

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

РОЗРАХУНКОВА РАБОТА

по курсу „Комп'ютерна логіка-2”

Виконав: Бедь Анатолій Михайлович

Група ІО-12, Факультет ІОТ,

Залікова книжка № 1202

Номер технічного завдання 10010110001

(підпис керівника)

Завдання:

1. Числа X_2 і Y_2 в прямому коді записати у формі з плаваючою комою (з порядком і мантисою, а також з характеристикою та мантисою), як вони зберігаються у пам'яті. На порядок відвести 8 розрядів, на мантису 16 розрядів (з урахуванням знакових розрядів).

2. Виконати 8 операцій з числами X_2 і Y_2 з плаваючою комою (чотири способи множення, два способи ділення, додавання та віднімання). Номери операцій (для п.3) відповідають порядку переліку (наприклад, 6 – ділення другим способом). Для обробки мантис кожної операції, подати:

2.1 теоретичне обґрунтування способу;

2.1 операційну схему;

2.2 змістовний мікроалгоритм;

2.3 таблицю станів регістрів (лічильника), довжина яких забезпечує одержання 15 основних розрядів мантиси результату;

2.4 функціональну схему з відображенням управляючих сигналів;

2.5 закодований мікроалгоритм (мікрооперації замінюються управл. сигналами);

2.6 граф управляючого автомата Мура з кодами вершин;

2.7 обробку порядків (показати у довільній формі);

2.8 форму запису нормалізованого результату з плаваючою комою в пам'ять.

Вказані пункти для операцій додавання та віднімання 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 $x_3x_2x_1$ побудувати управляючий автомат Мура на тригерах (тип вибрати самостійно) і елементах булевого базису.

Визначення та обґрунтування варіанту:

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

$$X_2 = -1x_{10}x_91x_8x_7x_61, x_5x_40x_31x_2x_1 \text{ і } Y_2 = +1x_{10}1x_9x_8, x_7x_61x_5x_40x_3x_2x_1,$$

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

$$1202_{10}=10010110001_2;$$

$$X_2 = -1x_{10}x_91x_8x_7x_61, x_5x_40x_31x_2x_1 = -10011011,1000101;$$

$$Y_2 = +1x_{10}1x_9x_8, x_7x_61x_5x_40x_3x_2x_1 = +10101,0111000011;$$

Основна частина:

Завдання №1

$$X_{\text{ПК}} = 1.10011011, 1000101;$$

$$Y_{\text{ПК}} = 0.10101, 0111000011;$$

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

X_2 :

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

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

Y_2 :

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

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

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

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

$$m = 7;$$

$$2^7 = 10000000_2$$

$$E_x = 10000000 + 1000 = 10001000$$

X_2 :

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

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

$$E_y = 10000000 + 101 = 10000101$$

Y_2 :

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

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

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

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

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

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

2.1.2 Операційна схема:

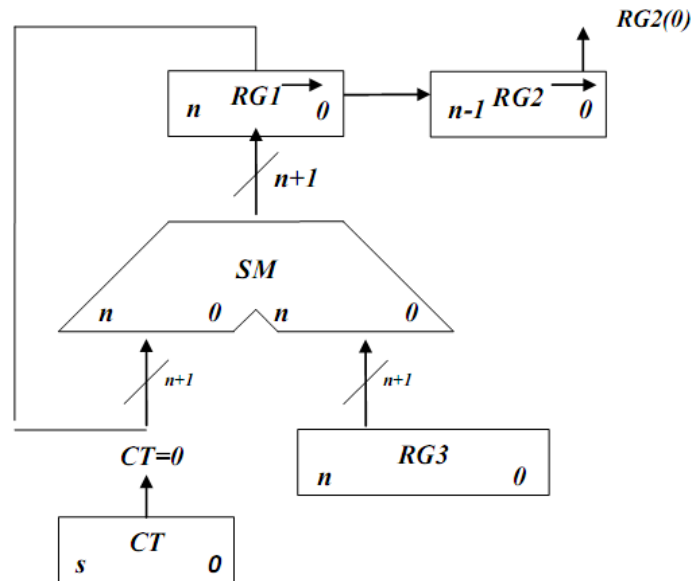


Рисунок 2.1.1- Операційна схема.

2.1.3 Змістовний мікроалгоритм:

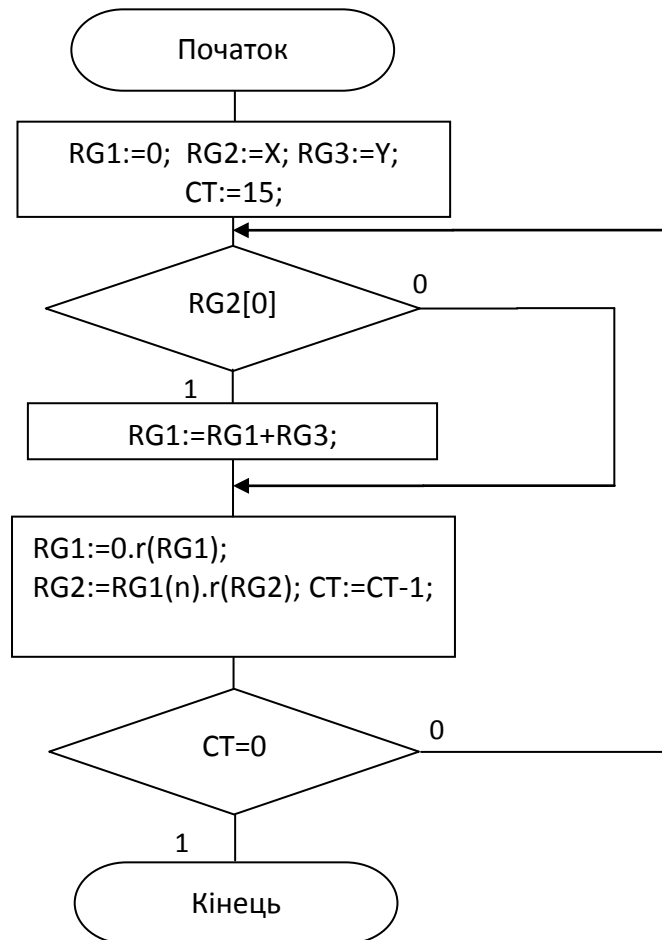


Рисунок 2.1.2 - Змістовний мікроалгоритм виконання операції множення першим способом.

2.1.4 Таблиця станів регістрів:

Таблиця 2.1.1-Таблиця станів регістрів для першого способу множення.

№	RG1	RG2	RG3	СТ
пс	0	100110111000101	0101010111000011	1111
1	+1010101111000011 =0101010111000011 0010101011100001	110011011100010		1110
2	001010101110000	111001101110001		1101
3	+1010101111000011 =01101011100110011 00110101110011001	111100110111000		1100
4	0001101011001100	111110011011100		1011
5	0000110101100110	011111001101110		1010
6	0000011010110011	001111100110111		1001
7	+1010101111000011 =0101110001110110 0010111000111011	000111110011011		1000
8	+1010101111000011 =1000001111111110 0100000111111111	000011111001101		0111
9	+1010101111000011 =10010111111000010 0100101111100001	000001111100110		0110
10	0010010111110000	100000111110011		0101
11	+1010101111000011 =0111101110110011 00111101111011001	110000011111001		0100
12	+1010101111000011 =10010011110011100 01001001111001110	011000001111100		0011
13	0010010011100111	001100000111110		0010
14	0001001001110011	100110000011111		0001
15	+1010101111000011 =0110100000110110 <u>0011010000011011</u>	<u>010011000001111</u>		0000

2.1.5 Функціональна схема:

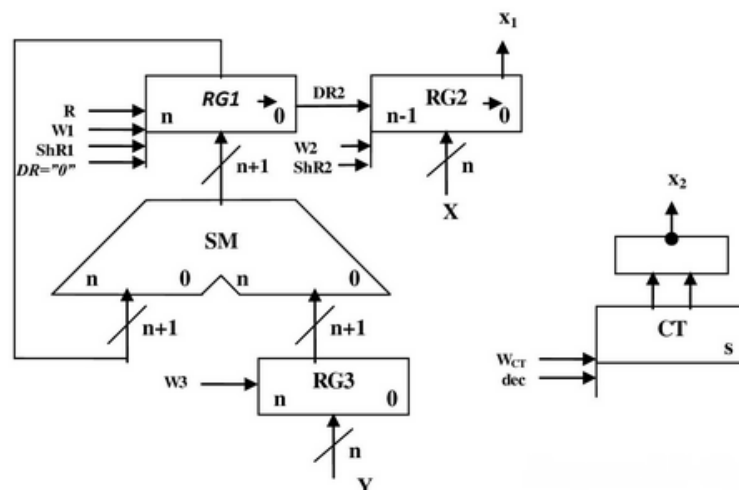


Рисунок 2.1.3- Функціональна схема.

2.1.6 Закодований мікроалгоритм

Таблиця 2.1.2-Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛУ	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		

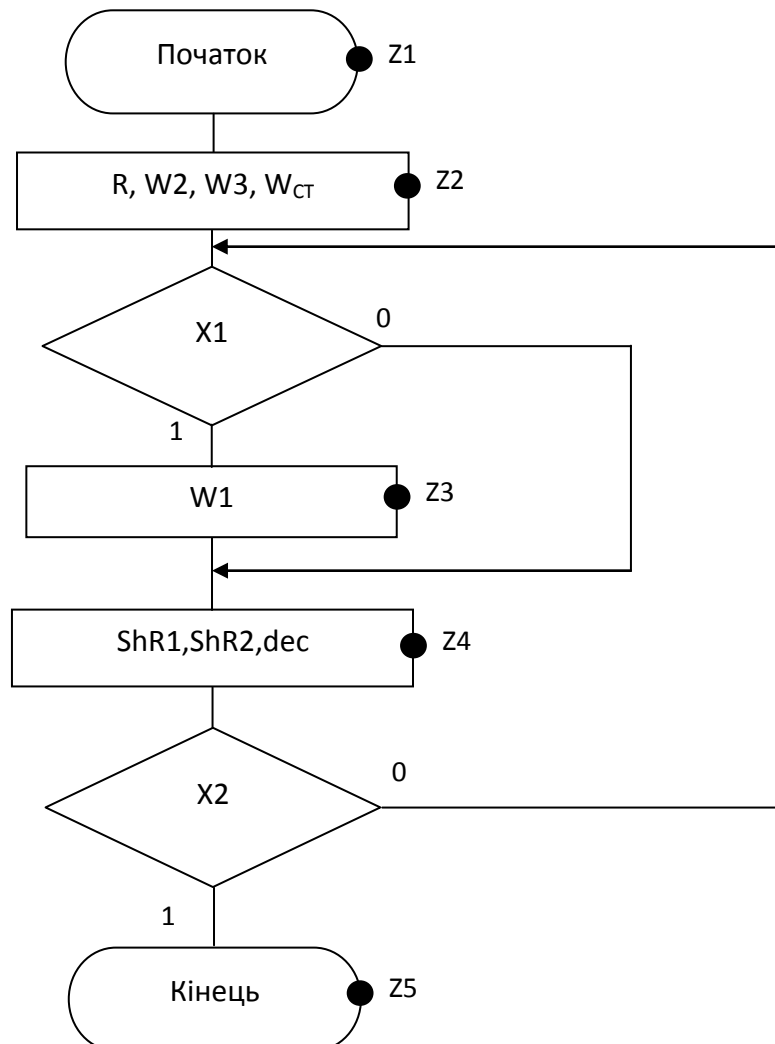


Рисунок 2.1.4-Закодований мікроалгоритм.

2.1.7 Граф управляючого автомата Мура з кодами вершин:

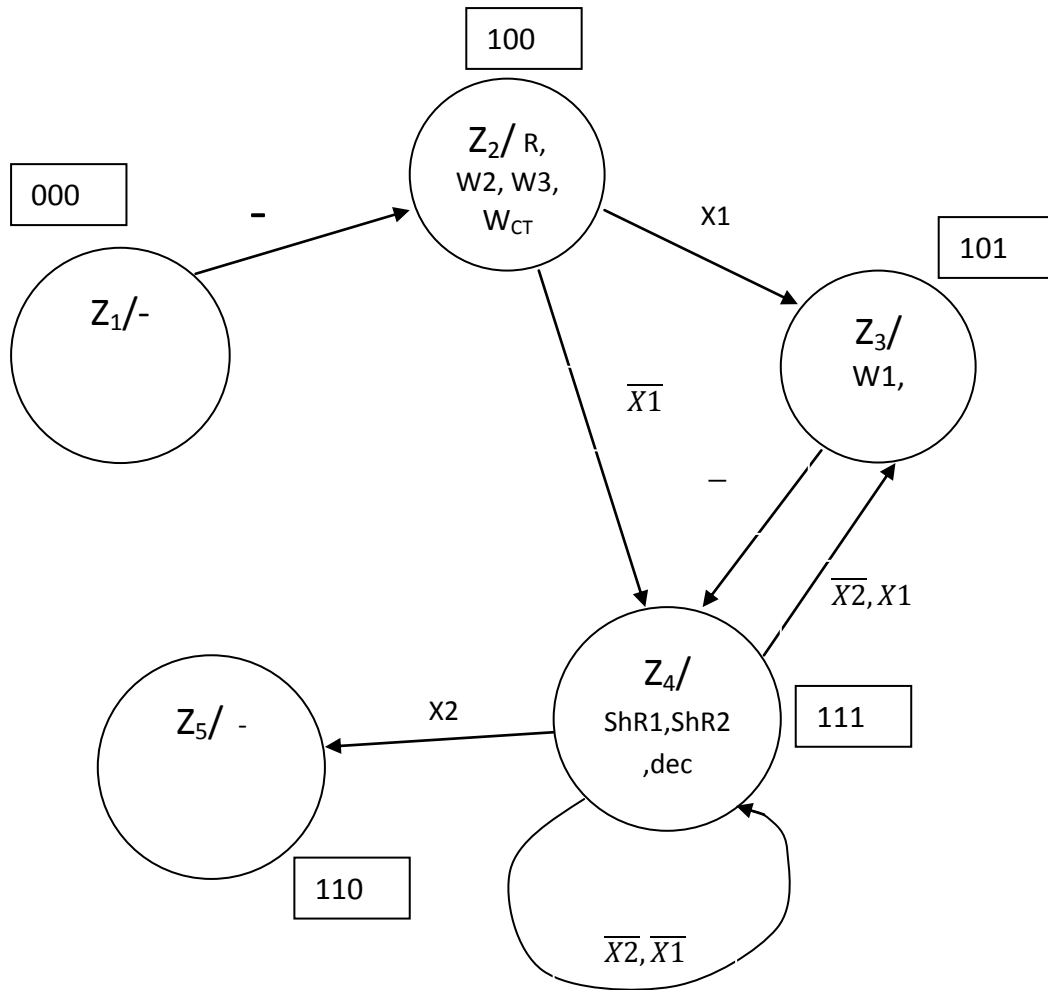


Рисунок 2.1.5-Граф автомата Мура

2.1.8 Обробка порядків:

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

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

2.1.9 Нормалізація результату:

Отримали результат: 011010000011011010011000001111

Знак мантиси: $1 \oplus 0 = 1$.

