

Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Комп'ютерна арифметика
Розрахункова робота по курсу «КЛ-2»

Виконав: студент групи ІО-82
Матвійчук Б. В.

Керівник: Жабін В.І.

2010 р.

Обґрунтування варіанту

Номер залікової книжки: $9108_{10} = 10001110010100_2$

$$X_2 = -11111001,1001100$$

$$Y_2 = +11111,0011001001$$

Виконання роботи

Завдання 1

В прямому коді:

X_2 :

$$\text{Зн.Р} \quad P = +8_{10} \quad \text{Зн.М} \quad M = -,111110011001100$$

Y_2 :

$$\text{Зн.Р} \quad P = +5_{10} \quad \text{Зн.М} \quad M = +,111110011001001$$

Завдання 2

1. Операція множення чисел.

$Z = Y \times X$, де Y -множене, X -множник.

1.1 Перший спосіб множення.

1.1.1 Теоретичне обґрунтування.

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:

$$Z = Y \cdot x_n \cdot 2^{-n+99} + Y \cdot x_{n-1} \cdot 2^{-n+1} + \dots + Y \cdot x_1 \cdot 2^{-1}, \text{ що рівносильно}$$

$$Z = \left(\left(\dots \left((0 + Y \cdot x_n) \cdot 2^{-1} + Y \cdot x_{n-1} \right) \cdot 2^{-1} \dots \right) + Y \cdot x_1 \right) \cdot 2^{-1} = \sum_{i=1}^n ((Z_{i-1} + Y \cdot x_{n-i+1}) \cdot 2^{-1}).$$

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу:

$$Z_i = (Z_{i-1} + Y \cdot x_{n-i+1}) \cdot 2^{-1}.$$

1.1.2 Операційна схема.

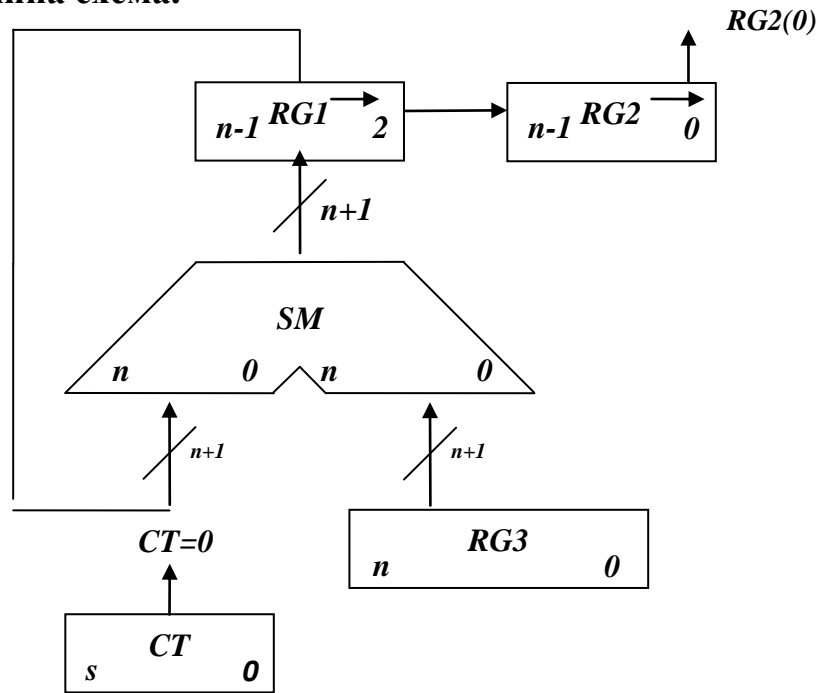


Рисунок 1.1- Операційна схема.

1.1.3 Змістовний мікроалгоритм.

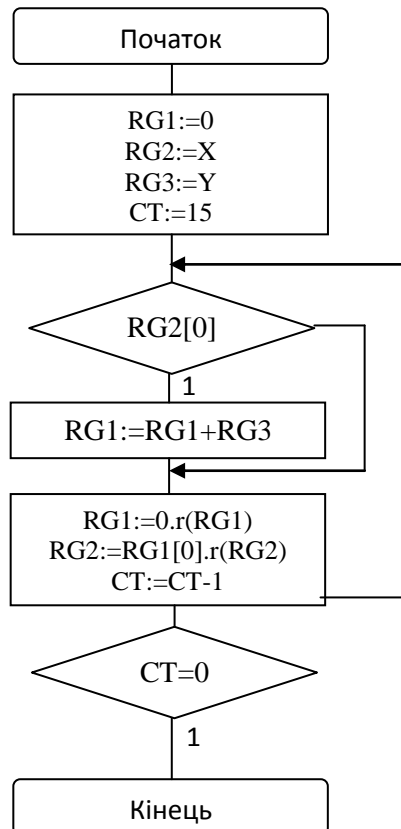


Рисунок 1.2- Змістовний мікроалгоритм.

1.1.4 Таблиця станів регістрів.

Таблиця 1.1-Таблиця станів регістрів.

№ ц.	RG1	RG2	RG3	СТ
П.С.	0000000000000000	011111001100110	0111110011001001	1111
1→	0000000000000000	001111100110011		1110
2 →	0000000000000000	001111100110011		1101
3	0011111001100100	100111110011001		1100
4→	+0111110011001001 1011101100101101 0101110110010110	110011111001100		1011
5→	0010111011001011	011001111100110		1010
6→	0001011101100101	101100111110011		1001
7 →	+0111110011001001 1001010000101110 0100101000010111	010110011111001		1000
8 →	+0111110011001001 1100011011100000 0110001101110000	001011001111100		0111
9→	0011000110111000	000101100111110		0110
10→	0001100011011100	000010110011111		0101
11→	+0111110011001001 1001010110100101 0100101011010010	100001011001111		0100
12 →	+0111110011001001 1100011110011011 0110001111001101	110000101100111		0011
13→	+0111110011001001 1110000010010110 0111000001001011	011000010110011		0010
14→	+0111110011001001 1110110100010100 0111011010001010	001100001011001		0001
15 →	+0111110011001001 1111001101010011 0,0111100110101001	100110000101100		0000

1.1.5 Функціональна схема з відображенням управляючих сигналів.

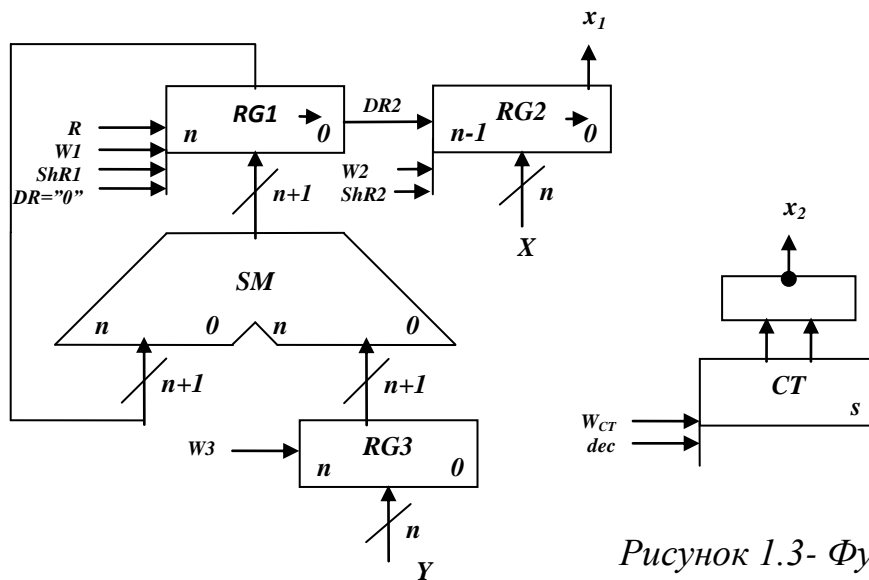


Рисунок 1.3- Функціональна схема.

1.1.6 Закодований мікроалгоритм.

Таблиця 1.2-Таблиця кодування операцій і логічних умов.

Таблиця кодування мікрооперацій		Таблиця кодування логічних умов	
МО	УС	ЛЮ	Позначення
G1:=0	R	RG2[0]	X1
RG2:=X	W2	CT=0	X2
RG3:=Y	W3		
CT:=15	W _{CT}		
RG1:=RG1+RG3	W1		
RG1:=0.r(RG1)	ShR1		
RG2:=RG1[0].r(RG2)	ShR2		
CT:=CT-1	dec		

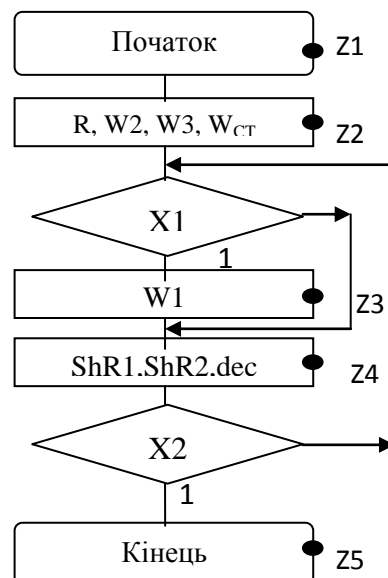


Рисунок 1.4-Закодований мікроалгоритм.

1.1.7 Граф управляючого автомата Мура з кодами вершин.

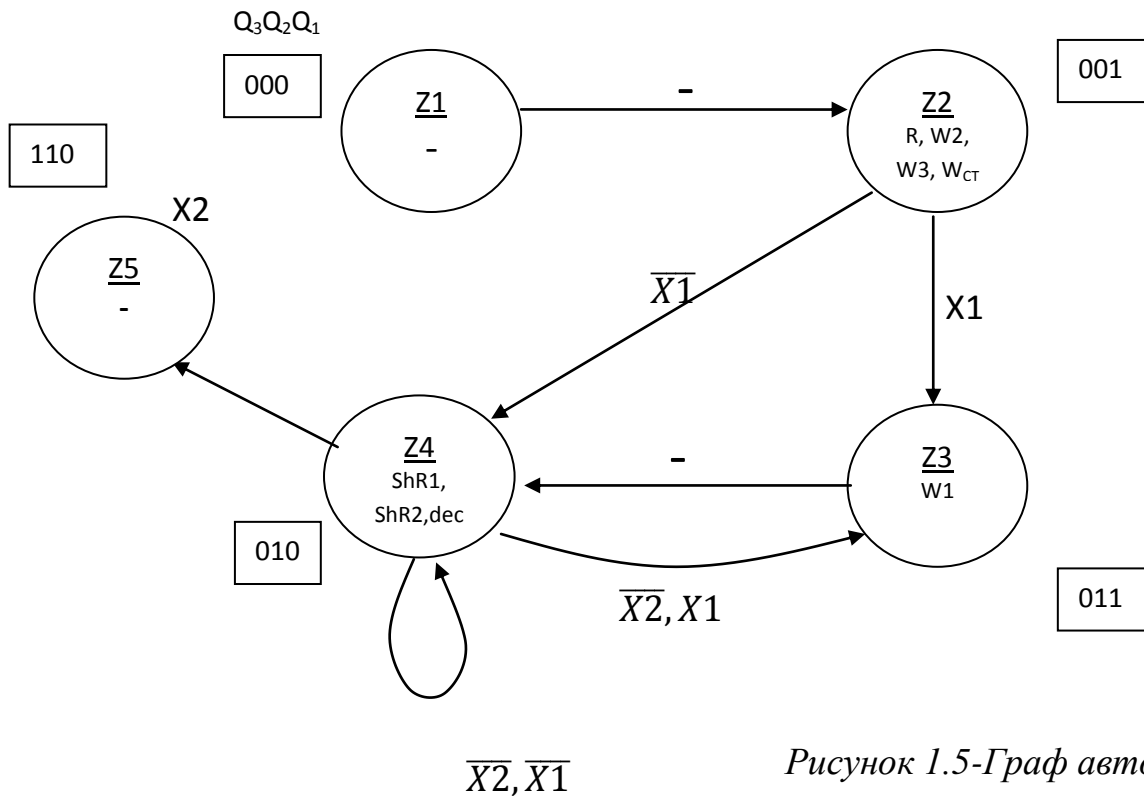


Рисунок 1.5-Граф автомата Мура

1.1.8 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 8 + 5 = 13_{10} = 1101_2.$$

Нормалізація мантиси.

$$M_z = , 0,111100110101001 \leftarrow , P_z := P_z - 1. \\ , 111100110101001 \quad P_z = 1100.$$

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

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

Зн.Р	P=+12 ₁₀	Зн.М	М
0. 0 0 0 1 1 0 0		0, 1 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1	

1.2 Другий спосіб множення

1.2.1 Теоретичне обґрунтування

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

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

Вираз

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

$$Z = \left(\left(\dots \left((0 + Y \cdot 2^{5n} \cdot x_n) + Y \cdot 2^{-n+1} \cdot x_{n-1} \right) + \dots \right) + Y \cdot 2^{-1} x_1 \right).$$

