

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Кафедра обчислювальної техніки

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА

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

Виконав

Бурділь Максим Андрійович

Факультет ІОТ,

Група ІО – 62

Залікова книжка № 6203

Керівник _____

(підпис керівника)

Київ – 2017 р.

Завдання:

1. Числа X і Y 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). 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).

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 запуску 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.

Обґрунтування варіанту

$$6203_{10} = 1\ 1000\ 0011\ 1011_2$$

$$X = -\ 011110,\ 0111$$

$$Y = +010011,\ 1011$$

Основна частина

Завдання №1

$$X_{\text{пк}} = 1.011110,\ 0111$$

$$Y_{\text{пк}} = 0.010011,\ 1011$$

Представлення чисел у формі з плаваючою комою з порядком і мантисою:

X:

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

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

Y:

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

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

Представлення чисел у формі з плаваючою точкою з характеристикою і мантисою:

$$E = P + 2^m ;$$

$$m = 7;$$

$$2^7 = 10000000_2;$$

$$E_x = 10000000 + 101 = 10000101;$$

X:

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

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

$$E_y = 10000000 + 101 = 10000101;$$

Y:

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

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

Представлення за стандартом ANSI/IEEE 754-2008

$X = -0,0111100111$

X в IEEE 754:

$S = 1$

Експонента = $5 + 128 = 133_{10} = 10000101_2$

Мантиса = 011000111_2

S	Експонента								Мантиса																					
1	1	0	0	0	0	1	0	1	0	1	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

$Y = +0,0100111011$

Y в IEEE 754:

$S = 0$

Експонента = $5 + 128 = 133_{10} = 10000101_2$

Мантиса = 01010011_2

S	Експонента								Мантиса																					
0	1	0	0	0	0	1	0	1	0	1	0	1	0	0	1	1	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:

Числа множаться у прямих кодах, знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Тоді добуток двох чисел представляється у вигляді:

$$Z = YX = Yx_n2^{-n} + Yx_{n-1}2^{-(n+1)} + \dots + Yx_12^{-1};$$

$$Z = YX = ((\dots((0 + Yx_n)2^{-1} + Yx_{n-1})2^{-1} + \dots + Yx_1)2^{-1};$$

$$Z = \sum_{i=1}^n (Z_{i-1} + Yx_{n-i+1})2^{-1}.$$

2.1.2 Операційна схема:

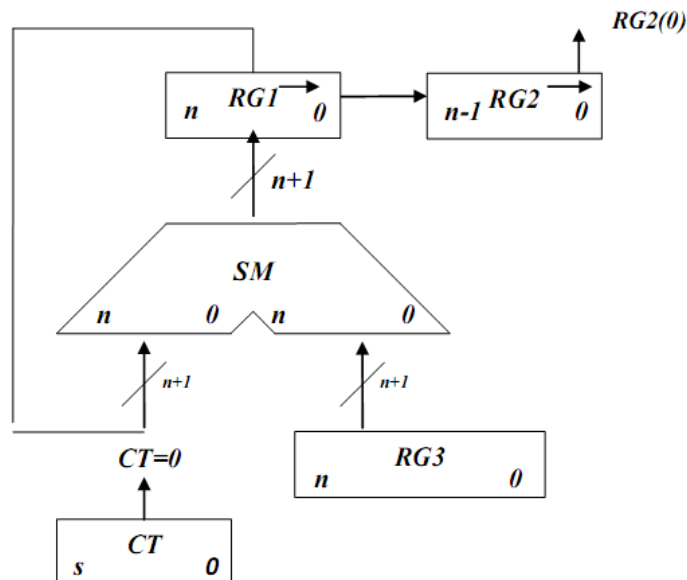


Рис. 2.1.1 – Операційна схема

2.1.3 Змістовний мікроалгоритм:

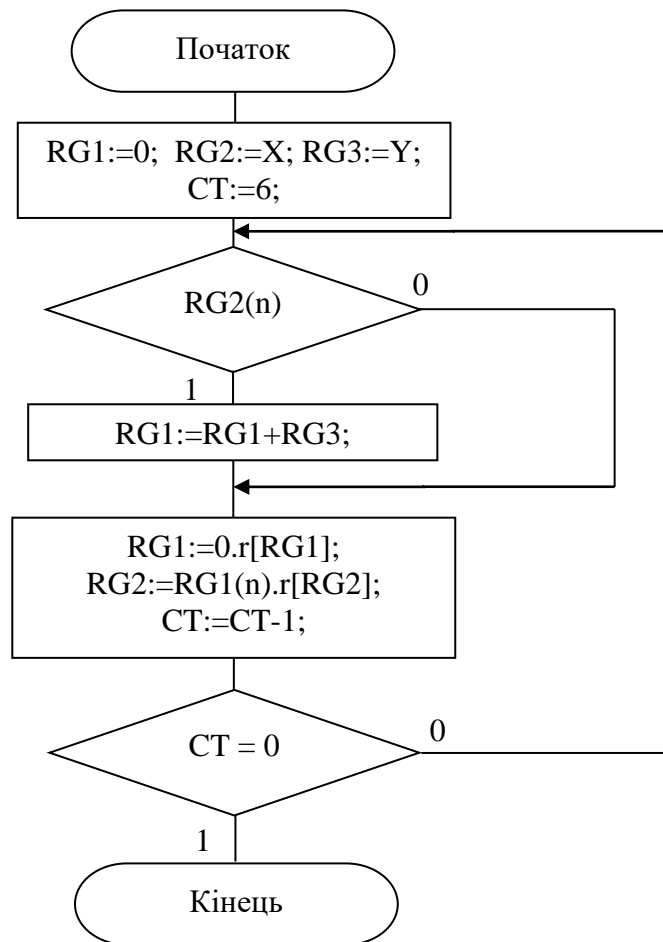


Рис. 2.1.2 – Змістовний мікроалгоритм виконання операції множення першим способом

2.1.4 Таблиця станів реєстрів:

Таблиця 2. 1. 1

Таблиця станів реєстрів для першого способу множення

№	RG1	RG2	RG3	СТ
ПС	0000000	111100	100111	110
1	0000000	011110		101
2	0000000	001111		100
3	0010011	100111		011
4	+ 0100111 = 0111010 0011101	010011		010
5	+ 0100111 = 1000100 0100010	001001		001
6	+ 0100111 = 1001001 0100100	100100		000

2.1.5 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: $P_x + P_y = P_z$.

$$P_x = 5; P_y = 5; P_z = 10_{10} = 1010_2$$

2.1.6 Нормалізація результату:

Отримали результат: **0100100100100**.

Знак мантиси: $1 \oplus 0 = 1$.

Робимо зсув результату вліво, доки у першому розряді не буде одиниця:

100100100100.

Порядок зменшуємо на 1: $P_z = 9$.

Запишемо нормалізований результат: 100101

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

2.2 Другий спосіб множення

2.2.1 Теоретичне обґрунтування другого способу множення:

Числа множаться у прямих кодах. Під час множення чисел у прямих кодах знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Множення другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою. Перед початком множення другим способом множник X записують у регістр RG2, а множене Y – в молодші розряди регістру RG3 (тобто в регістрі RG3 установлюють $Y_0 = Y2^{-n}$). В кожному i -му циклі множення додаванням кодів RG3 і RG1 керує цифра RG2(n), а в регістрі RG3 здійснюється зсув вліво на один розряд, у результаті чого формується величина $Y_i = 2Y_{i-1}$. Оскільки сума часткових добутків у процесі множення нерухома, зсув у регістрі RG3 можна сполучити в часі з підсумовуванням (як правило, $t_n > t_3$). Завершення операції множення визначається за нульовим вмістом регістру RG2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

Вираз: $Z = YX = Yx_n2^{-n} + Yx_{n-1}2^{-n+1} + \dots + Yx_12^{-1}$ подамо у вигляді

$$Z = (((0 + Y2^{-n}x_n) + Y2^{-n+1}x_{n-1}) + \dots + Y2^{-1}x_1.$$

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу: $Z_i = Z_{i-1} + Y_i \cdot x_{n-i+1}$, $Y_i = 2Y_{i-1}$.

