

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-44
Онищенко Артемій Олександрович
Керівник: Клименко І.А.

ЗМІСТ

Обґрунтування варіанту.....	6
1 .Операція множення чисел.....	6
1.1 Перший спосіб множення.....	6
1.1.1 Теоретичне обґрунтування способу.....	6
1.1.2 Операційна схема.....	7
1.1.3 Змістовний мікроалгоритм.....	7
1.1.4 Таблиця станів регістрів.....	8
1.1.5 Функціональна схема пристрою.....	9
1.1.6 Закодований мікроалгоритм.....	9
1.1.7 Граф управляючого автомата.....	10
1.1.8 Обробка порядків	10
1.1.9 Форма подання результату в пам'яті.....	10
1.2 Другий спосіб множення.....	11
1.2.1 Теоретичне обґрунтування способу.....	11
1.2.2 Операційна схема.....	11
1.2.3 Змістовний мікроалгоритм.....	12
1.2.4 Таблиця станів регістрів.....	12
1.2.5 Функціональна схема пристрою.....	13
1.2.6 Закодований мікроалгоритм.....	13
1.2.7 Граф управляючого автомата.....	14
1.2.8 Обробка порядків	15
1.2.9 Форма подання результату в пам'яті.....	15
1.3 Третій спосіб множення.....	16
1.3.1 Теоретичне обґрунтування способу.....	16
1.3.2 Операційна схема.....	16
1.3.3 Змістовний мікроалгоритм.....	17
1.3.4 Таблиця станів регістрів.....	17

1.3.5 Функціональна схема пристрою.....	18
1.3.6 Закодований мікроалгоритм.....	19
1.3.7 Граф управляючого автомата.....	19
1.3.8 Обробка порядків	19
1.3.9 Форма подання результату в пам'яті.....	20
1.4 Четвертий спосіб множення.....	20
1.4.1 Теоретичне обґрунтування способу.....	20
1.4.2 Операційна схема.....	20
1.4.3 Змістовний мікроалгоритм.....	21
1.4.4 Таблиця станів регістрів.....	21
1.4.5 Функціональна схема пристрою.....	22
1.4.6 Закодований мікроалгоритм.....	22
1.4.7 Граф управляючого автомата.....	23
1.4.8 Обробка порядків	24
1.4.9 Форма подання результату в пам'яті.....	24
2 Операція ділення чисел.....	24
2.1 Перший спосіб ділення.....	24
2.1.1 Теоретичне обґрунтування способу.....	24
2.1.2 Операційна схема.....	24
2.1.3 Змістовний мікроалгоритм.....	25
2.1.4 Таблиця станів регістрів.....	25
2.1.5 Функціональна схема пристрою.....	27
2.1.6 Закодований мікроалгоритм.....	27
2.1.7 Граф управляючого автомата.....	28
2.1.8 Обробка порядків	28
2.1.9 Форма подання результату в пам'яті.....	29
2.2 Другий спосіб ділення.....	29
1.1.1 Теоретичне обґрунтування способу.....	29

1.1.2 Операційна схема.....	29
1.1.3 Змістовний мікроалгоритм.....	30
1.1.4 Таблиця станів регістрів.....	30
1.1.5 Функціональна схема пристрою.....	32
1.1.6 Закодований мікроалгоритм.....	32
1.1.7 Граф управляючого автомата.....	33
1.1.8 Обробка порядків	33
1.1.9 Форма подання результату в пам'яті.....	34
3 Операція додавання чисел.....	34
3.1 Теоретичне обґрунтування способу.....	34
3.2 Операційна схема.....	34
3.3 Змістовний мікроалгоритм.....	35
3.4 Таблиця станів регістрів.....	36
3.5 Функціональна схема пристрою.....	36
3.6 Закодований мікроалгоритм.....	36
3.7 Граф управляючого автомата.....	37
3.8 Обробка порядків	37
3.9 Форма подання результату в пам'яті.....	38
4 Операція добування кореня.....	38
4.1 Теоретичне обґрунтування способу.....	38
4.2 Операційна схема.....	39
4.3 Змістовний мікроалгоритм.....	39
4.4 Таблиця станів регістрів.....	40
4.5 Функціональна схема пристрою.....	41
4.6 Закодований мікроалгоритм.....	41
4.7 Граф управляючого автомата.....	42
4.8 Обробка порядків	43
4.9 Форма подання результату в пам'яті.....	43

5 Синтез управляючого автомата для операційного пристрою	43
5.1 Таблиця співвідношення управляючих входів операційного автомата і виходів управляючого автомата	43
5.2 Мікроалгоритм в термінах управляючого автомата.....	43
5.3 Структурна таблиця автомата.....	44
5.4 Синтез функцій виходів і переходів.....	44
5.5 Функціональна схема пристрою	46
Висновок.....	47

Обґрунтування варіанту

Номер залікової книжки: $4419_{10}=1000101000011_2$

$$X_2 = -10110101,0000111$$

$$Y_2 = +10110,1010000111$$

Виконання роботи

Завдання 1

В прямому коді:

В прямому коді:

$$X_2: \text{Зн.Р} \quad P = +0001000_2 \quad \text{Зн.М} \quad M = -,101101010000111$$

$$Y_2: \text{Зн.Р} \quad P = +0000101_2 \quad \text{Зн.М} \quad M = +,101101010000111$$

$$E = P + 2^8$$

$$X_2: \text{Зн.Е} \quad E = 10001000_2 \quad \text{Зн.М} \quad M = -,101101010000111$$

$$Y_2: \text{Зн.Р} \quad E = 10000101_2 \quad \text{Зн.М} \quad M = +,101101010000111$$

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. Під час множення чисел 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 двох чисел Y та X може бути реалізоване шляхом виконання визначеного циклічного процесу, характер якого залежить від конкретної форми заданого виразу. Множення мантис першим способом здійснюється з молодших розрядів множника, сума часткових добутків зсувається вправо, а множене залишається нерухомим. Під час множення першим способом в першому такті i -го циклу аналізується значення $RG2(n)$ — молодшого n -го розряду регістру $RG2$, в якому знаходиться чергова цифра множника. Вміст $RG3$ додається до суми часткових добутків, що зняються в регістрі $RG1$, якщо $RG2(n)=1$, або не додається, якщо $RG2(n)=0$. В другому такті здійснюється правий зсув у регістрах $RG1$ і $RG2$, що еквівалентно множенню їх вмісту на $2(-1)$. За зсуву цифра молодшого розряду регістру $RG1$ записується у вивільнюваний старший розряд регістру $RG2$. Після виконання n циклів молодші розряди $2n$ -розрядного добутку будуть записані в регістр $RG2$, а старші — в $RG1$. Час множення, якщо не застосовуються методи прискорення операції, визначається виразом:

$$t_m = n(t_n + t_3),$$

де t_n і t_3 — тривалості тактів підсумовування і зсуву відповідно.

Добуток двох чисел представляється у вигляді:

$$Z = \left(\left(\dots \left((0 + Y \cdot x_n) \cdot 2^{-1} + Y \cdot x_{n-1} \right) \cdot 2^{-1} \dots \right) + Y \cdot x_1 \right) \cdot 2^{-1}. \text{ Отже, сума часткових добутків}$$

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.

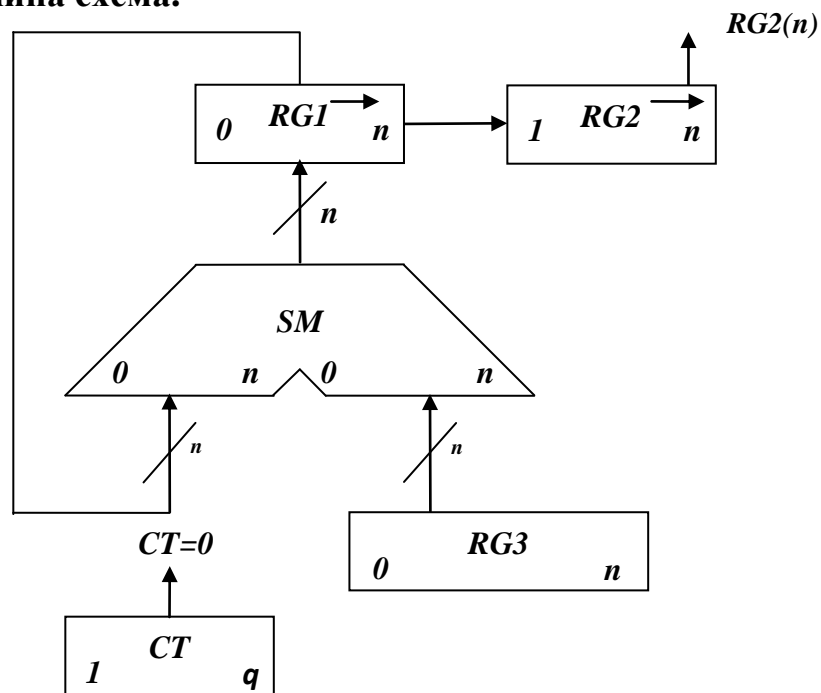


Рисунок 1.1- Операційна схема.

1.1.3 Змістовний мікроалгоритм.

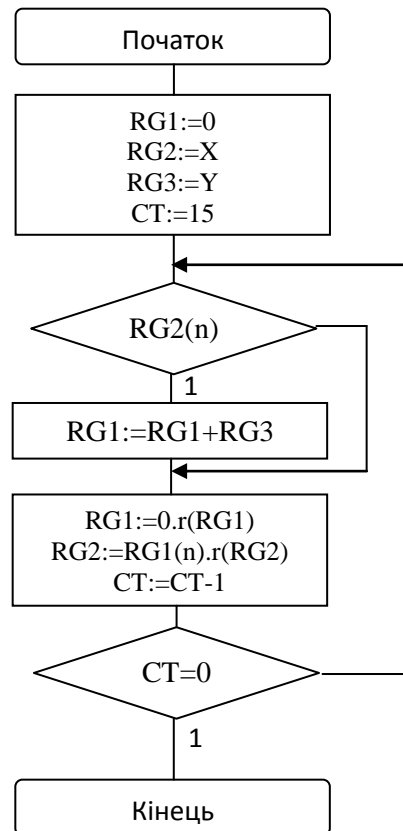


Рисунок 1.2- Змістовний мікроалгоритм.

1.1.4 Таблиця станів регістрів.

Таблиця 1.1-Таблиця станів регістрів.

№ ц.	RG1	RG2	RG3	СТ
П.С.	0000000000000000	101101010000111	10110101000011	1111
1→	00010110101000011	110110101000011		1110
2→	+ <u>00101101010000111</u> 1000011111001010 0100001111100101	011011010100001		1101
3→	+ <u>00101101010000111</u> 1001111001101100 0100111100110110	001101101010000		1100
4→	0010011110011011	000110110101000		1011
5→	0001001111001101	100011011010100		1010
6→	0000100111100110	110001101101010		1001
7→	0000010011110011	011000110110101		1000
8→	+ <u>00101101010000111</u> 0101111101111010 0010111110111101	001100011011010		0111
9→	0001011111011110	100110001101101		0110
10→	+ <u>00101101010000111</u> 0111001001100101 0011100100110010	110011000110110		0101
11→	0001110010011001	011001100011011		0100
12→	+ <u>00101101010000111</u> 0111011100100000 0011101110010000	001100110001101		0011
13→	+ <u>00101101010000111</u> 1001011000010111 0100101100001011	100110011000110		0010
14→	0010010110000101	110011001100011		0001
15→	+ <u>00101101010000111</u> 1000000000001100 0100000000000110	011001100110001		0000

1.1.5 Функціональна схема з відображенням управляючих сигналів.

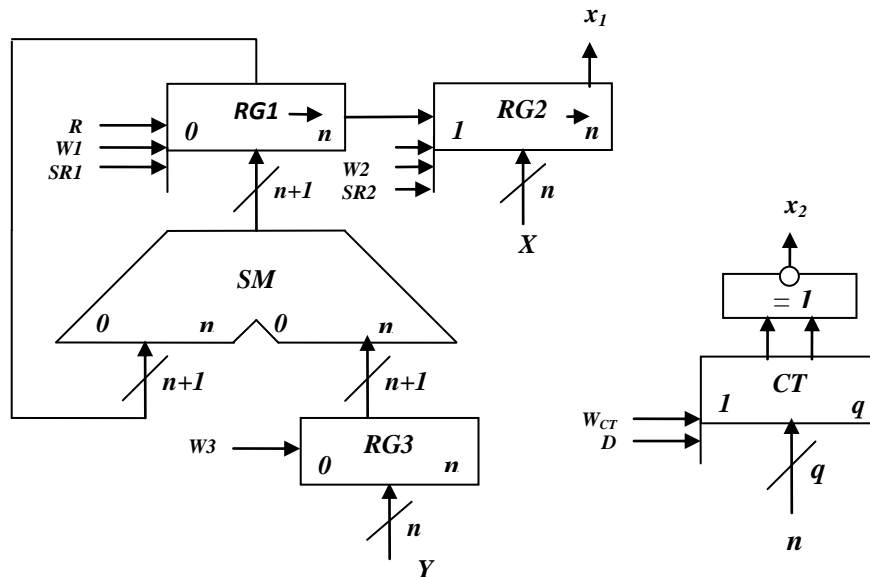


Рисунок 1.3- Функціональна схема.

