

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

РОЗРАХУНКОВА ГРАФІЧНА РОБОТА
з дисципліни "Комп'ютерна логіка 2. Комп'ютерна арифметика "

Виконала
Вернер Анна Ігорівна
Факультет ІОТ,
Група ІО-42
Залікова книжка № 4203

Керівник Жадін Валерій Іванович

(підпис керівника)

Завдання

1. Числа i 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 (характеристику) відвести 8 розрядів, на мантису 16 розрядів (з урахуванням знакових розрядів).

2. Виконати 8 операцій з числами i з плаваючою комою (чотири способи множення, два способи ділення, додавання та добування кореня). Номери операцій (для п.3) відповідають порядку переліку (наприклад, 1 – множення першим способом; 6 – ділення другим способом; 8 – добування кореня). 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 управляючими 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.

Операцію додавання до етапу нормалізації результату можна проілюструвати у довільній формі. Вказані пункти виконати для етапу нормалізації результату з урахуванням можливого нульового результату.

3. Для операції з двійковим номером $x_3x_2x_1$ побудувати управляючий автомат Мура на тригерах (тип тригера вибрати самостійно) і елементах булевого базису.

Визначення та обґрунтування варіанту:

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

$$X_2 = -1x_{10}x_91x_8x_7x_61x_5x_40x_31x_2x_1 \quad \text{і} \quad Y_2 = +1x_{10}1x_9x_8x_7x_61x_5x_40x_3x_2x_11;$$

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

$$4203_{10} = 1000001101011_2;$$

$$x_{10}=0, x_9=0, x_8=0, x_7=1, x_6=1, x_5=0, x_4=1, x_3=0, x_2=1, x_1=1.$$

$$X_2 = -10010111,0100111;$$

$$Y_2 = +10100,1110100111.$$

Завдання №1

$$X_{\text{нк}} = 1.10010111,0100111;$$

$$Y_{\text{нк}} = 0.10100,0110001101;$$

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

$$P_x = +1000_2; \quad M_x = -100101110100111_2;$$

$$P_y = +0101_2; \quad M_y = +101000110001101_2;$$

X_2 :

3.P	Ціла частина								3.P	Дробова частина															
0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1	1	0	1	0	0	0	1	1	1

Y_2 :

3.P	Ціла частина								3.P	Дробова частина															
0	0	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	0	0	1	1	0	1

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

$$E = P_x + 2^m, \quad m = 7, \quad 2^7 = 10000000_2$$

$$E_x = 10000000 + 1000 = 10001000$$

$$E = P_y + 2^m, \quad m = 7; \quad 2^7 = 10000000_2$$

$$E_y = 10000000 + 0101_2 = 10000101$$

X_2 :

3.P Ціла частина								3.P Дробова частина															
1	0	0	0	1	0	0	0	1	1	0	0	1	0	1	1	1	0	1	0	0	1	1	1

Y_2 :

3.P Ціла частина								3.P Дробова частина															
1	0	0	0	0	1	0	1	0	1	0	1	0	0	0	1	1	0	0	0	1	1	0	1

Завдання №2

2.1 Перший спосіб множення.

2.1.1 Теоретичне обґрунтування першого способу множення:

Під час множення першим способом в першому такті аналізується значення молодшого розряду регістру **RG2(n)**, в якому знаходиться чергова цифра множника. Вміст регістру **RG3** додається до вмісту регістру **RG1** якщо значення молодшого розряду в другому регістрі рівне 1, або не додається якщо значення рівне 0. В другому такті здійснюється правий зсув в регістрах **RG1** і **RG2** (те саме, що поділити їх на 2). За зсуву цифра молодшого розряду регістру **RG1** записується на місце звільненого старшого розряду **RG2**. Після виконання останнього такту молодші розряди добутку будуть записані у **RG2**, а старші в **RG1**.

2.1.2 Операційна схема:

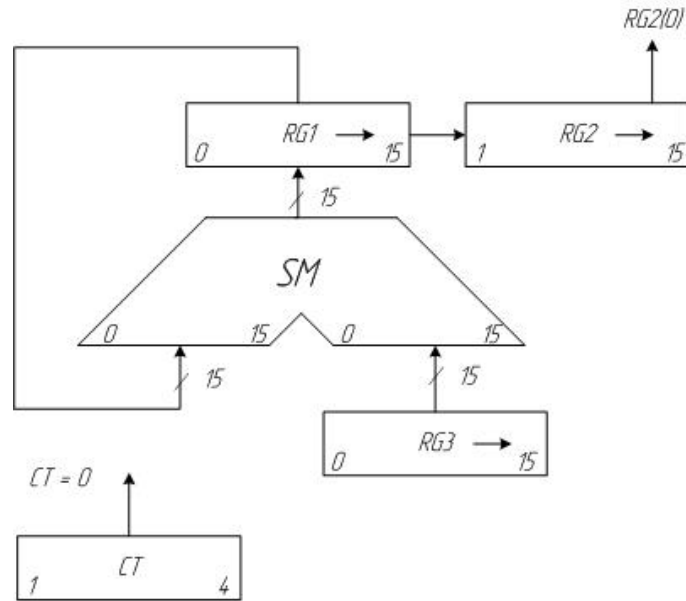


Рисунок 2.1.1 – Операційна схема.

2.1.3 Змістовний мікроалгоритм:

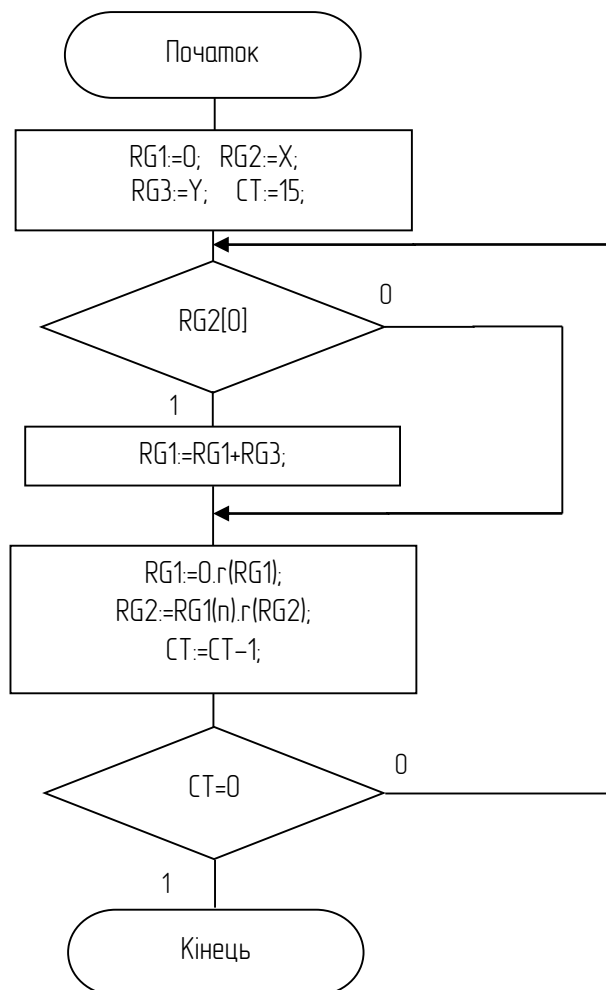


Рисунок 2.1.2 – Змістовний мікроалгоритм виконання операції множення першим способом.

2.1.4 Таблиця станів регістрів:

Таблиця 2.1-Таблиця станів регістрів для першого способу множення.

№	RG1	RG2	RG3	СТ
ПС	0000000000000000	100101110100111	101000110001101	1111
1	0010100011000110	110010111010011	101000110001101	1110
2	+0101000110001101 =0111101001010011 0011110100101001	111001011101001	101000110001101	1101
3	+0101000110001101 =1000111010110110 0100011101011011	011100101110100	101000110001101	1100
4	0010001110101101	101110010111010	101000110001101	1011
5	0001000111010110	110111001011101	101000110001101	1010
6	+0101000110001101 =0110001101100011 0011000110110001	111011100101110	101000110001101	1001
7	0001100011011000	111101110010111	101000110001101	1000
8	+0101000110001101 =0110101001100101 0011010100110010	111110111001011	101000110001101	0111
9	+0101000110001101 =1000011010111111 0100001101011111	111111011100101	101000110001101	0110
10	+0101000110001101 =1001010011101100 0100101001110110	011111101110010	101000110001101	0101
11	0010010100111011	001111110111001	101000110001101	0100
12	+0101000110001101 =0111011011001000 0011101101100100	000111111011100	101000110001101	0011
13	0001110110110010	000011111101110	101000110001101	0010
14	0000111011011001	000001111110111	101000110001101	0001
15	+0101000110001101 =0110000001100110 <u>0011000000110011</u>	<u>000000111111011</u>	101000110001101	0000

2.1.5 Функціональна схема:

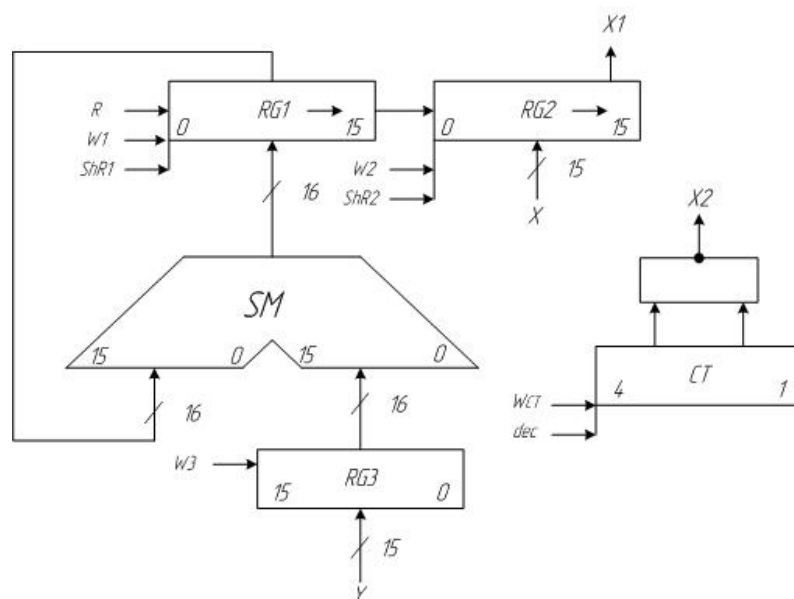


Рисунок 2.13 – Функціональна схема.

2.1.6 Закодований мікроалгоритм

Таблиця 2.12–Таблиця кодування операцій і логічних умов.

