

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
"КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ізоря Сікорського"  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ  
Кафедра обчислювальної техніки

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА  
з дисципліни „Комп’ютерна логіка 2. Комп’ютерна арифметика”

Виконала

Бровченко Анастасія Вікторівна

Група ІО-64

Факультет ІОТ

Залікова книжка № 6403

Керівник: Жадін В. І.

## 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 мантис 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.

## Визначення та обґрунтування 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).

$$6403_{10} = 1100100000011_2;$$

$$X = -x_7x_61x_5x_40, x_31x_2x_1 = -001000,0111$$

$$Y = +x_91x_8x_7x_6x_5, x_4x_3x_2x_1 = +110000,0011$$

## Основна частина:

### Завдання №1

$$X_{\text{пк}} = 1.001000,0111; \quad Y_{\text{пк}} = 0.110000,0011;$$

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

$X_2$ :

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

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

$Y_2$ :

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

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

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

$$E_x = 127 + 4 = 131_{10} = 10000011_2$$

$X_2$

1	1	0	0	0	0	0	1	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Порядок										Мантиса																						

$$E_y = 127 + 6 = 133_{10} = 10000101_2$$

$Y_2$

1	1	0	0	0	0	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Порядок										Мантиса																						

## 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:

$$\begin{aligned}
 Z &= YX = Yx_n 2^{-n} + Yx_{n-1} 2^{-n+1} \dots + Yx_1 2^{-1} = \\
 &= ((\dots((0 + Yx_n) 2^{-1} + Yx_{n-1}) 2^{-1} + \dots + Yx_i) 2^{-1} + \dots + Yx_1) 2^{-1}; \\
 Z &= \sum_{i=1}^n (Z_{i-1} + Yx_{n-i+1}) 2^{-1};
 \end{aligned}$$

#### 2.1.2 Операційна схема:

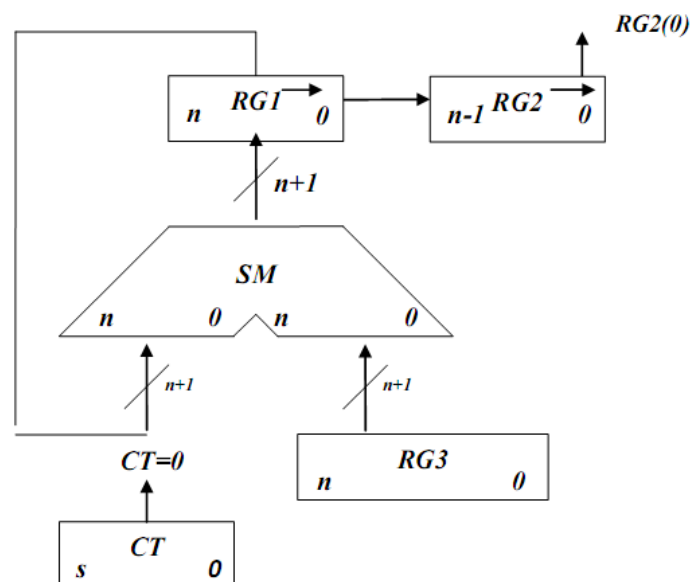


Рисунок 2.1.1- Операційна схема

### 2.1.3 Змістовний мікроалгоритм:

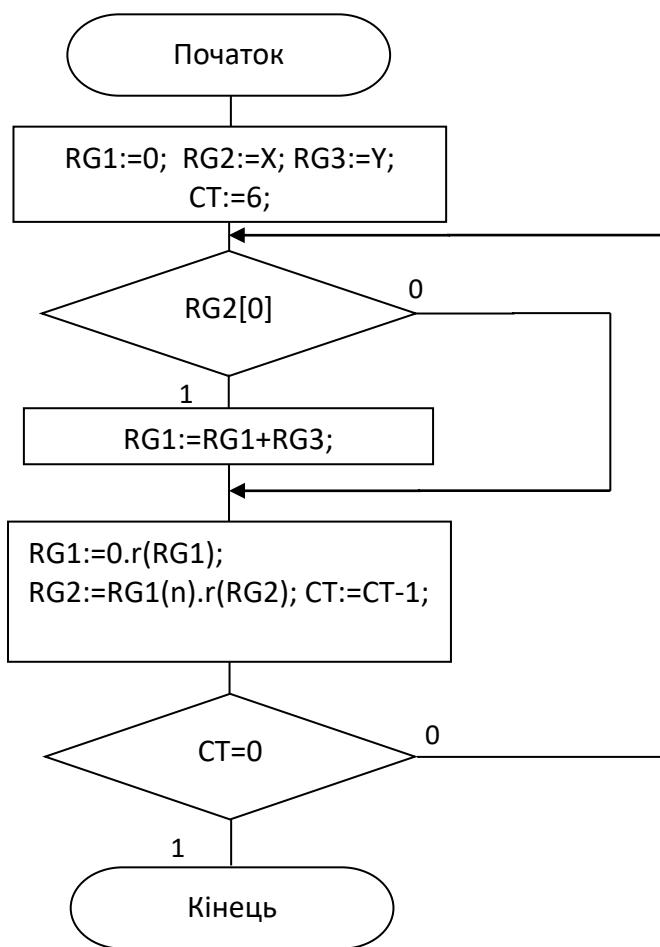


Рисунок 2.1.2 – Змістовний мікроалгоритм виконання операції множення першим способом

### 2.1.4 Таблиця станів регістрів:

Таблиця 2.1.1 – Таблиця станів регістрів

№	RG1	RG2	RG3	CT
пс	0	100001	0110000	110
1	0011000	010000		101
2	0001100	001000		100
3	0000110	000100		011
4	0000011	000010		010
5	0000001	100001		001
6	+ 0110000 = 0110001 0011000	110000		000

### 2.1.5 Обробка порядків:

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

$$P_x=4; \quad P_y=6; \quad P_z=10_{10}=1011_2$$

## 2.1.6 Нормалізація результату:

Отримали результат: 0011000110000

Знак мантиси:  $1 \oplus 0 = 1$ .

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

Порядок зменшуємо на 2:

11000110000;  $P_z = 8_2 = 1000_2$ ;

Запишемо нормалізований результат:

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

## 2.2 Другий спосіб множення.

### 2.2.1 Теоретичне обґрунтування другого способу множення:

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

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

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

$$Z = \sum_{i=1}^n Z_{i-1} + YX_{n-i+1} 2^{-n+i-1}; \quad Z_0 = 0; \quad Y_0 = 0$$

### 2.2.2 Операційна схема:

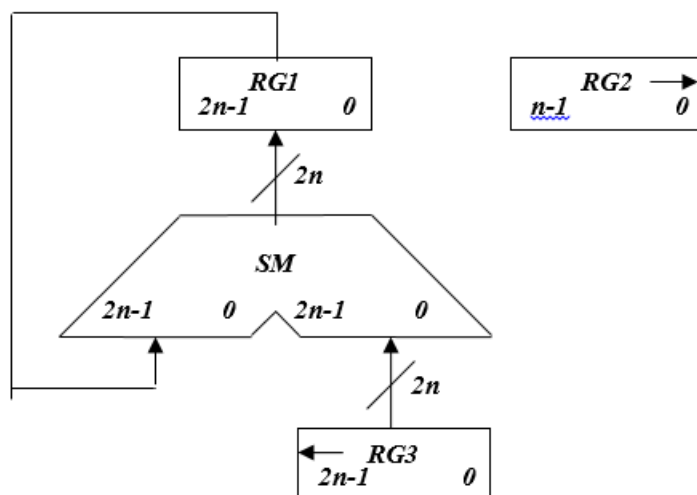


Рисунок 2.2.1 - Операційна схема

### 2.2.3 Змістовний мікроалгоритм:

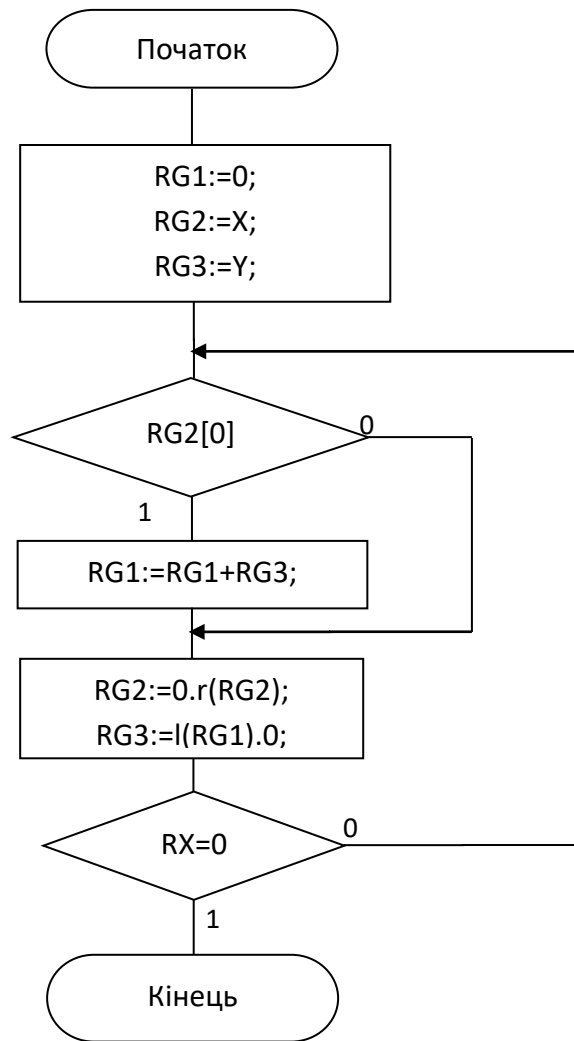


Рисунок 2.2.2 – Змістовний мікроалгоритм.