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
RG1:=0	R	RG2(n)	X1
RG2:=X	W2	CT=0	X2
RG3:=Y	W3		
CT:=15	W _{CT}		
RG1:=RG1+RG3	W1		
RG1:=0.r(RG1)	SR1		
RG2:=RG1(n).r(RG2)	SR2		
CT:=CT-1	D		

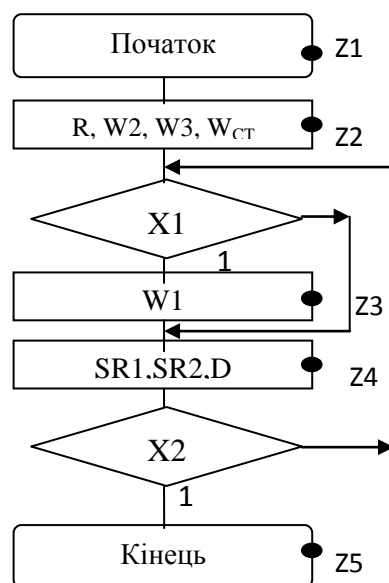


Рисунок 1.4-Закодований мікроалгоритм.

1.1.7 Граф управляючого автомата Мура з кодами вершин.

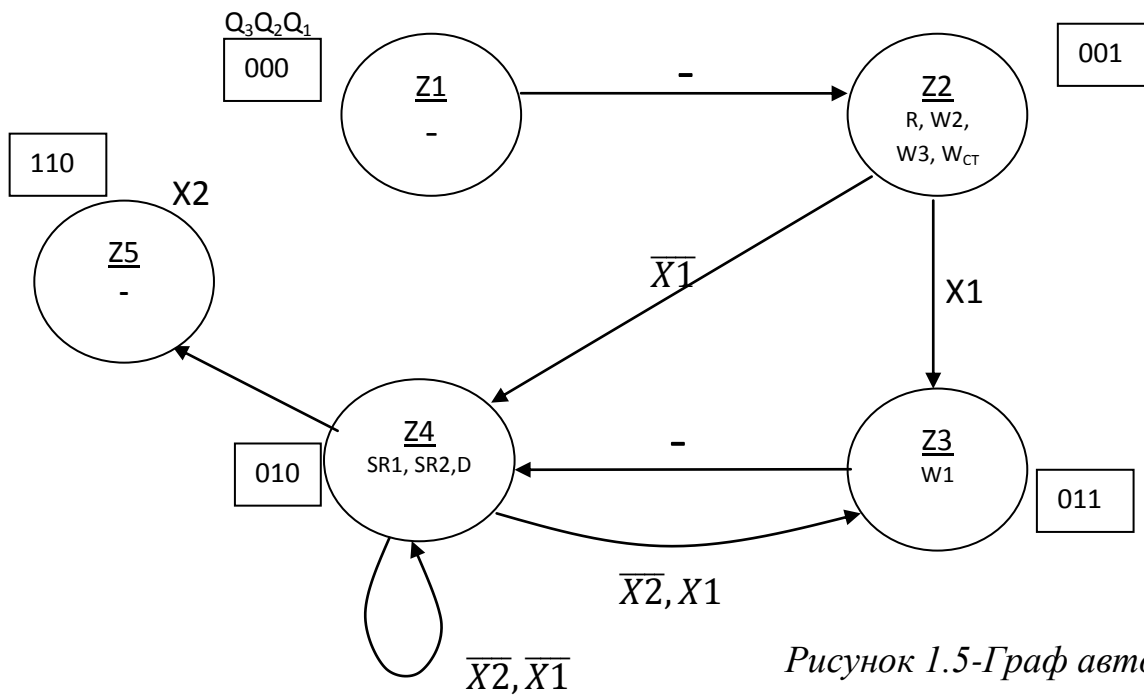


Рисунок 1.5-Граф автомата Мура

1.1.8 Обробка порядків і нормалізація

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

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

$$M_z = , 1000000000001100$$

$$P_z = 1101.$$

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

Зн.Р	Р=+13 ₁₀							Зн.М	М															
0	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	0		

1.2 Другий спосіб множення

1.2.1 Теоретичне обґрунтування

Числа множаться у прямих кодах. Під час множення чисел 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 другим способом здійснюється з молодших розрядів, множене зсувається вліво, а сума часткових добутків залишається нерухомою. Перед початком множення другим способом множник X записують у регістр RG2, а множене Y – в молодші розряди регістру RG3(тобто в регістрі RG3

установлюють $Y_0=Y2^{-n}$). В кожному i -му циклі множення додаванням кодів RG3 і RG1 керує цифра RG2(n), а в регістрі RG3 здійснюється зсув вліво на один розряд, у результаті чого формується величина $Y_i=2Y_{i-1}$. Оскільки сума часткових добутоків у процесі множення нерухома, зсув у регістрі RG3 можна сполучити в часі з підсумовуванням (як правило, $t_{\Pi} > t_3$). Завершення операції множення визначається за нульовим вмістом регістру RG2, що також приводить до збільшення швидкодії, якщо множник ненормалізований.

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 добутоків 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

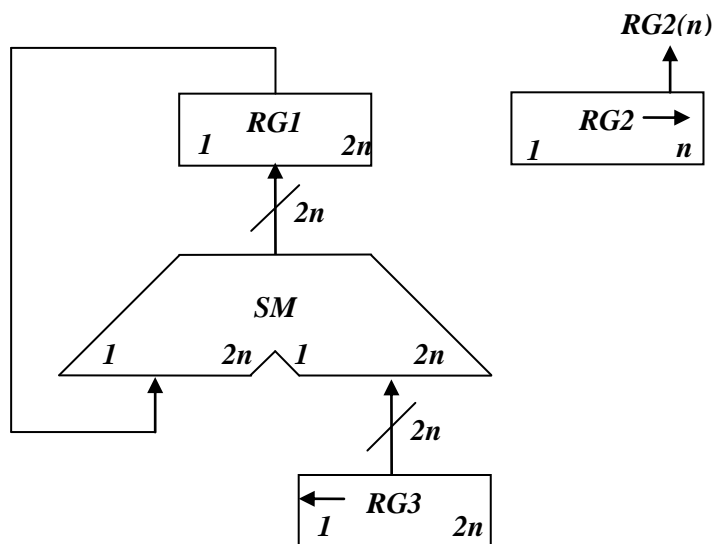


Рисунок 1.6-Операційна схема

1.2.3 Змістовний мікроалгоритм

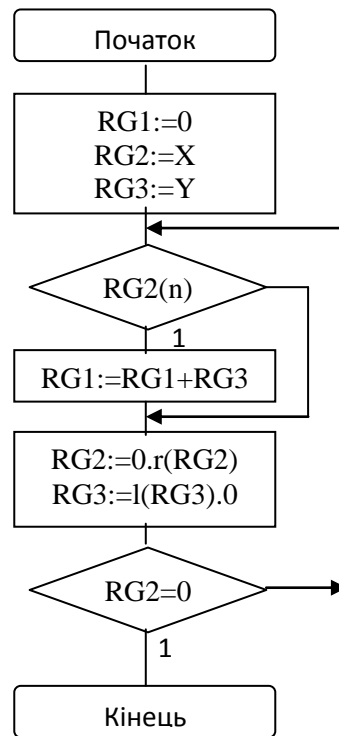


Рисунок 1.7-Змістовний мікроалгоритм

1.2.4 Таблиця станів регістрів

Таблиця 1.3- Таблиця станів регістрів

№п.	RG1	RG2→	RG3←
П.С.	00000000000000000000000000000000	101101010000111	0000000000000000101101010000111
1	00000000000000000101101010000111	010110101000011	00000000000000001011010100001110
2	+ 000000000000000001011010100001110 0000000000000010000111110010101	001011010100001	000000000000000010110101000011100
3	+ 0000000000000000010110101000011100 0000000000000100111100110110001	000101101010000	0000000000000000101101010000111000
4	0000000000000100111100110110001	000010110101000	00000000000001011010100001110000
5	0000000000000100111100110110001	000001011010100	00000000000010110101000011100000
6	0000000000000100111100110110001	000000101101010	00000000001011010100001110000000
7	0000000000000100111100110110001	000000010110101	00000000010110101000011100000000
8	+ 00000000010110101000011100000000 000000001011111011110100110001	000000001011010	00000000101101010000111000000000
9	000000001011111011110100110001	000000000101101	00000001011010100001110000000000
10	+ 00000001011010100001110000000000 000000111001001100101100110001	000000000010110	00000010110101000011100000000000
11	000000111001001100101100110001	000000000001011	00000101101010000111000000000000
12	+ 00000101101010000111000000000000 0000111011100100000001100110001	000000000000101	00001011010100001110000000000000
13	+	000000000000010	00010110101000011100000000000000

	<u>0000101101010000111000000000000</u> <u>001001011000010111001100110001</u>		
14	001001011000010111001100110001	0000000000000001	0010110101000011100000000000000
15	+ <u>0010110101000011100000000000000</u> <u>100000000000110011001100110001</u>	0000000000000000	0101101010000111000000000000000

1.2.5 Функціональна схема з відображенням управляючих сигналів

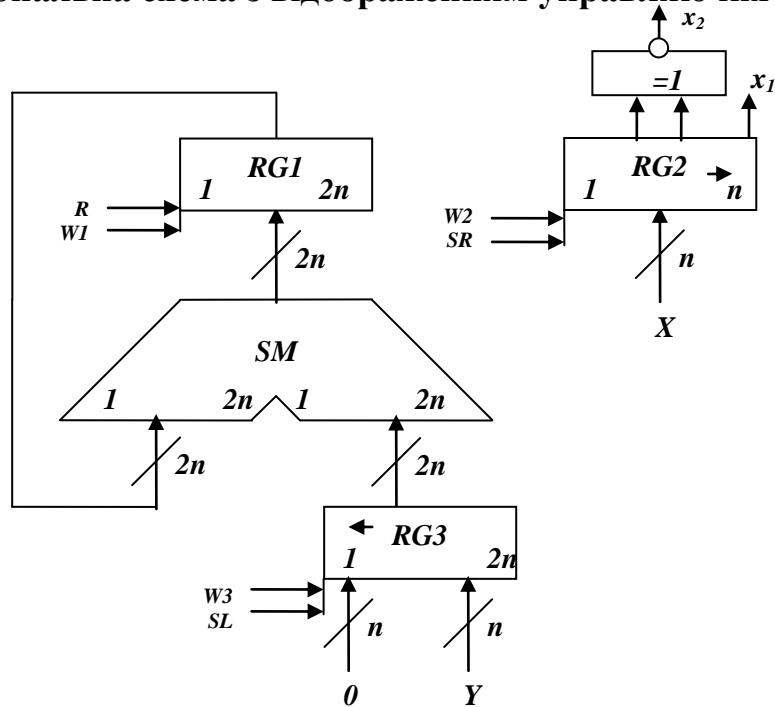


Рисунок 1.8- Функціональна схема

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(RG2)	SR
RG3:=l(RG3).0	SL

Таблиця кодування логічних умов	
ЛУ	Позначення
RG2(n)	X1
RG2=0	X2

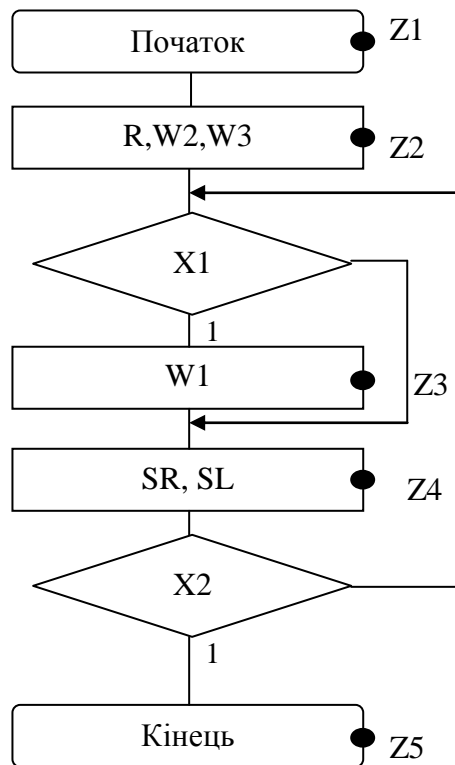
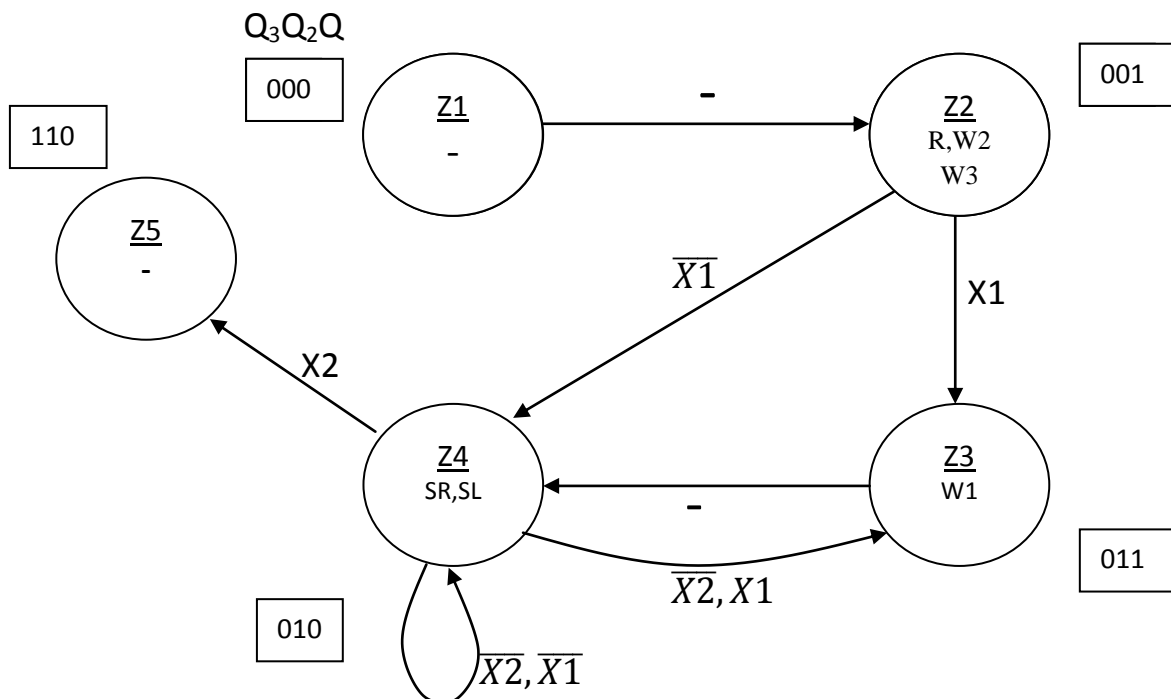


Рисунок 1.9- Закодований мікроалгоритм.

