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
Мазан Ян Владиславович
Факультет ІОТ,

Група ІВ-71

Залікова книжка № ІВ-7109

Керівник Верба О. А.

____
(підпис керівника)

Київ – 2018 р.

I. 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 Y). 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 першим операндом є результат додавання).

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 нульового 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.

II.Обґрунтування варіанту:

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

$$X = -x_7 x_6 1 x_5 x_4 0, x_3 1 x_2 x_1 \quad \text{і} \quad Y = +x_9 1 x_8 x_7 x_6 x_5, x_4 x_3 x_2 x_1 \, ,$$

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

$7109_{10} = 1101111000101_2$

X = -101000,1101 (~ -40)   Y = +111100,0101 (~60)

III.Основна частина:

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).

X = -101000,1101 Y = +111100,0101

$X_{ПК}$ = 1.101000,1101 $Y_{ПК}$ = 0.111100,0101

Запис у класичному коді:

Зсув порядків

$P_X = 6_{10} = 0.110$ $M_X = 1.101000\cancel{1101}$

$P_y = 6_{10} = 0.110$ $M_Y = 0.111100\cancel{0101}$

X:

0	1	1	0

1	1	0	1	0	0	0

Y:

0	1	1	0

0	1	1	1	1	0	0

Запис X і Y за стандартом ANSI/IEEE 754-2008:

$E_X = P_X + (2^{8-1}-1) = P_X + 127 = 110+1111111=10000101$

$\_M_X$ = 10100011010000000000000

$E_Y = P_Y + (2^{8-1}-1) = P_Y + 127 = 110+1111111=10000101$

$\_M_Y$ = 11110001010000000000000

X:

1.	1	0	0	0	0	1	0	1	1	0	1	0	0	0	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Y:

0.	1	0	0	0	0	1	0	1	1	1	1	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Порядок Мантиса

Завдання 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).

2.1. Перший спосіб множення

2.1.1 Теоретичне обґрунтування способу:

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.

Добуток Z модулів чисел дорівнює

$$Z = YX = Yx_1 2^{-1} + Yx_2 2^{-2} + \ldots + Yx_i 2^{-i} + \ldots + Yx_n 2^{-n}. \qquad (1)$$

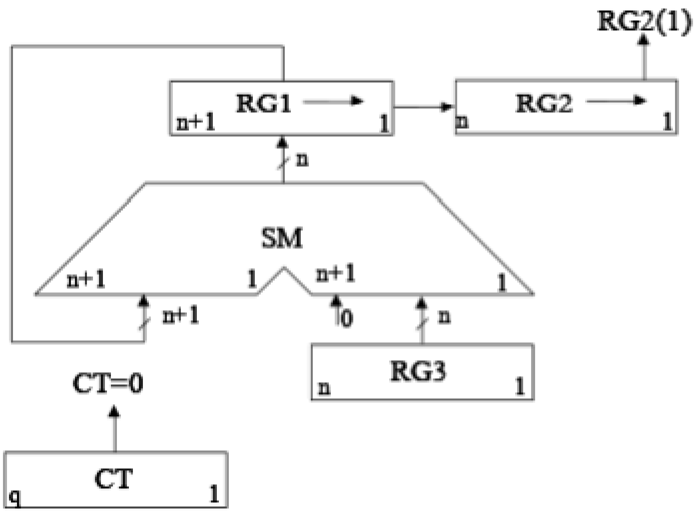
Цей вираз можна представити у вигляді:

$$Z=(((0+YX_n)2^{-1}+ YX_{n-1})2^{-1}\ldots+ YX_1)2^{-1};$$

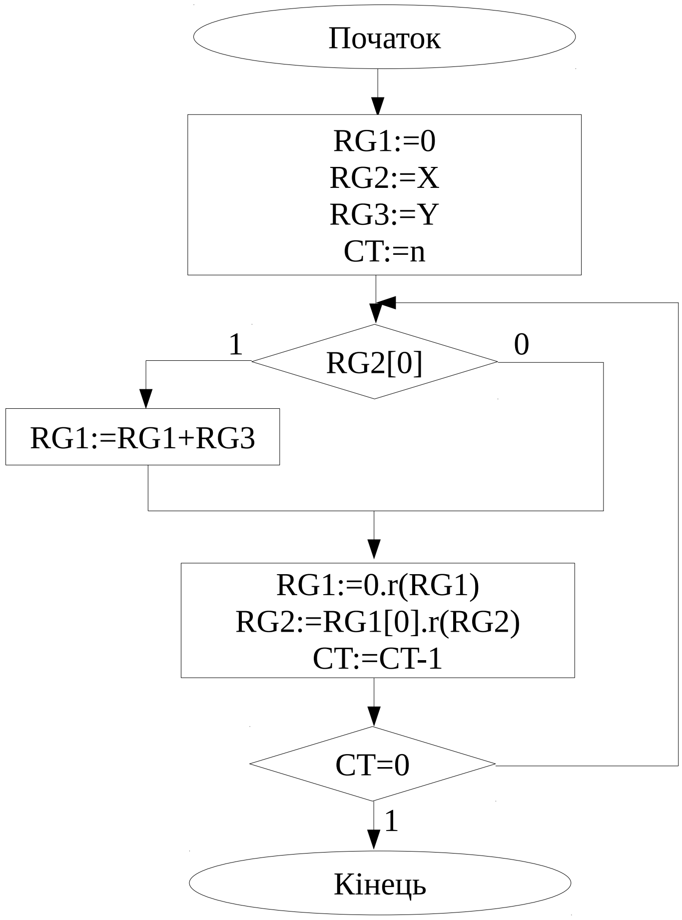
$$Z=\sum_{i=1}^{n}(Z_{i-1}+Y X_{n-i+1})2^{-1};$$

Звідси випливає, що отримані суми часткових добутків в *i*-му циклі ($i=\overline{1,n}$) зводиться до обчислення $Z_i = (Z_{i-1} + Yx_{n-i+1})2^{-1}$ з початковими значеннями i=1, Z_0=0, причому $Z_n=Z=YX$.

2.1.2 Операційна схема:



2.1.3 Змістовний мікроалгоритм:



2.1.4 Таблиця станів регістрів:

В моєму випадку X = 101000, Y = 111100, n = 110

№	RG1	RG2	RG3	CT
ПС	000000	10100**0**	111100	110
1	000000	01010**0**	111100	101
2	000000	00101**0**	111100	100
3	000000	00010**1**	111100	011
4	111100 →011110	00001**0**	111100	010
5	001111	00000**1**	111100	001
6	1001011 →100101	100000	111100	000

Z = 100101100000

2.1.5 Обробка порядків:

$P_Z = P_X + P_Y = 12 = 0.1100$

2.1.6 Нормалізація результату:

Отримано результат Z = 100101100000

Так, як у старшому розряді Z стоїть одиниця, то зсув непотрібен.

Знак мантиси: 1⊕0=1(додавання за модулем знаків множеного і множника)

Можемо записати мантису: M_Z = 1.100101

Z:

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

2.2 Другий спосіб множення

2.2.1 Теоретичне обґрунтування способу:

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.