### 2.2.4 Таблиця станів регістрів:

Таблиця 2.2.1 – Таблиця станів регістрів

№	RG1	RG3 ←	RG2 →
ПС	0	000000110000	110001
1	000000110000	000001100000	011000
2	000000110000	000011000000	001100
3	000000110000	000110000000	000110
4	000000110000	001100000000	000011
5	+ 001100000000 = 001100110000	011000000000	000001
6	+ 011000000000 = 100100110000	110000000000	000000

### 2.2.5 Обробка порядків:

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

$$P_x=8; \quad P_y=6; \quad P_z=14_{10}=1110_2$$

### 2.2.6 Нормалізація результату:

Отримали результат: 100100110000

Знак мантиси:  $1 \oplus 0 = 1$ .

Нормалізація мантиси не потрібна.

Запишемо нормалізований результат:

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

## 2.3 Третій спосіб множення.

### 2.3.1 Теоретичне обґрунтування третього способу множення:

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

Множення мантис третім способом здійснюється зі старших розрядів множника, сума часткових добутків і множник зсуваються вліво, а множене нерухоме.

$$Z = YX_n 2^{-n} + YX_{n-1} 2^{-n+1} \dots + YX_1 2^{-1}; \quad Z_0 = 0; \quad Y_0 = 0;$$

$$Z = YX_n 2^{-n} + 2(YX_{n-1} 2^{-n} + 2(YX_{n-2} 2^{-n} \dots + 2YX_1 2^{-n})); \quad Z = \sum_{i=1}^n 2Z_{i-1} + YX_i 2^{-n};$$

### 2.3.2 Операційна схема:

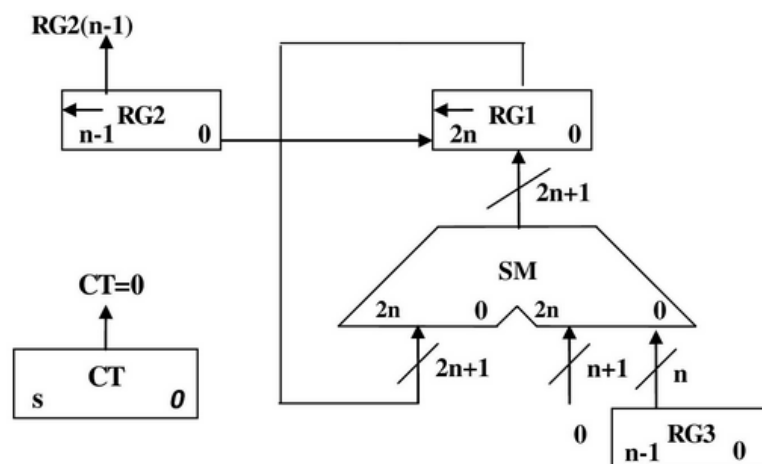


Рисунок 2.3.1 - Операційна схема



### 2.3.3 Змістовний мікроалгоритм:

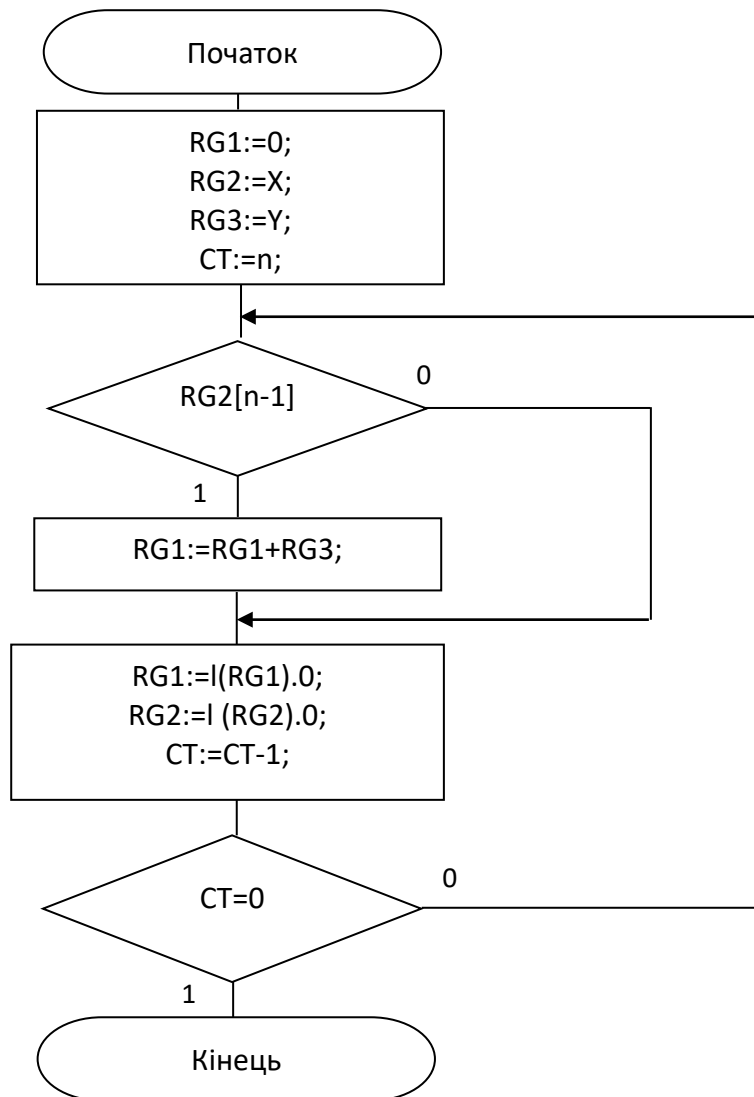


Рисунок 2.3.2 - Змістовний мікроалгоритм

### 2.3.4 Таблиця станів регістрів:

Таблиця 2.3.1- Таблиця станів регістрів

№	RG1 ←	RG2 ←	RG3	CT
пс	0	100100	110000	110
1	000001100000	001000		101
2	000011000000	010000		100
3	000110000000	100000		011
4	+ 000000110000 = 000110110000 001101100000	000000		010
5	011011000000	000000		001
6	110110000000	000000		000

### 2.3.5 Обробка порядків:

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

$$P_x=14; \quad P_y=6; \quad P_z=20_{10}=10100_2$$

### 2.3.6 Нормалізація результату:

Отримали результат: 110110000000

Знак мантиси:  $1 \oplus 0 = 1$ .

Нормалізація мантиси не потрібна.