1.2.7 Граф управляючого автомата Мура з кодами вершин



1.2.8 Обробка порядків і нормалізація

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

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

Нормалізація мантиси не потрібна.

$$M_z = , 10000000000001100$$

$$P_z = 1101.$$

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

Зн.Р								Зн.М																М															
0.	0	0	0	1	1	0	1	1,	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0							

1.3 Третій спосіб множення

1.3.1 Теоретичне обґрунтування

Числа множаться у прямих кодах. Під час множення чисел 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 третім способом здійснюється зі старших розрядів множника, сума часткових добутків зсувається вліво, а множене нерухоме. Під час множення третім способом вага молодшого розряду RG3 дорівнює 2^{-2n} , тому код у регістрі RG3 являє собою значення $Y2^{-n}$. На початку кожного циклу множення здійснюється лівий зсув у регістрах RG1 і RG2, а потім виконується додавання, яким керує RG1(1). У результаті підсумовування вмісту RG3 і RG1 може виникнути перенос у молодший розряд регістру RG1. У старшій частині суматора, на якому здійснюється підсумовування коду RG2 з нулями, відбувається поширення переносу. Збільшення довжини RG2 на один розряд усуває можливість поширення переносу в розряди множника. Після виконання n циклів молодші розряди добутку будуть знаходитися в регістрі RG2, а старші – в регістрі RG1. Час множення третім способом визначається аналогічно першому способу і дорівнює $t_m = n(t_{\Pi} + t_3)$.

Вираз

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

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

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

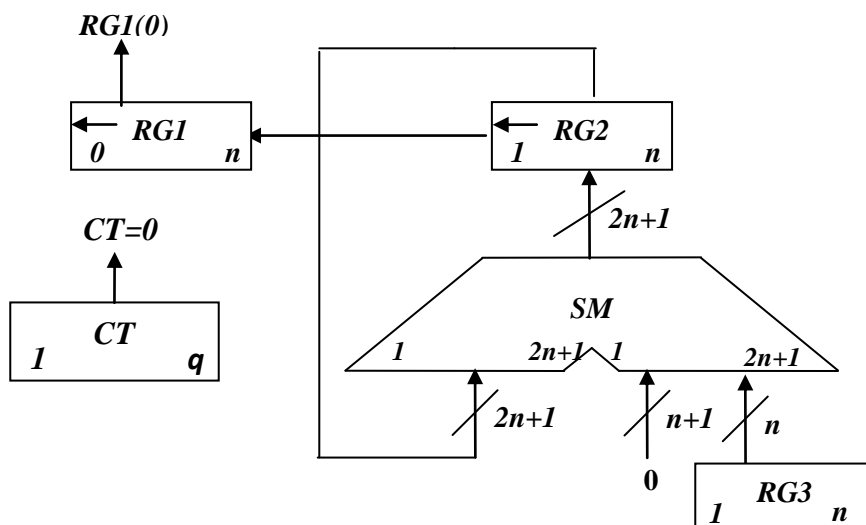


Рисунок 1.11-Операційна схема

1.3.3 Змістовний мікроалгоритм

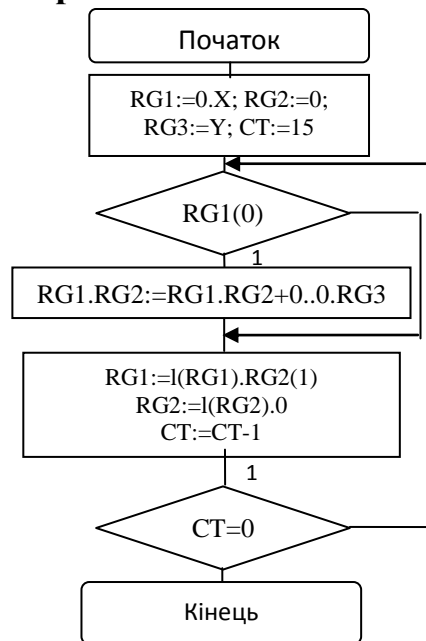


Рисунок 1.12-Змістовний мікроалгоритм

1.3.4 Таблиця станів регістрів

Таблиця 1.5- Таблиця станів регістрів

№ ц.	RG1	RG2	RG3	CT
П.С.	001101010000111	00000000000000000000000000000000	10110101000011	1111
1←	011010100001110	00000000000000001011010100001110		1110
2	110101000011100	00000000000000010110101000011100		1101
3	101010000111000	+ 0000000000000000101101010000111 000000000000011100010010100011 000000000000111000100101000110		1100
4←	010100001110000	+ 0000000000000000101101010000111 000000000000111110001111001101 000000000001111100011110011010		1011
5	101000011100000	000000000011111000111100110100		1010
6	010000111000000	+ 0000000000000000101101010000111 00000000001111110100110111011 000000000111111101001101110110		1001
7	100001110000000	000000001111111010011011101100		1000
8←	000011100000000	+ 0000000000000000101101010000111 0000000100000000000000101110011 00000010000000000000001011100110		0111
9←	000111000000000	0000010000000000000010111001100		0110
10	001110000000000	000010000000000000101110011000		0101
11	011100000000000	0001000000000000001011100110000		0100
12	111000000000000	001000000000000010111001100000		0011

13←	1100000000000000	+ 00000000000000000101101010000111 001000000000000001000100011100111 0100000000000000010001000111001110		0010
14←	1000000000000000	+ 00000000000000000101101010000111 0100000000000000010110110001010101 10000000000000000101101100010101010		0001
15	0000000000000000	+ 00000000000000000101101010000111 10000000000000000110011001100110001		0000

1.3.5 Функціональна схема з відображенням управляючих сигналів

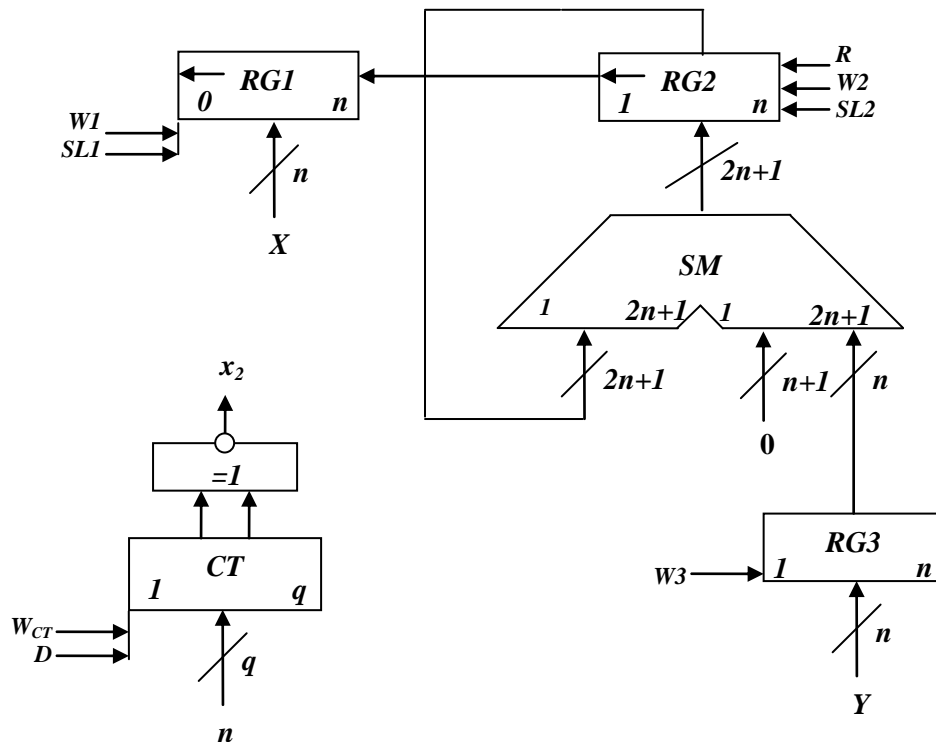


Рисунок 1.13-Функціональна схема

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).RG2(1)	SL1
RG2:=l(RG2).0	SL2
CT:=CT-1	D

Таблиця кодування логічних умов	
ЛІУ	Позначення
RG1(0)	X1
CT=0	X2

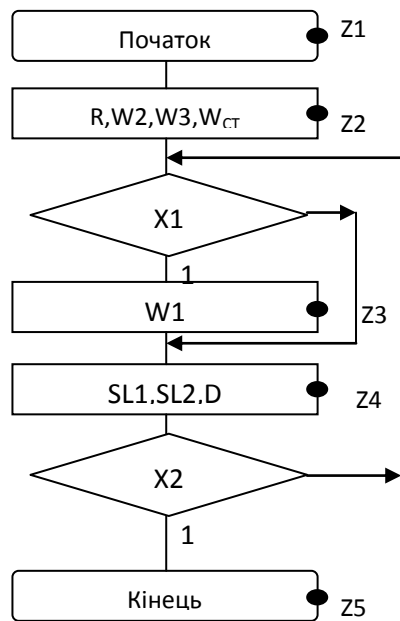


Рисунок 1.14- Закодований мікроалгоритм

1.3.7 Граф управляючого автомата Мура з кодами вершин

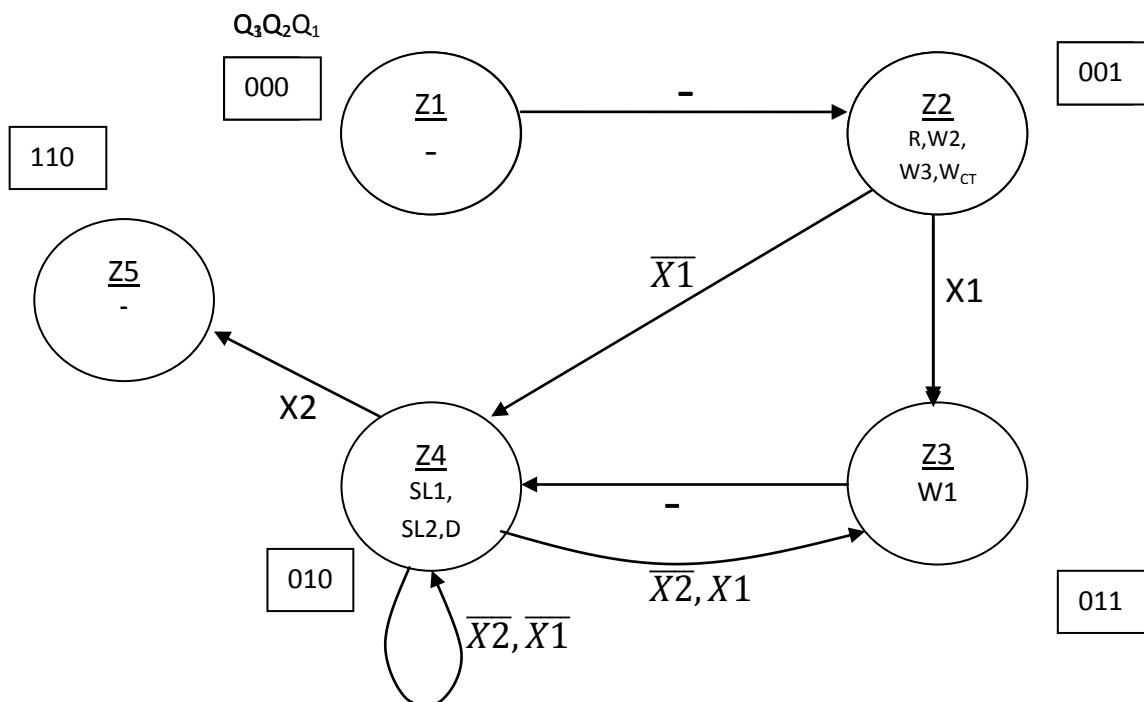


Рисунок 1.15- Граф автомата Мура

1.3.8 Обробка порядків і нормалізація

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

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

$$M_z = , 1000000000001100$$

$$P_z = 1101.$$

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

Зн.Р	P=+13 ₁₀	Зн.М	М
0 0 0 0 1 1 0 1		1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0	

1.4 Четвертий спосіб множення

1.4.1 Теоретичне обґрунтування

Числа множаться у прямих кодах. Під час множення чисел 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 виконується зі старших розрядів множника, сума часткових добутків залишається нерухомою, а множене зсувається вправо. Перед множенням четвертим способом множник записують у регістр RG2, а множене – в старші розряди регістру RG3 (тобто в RG3 установлюють $Y_0=Y \cdot 2^{-1}$). У кожному циклі цифра RG2(1), що знаходиться в старшому розряді регістру RG2, керує підсумовуванням, а в RG3 здійснюється правий зсув на один розряд, що еквівалентно множенню вмісту цього регістра на 2^{-1} . Час виконання множення четвертим способом складає $t_M = nt_P$, визначається аналогічно до другого способу. Запишу четвертий спосіб в аналітичній формі. Вираз $Z = Y \cdot x_n \cdot 2^{-n} + Y \cdot x_{n-1} \cdot 2^{-n+1} + \dots + Y \cdot x_1 \cdot 2^{-1}$, подамо у вигляді $Z = ((\dots ((0 + Y \cdot 2^{-1} \cdot x_1) + Y \cdot 2^{-2} \cdot x_2) + \dots + Y \cdot 2^{-k} \cdot x_k) + \dots + Y \cdot 2^{-n} \cdot x_n)$.