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу:

$$Z_i = Z_{i-1} + 2Y_{i-1} \cdot x_{n-i+1}.$$

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

1.2.2 Операційна схема

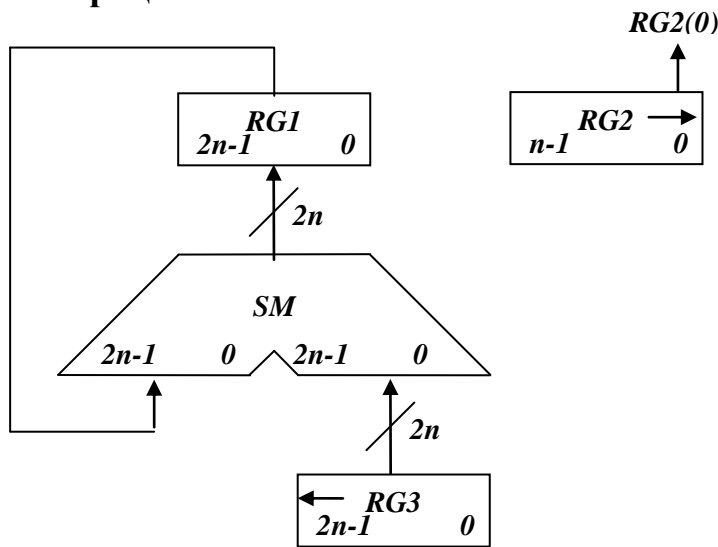
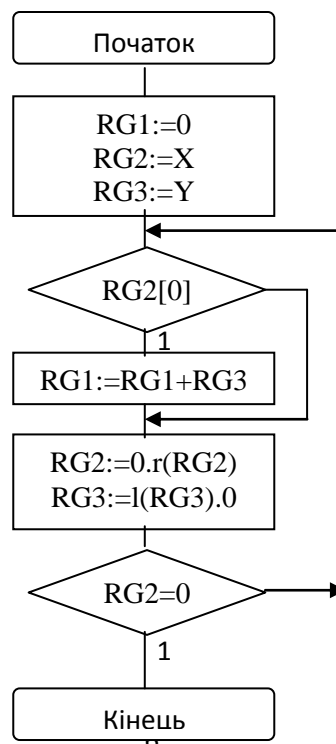


Рисунок 1.6-Операційна схема

1.2.3 Змістовний мікроалгоритм

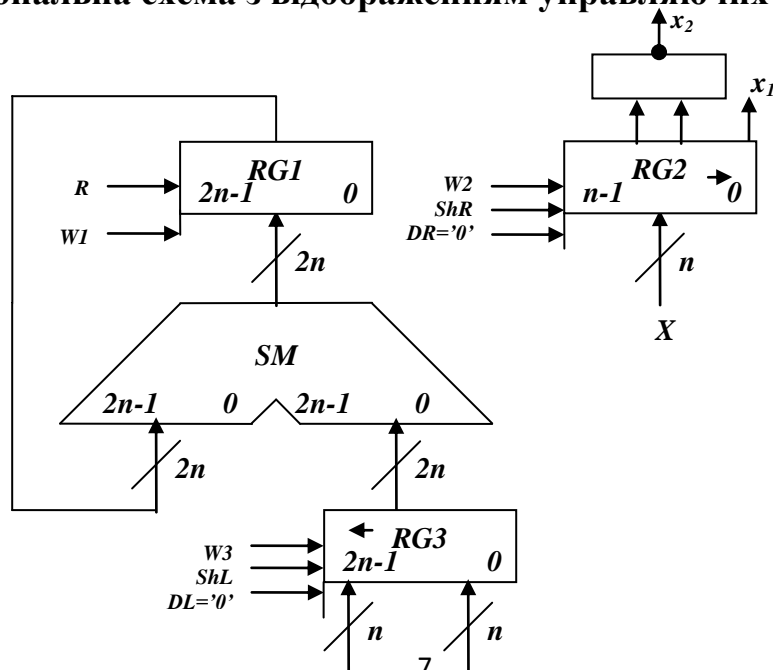


1.2.4 Таблиця станів регістрів

Таблиця 1.3- Таблиця станів регістрів

№ ц.	RG1	RG3 ←	RG2 →
П.С.	00000000000000000000000000000000	00000000000000000000111110011001001	1111100110010010
1	00000000000000000000000000000000	000000000000000000001111100110010010	011111001100110
2	00000000000000000000000000000000	0000000000000000000011111001100100100	001111100110011
3	0000000000000000000011111001100100100	00000000000000000000111110011001001000	000111110011001
4	+00000000000000000000111110011001001000 00000000000001011101100101101100	000000000000000000001111100110010010000	000011111001100
5	00000000000001011101100101101100	0000000000000000000011111001100100100000	000001111100110
6	00000000000001011101100101101100	00000000000000000000111110011001001000000	000000111110011
7	+0000000000000111110011001001000000 00000000010010100000101110101100	0000000000000000000011111001100100100000000	0000000011111001
8	+0000000000000111110011001001000000 0000000011000110111000000101100	0000000000000000000011111001100100100000000	0000000001111100
9	0000000011000110111000000101100	00000000000000000000111110011001001000000000	0000000000111110
10	0000000011000110111000000101100	000000000000000000001111100110010010000000000	0000000000011111
11	+00000011111001100100100000000000 0000100101011010001010000101100	000000000000000000001111100110010010000000000	0000000000001111
12	+00000111110011001001000000000000 0001100011110011011100000101100	00011111001100100100000000000000	0000000000000111
13	+000111110011001001000000000000 0011100000100101101100000101100	001111100110010010000000000000	0000000000000011
14	+001111100110010010000000000000 0111011010001010001100000101100	011111001100100100000000000000	0000000000000001
15	+011111001100100100000000000000 1111001101010011001100000101100	111110011001001000000000000000	0000000000000000

1.2.5 Функціональна схема з відображенням управляючих сигналів



1.2.6 Закодований мікроалгоритм

Таблиця 1.4 – Таблиця кодування мікрооперацій.

Таблиця кодування мікрооперацій	
МО	УС
RG1:=0	R
RG2:=X	W2
RG3:=Y	W3
RG1:=RG1+RG3	W1
RG2:=0.r(PG2)	ShR
RG3:=l(RG3).0	ShL

Таблиця кодування логічних умов	
ЛЮ	Позначення
RG2[0]	X1
RG2=0	X2

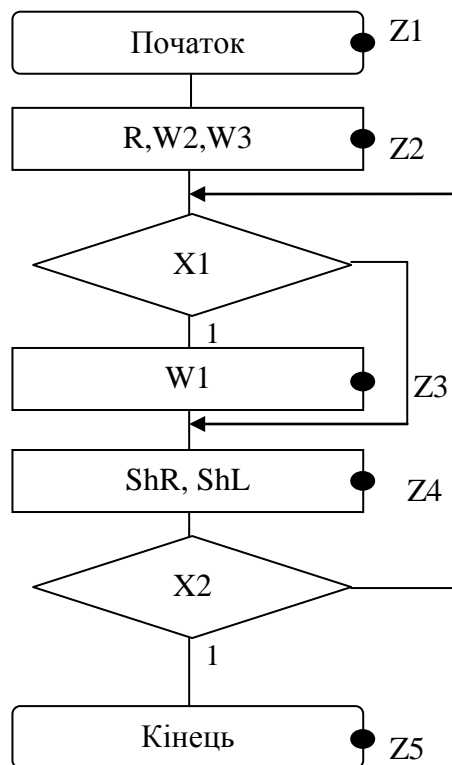


Рисунок 1.9- Закодований мікроалгоритм.

1.2.7 Граф управляющего автомата Мура з кодами вершин

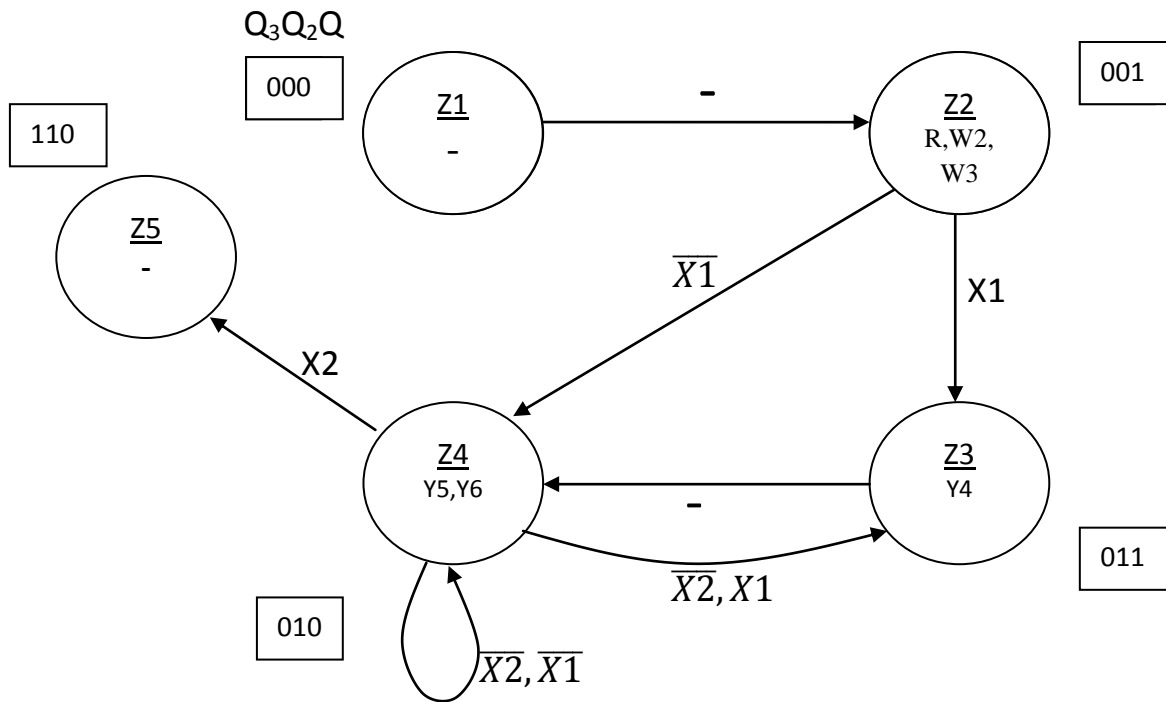


Рисунок 1.10- Граф автомата Мура

1.2.8 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 8 + 5 = 13_{10} = 1101_2$$

Нормалізація мантиси.

$$M_z = , 0,111100110101001 \leftarrow , P_z := P_z - 1. \\ , 111100110101001 \quad P_z = 1100.$$

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

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

Зн.Р	P=+12 ₁₀	Зн.М	М
0	0 0 0 1 1 0 0	0	1 1 1 1 0 0 1 1 0 1 0 1 0 0 1

1.3 Третій спосіб множення

1.3.1 Теоретичне обґрунтування

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

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

Вираз

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

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

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу:

$$Z_i = 2Z_{i-1} + Y \cdot x_i \cdot 2^{-n}.$$

з початковими умовами $Z_0=0, i=1$.

1.3.2 Операційна схема

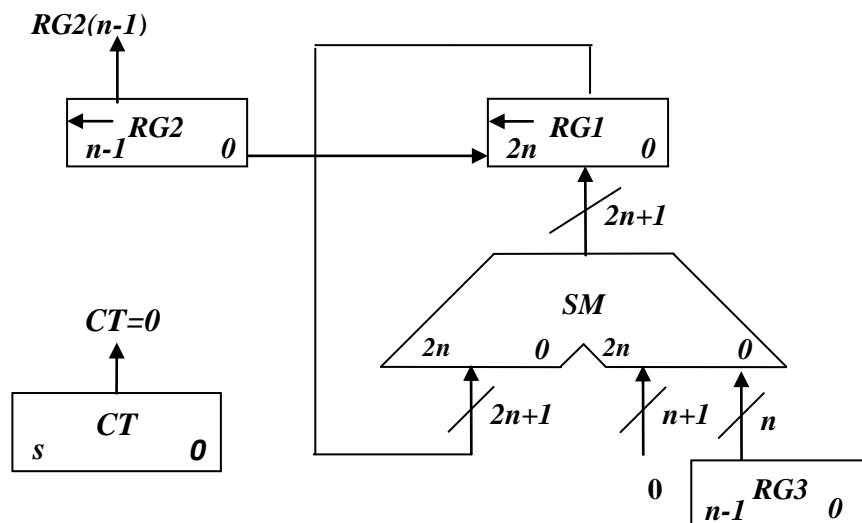


Рисунок 1.11-Операційна схема

1.3.3 Змістовний мікроалгоритм

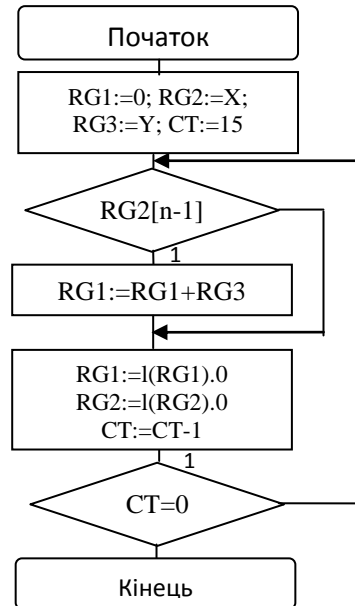


Рисунок 1.12-Змістовний мікроалгоритм

1.3.4 Таблиця станів регістрів

Таблиця 1.5- Таблиця станів регістрів

№ ц.	RG1	RG3	RG2	CT
П.С.	00000000000000000000000000000000	101000011000101	111110011001100	1111
1	00000000000000001111100110010010		111100110011000	1110
2	+0000000000000000111110011001001 0000000000000010111011001011011 0000000000000101110110010110110		111001100110000	1101
3	+0000000000000000111110011001001 0000000000000110110100101111111 0000000000001101101001011111110		110011001100000	1100
4	+0000000000000000111110011001001 0000000000001110100111111000111 0000000000011101001111110001110		100110011000000	1011
5	+0000000000000000111110011001001 0000000000011110001110001010111 0000000000111100011100010101110		001100110000000	1010
6	000000001111000111000101011100		011001100000000	1001
7	000000011110001110001010111000		110011000000000	1000
8	+0000000000000000111110011001001 000000011110010101111110000001 000000111100101011111100000010		100110000000000	0111
9	+0000000000000000111110011001001 0000000111100110011101111001011 000001111001100111011110010110		001100000000000	0110
10	000011110011001110111100101100		011000000000000	0101
11	000111100110011101111001011000		110000000000000	0100
12	+0000000000000000111110011001001 000111100110100101101100100001 001111001101001011011001000010		100000000000000	0011

