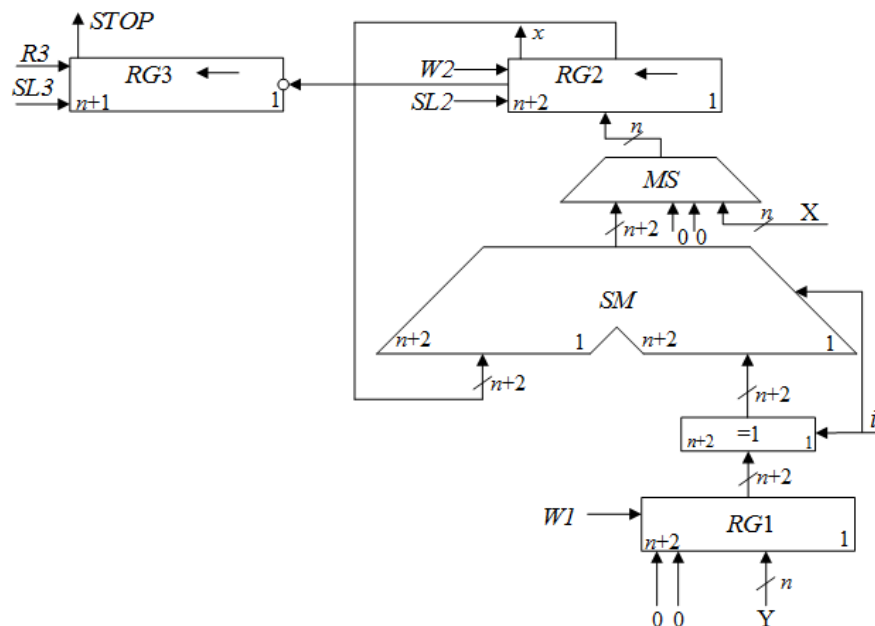


Рисунок 3.1. Функціональна схема пристрою ділення другим способом

3.2 Закодований мікроалгоритм

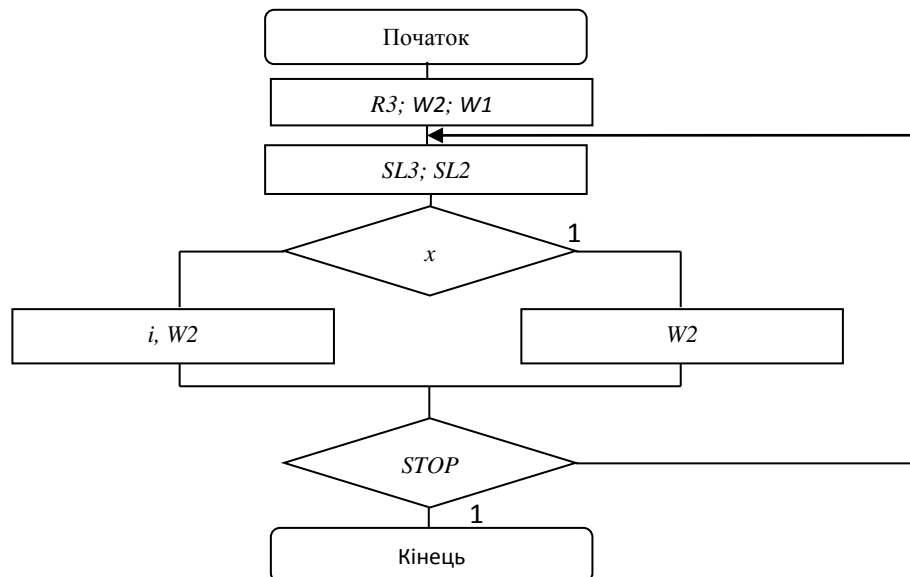


Рисунок 3.2. Змістовний мікроалгоритм пристрою ділення першим способом

За закодованим мікроалгоритмом складемо таблицю:

Таблиця 3.1 Таблиця кодування сигналів

Сигнали операційного автомата	Сигнали управляючого автомата
$R3, W1$	$Y1$
$SL3, SL2$	$Y2$
$W2$	$Y3$
i	$Y4$
x	$X1$
$STOP$	$X2$