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

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

1.4.2 Операційна схема

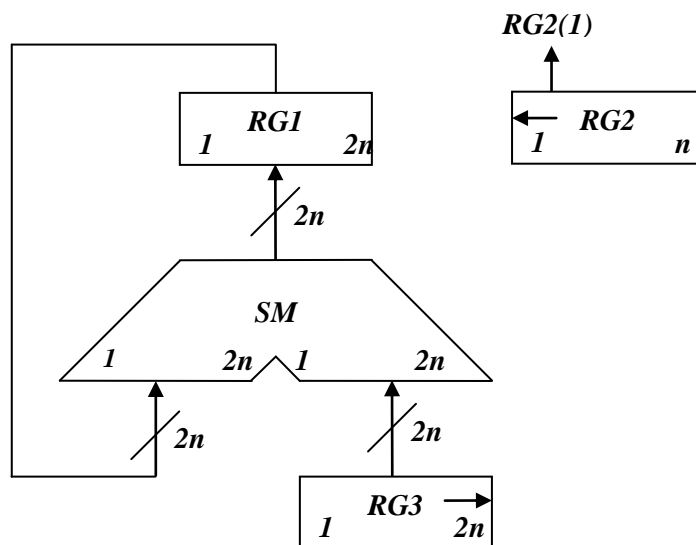


Рисунок 1.16-Операційна схема

1.4.3 Змістовний мікроалгоритм

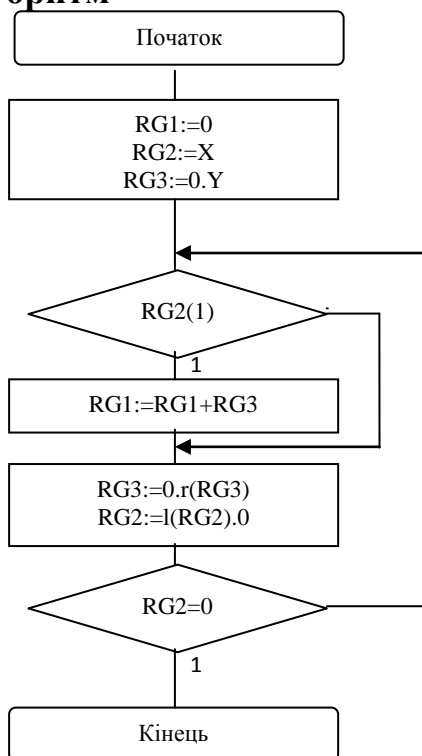


Рисунок 1.17-Змістовний мікроалгоритм

1.4.4 Таблиця станів регістрів

Таблиця 1.7- Таблиця станів регістрів

№ ц.	RG1	RG2←	RG3→
П.С.	00000000000000000000000000000000	001101010000111	00101101010000111000000000000000
1	00101101010000111000000000000000	011010100001110	00010110101000011100000000000000
2	00101101010000111000000000000000	110101000011100	00001011010100001110000000000000
3	+ 00001011010100001110000000000000 01110001001010001100000000000000	101010000111000	00000101101010000111000000000000
4	+ 00000101101010000111000000000000 01111100011110011010000000000000	010100001110000	00000010110101000011100000000000
5	01111100011110011010000000000000	101000011100000	00000001011010100001110000000000
6	+ 00000000101101010000111000000000 01111111010011011101100000000000	010000111000000	00000000101101010000111000000000
7	01111111010011011101100000000000	100001110000000	00000000010110101000011100000000
8	+ 00000000001011010100001110000000 10000000000000010111001100000000	000011100000000	00000000001011010100001110000000
9	10000000000000010111001100000000	000111000000000	00000000000101101010000111000000
10	10000000000000010111001100000000	001110000000000	00000000000010110101000011100000
11	10000000000000010111001100000000	011100000000000	00000000000001011010100001110000
12	10000000000000010111001100000000	111000000000000	00000000000000101101010000111000

13	+ 000000000000000010110101000011100 1000000000000100010001110011100	1100000000000000	00000000000000001011010100001110
14	+ 00000000000000001011010100001110 1000000000000101101100010101010	1000000000000000	0000000000000000101101010000111
15	+ 0000000000000000101101010000111 1000000000000110011001100110001	0000000000000000	000000000000000010110101000011

1.4.5 Функціональна схема з відображенням управляючих сигналів

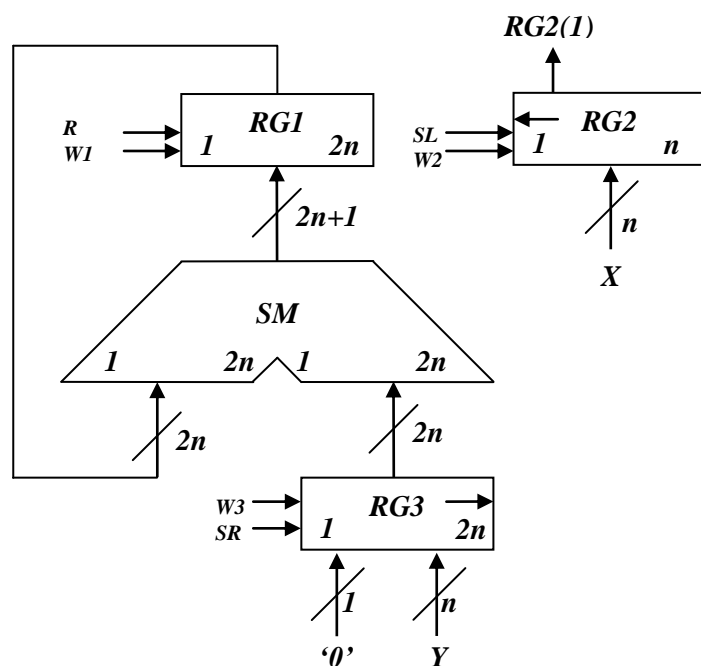


Рисунок 1.18-Функціональна схема

1.4.6 Закодований мікроалгоритм

Таблиця 1.8- Таблиця кодування мікрооперацій

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

Таблиця кодування логічних умов	
ЛЮ	Позначення
RG2(1)	X1
RG2=0	X2

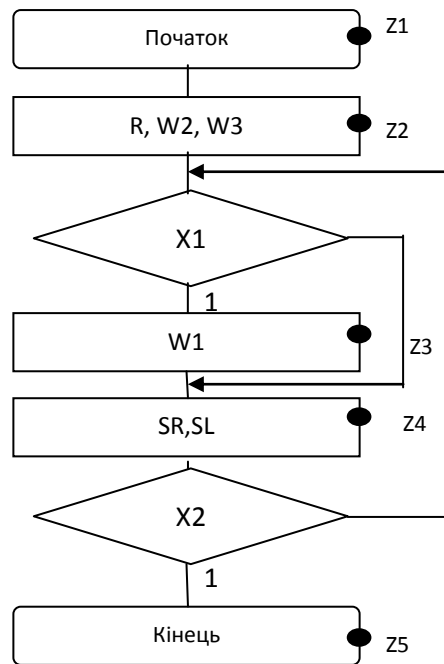


Рисунок 1.19- Закодований мікроалгоритм

1.4.7 Граф управляючого автомата Мура з кодами вершин

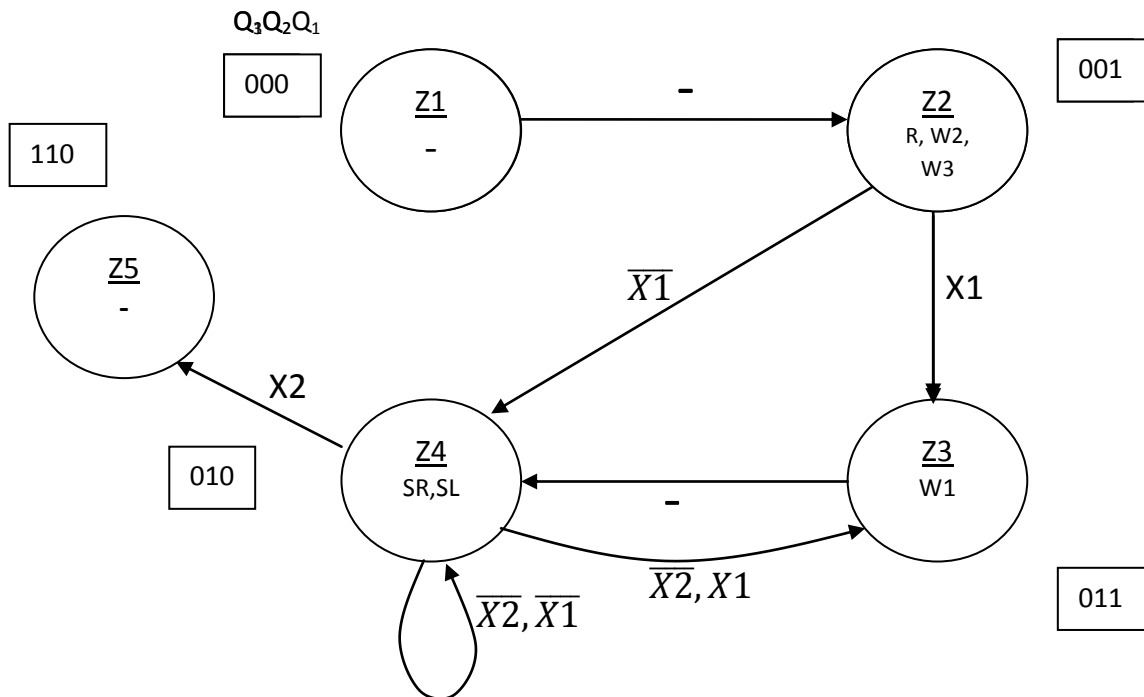


Рисунок 1.20- Граф автомата Мура

1.4.8 Обробка порядків і нормалізація

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

Знак мантиси: $1 \oplus 0 = 1$.

Нормалізація мантиси не потрібна.

$$M_z = , 10000000000001100$$

$$P_z = 1101.$$

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

3_H.P P=+13₁₀

0.	0	0	0	1	1	0	1
----	---	---	---	---	---	---	---

3_H.M M

1,	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
----	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

2. Операція ділення чисел

$Z=X: Y$, X -ділене, Y – дільник.

2.1 Перший спосіб ділення

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. Такий спосіб називається діленням із зсувом залишку. Черговий залишок формується в регістрі RG2 (у вихідному стані в цьому регістрі записаний X), дільник Y знаходиться в регістрі RG1. Виходи RG2 підключені до входів SM безпосередньо, тобто ланцюги видачі коду з RG2 не потрібні. Час для підключення $n+1$ цифри частки визначається виразом $t=(n+1)(tt+tc)$, де tt - тривалість виконання мікрооперації додавання-віднімання; tc - тривалість виконання мікрооперації зсуву. Результат формується в регістрі RG3.

2.1.2 Операційна схема

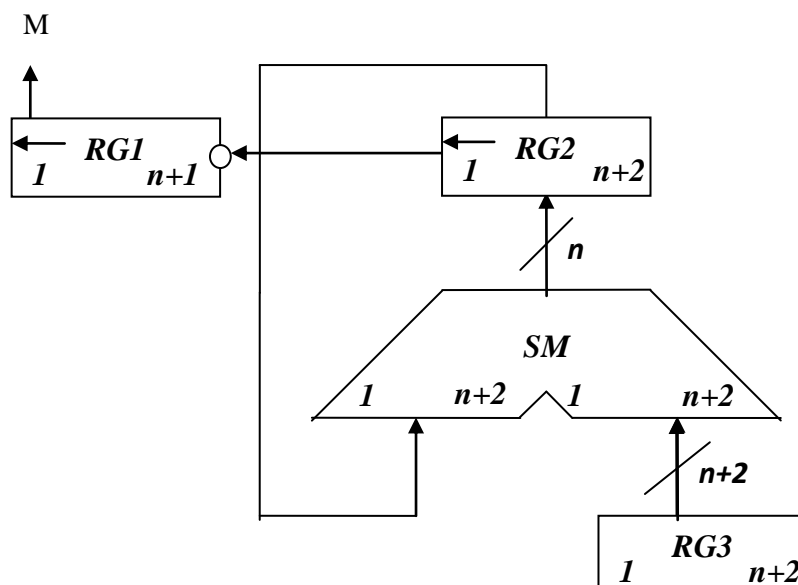


Рисунок 2.1-Операційна схема

2.1.3 Змістовний мікроалгоритм

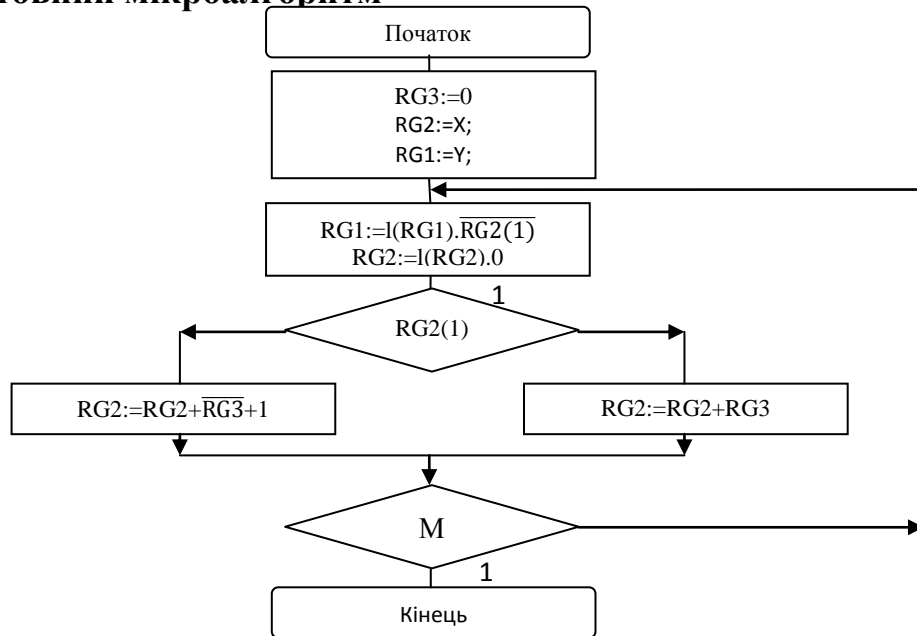


Рисунок 2.2-Змістовний мікроалгоритм

2.1.4 Таблиця станів регістрів

Таблиця 2.1- Таблиця станів регістрів

№ ц.	RG3	RG2	RG1
П.С.	0000000000000000	00101101010000111	00101101010000111
1	0000000000000001	01011010100001110 + <u>11101001010111101</u> 01000011111001011	
2	0000000000000011	10000111110010110 + <u>000101101010000111</u> 10110101000011101	
3	0000000000000110	01101010000111010 + <u>11101001010111101</u> 01010011011110111	
4	0000000000001101	10100110111101110 + <u>000101101010000111</u> 11010100001110101	
5	0000000000011010	10101000011101010 + <u>000101101010000111</u> 11010101101110001	