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

Порядок зменшуємо на 1:

$$11010000011011010011000001111; P_z=12;$$

Запишемо нормалізований результат:

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

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.

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

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

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

$$Z_0 = 0;$$

$$Y_0 = 0$$

2.2.2 Операційна схема:

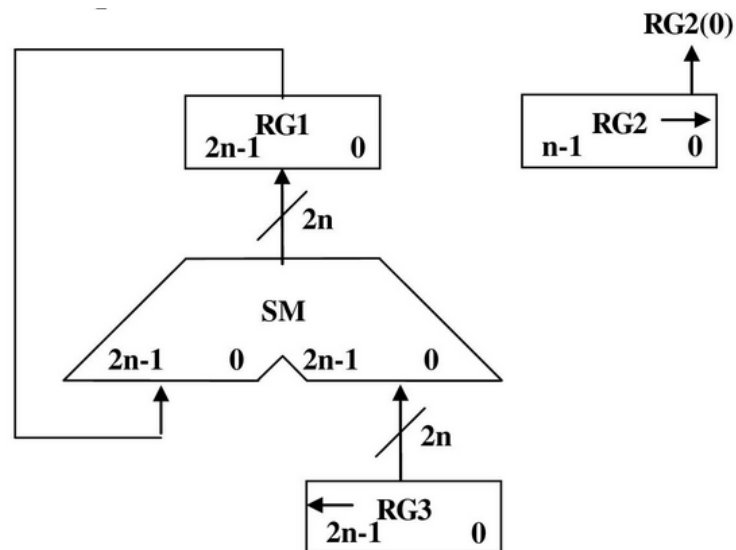


Рисунок 2.2.1- Операційна схема.

2.2.3 Змістовний мікроалгоритм:

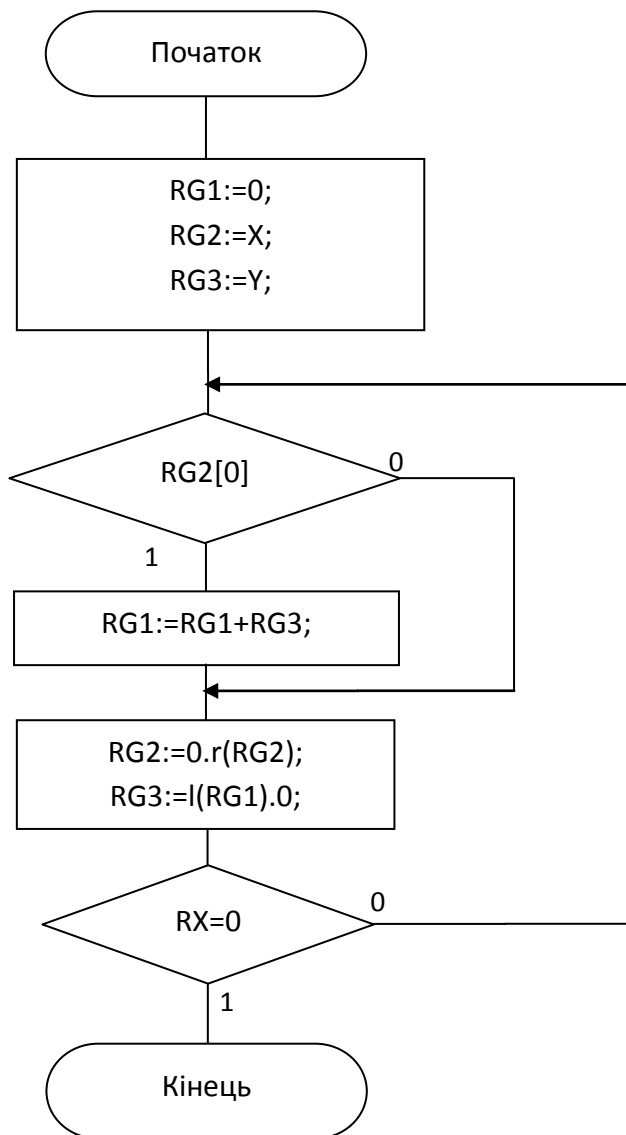


Рисунок 2.2.2 - Змістовний мікроалгоритм.

2.2.4 Таблиця станів регістрів:

Таблиця 2.2.1-Таблиця станів регістрів.

№	RG1	RG3 ←	RG2 →
ПС	0	0000000000000000101010111000011	100110111000101
1	+0000000000000000101010111000011 =0000000000000000101010111000011	00000000000000001010101110000110	010011011100010
2	0000000000000000101010111000011	000000000000000010101011100001100	001001101110001
3	+000000000000000010101011100001100 =000000000000000011010110011001111	0000000000000000101010111000011000	000100110111000
4	000000000000000011010110011001111	0000000000001010101110000110000	000010011011100
5	000000000000000011010110011001111	0000000000010101011100001100000	000001001101110
6	000000000000000011010110011001111	0000000000101010111000011000000	000000100110111

7	+000000000101010111000011000000 =000000000101110001110110001111	000000001010101110000110000000	000000010011011
8	+000000001010101110000110000000 =000000010000011111111100001111	000000010101011100001100000000	000000001001101
9	+000000010101011100001100000000 =000000100101111100001000001111	000000101010111000011000000000	000000000100110
10	000000100101111100001000001111	000001010101110000110000000000	000000000010011
11	+000001010101110000110000000000 =000001111011101100111000001111	000010101011100001100000000000	000000000001001
12	+000010101011100001100000000000 =000100100111001110011000001111	000101010111000011000000000000	000000000000100
13	000100100111001110011000001111	001010101110000110000000000000	000000000000010
14	000100100111001110011000001111	010101011100001100000000000000	000000000000001
15	+010101011100001100000000000000 =011010000011011010011000001111	101010111000011000000000000000	000000000000000

2.2.5 Функціональна схема:

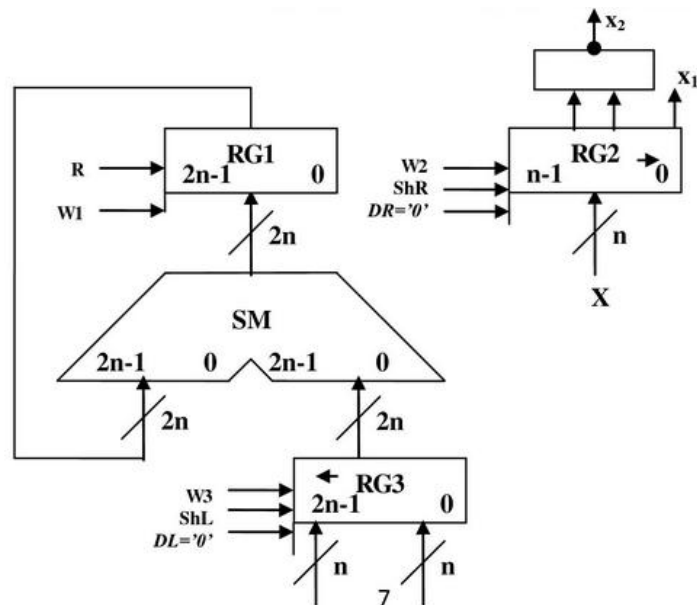


Рисунок 2.2.3- Функціональна схема.

2.2.6 Закодований мікроалгоритм

Таблиця 2.2.2-Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛУ	Позначення
RG1:=0	R	RG2[0]	X1
RG2:=X	W2	RG2=0	X2
RG3:=Y	W3		
RG1:=RG1+RG3	W1		
RG2:=0.r(PG2)	ShR		
RG3:=l(RG3).0	ShL		

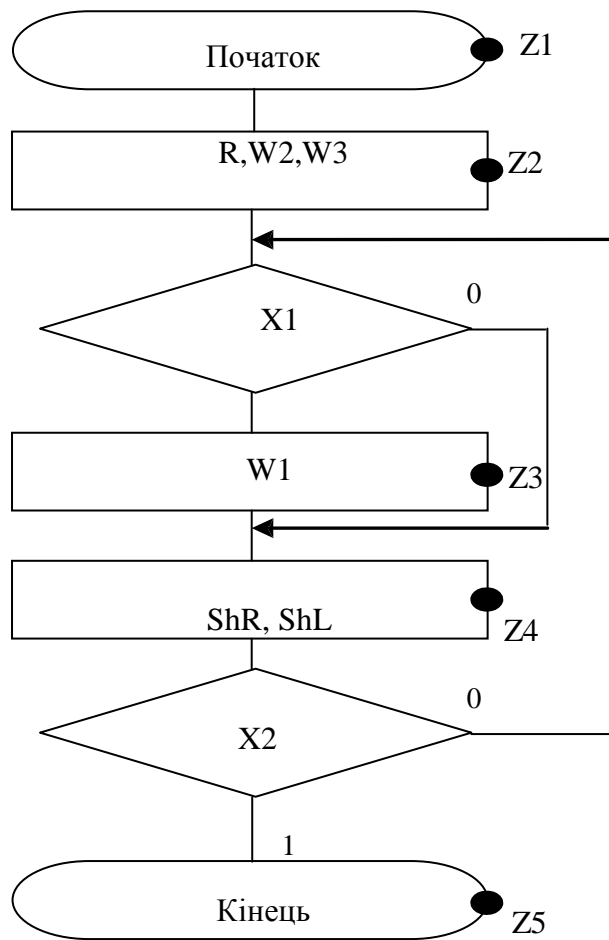


Рисунок 2.2.4-Закодований мікроалгоритм.

2.2.7 Граф управляючого автомата Мура з кодами вершин:

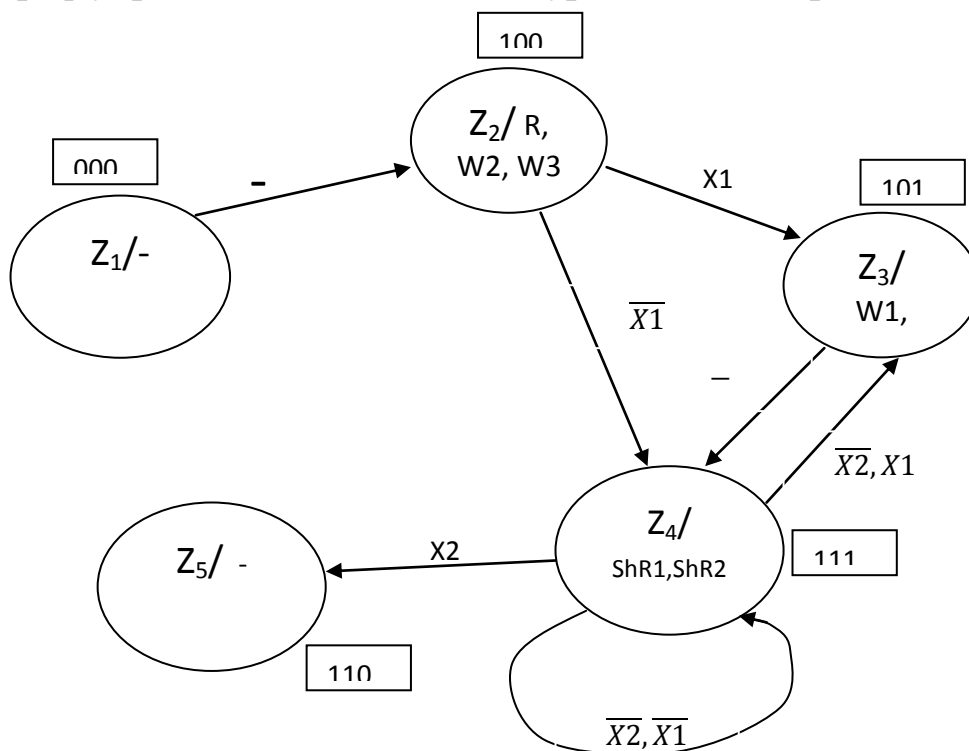


Рисунок 2.2.5 - Граф автомата Мура

2.2.8 Обробка порядків:

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

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

2.2.9 Нормалізація результату:

Отримали результат: 011010000011011010011000001111

Знак мантиси: $1 \oplus 0 = 1$.