13	+0000000000000000111110011001001 001111001101010011001100001011 011110011010100110011000010110		0000000000000000	0010
14	111100110101001100110000101100		0000000000000000	0001
15	111001101010011001100001011000		0000000000000000	0000

1.3.5 Функціональна схема з відображенням управляючих сигналів

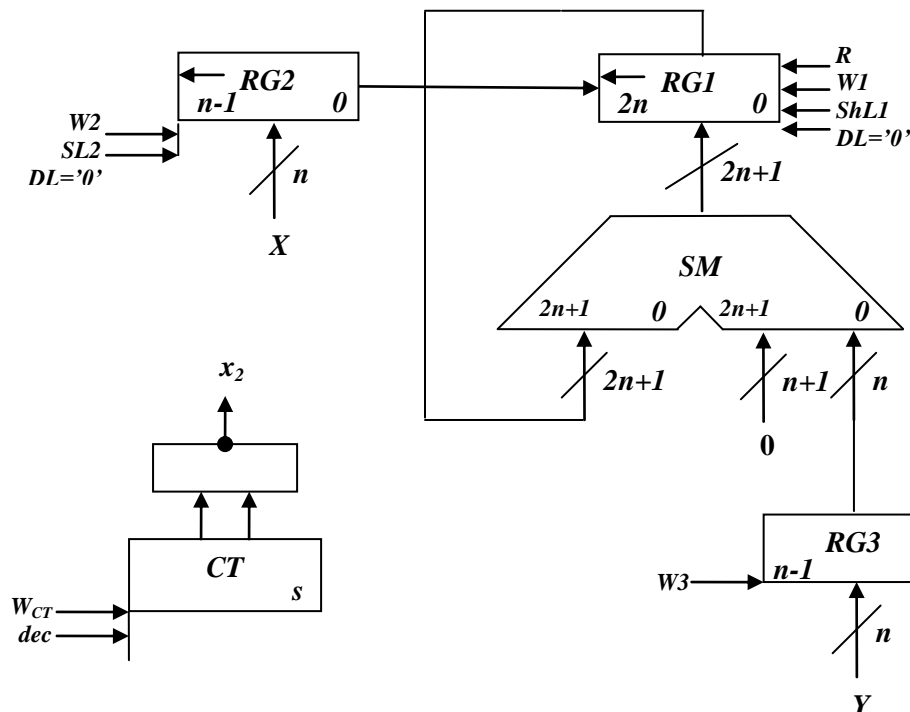


Рисунок 1.13-Функціональна схема

1.3.6 Закодований мікроалгоритм

Таблиця 1.6- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
RG1:=0	R
RG2:=X	W2
RG3:=Y	W3
CT:=15	W _{CT}
RG1:=RG1+RG3	W1
RG1:=l(RG1).0	ShL1
RG2:=l(RG2).0	ShL2
CT:=CT-1	dec

Таблиця кодування логічних умов	
ЛЮ	Позначення
RG2[n-1]	X1
CT=0	X2

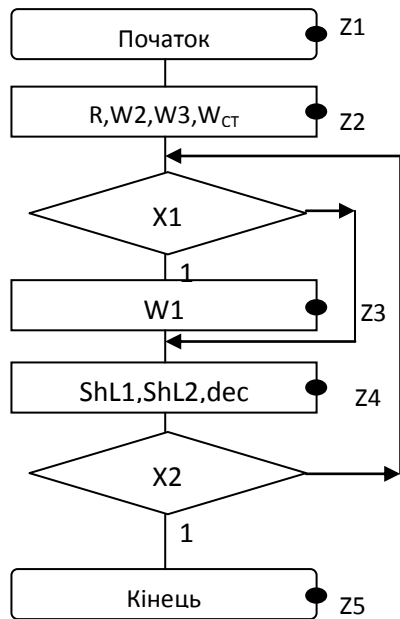


Рисунок 1.14- Закодований мікроалгоритм

1.3.7 Граф управляючого автомата Мура з кодами вершин

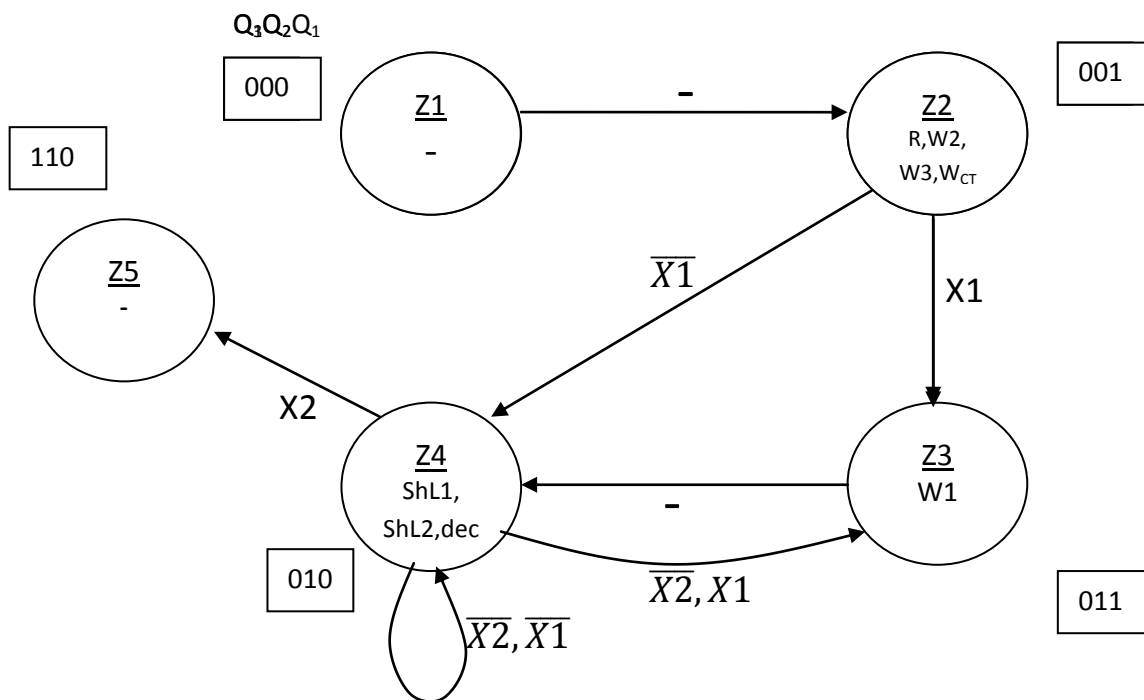


Рисунок 1.15- Граф автомата Мура

1.3.8 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 8 + 5 = 13_{10} = 1101_2.$$

Нормалізація мантиси.

$$M_z = , 0, 111100110101001 \leftarrow , P_z := P_z - 1.$$

$$, 111100110101001 \quad P_z = 1100.$$

Знак мантиси: $1 \oplus 0 = 1$.

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

Зн.Р	P=+12 ₁₀	Зн.М	М
0. 0 0 0 1 1 0 0		0, 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1	

1.4 Четвертий спосіб множення

1.4.1 Теоретичне обґрунтування

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.

Вираз

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

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

Отже, сума часткових добутків в i -му циклі, де $i = \overline{1, n}$, зводиться до обчислення виразу:

$$Z_i = Z_{i-1} + 2^{-1} Y_{i-1} \cdot x_i.$$

з початковими значеннями $i=1$, $Y_0=2^{-1}Y$, $Z_0=0$.

1.4.2 Операційна схема

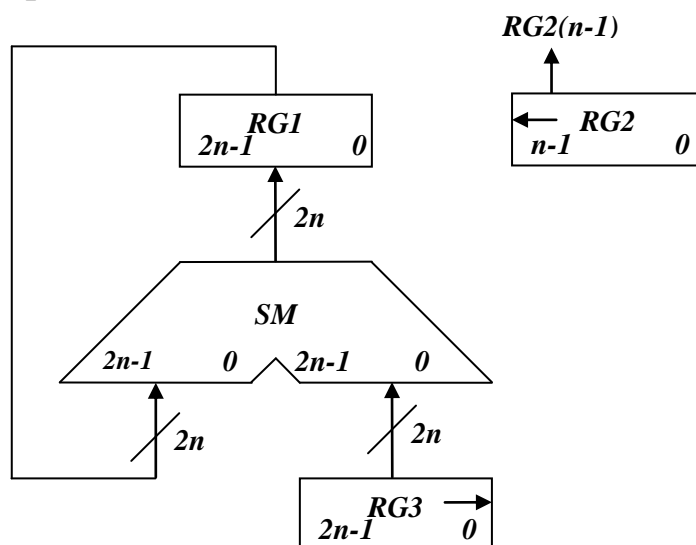


Рисунок 1.16-Операційна схема

1.4.3 Змістовний мікроалгоритм

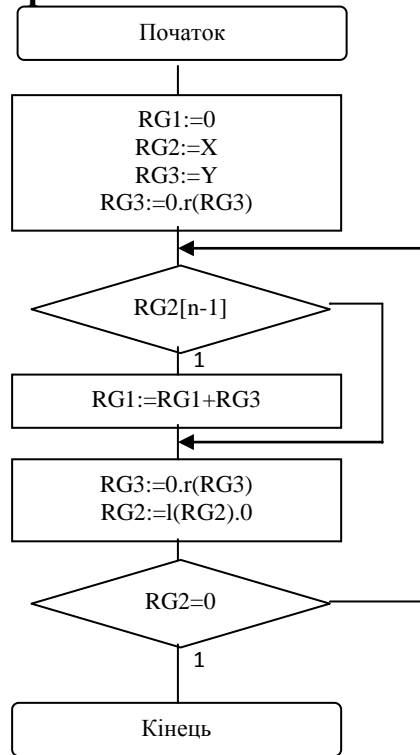


Рисунок 1.17-Змістовний мікроалгоритм

1.4.4 Таблиця станів регістрів

Таблиця 1.7- Таблиця станів регістрів

№ ц.	RG1	RG3	RG2
П.С.	00000000000000000000000000000000	01111100110010010000000000000000	111110011001100
1	01111100110010010000000000000000	00111110011001001000000000000000	111100110011000
2	+00111110011001001000000000000000 10111011001011011000000000000000	00011111001100100100000000000000	111001100110000
3	+00011111001100100100000000000000 11011010010111111100000000000000	00001111100110010010000000000000	110011001100000
4	+00001111100110010010000000000000 11101001111110001110000000000000	00000111110011001001000000000000	100110011000000
5	+00000111110011001001000000000000 11110001110001010111000000000000	00000011111001100100100000000000	001100011000000
6	11110001110001010111000000000000	00000001111100110010010000000000	011000110000000
7	11110001110001010111000000000000	00000000111110011001001000000000	110001100000000
8	+00000000111110011001001000000000 11110010101111110000001000000000	00000000011111001100100100000000	100011000000000
9	+00000000011111001100100100000000 11110011001110111100101100000000	00000000001111100110010010000000	000110000000000
10	11110011001110111100101100000000	00000000000111110011001001000000	110000000000000
11	11110011001110111100101100000000	00000000000011111001100100100000	100000000000000
12	+00000000000011111001100100100000 111100110100101101100100001000	00000000000001111100110010010000	000000000000000
13	+00000000000001111100110010010010 111100110101001100110000101100	00000000000000111110011001001010	000000000000000
14	111100110101001100110000101100	00000000000000011111001100100101	000000000000000