З початковими умовами $Z_0 = 0$, $Y_0 = Y2^{-n}$, $i = 1$.

2.2.2 Операційна схема:

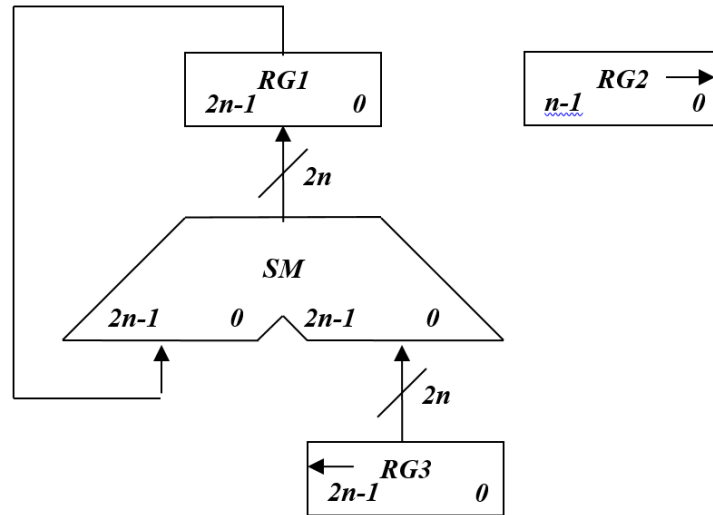


Рис. 2.2.1 - Операційна схема

2.2.3 Змістовний мікроалгоритм:

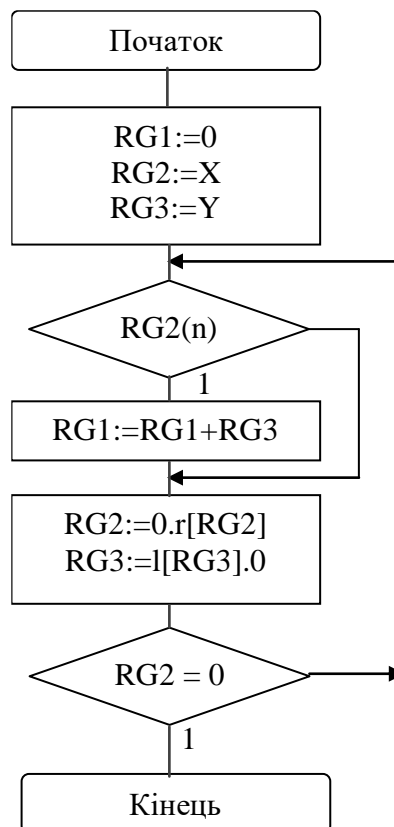


Рис. 2.2.2 - Змістовний мікроалгоритм

2.2.4 Таблиця станів регістрів:

Таблиця 2.2.1

Таблиця станів регістрів

№ц.	RG1	RG2→	RG3←
П.С.	000000000000	100101	000000100111
1	000000100111	010010	000001001110
2	000000100111	001001	000010011100
3	<div>+</div> <div>000010011100</div> <div>=</div> <div>000011000011</div>	000100	000100111000
4	000011000011	000010	001001110000
5	000011000011	000001	010011100000
6	<div>+</div> <div>010011100000</div> <div>=</div> <div>010110100011</div>	000000	100111000000

2.2.5 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: $P_x + P_y = P_z$.

$$P_x = 9; P_y = 5; P_z = 14_{10} = 1110_2.$$

2.2.6 Нормалізація результату:

Отримали результат: **010110100011**.

$$\text{Знак мантиси: } 1 \oplus 0 = 1.$$

Робимо зсув результату вліво, доки у першому розряді не буде одиниця:

10110100011.

Порядок зменшуємо на 1:

$$P_z = 13.$$

Запишемо нормалізований результат:

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

2.3 Третій спосіб множення

2.3.1 Теоретичне обґрунтування третього способу множення:

Числа множаться у прямих кодах. Під час множення чисел у прямих кодах знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Множення третім способом здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме. Під час множення третім способом вага молодшого розряду RG3 дорівнює 2^{-2n} , тому код у регістрі RG3 являє собою значення $Y2^{-n}$. На початку кожного циклу множення здійснюється лівий зсув у регістрах RG1 і RG2, а потім виконується додавання, яким керує RG1(1). У результаті підсумовування вмісту RG3 і RG1 може виникнути перенос у молодший розряд регістру RG1. У старшій частині суматора, на якому здійснюється підсумовування коду RG2 з нулями, відбувається поширення переносу. Збільшення довжини RG2 на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розряди добутку будуть знаходитися в регістрі RG2, а старші – в регістрі RG1. Час множення третім способом визначається аналогічно першому способу і дорівнює $t_m = n(t_1 + t_3)$.

Вираз $Z = YX = Yx_n2^{-n} + Yx_{n-1}2^{-n+1} + \dots + Yx_12^{-1}$ подамо у вигляді

$$Z = ((\dots ((0 + Y2^{-n}x_1)2 + Y2^{-n}x_2)2 + \dots + Y2^{-n}x_i)2 + \dots + Y2^{-n}x_n.$$

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу: $Z_i = 2Z_{i-1} + Y2^{-n}x_i$, $Y_i = 2Y_{i-1}$, з початковими умовами $Z_0 = 0$, $i = 1$.

2.3.2 Операційна схема:

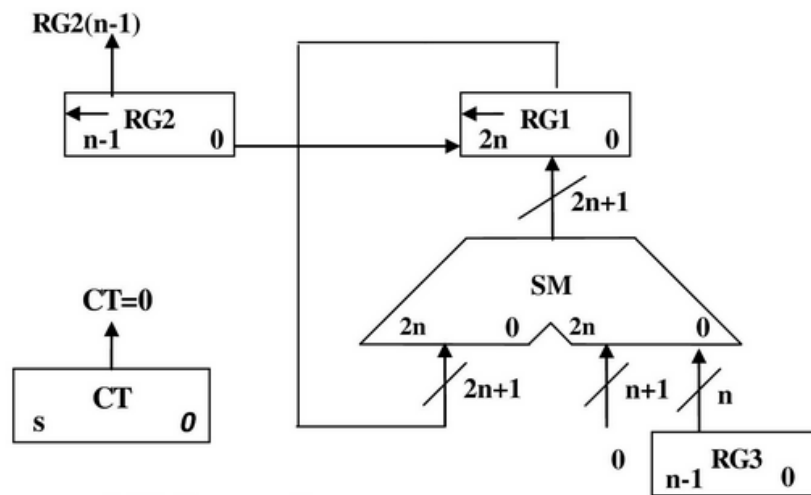


Рис. 2.3.1 - Операційна схема

2.3.3 Змістовний мікроалгоритм:

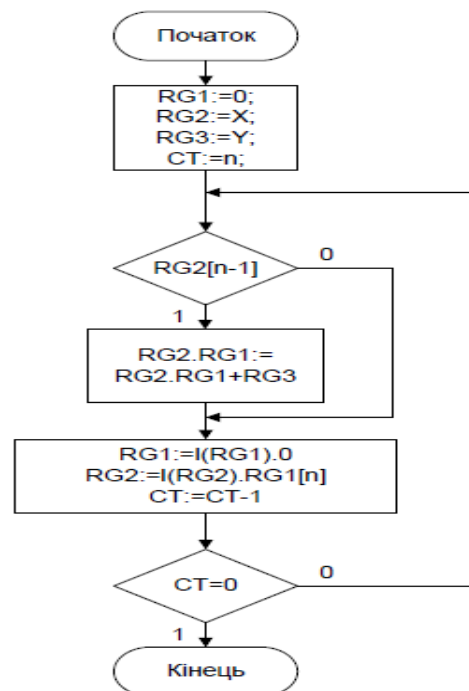


Рис. 2.3.2 - Змістовний мікроалгоритм

2.3.4 Таблиця станів регістрів:

Таблиця 2.3.1

Таблиця станів регістрів

№ ц.		RG1 ←	RG2 ←	RG3	СТ
П.С.		000000000000	101101	100111	110
1		000001001110	011010		101
2		000010011100	110100		100
3	+				
		000000100111			
	=				
		000011000011 000110000110	101000		011
4	+				
		000000100111			
	=				
		000110101101 001101011010	010000		010
5		011010110100	100000		001
6	+				
		000000100111			
	=				
		011011011011 110110110110	000000		000

2.3.5 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: $P_x + P_y = P_z$.

$$P_x = 13; P_y = 5; P_z = 18_{10} = 10010_2.$$

2.3.6 Нормалізація результату:

Отримали результат: **110110110110**.

Знак мантиси: $1 \oplus 0 = 1$.

Корекція не потрібна:

110110110110.

$$P_z = 18.$$

Запишемо нормалізований результат:

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

2.4 Четвертий спосіб множення

2.4.1 Теоретичне обґрунтування четвертого способу множення:

Числа множаться у прямих кодах. Під час множення чисел у прямих кодах знакові та основні розряди обробляються окремо. Для визначення знака добутку здійснюють підсумування по модулю 2 цифр, що розміщуються в знакових розрядах співмножників.

Множення виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо. Перед множенням четвертим способом множник записують у регістр RG2, а множене – в старші розряди регістру RG3 (тобто в RG3 установлюють $Y_0 = Y_{2-1}$). У кожному циклі цифра RG2(1), що знаходиться в старшому розряді регістру RG2, керує 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 $t_M = nt_P$, визначається аналогічно до другого способу. Запишу четвертий спосіб в аналітичній формі. Вираз

$$Z = Y \cdot x_n \cdot 2^{-n} + Y \cdot x_{n-1} \cdot 2^{-n+1} + \dots + Y \cdot x_1 \cdot 2^{-1}, \text{ подано у вигляді}$$

$$Z = (((((0 + Y \cdot 2^{-1} \cdot x_1) + Y \cdot 2^{-2} \cdot x_2) + \dots + Y \cdot 2^{-k} \cdot x_k) + \dots + Y \cdot 2^{-n} \cdot x_n))$$

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу: $Z_i = Z_{i-1} + 2^{-i} Y_{i-1} \cdot x_i$, з початковими значеннями $i=1$, $Y_0 = 2^{-1} Y$, $Z_0 = 0$.

2.4.2 Операційна схема:

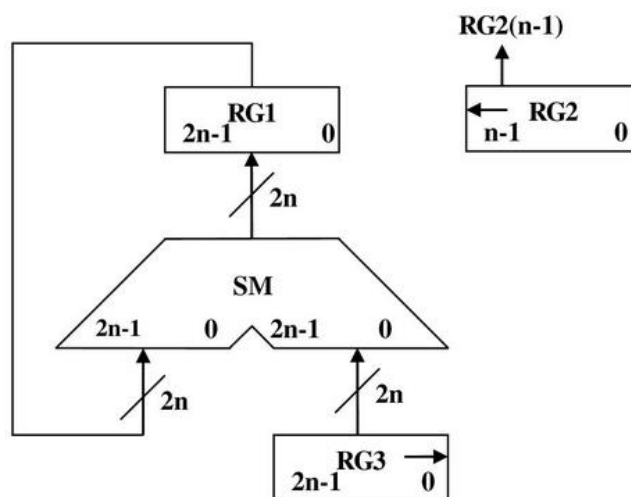


Рисунок 2.4.1- Операційна схема

2.4.3 Змістовний мікроалгоритм:

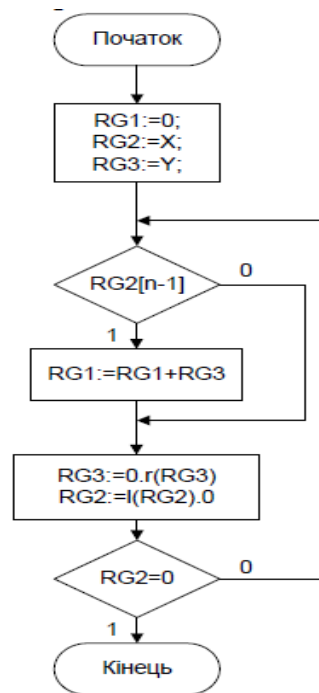


Рис. 2.4.2 - Змістовний мікроалгоритм

2.4.4 Таблиця станів регістрів:

Таблиця 2.4.1

Таблиця станів регістрів

№ ц.		RG1	RG2←	RG3 →
П.С.		000000000000	10110	010011100000
1		010011100000	101110	001001110000
2	+ 001001110000 = 011101010000		011100	000100111000
3		011101010000	111000	000010011100
4	+ 000010011100 = 011111101100		110000	000001001110
5	+ 000001001110 = 100000111010		100000	000000100111
6	+ 000000100111 = 100001100001		000000	000000010011

2.4.5 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків: $P_x + P_y = P_z$.

$$P_x = 18; P_y = 5; P_z = 23_{10} = 10111_2.$$

2.4.6 Нормалізація результату:

Отримали результат: 100001100001.

Знак мантиси: $1 \oplus 0 = 1$.

Корекція не потрібна:

100001100001;

$$P_z = 23.$$

Запишемо нормалізований результат:

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

2.5. Перший спосіб ділення

2.5.1 Теоретичне обґрунтування першого способу ділення:

Нехай ділене X і дільник Y є n -розрядними правильними дробами, поданими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Такий спосіб називається діленням із зсувом залишку. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний X), дільник Y знаходиться в регістрі RG1. Виходи RG2 підключені до входів SM безпосередньо, тобто ланцюги видачі коду з RG2 не потрібні. Час для підключення $n+1$ цифри частки визначається виразом $t=(n+1)(tt+tc)$, де tt – тривалість виконання мікрооперації додавання-віднімання; tc – тривалість виконання мікрооперації зсуву. Результат формується в регістрі RG3.