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

Порядок зменшуємо на 1:

$$11010000011011010011000001111; P_z=12;$$

Запишемо нормалізований результат:

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

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

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

$$Z = \sum_{i=1}^n 2Z_{i-1} + YX_i 2^{-n};$$

$$Z_0 = 0;$$

$$Y_0 = 0$$

2.3.2 Операційна схема:

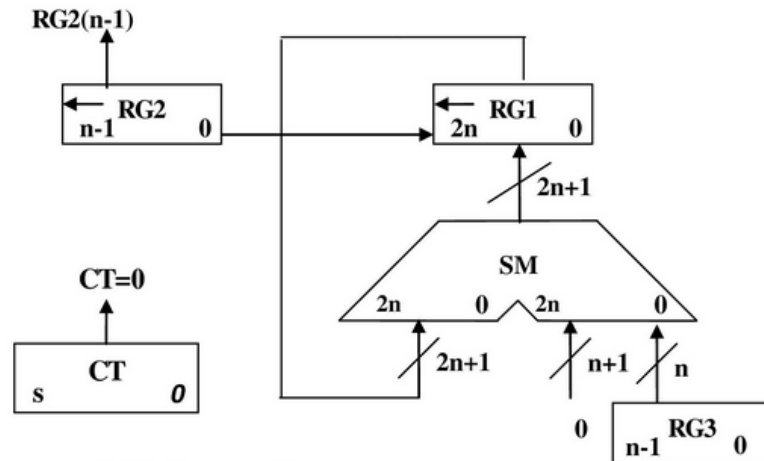


Рисунок 2.3.1 - Операційна схема

2.3.3 Змістовний мікроалгоритм:

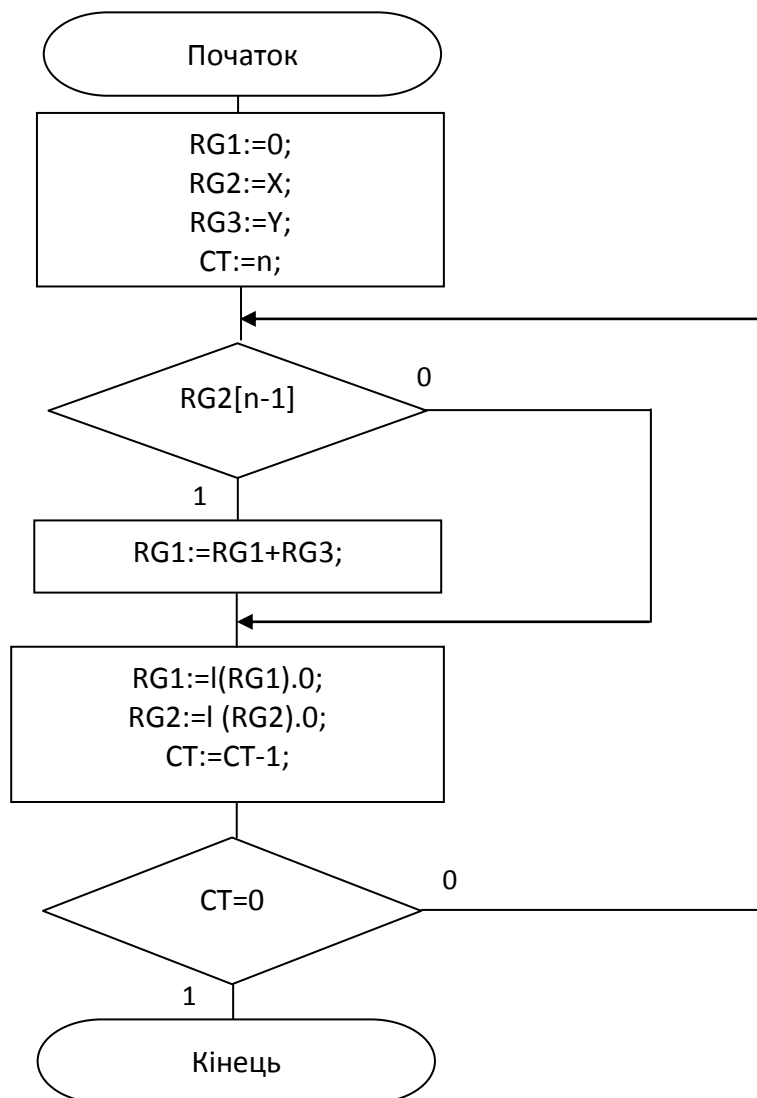


Рисунок 2.3.2 - Змістовний мікроалгоритм.

2.3.4 Таблиця станів регістрів:

Таблиця 2.3.1- Таблиця станів регістрів

№	RG1 ←	RG2 ←	RG3	CT
пс	00000000000000000000000000000000	100110111000101	101010111000011	1111
1	+0000000000000000101010111000011 =0000000000000000101010111000011 00000000000000001010101110000110	001101110001010		1110
2	000000000000000010101011100001100	011011100010100		1101
3	0000000000000000101010111000011000	110111000101000		1100
4	+0000000000000000101010111000011 =0000000000000110000001111011011 0000000000001100000011110110110	101110001010000		1011
5	+0000000000000000101010111000011 =0000000000001100101110101111001 0000000000011001011101011110010	011100010100000		1010
6	000000000110010111010111100100	111000101000000		1001
7	+0000000000000000101010111000011 =000000000110011100101110100111 0000000011001110010111101001110	110001010000000		1000
8	+0000000000000000101010111000011 =000000001100111110110100010001 000000011001111101101000100010	100010100000000		0111
9	+0000000000000000101010111000011 =000000011010000010111111100101 000000110100000101111111001010	000101000000000		0110
10	000001101000001011111110010100	001010000000000		0101
11	000011010000010111111100101000	010100000000000		0100
12	000110100000101111111001010000	101000000000000		0011
13	+0000000000000000101010111000011 =000110100000110101010000010011 001101000001101010100000100110	010000000000000		0010
14	011010000011010101000001001100	100000000000000		0001
15	+0000000000000000101010111000011 =011010000011011010011000001111	000000000000000		0

2.3.5 Функціональна схема:

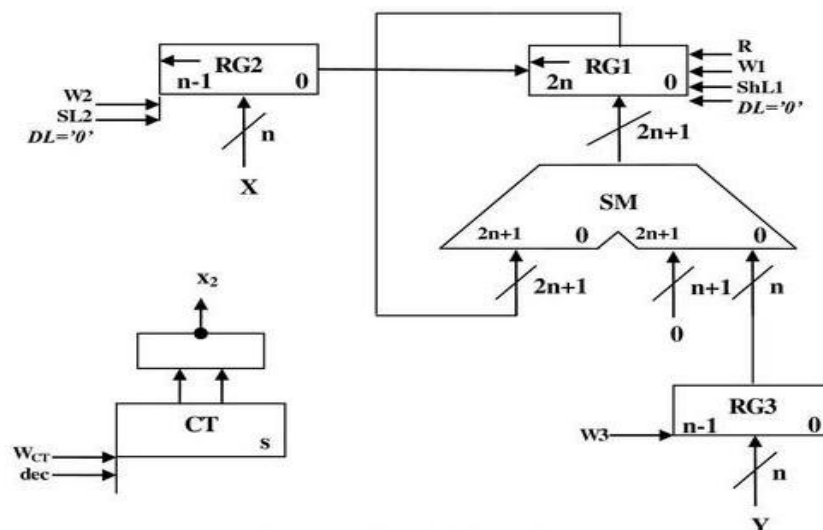


Рисунок 2.3.3 - Функціональна схема.

2.3.6 Закодований мікроалгоритм:

Таблиця 2.3.2-Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛЮ	Позначення
RG1:=0	R	RG2[n-1]	X1
RG2:=X	W2	CT=0	X2
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		

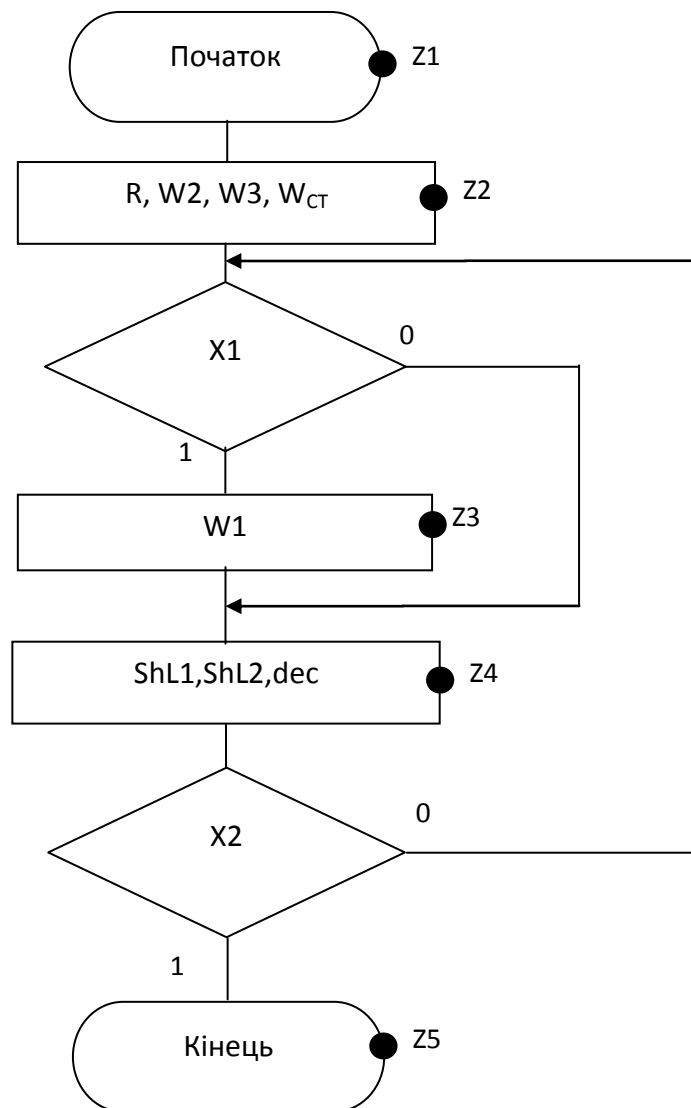


Рисунок 2.3.4-Закодований мікроалгоритм.

2.3.7 Граф управляючого автомата Мура з кодами вершин:

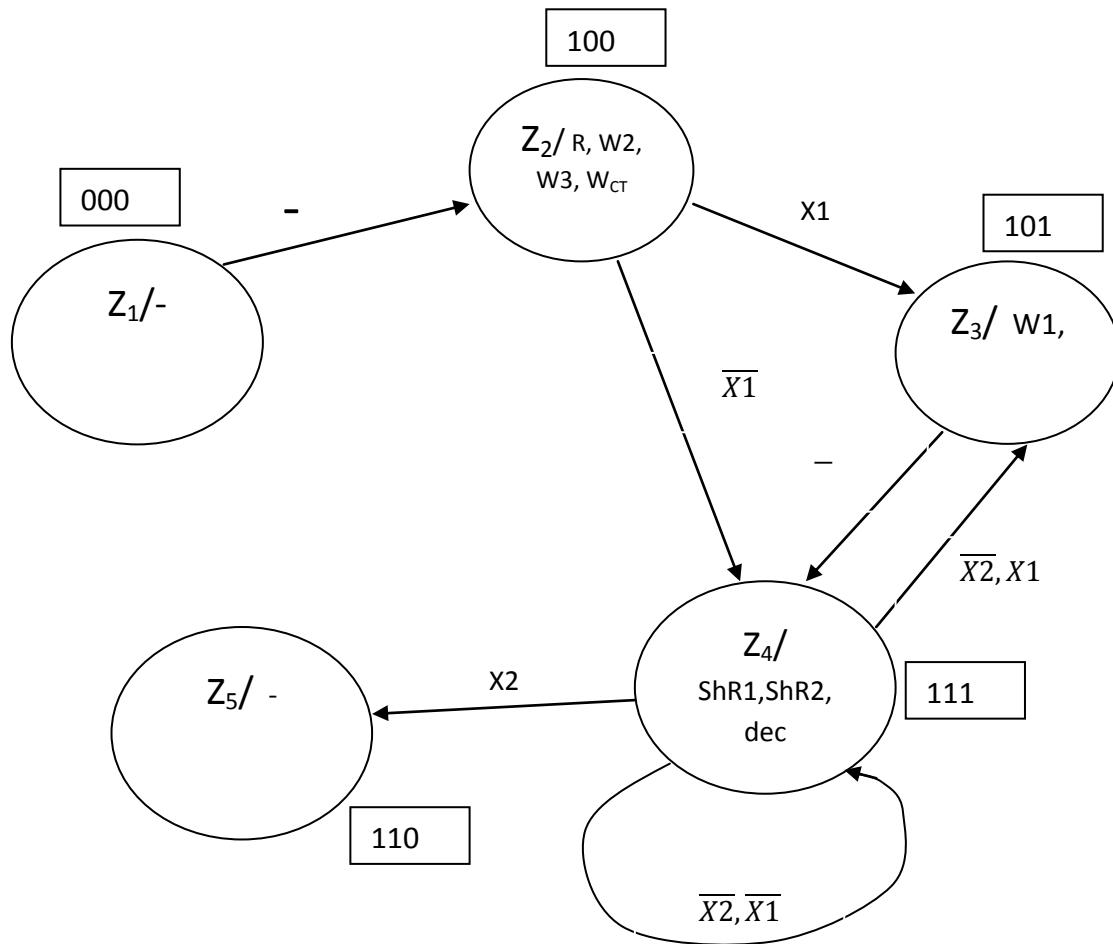


Рисунок 2.3.5 - Граф автомата Мура

2.3.8 Обробка порядків:

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

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

2.3.9 Нормалізація результату:

Отримали результат: 011010000011011010011000001111

Знак мантиси: $1 \oplus 0 = 1$.