Кодування мікрооперацій		Кодування логічних умов	
Мікрооперації	Управляючі сигнали	Логічні умови	Позначення
G1:=0 RG2:=X RG3:=Y CT:=15 RG1:=RG1+RG3 RG1:=0.r(RG1) RG2:=RG1[0].r(RG2) CT:=CT-1	R W2 W3 W _{CT} W1 ShR1 ShR2 dec	RG2[0] CT=0	X1 X2

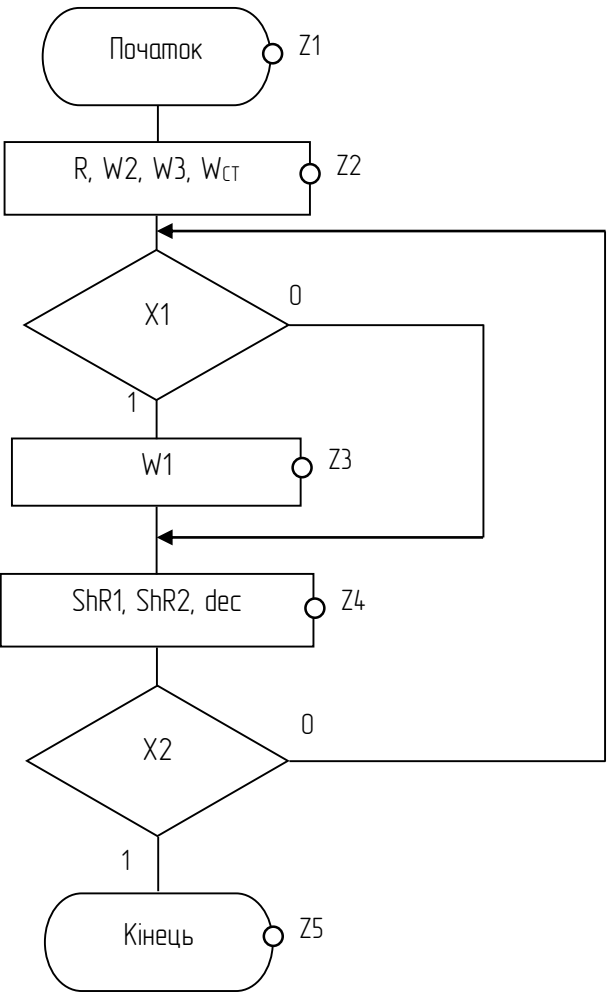


Рисунок 2.14 – Закодований мікроалгоритм.

2.1.7 Граф управляючого автомата Мура з кодами вершин:

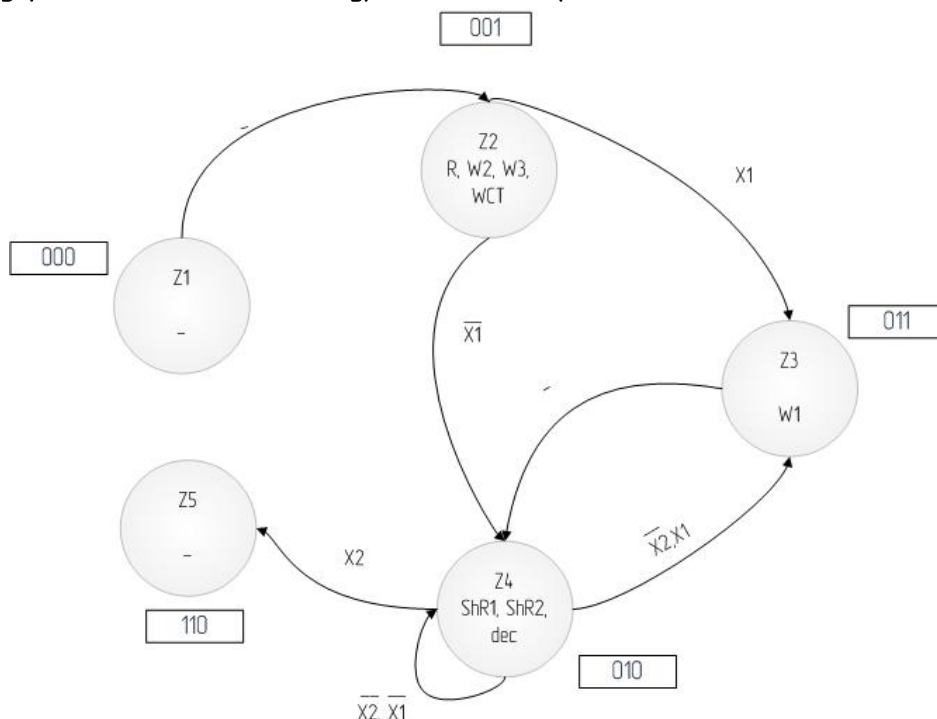


Рисунок 2.15 – Граф автомата Мура

2.1.8 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

$$P_z = P_x + P_y;$$

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

2.1.9 Нормалізація результату:

Отриманий результат: 0011000000110011000000111111011

Знак мантиси: $1 \oplus 0 = 1$.

Робимо зсув результату вліво, доки у першому розряді не буде одиниця, порядок щоразу зменшуємо на 1. В даному числі для нормалізації потрібно виконати 2 зсуви. Тоді отримаємо нормалізовану мантису такого вигляду: 110000001100110000001111101100;

$$P_z = 13 - 2 = 11 = 1011_2;$$

Запишемо нормалізований результат:

ЗР								Порядок								ЗР																Нормалізована мантиса															
0	0	0	0	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0																						

2.2 Другий спосіб множення.

2.2.1 Теоретичне обґрунтування другого способу множення:

Перед початком X записують у $RG2$, а Y записують у молодші розряди регістру $RG3$. В кожному циклі множення додаванням кодів у третьому і першому регістрах керує цифра в $RG2(n)$, а регістр $RG3$ зсувається на один вліво у результаті чого формується значення Y . Оскільки сума часткових добутоків у процесі множення нерухома, зсув у третьому регістрі можна сполучити з підсумовуванням. Завершується операція за визначенням нульового вмісту у регістрі $RG2$.

2.2.2 Операційна схема:

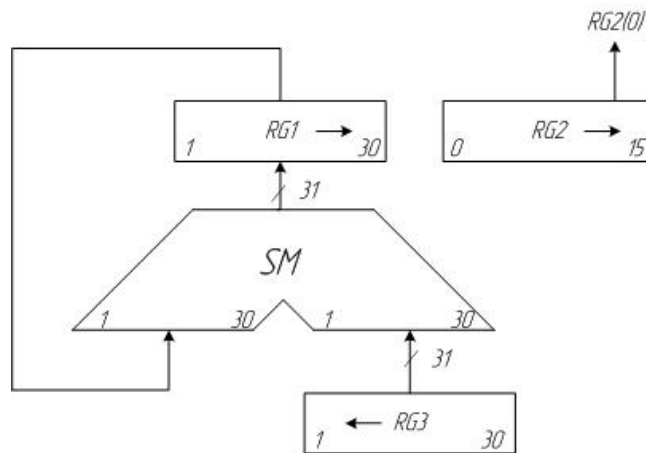


Рисунок 2.2.1- Операційна схема

2.2.3 Змістовний мікроалгоритм:

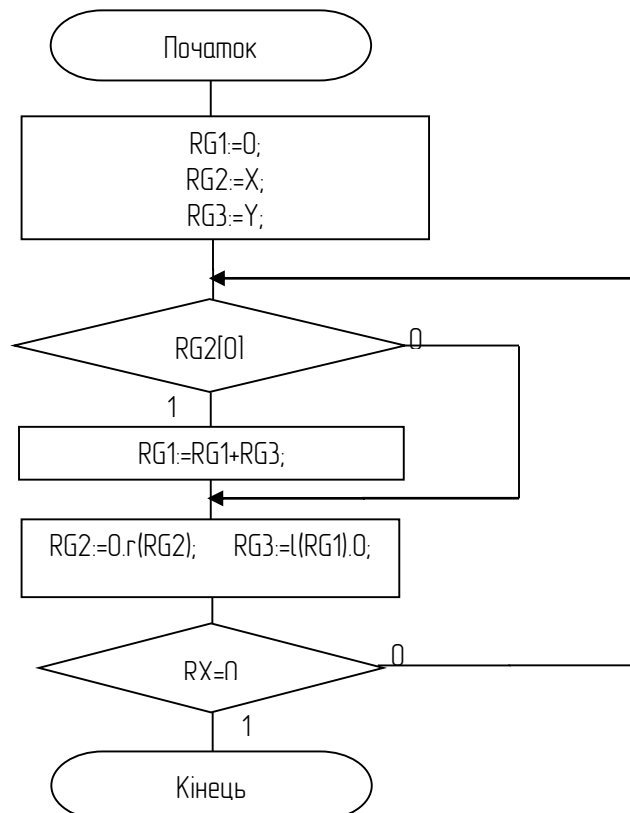


Рисунок 2.2.2 – Змістовний мікроалгоритм.

2.2.4 Таблиця станів резистрів:

Таблиця 2.2.1–Таблиця станів реєстрів.

Nº	RG1	RG3←	RG2→
π	000000000000000000000000000000	000000000000001010001100011010	100100110000110
1	000000000000000101000110001101	000000000000001010001100011010	010010111010011
2	+000000000000001010001100011010 =000000000000001111010010100111	0000000000000010100011000110100	001001011101001
3	+0000000000000010100011000110100 =00000000000000100011101011011011	00000000000000101000110001101000	000100101110100
4	000000000000100011101011011011	0000000000001010001100011010000	000010010111010
5	000000000000100011101011011011	000000000010100011000110100000	000001001011101
6	+0000000000010100011000110100000 =0000000000011000110110001111011	0000000000101000110001101000000	000000100101110
7	0000000000011000110110001111011	0000000001010001100011010000000	000000010010111
8	+0000000001010001100011010000000 =0000000001101010011001011111011	0000000010100011000110100000000	0000000001001011
9	+0000000010100011000110100000000 =00000001000011010111111111011	0000000101000110001101000000000	0000000000100101
10	+0000000101000110001101000000000 =00000001001010011101100111111011	00000001010001100011010000000000	0000000000010010
11	00000100101001110111011111011	0000010100011000110100000000000	0000000000001001
12	+0000010100011000110100000000000 =000001101101100100000011111011	0000010100011000110100000000000	0000000000000100
13	0000011011011001000000111111011	001010001100011010000000000000	0000000000000010
14	0000011011011001000000111111011	010100011000110100000000000000	0000000000000001
15	+0101000110001101000000000000000 =01100000001100110000000011111011	1010001100011010000000000000000	0000000000000000

2.2.5 Функціональна схема:

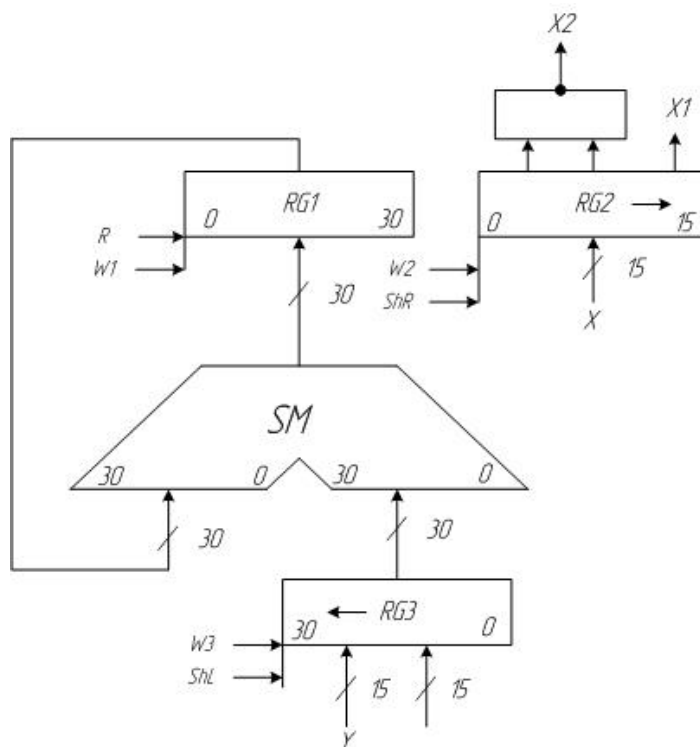


Рисунок 2.2.3– Функціональна схема.