6	0000000000110100	10101011011100010 + <u>000101101010000111</u> 11011000101101001	
7	00000000001101000	10110001011010010 + <u>000101101010000111</u> 11011110101011001	
8	00000000011010000	10111101010110010 + <u>000101101010000111</u> 11101010100111001	
9	00000000110100000	11010101001110010 + <u>000101101010000111</u> 00000010011111001	
10	00000001101000001	00000100111110010 + <u>11101001010111101</u> 11101110010101111	
11	00000011010000010	11011100101011110 + <u>000101101010000111</u> 00001001111100101	
12	0000110100000101	00010011111001010 + <u>11101001010111101</u> 11111101010000111	
13	0001101000001010	11111010100001110 + <u>000101101010000111</u> 00100111110010101	
14	0011010000010101	01001111100101010 + <u>11101001010111101</u> 00111000111100111	
15	0110100000101011	01110001111001110 + <u>11101001010111101</u> 01011011010001011	
16	1101000001010111	10110110100010110 + <u>000101101010000111</u> 11100011110011101	

2.1.5 Функціональна схема з відображенням управляючих сигналів

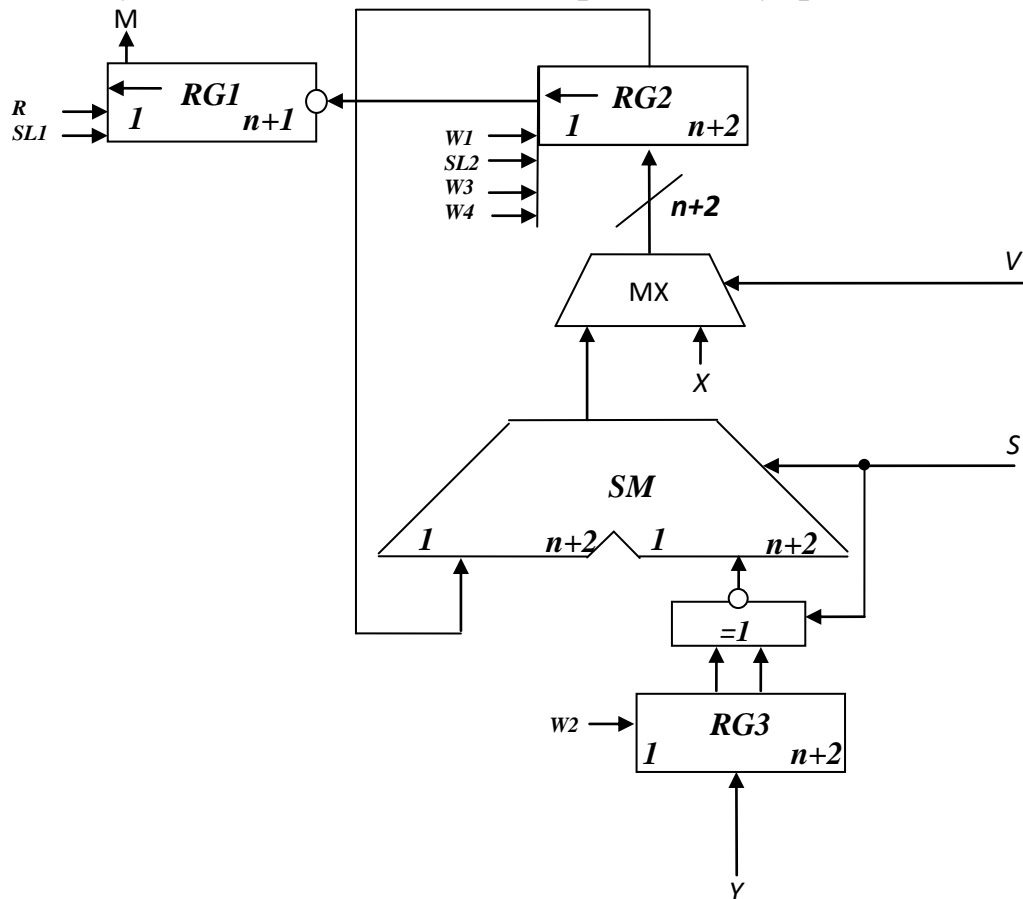


Рисунок 2.3-Функціональна схема

2.1.6 Закодований мікроалгоритм

Таблиця 2.2- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
RG3:=0	R
RG2:=X;	W1
RG1:=Y;	W2
$RG1:=l(RG1).RG2(1)$	SL1
$RG2:=l(RG2).0$	S2
$RG2:=RG2+\overline{RG1}+1$	W3
$RG2:=RG2+RG1$	W4

Таблиця кодування логічних умов	
ЛЮ	Позначення
RG2(1)	X1
M	X2

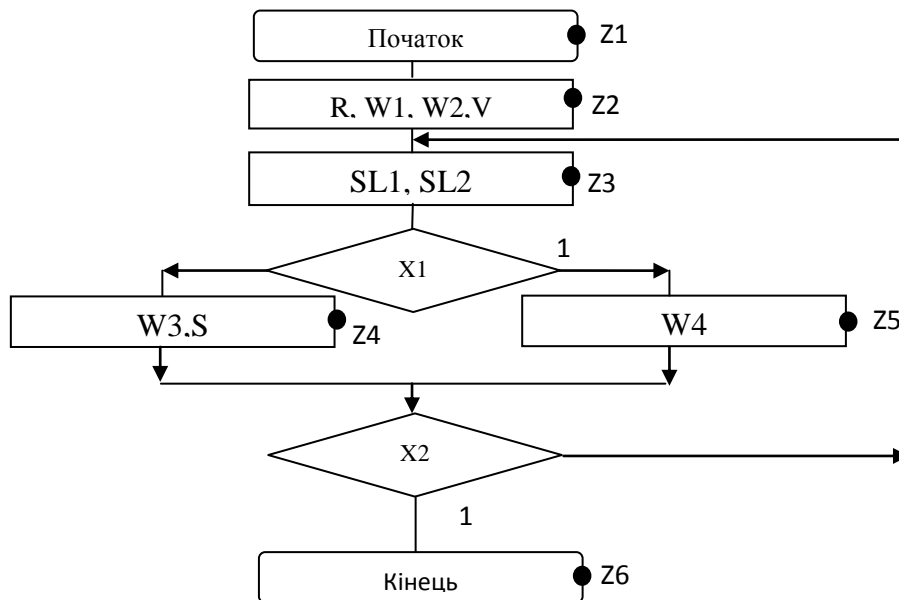


Рисунок 2.4- Закодований мікроалгоритм

2.1.7 Граф управляючого автомата Мура з кодами вершин

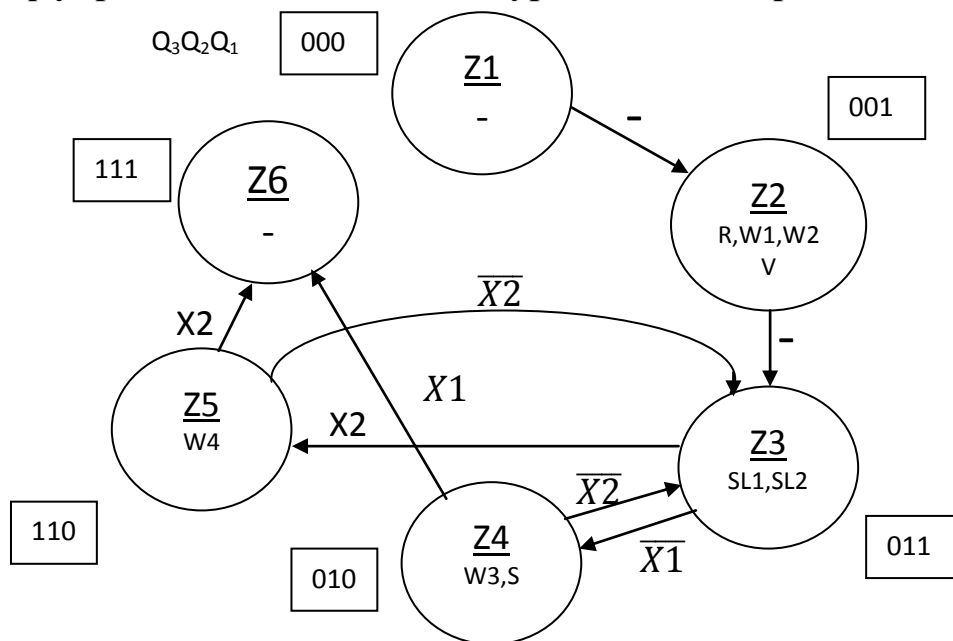


Рисунок 2.5- Граф автомата Мура

2.1.8 Обробка порядків і нормалізація

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

Нормалізація мантиси не потрібна.

$$M_z = , 1101000001010111$$

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

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

Зн.Р	P=+11 ₂	Зн.М	М
0. 0 0 0 0 0 1 1		1, 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1	

2.2 Другий спосіб ділення

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 підсумування 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

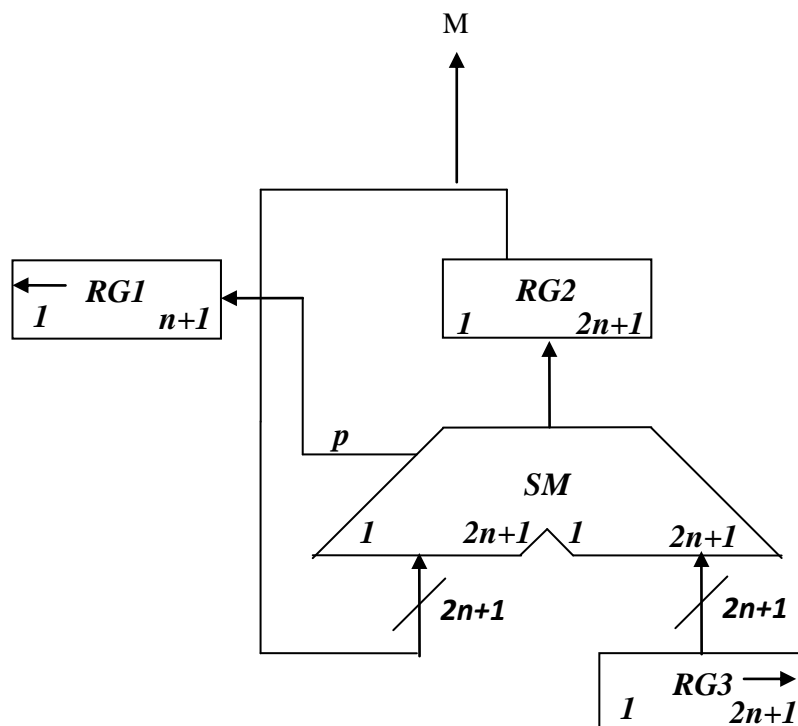


Рисунок 2.6-Операційна схема

2.2.3 Змістовний мікроалгоритм

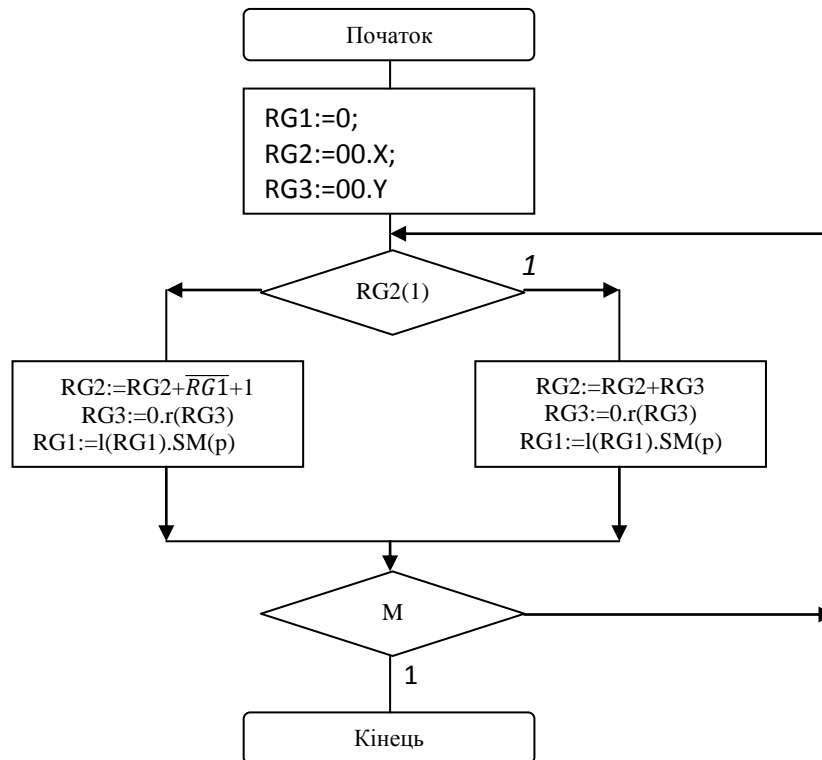


Рисунок 2.7-Змістовний мікроалгоритм

2.2.4 Таблиця станів регістрів

Таблиця 2.3- Таблиця станів регістрів

№ ц.	RG1	RG2	RG3
П.С.	0000000000000001	010110101000011100000000000000	000101101010000111000000000000 0
1	0000000000000011	010110101000011100000000000000 + <u>111010010101111001000000000000</u> 010000111110010101000000000000	000010110101000011100000000000 0
2	0000000000000110	010000111110010101000000000000 + <u>111101001010111100100000000000</u> 001110001001010001100000000000	000001011010100001110000000000 0
3	0000000000001101	001110001001010001100000000000 + <u>111110100101011110010000000000</u> 001100101110101111110000000000	000000101101010000111000000000 0
4	0000000000011010	001100101110101111110000000000 + <u>111111010010101111001000000000</u> 001100000001011110111000000000	000000010110101000011100000000 0
5	0000000000110100	001100000001011110111000000000 + <u>111111101001010111100100000000</u> 001011101010110110011100000000	000000001011010100001110000000 0