Представимо вираз (1) у вигляді:

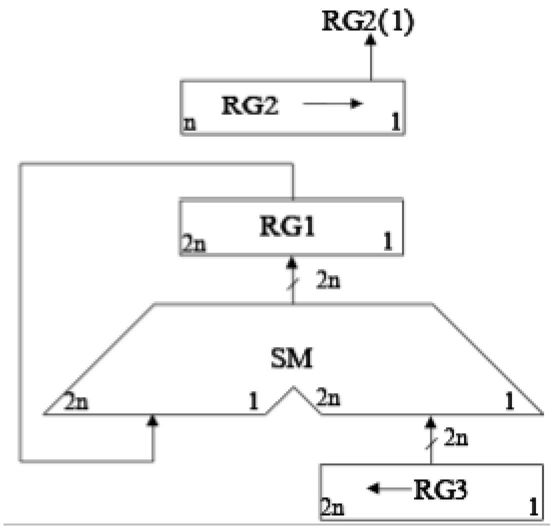
$$Z = ((\ldots((0 + Y2^{-n}x_n) + Y2^{-n+1}x_{n-1}) + \ldots + Y2^{-1}x_1.$$

Очевидно, що процес множення може бути зведений до n-кратного виконання циклу

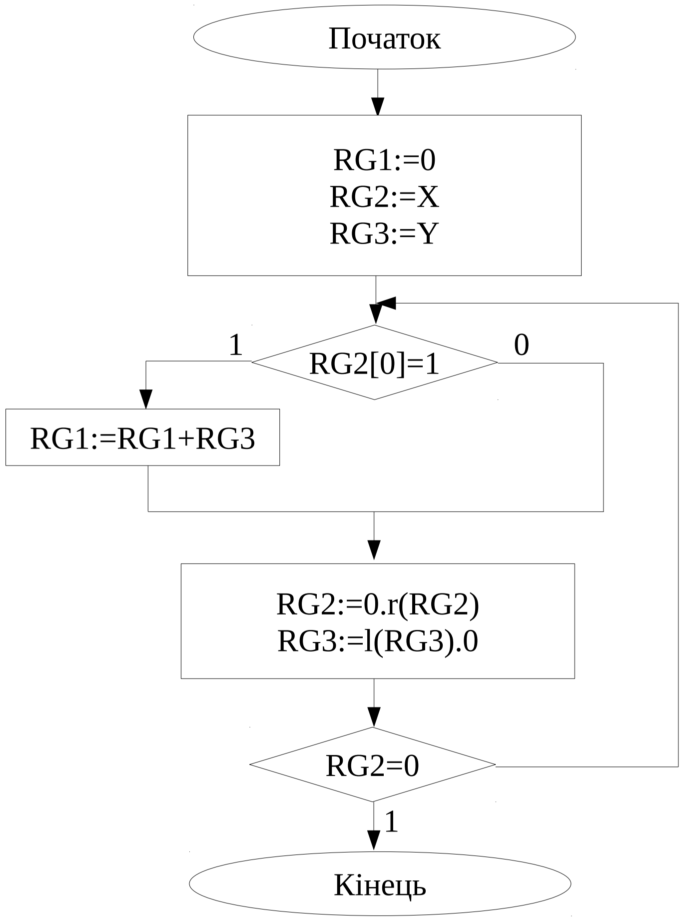
$$Z_i = Z_{i-1} + Y_i x_{n-i+1},\quad Y_i = 2Y_{i-1},$$

з початковими значеннями i=1, $Y_0=Y2^{-n}$, $Z_0=0$. Множення здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою.

2.2.2 Операційна схема:



2.2.3 Змістовний мікроалгоритм:



2.2.4 Таблиця станів регістрів:

В моєму випадку X = 1.100101, Y = 0.111100

№	RG1	RG2	RG3
ПС	000000000000	10010**1**	000000111100
1	000000111100	01001**0**	000001111000
2	000000111100	00100**1**	000011110000
3	000100101100	00010**0**	000111100000
4	000100101100	00001**0**	001111000000
5	000100101100	00000**1**	011110000000
6	100010101100	000000	111100000000

Z = 100010101100

2.2.5 Обробка порядків:

$P_Z = P_X + P_Y = 12 + 6 = 18_{10} = 0.10010$

2.2.6 Нормалізація результату:

Отримано результат Z = 100010101100

Так, як у старшому розряді Z стоїть одиниця, то зсув непотрібен.

Знак мантиси: 1⊕0=1(додавання за модулем знаків множеного і множника)

Можемо записати мантису: M_Z = 1.100010

Z:

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

2.3 Третій спосіб множення

2.3.1 Теоретичне обґрунтування способу:

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.

Представимо вираз (1) у вигляді:

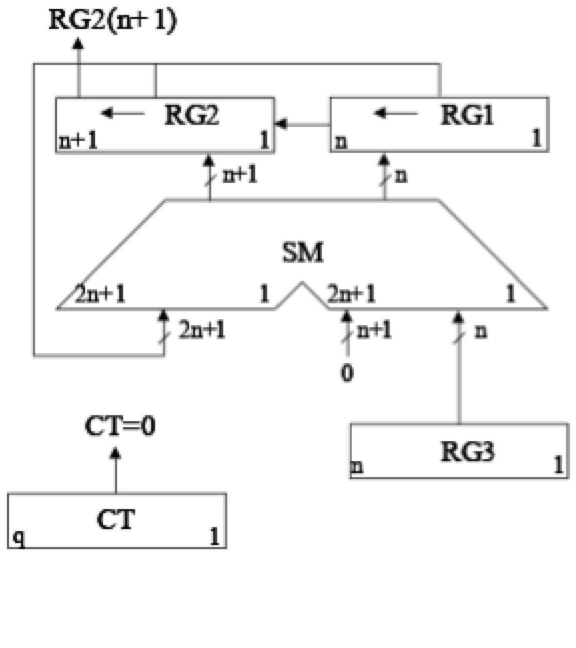
$$Z = ((\ldots((0 + Y2^{-n}x_1)2 + Y2^{-n}x_2)2 + \ldots + Y2^{-n}x_i)2 + \ldots + Y2^{-n}x_n.$$

Отже, суму часткових добутків у і-м циклі (і=$\overline{1,n}$) можна одержати за формулою