Робимо здви́г результату влі́во, доки у першому розряді не буде одиниця,

порядок зменшуємо на 1:

11010000011011010011000001111; $P_z=12$;

Запишемо нормалізований результат:

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

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

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

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

2.4.2 Операційна схема:

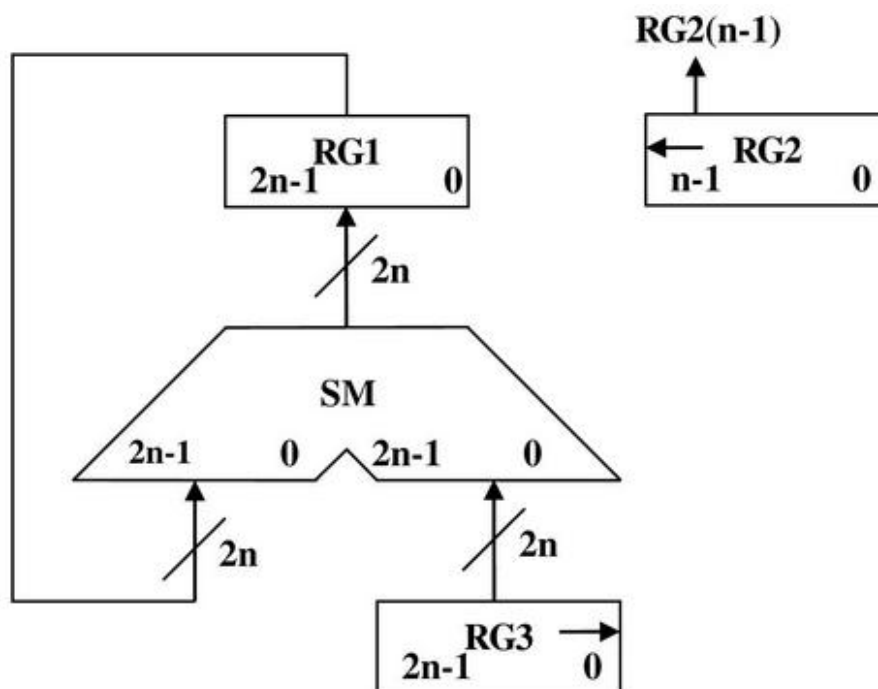


Рисунок 2.4.1- Операційна схема

2.4.3 Змістовний мікроалгоритм:

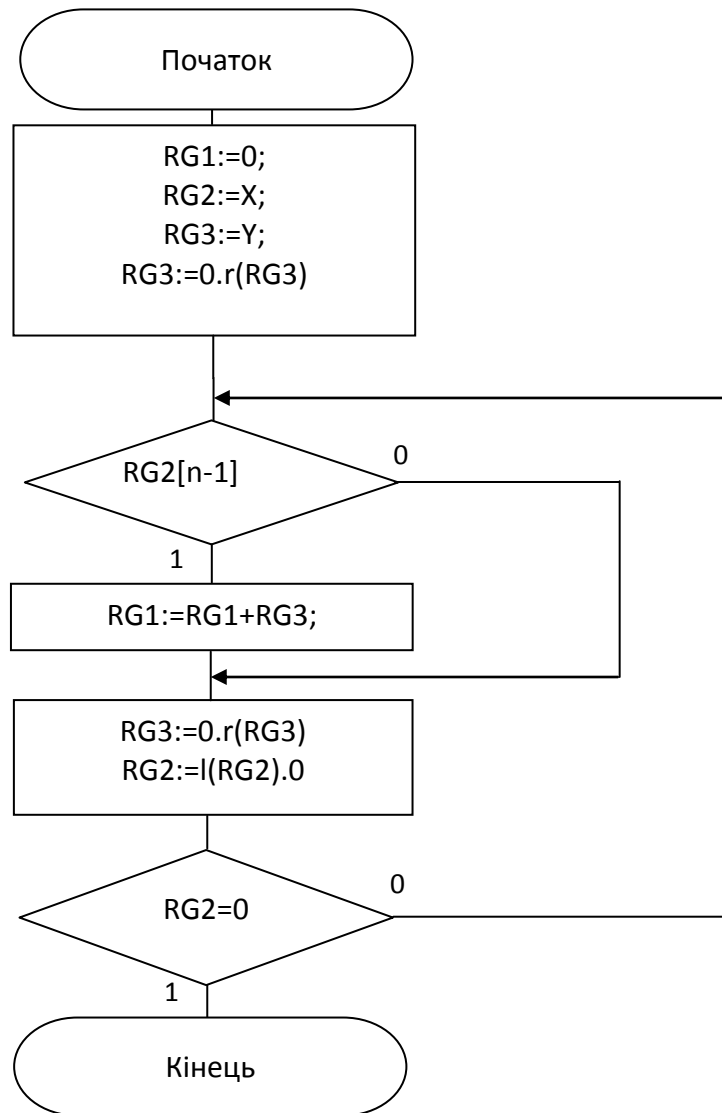


Рисунок 2.4.2 - Змістовний мікроалгоритм.

2.4.4 Таблиця станів регістрів:

Таблиця 2.4.1- Таблиця станів регістрів

№	RG1	RG3 →	RG2 ←
ПС	00000000000000000000000000000000	01010101110000110000000000000000	100110111000101
1	+01010101110000110000000000000000 =01010101110000110000000000000000	00101010111000011000000000000000	001101110001010
2	01010101110000110000000000000000	00010101011100001100000000000000	011011100010100
3	01010101110000110000000000000000	00001010101110000110000000000000	110111000101000
4	+00001010101110000110000000000000 =01100000011110110110000000000000	00000101010111000011000000000000	101110001010000
5	+00000101010111000011000000000000 =01100101110101111001000000000000	00000010101011100001100000000000	011100010100000

6	011001011101011110010000000000	0000000010101011100001100000000	1110001010000000
7	+000000010101011100001100000000 =011001110010111010011100000000	0000000001010101110000110000000	1100010100000000
8	+000000001010101110000110000000 =011001111101101000100010000000	0000000000101010111000011000000	1000101000000000
9	+000000000101010111000011000000 =011010000010111111100101000000	0000000000010101011100001100000	0001010000000000
10	011010000010111111100101000000	0000000000001010101110000110000	0010100000000000
11	011010000010111111100101000000	0000000000000101010111000011000	0101000000000000
12	011010000010111111100101000000	0000000000000010101011100001100	1010000000000000
13	+000000000000010101011100001100 =011010000011010101000001001100	0000000000000001010101110000110	0100000000000000
14	011010000011010101000001001100	0000000000000000101010111000011	1000000000000000
15	+0000000000000000101010111000011 =011010000011011010011000001111	0000000000000000010101011100001	0000000000000000

2.4.5 Функціональна схема:

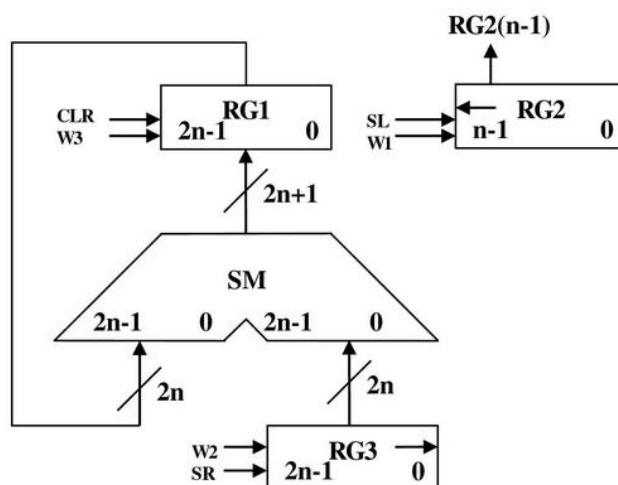


Рисунок 2.4.3 - Функціональна схема.

2.4.6 Закодований мікроалгоритм

Таблиця 2.4.2-Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛЮ	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		

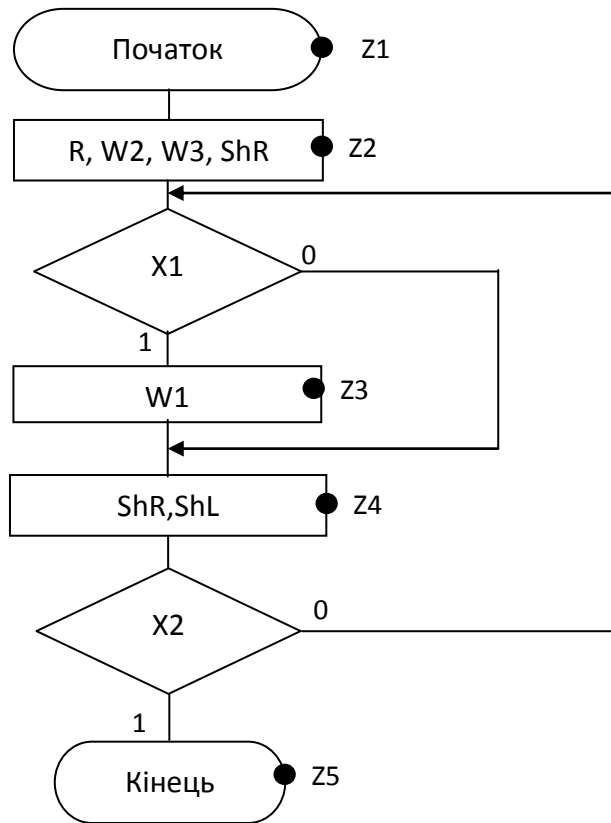


Рисунок 2.4.4-Закодований мікроалгоритм.

2.4.7 Граф управляючого автомата Мура з кодами вершин:

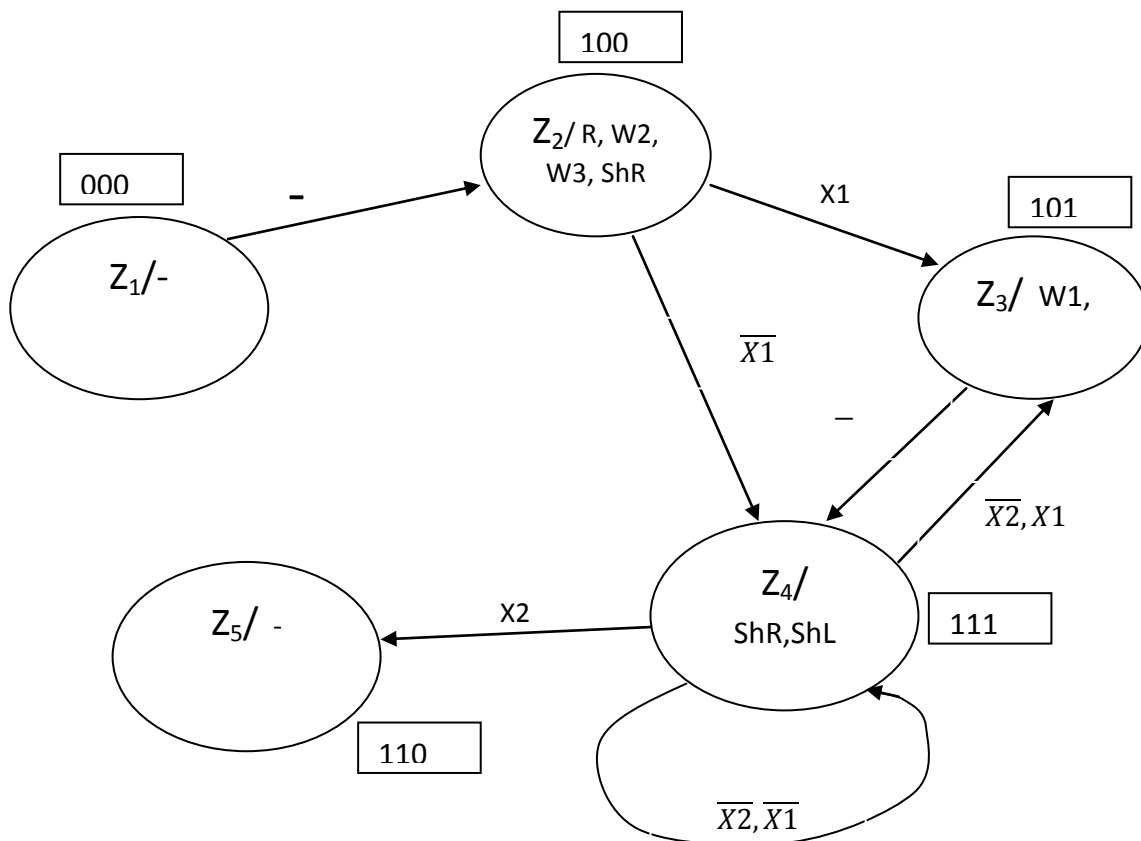


Рисунок 2.4.5 - Граф автомата Мура

2.4.8 Обробка порядків:

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

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

2.4.9 Нормалізація результату:

Отримали результат: 011010000011011010011000001111

Знак мантиси: $1 \oplus 0 = 1$.

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

Порядок понижаємо на 1:

$$11010000011011010011000001111; P_z=12;$$

Запишемо нормалізований результат:

0	0	0	0	1	1	0	0	1	1	1	0	1	0	0	0	0	1	1	0	1	1	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 СМ 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:

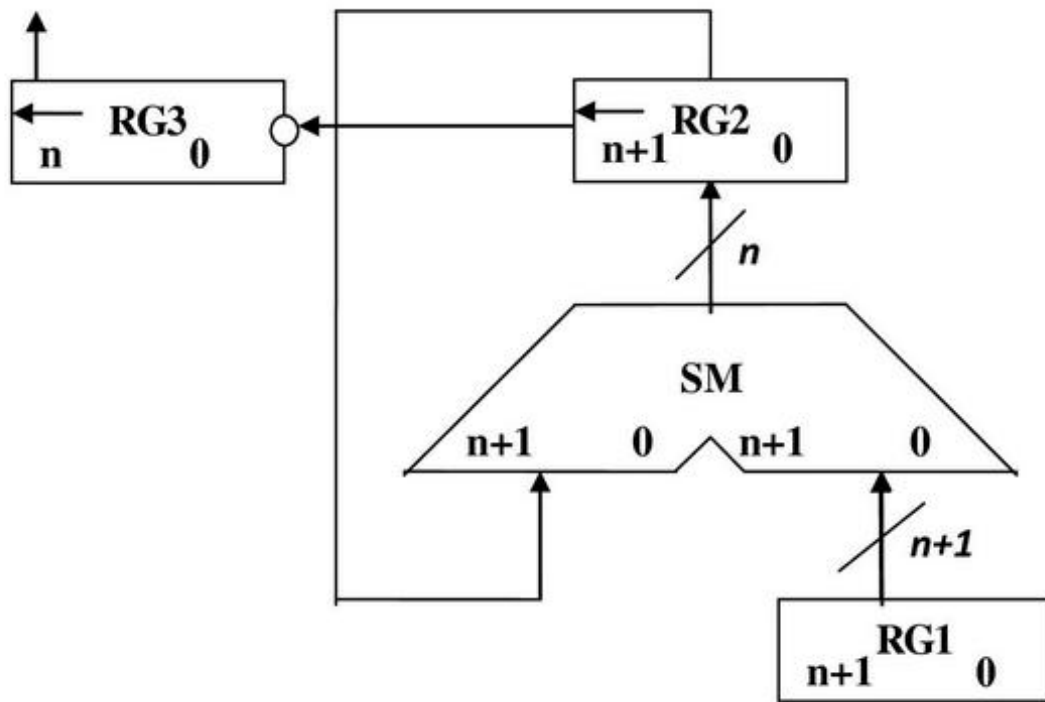


Рисунок 2.5.1- Операційна схема

2.5.3 Змістовний мікроалгоритм:

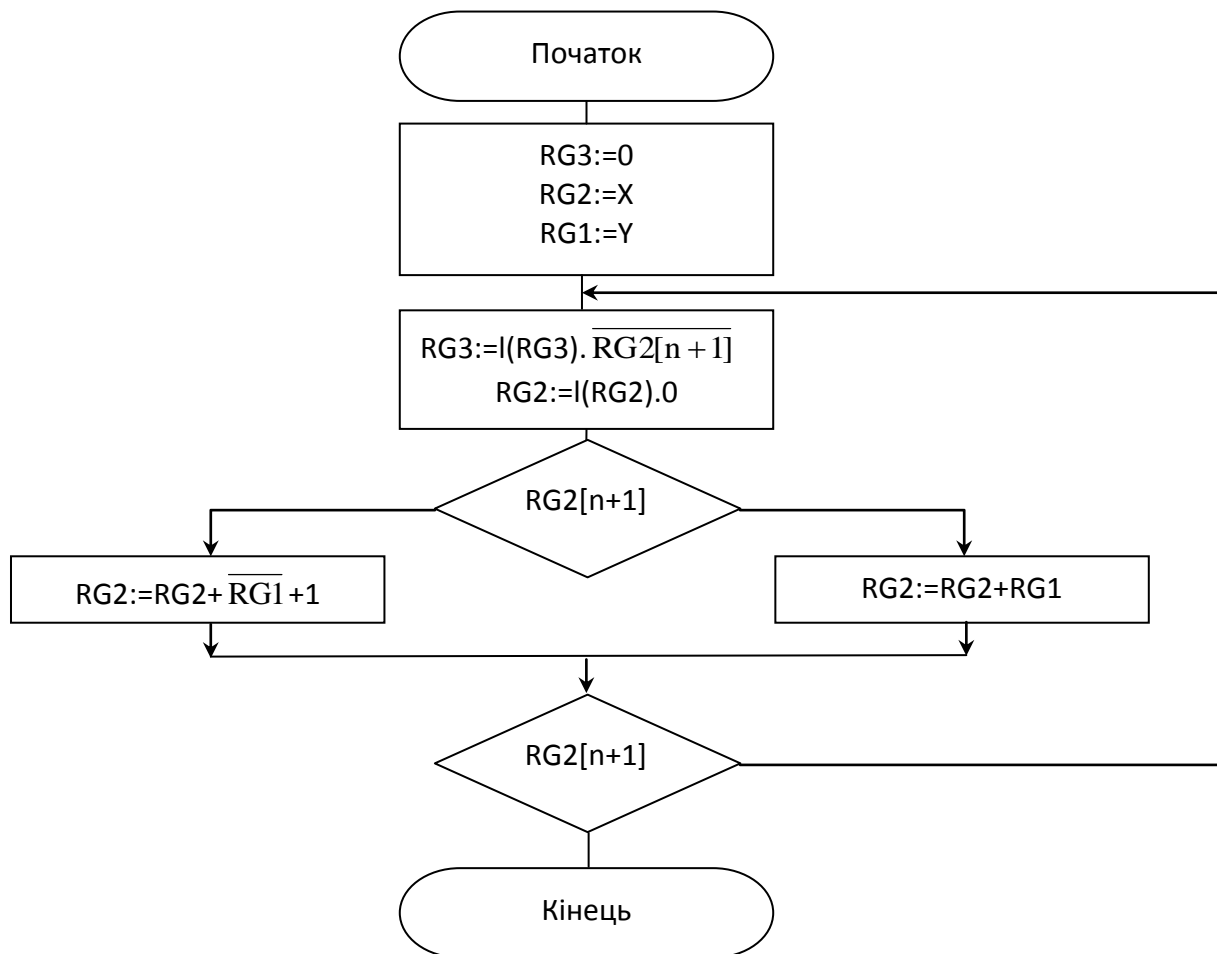


Рисунок 2.5.2-Змістовний мікроалгоритм

2.5.4 Таблиця станів реєстрів:

№	RG3 (Z)	RG2 (X)	RG1 (Y)
пс	0000000000000000	00100110111000101	00101010111000011
1	0000000000000001	01001101110001010 +11010101000111101 =00100010111000111	
2	0000000000000011	01000101110001110 +11010101000111101 =00011010111001011	
3	0000000000000111	00110101110010110 +11010101000111101 =00001010111010011	
4	0000000000001111	00010101110100110 +11010101000111101 =11101010111100011	
5	0000000000011110	11010101111000110 +00101010111000011 =00000000110001001	
6	0000000000111101	00000001100010010 +11010101000111101 =11010110101001111	
7	0000000001111010	10101101010011110 +00101010111000011 =11011000001100001	
8	0000000011110100	10110000011000010 +00101010111000011 =11011011010000101	
9	0000000111101000	10110110100001010 +00101010111000011 =11100001011001101	
10	0000001111010000	11000010110011010 +00101010111000011 =11101101101011101	
11	0000011110100000	11011011010111010 +00101010111000011 =00000110001111101	
12	0000111101000001	00001100011111010 +11010101000111101 =11100001100110111	
13	0001111010000010	11000011001101110 +00101010111000011 =11101110000110001	
14	0011110100000100	11011100001100010 +00101010111000011 =00000111000100101	
15	0111101000001001	00001110001001010 +11010101000111101 =11100011010000111	
16	1111010000010010	11000110100001110 +00101010111000011 =11110001011010001	

2.5.5 Функціональна схема:

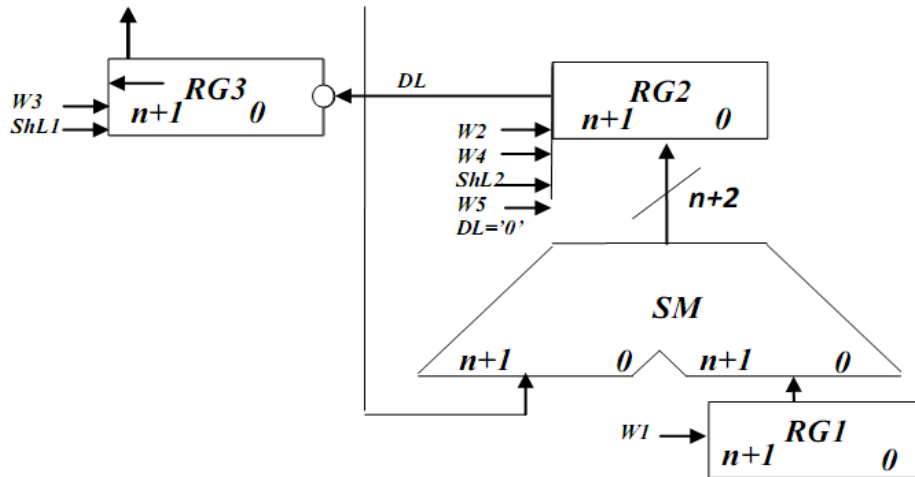


Рисунок 2.5.3 – Функціональна схема

2.5.6 Закодований мікроалгоритм

Таблиця 2.5.2-Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
МО	УС	ЛУ	Позначення
RG3:=0	W3	RG2[n-1]	X1
RG2:=X;	W2	RG2=0	X2
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		

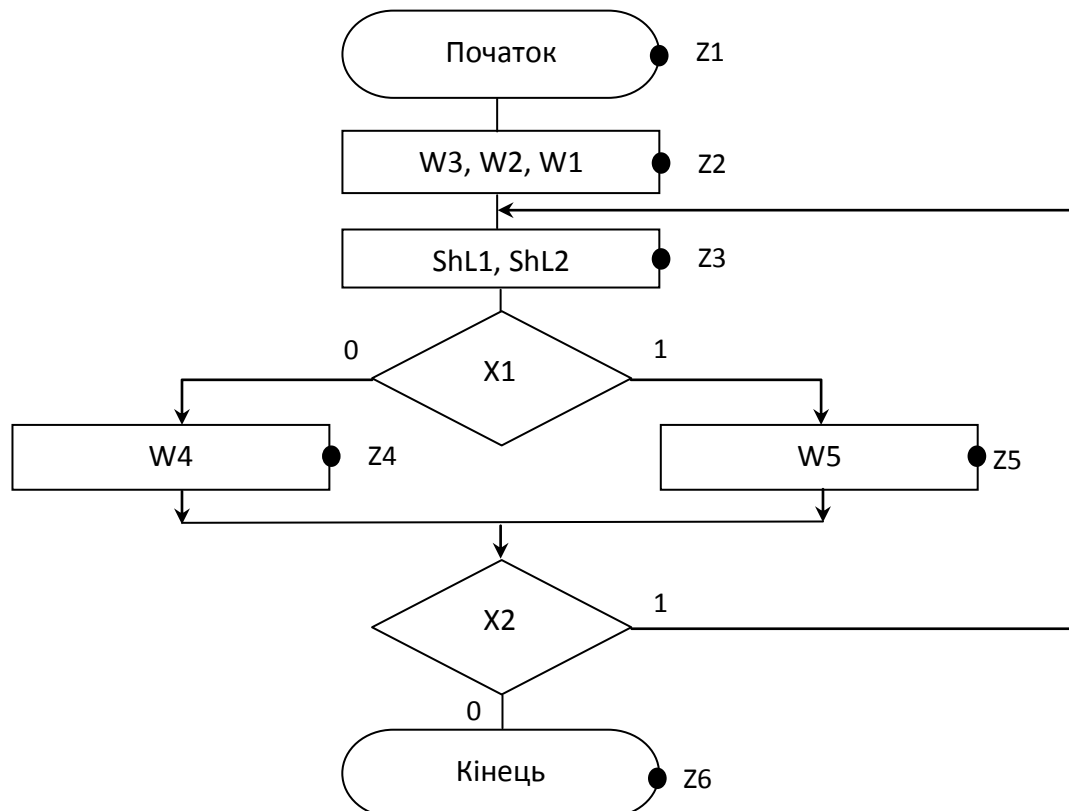


Рисунок 2.5.4-Закодований мікроалгоритм.

2.5.7 Граф управляющего автомата Мура з кодами вершин:

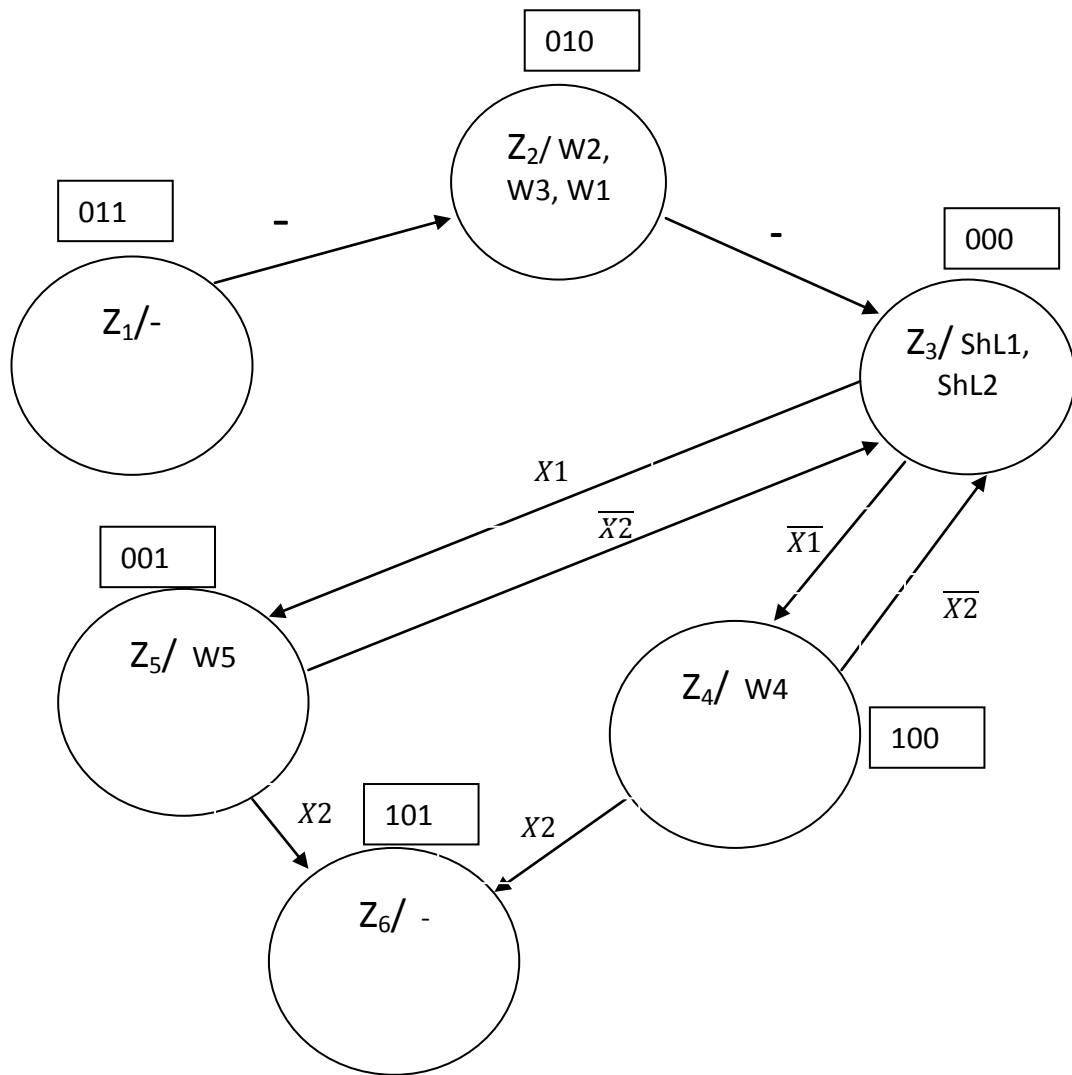


Рисунок 2.5.5 - Граф управляющего автомата.