Запишемо нормалізований результат:

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

## 2.4 Четвертий спосіб множення.

### 2.4.1 Теоритичне обґрунтування четвертого способу множення:

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

Множення здійснюється зі старших розрядів множника, сума часткових добутків залишається нерухомою, множене зсувається праворуч, множник ліворуч.

$$Z = Y \cdot x_n \cdot 2^{-n} + Y \cdot x_{n-1} \cdot 2^{-n+1} + \dots + Y \cdot x_1 \cdot 2^{-1}.$$

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

$$Z_i = Z_{i-1} + 2^{-1}Y_{i-1} \cdot x_i \quad \text{з початковими значеннями } i=1, Y_0=2^{-1}Y, Z_0=0.$$

### 2.4.2 Операційна схема:

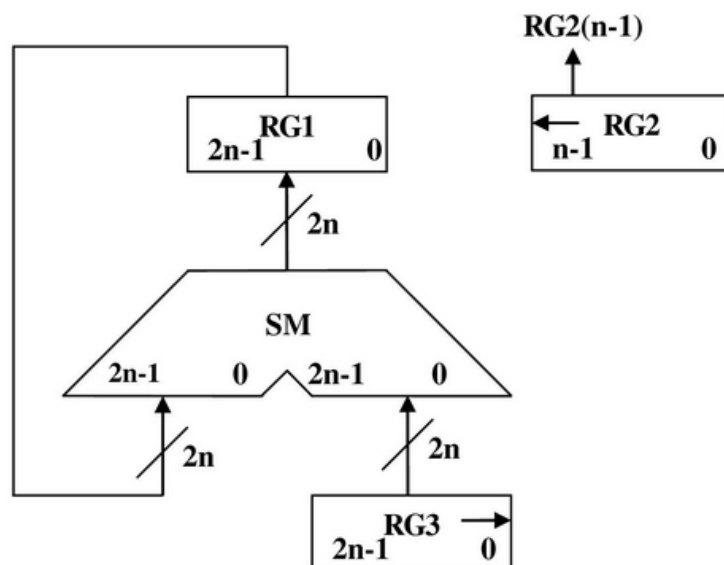


Рисунок 2.4.1- Операційна схема

### 2.4.3 Змістовний мікроалгоритм:

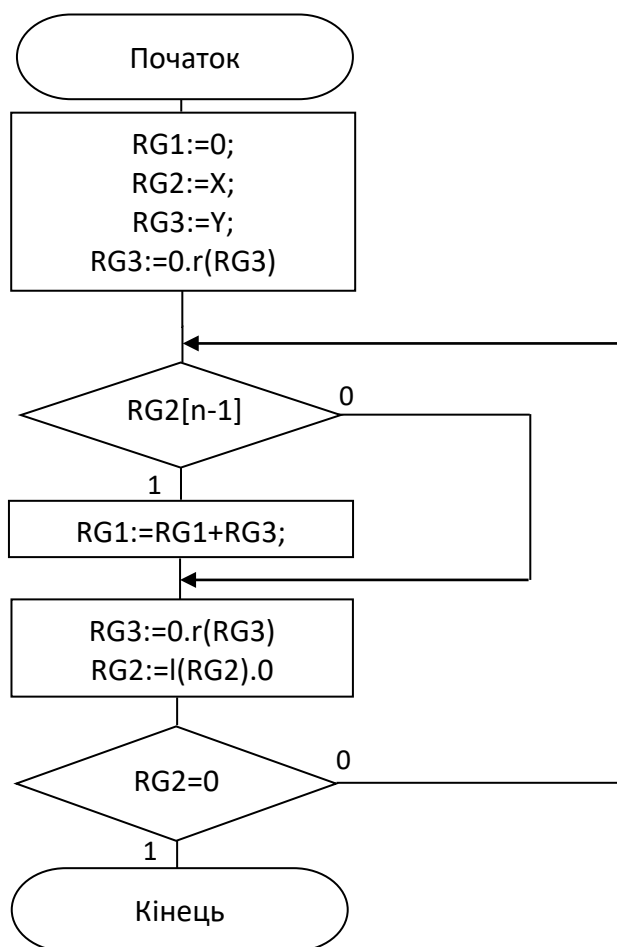


Рисунок 2.4.2 – Змістовний мікроалгоритм

### 2.4.4 Таблиця станів регістрів:

Таблиця 2.4.1– Таблиця станів регістрів

№	RG1	RG3 →	RG2 ←
ПС	0	011000000000	110110
1	011000000000	001100000000	101100
2	+ 001100000000 = 100100000000	000110000000	011000
3	100100000000	000011000000	110000
4	+ 000011000000 = 100111000000	000001100000	100000
5	+ 000001100000 = 101000100000	000000110000	000000
6	101000100000	000000011000	000000

## 2.4.5 Обробка порядків:

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

$$P_x=20; \quad P_y=6; \quad P_z=26_{10}=11010_2$$

## 2.4.6 Нормалізація результату:

Отримали результат: 101000100000

Знак мантиси:  $1 \oplus 0 = 1$ .

Нормалізація мантиси не потрібна.

Запишемо нормалізований результат:

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

## 2.5. Перший спосіб ділення.

### 2.5.1 Теоритичне 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. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний  $X$ ). 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.

### 2.5.2 Операційна схема:

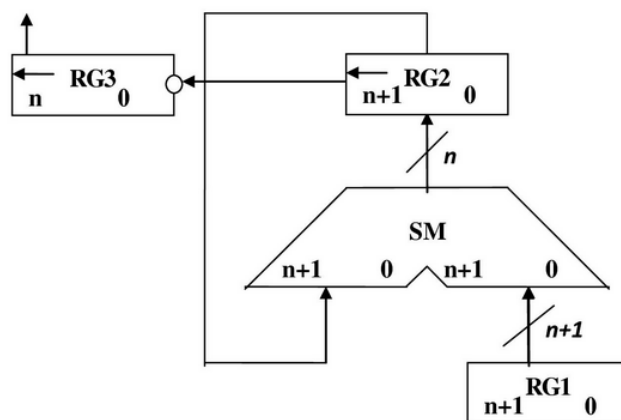


Рисунок 2.5.1- Операційна схема

### 2.5.3 Змістовний мікроалгоритм:

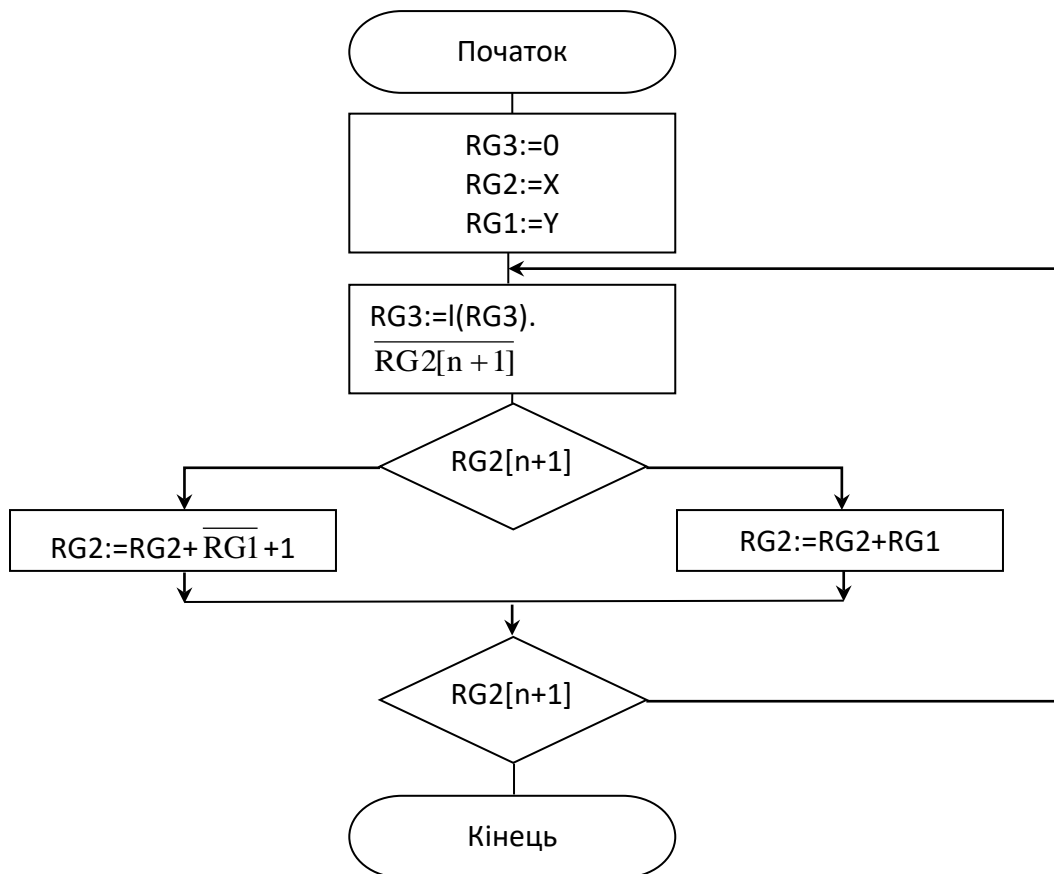


Рисунок 2.5.2-Змістовний мікроалгоритм

### 2.5.4 Таблиця станів регістрів:

Таблиця 2.5.1- Таблиця станів регістрів

№	RG3(Z)	RG2(X)	RG1(Y)
пс	000000	00101000	110000
1		01010000 + 11010000 = 00100000	
2	0000001	01000000 + 11010000 = 00010000	
3		00100000 + 11010000 = 11110000	
4	0000111	11100000 + 00110000 = 00010000	
	0001110		

5	0011101	00100000 + 11010000 = 11110000	
6	0111010	11100000 + 00110000 = 00010000	
7	1110101	00100000 + 11010000 = 11110000	

### 2.5.5 Обробка порядків:

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

В моєму випадку  $P_x=26$ ;  $P_y=6$ ;  $P_z=20_{10}=10100_2$ ;

### 2.5.6 Нормалізація результату:

Отримали результат: 11110000

Знак мантиси:  $1 \oplus 0 = 1$ .