2.5.2 Операційна схема:

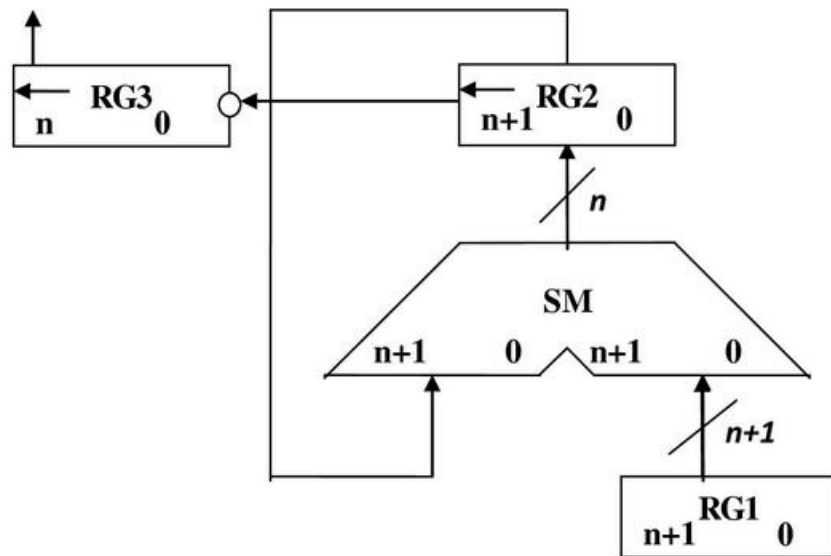


Рис. 2.5.1 – Операційна схема

2.5.3 Змістовний мікроалгоритм:

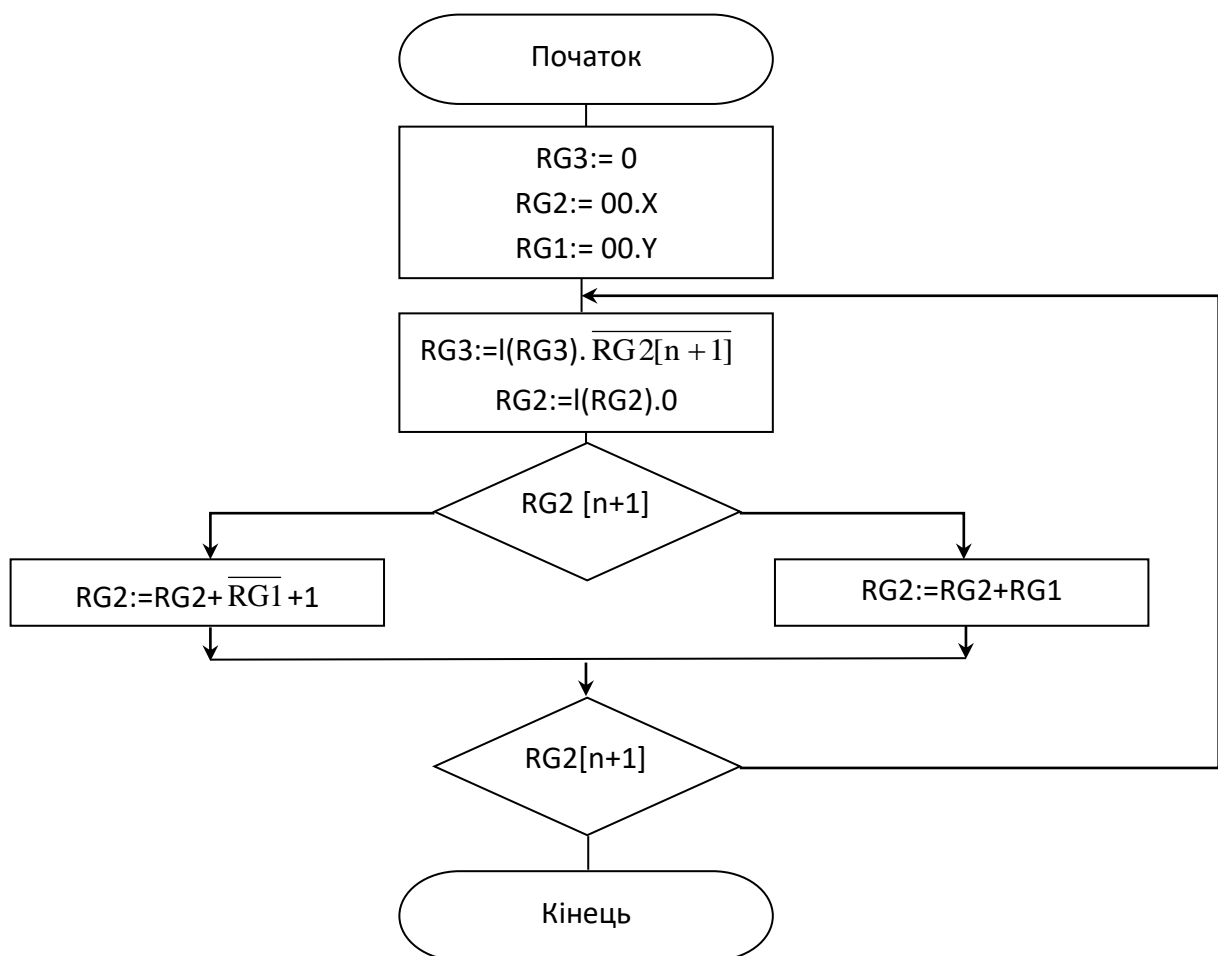


Рис. 2.5.2 – Змістовний мікроалгоритм

2.5.4 Таблиця станів регістрів:

Таблиця 2.5.1

Таблиця станів регістрів

№ ц.	RG3(Z)	RG2(X)	RG1(Y)
П.С.	000000	00100010	00100111
1	0000001	01000100 + 11011001 = 00011101	
2	0000011	00111010 + 11011001 = 00010011	
3	0000111	00100110 + 11011001 = 11111111	
4	0001110	11111110 + 00100111 = 00100101	
5	0011101	01001010 + 11011001 = 00100011	
6	0111011	01000110 + 11011001 = 00011111	
7	1110111	00111110 + 11011001 = 00010111	

2.5.8 Обробка порядків:

Порядок частки буде дорівнювати: $P_z = P_x - P_y$;

В моєму випадку $P_x=23$; $P_y=5$; $P_z=18$;

2.5.9 Нормалізація результату:

Отримали результат: 1110111.

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

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

2.6. Другий спосіб ділення

2.6.1 Теоретичне обґрунтування другого способу ділення:

Нехай ділене X і дільник Y є n -розрядними правильними дробами, поданими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

Остача нерухома, дільник зсувається праворуч. Як і при множенні з нерухомою сумою часткових добутків можна водночас виконувати підсумування і віднімання, зсув в регістрах Y, Z . Тобто 1 цикл може складатися з 1 такту, це дає прискорення відносно 1-го способу.