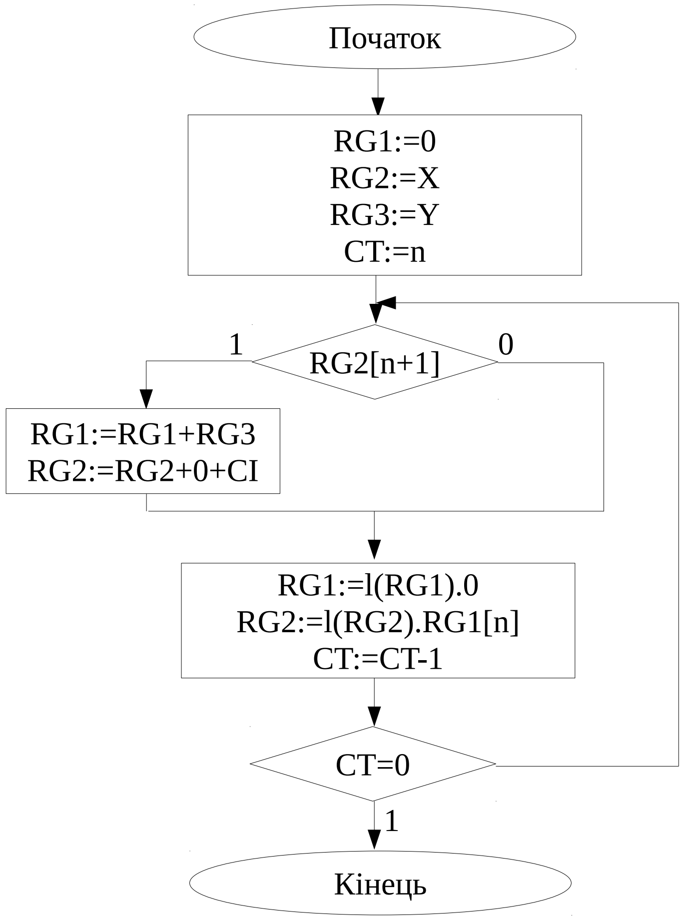
$$Z_i = 2Z_{i-1} + Y2^{-n}x_i.$$

Початковими значеннями є і=1, Z0=0. Множення здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме.

2.3.2 Операційна схема:



2.3.3 Змістовний мікроалгоритм:



2.3.4 Таблиця станів регістрів:

В моєму випадку X=1.100010, Y = 0.111100, n = 6_{10} = 110

№	RG2	RG1	RG3	CT
ПС	**1**00010	000000	111100	110
1	100010 **0**00101	111100 111000	111100	101
2	**0**01011	110000	111100	100
3	**0**10111	100000	111100	011
4	**1**01111	000000	111100	010
5	101111 **0**11111	111100 111000	111100	001
6	111111	110000	111100	000

Z = 11111111000

2.3.5 Обробка порядків:

$P_Z = P_X + P_Y = 18 + 6 = 24_{10} = 0.11000$

2.3.6 Нормалізація результату:

Отримано результат Z = 11111111000

Так, як у старшому розряді Z стоїть одиниця, то зсув непотрібен.

Знак мантиси: 1⊕0=1(додавання за модулем знаків множеного і множника)

Можемо записати мантису: M_Z = 1.111111

Z:

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

2.4 Четвертий спосіб множення

2.4.1 Теоретичне обґрунтування способу:

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.

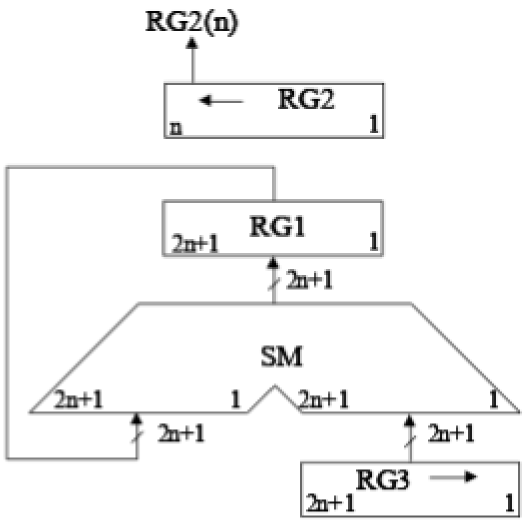
Представимо вираз (1) у вигляді:

$$Z=((...((0+Y2^{-1}x_1)+Y2^{-2}x_2)+...+Y2^{-i}x_i)+...+Y2^{-n}x_n.$$

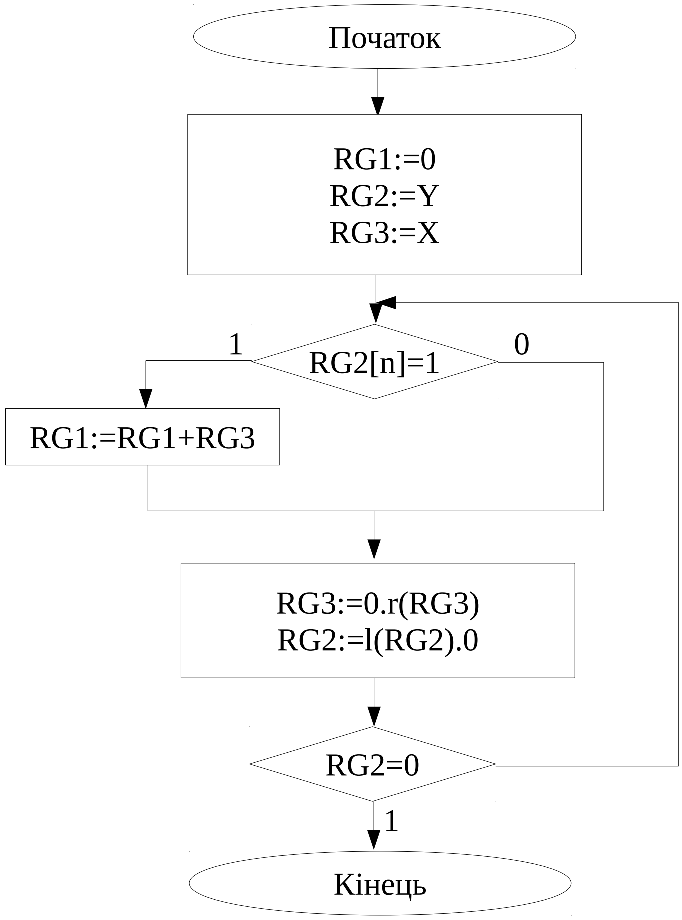
Процес множення може бути зведений до *n*-кратного виконання циклу

$Z_i=Z_{i-1}+Y_{i-1}x_i,\ Y_i=Y_{i-1}2^{-1}$ з початковими значеннями i=1, $Y_0=Y2^{-1}$, Z_0=0.

2.4.2 Операційна схема:



2.4.3 Змістовний мікроалгоритм:



2.4.4 Таблиця станів регістрів:

В моєму випадку X=1.111111, Y = 0.111100

№	RG1	RG2	RG3
ПС	000000000000	**1**11100	011111100000
1	011111100000	**1**11000	001111110000
2	101111010000	**1**10000	000111111000
3	110111001000	**1**00000	000011111100
4	111011000100	**0**00000	000001111110

Z=111011000100

2.4.5 Обробка порядків:

$P_Z = P_X + P_Y = 24+6 = 32_{10} = 0.100000$

2.4.6 Нормалізація результату:

Отримано результат Z = 111011000100

Так, як у старшому розряді Z стоїть одиниця, то зсув непотрібен.