Нормалізація мантиси не потрібна.

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

## 2.6. Другий спосіб ділення.

### 2.6.1 Теоритичне 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 добутоків 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

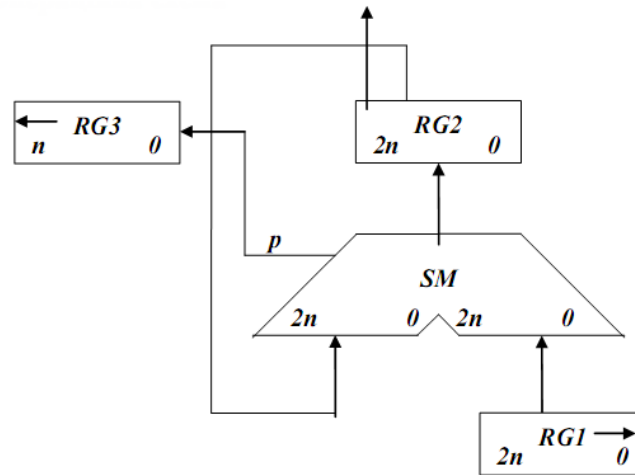


Рисунок 2.6.1-Операційна схема

## 2.6.3 Змістовний мікроалгоритм

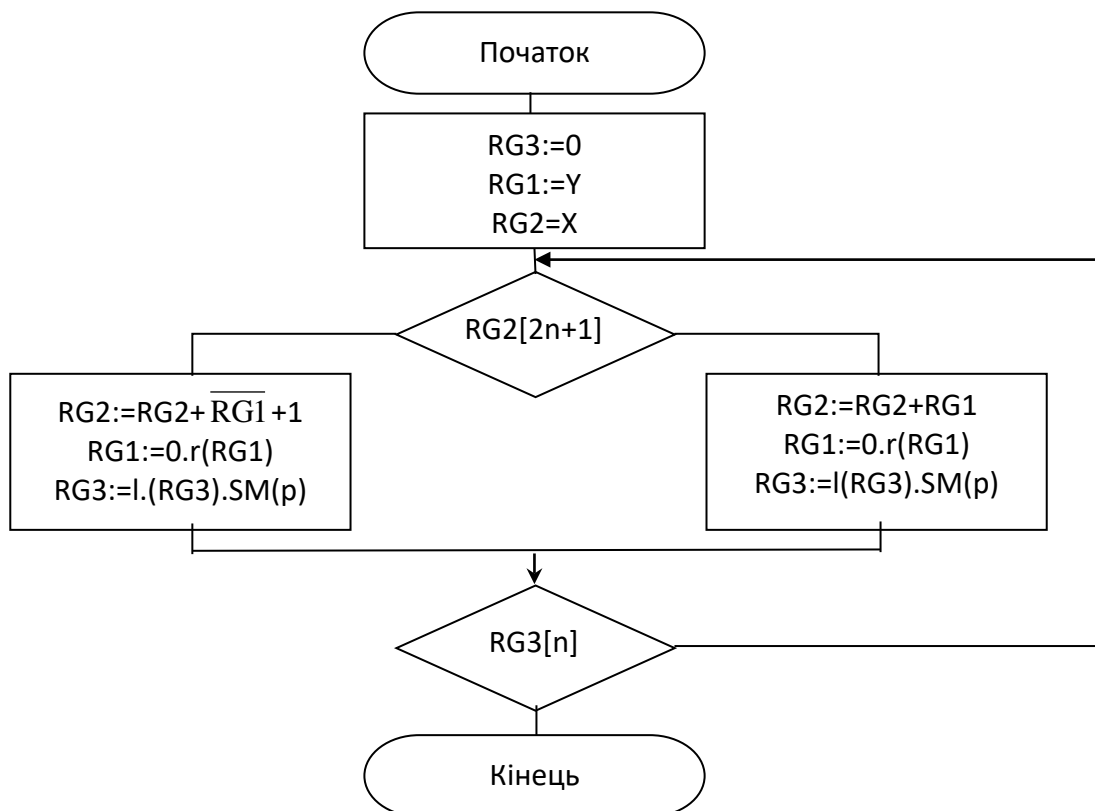


Рисунок 2.6.2-Змістовний мікроалгоритм

## 2.6.4 Таблиця станів регістрів

Таблиця 2.6.1- Таблиця станів регістрів

№	RG3(Z)	RG2(X)	RG1(Y)
ПС	000000	001111000000	011000000000
1			011000000000
			+
			110001000000
			=
	000001	000111100000	001001000000

2			001001000000 + 111000100000 = 000001100000
	000011	000011110000	
3			000001100000 + 111100010000 = 111101110000
	000110	000001111000	
4			111101110000 + 000001111000 = 111111010000
	001100	000000111100	
5			111111010000 + 000000111100 = 000000100100
	011001	000000011110	
6			000000100100 + 111111000010 = 000000000110
	110011	000000001111	

### 2.6.5 Обробка порядків:

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

В моєму випадку  $P_x=20$ ;  $P_y=6$ ;  $P_z=14$ ;

### 2.6.6 Нормалізація результату:

Отримали результат: 110011

Знак мантиси:  $1 \oplus 0 = 1$ .

Нормалізація мантиси не потрібна.

0	1	1	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 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.

1. Порівняння порядків.

$$P_x = +14_{10} = +1110_2$$

$$P_y = +6_{10} = +0110_2$$

$$P_x > P_y \Rightarrow \Delta = P_x - P_y = 14_{10} - 6_{10} = 8_{10} = 1000_2$$

2. Вирівнювання порядків.

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

Таблиця 2.7.1- Таблиця зсуву мантиси на етапі вирівнювання порядків

$M_Y$	$\Delta$	Мікрооперація
0,110000	1000	Початковий стан
0,011000	111	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,001100	110	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000110	101	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000011	100	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000001	11	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000000	10	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000000	01	$M_y = 0_r(M_y); \Delta := \Delta - 1$
0,000000	00	$M_y = 0_r(M_y); \Delta := \Delta - 1$

3. Додавання мантис у модифікованому ДК.

$$X_{\text{мдк}} = 11.001101$$

$$Y_{\text{мдк}} = 00.000000$$

Таблиця 2.7.2-Додавання мантис(для додавання)

$M_X$	1	1.	0	0	1	1	0	1
$M_Y$	0	0.	0	0	0	0	0	0
$M_Z$	1	1.	0	0	1	1	0	1

$$Z_{\text{пк}} = 1.110011$$

4. Нормалізація результату (В ПК).

Для даного результату додавання нормалізація не потрібна.

## 2.7.2 Операційна схема

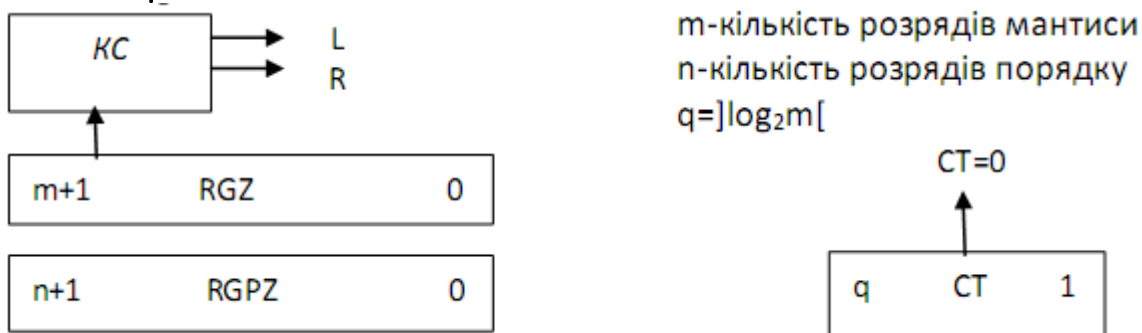


Рисунок 2.7.1-Операційна схема

Виконаємо синтез КС для визначення порушення нормалізації.