2.6.2 Операційна схема

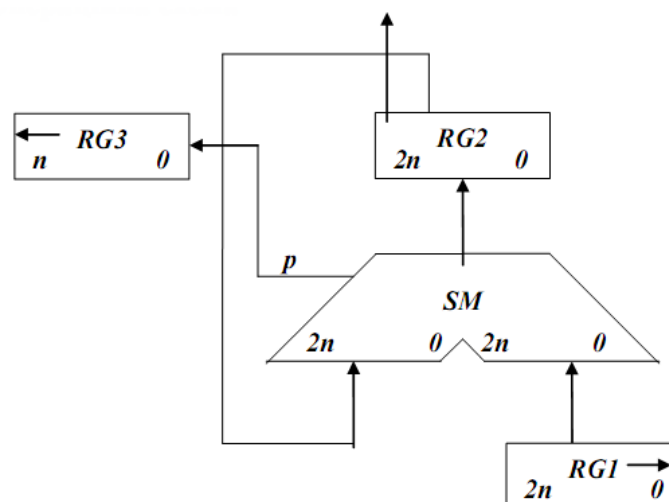


Рис. 2.6.1 – Операційна схема для операції ділення другим способом

2.6.3 Змістовний мікроалгоритм

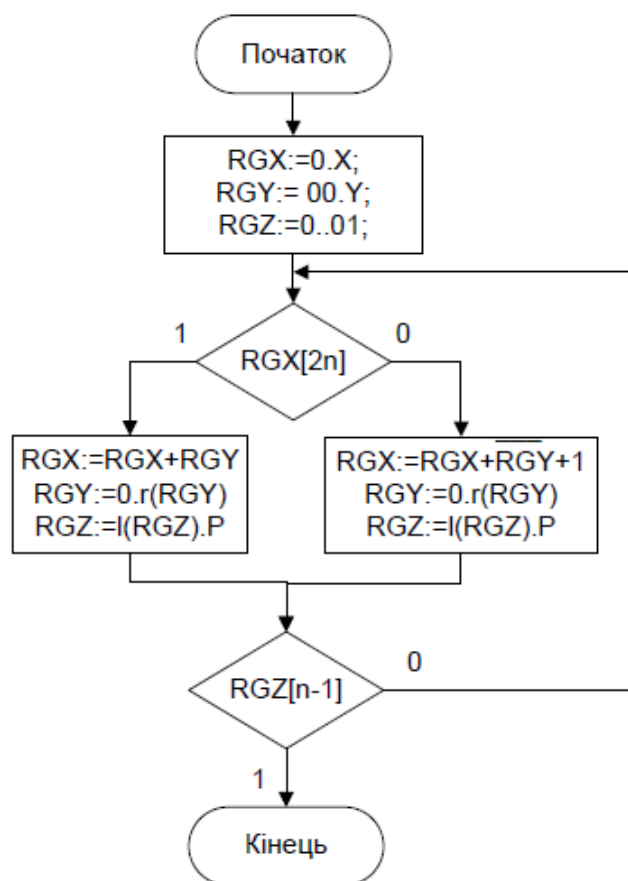


Рис. 2.6.2 – Змістовний мікроалгоритм

2.6.4 Таблиця станів регістрів

Таблиця 2.6.1

Таблиця станів регістрів

№ ц.	RG3(Z)		RG2(X)	RG1(Y)
П.С.	0000001		011110000000	001001110000
1	0000011		011110000000	000100111000
		+	110110010000	
		=	010100010000	
2	0000111		010100010000	000010011100
		+	111011001000	
		=	001110111000	
3	0001111		001110111000	000001001110
		+	111101100100	
		=	001100111100	

4	0011111	$ \begin{array}{r} 001100111100 \\ + \\ 111110110010 \\ \hline 001011101110 \end{array} $	000000100111
5	0111111	$ \begin{array}{r} 001011101110 \\ + \\ 11111011001 \\ \hline 001011000111 \end{array} $	000000010011
6	1111111	$ \begin{array}{r} 001011000111 \\ + \\ 11111101101 \\ \hline 001010110100 \end{array} $	000000001001

2.6.8 Обробка порядків:

Порядок частки буде дорівнювати: $P_z = P_x - P_y$;

В моєму випадку $P_x=18$; $P_y=5$; $P_z=13$;

2.6.9 Нормалізація результату:

Отримали результат: 111111.

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

0	1	1	0	1	1	1	1	1	1	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 мантис. Додавання мантис виконується у доповнювальних кодах, при необхідності числа у ДК переводяться в АЛП. Додавання виконується порозрядно на n -розрядному суматорі з переносом. Останній етап – нормалізація результату. Виконується за допомогою зсуву мантиси результату і коригування порядку результату. Порушення нормалізації можливо вліво і вправо, на 1 розряд вліво і на n розрядів вправо.

Виконання етапів вирівнювання порядків і додавання мантис:

1. Порівняння порядків.

$$P_x = +13_{10} = +1101_2$$

$$P_y = +5_{10} = +0101_2$$

$$P_x > P_y \Rightarrow$$

$$\Delta = P_x - P_y = 13_{10} - 5_{10} = 8_{10} = 1000_2$$

2. Вирівнювання порядків.

Робимо зсув вправо мантиси числа Y , зменшуючи Δ на кожному кроці, доки Δ не стане 0.

Таблиця 2.7.1

Таблиця зсуву мантиси на етапі вирівнювання порядків

M_y	Δ	Мікрооперація
0, 100111	1000	Початковий стан
0, 010011	0111	$M_y \rightarrow \Delta := \Delta - 1$
0, 001001	0110	$M_y \rightarrow \Delta := \Delta - 1$
0, 000100	0101	$M_y \rightarrow \Delta := \Delta - 1$
0, 000010	0100	$M_y \rightarrow \Delta := \Delta - 1$
0, 000001	0011	$M_y \rightarrow \Delta := \Delta - 1$
0, 000000	0010	$M_y \rightarrow \Delta := \Delta - 1$
0, 000000	0001	$M_y \rightarrow \Delta := \Delta - 1$
0, 000000	0000	$M_y \rightarrow \Delta := \Delta - 1$

3. Додавання мантис у модифікованому ДК.

Таблиця 2.7.2

Додавання мантис

M_x	1	1,	0	0	0	0	0	1
M_y	0	0,	0	0	0	0	0	0
M_z	1	1,	0	0	0	0	0	1

$$M_X = 11, 111111_{\text{ПК}} = 11, 000001_{\text{ДК}}$$

$$M_Y = 00, 000000_{\text{ПК}} = 00, 000000_{\text{ДК}}$$

$$M_Z = 11, 000001_{\text{ДК}} = 11, 111111_{\text{ПК}}$$

4. Нормалізація результату (В ПК).

Для даного результату додавання нормалізація не потрібна.

2.7.2 Операційна схема

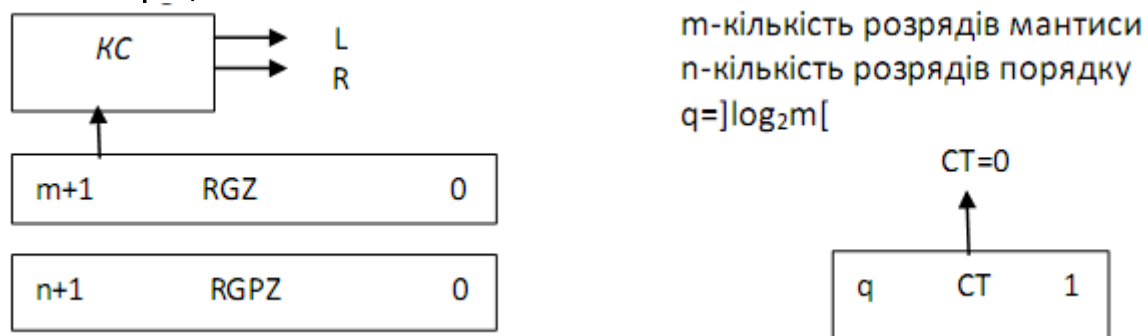


Рис. 2.7.1-Операційна схема

Виконаємо синтез КС для визначення порушення нормалізації.

Таблиця 2.7.4

Визначення порушення нормалізації

Розряди регістру RGZ			Значення функцій	
Z'0	Z0	Z1	L	R
0	0	0	0	1
0	0	1	0	0
0	1	0	1	1
0	1	1	1	0

$$L = Z_0, R = \overline{Z_1}.$$

Результат беремо по модулю, знак встановлюємо за Z'0 до нормалізації.