1.4.5 Функціональна схема з відображенням управляючих сигналів

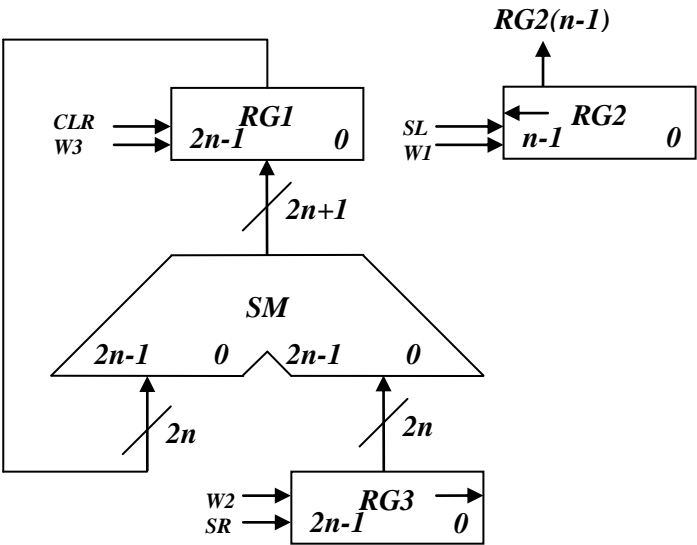


Рисунок 1.18-Функціональна схема

1.4.6 Закодований мікроалгоритм

Таблиця 1.8- Таблиця кодування мікрооперацій

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

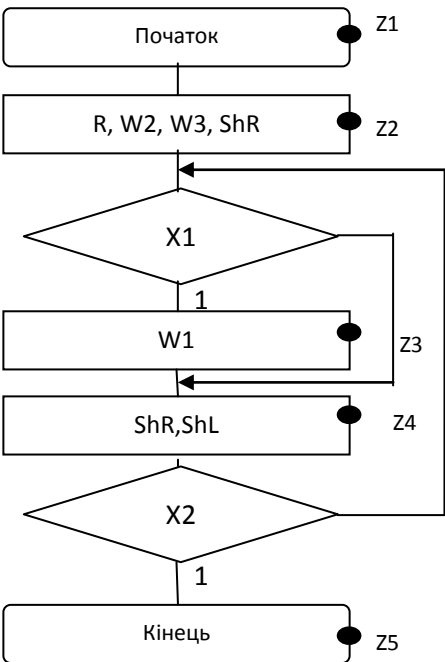


Рисунок 1.19- Закодований мікроалгоритм

1.4.7 Граф управляючого автомата Мура з кодами вершин

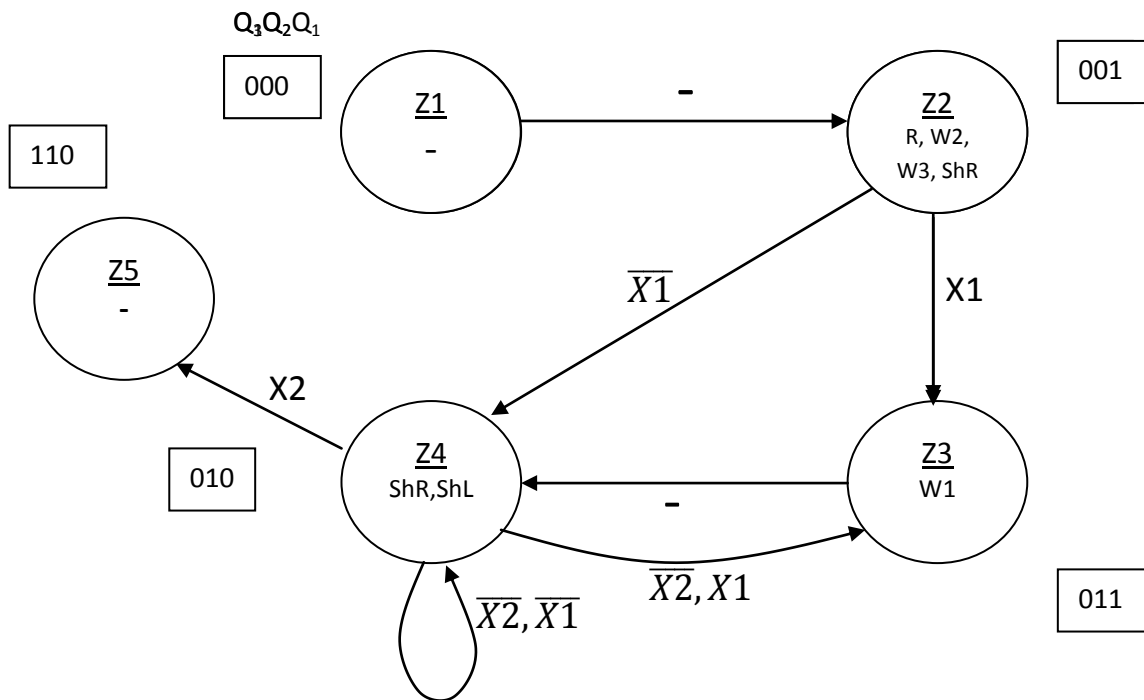


Рисунок 1.20- Граф автомата Мура

1.4.8 Обробка порядків і нормалізація

$$P_z = P_x + P_y = 8 + 5 = 13_{10} = 1101_2.$$

Нормалізація мантиси.

$$M_z = , 0,111100110101001 \leftarrow , P_z := P_z - 1. \\ , 111100110101001 \quad P_z = 1100.$$

Знак мантиси: $1 \oplus 0 = 1$.

1.4.9 Форма запису нормалізованого результату з плаваючою комою в пам'яті

Зн.Р	P=+12 ₁₀	Зн.М	М
0. 0 0 0 1 1 0 0		0, 1 1 1 1 0 0 1 1 0 1 0 1 0 0 1	

2. Операція ділення чисел

$Z=X: Y$, X-ділене, Y – дільник.

2.1 Перший спосіб ділення (з відновленням від'ємного залишку)

2.1.1 Теоретичне обґрунтування

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 P2 (у вихідному стані в цьому регістрі записаний X). Виходи P2 підключені до входів СМ безпосередньо, тобто ланцюги видачі коду з P2 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.

2.1.2 Операційна схема

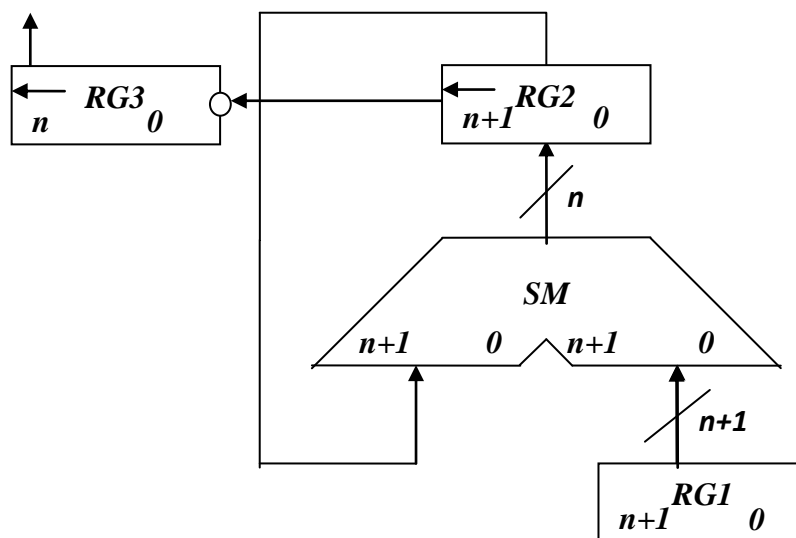


Рисунок 2.1-Операційна схема

2.1.3 Змістовний мікроалгоритм

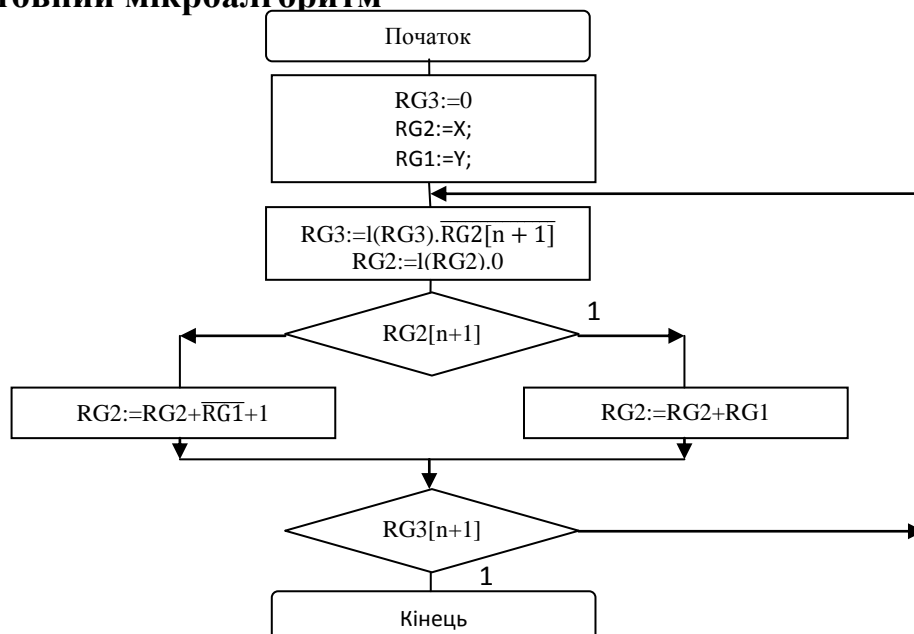


Рисунок 2.2-Змістовний мікроалгоритм

2.1.4 Таблиця станів регістрів

Таблиця 2.1- Таблиця станів реєстрів

№ ц. П.С.	RG3	RG2	RG1	МО
	0000000000000000	00111110011001100	00111110011001100 _{ПК} 11000001100110111 _{ДК}	ПС
1	0000000000000000 <u>1</u>	01111100110011000 +11000001100110111 00111110011001111		← RG2←RG3 RG2:=RG2- RG1
2	0000000000000000 <u>11</u>	01111100110011110 +11000001100110111 00111110011010101		← RG2←RG3 RG2:=RG2- RG1
3	0000000000000000 <u>111</u>	01111100110101010 +11000001100110111 00111110011100001		← RG2←RG3 RG2:=RG2- RG1
4	0000000000000000 <u>1111</u>	01111100111000010 +11000001100110111 00111110011111001		← RG2←RG3 RG2:=RG2- RG1
5	00000000000011111	01111100111110010 +11000001100110111 00111110100101001		← RG2←RG3 RG2:=RG2- RG1
6	00000000000111111	01111101001010010 +11000001100110111 00111110110001001		← RG2←RG3 RG2:=RG2- RG1
7	00000000011111111	01111101100010010 +11000001100110111 00111111001001001		← RG2←RG3 RG2:=RG2- RG1
8	00000000111111111	01111110010010010 +11000001100110111 00111111111001001		← RG2←RG3 RG2:=RG2- RG1
9	00000001111111111	01111111110010010 +11000001100110111 01000001011001001		← RG2←RG3 RG2:=RG2- RG1
10	00000011111111111	10000010110010010 +00111110011001001 11000001001011011		← RG2←RG3 RG2:=RG2- RG1
11	00000111111111110	10000010010110110 +00111110011001001 11000000101111111		← RG2←RG3 RG2:=RG2- RG1
12	00001111111111100	10000001011111110 +00111110011001001 10111111111000111		← RG2←RG3 RG2:=RG2- RG1
13	00011111111111000	01111111110001110 +11000001100110111 01000001011000101		← RG2←RG3 RG2:=RG2- RG1
14	00111111111110001	10000010110001010 +00111110011001001 11000001001010011		← RG2←RG3 RG2:=RG2- RG1
15	01111111111100010	10000010010100110 +00111110011001001 11000000101101111		← RG2←RG3 RG2:=RG2- RG1
16	<u>1</u> ,1111111111000100	10000001011011110 +00111110011001001 10111111110100111		← RG2←RG3 RG2:=RG2- RG1

2.1.5 Функціональна схема з відображенням управляючих сигналів

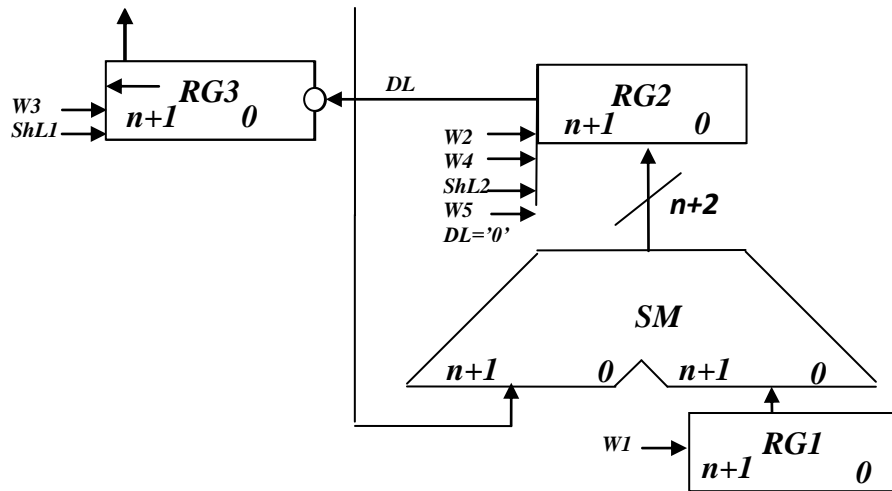


Рисунок 2.3- Функціональна схема

2.1.6 Закодований мікроалгоритм

Таблиця 2.2- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
RG3:=0	W3
RG2:=X;	W2
RG1:=Y;	W1
$RG3:=l(RG3).\overline{RG2[n+1]}$	ShL1
$RG2:=l(RG2).0$	ShL2
$RG2:=RG2+\overline{RG1}+1$	W4
$RG2:=RG2+RG1$	W5