Таблиця 2.7.4-Визначення порушення нормалізації

Розряди регістру RGZ			Значення функцій	
$Z'_0$	$Z_0$	$Z_1$	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 Таблиця станів регістрів

Додавання

Таблиця 2.7.5- Таблиця станів регістрів

№ такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001001	$\begin{array}{r} 11,001101 \\ + \\ 00,000000 \\ = \\ 11,001101 \end{array}$	1	0	110	

### 2.7.4 Обробка порядків

$$P_{X+Y} = 14_{10} = 1110_2$$

### 2.7.5 Форма запису результату з плаваючою комою

Результат додавання  $Z = X + Y$ .

$$Z_{\text{пк}} = 1.110011$$

$$P_z = 14_{10} = 1110_2$$

$$M_z = 110011_2$$

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

## 2.7.6 Змістовний алгоритм

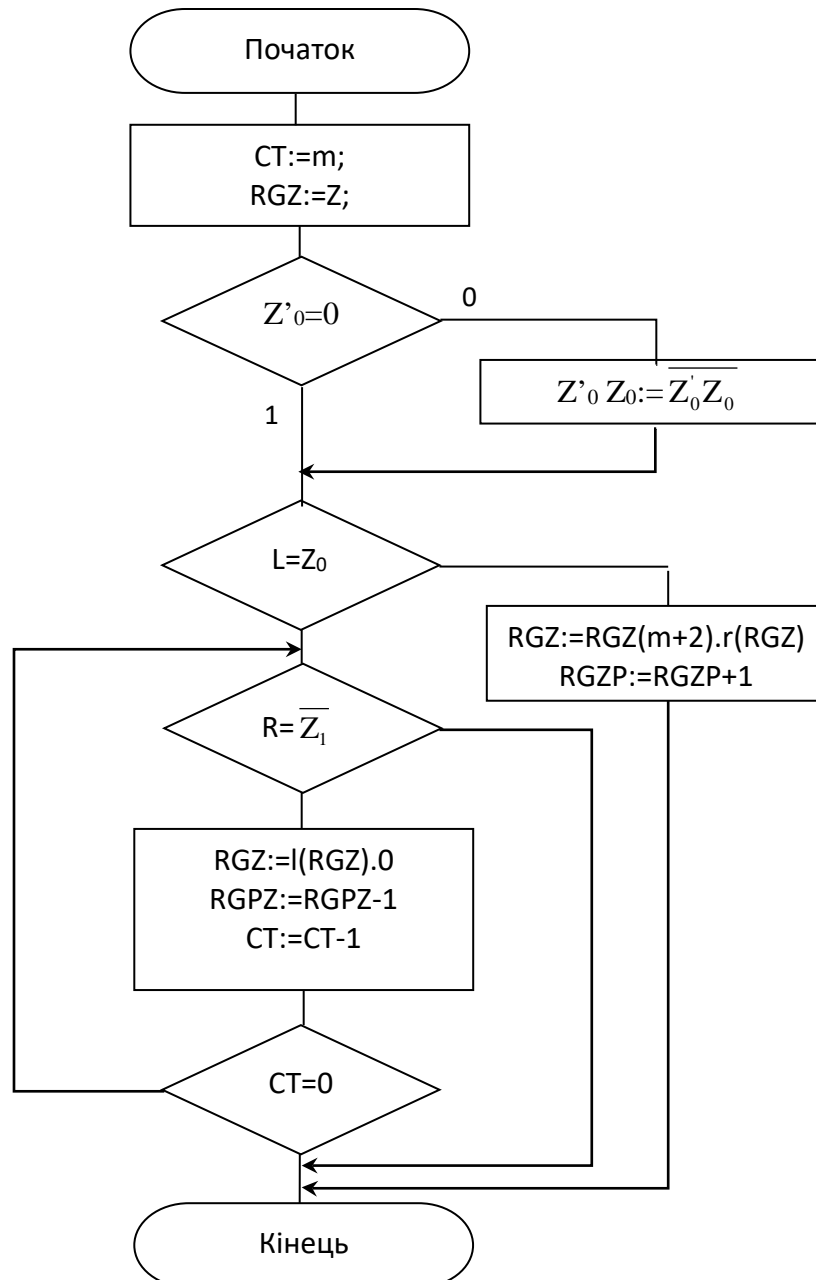


Рисунок 2.7.2-Змістовний мікроалгоритм

## 2.8.Операція добування кореня

### 2.8.1Теоритичне обґрунтування операції обчислення квадратного кореня

Аргумент вводитьься зі старших розрядів. Порядок результату дорівнює поділеному на 2 порядку аргумента. З мантиси добувається корінь завдяки нерівностям:

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

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

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

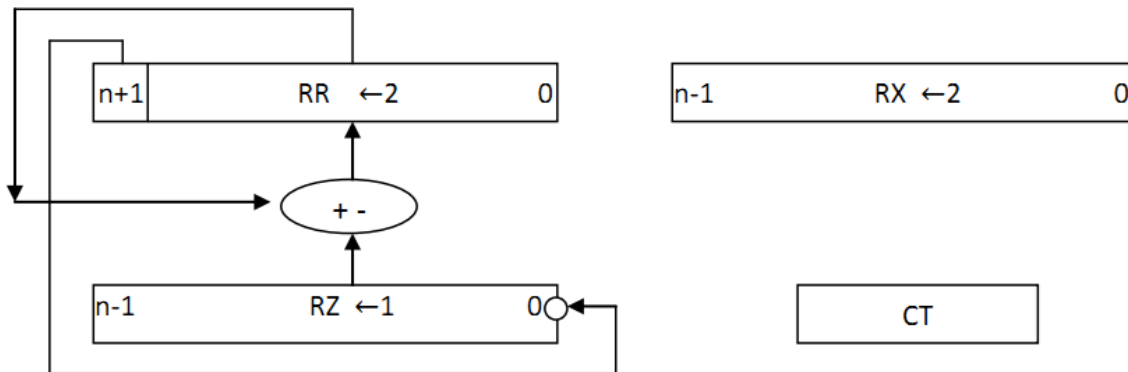


Рисунок 2.8.1 – Операційна схема

## 2.8.3 Змістовний мікроалгоритм

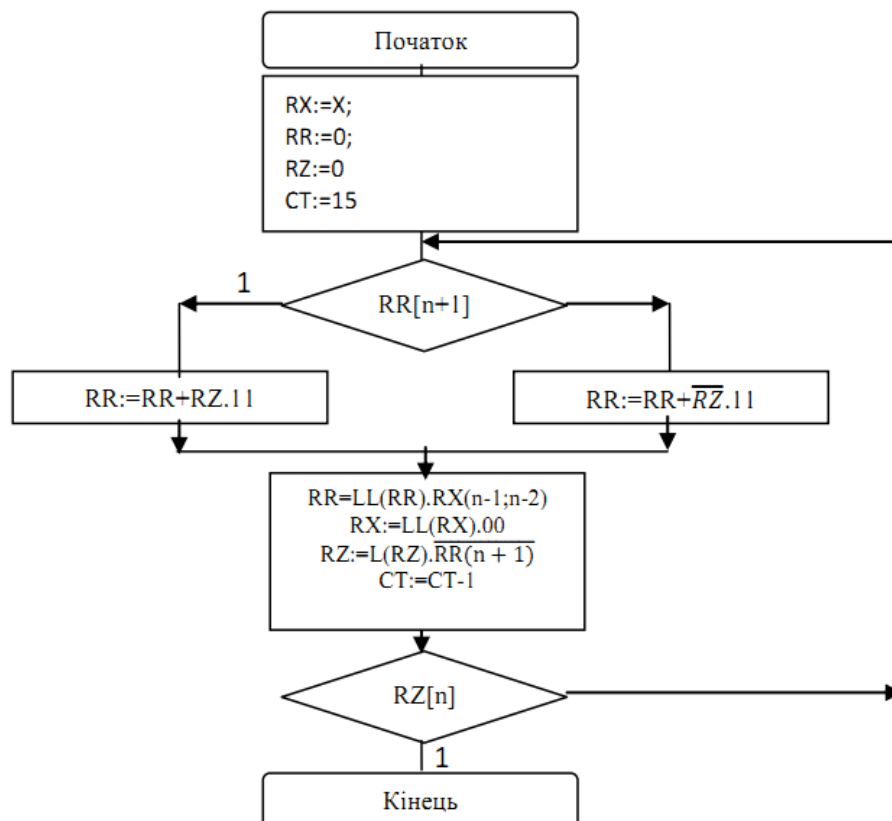


Рисунок 2.8.2 – Змістовний мікроалгоритм

## 2.8.4 Таблиця станів регістрів

Таблиця 2.8.1 – Таблиця станів регістрів

№	RZ	RR	RX	CT
ПС		00000000		
ПЗ	000000	00000011	110011	110
1		00000011 + 11111111 = 00000010 00001000		
	000001		011001	101
2		00001000 + 11111011 = 00000011 00001111		
	000011		001100	100
3		00001111 + 11110011 = 00000010 00001000		
	000111		000110	011
4		00001000 + 11100011 = 11101011 10101100		
	001110		000011	010
5		10101100 + 00111011 = 11100111 10011100		
	011100		000001	001
6		10011100 + 01110011 = 00001111 00111100		
	111001		000000	000

## 2.8.8 Обробка порядків

$$P_z = P_x/2;$$

В моєму випадку  $P_z=7$ ;

## 2.8.9 Запис результату

Отримали результат  $Z = 111001$ ;

Результат нормалізований, готовий до запису у мантису:

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

### Завдання 3

$x_3 x_2 x_1 = 011_2 = 3_{10}$  — непарне.