2.7.3 Змістовний алгоритм

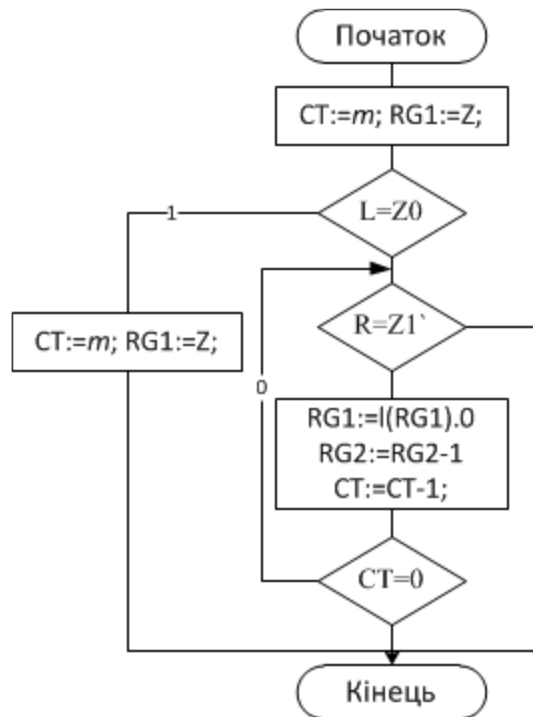


Рис. 2.7.2 – Змістовний мікроалгоритм

2.7.4 Таблиця станів регістрів

1) Додавання

Таблиця 2.7.5

Таблиця станів регістрів

№	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001000	11, 111111	0	1	01	
1	001000	00, 000001 <u>00, 000000</u> 00, 000001	0	0	00	$Z'_0 Z_0 := \overline{Z'_0 Z_0}$ $RGZ := !(RGZ).0$ $RGPZ := RGPZ - 1$ $CT := CT - 1$

2.7.8 Обробка порядків

$$P_{X+Y} = 8_{10} = 1000_2$$

2.7.9 Форма запису результату з плаваючою комою

Результат додавання $Z = X + Y$.

$$Z_{\text{пк}} = 1.11111_{\text{пк}}$$

$$P_z = 8_{10} = 1000_2$$

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

2.8. Операція добування кореня

2.8.1 Теоретичне обґрунтування операції обчислення квадратного кореня

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 операції обчислення квадратного кореня

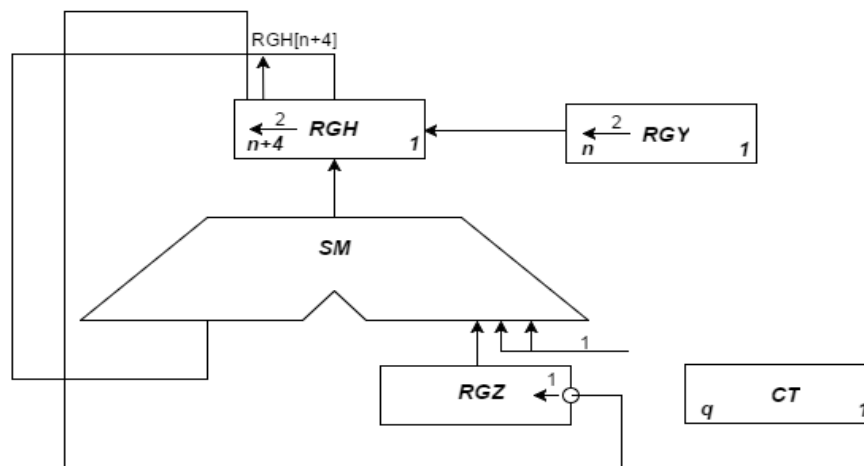


Рис. 2.8.1 – Операційна схема

2.8.3 Змістовний мікроалгоритм

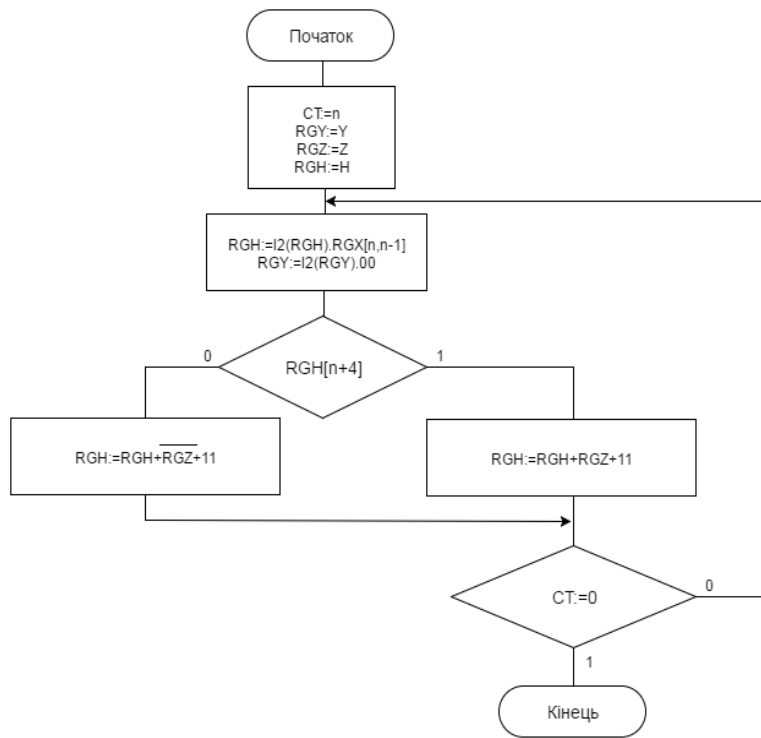


Рис.2.8.2 – Змістовний мікроалгоритм

2.8.4 Таблиця станів регістрів

Таблиця 2.8.1

Таблиця станів регістрів

№	RZ	RR	RX	CT
пс		00000000		
пз	000000	00000011	100111	110
1		00000011 + 11111111 = 00000010		
	000001	00001011	011111	101
2		00001011 + 11111011 = 00000110		
	000011	00011011	001111	100
3		00011011 + 11110011 =	000111	
	000111			011

		00001110		
		00111000		
4		00111000 + 11100011 = 00011011		
	001111	01101100	000011	010
5		01101100 + 11000011 = 00101111		
	011111	10111100	000001	001
6		10111100 + 01111111 = 00111011		
	111111	11101100	000000	000

2.8.8 Обробка порядків

$$P_z = \frac{P_x}{2};$$

В моєму випадку $P_z = 4;$

2.8.9 Запис результату

Отримали результат $Z = 11111.$

Результат нормалізований, готовий до запису у мантису:

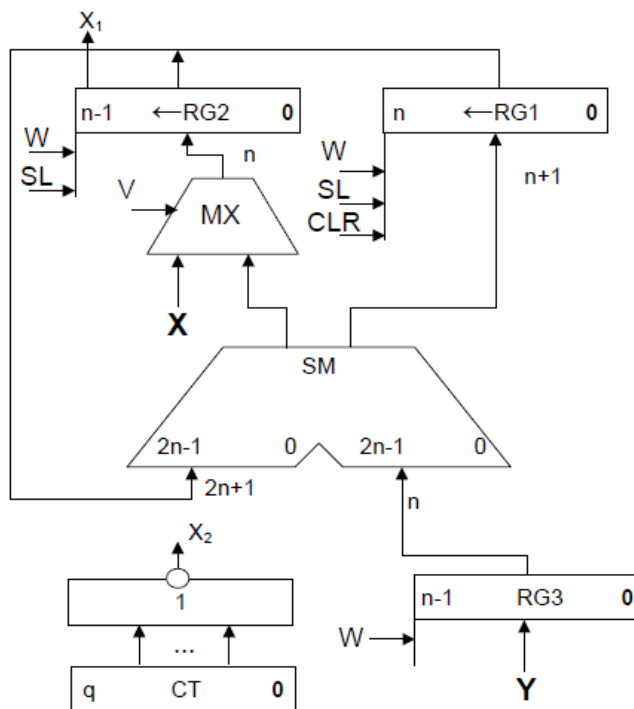
0	0	1	0	0	0	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---

Завдання №3

Варіант завдання: четвертий спосіб множення.

Синтез управляючого автомату Мілі на RS – тригерах для операції четвертого способу множення.