Таблиця кодування логічних умов	
ЛУ	Позначення
$RG2[n+1]$	X1
$RG3[n+1]$	X2

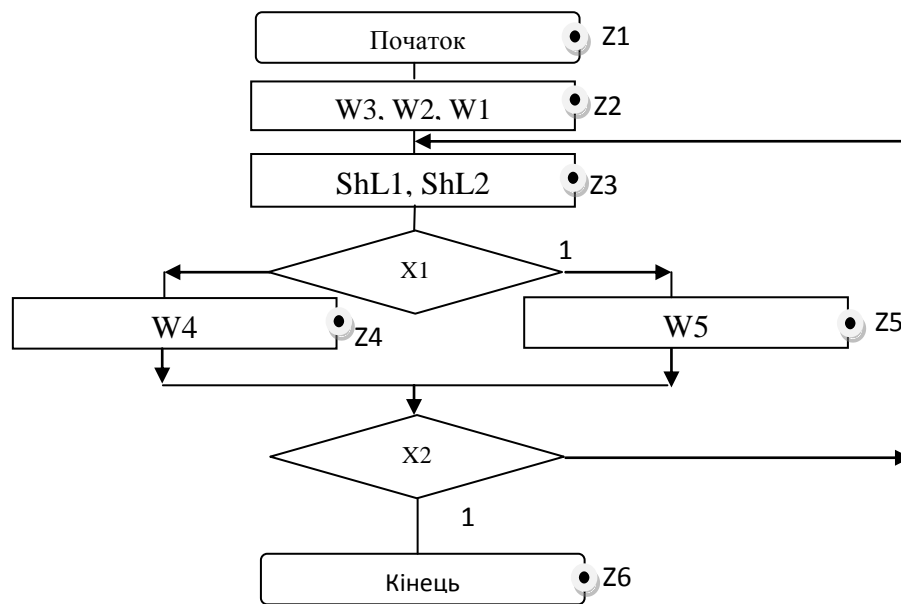


Рисунок 2.4- Закодований мікроалгоритм

2.1.7 Граф управляючого автомата Мура з кодами вершин

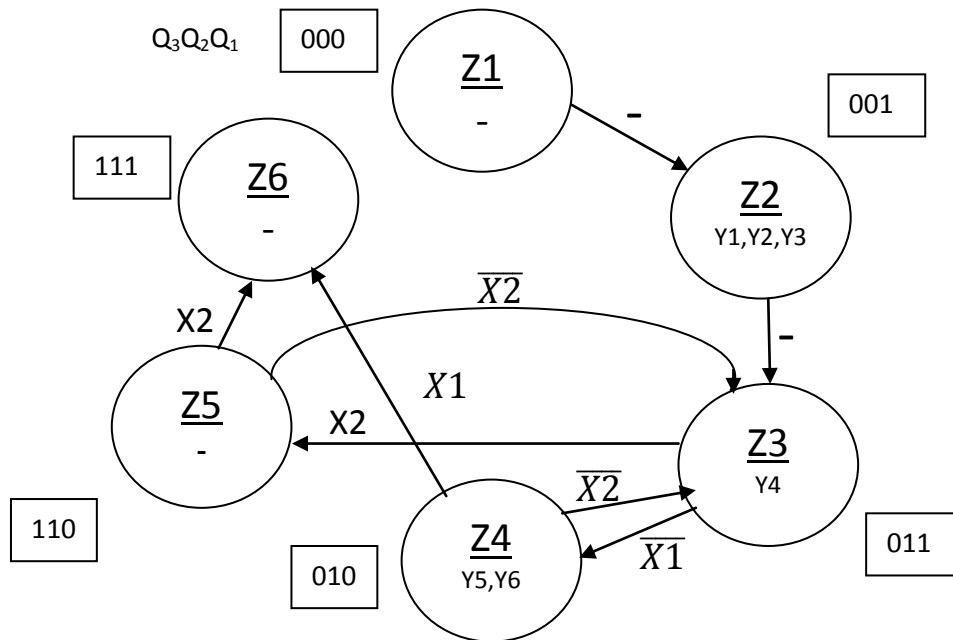


Рисунок 2.5- Граф автомата Мура

2.1.8 Обробка порядків і нормалізація

$$P_z = P_x - P_y = 8 - 5 = 3_{10} = 11_2.$$

Нормалізація мантиси не потрібна.

$$M_z = , 111111111000100.$$

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

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

3 _H .P	P=+12 ₁₀							3 _H .M	M													
0.	0	0	0	0	0	1	1	1,	1	1	1	1	1	1	1	1	0	0	0	1	0	0

2.2 Другий спосіб ділення(без відновлення остачі)

2.2.1 Теоретичне обґрунтування

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 підсумовування і віднімання, зсув в регістрах Y,Z. Тобто 1 цикл може складатися з 1 такту, це дає прискорення відносно 1-го способу.

2.2.2 Операційна схема

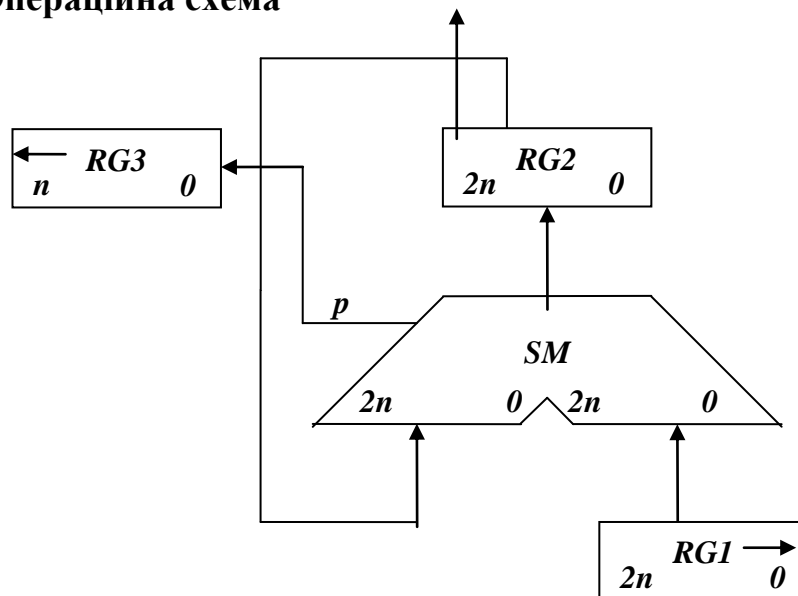


Рисунок 2.6-Операційна схема

2.2.3 Змістовний мікроалгоритм

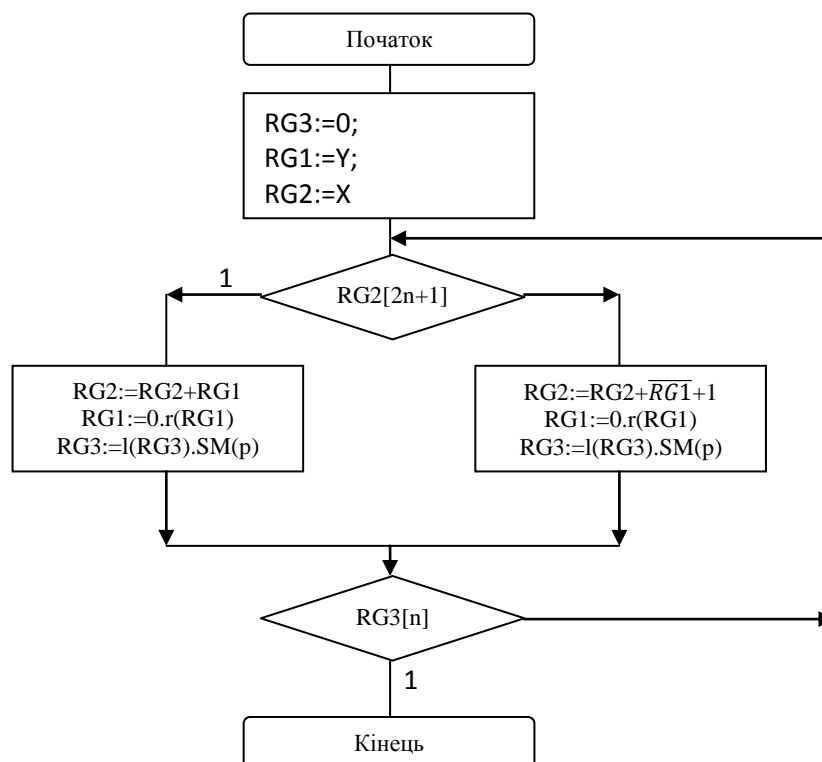


Рисунок 2.7-Змістовний мікроалгоритм

2.2.4 Таблиця станів регістрів

Таблиця 2.3- Таблиця станів регістрів

№ ц.	RG3	RG2	RG1
П.С.	000000000000000 <u>1</u>	011111001100110000000000000000	001111100110010010000000000000 _{ПК} 110000011001101110000000000000 _{ДК}
1	000000000000000 <u>11</u>	+110000011001101110000000000000 001111100110011110000000000000	000111110011001001000000000000
2	000000000000000 <u>111</u>	+111000001100110111000000000000 000111110011010101000000000000	000011111001100100100000000000
3	000000000000000 <u>1111</u>	+111100000110011011100000000000 000011111001110000100000000000	000001111100110010010000000000
4	0000000000000 <u>11111</u>	+111110000011001101110000000000 000001111100111110010000000000	000000111110011001001000000000
5	00000000000 <u>111111</u>	+111111000001100110111000000000 000000111110100101001000000000	000000011111001100100100000000
6	0000000000 <u>1111111</u>	+111111100000110011011100000000 000000011111011000100100000000	000000001111100110010010000000
7	000000000 <u>11111111</u>	+111111110000011001101110000000 000000001111110010010010000000	000000000111110011001001000000
8	00000000 <u>111111111</u>	+111111111000001100110111000000 000000000111111111001001000000	000000000011111001100100100000
9	0000000 <u>1111111111</u>	+111111111100000110011011100000 000000000100000101100100100000	000000000001111100110010010000
10	000000 <u>11111111111</u>	+111111111110000011001101110000 000000000010001000110010010000	000000000000111110011001001000
11	00000 <u>111111111111</u>	+111111111111000001100110111000 000000000001001010011001001000	000000000000011111001100100100
12	0000 <u>1111111111111</u>	+111111111111100000110011011100 000000000000101011001100100100	000000000000001111100110010010
13	000 <u>11111111111111</u>	+111111111111110000011001101110 000000000000011011100110010010	000000000000000111110011001001
14	00 <u>111111111111111</u>	+111111111111111000001100110111 0000000000000010011110011001001	000000000000000011111001100100
15	0 <u>1111111111111111</u>	+111111111111111100000110011100 000000000000000111111001100101	000000000000000001111100110010

2.2.5 Функціональна схема з відображенням управляючих сигналів

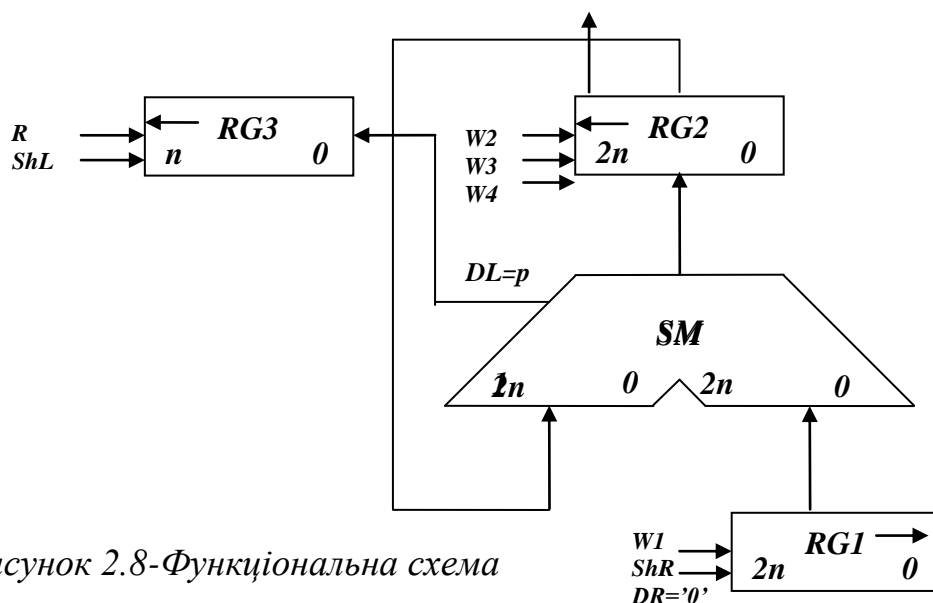


Рисунок 2.8-Функціональна схема

2.2.6 Закодований мікроалгоритм

Таблиця 2.4- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій		Таблиця кодування логічних умов	
МО	УС	ЛУ	Позначення
RG3:=0;	R	RG2[2n+1]	X1
RG1:=Y;	W1	RG3[n]	X2
RG2:=X	W2		
RG2:=RG2+RG1	W3		
RG1:=0.r(RG1)	ShR		
RG3:=l(RG3).SM(p)	ShL		
RG2:=RG2+RG1+1	W4		