Синтез управляючого автомату Мілі на RS-тригерах для операції множення 4 способом.

#### 3.1 Таблиця кодування сигналів

##### 3.1.1 Функціональна схема:

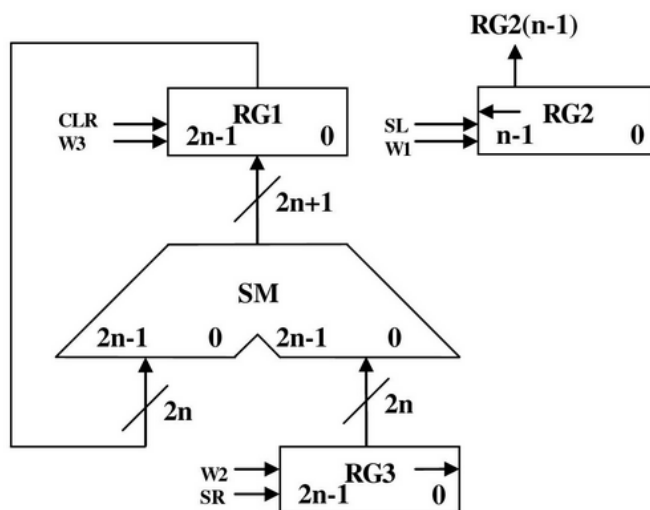


Рисунок 3.1.1 – Функціональна схема.

##### 3.1.2 Закодований мікроалгоритм

Таблиця 3.1.1 – Таблиця кодування операцій і логічних умов

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

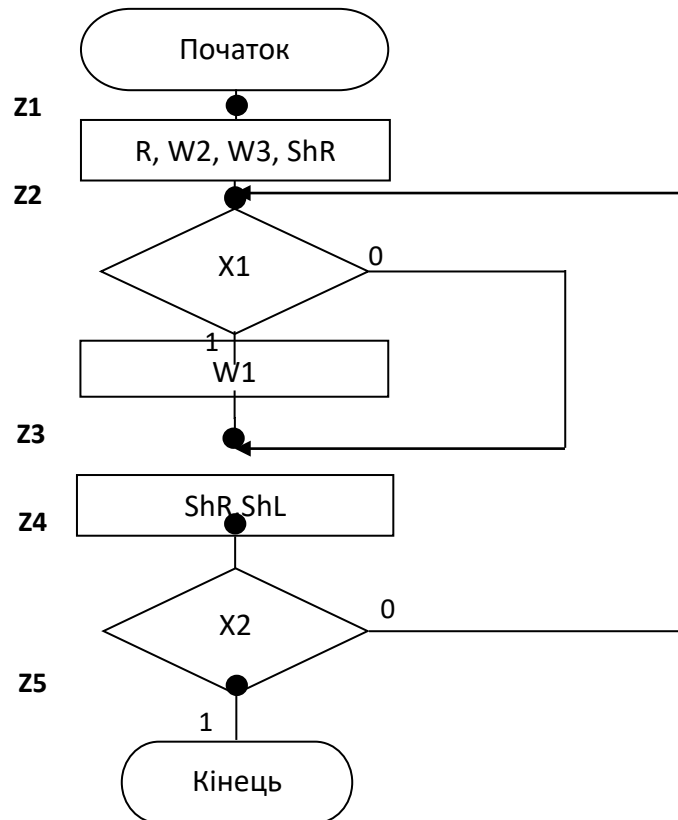


Рисунок 3.1.3 -Закодований мікроалгоритм.

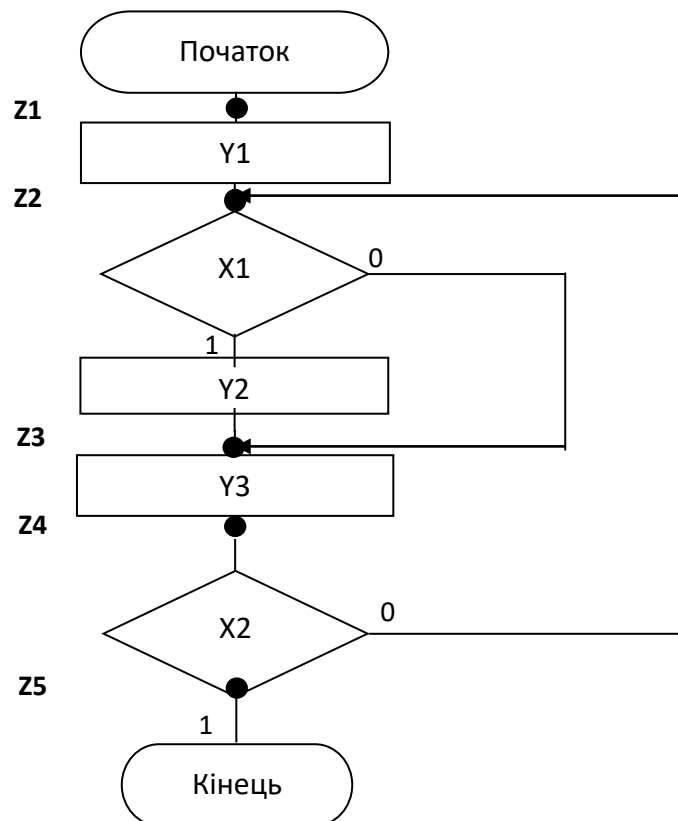


Рисунок 3.1.3 -Закодований мікроалгоритм для автомата.