6	0000000001101000	001011101010110110011100000000 + <u>111111110100101011110010000000</u> 001011011111100010001110000000	000000000101101010000111000000 0
7	0000000011010000	001011011111100010001110000000 + <u>111111110100101011110010000000</u> 001011011001111000000111000000	000000000010110101000011100000 0
8	0000000110100000	001011011001111000000111000000 + <u>111111111010010101111001000000</u> 001011010111000011000011100000	000000000001011010100001110000 0
9	0000001101000001	001011010111000011000011100000 + <u>111111111101001010111100100000</u> 001011010101101000100001110000	000000000000101101010000111000 0
10	0000011010000010	001011010101101000100001110000 + <u>111111111110100101011110010000</u> 001011010100111011010000111000	000000000000010110101000011100 0
11	0000110100000101	001011010100111011010000111000 + <u>111111111111010010101111001000</u> 001011010100100100101000011100	000000000000001011010100001110 0
12	0001101000001010	001011010100100100101000011100 + <u>111111111111101001010111100100</u> 001011010100011001010100001110	000000000000000101101010000111 0
13	0011010000010101	001011010100011001010100001110 + <u>111111111111110100101011110010</u> 001011010100010011101010000111	000000000000000010110101000011 1
14	0110100000101011	001011010100010011101010000111 + <u>111111111111111010010101111010</u> 001011010100010000110101000100	000000000000000001011010100001 1
15	1101000001010111	001011010100010000110101000100 + <u>111111111111111101001010111111</u> 001011010100001111011010100011	000000000000000000101101010000 1

2.2.5 Функціональна схема з відображенням управляючих сигналів

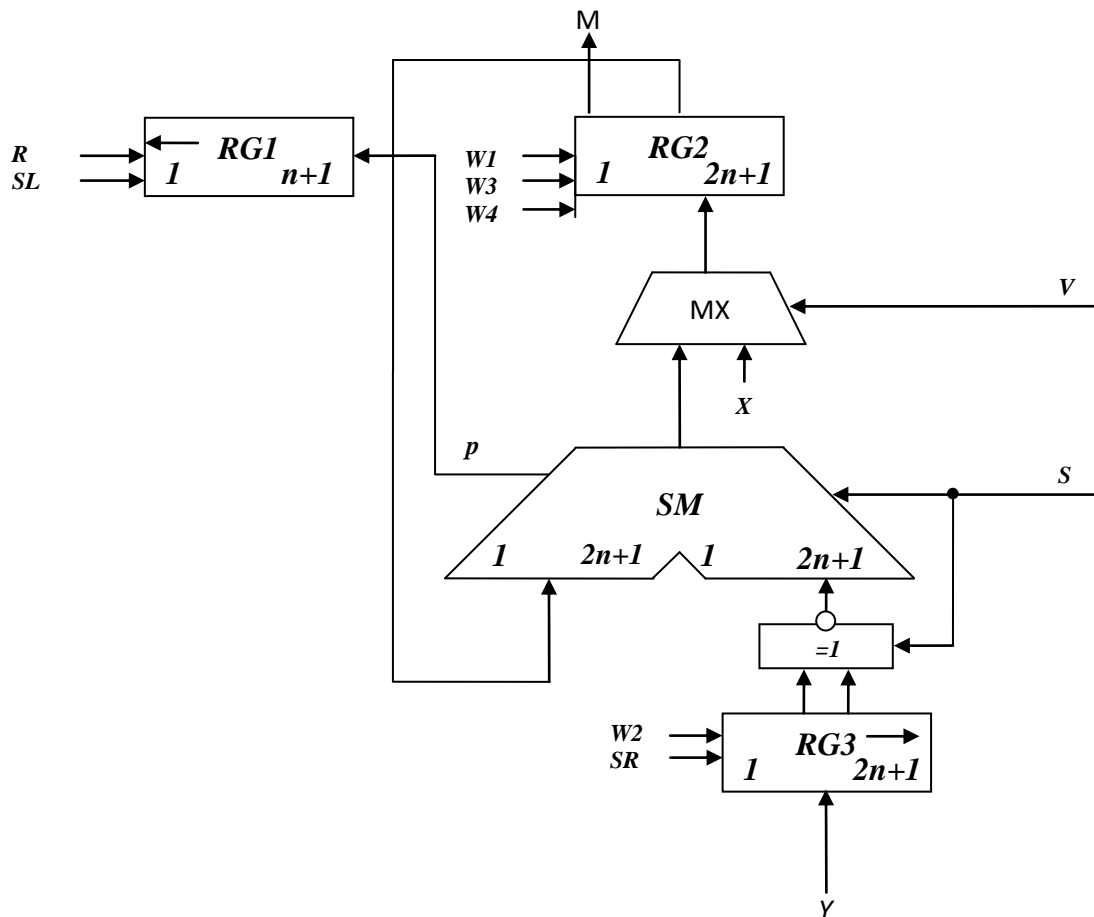


Рисунок 2.8-Функціональна схема

2.2.6 Закодований мікроалгоритм

Таблиця 2.4- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій		Таблиця кодування логічних умов	
МО	УС	ЛУ	Позначення
RG1:=0;	R	RG2(1)	X1
RG2:=00.X	W1	M	X2
RG3:=00.Y	W2		
RG2:=RG2+RG1	W3		
RG2:=RG2+ $\overline{RG1}+1$	W4		
RG1:=0.r(RG1)	SR		
RG3:=l(RG3).SM(p)	SL		

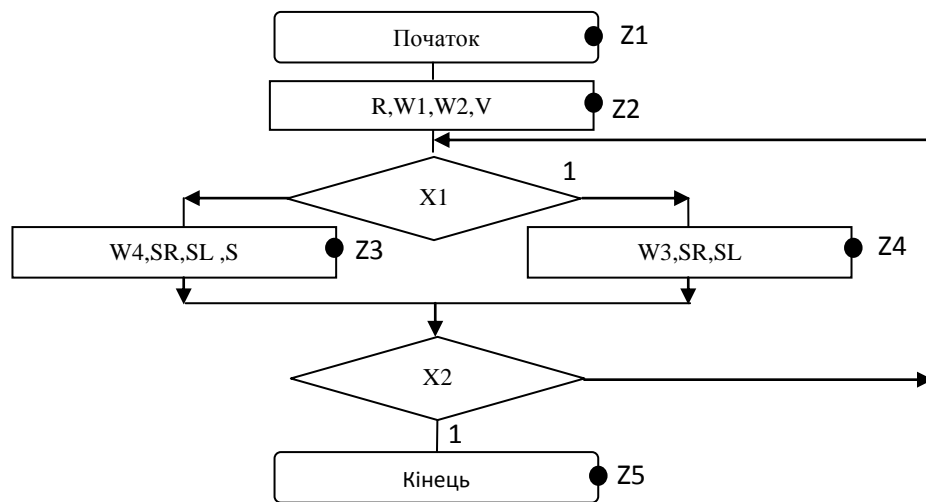


Рисунок 2.9- Закодований мікроалгоритм

2.2.7 Граф управляючого автомата Мура з кодами вершин

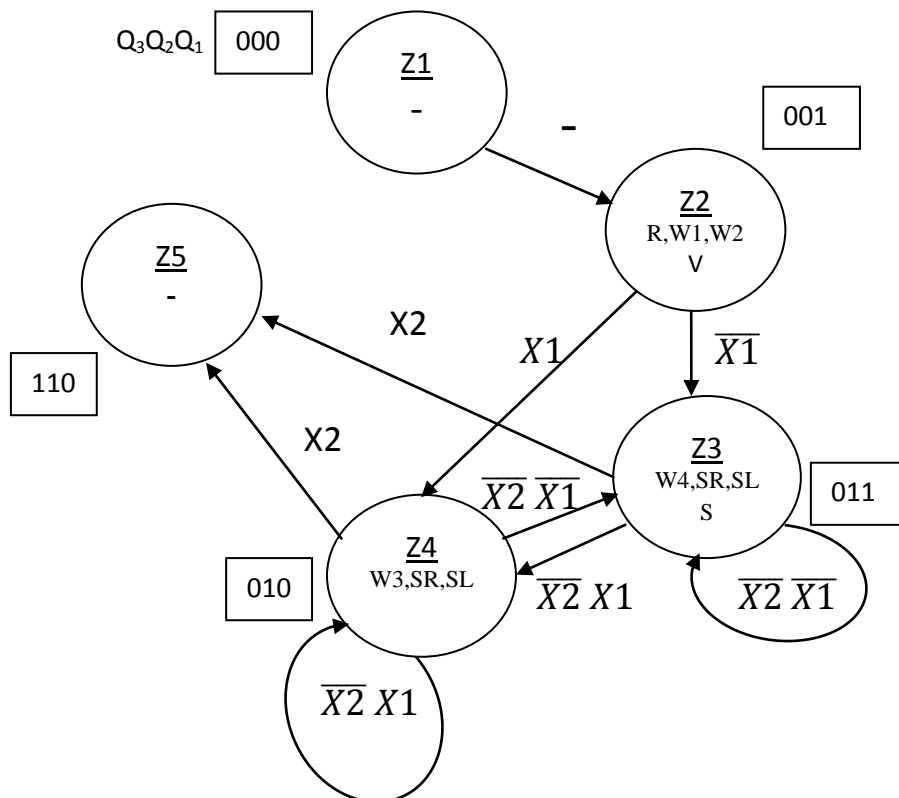


Рисунок 2.10- Граф автомата Мура

2.2.8 Обробка порядків і нормалізація

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

Нормалізація мантиси не потрібна.

$$M_z = , 1101000001010111$$

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

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

Зн.Р	P=+11 ₂	Зн.М	М
0, 0 0 0 0 0 1 1		1, 1 1 0 1 0 0 0 0 0 1 0 1 0 1 1	

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 мантий. Додавання мантий 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 мантий 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 мантий:

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 мантий числа Y, зменшуючи Δ на кожному кроці, доки Δ стане 0.

Таблиця 3.1- Таблиця зсуву мантий на етапі вирівнювання порядків

M_Y	Δ	Мікрооперація
0,101101010101001	11	П.С.
0,010110101010100	10	$M_Y \rightarrow \Delta := \Delta - 1$
0,001011010101010	01	$M_Y \rightarrow \Delta := \Delta - 1$
0,000101101010101	00	$M_Y \rightarrow \Delta := \Delta - 1$

3. Додавання мантий у модифікованому ДК.

Таблиця 3.2-Додавання мантий

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

$$M_x = 11,101101010000111_{ПК} = 11,010010101111001_{ДК}$$

$$M_y = 00,000101101010101_{ПК} = 00,000101101010101_{ДК}$$

$$M_z = 11,110010111011100_{ПК} = 11,001101000100111_{ДК}$$

4. Нормалізація результату (В ПК).

3.2 Операційна схема

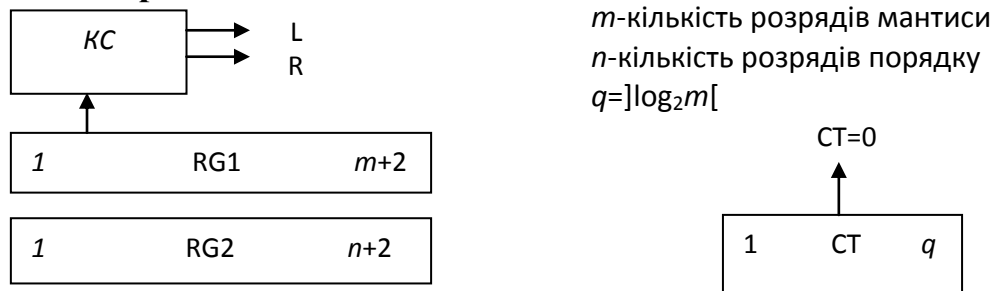


Рисунок 3.1-Операційна схема

Виконаємо синтез КС для визначення порушення нормалізації.