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		

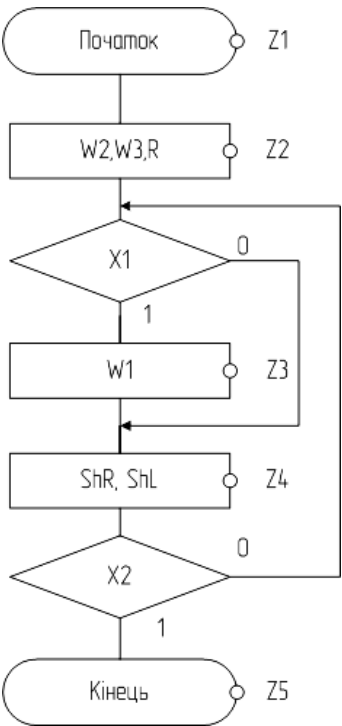


Рисунок 2.2.4 – Закодований мікроалгоритм

2.2.7 Граф управляючого автомата Мура з кодами вершин:

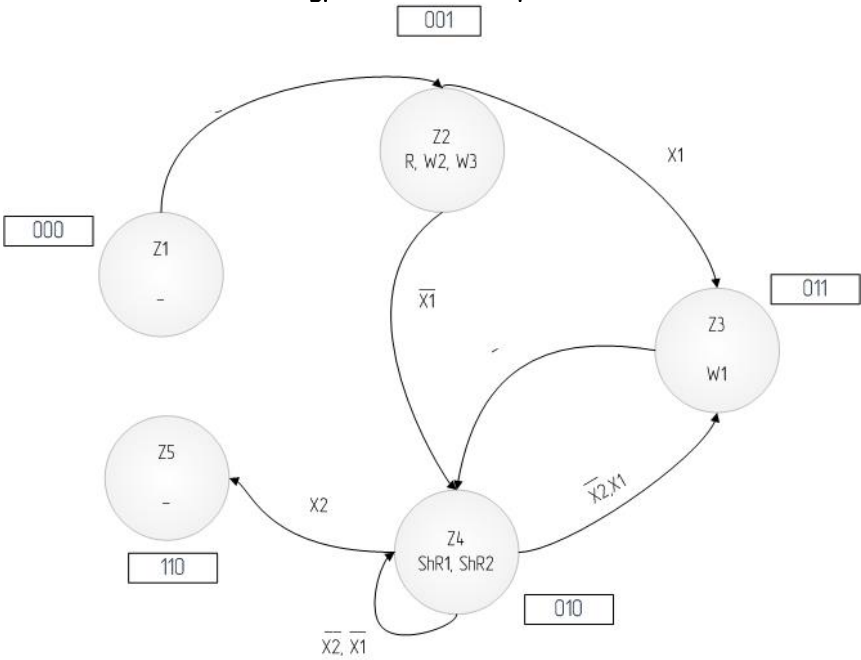


Рисунок 2.2.5 – Граф автомата Мура

2.2.8 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

$$P_z = P_x + P_y;$$

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

2.2.9 Нормалізація результату

Отриманий результат: 011000000110011000000111111011

Знак мантиси: $1 \oplus 0 = 1$.

Робимо зсув результату вліво, доки у першому розряді не буде одиниця, порядок щоразу зменшуємо на 1. В даному числі для нормалізації потрібно виконати 1 зсув. Тоді отримаємо нормалізовану мантису такого вигляду: 110000001100110000001111101100;

$$P_z = 13 - 1 = 12 = 1100_2;$$

Запишемо нормалізований результат:

З.Р				Порядок				З.Р				Нормалізована мантиса											
0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0

2.3 Третій спосіб множення.

2.3.1 Теоретичне обґрунтування третього способу множення:

На початку кожного циклу множення виконується зсув у першому та другому регістрах, а потім виконується додавання, яким керує 1 або нуль записані у старших розрядах другого регістра. У результаті підсумовування першого та третього регістрів може виникнути перенос у молодший розряд другого регістра. Після закінчення множення молодші розряди добутку будуть знаходитися в першому регістрі, а старші – другому.

2.3.2 Операційна схема:

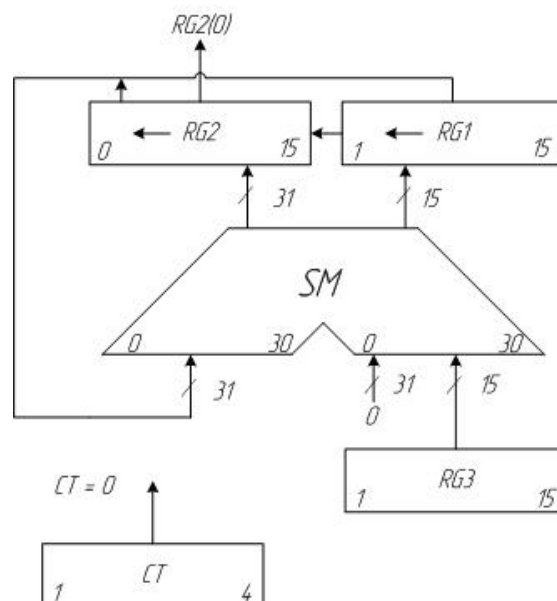


Рисунок 2.3.1 – Операційна схема

2.3.3 Змістовний мікроалгоритм:

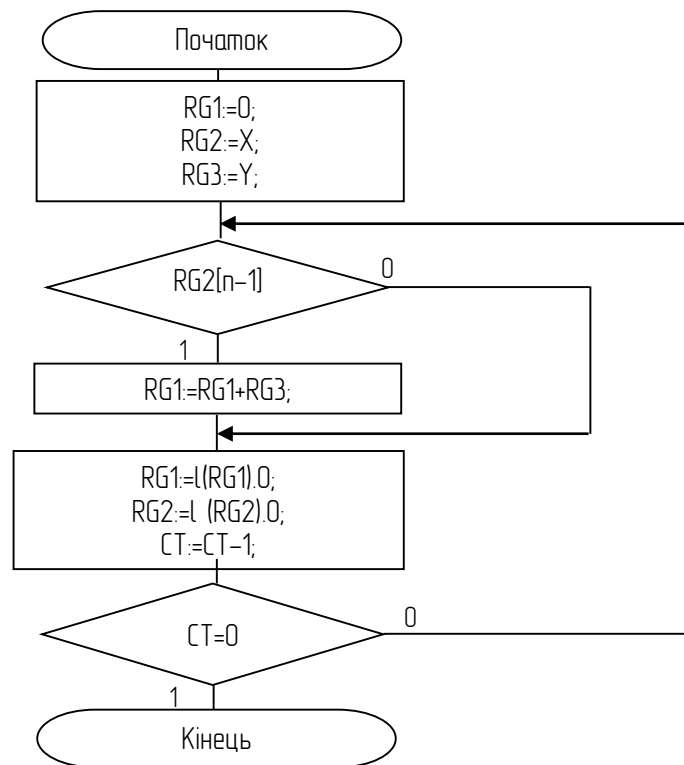


Рисунок 2.3.3- Змістовний мікроалгоритм.

2.3.4 Закодований мікроалгоритм.

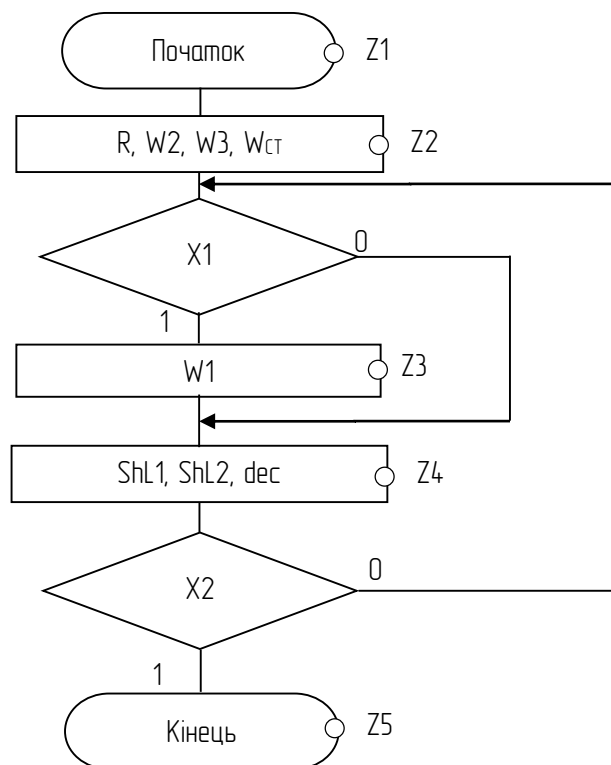


Рисунок 2.3.4 – Закодований мікроалгоритм.

2.3.4 Таблиця станів реєстрів:

Таблиця 2.3.1 – Таблиця станів реєстрів

№	RG1	RG2	RG3	CT
ПС	00000000000000000000000000000000	100101110100111	101000110001101	1111
1	000000000000001010001100011010	001011101001110	101000110001101	1110
2	0000000000000010100011000110100	010111010011100	101000110001101	1101
3	00000000000000101000110001101000	101110100111000	101000110001101	1100
4	+000000000000000101000110001101 =000000000000010110111011110101 000000000000101101110111101010	011101001110000	101000110001101	1011
5	00000000001011011101111010100	111010011100000	101000110001101	1010
6	+000000000000000101000110001101 =0000000000010111100100101100001 000000000101111001001011000010	110100111000000	101000110001101	1001
7	+000000000000000101000110001101 =000000000010111110010001001111 00000000101111100100010011110	101001110000000	101000110001101	1000
8	+000000000000000101000110001101 =000000001100000001101000101011 000000011000000011010001010110	010011100000000	101000110001101	0111
9	000000110000000110100010101100	100111000000000	101000110001101	0110
10	+000000000000000101000110001101 =0000001100000001011101000111001 000001100000010111010001110010	001110000000000	101000110001101	0101
11	000011000000101110100011100100	011100000000000	101000110001101	0100
12	000110000001011101000111001000	111000000000000	101000110001101	0011
13	+000000000000000101000110001101 =000110000001100010001101010101 001100000011000100011010101010	110000000000000	101000110001101	0010
14	+000000000000000101000110001101 =001100000011001001100000110111 011000000110010011000001101110	100000000000000	101000110001101	0001
15	+000000000000000101000110001101 =01100000011001100000011111011	000000000000000	101000110001101	0000

2.3.6 Закодований мікроалгоритм:

Таблиця 2.3.2–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	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		

2.3.5 Функціональна схема:

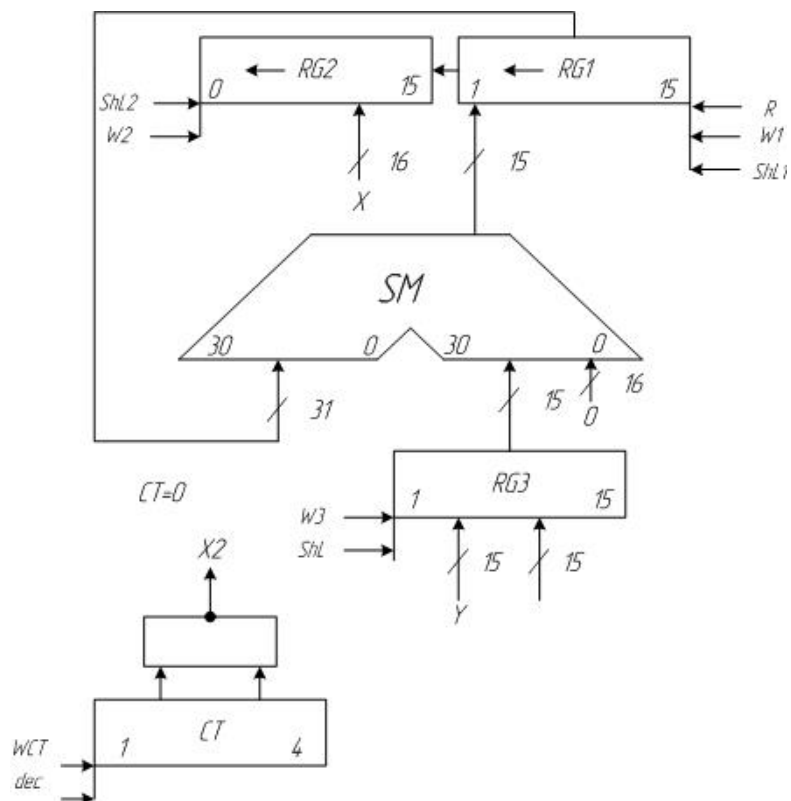


Рисунок 2.3.3 – Функціональна схема.