Знак мантиси: 1⊕0=1(додавання за модулем знаків множеного і множника)

Можемо записати мантису: M_Z = 1.111011

Z:

0	1	0	0	0	0	0

1	1	1	1	0	1	1

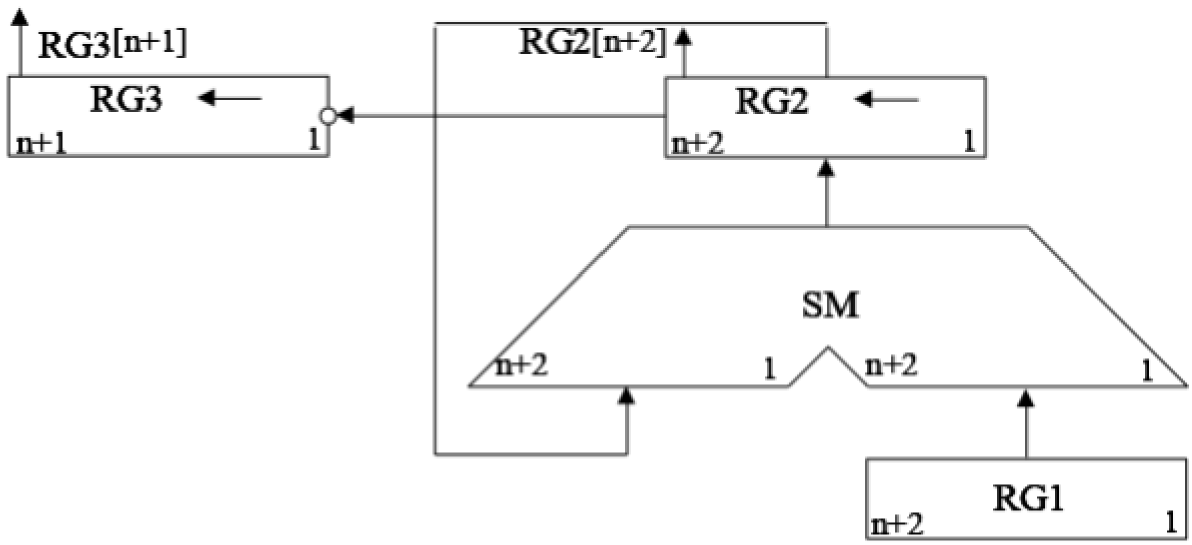
2.5 Перший спосіб ділення

2.5.1 Теоретичне обґрунтування способу:

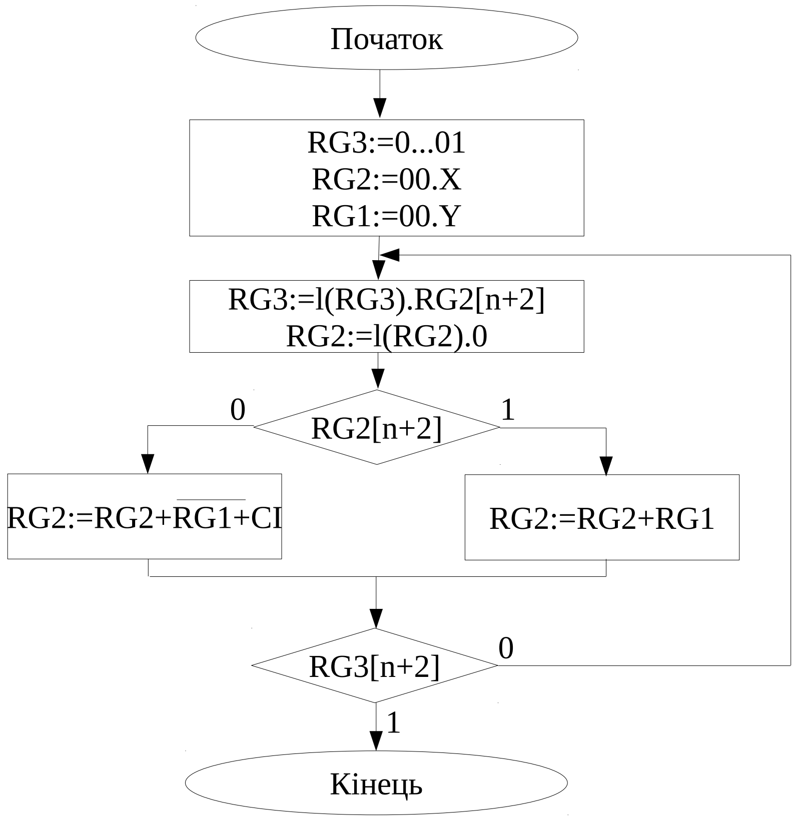
Нехай ділене X і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

При реалізації ділення за першим методом здійснюється зсув вліво залишку при нерухомому дільнику. Черговий залишок формується в регістрі Р2 (у вихідному стані в цьому регістрі записаний X). Виходи Р2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з Р2 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.

2.5.2 Операційна схема:



2.5.3 Змістовний мікроалгоритм:



2.5.4 Таблиця станів регістрів:

В моєму випадку X = 1.111011, Y= 0.111100

№	RG3	RG2	RG1
ПС	**0**000001	**0**0111011	00111100
1	**0**000010	**0**1110110 +11000011 + 1 00111010	00111100
2	**0**000101	**0**1110100 +11000011 + 1 00111000	00111100
3	**0**001011	**0**1110000 +11000011 + 1 00110100	00111100
4	**0**010111	**0**1101000 +11000011 + 1 00101011	00111100
5	**0**101111	**0**1010110 +11000011 + 1 00011001	00111100
6	**1**011111	**0**0110010 +11000011 + 1 11110101	00111100

Z = 011111

2.5.5 Обробка порядків:

$P_Z = P_X - P_Y = 32 - 6 = 24_{10} = 0.11000$

2.5.6 Нормалізація результату:

Отримано результат Z = 011111

Так, як у старшому розряді Z не стоїть одиниця, то потрібно виконати зсув мантиси:

Виконуємо зсув результату вліво, доки у першому розряді не опиниться одиниця, при цьому порядок числа зменшуємо на 1:

$M_Z = 0.011111 \rightarrow M_Z = 0.111110$; $P_Z = P_Z - 1 = 23_{10} = 0.10111$

Знак мантиси: $1 \oplus 0 = 1$(додавання за модулем знаків множеного і множника)

Можемо записати мантису: $M_Z = 1.111110$

Z:

0	1	0	1	1	1

1	1	1	1	1	1	0

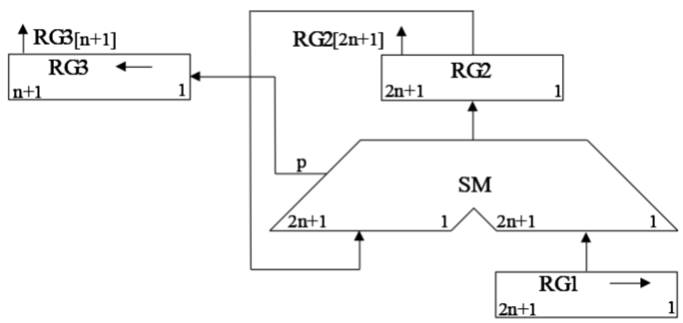
2.6 Другий спосіб ділення

2.6.1 Теоретичне обґрунтування способу:

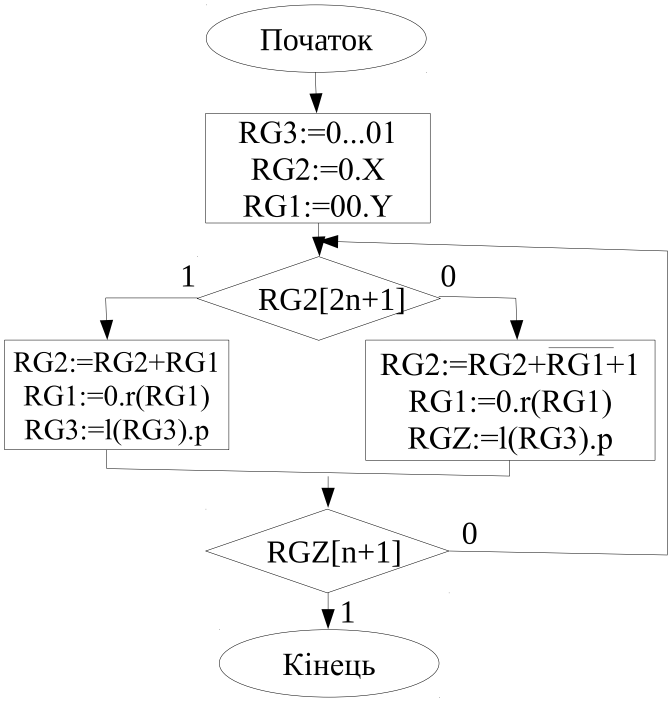
Нехай ділене X і дільник Y є n-розрядними правильними дробами, поданими в прямому коді. В цьому випадку знакові й основні розряди операндів обробляються окремо. Знак результату визначається шляхом підсумовування по модулю 2 цифр, записаних в знакових розрядах.

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:



2.6.3 Змістовний мікроалгоритм:



2.6.4 Таблиця станів регістрів:

В моєму випадку X = 1.111110, Y= 0.111100, n = 6

(RG1 інвертується до останньої одиниці, нулі далі залишаються)

№	RG3	RG2	RG1
ПС	**0**000001	0111110000000	0111100000000
1	**0**000011	+1000100000000 0000010000000	0001111000000
2	**0**000110	+1110001000000 1110011000000	0000111100000
3	**0**001100	+0000011110000 1110110110000	0000011110000
4	**0**011000	+0000001111000 1111000101000	0000001111000
5	**0**110000	+0000001111000 1111010100000	0000000111100
6	**1**100000	+0000000011110 1111010111110	0000000011110

Результат: Z = 100000

2.6.5 Обробка порядків:

$P_Z = P_X - P_Y = 23 - 6 = 17_{10} = 0.10001$

2.6.6 Нормалізація результату:

Отримано результат Z = 100000

Так, як у старшому розряді Z стоїть одиниця, то зсув непотрібен.

Знак мантиси: 1⊕0=1(додавання за модулем знаків множеного і множника)

Можемо записати мантису: M_Z = 1.100000

Z:

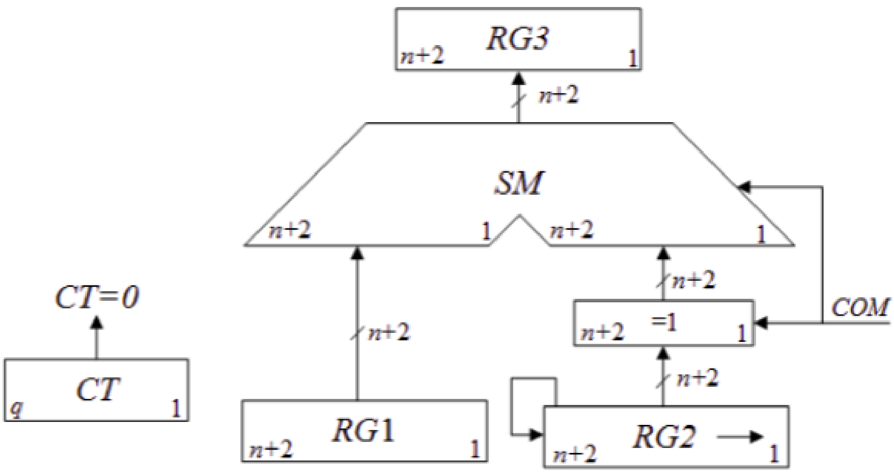
0	1	0	0	0	1		1	1	0	0	0	0	0

2.7 Додавання чисел, поданих у формі з плаваючою комою

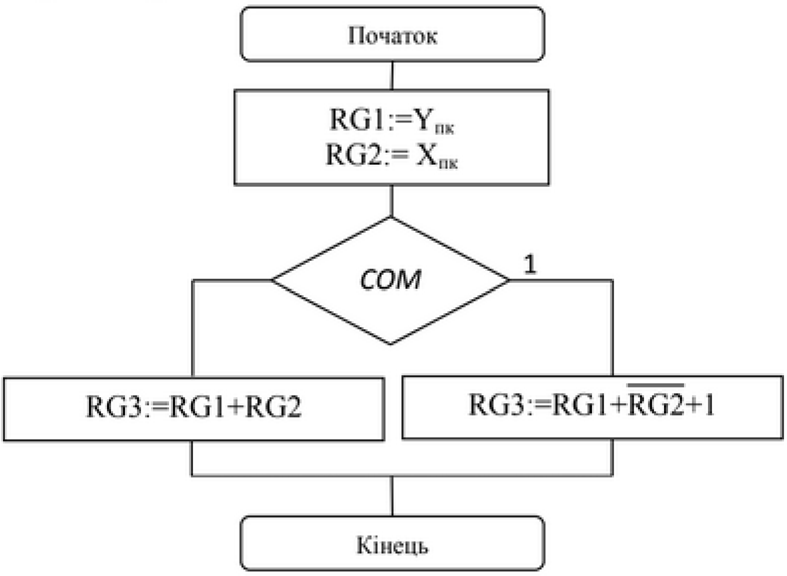
2.7.1 Теоретичне обґрунтування способу:

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 доповняльних кодах.

2.7.2 Операційна схема:



2.7.3 Змістовний мікроалгоритм:



2.7.4 Обчислення порядку суми, вирівнювання порядків та додавання мантисою

$P_X = 17_{10} = 0.10001$

$P_y = 6_{10} = 0.110$

$P_Z = \max(P_X,P_Y) = 0.10001$

$\Delta = P_X - P_Y = 17 - 6 = 11_{10} = 0.1011$

$M_X = 1.100000$

$M_Y = 0.111100$

Вирівнювання порядків

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

M_Y	Δ
0,111100	1011
0,011110	1010
0,001111	1001
0,000111	1000
0,000011	0111
0,000001	0110
0,000000	0101
0,000000	0100
0,000000	0011
0,000000	0010
0,000000	0001
0,000000	0000

Додаємо мантиси у ДК

$M_{XДК}$ = 11.011111 + 1=11.100000

$M_{YДК} = M_{YПК}$ = 0.000000

$M_X + M_Y$ = 11.100000 = $M_{ZДК}$

$M_{ZПК}$ = $\overline{11.100000\text{-}1}$ = 1.100000

2.7.5 Обробка порядків:

P_Z = 0.10001

M_Z = 1.100000

2.7.6 Нормалізація результату:

Нормалізація непотрібна

Z:

0	1	0	0	0	1

1	1	0	0	0	0	0

2.8 Обчислення квадратного кореня

2.8.1 Теоретичне обґрунтування способу:

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

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

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

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

Виконання операції зводиться до послідовності дій:

1. Одержання остачі.