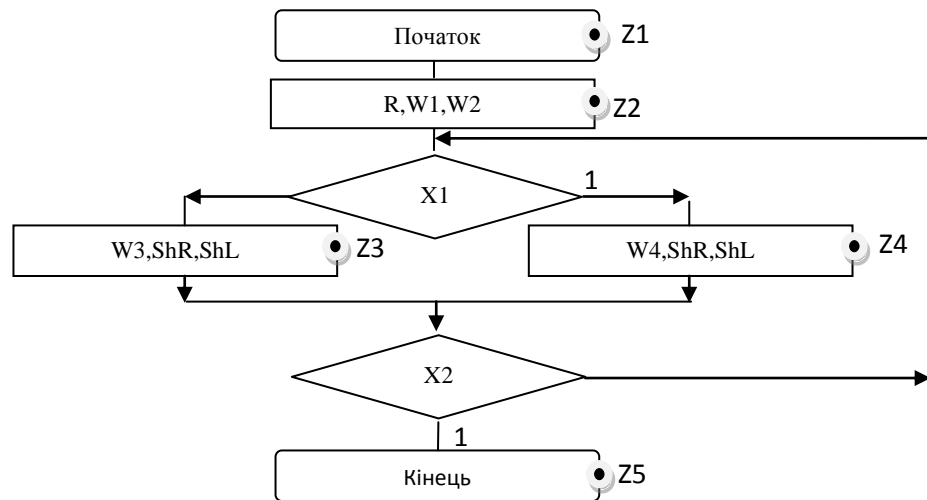
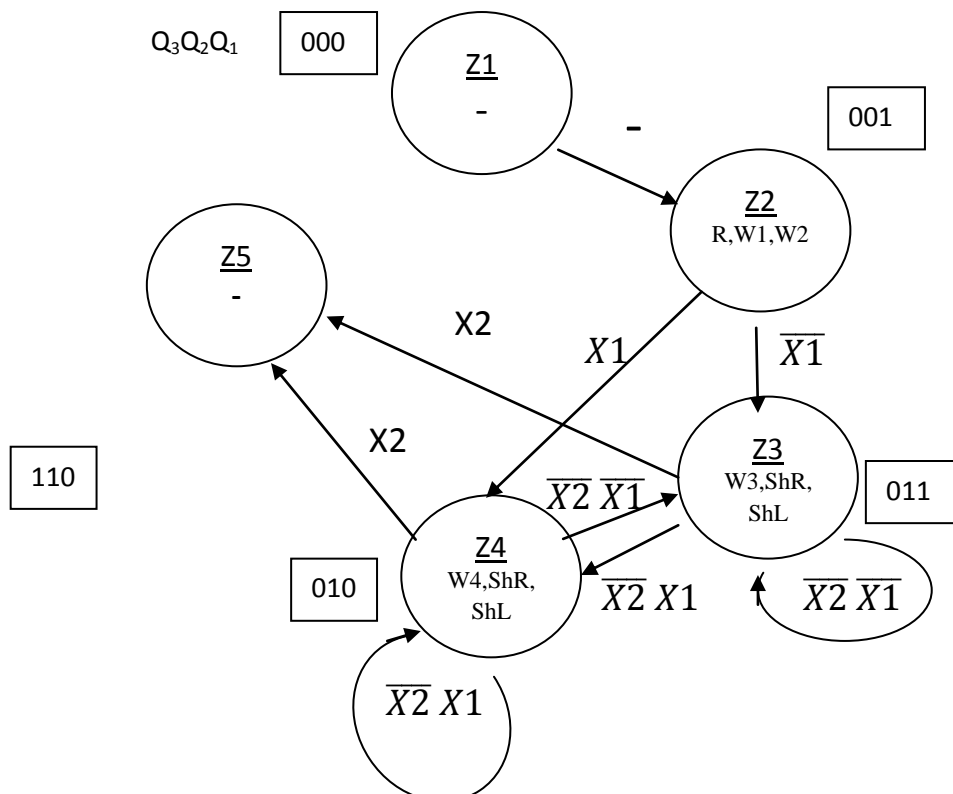


Рисунок 2.9- Закодований мікроалгоритм

2.2.7 Граф управляючого автомата Мура з кодами вершин



2.2.8 Обробка порядків і нормалізація

$$P_z = P_x - P_y = 8 - 5 = 3_{10} = 11_2$$

Нормалізація мантиси не потрібна.

$$M_z = , 1111111111000100.$$

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

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

Зн.Р Р=+3 ₁₀								Зн.М								М							
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0		

3. Операція додавання чисел

$$Z=X+Y.$$

3.1 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.

$$P_x > P_y \rightarrow P_z = P_x = +8_{10} = +1000_2,$$

$$\Delta = P_x - P_y = 8 - 5 = 3_{10} = 11_2.$$

2. Вирівнювання порядків.

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

Таблиця 3.1- Таблиця зсуву мантиси на етапі вирівнювання порядків

M_Y	Δ	Мікрооперація
0.1111110011001001	11	П.С.
0.0111111001100100	01	$M_Y \rightarrow \Delta := \Delta - 1$
0.0011111100110010	00	$M_Y \rightarrow \Delta := \Delta - 1$

3. Додавання мантис у модифікованому ДК.

Таблиця 3.2-Додавання мантис

111110011001100

1	1,	1	1	1	1	1	0	0	1	1	0	0	1	1	0	0
0	0,	0	0	1	1	1	1	1	0	0	1	1	0	0	1	0
0	0,	0	0	1	1	0	1	1	1	1	1	1	1	1	1	0

$$M_x = 11,111110011001100_{ПК} = 11,000001100110100_{ДК}$$

$$M_y = 00,001111100110010_{ПК} = 00,001111100110010_{ДК}$$

$$M_z = 00,001101111111110_{ДК} = 00,001101111111110_{ПК}$$

4. Нормалізація результату (В ПК).

3.2 Операційна схема

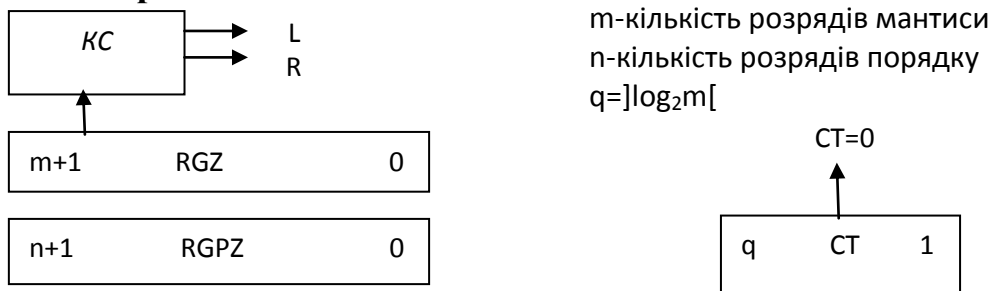


Рисунок 3.1-Операційна схема

Виконаємо синтез КС для визначення порушення нормалізації.

Таблиця 3.3-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.

3.3 Змістовний мікроалгоритм

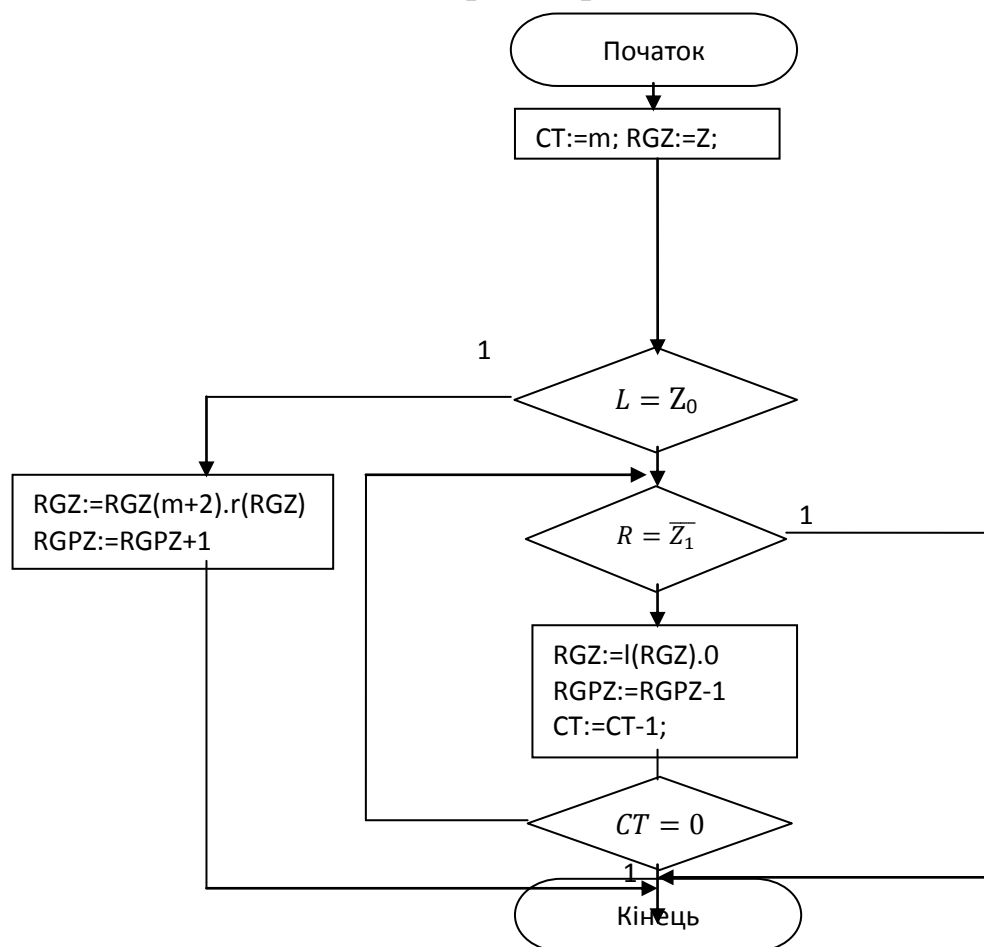


Рисунок 3.2-Змістовний мікроалгоритм

3.4 Таблиця станів регістрів

Таблиця 3.4- Таблиця станів регістрів

№ такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001000	00.111110011001001	0	1	100	
1	000111	00.1111100110010010 00.111001100100100	0	0	011	$Z'_0 Z_0 := \overline{Z'_0 Z_0}$ RGZ:=l(RGZ).0 RGPZ:=RGPZ-1 CT:=CT-1

3.5 Функціональна схема з відображенням управляючих сигналів

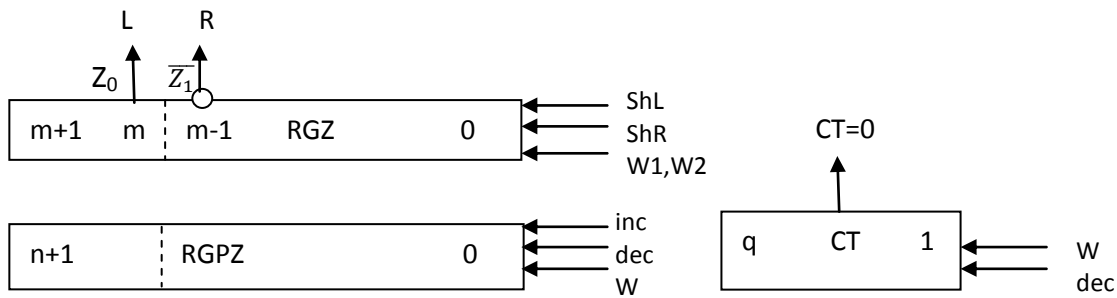


Рисунок 3.3- Функціональна схема

3.6 Закодований мікроалгоритм

Таблиця 3.5- Таблиця кодування мікрооперацій

МО	УС
CT:=m;	W
RGZ:=Z;	W1
$Z'_0 Z_0 := \overline{Z'_0} Z_0$	W2
RGZ:=RGZ(m+2).r(RGZ)	ShR
RGPZ:=RGPZ+1	inc
RGZ:=l(RGZ).0	ShL
RGPZ:=RGPZ-1	dec
CT:=CT-1;	dec