Таблиця 3.3-Визначення порушення нормалізації

Розряди регістру RG1			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

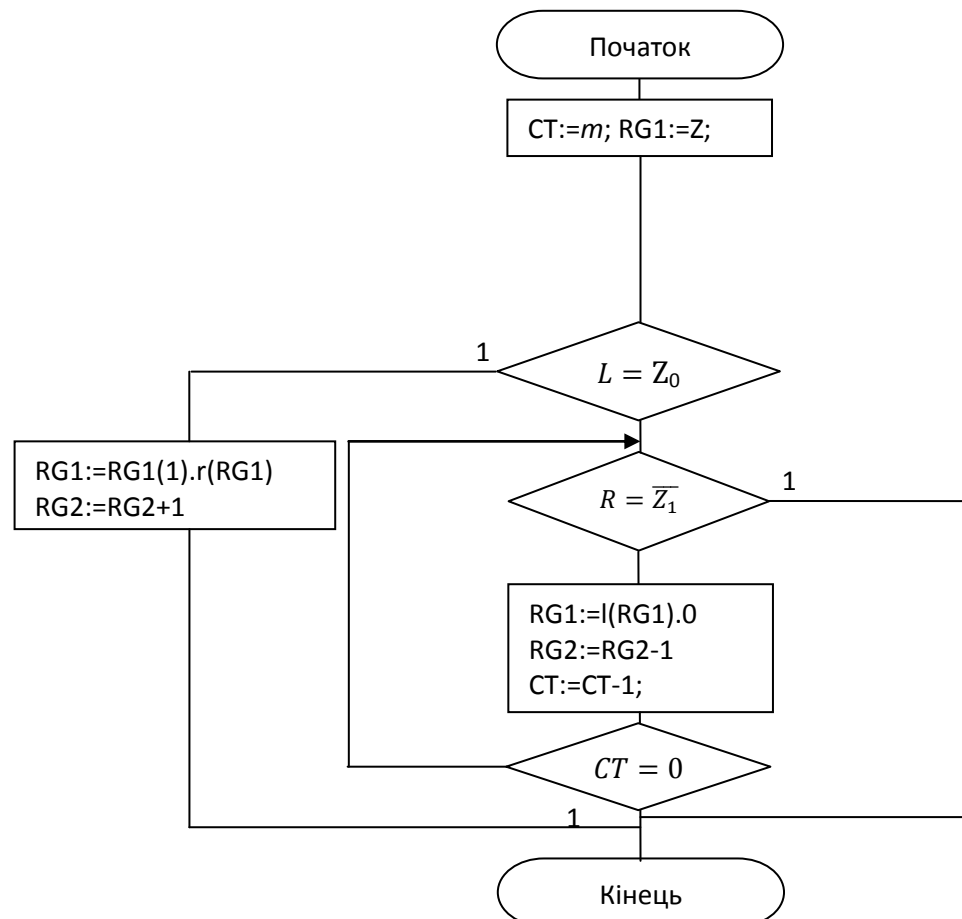


Рисунок 3.2-Змістовний мікроалгоритм

3.4 Таблиця станів регістрів

Таблиця 3.4- Таблиця станів регістрів

№ такту	RG2	RG1	ЛПН(L)	ППН(R)	СТ	Мікрооперація
ПС	001000	00,101101010101001	0	1	1111	

3.5 Функціональна схема з відображенням управляючих сигналів

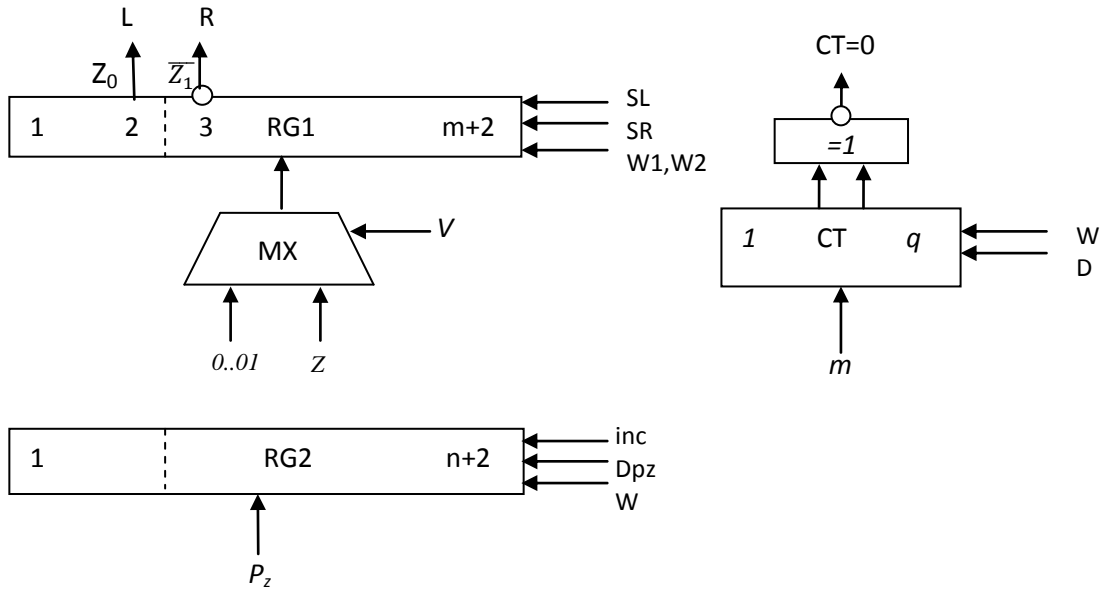


Рисунок 3.3-Функціональна схема

3.6 Закодований мікроалгоритм

Таблиця 3.5- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
CT:=m;	W
RG1:=Z;	W1
$Z'_0 Z_0 := \overline{Z'_0} \overline{Z_0}$	W2
RG1:=RG1(1).r(RG1)	SR
RG2:=RG2+1	inc
RG1:=l(RG1).0	SL
RG2:=RG2-1	D ₂
CT:=CT-1;	D

Таблиця кодування логічних умов	
ЛУ	Позначення
$Z'_0=0$	X1
$L = Z_0$	X2
$R = \overline{Z_1}$	X3
CT = 0	X4