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

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

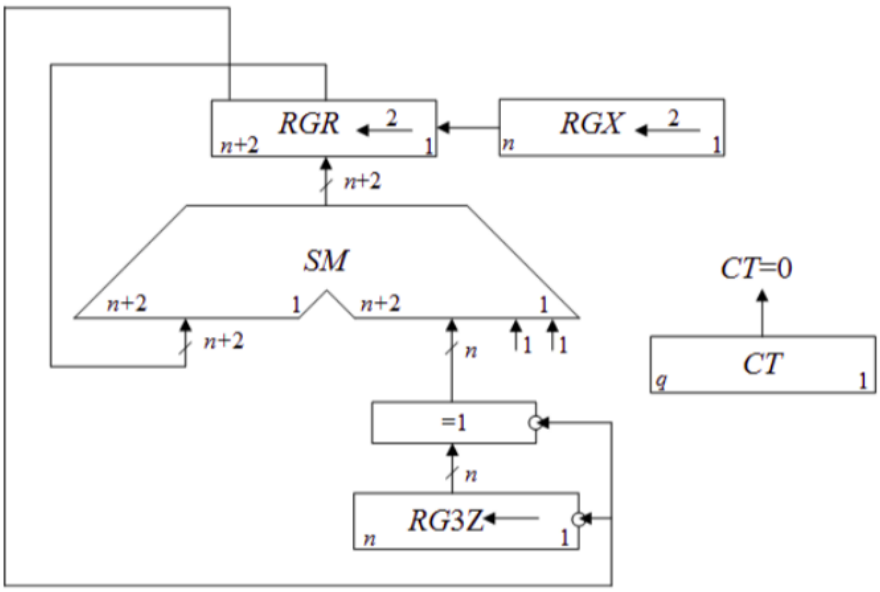
3. Якщо $R_{i+1}{}' < 0$, то $Z_{i+1} = 0$, $R_{i+1} = R_{i+1}{}' + Z_i - 2^{-i-2}$.

Відновлення остачі додає зайвий такт, але можна зробити інакше:

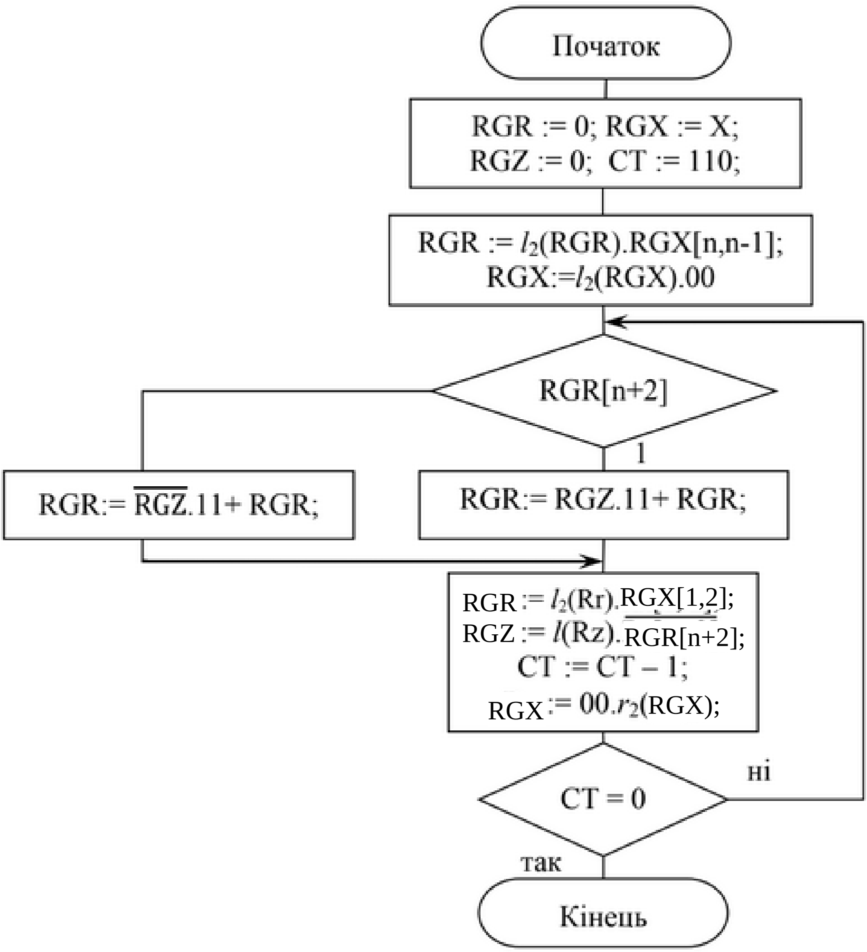
$R_{i+2} = 2R_{i+1}{}' + Z_i + 2^{-i-2} + 2^{-i-3}$, тоді корінь добувається без відновлення залишку.

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:



2.8.3 Змістовний мікроалгоритм:



2.8.4 Таблиця станів регістрів:

В моєму випадку X = 100000

Так як порядок X дорівнює 17 (P_X = 0.10001), то я повине зсунути мантису на один порядок праворуч, щоб отримати парний порядок.

P_X = 0.10010 (18_{10}), X = 010000

№	RGZ	RGR	RGX	CT
ПС	000000	00000000 00000001	010000 000000	110
1	 000001	**0**0000001 +11111111 00000000 00000000	000000	101
2	 000010	**0**0000000 +11111011 11111011 11101100	000000	100
3	 000100	**1**1101100 +00001011 11110111 11011100	000000	011
4	 001000	**1**1011100 +00010011 11101111 10111100	000000	010
5	 010001	**1**0111100 +00100011 11011111 01111100	000000	001
6	 100010	**0**1111100 +10111011 00110111 11011100	000000	000

Результат: Z = 111011

2.8.5 Обробка порядків:

$P_Z = P_X/2 = 9_2 = 0.1001_2$

2.8.6 Нормалізація результату

Так, як в результаті Z в найстаршому розряді стоїть одиниця, то нормалізація непотрібна.