2.5.8 Обработка порядків:

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

В моєму випадку $P_x=8$; $P_y=5$; $P_z=3$;

2.5.8 Нормалізація результату:

Отримали результат: 1111010000010010

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

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

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 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

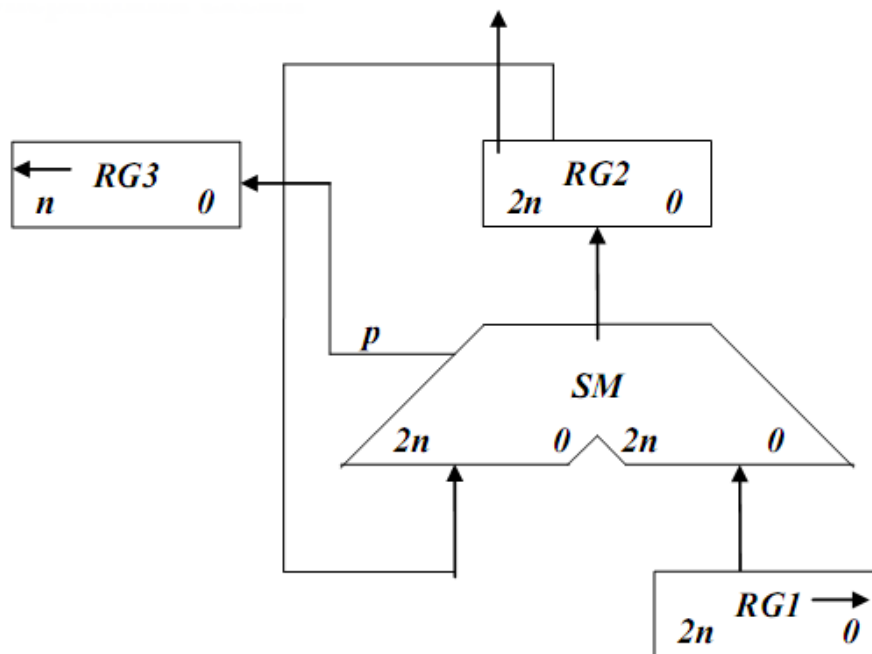


Рисунок 2.6.1-Операційна схема

2.6.3 Змістовний мікроалгоритм

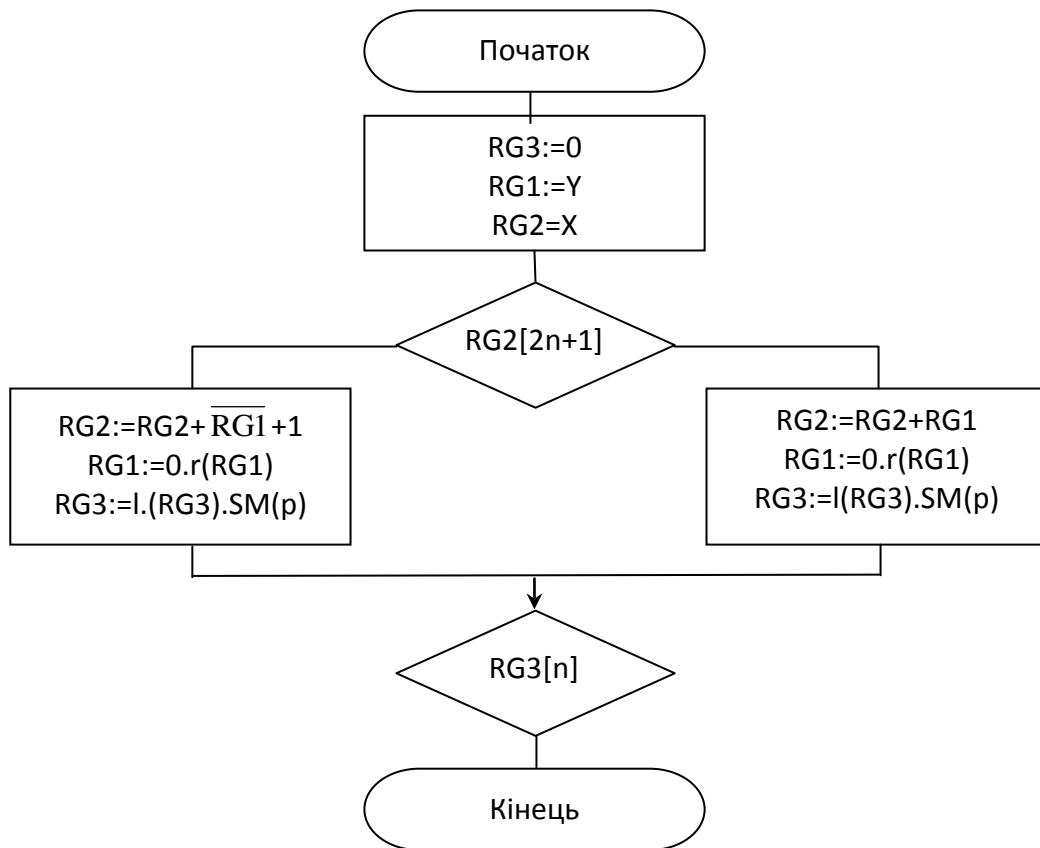


Рисунок 2.6.2-Змістовний мікроалгоритм

2.6.4 Таблиця станів регістрів

Таблиця 2.6.1- Таблиця станів регістрів

№	RG3 (Z)	RG2 (X)	RG1 (Y)
ПС			
	000000000000000001	010011011100010100000000000000	001010101110000110000000000000
1	000000000000000011	010011011100010100000000000000 +110101010001111010000000000000 =001000101110001110000000000000	000101010111000011000000000000
2	000000000000000111	001000101110001110000000000000 +111010101000111101000000000000 =000011010111001011000000000000	000010101011100001100000000000
3	000000000000001111	000011010111001011000000000000 +111101010100011110100000000000 =000000101011101001100000000000	000001010101110000110000000000
4	00000000000011110	000000101011101001100000000000 +111110101010001111010000000000 =111111010101110001100000000000	000000101010111000011000000000
5	00000000000111101	111111010101110001100000000000 +000000101010111000011000000000 =000000000000110001001000000000	000000010101011100001100000000
6	0000000001111010	000000000000110001001000000000 +111111101010100011110100000000 =111111101011010100111100000000	000000001010101110000110000000
7	0000000011110100	111111101011010100111100000000 +000000001010101110000110000000 =111111110110000011000010000000	000000000101010111000011000000

8	0000000111101000	111111110110000011000010000000 +000000000101010111000011000000 =111111111011011010000101000000	000000000010101011100001100000
9	0000001111010000	111111111011011010000101000000 +000000000010101011100001100000 =111111111100001011001101000000	000000000001010101110000110000
10	0000011110100000	111111111100001011001101000000 +000000000001010101110000110000 =111111111110110110101110100000	000000000000101010111000011000
11	0000111101000001	111111111110110110101110100000 +000000000000101010111000011000 =000000000000000110001111010000	000000000000010101011100001100
12	0001111010000010	000000000000000110001111010000 +111111111111010101000111101000 =111111111111100001100110111000	0000000000000001010101110000110
13	0011110100000100	111111111111100001100110111000 +0000000000000001010101110000110 =11111111111111011100001100010	0000000000000000101010111000011
14	0111101000001001	11111111111111011100001100010 +000000000000000101010111000011 =00000000000000000111000100101	0000000000000000010101011100001
15	1111010000010010	0000000000000000000111000100101 +11111111111111101010100011111 =11111111111111110001101000100	000000000000000001010101110000

2.6.5 Функціональна схема з відображенням управляючих сигналів

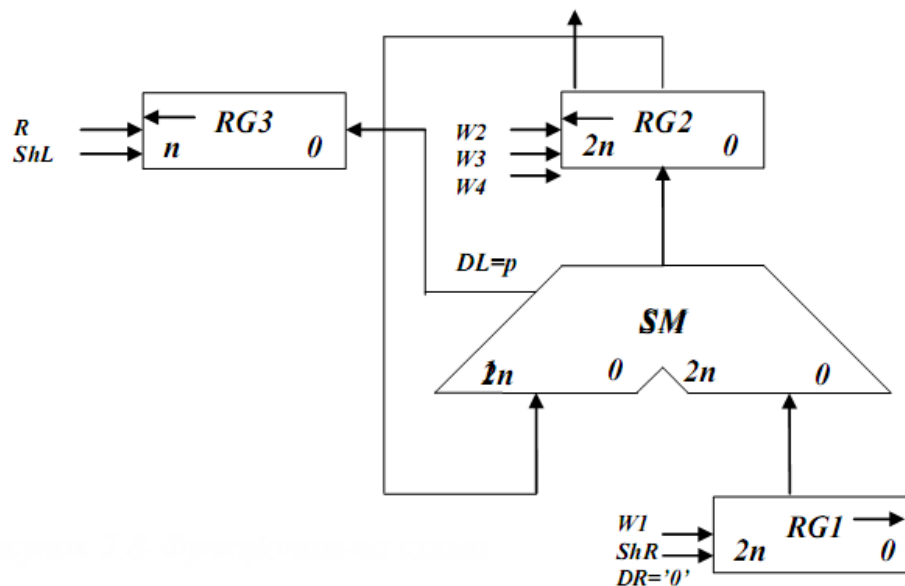


Рисунок 2.6.3-Функціональна схема

2.6.6 Закодований мікроалгоритм

Таблиця 2.6.2- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій			Таблиця кодування логічних умов
МО	УС	ЛУ	Позначення
RG3:=0 RG1:=Y RG2:=X RG2:=RG2+RG1	R W1 W2 W3	RG2[2n+1] RG3[n]	X1 X2

RG1:=0.r(RG1) RG3:=l(RG3).SM(p) RG2:==RG2+RG1+1	ShR ShL W4		
---	------------------	--	--

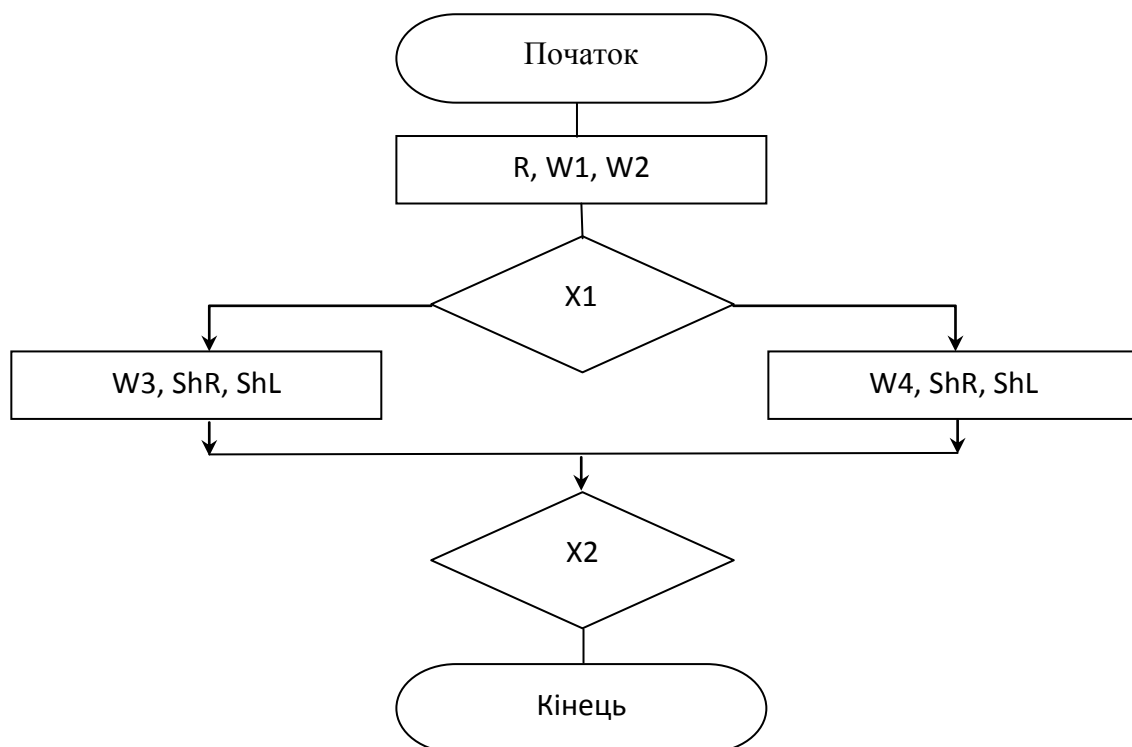


Рисунок 2.6.4- Закодований мікроалгоритм

2.6.7 Граф управляючого автомата Мура з кодами вершин

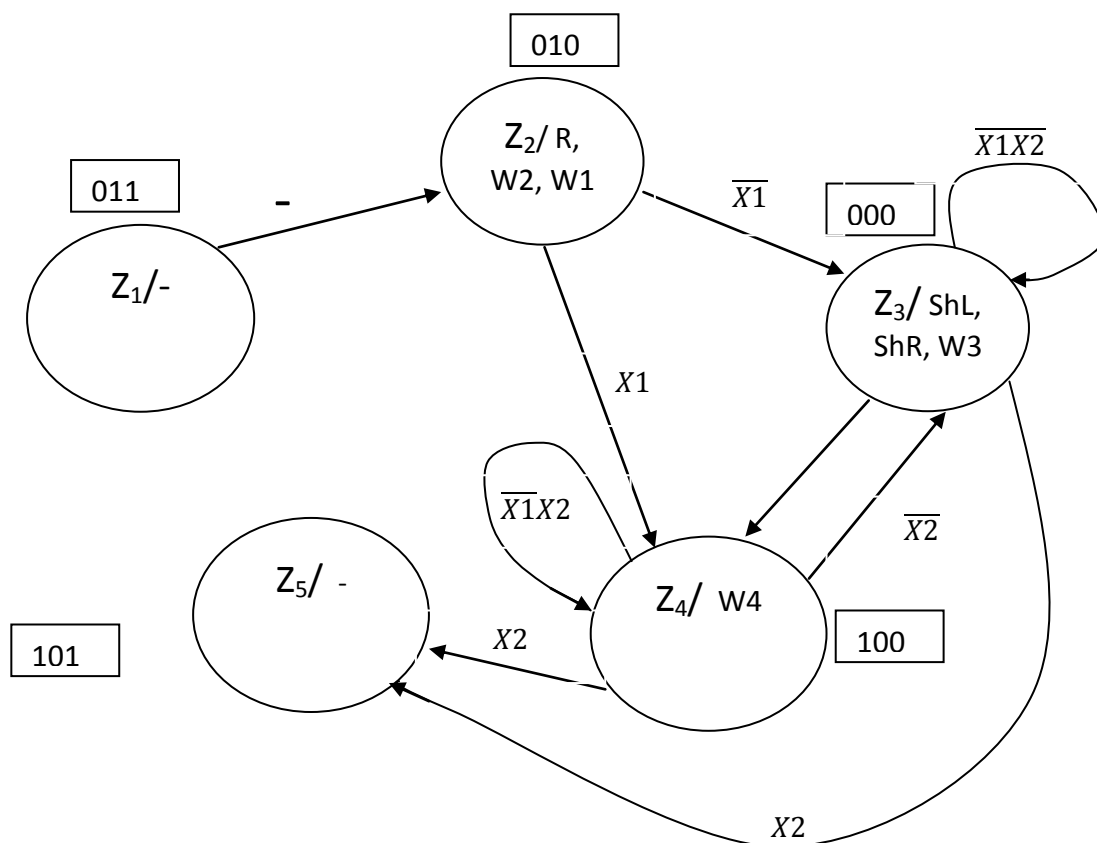


Рисунок 2.6.5- Граф автомата Мура

2.6.8 Обробка порядків:

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

В моєму випадку $P_x=8$; $P_y=5$; $P_z=3$;

2.6.9 Нормалізація результату:

Отримали результат: 1111010000010010

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

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

2.7. Операція додавання та віднімання 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. Віднімання будемо виконувати як операцію додавання але змінивши знак від'ємника на протилежний $Z = X - Y = X + (-Y)$

1. Порівняння порядків.

$$P_x = +8_{10} = +1000_2$$

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

$$P_x > P_y \Rightarrow$$

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

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
0,101010111000011	11	Початковий стан
0,010101011100001	10	$M_y = 0r(M_y)$; $\Delta = \Delta - 1$
0,001010101110000	01	$M_y = 0r(M_y)$; $\Delta = \Delta - 1$
0,000101010111000	00	$M_y = 0r(M_y)$; $\Delta = \Delta - 1$

3. Додавання мантис у модифікованому ДК.

$$X_{\text{мдк}} = 11.011001000111011$$

$$Y_{\text{мдк}} = 00.000101010111000$$

Таблиця 2.7.2-Додавання мантис(для додавання)

M_X	1	1,	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1
M_Y	0	0,	0	0	0	1	0	1	0	1	0	1	1	1	0	0	0
M_Z	1	1,	0	1	1	1	1	0	0	1	1	1	1	0	0	1	1

$$Z_{ПК} = 1.100001100001101$$

4. Віднімання мантис у модифікованому ДК.

$$X_{мдк} = 11.011001000111011$$

$$Y_{мдк} := -Y_{мдк} = 11.111010101001000$$

Таблиця 2.7.3-Додавання мантис(для віднімання)

M_X	1	1,	0	1	1	0	0	1	0	0	0	1	1	1	0	1	1
M_Y	1	1,	1	1	1	0	1	0	1	0	1	0	0	1	0	0	0
M_Z	1	1	0	1	0	0	1	1	1	1	0	0	0	0	0	1	1

$$Z_{ПК} = 1.101100001111100$$

5. Нормалізація результату (В ПК).

Для даних результатів додавання та віднімання нормалізація не потрібна.

2.7.2 Операційна схема

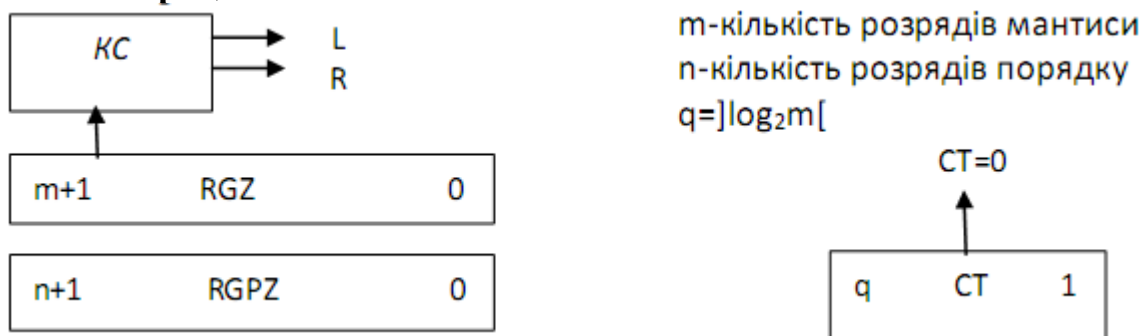


Рисунок 2.7.1-Операційна схема

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 Змістовний алгоритм

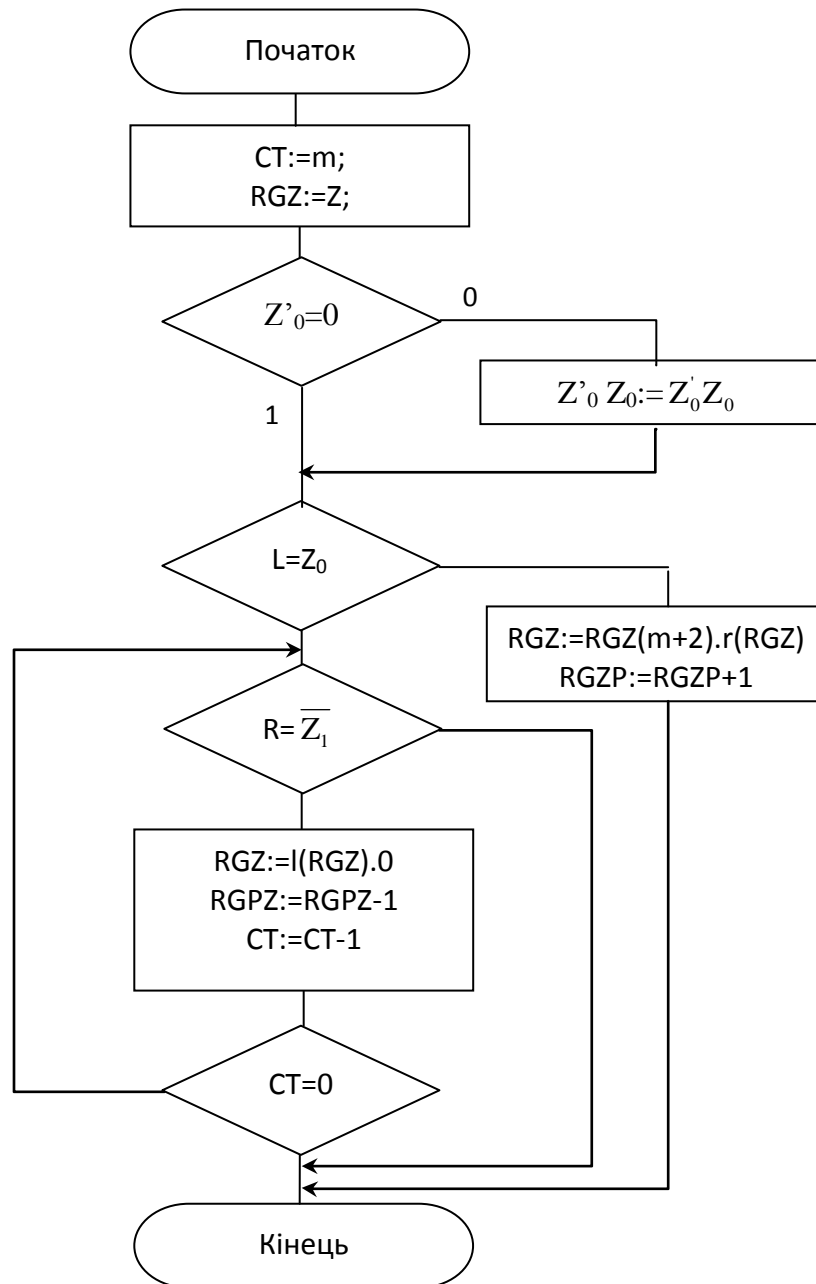


Рисунок 2.7.2-Змістовний мікроалгоритм

2.7.4 Таблиця станів регістрів

1) Додавання

Таблиця 2.7.5- Таблиця станів регістрів

№ такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001000	11.100001100001101	0	0	100	

2) Віднімання

Таблиця 2.7.6- Таблиця станів регістрів

№ такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001000	11.101100001111100	0	0	100	

2.7.5 Функціональна схема з відображенням керуючих сигналів

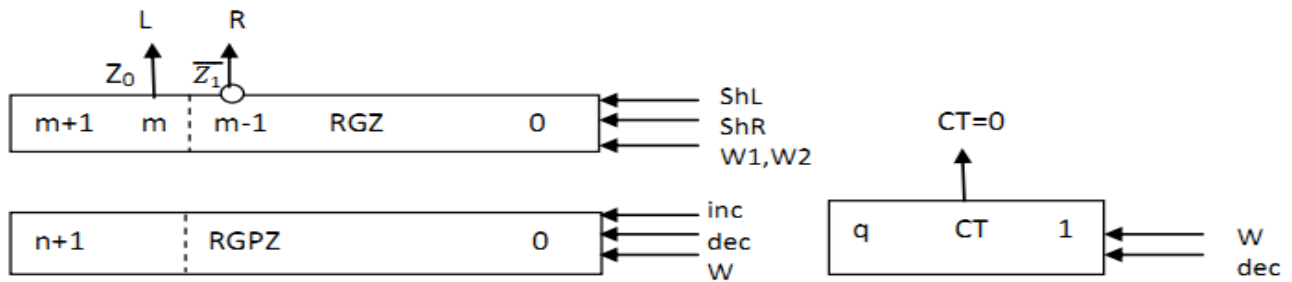
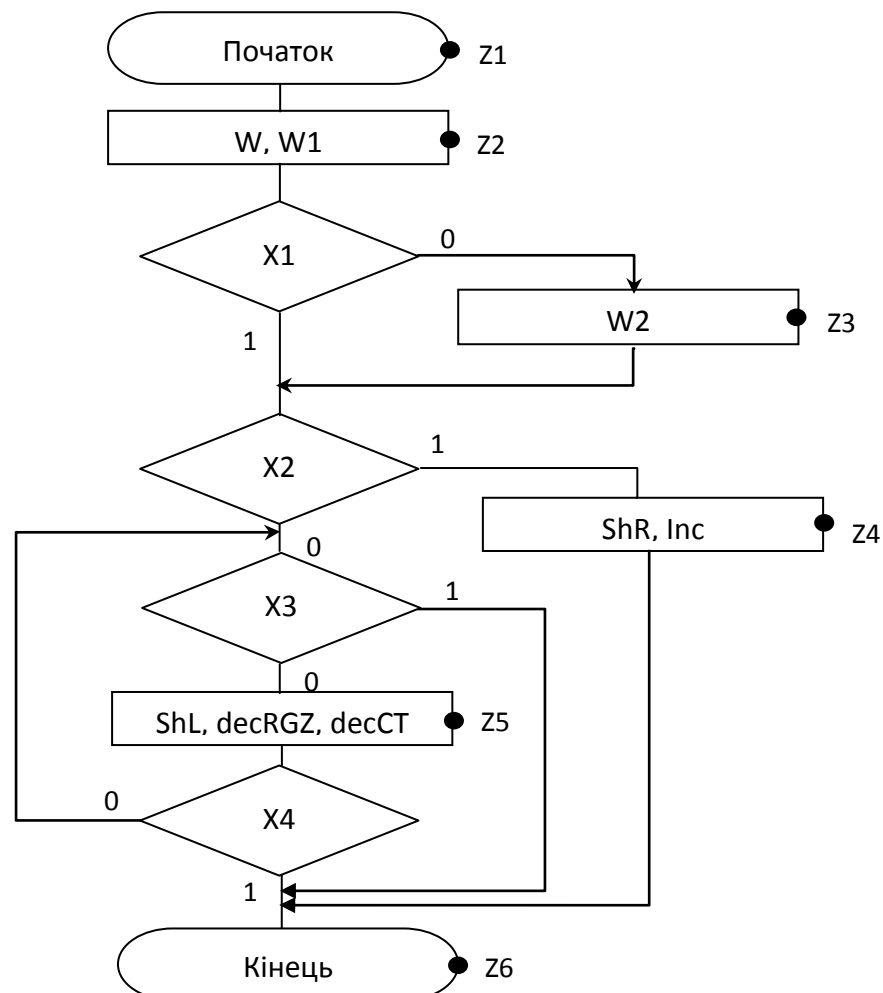


Рисунок 2.7.3 – Функціональна схема

2.7.6 Закодований мікроалгоритм

Таблиця 2.7.7– Таблиця кодування

Таблиця кодування мікрооперацій			Таблиця кодування логічних умов
МО	УС	ЛУ	Позначення
CT:=m	W	Z ₀ '=0	X1
RGZ:=Z	W1	L=Z ₀	X2
Z ₀ ' Z ₀ :=Z ₀ ' Z ₀	W2	R=Z ₁	X3
RGZ:=RGZ(m+2).r(RGZ)	ShR	CT=0	X4
RGPZ:=RGPZ+1	Inc		
RGZ:=l(RGZ).0	ShL		
RGPZ:=RGPZ-1	decRGZ		
CT:=CT-1	decCT		