### 3.1.3 Граф управляючого автомата Мілі з кодами вершин:

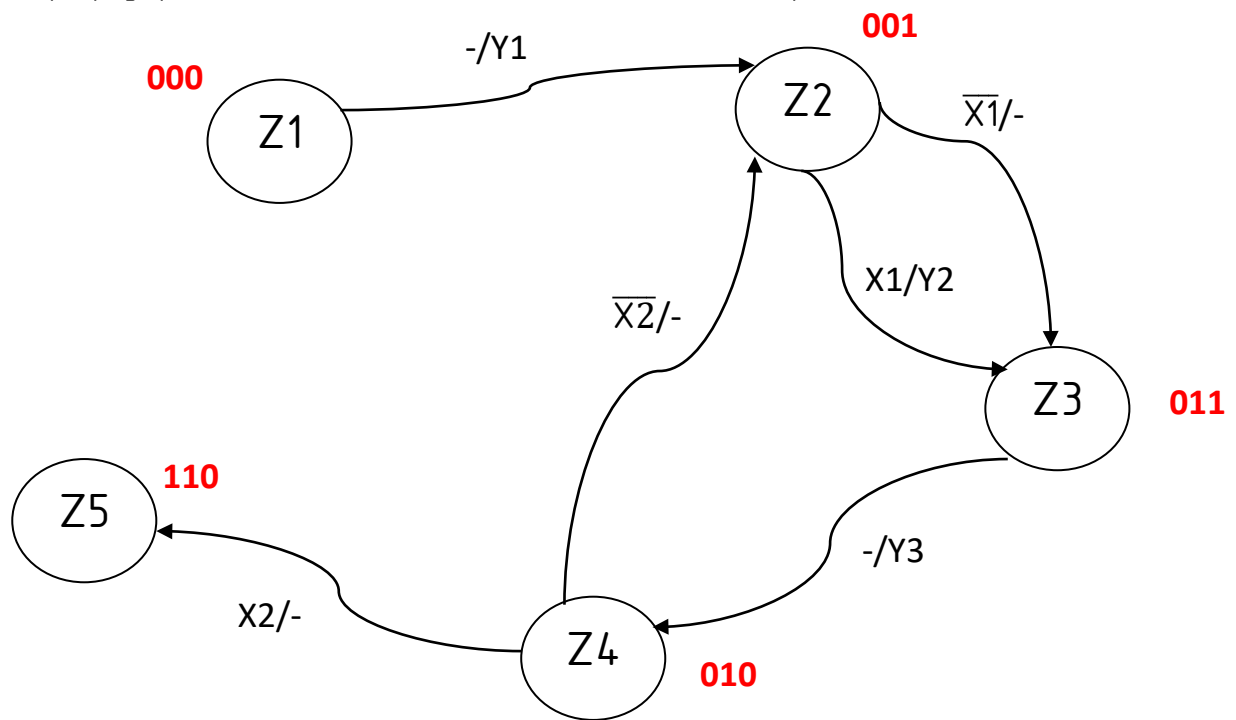


Рисунок 3.2 – Граф автомата

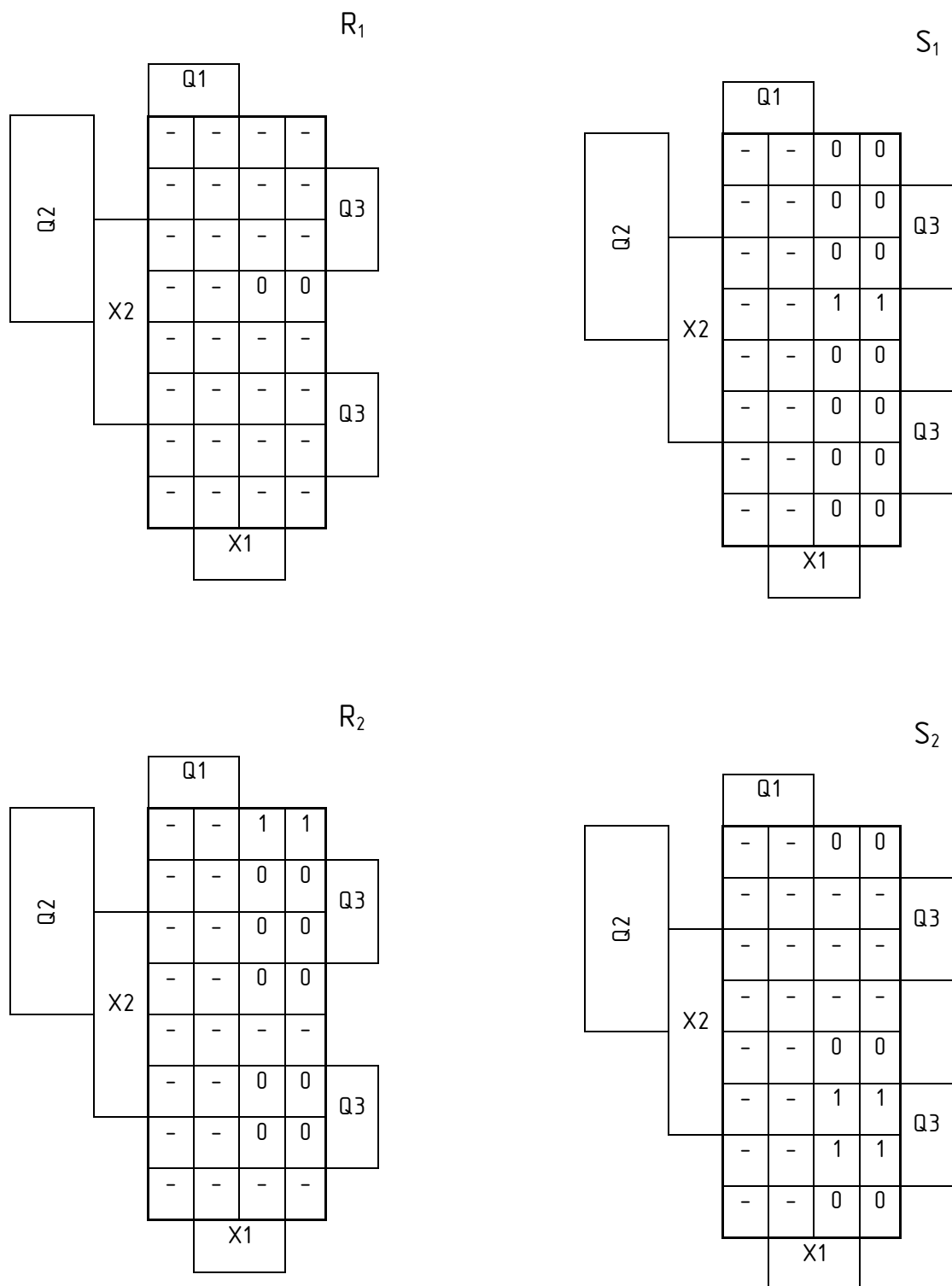
### 3.4 Таблиця переходів циклічного автомата на RS-тригерах

Таблиця 3.2 – Таблиця переходів

Пер.	Ст. ст.	Нов. стан	Вх. сузн.	Вих. сузн.	Функції тригерів		
	$Q_3Q_2Q_1$	$Q_3Q_2Q_1$	$X_2X_1$	$Y_1Y_2Y_3$	$R_1S_1$	$R_2S_2$	$R_3S_3$
$Z_1 \rightarrow Z_2$	000	001	--	100	-0	-0	01
$Z_2 \rightarrow Z_3$	001	011	0-	000	-0	01	0-
$Z_2 \rightarrow Z_3$	001	011	1-	010	-0	01	0-
$Z_3 \rightarrow Z_4$	011	010	--	001	-0	0-	10
$Z_4 \rightarrow Z_2$	010	001	-0	000	-0	10	01
$Z_4 \rightarrow Z_5$	010	110	-1	000	01	0-	-0

### 3.5 Мінімізація функцій тригерів





*Рисунок 3.3 – Мінімізація функцій тригерів*

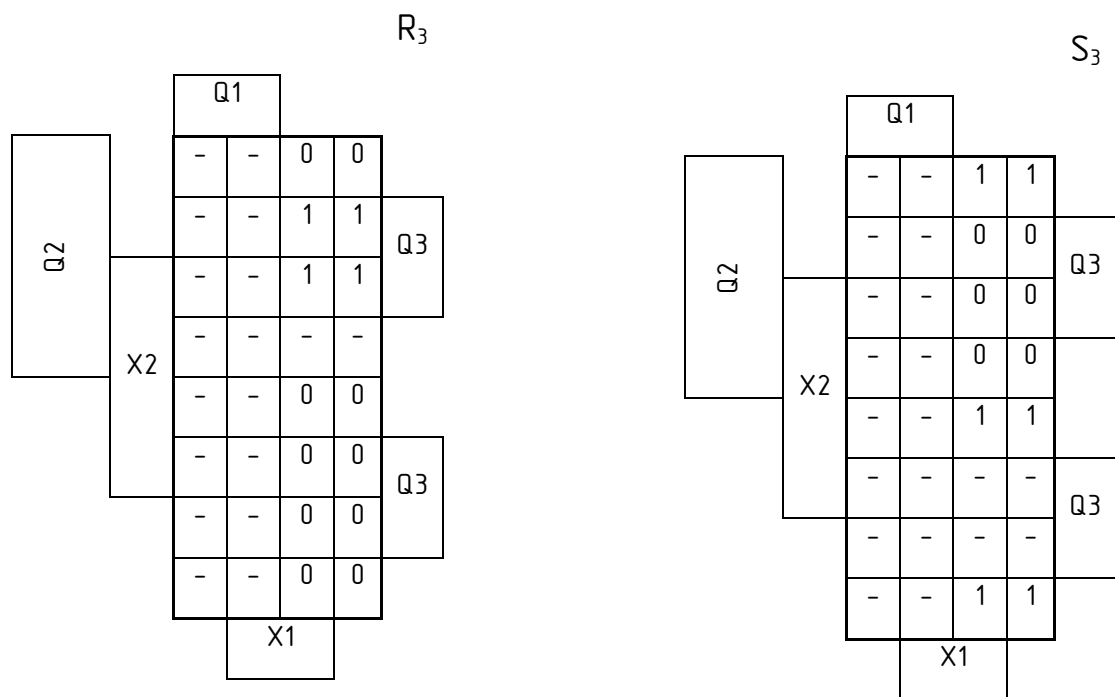


Рисунок 3.3 – Мінімізація функцій тригерів (продовження)