M_Z = 0.111011

Z:

0	1	0	0	0	1

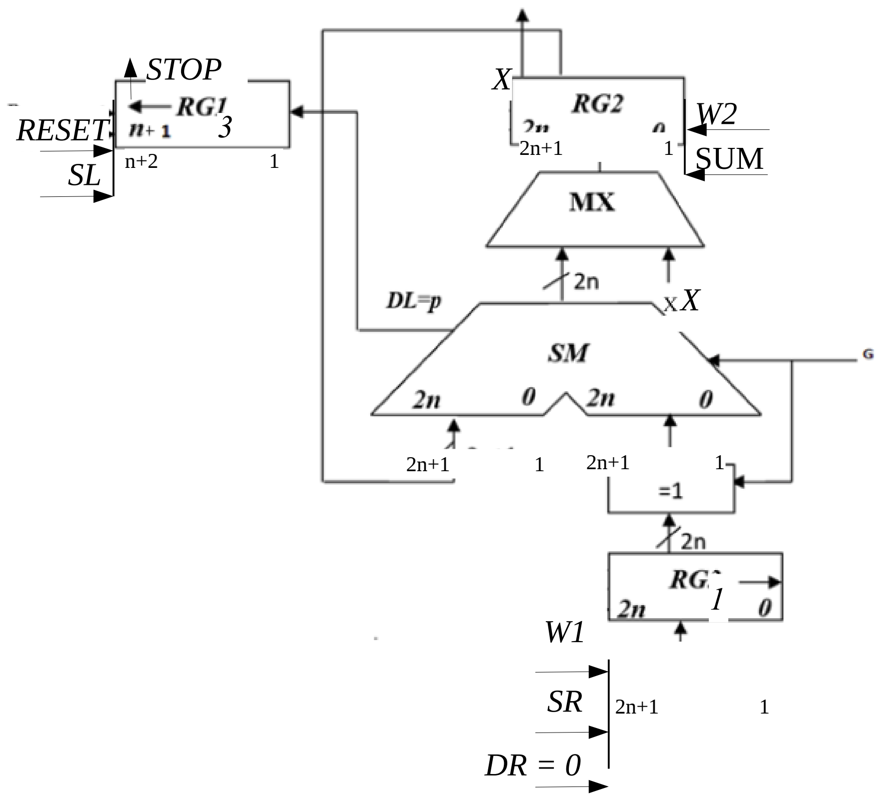
1	1	1	1	0	1	1

Завдання 3

$x_3x_2x_1=101_2=5_{10}$

Мені потрібно виконати індивідуальне завдання для ділення другим способом і побудувати автомат Мілі на RS-тригерах.

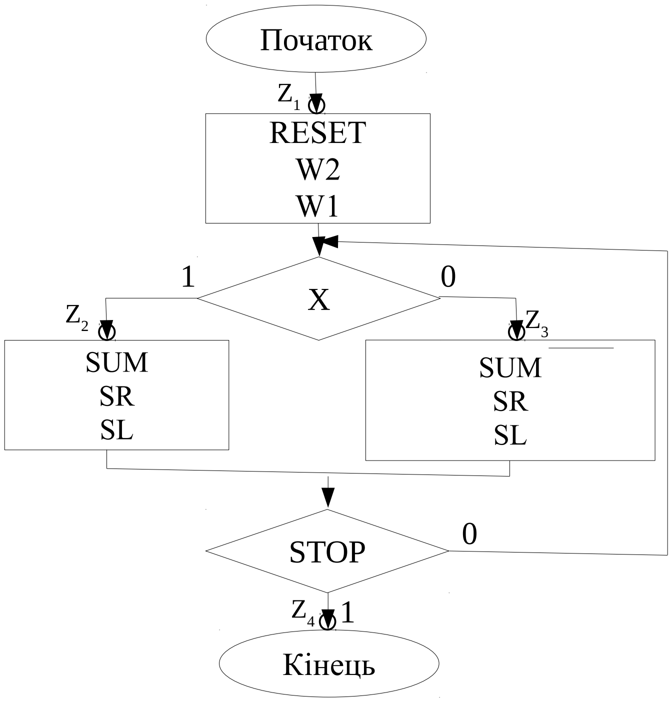
3.1 Функціональна схема



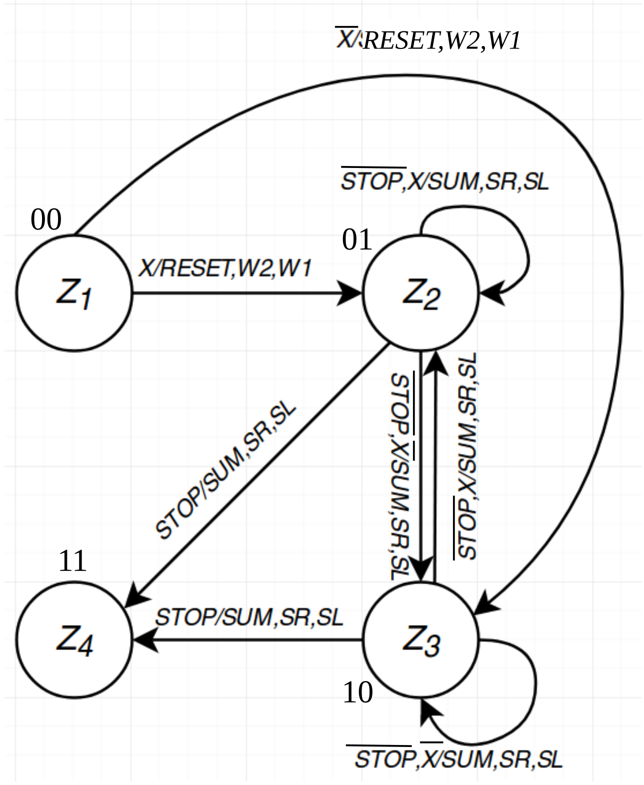
3.2 Закодований (структурний) мікроалгоритм

Кодування мікрооперацій та логічних умов

Кодування мікрооперацій		Кодування логічних умов	
RG3:=00...01	RESET	RG2[2n+1]	X
RG2:=0.X	W2	RG3[n+2]	STOP
RG1:=00.Y	W1		
RG2:=RG2+RG1 RG2:=RG2+$\overline{RG1}$+1	SUM		
RG1:=0.r(RG1)	SR		
RGZ:=l(RG3).p	SL		