ЛУ	Позначення
$Z'_0 = 0$	X1
$L = Z_0$	X2
$R = \overline{Z'_1}$	X3
$CT = 0$	X4

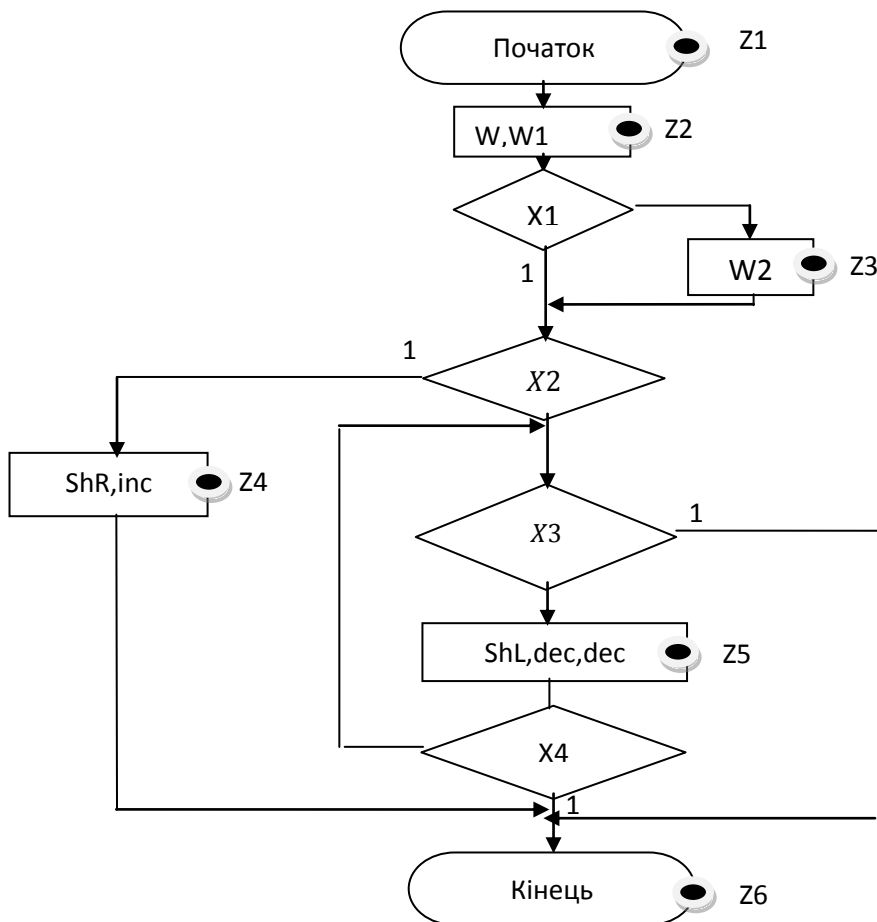


Рисунок 3.4- Закодований мікроалгоритм

3.7 Граф управляючого автомата Мура з кодами вершин

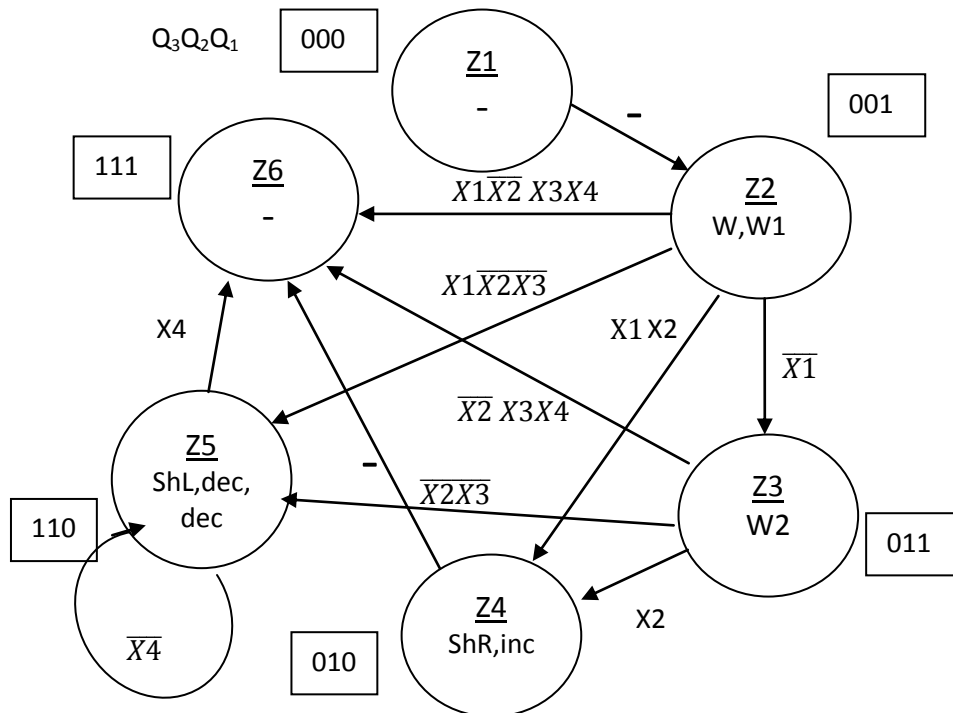
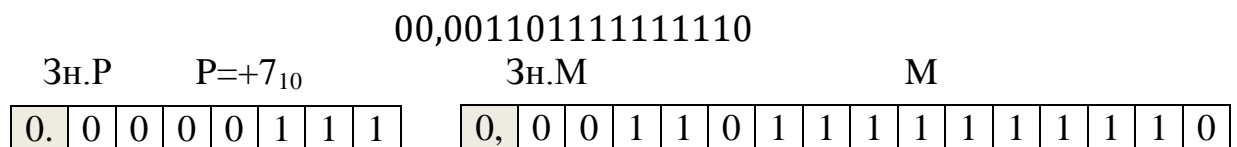


Рисунок 3.5- Граф автомата Мура

3.8 Обробка порядків

PZ=111.

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



4. Операція добування кореня

$$Z = \sqrt{|X|}$$

4.1 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.

4.2 Операційна схема

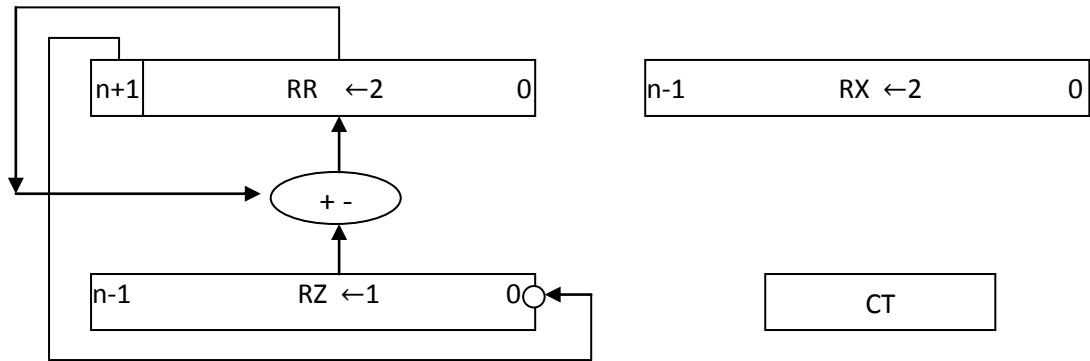


Рисунок 4.1-Операційна схема

4.3 Змістовний мікроалгоритм

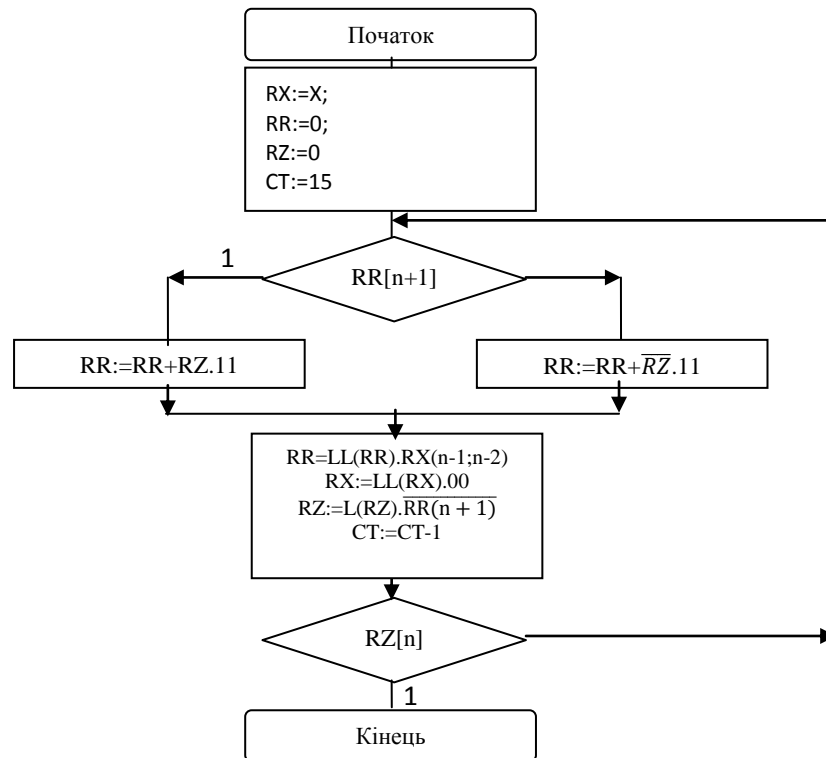


Рисунок 4.2-Змістовний мікроалгоритм

4.4 Таблиця станів регістрів

Таблиця 4.1- Таблиця станів регістрів

№ ц.	RZ	RR	RX	CT
П.С. Пзсув	0000000000000000	0000000000000000 0000000000000011	111110011001100	1111
1	0000000000000001	+1111111111111111 0000000000000010 0000000000000101	011111001100110	1110
2	0000000000000011	+1111111111111101 0000000000000011 0000000000001101	001111100110011	1101
3	0000000000000111	+1111111111111001 0000000000000110 0000000000011010	000111110011001	1100
4	0000000000001111	+1111111111110001 0000000000001100 0000000000110010	000011111001100	1011
5	0000000000111111	+1111111111100001 0000000000010010 0000000001001010	000001111100110	1010
6	0000000001111111	+1111111111000001 0000000000011000 0000000000110001	000000111110011	1001
7	0000000011111110	+1111111110000001 1111111101100101 1111110110010100	000000011111001	1000
8	0000000111111100	+0000000011111101 1111111111000111 1111111000111100	000000001111100	0111
9	0000001111111001	+0000000111111001 0000000100010111 0000010001011100	000000000111110	0110
10	0000011111110011	+1111110000001101 0000000001101011 0000000110101100	000000000011111	0101
11	0000111111100110	+1111100000011001 1111100111000111 1110011100011110	000000000001111	0100
12	0001111111001100	+0000111111001101 1111011011101011 1101101110101100	000000000000111	0011
13	0011111110011000	+0001111111001100 1111101101000111 1110110100011110	000000000000011	0010
14	0111111100110001	+0011111100110001 0010110001001111 1011000100111100	000000000000001	0001
15	1111111001100011	+0111111100110001 0010111110100001	000000000000000	0000

4.5 Функціональна схема з відображенням управляючих сигналів

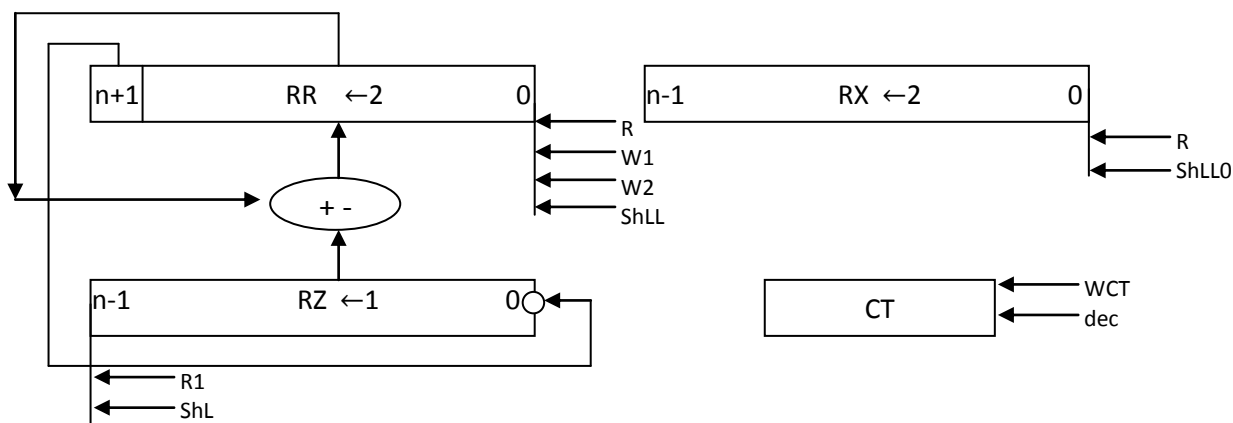


Рисунок 4.3-Функціональна схема

4.6 Закодований мікроалгоритм

Таблиця 4.2- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
RX:=X;	WX
RR:=0;	R
RZ:=0	R1
CT:=15	WCT
RR:=RR+RZ.11	W1
RR:=RR+ \overline{RZ} .11	W2
RR=LL(RR).RX(n-1;n-2)	ShLL
RX:=LL(RX).00	ShLL0
RZ:=L(RZ). $\overline{RR(n+1)}$	ShL
CT:=CT-1	dec