2.7.7 Граф управляючого автомата Мура з кодами вершин

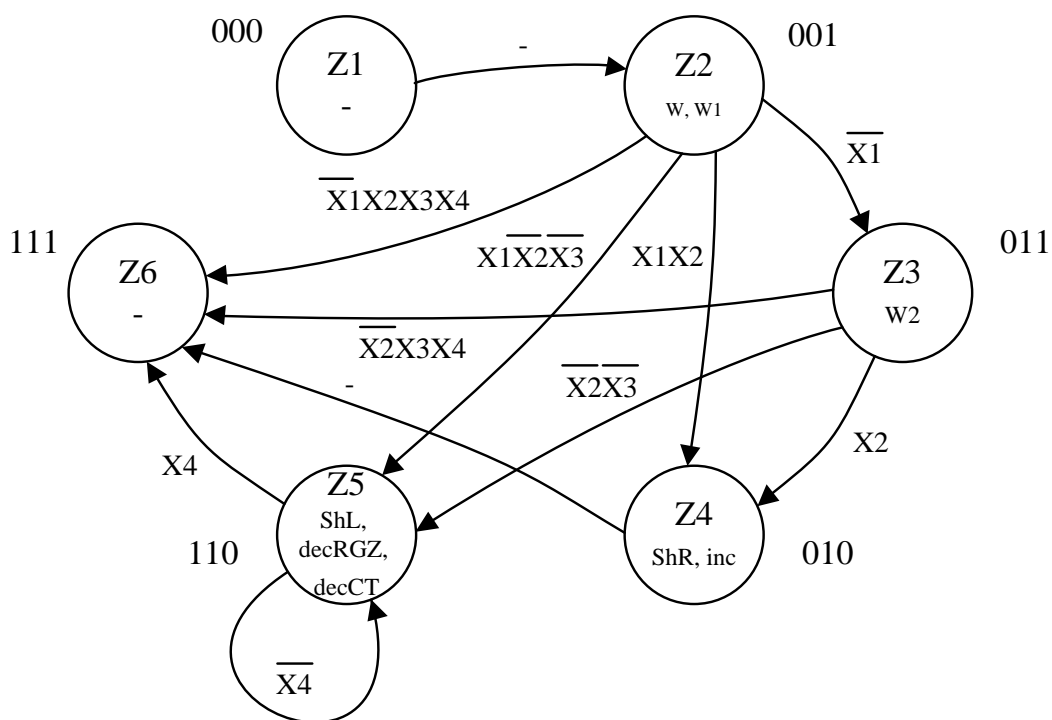


Рисунок 2.7.5 – Граф автомата Мура

2.7.8 Обробка порядків

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

2) $P_{X-Y} = 8_{10} = 1000_2$

2.7.9 Форма запису результату з плаваючою комою

1) Результат додавання $Z = X + Y$.

$$Z_{\text{пк}} = 1.100001100001101$$

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

$$M_z = 100001100110001_2$$

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

2) Результат віднімання $Z = X - Y$.

$$Z_{\text{пк}} = 1.101100001111100$$

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

$$M_z = 101100001111100_2$$

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

Завдання 3

$$x_3 x_2 x_1 = 001_2 = 1_{10}.$$

Синтез управляючого автомату Мура на тригерах для операції множення першим способом

3.1 Таблиця кодування сигналів

Таблиця 3.1 – Таблиця кодування сигналів

R, W2, W3, W _{CT}	Y1
W1	Y2
ShR1, ShR2, dec	Y3

3.2 Мікроалгоритм в термінах управляючого автомата

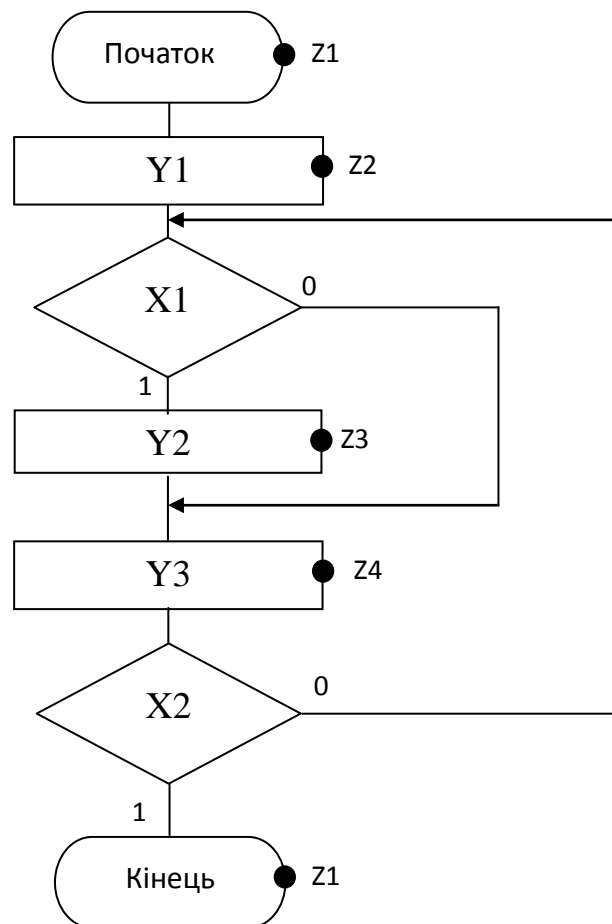


Рисунок 3.1 – Закодований мікроалгоритм

3.3 Граф автомата

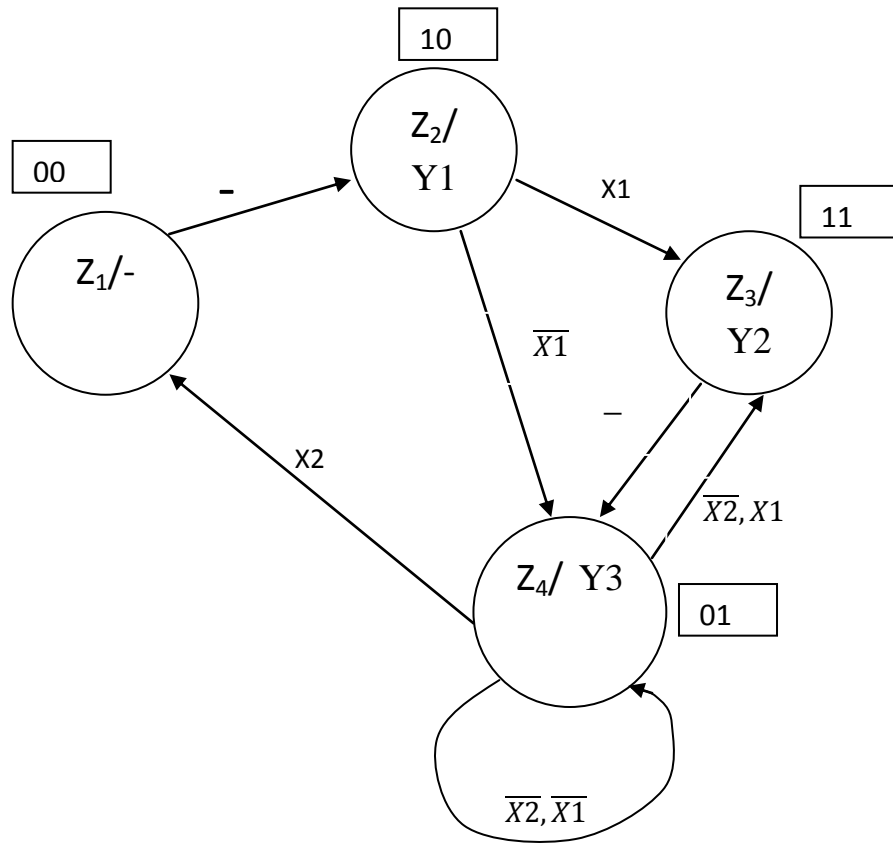


Рисунок 3.2 – Граф циклічного автомата

3.4 Таблица переходів циклічного автомата на D-тригерах

Таблиця 3.2 – Таблица переходів

Пер.	Ст. ст.	Нов. стан	Вх. сигн.	Вих. сигн.	Функції тригерів	
	Q_2Q_1	Q_2Q_1	X_2X_1	$Y_1Y_2Y_3$	D_2	D_1
$Z_1 \rightarrow Z_2$	00	10	- -	0 0 0	1	0
$Z_2 \rightarrow Z_3$	10	11	- 1	1 0 0	1	1
$Z_2 \rightarrow Z_4$	10	01	- 0	1 0 0	0	1
$Z_3 \rightarrow Z_4$	11	01	- -	0 1 0	0	1
$Z_4 \rightarrow Z_3$	01	11	0 1	0 0 1	1	1
$Z_4 \rightarrow Z_4$	01	01	0 0	0 0 1	0	1
$Z_4 \rightarrow Z_1$	01	00	1 -	0 0 1	0	0

3.5 Мінімізація функцій тригерів

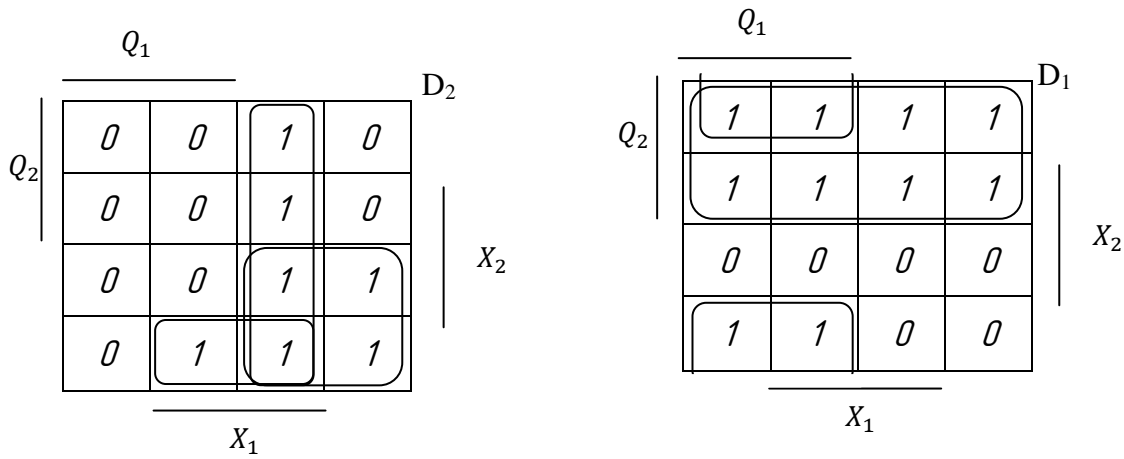


Рисунок 3.3 – Мінімізація функцій тригерів

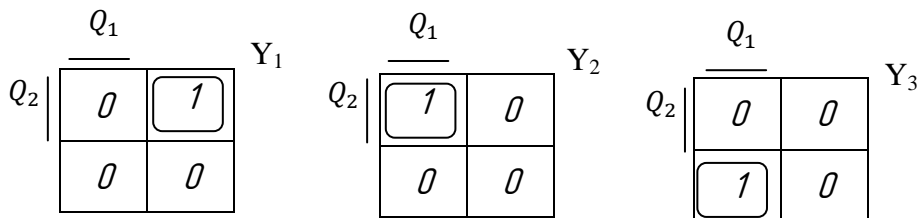


Рисунок 3.4 – Діаграми Вейча для вихідних сигналів

$$D_2 = \overline{Q_2} \overline{Q_1} \vee \overline{Q_1} X_1 \vee \overline{Q_2} \overline{X_2} X_1$$

$$D_1 = Q_2 \vee Q_1 \overline{X_2}$$

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

$$Y_2 = Q_2 Q_1$$

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

3.6 Функціональна схема автомата

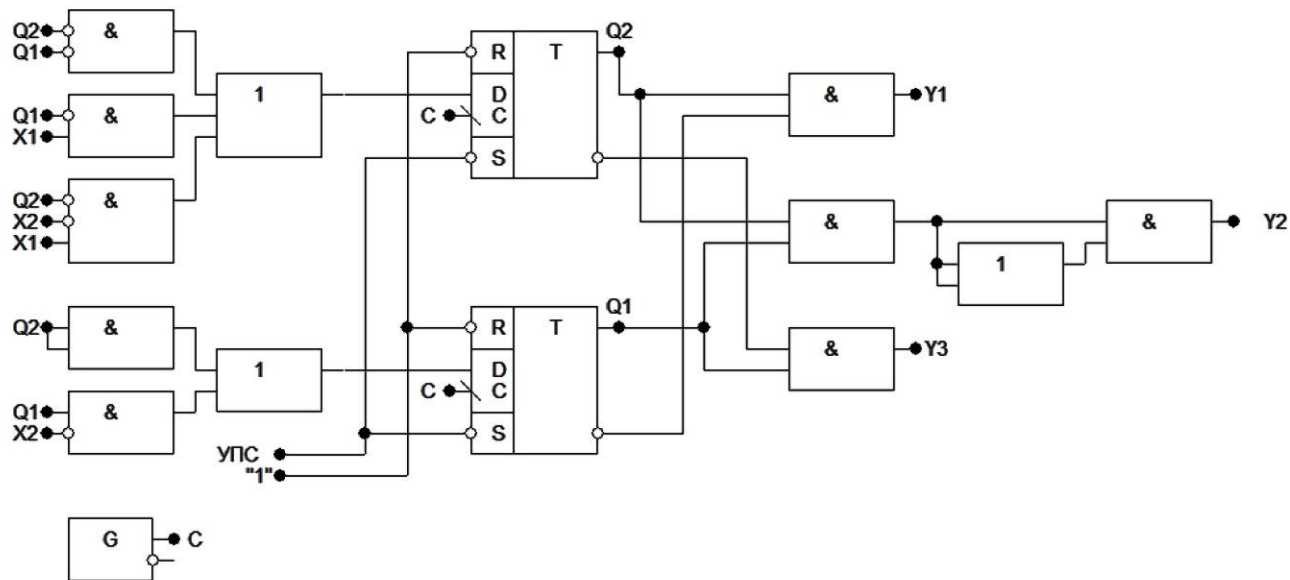


Рисунок 3.5 - Функціональна схема