2.3.7 Граф управляющего автомата Мура з кодами вершин:

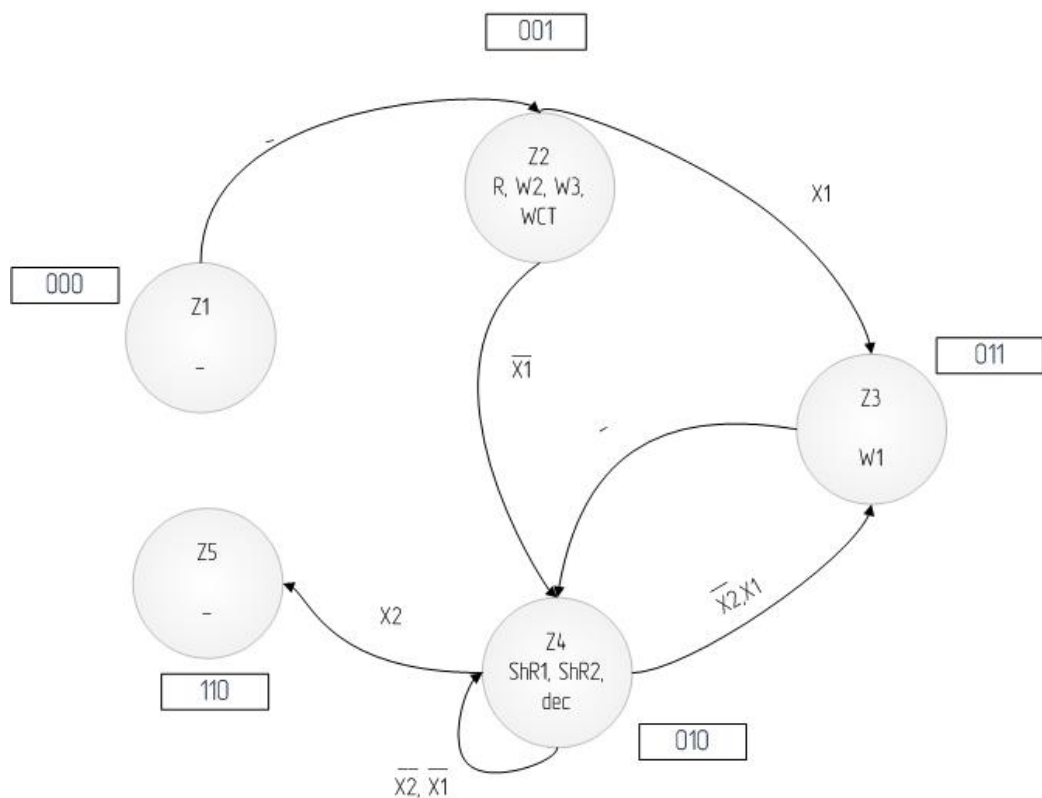


Рисунок 2.3.5 – Граф автомата Мура

2.3.8 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

$$P_z = P_x + P_y;$$

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

2.3.9 Нормалізація результату

Отриманий результат: 011000000110011000000111111011

Знак мантиси: $1 \oplus 0 = 1$.

Робимо зсув результату вліво, доки у першому розряді не буде одиниця, порядок щоразу зменшуємо на 1. В даному числі для нормалізації потрібно виконати 1 зсув. Тоді отримаємо нормалізовану мантису такого вигляду: 110000001100110000001111101100;

$$P_z = 13 - 1 = 12 = 1100_2;$$

Запишемо нормалізований результат:

З.Р								Порядок								З.Р								Нормалізована мантиса								
0	0	0	0	0	1	1	0	0									1	1	1	0	0	0	0	0	0	1	1	0	0	1	1	0

2.4 Четвертий спосіб множення.

2.4.1 Теоретичне обґрунтування четвертого способу множення:

Перед початком множення множник записують у другий регістр, а множене – в старші розряди третього. Під час кожного такту цифра, що знаходиться у старшому розряді другого регістра керує, за встановленої там одиниці, підсумовуванням, а в третьому регістрі відбувається правий зсув числа на один розряд. Завершується операція за визначенням нульового вмісту у регістрі RG2.

2.4.2 Операційна схема:

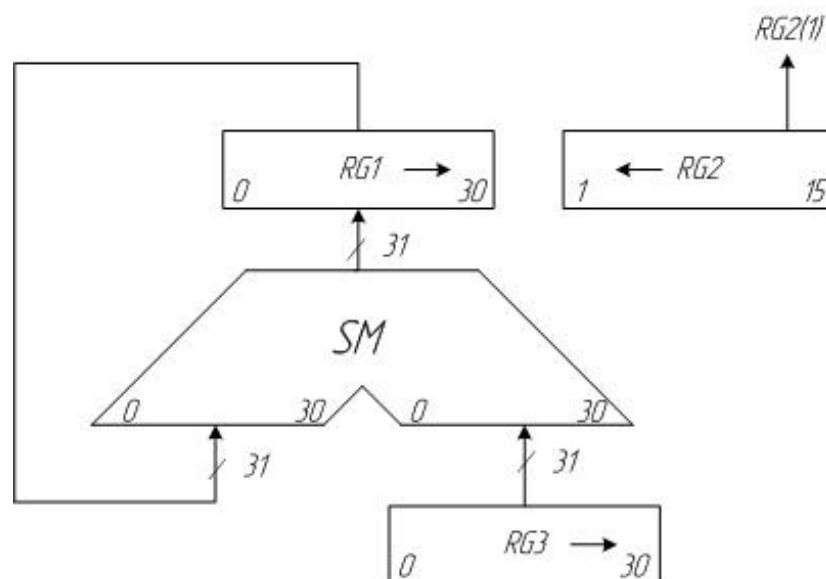


Рисунок 2.4.2 – Операційна схема

2.4.3 Змістовний мікроалгоритм:

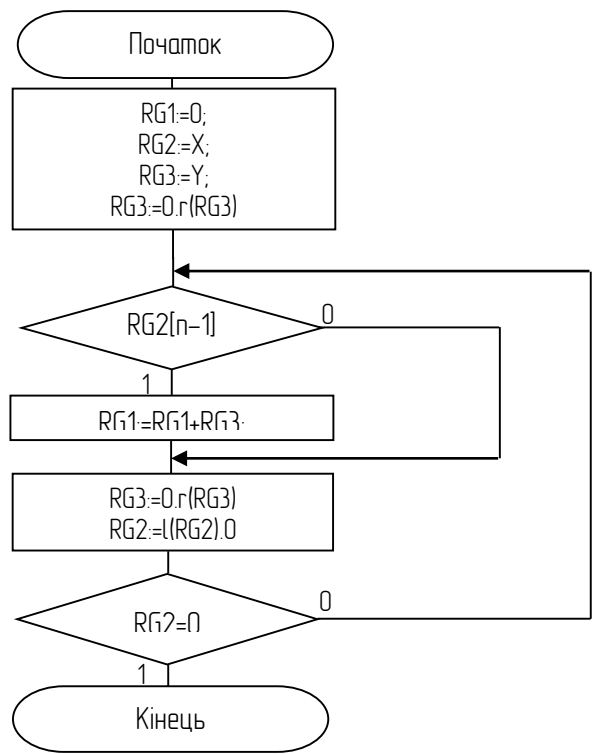


Рисунок 2.4.3 – Змістовний мікроалгоритм.

2.4.4 Таблиця станів регістрів:

Таблиця 2.4.1 – Таблиця станів регістрів

№	RG1	RG2	RG3
ПС	000000000000000000000000	0100101110100111000000000000	101000110001101
1	0101000110001101000000000000	0010100011000110100000000000	001011101001110
2	0101000110001101000000000000	0001010001100011010000000000	010111010011100
3	0101000110001101000000000000	0000101000110001101000000000	101110100111000
4	+0000101000110001101000000000 =0101101110111110101000000000	0000010100011000110100000000	011101001110000
5	0101101110111110101000000000	0000001010001100011010000000	111010011100000
6	+0000001010001100011010000000 =0101111001001011000010000000	0000000101000110001101000000	110100111000000
7	+0000000101000110001101000000 =0101111100100010011110000000	0000000010100011000110100000	101001110000000
8	+0000000010100011000110100000 =0110000000110100010101100000	0000000001010001100011010000	01001100000000
9	0110000000110100010101100000	0000000000101000110001101000	10011000000000
10	+0000000000101000110001101000 =0110000000101101000111001000	0000000000010100011000110100	00110000000000
11	0110000000101101000111001000	0000000000001010001100011010	01100000000000
12	0110000000101101000111001000	0000000000000101000110001101	11000000000000
13	+0000000000000101000110001101 =0110000000110001000110101010	00000000000000101000110001101	11000000000000
14	+000000000000000101000110001101 =011000000011001001100000110110	000000000000000101000110001101	10000000000000
15	+0000000000000000010100011001 =011000000011001100000011111011	000000000000000010100011000110	00000000000000

2.4.5 Функціональна схема:

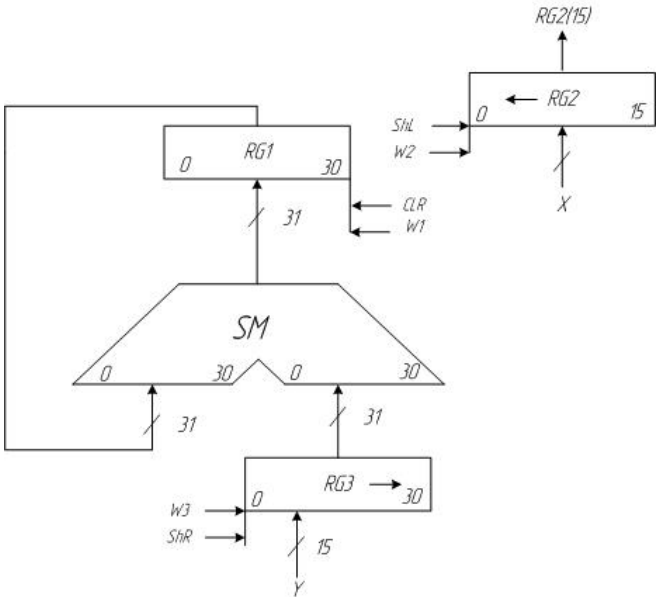


Рисунок 2.4.3 – Функціональна схема.