Закодований мікроалгоритм з управляючими сигналами автомата

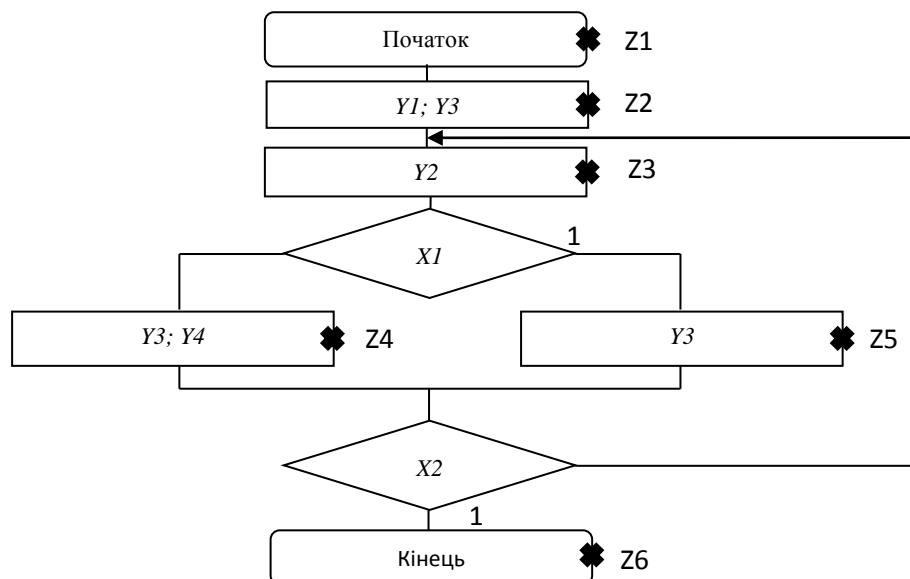


Рисунок 3.3. Змістовний мікроалгоритм пристрою ділення першим способом з управляючими сигналами автомата

3.3 Граф управляющего автомата Мура

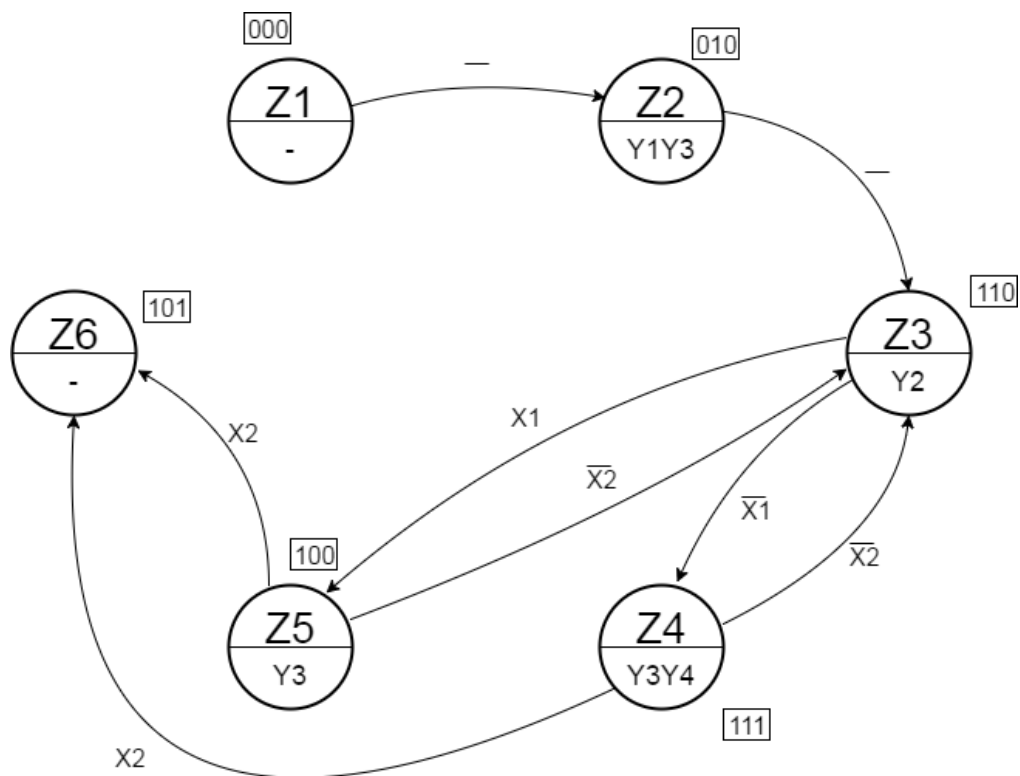


Рисунок 3.4 Граф автомата Мура

3.4 Побудова автомата

Таблиця 3.2. Структурна таблиця автомата

Перехід	$Q_3Q_2Q_1$	$Q_3Q_2Q_1$	x_1x_2	$y_4y_3y_2y_1$	J_3K_3	J_2K_2	J_1K_1
$z_1 \rightarrow z_2$	000	010	--	0000	0-	1-	0-
$z_2 \rightarrow z_3$	010	110	--	0101	1-	-0	0-
$z_3 \rightarrow z_4$	110	111	0-	0010	-0	-0	1-
$z_3 \rightarrow z_5$	110	100	1-	0010	-0	-1	0-
$z_4 \rightarrow z_3$	111	110	-0	1100	-0	-0	-1
$z_4 \rightarrow z_6$	111	101	-1	1100	-0	-1	-0
$z_5 \rightarrow z_3$	100	110	-0	0100	-0	1-	0-
$z_5 \rightarrow z_6$	100	101	-1	0100	-0	0-	1-