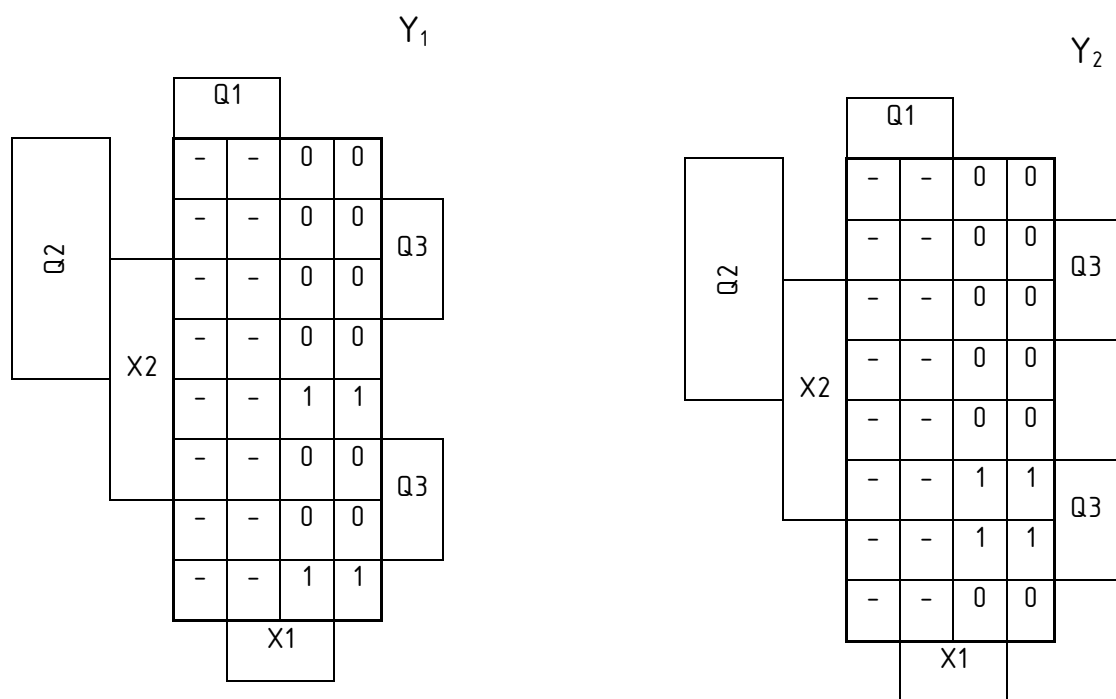


Рисунок 3.4 – Діаграми Вейча для вихідних сигналів

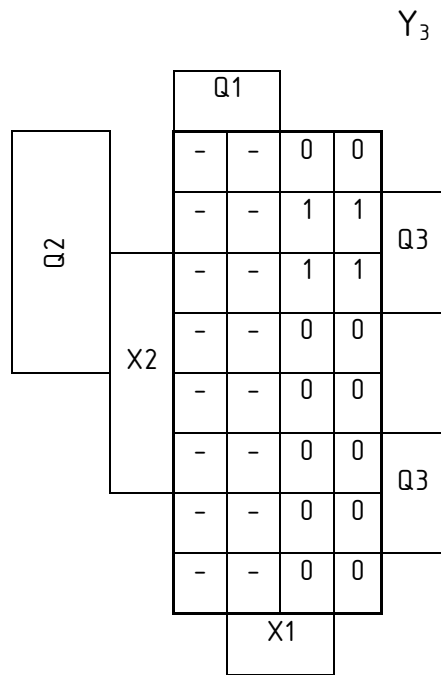


Рисунок 3.4 – Діаграми Вейча для вихідних сигналів (продовження)

$$R_1 = Q_3$$

$$R_2 = Q_2 \overline{Q_3} X_2$$

$$R_3 = Q_3 Q_2$$

$$S_1 = Q_2 \overline{Q_3} X_2$$

$$S_2 = Q_3 \overline{Q_2}$$

$$S_3 = \overline{Q_2} \vee Q_2 \overline{Q_3} X_2$$

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

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

$$Y_3 = Q_3 Q_2$$

### 3.6 Функціональна схема автомата

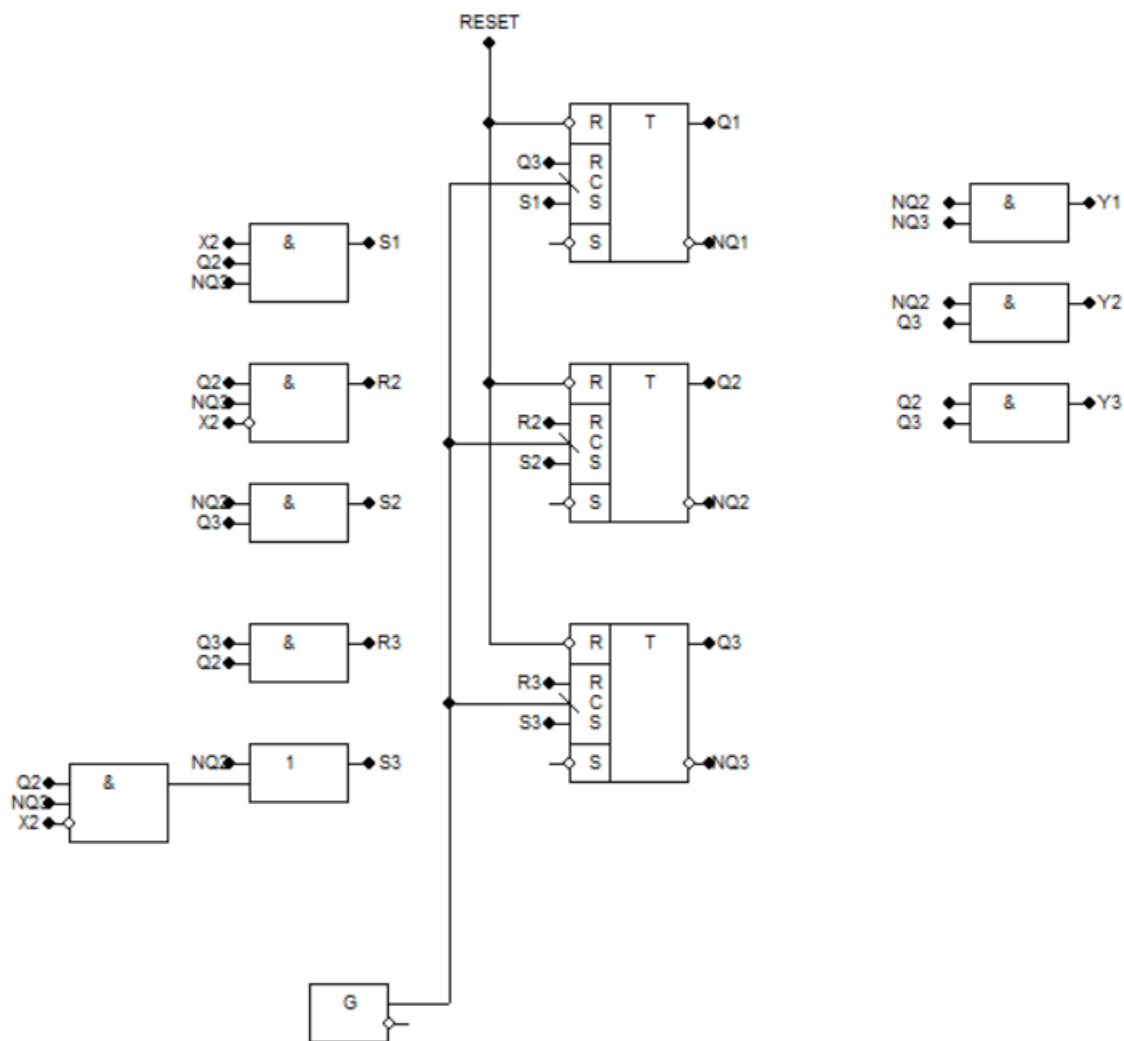


Рисунок 3.5 – Функціональна схема

## Висновок

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

Під час виконання даної розрахункової роботи я повторив для себе матеріал курсу «Компютерна логіка – 1», а також закріпив знання з курсу «Компютерна логіка – 2» (Комп'ютерна арифметика).

#### Використана література:

1	Жабін В.І., Жуков І.А., Клименко І.А.,Ткаченко В.В. Прикладна теорія цифрових автоматів: Навчальний посібник.–К.: Книжкове вид-во НАУ, 2009. – 360 с.
---	---