Таблиця кодування логічних умов	
ЛЮ	Позначення
RR[n+1]	X1
RZ[n]	X2

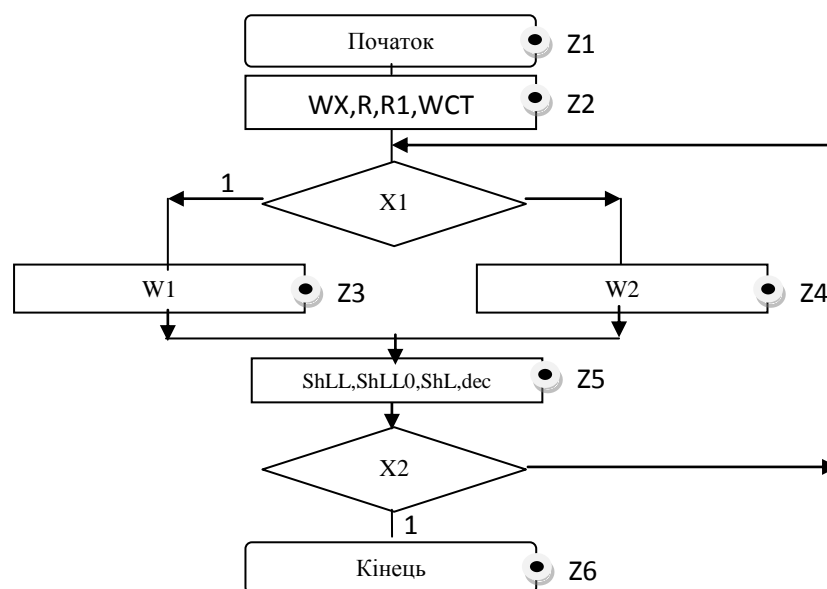


Рисунок 4.4- Закодований мікроалгоритм

4.7 Граф управляющего автомата Мура з кодами вершин

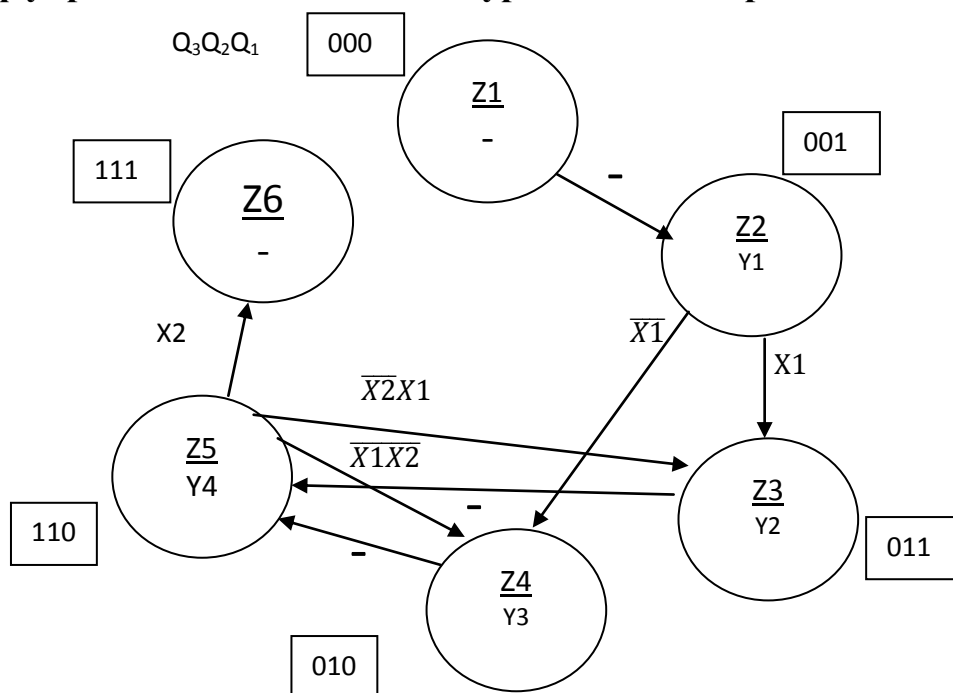


Рисунок 4.5- Граф автомата Мура

4.8 Обработка порядків

$PZ=PX:2=8:2=4_{10}=100_2$.

4.9 Форма запису нормалізованого результату з плаваючою комою в пам'яті

Зн.Р	P=+4 ₁₀	Зн.М	М
0. 0 0 0 0 1 0 0		0, 1 1 1 1 1 1 0 0 1 1 0 0 0 1 1	

5. Синтез управляющего автомата для операционного пристрою множення третім способом.

$x_3x_2x_1+1=101=5$ – операція множення третім способом.

5.1 Таблица співвідношення управляючих входів операционного автомата і виходів управляющего автомата

За закодованим мікроалгоритмом (Рис. 1.14) складемо таблицю:

Таблиця 5.1 Таблица кодування сигналів

Входи операционного автомата	Виходи управляющего автомата
R,W2,W3,W _{СТ}	Y1
W1	Y2
ShL1,ShL2,dec	Y3

5.2 Мікроалгоритм в термінах управляючого автомата

Зробимо автомат Мура циклічним задля зменшення кількості вершин.

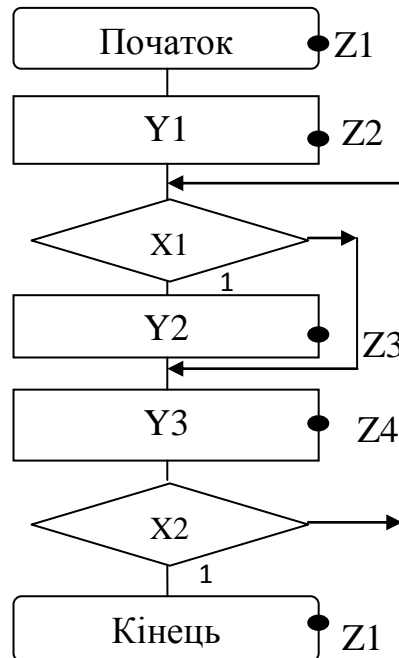


Рисунок 5.1- Закодований мікроалгоритм

Будуємо граф автомата Мура

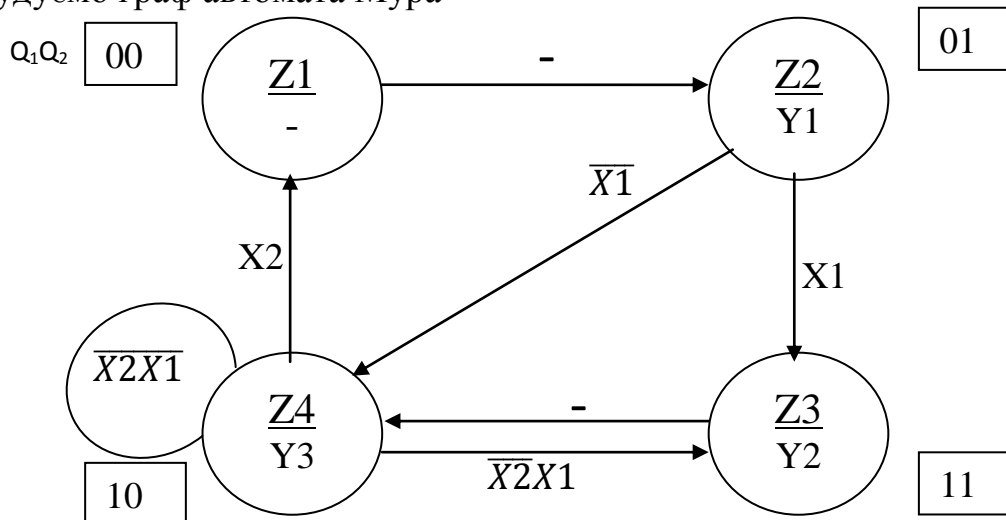


Рисунок 5.2- Граф автомата Мура

5.3 Структурна таблиця автомата

За графом автомата мура складаємо структурну таблицю автомата. Значення функцій збудження тригерів визначаються відповідно до графічної схеми переходів JK-тригера.

Таблиця 5.2-Структурна таблиця автомата

Перехід	Q_2Q_1	Q_2Q_1	x_1x_2	$y_1y_2y_3$	J_2	K_2	J_1	K_1
$z_1 \rightarrow z_2$	00	01	--	000	0	-	1	-
$z_2 \rightarrow z_3$	01	11	1-	100	1	-	-	0
$z_2 \rightarrow z_4$	01	10	0-	100	1	-	-	1
$z_3 \rightarrow z_4$	11	10	--	010	-	0	-	1
$z_4 \rightarrow z_3$	10	11	10	001	-	0	1	-
$z_4 \rightarrow z_1$	10	00	-1	001	-	1	0	-
$z_4 \rightarrow z_4$	10	10	00	001	-	0	0	-

JK-тригер:

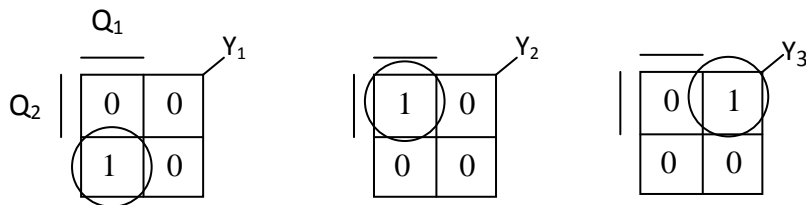
$$0 \xrightarrow{0-} 0$$

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

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

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

5.4 Синтез функцій виходів і переходів



$$Y_1 = \overline{Q_2}Q_1$$

$$Y_2 = Q_2Q_1$$

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

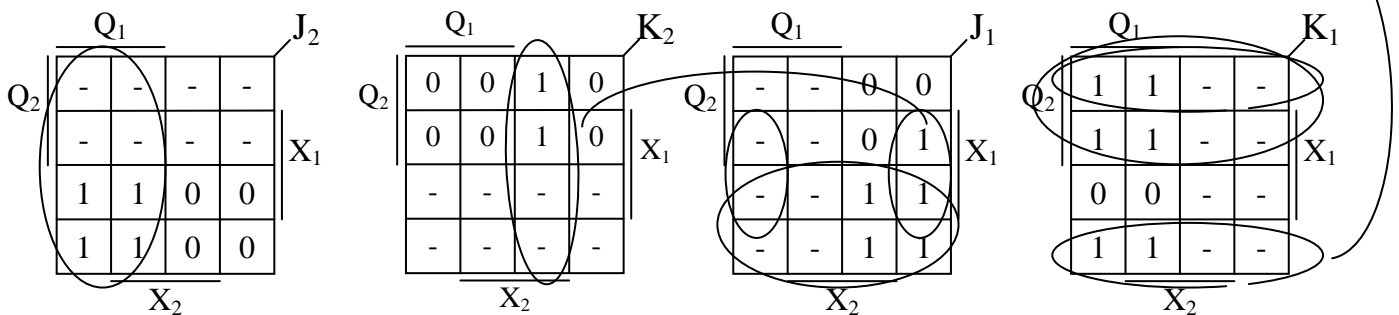


Рисунок 5.3- Діаграми Вейча

$$J_2 = Q_1$$

$$K_2 = \overline{Q_1}X_2$$

$$J_1 = \overline{Q_1} \cup X_1\overline{X_2}$$

$$K_1 = Q_2 \vee \overline{X_1}$$

5.5 Функціональна схема пристрою (виходи управляючого автомата підключені до входів операційного автомата)

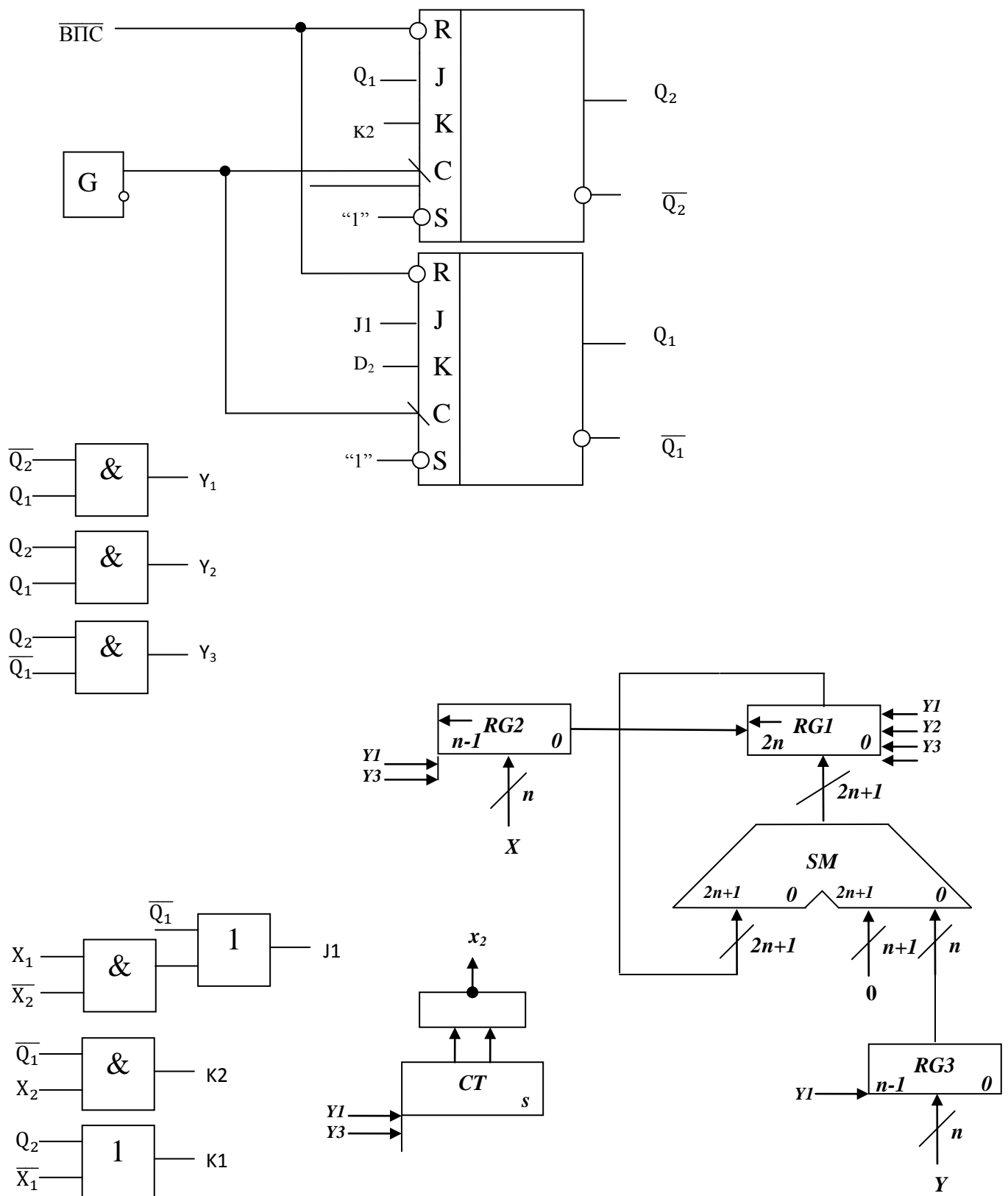


Рисунок 5.5- Функціональна схема пристрою