3.3 Граф управляючого автомата Мура

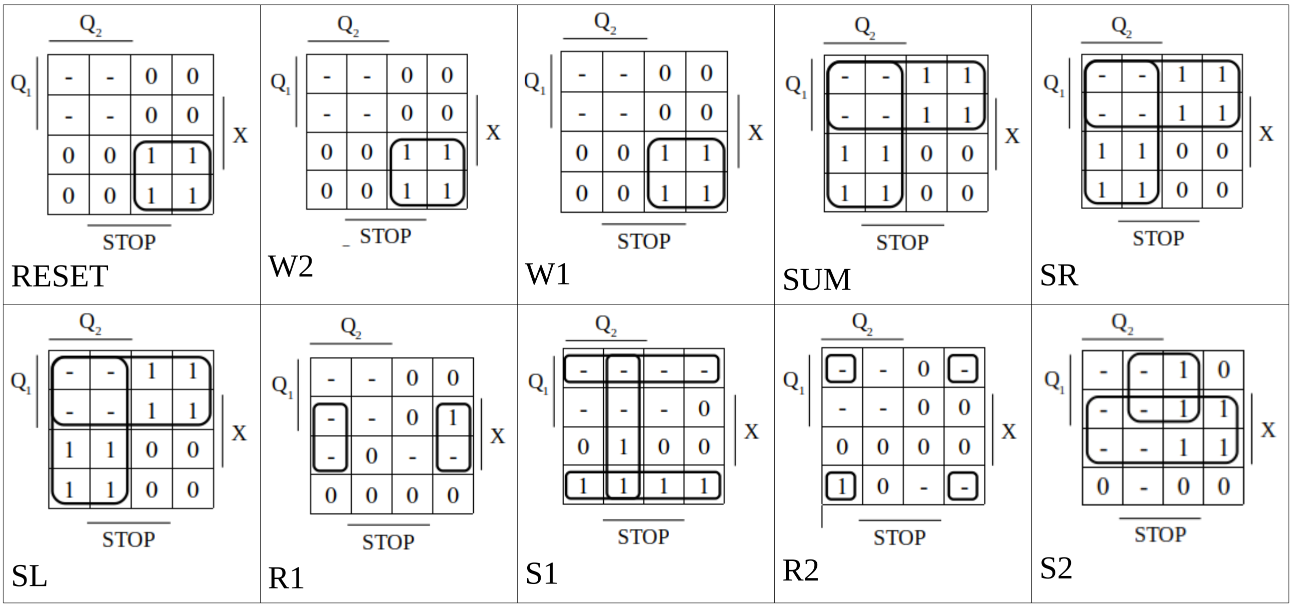


3.4 Структурна таблиця станів автомата, спрощення управляючих сигналів, побудова управляючого автомата

	R	S	
0	-	0	0
0	0	1	1
1	1	0	0
1	0	-	1

Перехід	Код ПС	Код СП	Лог. умови		Керуючі сигнали						Функції збудження тригерів			
	Q^1Q^2	$Q^{1'}Q^{2'}$	X	STOP	RESET	W2	W1	SUM	SR	SL	R_1	S_1	R_2	S_2
Z_1Z_2	00	01	1	-	1	1	1	0	0	0	-	0	0	1
Z_1Z_3	00	10	0	-	1	1	1	0	0	0	0	1	-	0
Z_2Z_2	01	01	1	0	0	0	0	1	1	1	-	0	0	-
Z_2Z_3	01	10	0	0	0	0	0	1	1	1	0	1	1	0
Z_2Z_4	01	11	-	1	0	0	0	1	1	1	0	1	0	-
Z_3Z_2	10	01	1	0	0	0	0	1	1	1	1	0	0	1
Z_3Z_3	10	10	0	0	0	0	0	1	1	1	0	-	-	0
Z_3Z_4	10	11	-	1	0	0	0	1	1	1	0	-	0	1

Спрощуватиму функції методом діаграм Вейча.



$$RESET = W2 = W1 = \overline{Q_2}\,\overline{Q_1}$$

$$SUM = SR = SL = Q_1 \vee Q_2$$

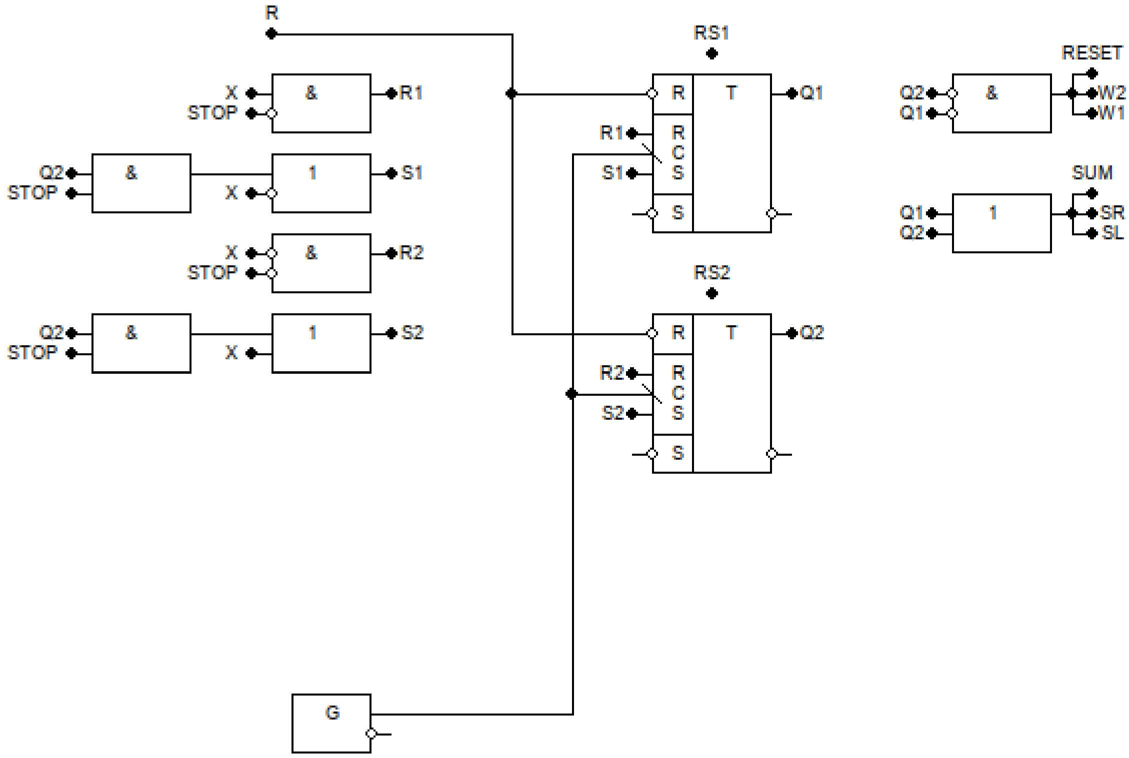
$$R_1 = x\,\overline{STOP}$$

$$S_1 = \bar{x} \vee Q_2 STOP$$

$$R_2 = \overline{STOP}\,\bar{x}$$

$$S_2 = x \vee Q_1\,STOP$$

3.5 Побудований управляючий автомат Мілі



3.6 Підрахунок часу виконання операції

Для другого способу $t_{діл} = t_{додавання} = (n+1)t_{0\ додавання}$, де n — кількість розрядів діленого і дільника

$(n+1)t_{0\ додавання} \approx 7n$, де n — кількість розрядів діленого і дільника

Під час виконання мого завдання розрахункової n дорівнював 6, тому в моєму випадку $t_{діл} = 7 \cdot 7 = 49$ умовних одиниць часу.

4. Висновок

Під час виконання даної графічної розрахункової роботи я навчився виконувати 8 простих операцій з числами, поданими у формі з плаваючою комою (чотири способи множення, два способи ділення, додавання та обчислення кореня додатного числа). Також, під час перетворень числа із форми з фіксованою у форму з плаваючою комою, я вивчив два способи подання другої вищепереліченої форми двійкових чисел: класичний варіант та стандарт ANSI/IEEE 754-2008 в короткому 32-розрядному форматі. Форма запису двійкового числа із плаваючою комою вигідніша за форму з фіксованою комою тим, що в ній відсутнє переповнення розрядної сітки, а замість цього операції з числами виконуються з певною, попередньо заданою точністю, яка визначається розміром розрядної сітки.