3.1 Функціональна схема:



3.2 Закодований мікроалгоритм:

Таблиця 2.3.2

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛУ	Позначення
RG1 := 0	R	RG2[n-1]	X1
RG2 := X	W2	RG2 = 0	X2
RG := Y	W3		
RG1 := RG1+RG3	W1		
RG1 := 0.r(RG3)	ShR		
RG2 := l(RG2).0	ShL		

Таблиця кодування сигналів

Таблиця 3.1

W2, W3, CLR1	Y1
W1	Y2
SR3, SL2	Y3

3.3 Мікроалгоритм в термінах управляючого автомата

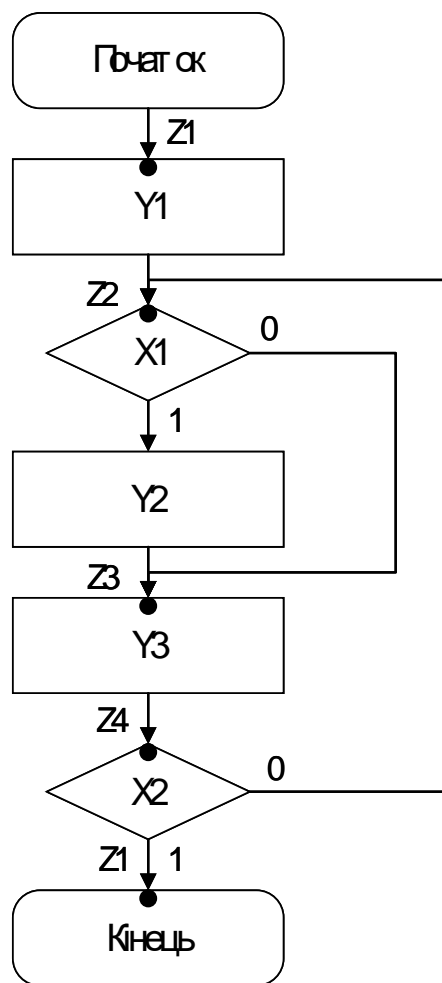


Рис. 3.1 – Закодований мікроалгоритм

3.3 Граф управляючого автомата

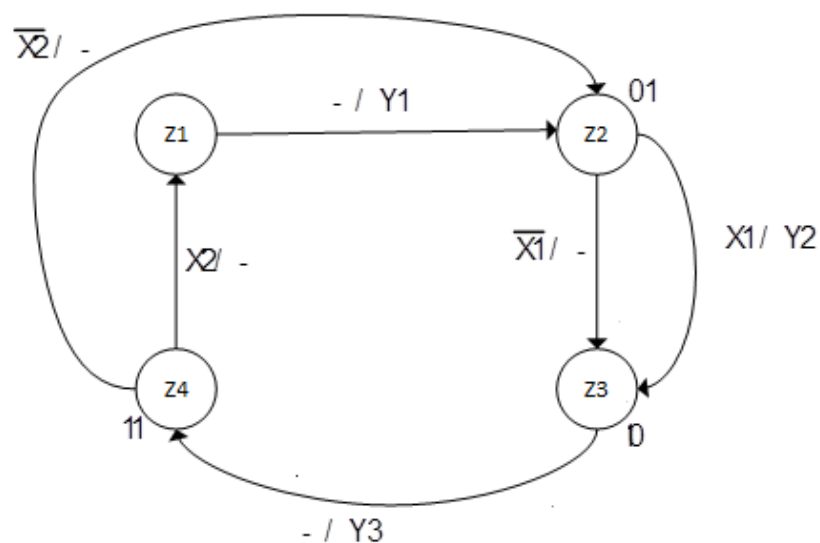


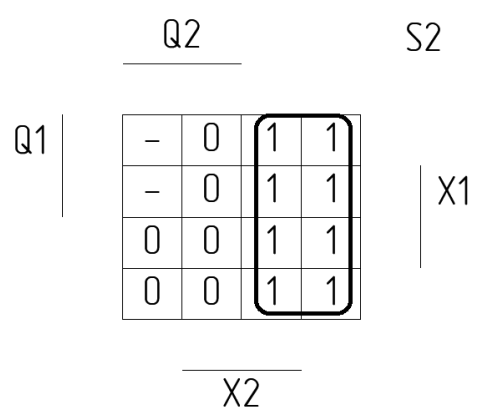
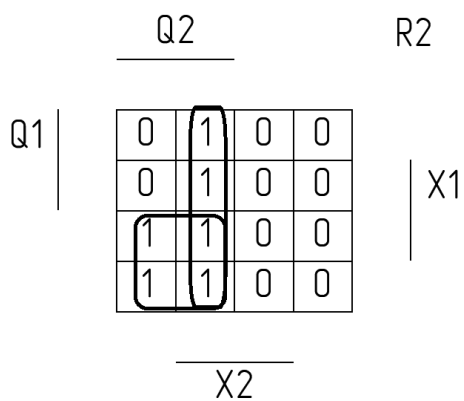
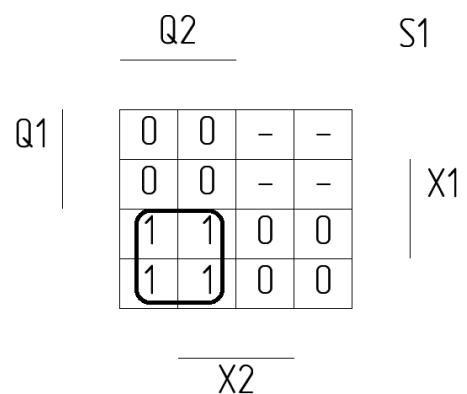
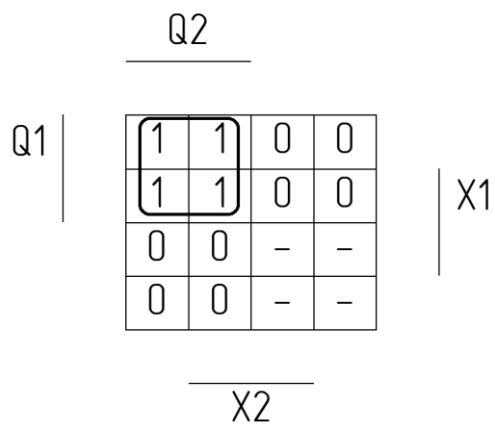
Рис. 3.2 – Граф циклічного автомата

3.4 Таблиця переходів циклічного автомата на RS-тригерах

Таблиця 3.2

Перехід	Старий стан	Новий стан	Вхідні сигнали	Вихідні сигнали	Функції збудження тригерів	
	$Q1^t \ Q2^t$	$Q1^{t+1} \ Q2^{t+1}$	$X1 \ X2$	$Y1 \ Y2 \ Y3$	$R1 \ S1$	$R2 \ S2$
$Z1 - Z2$	0 0	0 1	- -	1 0 0	- 0	0 1
$Z2 - Z3$	0 1	1 0	0 -	0 0 0	0 1	1 0
$Z2 - Z3$	0 1	1 0	1 -	0 1 0	0 1	1 0
$Z3 - Z4$	1 0	1 1	- -	0 0 1	0 -	0 1
$Z4 - Z2$	1 1	0 1	- 0	0 0 0	1 0	0 -
$Z4 - Z1$	1 1	0 0	- 1	0 0 0	1 0	1 0

3.5 Функції збудження тригерів. Мінімізація функцій методом діаграм Вейча :