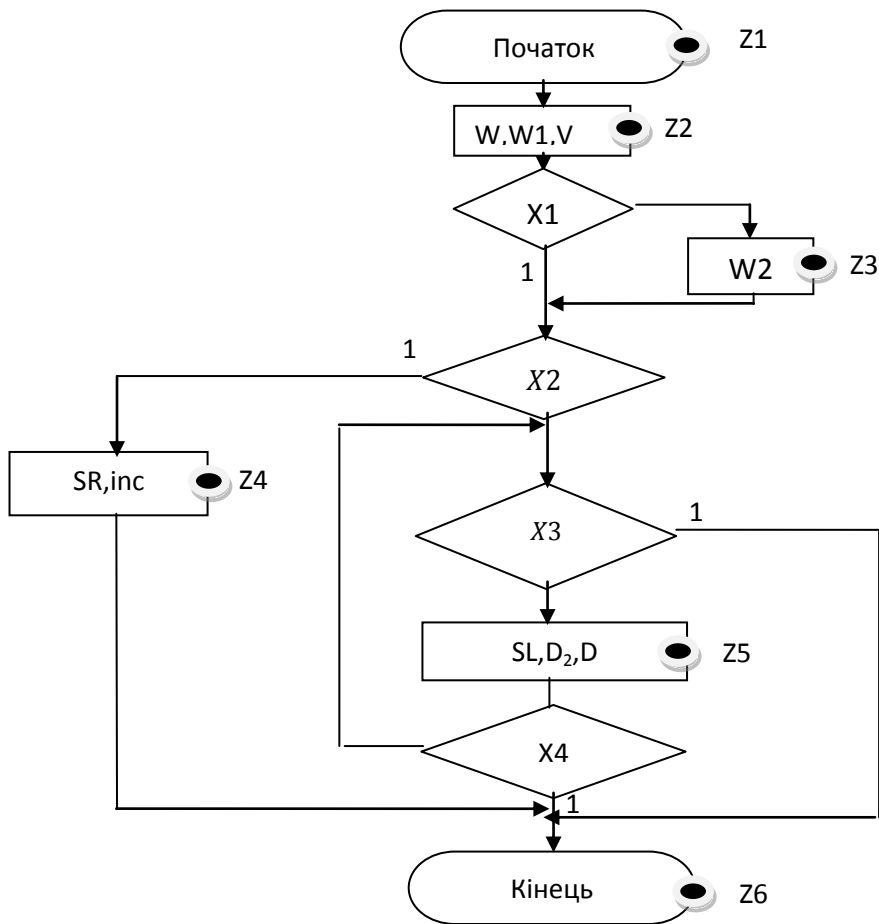


Рисунок 3.4- Закодований мікроалгоритм

3.7 Граф управляючого автомата Мура з кодами вершин

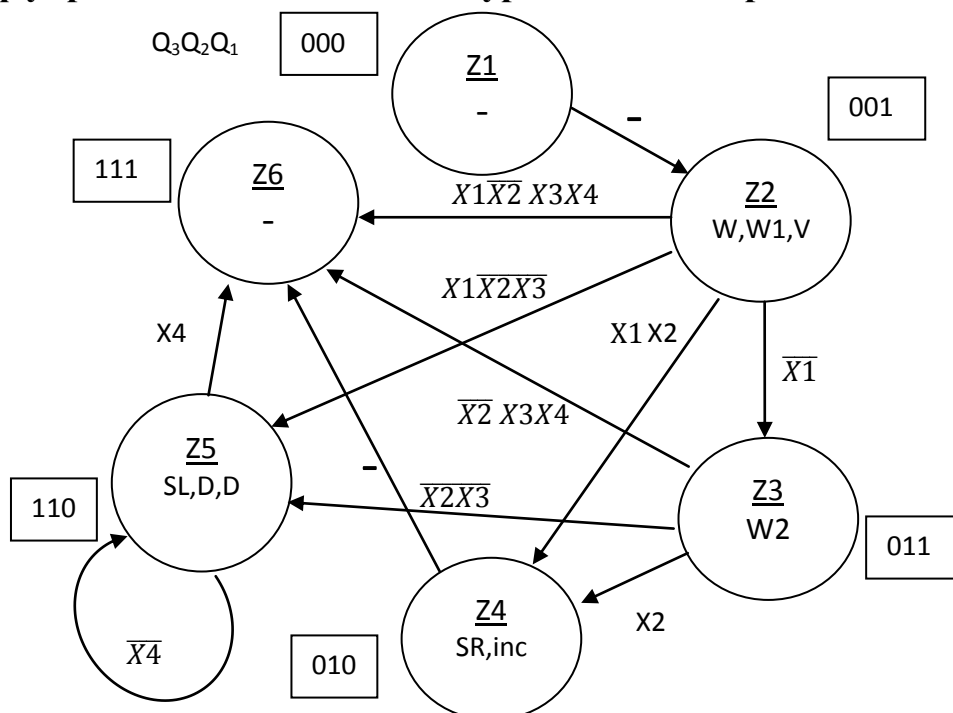


Рисунок 3.5- Граф автомата Мура

3.8 Обробка порядків

PZ=1000

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

Зн.Р								Зн.М								М							
0	0	0	0	1	0	0	0	1	1	1	0	0	1	0	1	1	1	0	1	1	1	0	0

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

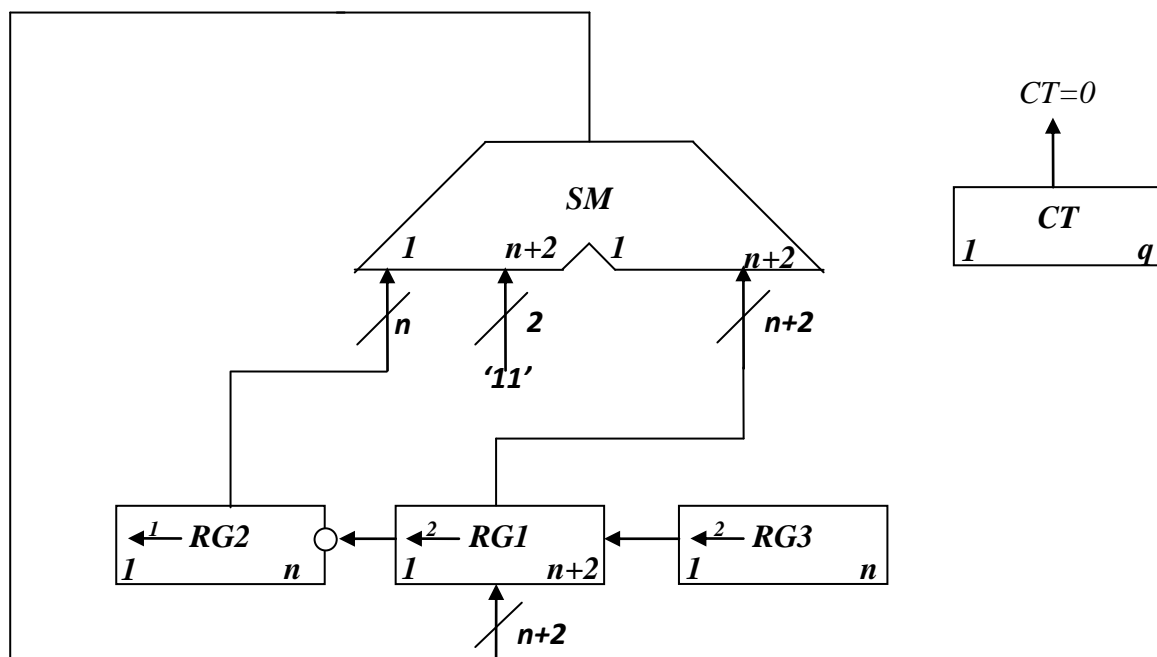


Рисунок 4.1-Операційна схема

4.3 Змістовний мікроалгоритм

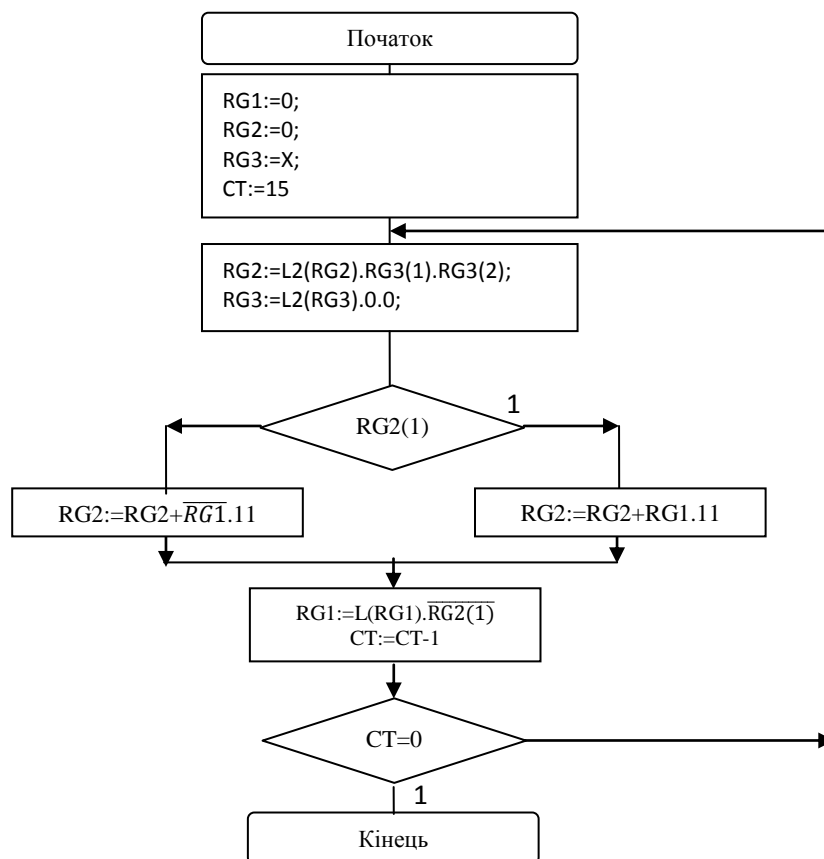


Рисунок 4.2-Змістовний мікроалгоритм

4.4 Таблиця станів реєстрів

Таблиця 4.1- Таблиця станів реєстрів

№ ц.	RG1	RG2	RG3	CT
П.С.	0000000000000000	000000000000000000 000000000000000010	101101010000111	1111
1	0000000000000001	00000000000000010 + <u>1111111111111111</u> 00000000000000001 000000000000000111	010110101000011	1110
2	0000000000000011	000000000000000111 + <u>11111111111111011</u> 00000000000000010 000000000000001001	001011010100001	1101
3	0000000000000110	000000000000001001 + <u>11111111111110011</u> 11111111111111100 11111111111110001	000101101010000	1100
4	0000000000001101	11111111111110001 + <u>00000000000011011</u> 00000000000001100 000000000000110000	000010110101000	1011
5	000000000011010	00000000000110000 + <u>11111111111001011</u> 11111111111111011 11111111111101100	000001011010100	1010
6	000000000110101	11111111111101100 + <u>00000000001101011</u> 00000000001010111 00000000101011111	000000101101010	1001
7	000000001101011	00000000101011111 + <u>11111111100101011</u> 00000000010001010 00000001000101010	000000010110101	1000
8	000000011010111	000000001000101010 + <u>11111111001010011</u> 0000000000111101 00000000111110100	000000001011010	0111
9	000000110101110	00000000111110100 + <u>11111110010100011</u> 11111111010010111 11111101001011100	000000000101101	0110
10	000001101011101	11111101001011100 + <u>00000011010111011</u>	000000000010110	0101

		00000000100010111 00000010001011100		
11	000011010111010	00000010001011100 + <u>11111001010001011</u> 11111011011100111 11011011110011100	000000000001011	0100
12	000110101110100	11101101110011100 + <u>00001101011101011</u> 11111011010000111 11101101000011100	000000000000101	0011
13	001101011101001	11101101000011100 + <u>00011010111010011</u> 00000111111101111 00011111110111100	000000000000010	0010
14	011010111010010	00011111110111100 + <u>11001010001011011</u> 11101010000010111 10101000001011100	000000000000001	0001
15	110101110100101	10101000001011100 + <u>01101011101001011</u> 00010011110100111 01001111010011100	000000000000000	0000

4.5 Функціональна схема з відображенням управляючих сигналів

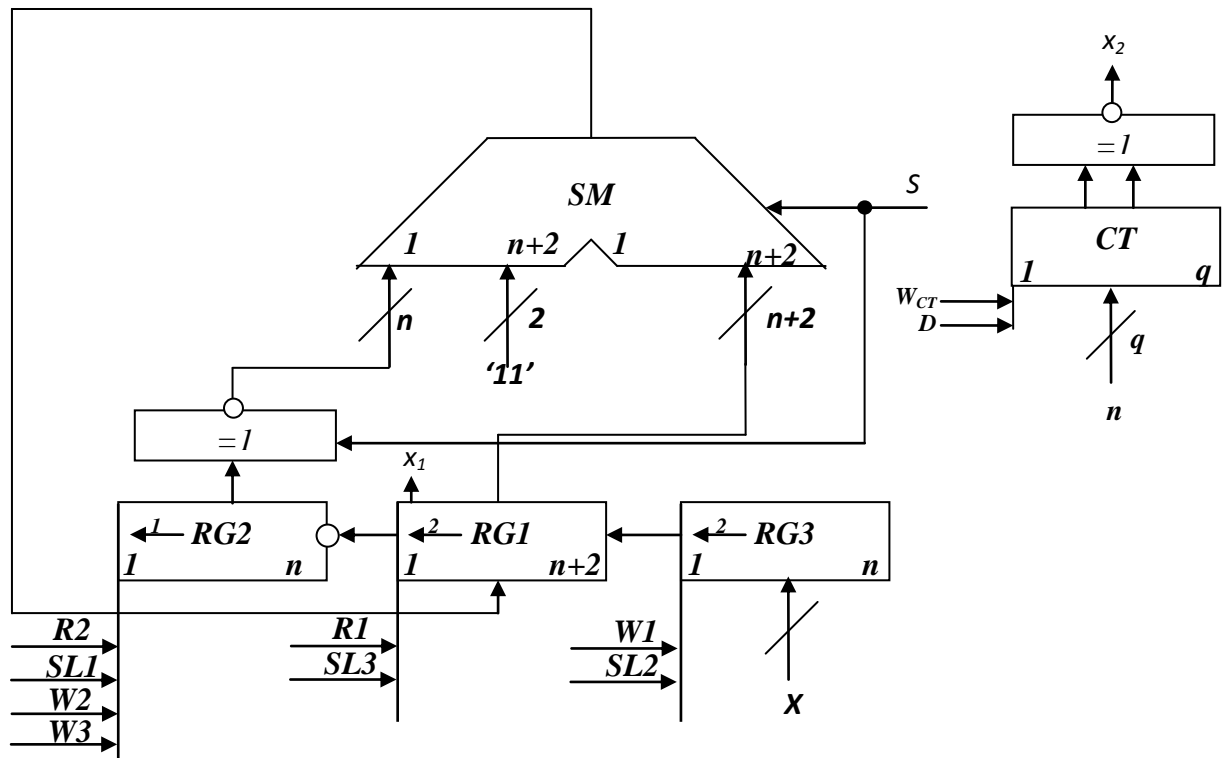


Рисунок 4.3-Функціональна схема

4.6 Закодований мікроалгоритм

Таблиця 4.2- Таблиця кодування мікрооперацій

Таблиця кодування мікрооперацій	
МО	УС
RG1:=0;	R1
RG2:=0;	R2
RG3:=X	W1
CT:=15	W _{CT}
RG2:=RG2+RG1.11	W2
RG2:=RG2+ $\overline{RG1}$.11	W3
RG2:=L2(RG2).RG3(1).RG3(2)	SL1
RG3:=L2(RG3).0.0	SL2
RG1:=L(RG1). $\overline{RG2(1)}$	SL3
CT:=CT-1	D

Таблиця кодування логічних умов	
ЛЮ	Позначення
RG2(1)	X1
CT=0	X2

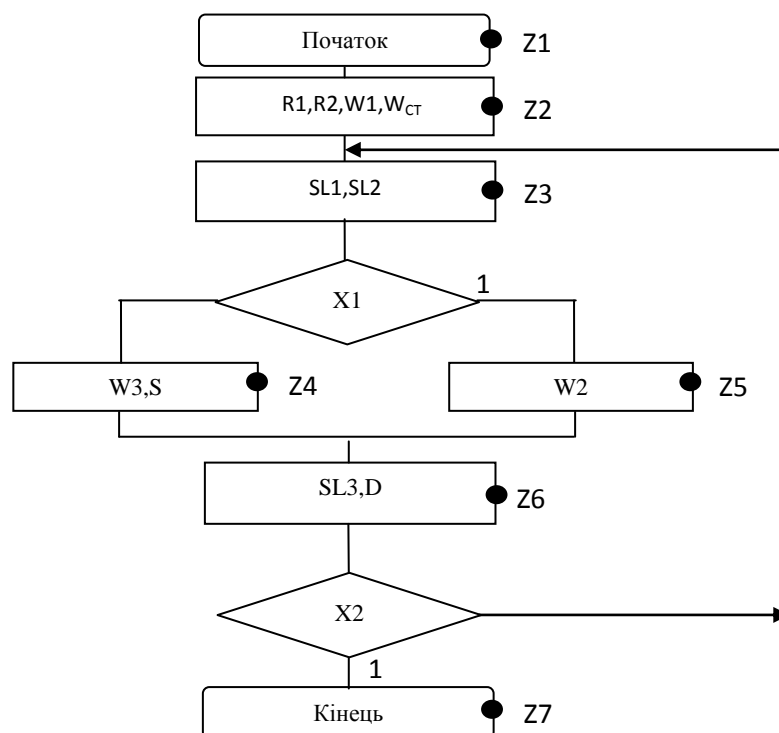


Рисунок 4.4- Закодований мікроалгоритм

4.7 Граф управляющего автомата Мура з кодами вершин

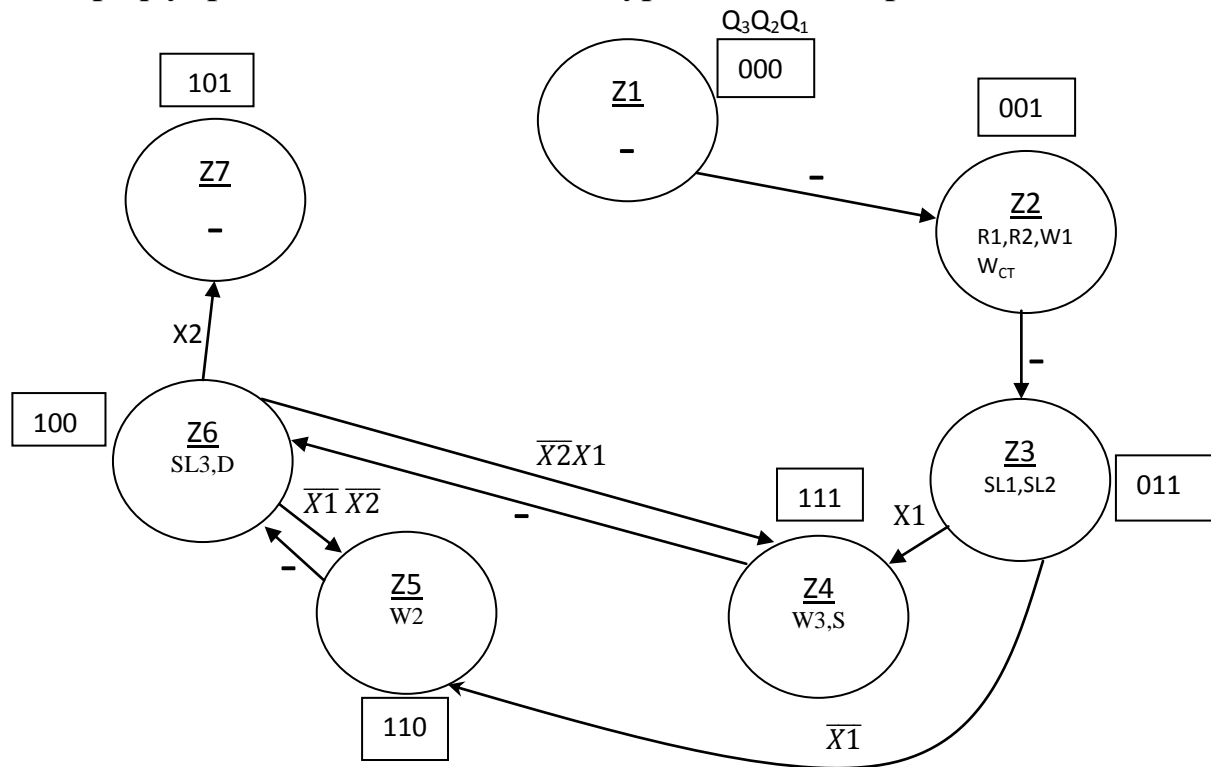


Рисунок 4.5- Граф автомата Мура

4.8 Обробка порядків

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

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

Зн.Р								Зн.М								М							
0.	0	0	0	0	1	0	0	0,	1	1	0	1	0	1	1	1	0	1	1	1	1	0	0