2.4.6 Закодований мікроалгоритм

Таблиця 2.4.2–Таблиця кодування операцій і логічних умов.

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

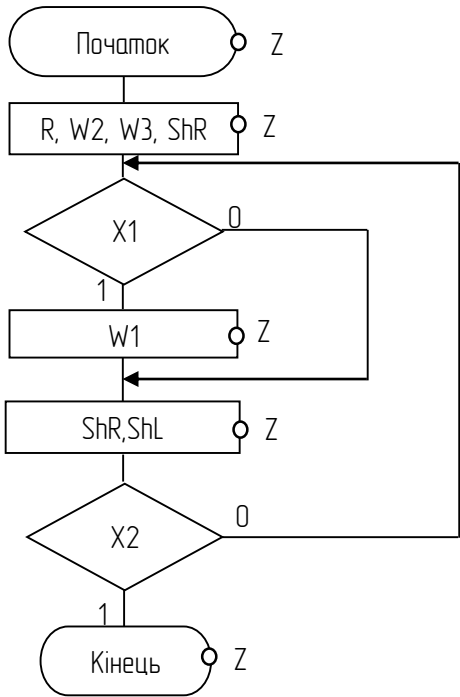


Рисунок 2.4.4–Закодований мікроалгоритм.

2.4.7 Граф управляючого автомата Мура з кодами вершин:

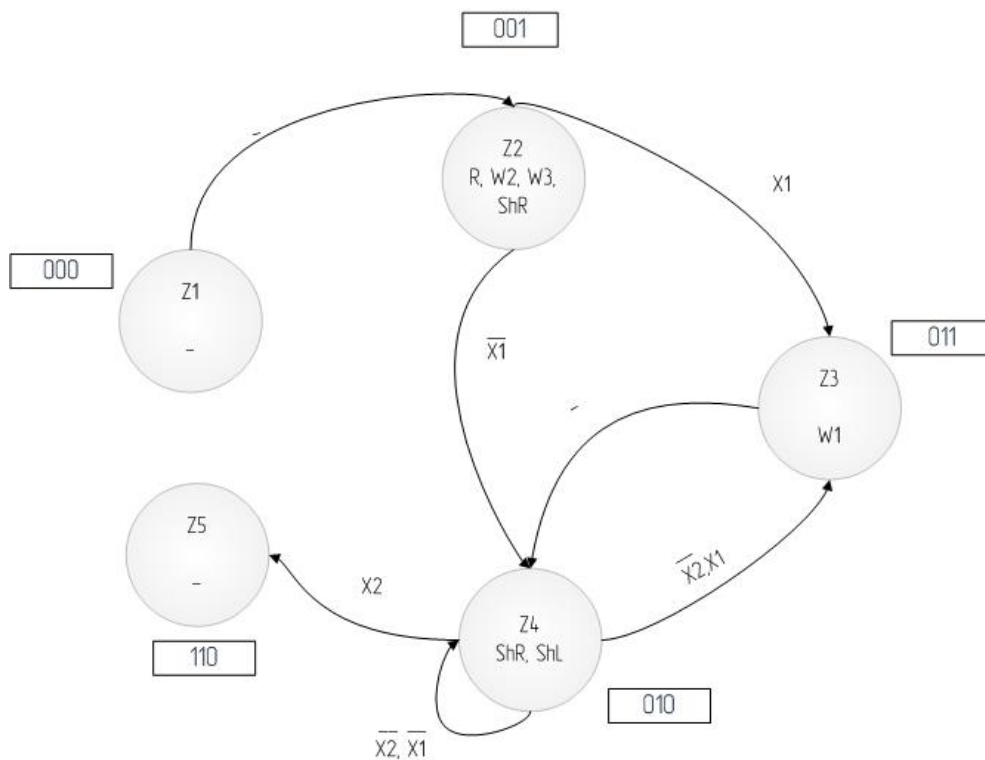


Рисунок 2.4.5 – Граф автомата Мура

2.4.8 Обробка порядків:

Порядок добутку буде дорівнювати сумі порядків множників з урахуванням знаку порядків:

$$P_z = P_x + P_y;$$

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

2.4.9 Нормалізація результату

Отриманий результат: 011000000110011000000111111011

Знак мантиси: $1 \oplus 0 = 1$.

Робимо зсув результату вліво, доки у першому розряді не буде одиниця, порядок щоразу зменшуємо на 1. В даному числі для нормалізації потрібно виконати 1 зсув. Тоді отримаємо нормалізовану мантису такого вигляду: 110000001100110000001111101100;

$$P_z = 13 - 1 = 12 = 1100_2;$$

Запишемо нормалізований результат:

З.Р								Порядок								З.Р								Нормалізована мантиса							
0	0	0	0	0	1	1	0	0	1	1	1	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0					

2.5. Перший спосіб ділення.

2.5.1 Теоритичне обґрунтування першого способу ділення:

Нехай ділене X і дільник $Y \in n$ -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).

2.5.2 Операційна схема:

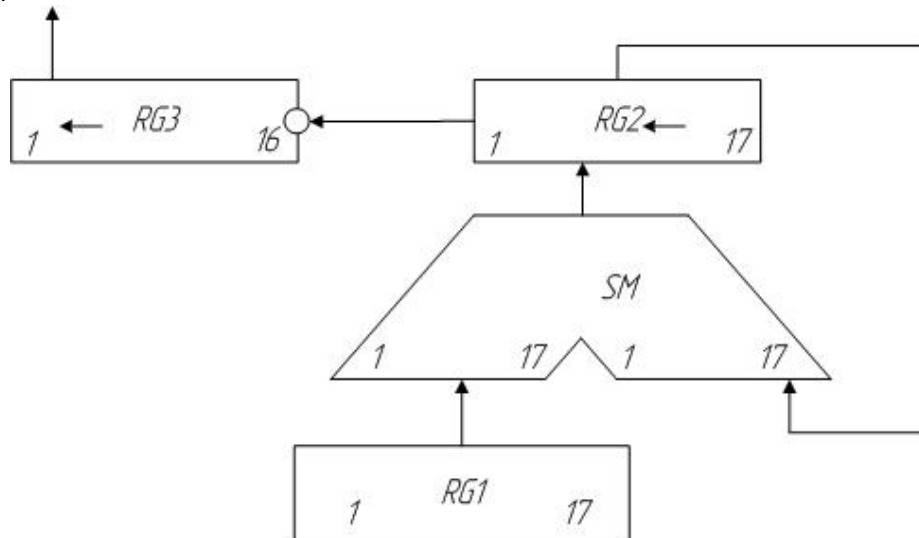


Рисунок 2.5.1 – Операційна схема

2.5.3 Змістовний мікроалгоритм:

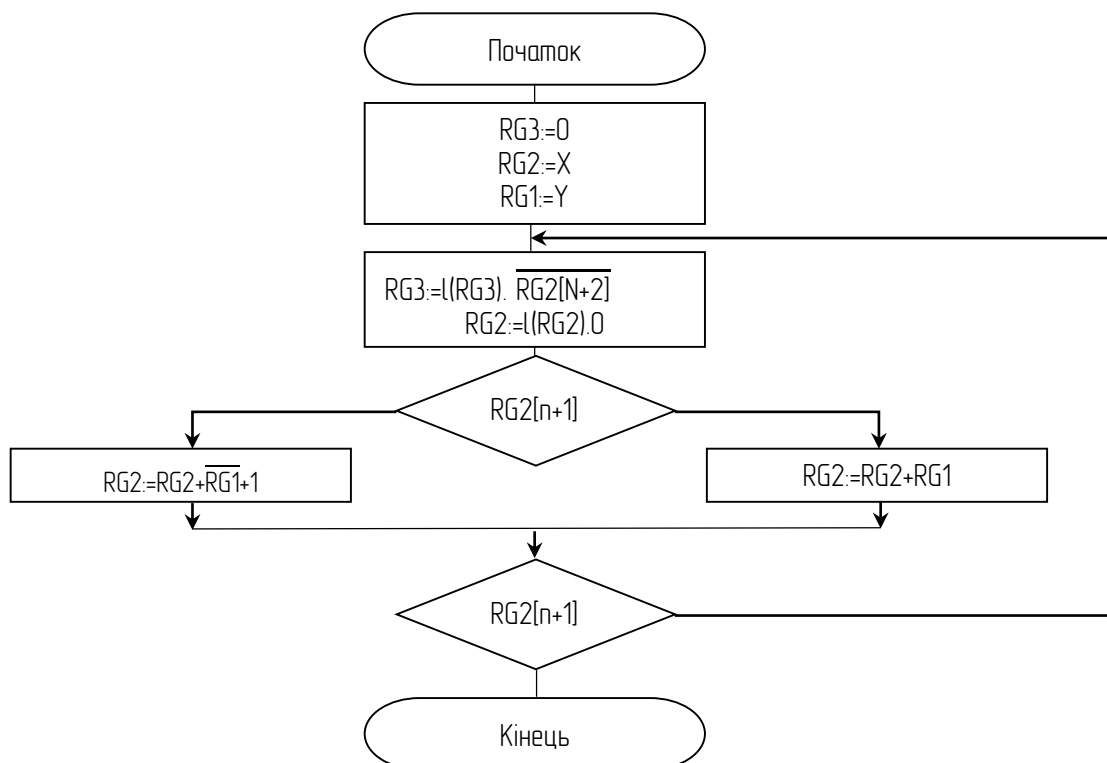


Рисунок 2.5.2–Змістовний мікроалгоритм

2.5.4 Таблиця станів резистрів:

№	RG3	RG2	RG1
ПС	0000000000000000	00100101111001100	101000110001101
1	0000000000000001	01001011101001110+ 11010111001110011= 00100010111000001	101000110001101
2	0000000000000011	01000101110000010+ 11010111001110011= 00011100111110101	101000110001101
3	0000000000000111	00111001111101010+ 11010111001110011= 00010001001011101	101000110001101
4	0000000000001111	00100010010111010+ 11010111001110011= 11111001100101101	101000110001101
5	0000000000011110	11110011001011010+ 00101000110001101= 00011011111100111	101000110001101
6	0000000000111101	00110111111001110+ 11010111001110011= 00001111001000001	101000110001101
7	0000000001111011	00011110010000010+ 11010111001110011= 11110101011110101	101000110001101
8	0000000011110110	11101010111101010+ 00101000110001101= 00010011101110111	101000110001101
9	0000000111101101	00100111011101110+ 11010111001110011= 1111110101100001	101000110001101
10	0000001111011010	11111101011000010+ 00101000110001101= 00100110001001111	101000110001101
11	0000011110110101	01001100010011110+ 11010111001110011= 00100011100010001	101000110001101
12	0000111101101011	01000111000100010+ 11010111001110011= 00011110010010101	101000110001101
13	0001111011010111	00111100100101010+ 11010111001110011= 00010011110011101	101000110001101
14	0011110110101111	00100111100111010+ 11010111001110011= 1111110110101101	101000110001101
15	0111101101011110	11111101101011010+ 00101000110001101= 00100110011100111	101000110001101
16	1111011010111101	01001100111001110+ 11010111001110011 = 00100100001000001	101000110001101

2.5.5 Функціональна схема:

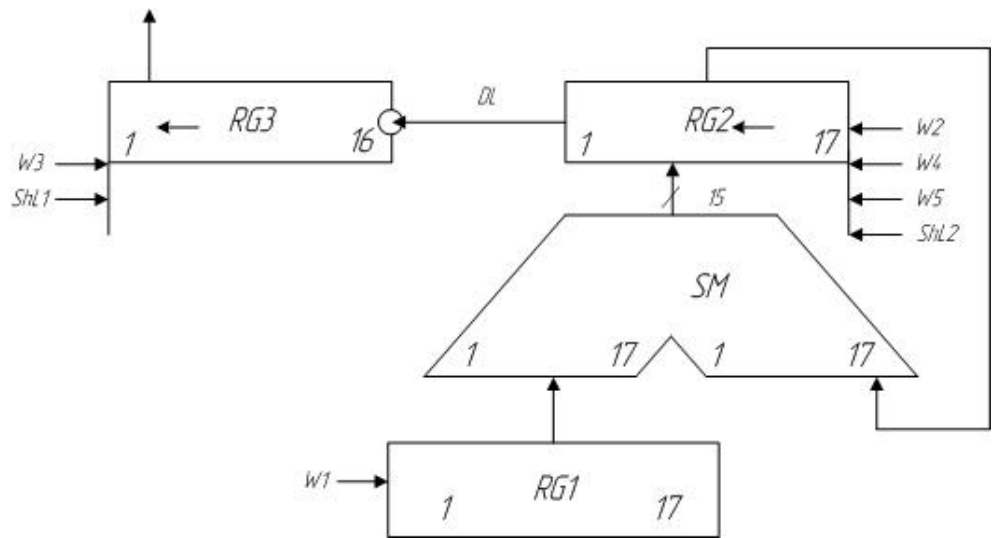


Рисунок 2.5.3 – Функціональна схема

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:=(RG3).RG2[n+1]	ShL1		
RG2:=l(RG2).0	ShL2		
RG2:=RG2+RG1+1	W4		
RG2:=RG2+RG1	W5		

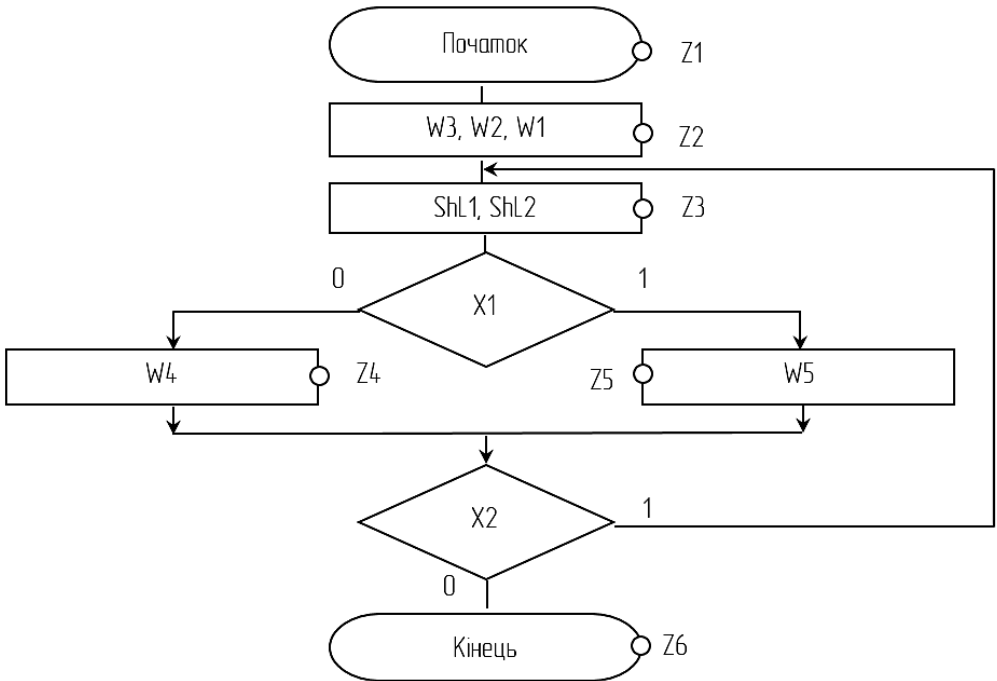


Рисунок 2.5.4 – Закодований мікроалгоритм

2.5.7 Граф управляючого автомата Мура з кодами вершин:

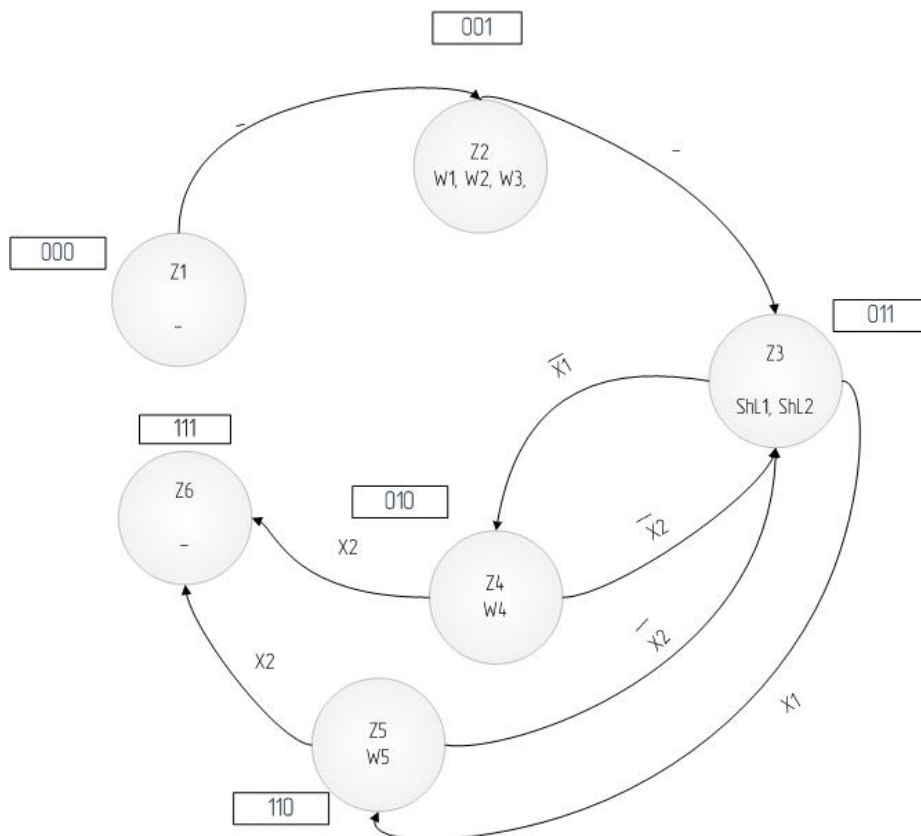


Рисунок 2.5.5 – Граф управляючого автомата.

2.5.8 Обробка порядків:

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

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

2.5.8 Нормалізація результату:

Отримали результат: 1111011010111101

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

З.Р	Порядок								З.Р	Мантиса															
0	0	0	0	0	0	0	1	1	1	1	1	1	0	1	1	0	1	0	1	1	1	0	1		

2.6. Другий спосіб ділення.

2.6.1 Теоритичне обґрунтування другого способу ділення:

Нехай ділене X і дільник $Y \in p$ -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. Швидший за перший спосіб ділення.