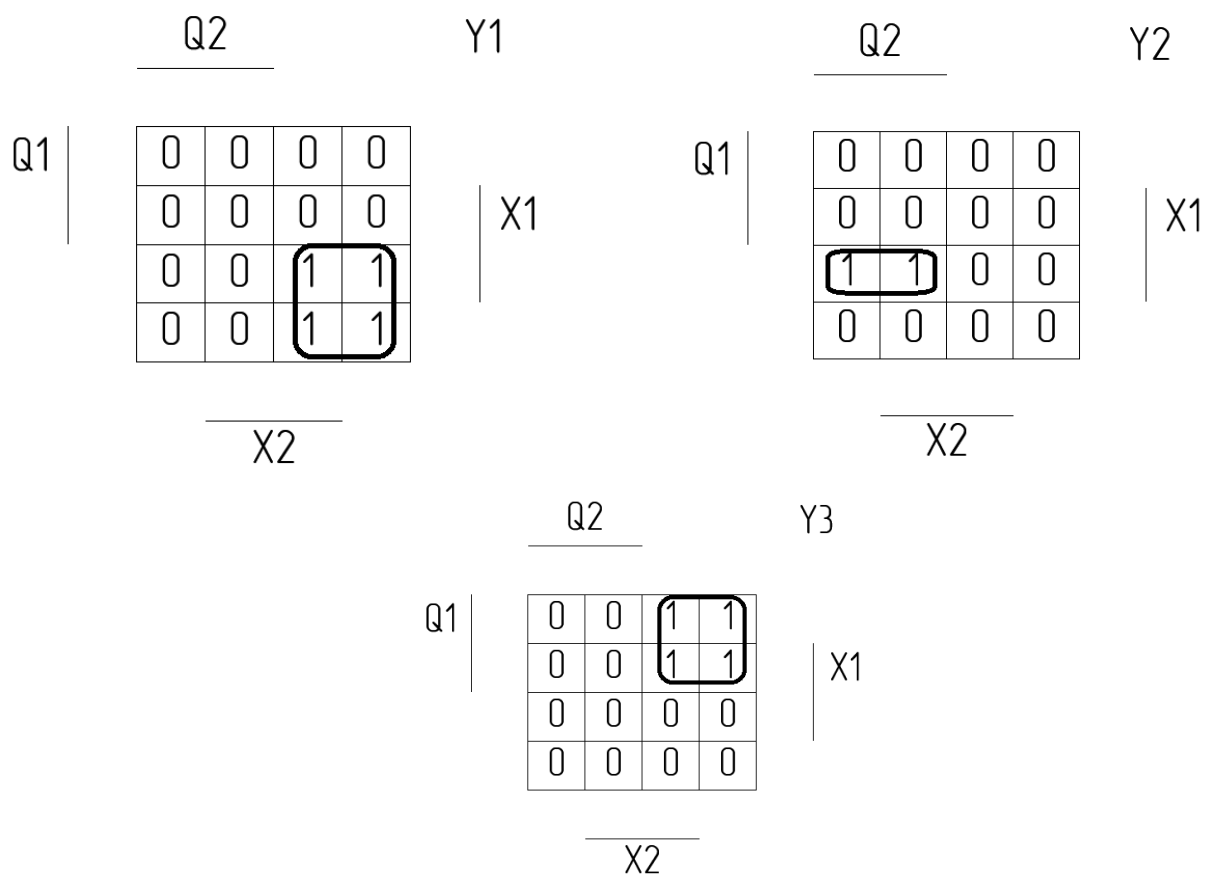


Рис. 3.3 – Мінімізація функцій

$$R_1 = Q_1 Q_2$$

$$R_2 = Q_2 X_2 \vee Q_2 \bar{Q}_1$$

$$S_1 = Q_2 \bar{Q}_1$$

$$S_2 = \bar{Q}_2$$

$$Y_1 = \bar{Q}_2 \bar{Q}_1$$

$$Y_2 = x_1 \bar{Q}_1 Q_2$$

$$Y_3 = Q_1 \bar{Q}_2$$

3.6 Функціональна схема автомата

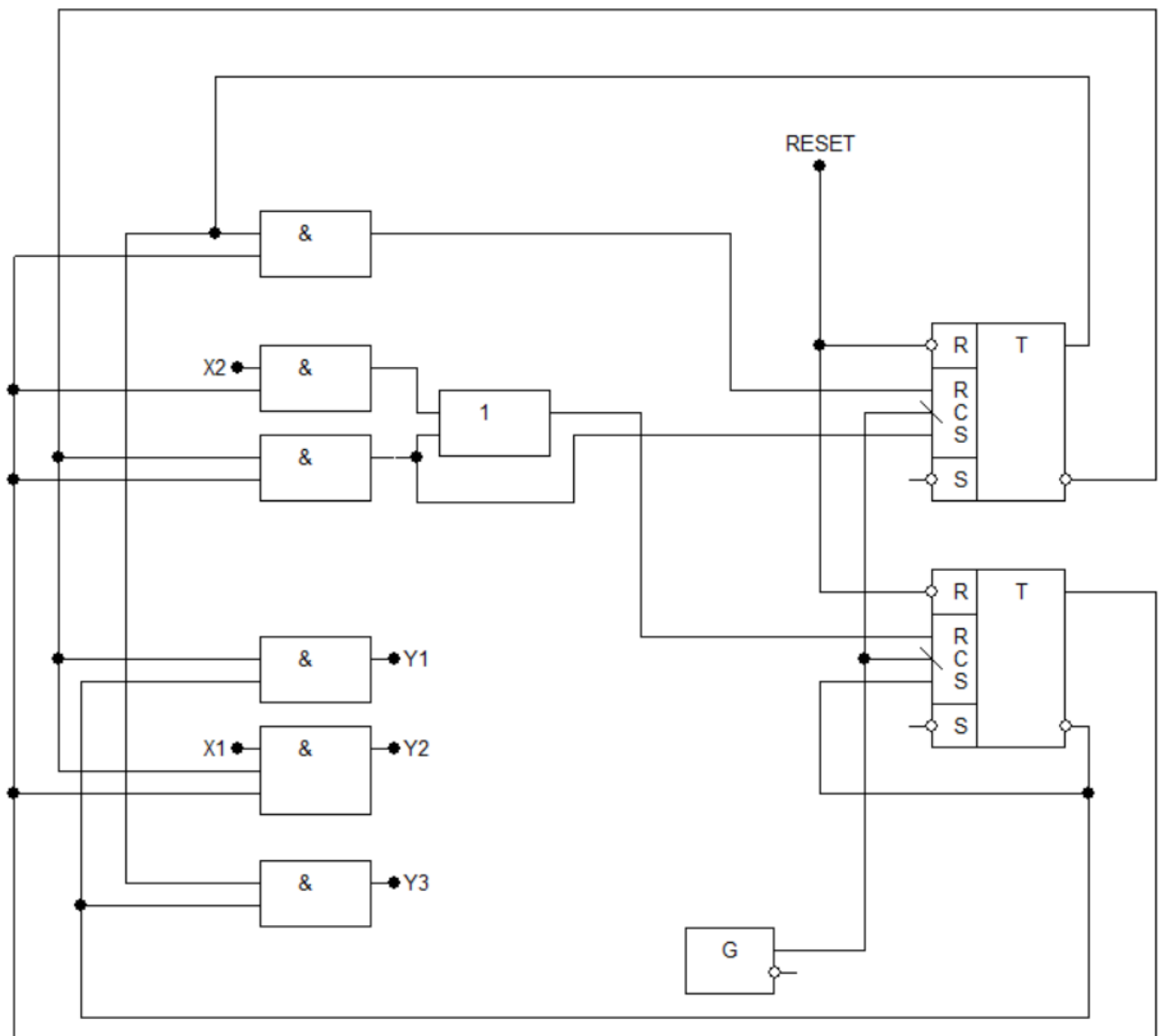


Рис. 3.5 – Функціональна схема

Висновок

У даній розрахунковій роботі було виконано операції з числами в двійковому коді з плаваючою комою, а саме: множення чотирма способами, ділення двома способами, додавання та віднімання. Для операції ділення другим способом було побудовано управляючий автомат Мілі на RS-тригерах і елементах булевого базису. Зроблено мінімізацію функцій тригерів і в середовищі AFDK побудована функціональна схема автомата.