JK:
 $0 \xrightarrow{0-} 0$
 $0 \xrightarrow{1-} 1$
 $1 \xrightarrow{-1} 0$
 $1 \xrightarrow{-0} 1$

Мінімізація за допомогою діаграм Вейча:

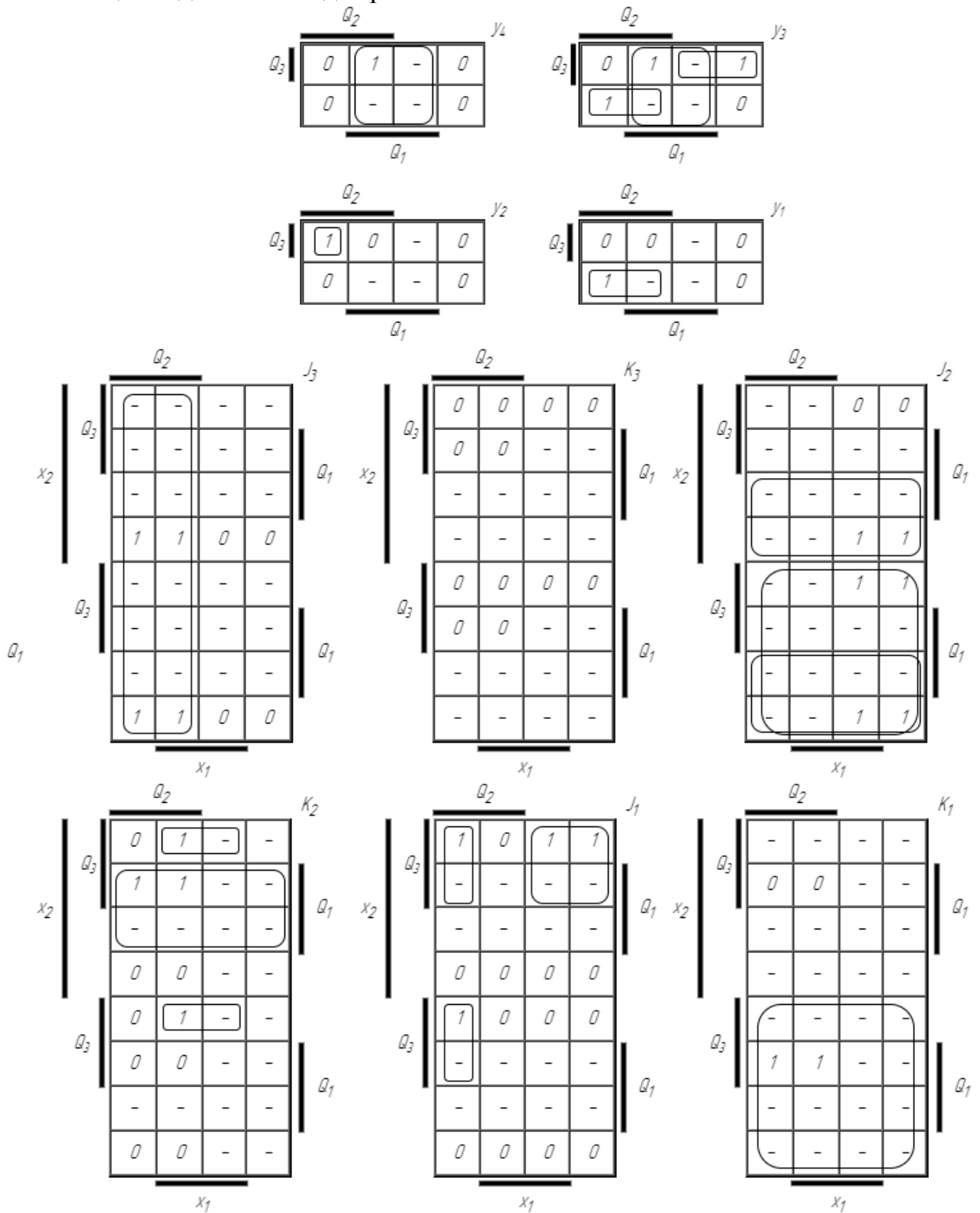


Рисунок 3.5 Діаграми Вейча

$$Y_4 = Q_1$$

$$Y_3 = Q_1 \vee \overline{Q_3} Q_2 \vee Q_3 \overline{Q_2}$$

$$Y_2 = Q_3 Q_2 \overline{Q_1}$$

$$Y_1 = \overline{Q_3} Q_2$$

$$J_3 = Q_2$$

$$K_3 = 0$$

$$J_2 = \overline{Q_3} \vee \overline{X_2}$$

$$K2 = Q1X2 \vee Q3\overline{Q1}X1$$

$$J1 = Q3Q2\overline{X1} \vee Q3\overline{Q2}X2$$

$$K1 = \overline{X2}$$

Управляющий автомат:

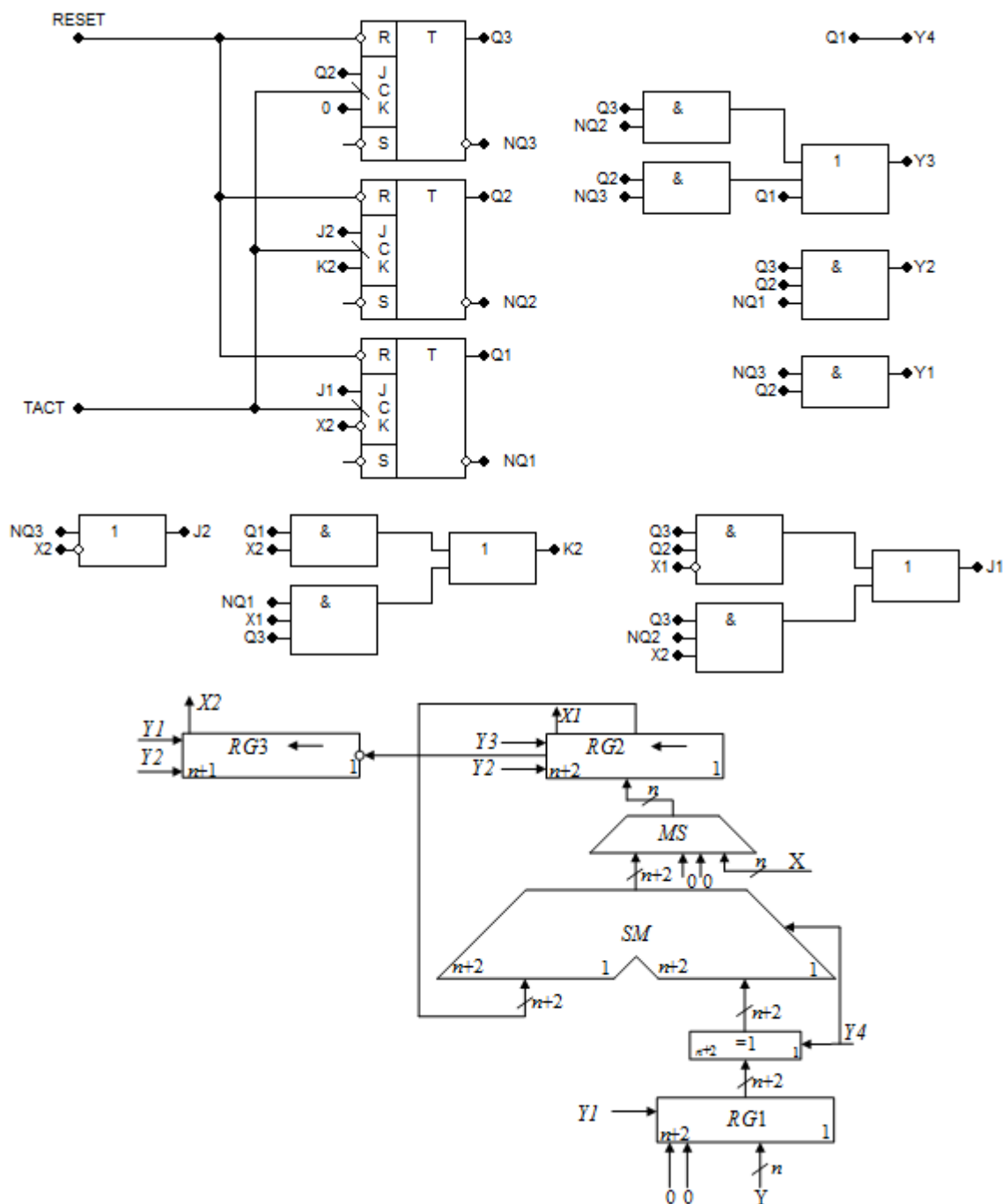


Рисунок 3.6 Управляющий автомат та пристрій для ділення другим способом

Висновок: Таким чином, виконуючи дану розрахункову роботу, я повторив такі операції над числами в двійковому коді, як множення, ділення, додавання та знаходження кореня. Для кожної операції була подана операційна схема та змістовний мікроалгоритм. Був синтезований управляючий автомат для операційного пристрою ділення першим способом, згідно з варіантом, даний операційний пристрій був побудований на JK-тригерах.