2.6.2 Операційна схема

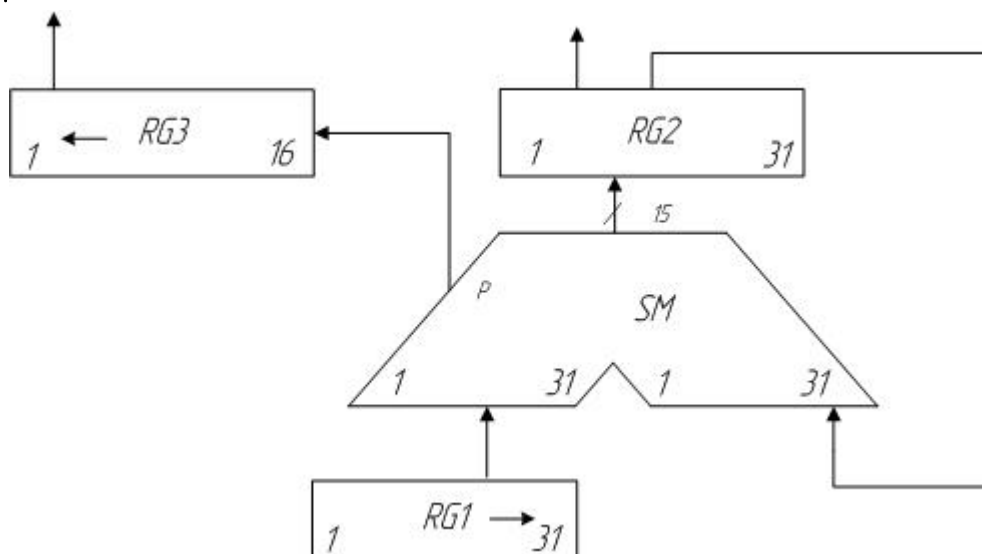


Рисунок 2.6.1-Операційна схема

2.6.3 Змістовний мікроалгоритм

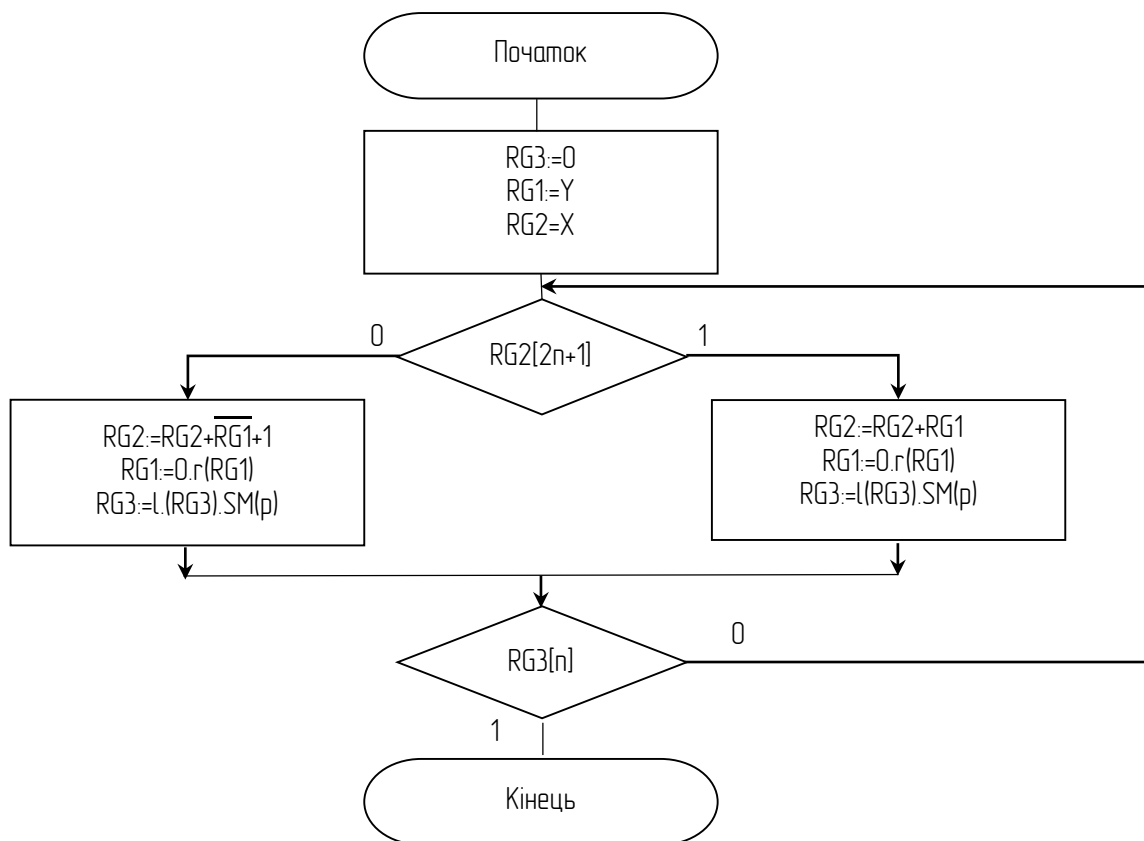


Рисунок 2.6.2-Змістовний мікроалгоритм

2.6.4 Таблиця станів реєстрів

Таблиця 2.6.1 – Таблиця станів реєстрів

№	RGZ	RGX	RGY
PC	0000000000000001	01001010100110000000000000	0010100011000110100000000000
1	0000000000000011	01001010100110000000000000+ 11010110011001100000000000= 0010001011000001000000000000	000101000110001101000000000000
2	0000000000000111	0010001011000001000000000000+ 11010110011001100000000000= 0000111001110101000000000000	000010100011000110100000000000
3	0000000000001111	0000111001110101000000000000+ 11101011001100110000000000= 000001000100101110100000000000	000001010001100011010000000000
4	0000000000011110	0000010001010110100000000000+ 11110101100110011000000000= 1111110011001010100000000000	000000101000110001101000000000
5	0000000001111011	1111110011001010100000000000+ 0000001010001100011010000000= 0000000101111100110000000000	000000010100011000110100000000
6	0000000001111011	0000000101111100110000000000+ 11111101011001100110000000= 0000000001110010000010000000	000000001010001100011010000000
7	00000000011110110	0000000001110010000010000000+ 11111110101100110010000000= 11111111010101101000000000	000000000101000110001101000000
8	00000000111101101	1111111101010110100000000000+ 000000000101000110001101000000= 0000000000100110110110000000	000000000010100011000110100000
9	00000001111011010	0000000000100111011010000000+ 11111111010110011100100000= 11111111110101100000100000	000000000001010001100011010000
10	00000011110110101	1111111111010110000010000000+ 000000000001010001100011010000= 000000000001001100010011100000	0000000000000101000110001101000
11	0000111101101011	000000000001001100001001110000+ 11111111110101100110001000= 0000000000010001100010001000	0000000000000010100011000110100
12	0001111011010111	000000000000100011000010001000+ 11111111110101100110001000= 0000000000000111001001010100	00000000000000001010001100011010
13	0011110110101111	000000000000001110010001010100+ 11111111110101100110001000= 00000000000000001001110011010	000000000000000000101000110001101
14	0111101101011110	000000000000000010011100110100+ 11111111110101100110001100= 11111111110101101010101	0000000000000000000010100011000110
15	1111011010111101	11111111110101101010101010+ 000000000000000001000110001100= 000000000000000001001100110011	00000000000000000000001010001100011

2.6.5 Функціональна схема з відображенням управляючих сигналів

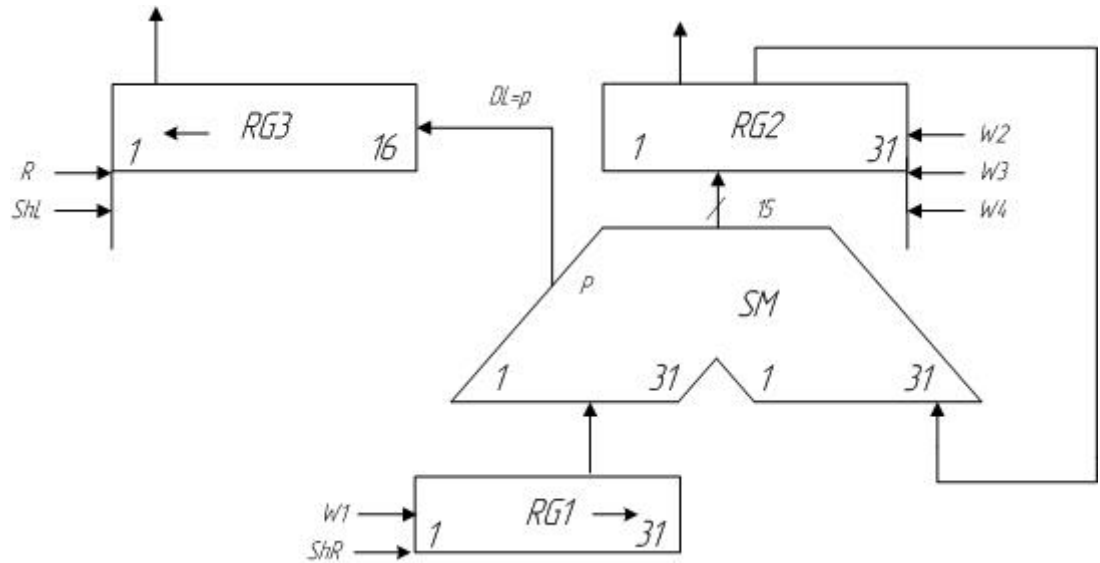


Рисунок 2.6.3–Функціональна схема

2.6.6 Закодований мікроалгоритм

Таблиця 2.6.2– Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій			Таблиця кодування логічних умов
MO	YC	AY	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		

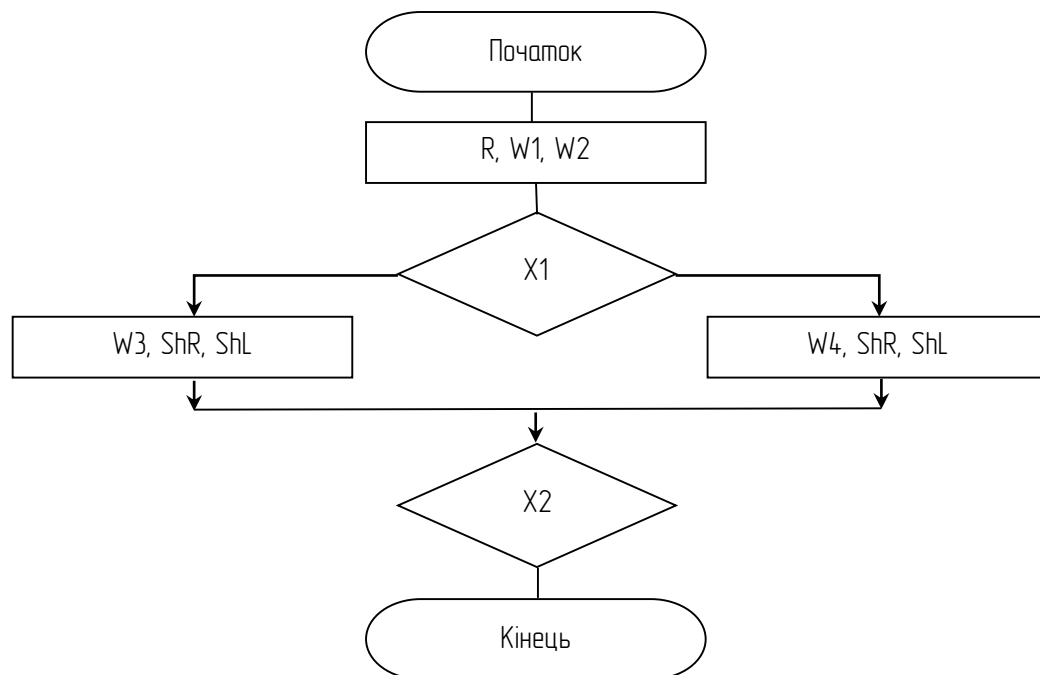


Рисунок 2.6.4– Закодований мікроалгоритм

2.6.7 Граф управляючого автомата Мура з кодами вершин

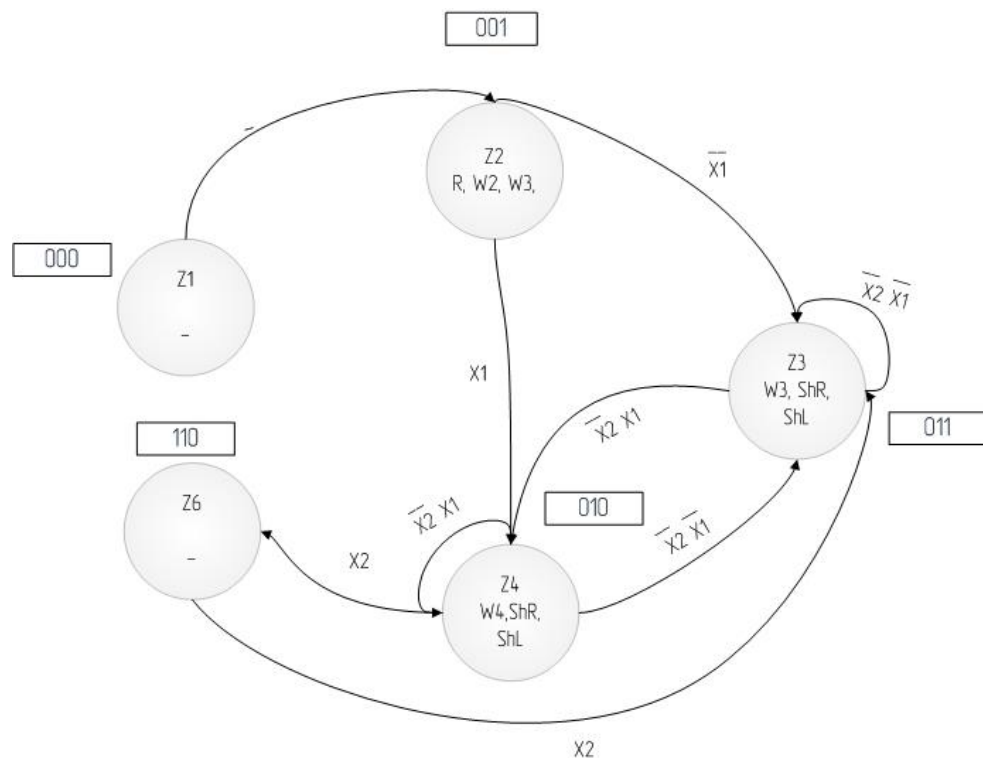


Рисунок 2.6.5– Граф автомата Мура

2.5.8 Обробка порядків:

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

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

2.5.8 Нормалізація результату:

Отримали результат: 1111011010111101

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

З.Р								Порядок								З.Р								Мантиса							
0	0	0	0	0	0	1	1									1	1	1	1	1	0	1	1	0	1	0	1	1	1	0	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.

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.

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

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

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

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, 101000110001101	11	Початковий стан
0, 010100011000110	10	$M_y = 0 \oplus M_y$, $\Delta = \Delta - 1$
0, 001010001100011	01	$M_y = 0 \oplus M_y$, $\Delta = \Delta - 1$
0,000101000110001	00	$M_y = 0 \oplus M_y$, $\Delta = \Delta - 1$

3.1 Додавання мантис у модифікованому ДК.

$$X_{\text{мдк}} = 11. 011010001011001$$

$$Y_{\text{мдк}} = 00. 000101000110001$$

Таблиця 2.7.2–Додавання мантис(для додавання)

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

$$Z_{\text{дк}} = 1. 01111010001010$$

$$Z_{\text{пк}} = 1. 100000101110110$$

3.2 Віднімання мантис у модифікованому ДК.

$$X_{\text{мдк}} = 11. 011010001011001$$

$$-Y_{\text{мдк}} = 11. 010111001110011$$

Таблиця 2.7.3–Віднімання мантис у модифікованому ДК

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

4. Нормалізація результату (В ПК).

$$P_z = 7_{10} = 111_2$$

$$M_z = - 100000101110110_2$$

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

$$P_z = 7_{10} = 111_2$$

$$M_z = - 001110100110100_2$$

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

2.7.2 Операційна схема

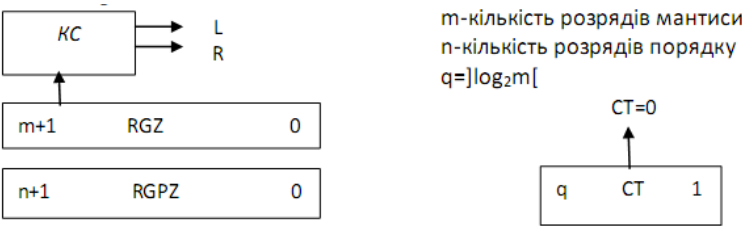


Рисунок 2.7.1-Операційна схема

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

Результат беремо по модулю, знак встановлюємо за Z₀ до нормалізації.

2.7.3 Змістовний алгоритм

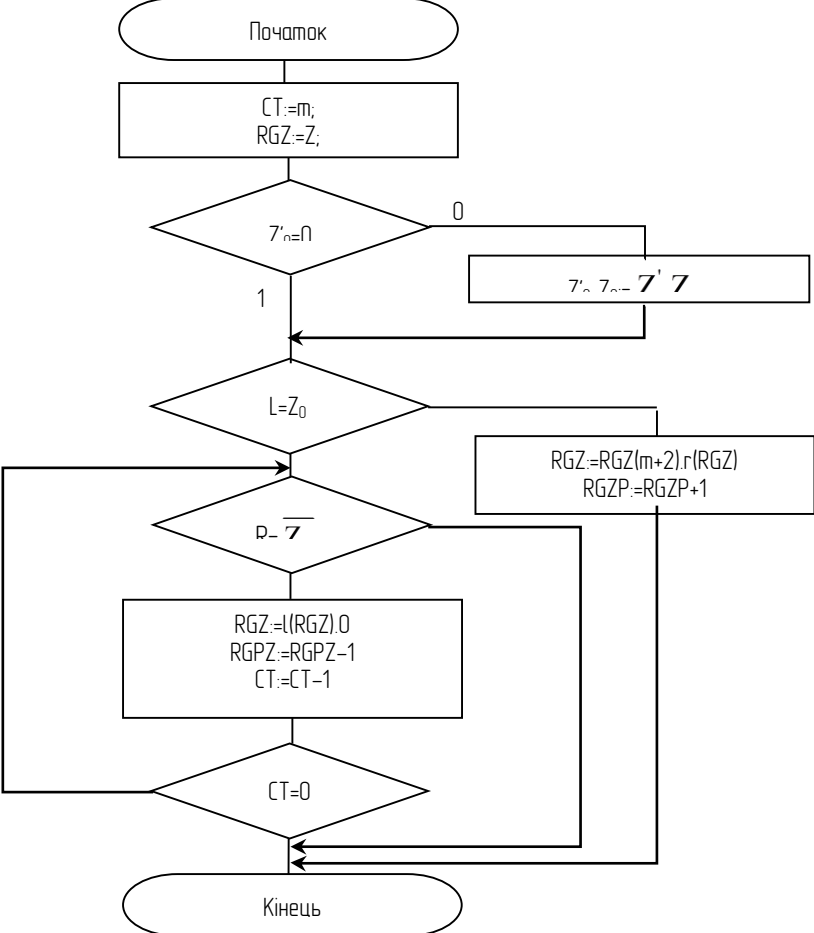


Рисунок 2.7.2-Змістовний мікроалгоритм

2.7.4 Таблиця станів регістрів

1) Додавання

Таблиця 2.7.5- Таблиця станів регістрів

№такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ
ПС	0001000	01111010001010	0	0	100

2) Віднімання

Таблиця 2.7.6- Таблиця станів регістрів

№такту	RGPZ	RGZ	ЛПН(L)	ППН(R)	СТ
ПС	0001000	110001011001100	1	0	100

2.7.5 Функціональна схема з відображенням керуючих сигналів

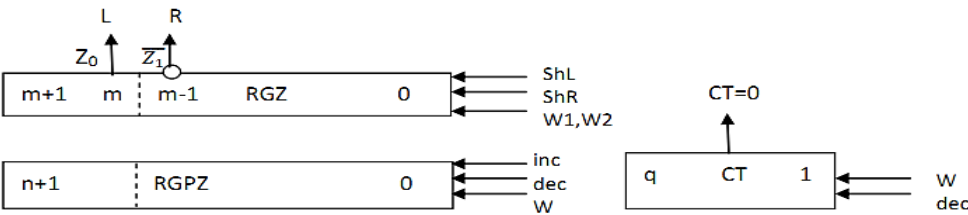


Рисунок 2.7.3 – Функціональна схема

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	L=Z ₀	X1
RGZ:=Z	W1	Z' ₀ =0	X2
RGZ:=RGZ(m+2).r(RGZ)	ShR	R=Z1	X3
RGPZ:=RGPZ+1	Inc	CT=0	X4
RGZ:=l(RGZ).0	ShL		
RGPZ:=RGPZ-1	Dec		
CT:=CT-1	Dec		
RGZ:=minM	W2		
RGPZ:=minP	W3		

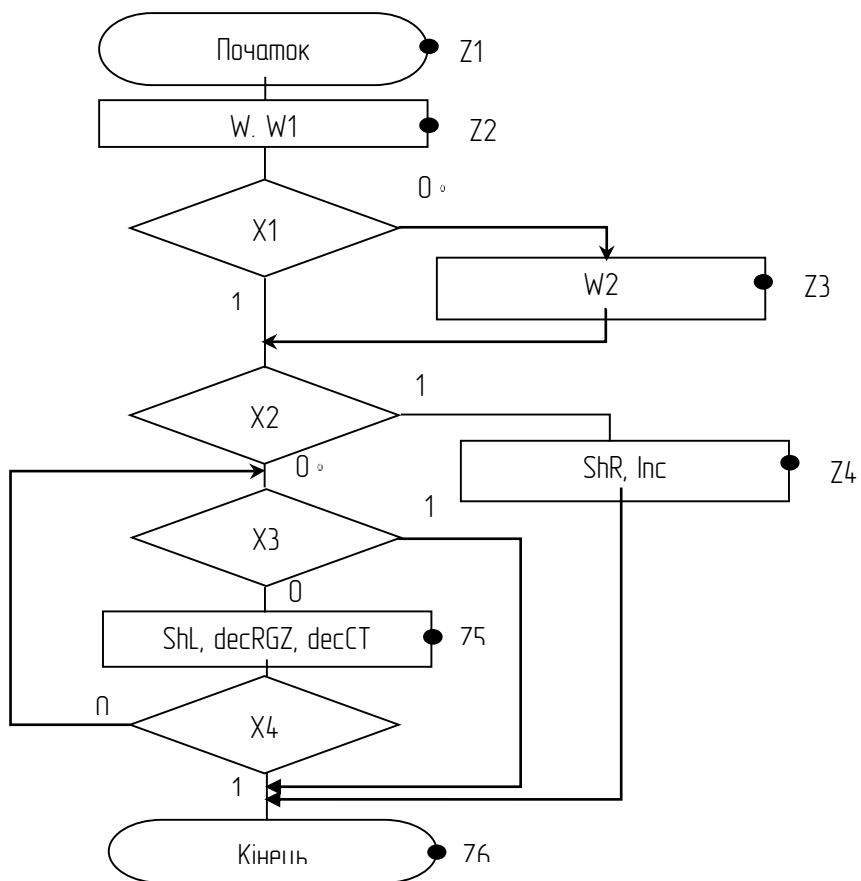


Рисунок 2.7.4 – Закодований мікроалгоритм

2.7.7 Граф управляючого автомата Мура з кодами вершин

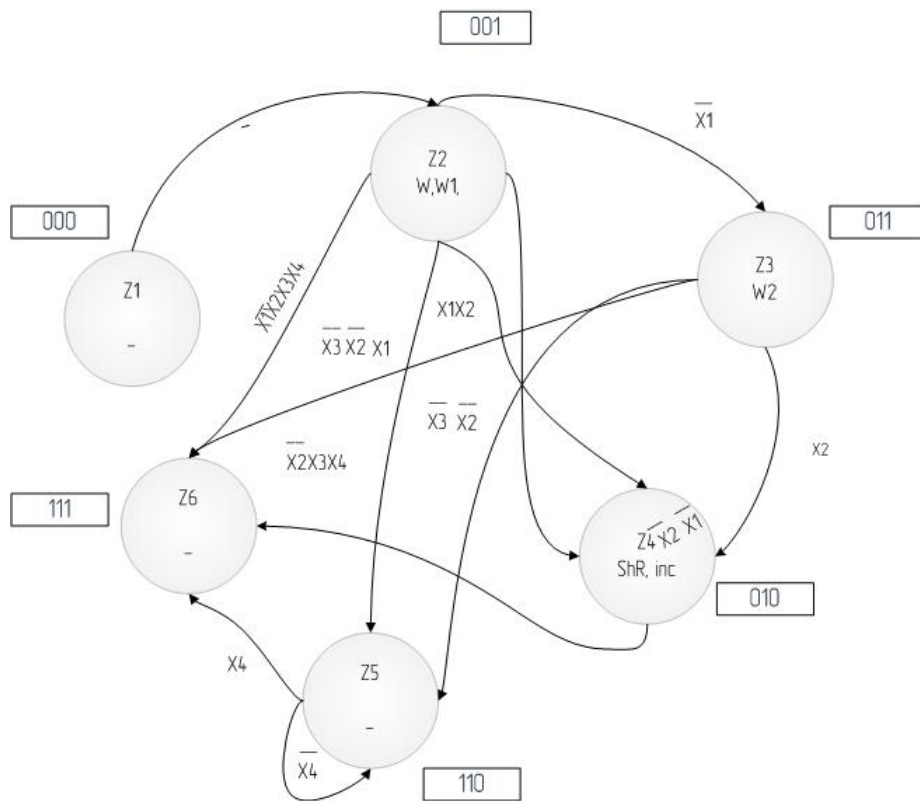


Рисунок 2.7.5 – Граф автомата Мура

2.7.8 Обробка порядків

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

2.7.9 Форма запису результату з плаваючою комою

Результат додавання $Z=X+Y$

$$Z_{\text{нк}} = 1.100000101110110$$

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

$$M_Z = -100000101110110_2$$

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

$$Z_{\text{нк}} = 1.001110100110100$$

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

$$M_Z = -001110100110100_2$$

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

Завдання 3

Синтез управляющего автомата Мура на триггерах

$x3x2x1=011_2=3_{10}$ – операція множення третім способом.

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.

Під час виконання даної розрахункової роботи я повторила матеріал курсу «Комп'ютерна логіка – 1», а також закріпила знання з курсу «Комп'ютерна логіка – 2».